



Universitat de Girona

# ENTORNO VIRTUAL PARA EL APRENDIZAJE Y LA EVALUACIÓN AUTOMÁTICA EN BASES DE DATOS

**Josep SOLER MASÓ**

**ISBN: 978-84-694-0260-3**

**Dipòsit legal: GI-I453-2010**

<http://www.tdx.cat/TDX-1118110-115911>

**ADVERTIMENT.** La consulta d'aquesta tesi queda condicionada a l'acceptació de les següents condicions d'ús: La difusió d'aquesta tesi per mitjà del servei TDX ([www.tesisenxarxa.net](http://www.tesisenxarxa.net)) ha estat autoritzada pels titulars dels drets de propietat intel·lectual únicament per a usos privats emmarcats en activitats d'investigació i docència. No s'autoritza la seva reproducció amb finalitats de lucre ni la seva difusió i posada a disposició des d'un lloc aliè al servei TDX. No s'autoritza la presentació del seu contingut en una finestra o marc aliè a TDX (framing). Aquesta reserva de drets afecta tant al resum de presentació de la tesi com als seus continguts. En la utilització o cita de parts de la tesi és obligat indicar el nom de la persona autora.

**ADVERTENCIA.** La consulta de esta tesis queda condicionada a la aceptación de las siguientes condiciones de uso: La difusión de esta tesis por medio del servicio TDR ([www.tesisenred.net](http://www.tesisenred.net)) ha sido autorizada por los titulares de los derechos de propiedad intelectual únicamente para usos privados enmarcados en actividades de investigación y docencia. No se autoriza su reproducción con finalidades de lucro ni su difusión y puesta a disposición desde un sitio ajeno al servicio TDR. No se autoriza la presentación de su contenido en una ventana o marco ajeno a TDR (framing). Esta reserva de derechos afecta tanto al resumen de presentación de la tesis como a sus contenidos. En la utilización o cita de partes de la tesis es obligado indicar el nombre de la persona autora.

**WARNING.** On having consulted this thesis you're accepting the following use conditions: Spreading this thesis by the TDX ([www.tesisenxarxa.net](http://www.tesisenxarxa.net)) service has been authorized by the titular of the intellectual property rights only for private uses placed in investigation and teaching activities. Reproduction with lucrative aims is not authorized neither its spreading and availability from a site foreign to the TDX service. Introducing its content in a window or frame foreign to the TDX service is not authorized (framing). This rights affect to the presentation summary of the thesis as well as to its contents. In the using or citation of parts of the thesis it's obliged to indicate the name of the author.



TESIS DOCTORAL

**ENTORNO VIRTUAL PARA EL  
APRENDIZAJE Y LA EVALUACIÓN  
AUTOMÁTICA EN BASES DE DATOS**

JOSEP SOLER MASÓ

2010





TESIS DOCTORAL

**ENTORNO VIRTUAL PARA EL  
APRENDIZAJE Y LA EVALUACIÓN  
AUTOMÁTICA EN BASES DE DATOS**

JOSEP SOLER MASÓ

2010

Programa de Doctorado en Tecnología

Dirigida por: RAMÓN FABREGAT GESA

Memoria presentada para optar al título de doctor/a por la Universidad de Girona



Per l'Imma  
sense ella res hauria estat possible

Per la Júlia, la Marta i l'Anna  
tot esperant que segueixin el nostre camí



## **AGRAÏMENTS**

Molt especialment per en Jordi Poch i en Ferran Prados, sense ells dos el projecte ACME no seria el que és.

Per en Josep Daunis (Pepus), per tota l'ajuda rebuda en els tractaments estadístics.

Per en Joan Surrell i l'Imma Boada, per la correcció i puntuació de molts exercicis.

Per en Jordi Conesa, pels molts bons comentaris i observacions sobre la tesi.

Per la Telvys Duliep, per la seva acurada correcció.

Per en Ramón Fabregat, per totes les seves aportacions i el suport rebut.

I finalment a tots aquells que d'alguna manera o altra heu aportat quelcom al projecte ACME.

Moltes gràcies a tots !



## ÍNDICE DE FIGURAS

Figura 1.1 Marco de trabajo de la tesis.....	14
Figura 2.1 Marco de la tesis. Evaluación.....	20
Figura 2.2 Taxonomía de Bloom.....	23
Figura 2.3 Taxonomía revisada de Bloom .....	24
Figura 2.4 Marco de la tesis. CBA .....	27
Figura 2.5 Relación entre tecnologías de e-learning .....	28
Figura 2.6 Niveles superiores Taxonomía de Bloom .....	33
Figura 2.7 Marco de la tesis. Bases de datos .....	40
Figura 2.8 Fases y resultados obtenidos en el diseño de bases de datos .....	42
Figura 2.9 Entorno de la tesis .....	47
Figura 2.10 Estructura modular de la plataforma ACME .....	65
Figura 2.11 Tipología de problemas soportados por la plataforma ACME .....	67
Figura 2.12 Actividades soportadas por ACME.....	69
Figura 3.1 Líneas generales de trabajo .....	75
Figura 3.2 Tipología problemas ACME.....	75
Figura 4.1. Notación modelo ER.....	80
Figura 4.2. Diagrama ER-1 del supuesto planteado .....	82
Figura 4.3 Diagrama ER-2 del supuesto planteado .....	82
Figura 4.4 Enunciado e interfaz de entrada de soluciones a un problema ER .....	88
Figura 4.5 Mensajes de error ACME-ER.....	89
Figura 4.6 Proceso de verificación de la solución en ACME-ER.....	90
Figura 4.7 Proceso de retorno de feedback en ACME-ER.....	92
Figura 4.8 Gráficas del modelo de puntuación obtenido.....	94
Figura 4.9 Interfaz de seguimiento del profesor.....	97
Figura 5.1 Diagrama ER al supuesto típico de una empresa.....	103
Figura 5.2 Diagrama de clases UML del mismo supuesto anterior .....	103
Figura 5.3 Interfaz entrada diagramas de clase .....	109
Figura 5.4 Pestañas interfaz entrada diagramas de clase.....	111
Figura 5.5 Mensajes de error ACME-DC.....	112
Figura 5.6 Resolución del problema ejemplo.....	112
Figura 5.7 Proceso de verificación de la solución en ACME-DC.....	113
Figura 5.8 Proceso de retorno de feedback en ACME-DC .....	115
Figura 6.1 Ejemplo de diagrama ER del supuesto planteado .....	122
Figura 6.2 Esquema de la base de datos del supuesto planteado.....	122
Figura 6.3 Interfaz entrada esquemas BD relacionales .....	128
Figura 6.4 Visualización de tablas entradas .....	129
Figura 6.5 Visualización mensajes de error ACME-EBD.....	129
Figura 6.6 Proceso de verificación de la solución en ACME-EBD .....	131
Figura 6.7 Proceso de retorno de feedback en ACME-EBD.....	131
Figura 6.8 Gráficas del modelo obtenido .....	133
Figura 6.9 Contraste de normalidad .....	133
Figura 7.1 Visualización ejercicio normalización e interfaz de entrada.....	144
Figura 8.1 Proceso de optimización y generación de código de una sentencia SQL ...	148
Figura 8.2 Estrategia asignación consultas álgebra relacional .....	155
Figura 8.3 Interfaz entrada soluciones a problemas álgebra relacional.....	156

Figura 8.4 Interfaz mensajes módulo ACME-AR.....	157
Figura 8.5 Proceso de corrección en ACME-AR.....	158
Figura 9.1 Interfaz entrada sentencias SQL.....	170
Figura 9.2 Interfaz visualización tablas en ACME-SQL.....	171
Figura 9.3 Zonas interfaz visualización resultados sentencias SQL.....	171
Figura 9.4 Proceso de corrección en ACME-SQL.....	172
Figura 9.5 Arquitectura del sistema ACME-SQL.....	173
Figura 10.1 Interfaz de seguimiento del profesor.....	179
Figura 10.2 Box-plot por el total de los alumnos de los cuatro cursos.....	182
Figura 10.3 Gráficas encuestas ACME-DB.....	184

## ÍNDICE DE TABLAS

Tabla 2.1 Descripción de los niveles de conocimiento según la taxonomía de Bloom..	25
Tabla 2.2 Criterios para una buena evaluación aplicados al CBA .....	31
Tabla 2.3 Comparativa de herramientas de aprendizaje de diagramas ER .....	50
Tabla 2.4 Comparativa de herramientas de aprendizaje de diagramas de clase.....	53
Tabla 2.5 Comparativa de herramientas de aprendizaje de esquemas de BD .....	54
Tabla 2.6 Comparativa de herramientas de aprendizaje de la normalización de BD .....	56
Tabla 2.7 Comparativa de herramientas de aprendizaje de álgebra relacional .....	58
Tabla 2.8 Comparativa de herramientas de aprendizaje de SQL .....	61
Tabla 4.1 Causas y mensajes de error del módulo ACME-ER .....	91
Tabla 4.2 Utilización del entorno ACME-ER .....	96
Tabla 4.3 Valoración sobre los ejercicios ACME-ER leídos .....	96
Tabla 4.4 Tipología de errores en función del número de intento.....	98
Tabla 5.1 Causas y mensajes de error del módulo ACME-DC .....	115
Tabla 5.2 Resultados obtenidos con el entorno ACME-DC.....	117
Tabla 5.3 Problemas resueltos con ACME-DC.....	118
Tabla 6.1 Utilización del entorno ACME-EBD .....	135
Tabla 6.2 Valoración sobre los ejercicios ACME-EBD leídos .....	135
Tabla 6.3 Tipología de errores. Problemas Grupo 1.....	136
Tabla 6.4 Tipología de errores. Problemas Grupo 2.....	136
Tabla 6.5 Tipología de errores. Problemas Grupo 3.....	136
Tabla 7.1 Utilización del entorno ACME-NOR.....	145
Tabla 7.2 Valoración sobre los ejercicios ACME-NOR leídos.....	145
Tabla 8.1 Resultados obtenidos con ACME-AR.....	161
Tabla 8.2 Problemas resueltos con ACME-AR.....	161
Tabla 9.1 Resultados obtenidos con ACME-SQL.....	175
Tabla 10.1 Tabla resumen utilización actual de ACME-DB en la asignatura de IFBD	177
Tabla 10.2 Tabla resumen utilización actual de ACME-DB en la asignatura de BD ..	178
Tabla 10.3 Información obtenida del uso de ACME-DB.....	181
Tabla 10.4 Resultados académicos.....	181
Tabla 10.5 Medias y desviaciones estándar .....	182
Tabla 10.6 Test de Levene y valores de p .....	182
Tabla 10.7 Test ANOVA. Tabla de los cuatro cursos .....	183
Tabla 10.8 Resumen encuesta ACME-DB .....	183
Tabla 10.9 Aspectos positivos de ACME-DB según encuestas .....	185
Tabla 10.10 Aspectos a mejorar de ACME-DB según encuestas .....	185
Tabla 11.1 Características del entorno ACME-DB .....	189
Tabla 11.2 Características de las herramientas de diagramas ER .....	189
Tabla 11.3 Características de las herramientas de diagramas de clase.....	190
Tabla 11.4 Características de las herramientas de esquemas de BD .....	190
Tabla 11.5 Características de las herramientas de diagramas de clase.....	191
Tabla 11.6 Características de las herramientas de álgebra relacional .....	191
Tabla 11.7 Características de las herramientas de SQL .....	192
Tabla 11.8 Tabla previsión utilización de ACME-DB en la nueva asignatura de BD .	193



# ÍNDICE

CAPÍTULO 1. INTRODUCCIÓN Y OBJETIVOS.....	13
1.1 INTRODUCCIÓN.....	13
1.2 MOTIVACIÓN .....	14
1.3 OBJETIVOS.....	14
1.4 CONTRIBUCIONES .....	15
1.5 ESTRUCTURA DE LA TESIS.....	17
CAPÍTULO 2. ANTECEDENTES .....	19
2.1 EVALUACIÓN.....	19
2.1.1 Definición y clasificación.....	19
2.1.2 Evaluación formativa y sumativa.....	21
2.1.3 Ventajas e inconvenientes de la evaluación formativa y sumativa. ....	22
2.1.4 Evaluación y taxonomía de Bloom.....	23
2.1.5 Evaluación y tecnología.....	26
2.1.6 Resumen .....	26
2.2 COMPUTER BASED ASSESSMENT.....	27
2.2.1 Definición.....	27
2.2.2 Clasificación de los sistemas Computer Based Assessment .....	29
2.2.3 Beneficios del Computer Based Assessment .....	30
2.2.4 Limitaciones del Computer Based Assessment.....	32
2.2.5 Evaluación de los niveles cognitivos superiores utilizando CBA .....	33
2.2.6 Entornos CBA en Computer Science .....	34
2.2.7 Puntuación automática.....	36
2.2.8 Resumen .....	39
2.3 DOCENCIA EN BASES DE DATOS .....	40
2.3.1 Contenido y metodología de una asignatura de bases de datos.....	40
2.3.2 Diseño de bases de datos .....	41
2.3.3 Consultas en bases de datos.....	43
2.3.4 Innovación y mejoras en la enseñanza de Bases de datos .....	44
2.3.5 Resumen .....	46
2.4 ESTADO DEL ARTE EN HERRAMIENTAS PARA EL APRENDIZAJE Y EVALUACIÓN EN MATERIA DE BD .....	47
2.4.1 Correctores de diagramas entidad / relación.....	48
2.4.2 Correctores de diagramas de clase.....	51
2.4.3 Correctores de esquemas de bases de datos relacionales .....	53
2.4.4 Correctores para la normalización de esquemas de BD. ....	54
2.4.5 Correctores de consultas de álgebra relacional.....	56
2.4.6 Correctores de sentencias SQL.....	58
2.4.7 Integración de entornos de aprendizaje en materia de BD. ....	61
2.4.8 Resumen .....	62
2.5 LA PLATAFORMA ACME .....	63
2.5.1 Características generales de la plataforma ACME.....	63
2.5.2 Tipología de usuarios .....	64
2.5.3 Estructura y funcionamiento de la plataforma ACME.....	65
2.5.4 Tipología de problemas y actividades soportadas por ACME .....	66
2.5.5 Aportaciones a la innovación docente de la plataforma ACME .....	69
2.5.6 Resumen .....	70
2.6 VALORACIÓN FINAL .....	71

CAPÍTULO 3. LÍNEAS GENERALES DE NUESTRA PROPUESTA .....	73
3.1 MARCO DE TRABAJO .....	73
3.2 NUESTRA PROPUESTA .....	74
3.3 METODOLOGÍA DE TRABAJO .....	76
CAPÍTULO 4. MÓDULO DE DIAGRAMAS ENTIDAD/ RELACIÓN .....	79
4.1 INTRODUCCIÓN .....	79
4.2 DESCRIPCIÓN DEL MODELO ENTIDAD/RELACIÓN .....	80
4.3 NECESIDADES Y PROBLEMÁTICA EN LA ENSEÑANZA DEL DISEÑO CONCEPTUAL .....	83
4.4 CORRECCIÓN AUTOMÁTICA DE DIAGRAMAS ER .....	83
4.5.1 Requisitos del sistema ACME-ER .....	84
4.5.2 Análisis inicial de ACME-ER .....	85
4.5.3 Estructura de un problema tipo ERD .....	86
4.5.4 Diseño de la interfaz ERD .....	87
4.5.5 El módulo corrector ACME-ER .....	89
4.5.6 Puntuación automática de los diagramas ERD .....	92
4.6 EVALUACIÓN DEL MÓDULO .....	94
4.6.1 Preparación de ejercicios ER .....	95
4.6.2 Actividades realizadas y resultados obtenidos .....	95
4.7 APORTACIONES Y CONCLUSIONES .....	99
CAPÍTULO 5. MÓDULO CORRECTOR DE DIAGRAMAS DE CLASE .....	101
5.1 INTRODUCCIÓN .....	101
5.2 DESCRIPCIÓN DE UN DIAGRAMA DE CLASES .....	102
5.3 NECESIDADES Y PROBLEMÁTICA DE LA ENSEÑANZA DEL MODELADO CONCEPTUAL CON DIAGRAMAS DE CLASE .....	104
5.4 CORRECCIÓN AUTOMÁTICA DE DIAGRAMAS DE CLASE .....	105
5.5 ACME-DC: MÓDULO DE DIAGRAMAS DE CLASE .....	106
5.5.1 Requisitos del sistema ACME-DC .....	106
5.5.2 Análisis inicial de ACME-DC .....	107
5.5.3 Estructura de un problema diagrama de clases .....	107
5.5.4 Diseño de la interfaz ACME-DC .....	109
5.5.5 El módulo corrector ACME-DC .....	113
5.5.6 Puntuación automática de los diagramas de clase .....	116
5.6 EVALUACIÓN DEL MÓDULO .....	117
5.6.1 Preparación de ejercicios de diagramas de clases .....	117
5.6.2 Actividades realizadas y resultados obtenidos .....	117
5.7 APORTACIONES Y CONCLUSIONES .....	118
CAPÍTULO 6. MÓDULO ESQUEMAS BASES DE DATOS RELACIONALES ....	119
6.1 INTRODUCCIÓN .....	119
6.2 DESCRIPCIÓN DEL MODELO RELACIONAL .....	120
6.3 NECESIDADES Y PROBLEMÁTICA EN LA OBTENCIÓN DE ESQUEMAS DE BASES DE DATOS .....	123
6.4 CORRECCIÓN AUTOMÁTICA DE ESQUEMAS DE BASES DE DATOS RELACIONALES .....	124
6.5 ACME-EBD: MÓDULO DE ESQUEMAS DE BASES DE DATOS RELACIONALES .....	124
6.5.1 Requisitos del sistema ACME-EBD .....	125

6.5.2	Análisis inicial de ACME-EBD .....	125
6.5.3	Estructura de un problema de esquemas de BD .....	126
6.5.4	Diseño de la interfaz ACME-EBD .....	127
6.5.5	El módulo corrector ACME-EBD .....	130
6.5.6	Puntuación automática de los esquemas de bases de datos .....	132
6.6	EVALUACIÓN DEL MÓDULO .....	134
6.6.1	Preparación de ejercicios de EBD relacionales .....	134
6.6.2	Actividades realizadas y resultados obtenidos .....	134
6.7	APORTACIONES Y CONCLUSIONES .....	137
CAPÍTULO 7. MÓDULO NORMALIZACIÓN DE BASES DE DATOS.....		139
7.1	INTRODUCCIÓN.....	139
7.2	NORMALIZACIÓN DE BD RELACIONALES .....	140
7.3	NECESIDADES Y PROBLEMÁTICA EN LA ENSEÑANZA DE LA NORMALIZACIÓN DE BASES DE DATOS.....	142
7.4	CORRECCIÓN AUTOMÁTICA DE EJERCICIOS DE NORMALIZACIÓN	143
7.5	ACME-NOR: MÓDULO DE NORMALIZACIÓN .....	143
7.6	EVALUACIÓN DEL MÓDULO.....	144
7.7	APORTACIONES Y CONCLUSIONES .....	146
CAPÍTULO 8. MÓDULO DE CONSULTAS EN ÁLGEBRA RELACIONAL .....		147
8.1	INTRODUCCIÓN.....	147
8.2	EL ÁLGEBRA RELACIONAL.....	148
8.3	NECESIDADES Y PROBLEMÁTICA EN LA ENSEÑANZA DEL ÀLGEBRA RELACIONAL.....	151
8.4	CORRECCIÓN AUTOMÁTICA DE CONSULTAS DE ÀLGEBRA RELACIONAL.....	152
8.5	ACME-AR: MÓDULO CORRECTOR DE CONSULTAS DE ÀLGEBRA RELACIONAL.....	152
8.5.1	Requisitos del sistema ACME-AR.....	152
8.5.2	Análisis inicial de ACME-AR.....	153
8.5.3	Estructura de un problema de álgebra relacional.....	154
8.5.4	Diseño de la interfaz ACME-AR .....	155
8.5.5	El módulo corrector ACME-AR.....	157
8.5.6	Puntuación automática de consultas de álgebra relacional.....	160
8.6	EVALUACIÓN DEL MÓDULO.....	160
8.6.1	Preparación de ejercicios de álgebra relacional.....	160
8.6.2	Actividades realizadas y resultados obtenidos .....	161
8.7	APORTACIONES Y CONCLUSIONES .....	161
CAPÍTULO 9. MÓDULO DE SENTENCIAS SQL .....		163
9.1	INTRODUCCIÓN.....	163
9.2	DESCRIPCIÓN DEL LENGUAJE SQL .....	164
9.2.1	Creación, modificación y borrado de tablas .....	164
9.2.2	Inserción, actualización y borrado de filas .....	164
9.2.3	Consultas .....	165
9.3	NECESIDADES Y PROBLEMÁTICA DE LA ENSEÑANZA DE SQL .....	166
9.4	CORRECCIÓN AUTOMÁTICA DE SENTENCIAS SQL .....	167
9.5	ACME-SQL. MÓDULO DE SENTENCIAS SQL.....	167
9.5.1	Requisitos del sistema ACME-SQL.....	167

9.5.2 Análisis inicial de ACME-SQL .....	168
9.5.3 Estructura problema SQL .....	169
9.5.4 Diseño de las interfaces SQL .....	170
9.5.5 El módulo corrector ACME-SQL .....	172
9.5.6 Puntuación automática de sentencias SQL .....	174
9.6 EVALUACIÓN DEL MÓDULO .....	174
9.6.1 Preparación de ejercicios SQL .....	174
9.6.2 Actividades realizadas y resultados obtenidos .....	175
9.7 APORTACIONES Y CONCLUSIONES .....	175
CAPÍTULO 10. RESULTADOS GENERALES .....	177
10.1 USO DE ACME_DB EN LA ASIGNATURA DE BASES DE DATOS.....	177
10.2 RESULTADOS OBTENIDOS EN EL DISEÑO DE BD .....	180
10.3 ENCUESTAS .....	183
CAPÍTULO 11. CONCLUSIONES FINALES Y TRABAJO FUTURO.....	187
11.1 CONCLUSIONES.....	187
11.2 APORTACIONES RESPECTO A LOS ENTORNOS EXISTENTES .....	188
11.3 TRABAJO FUTURO .....	192
11.4 REFLEXIONES FINALES .....	195
BIBLIOGRAFÍA .....	197

## RESUMEN

La enseñanza y evaluación automática a través de un sistema Computer Based Assessment (CBA) requiere de software especializado que se adapte a la tipología de actividades a tratar y evaluar. La mayor parte de estos sistemas realizan la evaluación automática a través de preguntas de respuesta fija (elección múltiple, preguntas tipo test, etc.). Con este tipo de preguntas se pueden evaluar los niveles de conocimiento básicos de la taxonomía de Bloom (conocimiento y comprensión) pero difícilmente se pueden evaluar niveles intermedios (aplicación y análisis) o superiores (síntesis y evaluación). Para estos niveles se necesitan entornos especializados que soporten actividades concretas en las que los alumnos puedan responder libremente sin tener las respuestas prefijadas de antemano.

El diseño de entornos de CBA especializados no es trivial. Deben desarrollarse correctores a medida con un grado de especialización muy elevado para ajustarse a las necesidades de la materia a evaluar. En este trabajo nos hemos centrado en el aprendizaje y evaluación de la materia de Bases de Datos en un entorno universitario. Las competencias en esta materia se logran cuando el alumno es capaz de realizar ciertas actividades que se enmarcan en los niveles de conocimiento intermedios de la taxonomía de Bloom.

Se ha desarrollado un entorno CBA que facilita el aprendizaje y evaluación de los principales temas de una asignatura de bases de datos. Para ello se han analizado las herramientas existentes en cada uno de estos temas (Diagramas Entidad/Relación, diagramas de clases, esquemas de bases de datos relacionales, normalización, consultas en álgebra relacional y lenguaje SQL) y para cada uno de ellos se ha analizado, diseñado e implementado un módulo de corrección y evaluación automática que aporta mejoras respecto a los existentes. Estos módulos se han integrado en un mismo entorno al que hemos llamado ACME-DB.

Este entorno ha sido utilizado como complemento a las clases presenciales en la asignatura de Bases de Datos de segundo curso de Ingeniería Técnica en Informática de Gestión de la Universidad de Girona. Se han comparado los resultados obtenidos con los de los cursos anteriores y se ha observado una mejora en los resultados académicos facilitando al profesor las tareas de corrección y evaluación.



## ABSTRACT

Teaching and automatic evaluation through Computer Based Assessment (CBA) requires of specialized software tools capable to correct different type of activities. Most of current systems only correct fixed-response questions, such as multiple choice or test-questions and can be used only for evaluating lowest levels of Bloom's taxonomy (knowledge and comprehension). The evaluation of intermediate (application and analysis) or highest (synthesis and evaluation) levels requires of more specialized environments able to support more complex activities where no predetermined answers are given.

The design of specialized CBA environments is not trivial. The main difficulty is on the development of the correction strategies. These require a high degree of specialization to satisfy the needs of the area to be evaluated. In this context, we have focused our interest on university database courses where intermediate skill levels of the taxonomy of Bloom need to be evaluated.

In this work, we have analyzed the main topics of a database course (entity/ relationship diagrams, class diagrams, relational database schemas, normalization, relational algebra and SQL) and for each one of them we have reported the state of art on CBA tools. Afterwards, we have proposed a new CBA environment for teaching and learning database. This environment, denoted ACME-DB, is composed of different modules capable to automatically correct and evaluate exercises of main topics of a database course providing improvements over existing tools.

ACME-DB has been used in our university as a complement to the classroom in the database course of the second year diploma in Computer Science of the University of Girona. We have compared the results of using this environment with those of previous years. It has been seen that better academic results are achieved when using the platform. Moreover, improvements on the teacher's correction and evaluation tasks have been obtained.



# CAPÍTULO 1. INTRODUCCIÓN Y OBJETIVOS

## 1.1 INTRODUCCIÓN

En el Espacio Europeo de Educación Superior (EEES) se plantean las diferentes carreras como un conjunto de competencias que el alumno debe adquirir y que los profesores tienen que evaluar. Se propone un nuevo modelo educativo en el cual los planes docentes y las metodologías a usar han de estar basados en el aprendizaje de los estudiantes, en contraposición con la tradicional programación centrada en la enseñanza del profesor. En este nuevo contexto, el profesor deberá llevar a cabo el control de la evolución del aprendizaje del alumno para garantizar así el logro de las competencias a adquirir. Por otra parte, el elevado número de alumnos en las clases hace más difícil el desarrollo de los cursos y más aún si tenemos en cuenta los recursos limitados de las universidades. En este nuevo modelo uno de los puntos clave es la evaluación.

De forma simplificada podemos considerar dos tipos de evaluación: la formativa y la sumativa. La *evaluación sumativa* tiene como objetivo fundamental calificar o certificar el nivel del alumno al terminar un determinado período (parte del curso, curso completo, etc.) [Kni01]. Puede haber varias evaluaciones sumativas, como por ejemplo los exámenes parciales o una única evaluación final. El método de evaluación utilizado debe ser riguroso y fiable ya que será el único elemento de evaluación. La *evaluación formativa* es una forma de evaluación en la que el principal objetivo es ayudar en todo el proceso de aprendizaje [Kni01]. Este tipo de evaluación aporta información a lo largo de todo el proceso enseñanza-aprendizaje. De este modo el profesor puede evaluar el nivel de aprendizaje, el ritmo de trabajo, detectar deficiencias y si es necesario cambiar las actividades propuestas. Por otra parte, al alumno le aporta información válida para su autoevaluación, se le informa sobre sus errores, sobre qué es lo importante y cómo debe afrontar el curso [JMM+04]. La calificación se realiza a partir de lo que el alumno sabe hacer al terminar el período de aprendizaje. Aunque desde el punto de vista pedagógico es mejor la evaluación formativa, este tipo de evaluación supone una mayor dedicación por parte del profesor por la cantidad de pruebas, controles, etc. que debe realizar, dificultando así su aplicación.

Para la evaluación de las competencias en el EEES, encaja mejor la evaluación formativa que la sumativa tradicional. Ahora bien, la evaluación formativa conlleva una gran carga de trabajo adicional al profesor y más en los cursos numerosos. En este contexto los profesores tienen pocas opciones a la hora de evaluar [Gib92], [BR96], [CEA+03]. Una alternativa es utilizar ayudantes o becarios para calificar las pruebas presentadas, con toda la problemática que ello conlleva (coordinación, homogenización de los criterios a seguir, falta de feedback, etc.). Otra alternativa es aplicar las tecnologías de la información y comunicación (TIC) para desarrollar entornos y plataformas educativas que faciliten la evaluación del alumno. Estos sistemas pueden ayudar a los profesores proporcionando mecanismos para automatizar la evaluación del trabajo de los estudiantes. Sin embargo el desarrollo, implementación y puesta en marcha de estos entornos no es trivial ya que requieren de técnicas y métodos especializados que han dado lugar a una nueva línea de investigación en el marco del e-learning, conocida como Computer Based Assessment.

El Computer Based Assessment (CBA) [CE98] [Tsi02] constituye una sección de las tecnologías del e-learning que se distingue por la automatización de todos los procesos que intervienen en la evaluación del alumno. Se define como un sistema integrado y on-line que permite la entrega de materiales y ejercicios, la entrada de soluciones por parte de los estudiantes y un proceso de evaluación y de entrega de comentarios (feedback) totalmente automatizado. En este proceso de evaluación el sistema debe corregir y puntuar automáticamente las soluciones enviadas. La complejidad de estos entornos depende de las actividades que deba evaluar. Por ejemplo, no tiene la misma dificultad la evaluación de una pregunta tipo test que la corrección de un programa informático o un ejercicio que requiera el desarrollo de un diagrama. La mayoría de los entornos CBA actuales están diseñados para la evaluación a través de ejercicios simples, siendo la evaluación de problemas complejos un tema de investigación a tratar.

En esta tesis nos centraremos en el desarrollo de herramientas que nos permitan llevar a cabo la evaluación automatizada de los alumnos en la materia de Bases de Datos (BD). Este tipo de herramientas deben integrar funcionalidades avanzadas que soporten, entre otras, la corrección de diagramas que representan la estructura de la base de datos y las consultas que puedan realizarse sobre la misma.

## 1.2 MOTIVACIÓN

La motivación que nos ha llevado al desarrollo de este trabajo ha sido fruto de nuestra experiencia en la docencia de BD. Dada la importancia de esta materia en los estudios de Informática consideramos que la incorporación de un sistema CBA podría facilitar el aprendizaje de la materia. Existen muchas herramientas que de forma independiente facilitan el aprendizaje de un determinado tema de BD, pero hasta donde llega nuestro conocimiento, desconocemos la existencia de plataformas que integren las herramientas necesarias para llevar a cabo procesos de evaluación (formativos y sumativos) en los principales temas de esta materia. Esta limitación se debe a la complejidad del desarrollo de estas herramientas. Debemos tener en cuenta que un entorno CBA debe permitir la corrección automática de ejercicios, facilitando el feedback oportuno y obteniendo una puntuación que pueda servir para la evaluación sumativa del alumno. Además de la corrección de ejercicios de respuesta fija (tipo test, elección múltiple, etc.) disponibles en la mayoría de plataformas de e-learning, debe soportar la corrección de ejercicios con respuesta libre, donde el alumno desarrollará su respuesta según el tipo de ejercicio. El alumno no deberá escoger una respuesta entre varias, sino que deberá crear su respuesta. Una dificultad añadida será que las respuestas de los ejercicios propuestos en este entorno tienen diferentes representaciones en forma de diagramas, de esquemas de BD, de expresiones matemáticas, sentencias en SQL (Structured Query Language), etc. Evaluar este tipo de ejercicios es necesario para verificar si el alumno ha adquirido las competencias en materia de BD que se sitúan en los niveles cognitivos de aplicación y análisis según la taxonomía de Bloom [Blo56].

## 1.3 OBJETIVOS

El objetivo principal de esta tesis es proponer y desarrollar herramientas web de soporte a la evaluación automatizada en BD que se integrarán en una plataforma web de fácil uso tanto para profesores como estudiantes. Para ello debemos desarrollar técnicas capaces de interpretar y corregir cualquier tipo de problemas, proporcionando feedback al alumno y facilitando la información oportuna para su evaluación automatizada. Centraremos nuestro trabajo en las materias específicas de un curso de bases de datos, siendo uno de nuestros retos la integración de las técnicas y métodos propuestos en una plataforma de e-learning, concretamente en la plataforma ACME (del catalán “Avaluació Continuada i Millora de l’Ensenyament”) [SPB+02][PBJ+03][PBS+05a], permitiendo de esta forma evaluar nuestras propuestas.

Nuestro trabajo combinará tres conceptos básicos: Evaluación, Computer Based Assessment y Enseñanza de Bases de Datos, siendo el resultado final un nuevo entorno de e-learning al que llamaremos ACME-DB (ver Figura 1.1).

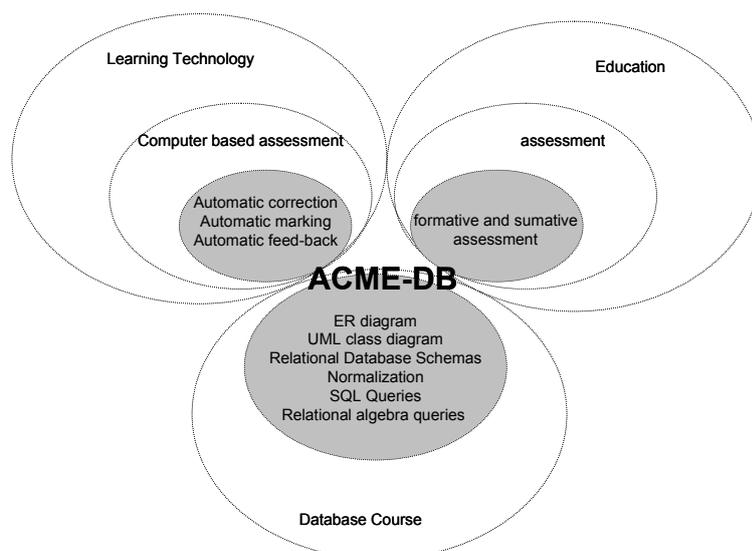


Figura 1.1 Marco de trabajo de la tesis.

ACME-DB permitirá la evaluación automática, es decir corrección, feedback y puntuación automática de ejercicios de los principales temas en materia de BD. Estos temas son:

- **Diseño de bases de datos.** Para ello se necesita disponer de un sistema para la evaluación de diagramas entidad/relación (ERD) y de diagramas de clases para poder realizar el diseño conceptual. Además para el diseño lógico el sistema debe determinar si un esquema de BD relacional es correcto y si está normalizado.
- **Consultas en bases de datos.** El sistema debe permitir la evaluación de sentencias SQL y de consultas en álgebra relacional.

Para alcanzar el objetivo propuesto, para cada tipología de problemas de un curso de BD se han desarrollado las siguientes tareas:

1. **Desarrollo de módulos correctores.** Dentro de un contexto de CBA, estos módulos deberán permitir la corrección automática de ejercicios de los temas previamente citados (diagramas entidad/relación, diagramas de clase, esquemas de BD, normalización, álgebra relacional y sentencias SQL). Esta tarea conlleva:
  - Investigación acerca de herramientas que faciliten el aprendizaje de BD.
  - Análisis, diseño e implementación de los módulos correctores.
2. **Integración en la plataforma ACME.** Para poder utilizar los módulos desarrollados se deben integrar en una plataforma de e-learning y crear una extensa colección de ejercicios. Esta tarea conlleva:
  - Integración en la plataforma ACME de los módulos desarrollados.
  - Desarrollo de una amplia colección de problemas.
3. **Evaluación en un entorno real.** Cada vez que se disponga de un nuevo módulo corrector operativo se testeará en la asignatura de BD. Se pretende que después de cada tema el profesor asigne ejercicios a los alumnos a través de la plataforma ACME-DB. El alumno enviará soluciones a las actividades propuestas que serán automáticamente corregidas por el sistema el cual además ofrecerá el feedback oportuno. Con la información guardada por el sistema, el profesor podrá realizar el seguimiento de forma individualizada de cada alumno permitiéndole detectar deficiencias, corregir errores, etc. Con ello se pretende llevar a cabo la evaluación formativa. Para cada uno de los módulos, esta tarea conlleva:
  - Puesta en marcha y utilización en la asignatura de BD.
  - Seguimiento de su utilización. Remarcar que esta tarea es de especial importancia ya que nos permite registrar los diferentes tipos de errores que cometen los alumnos. Creemos que esta información será de gran valor para diseñar actuaciones que permitan solventarlos.
  - Análisis de los resultados parciales obtenidos.
4. **Valoración final y conclusiones.** Una vez evaluados los diferentes módulos en un entorno real se analizarán los datos obtenidos de la utilización de estas herramientas como complemento a las clases presenciales (blended learning) en materia de bases de datos y se extraerán las conclusiones pertinentes.

## 1.4 CONTRIBUCIONES

En esta sección se presentan de forma resumida las contribuciones que nuestro trabajo aporta respecto a los entornos existentes que se verán en el Capítulo 2. Éstas radican en el desarrollo de un entorno CBA especializado en BD que permite la evaluación formativa y sumativa. Los puntos clave en el proceso de evaluación son la capacidad del entorno de poder realizar la corrección, ofrecer feedback y calcular la puntuación de forma totalmente automática de las distintas actividades. Teniendo en cuenta que las actividades son complejas y requiere respuestas libres (diagramas, expresiones, etc), las contribuciones aportadas se pueden desglosar en los siguientes apartados:

- **Entorno web, especializado en bases de datos.** Para el uso de la mayoría de entornos especializados en BD es necesario descargar e instalar una determinada aplicación en el ordenador donde se utilizará. Por otra parte, los correctores automáticos existentes suelen ser trabajos aislados, a menudo no enmarcados en la docencia de BD, sino que aparecen como aplicaciones en medio de otro entorno de investigación. En este sentido encontramos:
  - Trabajos de investigación en la corrección de diagramas, en donde se tratan como casos particulares los diagramas utilizados en BD.
  - Trabajos referentes a la interpretación del lenguaje natural que son utilizados en aplicaciones de corrección automática.
  - Aplicaciones que de forma individual se utilizan en un tema de un curso de BD.
  - Aplicaciones que necesitan un software adicional para realizar parte de la actividad. Por ejemplo un sistema de gestión de bases de datos (SGBD), software específico para la realización de diagramas, etc.

En este sentido nosotros proponemos un único entorno web que aglutine todas las actividades que se puedan corregir de forma automática en materia de BD, sin más requisitos que una conexión Internet y un navegador cualquiera. De esta forma solventamos una de las limitaciones de las herramientas actuales que no engloban todos estos correctores.

- **Corrección automática de problemas complejos.** La corrección automática de ejercicios y el correspondiente feedback automatizan la evaluación formativa, permitiendo al profesor conocer la situación de cada alumno, los errores más frecuentes, el seguimiento del grupo, etc. Por otro lado el alumno sabe su nivel de aprendizaje, se le permite corregir los errores y volver a enviar una nueva solución. Aunque la mayoría de entornos desarrollados facilitan el aprendizaje en materia de BD, son pocos los que corrigen de forma automática las soluciones enviadas por los alumnos.

Esta es una de las aportaciones clave de nuestro trabajo: ofrecemos al alumno un entorno en el que debe resolver actividades y entrar la solución que será corregida on-line, facilitando el feedback adecuado a la solución del alumno. Con esta característica el alumno dispondrá de un tutor virtual disponible a todas horas, que le permite saber el estado de la resolución de sus actividades y, en caso de error, recibir las indicaciones que le guíen hacia la solución correcta.

- **Puntuación automática.** En la mayoría de plataformas de e-learning la corrección es binaria (correcto/incorrecto). Esta estrategia es eficaz en el caso de problemas de respuesta fija como por ejemplo los de elección múltiple. En el caso de problemas complejos, como es el nuestro, los ejercicios que debe realizar el alumno pueden ser muy extensos y un pequeño error es asumible. Por lo tanto la opción correcto/incorrecto no nos sirve y se deben buscar métodos que califiquen un ejercicio y que la nota obtenida sea muy parecida a la que pondría un profesor.

En este sentido, nuestra contribución se basa en un sistema que facilita una nota entre 0 y 10 en función de los errores cometidos. Los parámetros para la obtención de esta nota deberán minimizar las diferencias con las notas obtenidas de forma manual por el profesor. Actualmente son muy pocos los correctores de esta temática que faciliten una puntuación automática.

- **Evaluación automática.** A partir de las actividades desarrolladas en cada tema y puntuadas de forma automática, se debe facilitar al profesor la evaluación final del alumno de forma totalmente automática.

Nuestra contribución es que a través de la corrección y la puntuación automática, aportamos al profesor una valiosa herramienta que le reducirá considerablemente la tediosa labor de corrección y evaluación de los ejercicios, reduciendo drásticamente el tiempo dedicado a estas labores.

## 1.5 ESTRUCTURA DE LA TESIS

Esta tesis, además de este capítulo introductorio, está formada por diez capítulos más.

En el capítulo 2 se presentan los antecedentes necesarios para situar el contexto de la tesis. Se hace un análisis de los aspectos más importantes del e-assessment y del **Computer Based Assessment (CBA)**, centrándonos básicamente en los entornos que permiten una corrección automática de ejercicios de respuesta libre. A continuación se presentan los temas más relevantes en **docencia de bases de datos** y las metodologías que suelen utilizarse, haciendo hincapié en los métodos innovadores que se están utilizando. Posteriormente se desarrolla el estado del arte en materia de **correctores automáticos** en los diferentes temas de bases de datos que se van a desarrollar en esta tesis. Finalmente se describe la **plataforma ACME** sobre la cual se van a integrar y utilizar en un contexto real los módulos correctores desarrollados.

En el capítulo 3 se presenta de forma simplificada una visión global de nuestra propuesta y se describen las líneas generales de la misma.

En los siguientes capítulos se describen los módulos correctores correspondientes a: diagramas entidad/relación (capítulo 4), diagramas de clase (capítulo 5), esquemas de BD relacionales (capítulo 6), normalización (capítulo 7), álgebra relacional (capítulo 8) y SQL (capítulo 9). En cada capítulo se describe el problema a tratar, se analizan posibles soluciones y finalmente se presenta la solución propuesta, describiendo el módulo corrector desarrollado y las estrategias de corrección usadas. La redacción de cada capítulo se ha realizado de forma que pueda leerse de manera independiente aunque esto implique que alguna parte pueda resultar repetitiva. Una vez descrito el corrector se describe su puesta en marcha en un entorno real y se detallan las actividades y ejercicios realizados por los alumnos en la asignatura de Bases de Datos a través de la plataforma ACME-DB. Finalmente se realiza una evaluación de los resultados obtenidos y las conclusiones alcanzadas.

En el capítulo 10 se realiza una evaluación global de los resultados obtenidos. Finalmente en el capítulo 11 se presentan las conclusiones finales y el trabajo futuro.



## CAPÍTULO 2. ANTECEDENTES

Las plataformas de e-learning facilitan la creación y distribución de los materiales docentes. La tendencia actual es ampliar y especializar más sus funcionalidades, especialmente en el área de Computer Science [Rob08]. Se pretende llegar a altos grados de automatización de todo el proceso enseñanza/aprendizaje siendo la evaluación automática de los estudiantes uno de los puntos clave a resolver. Dada la dificultad del proceso de evaluación, el diseño de métodos capaces de llevarla a cabo se ha convertido en una línea emergente de investigación. A esta dificultad cabe añadir la especialización de las materias a evaluar y más en un entorno universitario. Por ejemplo la evaluación automática de un programa informático o de un diagrama de clases son tareas complejas. En nuestro caso nos centraremos en la materia de bases de datos ya que, por su dificultad, requiere de herramientas específicas para llevar a cabo el proceso de evaluación automatizada. Así pues en el contexto de esta tesis remarcamos tres conceptos básicos: evaluación, Computer Based Assessment (CBA) y la docencia en Bases de Datos (BD).

En este capítulo se presentan los antecedentes necesarios para situar el contexto de la tesis. Inicialmente se analizan los aspectos más importantes de la Evaluación y del Computer Based Assessment, centrándonos básicamente en los entornos que permiten una corrección automática de ejercicios complejos. Se presentan también los temas más relevantes en docencia de BD y las metodologías que suelen utilizarse. A continuación, se describe el estado del arte en la utilización de herramientas de e-learning específicas para la enseñanza/aprendizaje de los distintos temas de BD, haciendo especial hincapié en las que nos permiten el CBA para la evaluación de los estudiantes. Finalmente se describe ACME que es la plataforma de e-learning sobre la que se van a integrar y validar los métodos propuestos en la tesis.

### 2.1 EVALUACIÓN

Actualmente la evaluación es uno de los temas con mayor protagonismo en el ámbito educativo, ya que tanto administradores, educadores, padres, alumnos y toda la sociedad en su conjunto, son más conscientes que nunca de la importancia y las repercusiones del hecho de evaluar o de ser evaluado. Existe quizá una mayor conciencia de la necesidad de alcanzar determinadas cotas de calidad educativa, y de aprovechar adecuadamente los recursos, el tiempo y los esfuerzos. Centrándonos en el ámbito universitario y en el actual marco del Espacio Europeo de Educación Superior (EEES) la evaluación por competencias es uno de los apartados con más relieve y que sin ninguna duda supone un gran esfuerzo y dedicación a los profesores. Por este motivo y como iremos viendo a lo largo de este trabajo, las tecnologías de la información y comunicación (TIC) juegan un papel muy importante.

En esta sección nos centraremos en describir las bases en materia de *evaluación*, considerada el primer pilar de nuestra investigación (ver Figura 2.1). Presentaremos una definición del concepto de evaluación y una clasificación de las diferentes técnicas (sección 2.1.1). A continuación nos centraremos en la evaluación formativa y sumativa (sección 2.1.2), presentando sus ventajas e inconvenientes (sección 2.1.3). También analizaremos la relación entre evaluación y la taxonomía de Bloom (sección 2.1.4) y nos centraremos en Computer Science. Concluiremos este primer bloque resaltando la importancia actual de la tecnología en las tareas de la evaluación (sección 2.1.5).

#### 2.1.1 Definición y clasificación

El Diccionario de la Real Academia de la Lengua define evaluar como “*estimar los conocimientos, aptitudes y rendimiento de los alumnos*”. Ya dentro del entorno educativo se dan varias definiciones de evaluación, entre ellas la de [Gar89] “*la evaluación es una actividad o proceso sistemático de identificación, recogida o tratamiento de datos sobre elementos o hechos educativos, con el objetivo de valorarlos primero y, sobre dicha valoración, tomar decisiones*”.

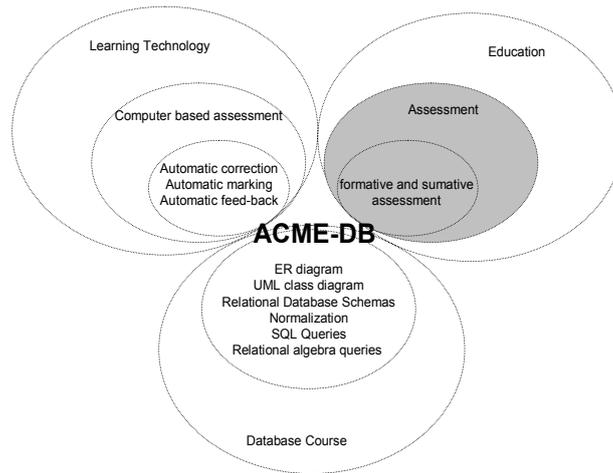


Figura 2.1 Marco de la tesis. Evaluación.

El concepto de evaluación no debe confundirse con el de calificación. El término calificación está referido exclusivamente a la valoración de los alumnos. Calificar, por tanto, es una actividad más restringida que evaluar. La calificación será la expresión cualitativa (apto/no apto) o cuantitativa (10, 9, 8, etc.) del juicio de valor que emitimos sobre la actividad y logros del alumno. En este juicio de valor se suele querer expresar el grado de suficiencia o insuficiencia, conocimientos, destrezas y habilidades del alumno, como resultado de algún tipo de prueba, actividad, examen o proceso. Podemos clasificar la evaluación según distintos criterios [Gar89]:

- **Según su finalidad y función:**
  - Función formativa: la evaluación se utiliza preferentemente como estrategia de mejora y para ajustar sobre la marcha, los procesos educativos de cara a conseguir las metas u objetivos previstos. Suele identificarse con la evaluación continua.
  - Función sumativa: suele aplicarse más en la evaluación final, es decir, una vez finalizado el periodo de enseñanza/aprendizaje. Con este tipo de evaluación no se pretende modificar, ajustar o mejorar el objeto de la evaluación, sino simplemente valorar el nivel obtenido, calificando al alumno.
- **Según su extensión:**
  - Evaluación global: se pretende abarcar todos los componentes o dimensiones del alumno, del centro educativo, del programa, etc. Cualquier modificación en uno de sus componentes o dimensiones tiene consecuencias en el resto. Con este tipo de evaluación, la comprensión de la realidad evaluada aumenta, pero no siempre es necesaria o posible.
  - Evaluación parcial: pretende el estudio o valoración de determinados componentes o dimensiones de un centro, de un programa educativo, de rendimiento de un alumno, etc.
- **Según los agentes evaluadores:**
  - Evaluación interna: es aquella que es llevada a cabo y promovida por los propios integrantes de un centro, un programa educativo, etc.
  - Evaluación externa: se da cuando agentes no integrantes de un centro o de un programa evalúan su funcionamiento. Suele ser el caso de la "evaluación de expertos".
- **Según el momento de aplicación:**
  - Evaluación inicial: se realiza al comienzo del curso académico, de la implantación de un programa educativo, del funcionamiento de una institución escolar, etc. Consiste en la recogida de datos en la situación de partida. Es imprescindible para iniciar cualquier cambio educativo, para decidir los objetivos que se pueden y deben conseguir, y también para valorar si al final de un proceso los resultados son satisfactorios o insatisfactorios.
  - Evaluación procesual: consiste en la valoración, a través de la recogida continua y sistemática de datos, del funcionamiento de un centro, de un programa educativo, del

proceso de aprendizaje de un alumno, de la eficacia de un profesor, etc. a lo largo del periodo de tiempo fijado para la consecución de unas metas u objetivos. La evaluación procesual es de gran importancia dentro de una concepción formativa de la evaluación, porque permite tomar decisiones de mejora sobre la marcha.

- Evaluación final: consiste en la recogida y valoración de unos datos al finalizar un periodo de tiempo previsto para la realización de un aprendizaje, un programa, un trabajo, un curso escolar, etc. o para la consecución de unos objetivos.

Con el auge de las TIC, nuestro sistema educativo en general y la evaluación como parte de él se han visto inmersos en un proceso de cambio, hablándose ahora de la evaluación automatizada como la que se realiza a través de un sistema informático. Tal como veremos en las siguientes secciones, la evaluación que suele usarse dentro del ámbito universitario suele ser una combinación de la evaluación formativa y sumativa. Así pues en este trabajo nos situaremos básicamente en la automatización de estos dos tipos de evaluaciones.

## 2.1.2 Evaluación formativa y sumativa

La *evaluación formativa* tiene como principal objetivo el ayudar en el proceso de aprendizaje [Kni01], aportando información a lo largo de todo el proceso enseñanza-aprendizaje. De este modo el profesor puede evaluar el nivel de aprendizaje, el ritmo de trabajo, detectar deficiencias y si es necesario cambiar las actividades propuestas. Por otra parte, al alumno le aporta información válida para su autoevaluación, se le informa sobre sus errores, sobre qué es lo importante y cómo debe afrontar el curso [JMM+04].

Entre las características de la evaluación formativa podemos citar:

- Se evalúa al terminar un tema o siempre que se considere oportuno.
- Los métodos de evaluación utilizados pueden ser sencillos e informales, como por ejemplo la corrección de ejercicios en clase, la realización de actividades a través de una plataforma de e-learning, etc.
- Se pretende evaluar el proceso sobre la marcha, para mejorarlo y conseguir un mayor rendimiento, evitando en la medida de lo posible un mayor número de fracasos por falta de información y de orientación durante el proceso de aprendizaje.
- Se aprende cuando conocemos nuestros errores en un tiempo y en una situación que nos permite corregirlos.
- Es muy frecuente no calificar estas evaluaciones ya que lo que importa es lo que el alumno sabe al final, no en la mitad del proceso. En cualquier caso estas evaluaciones deberían tener un peso menor en las calificaciones, aunque sí se puede tener en cuenta el haberlas hecho, como indicador de asistencia y participación. Por otra parte evaluaciones muy frecuentes y con nota pueden crear un clima de ansiedad que no es positivo.

La *evaluación sumativa* tiene como objetivo fundamental calificar o certificar el nivel del alumno al terminar un determinado período (parte del curso, curso completo, etc.) [Kni01]. Puede haber varias evaluaciones sumativas, como por ejemplo los exámenes parciales o una evaluación final. El método de evaluación utilizado debe ser riguroso y fiable ya que a partir de él se califica al alumno. Existen varios tipos de criterios que pueden combinarse entre sí: por objetivos conseguidos, en relación a los resultados de todo el grupo, etc. A partir de los criterios establecidos se califica y ésta suele ser la finalidad de este tipo de evaluación.

La diferencia más radical entre los dos enfoques está en la finalidad. La evaluación sumativa es la convencional, con los exámenes finales para ver dónde está cada alumno y que nota se merece. La evaluación formativa tiene como finalidad fundamental informar al profesor, replantear el ritmo con que lleva la asignatura, ver qué debe volver a explicar, qué ejercicios adicionales preparar, etc. y al alumno para que tome conciencia de dónde está y pueda corregir sus deficiencias y errores.

La evaluación sumativa suele ser prioritaria desde el punto de vista institucional debido a la necesidad de indicar el rendimiento del estudiante al final del curso académico. Sin embargo, la evaluación formativa tiene considerables ventajas pedagógicas y con ella se obtienen mejores resultados académicos.

### 2.1.3 Ventajas e inconvenientes de la evaluación formativa y sumativa.

A la hora de determinar ventajas e inconvenientes de la evaluación formativa y sumativa hemos de tener en cuenta diferentes aspectos. De esta forma Brown en [Bro01] considera que la evaluación da forma al aprendizaje, de manera que se puede cambiar el aprendizaje simplemente cambiando la forma de evaluación. En detrimento de la evaluación sumativa, considera que los estudiantes cada vez más aprenden “a aprobar exámenes” más que a adquirir un aprendizaje amplio de las competencias del curso. Por su naturaleza la evaluación sumativa invita al engaño ya que el estudiante es consciente de que se le está evaluando su trabajo en un determinado momento y no sus conocimientos y habilidades en la materia. El estudiante trata de resaltar lo que sabe y ocultar sus deficiencias para obtener una mejor nota. Por otra parte, se centra únicamente en los apartados del programa que se le van a evaluar y pasa por alto el resto.

Rowntree en [Row87] considera la relación entre los objetivos de aprendizaje de un sistema educativo con sus mecanismos de evaluación. De esta manera opina que para valorar un sistema de educación es necesario indagar sobre sus procedimientos de evaluación. Considera que la evaluación define lo que los estudiantes han aprendido, ya que son muy pocos los alumnos que dedican esfuerzos a adquirir conocimientos y habilidades que no son recompensados por el sistema.

Knight en [Kni01] señala que la evaluación formativa fomenta la legalidad más que el engaño. Los alumnos están más predispuestos a admitir su ignorancia sobre un tema si a continuación les damos ayuda que a aceptar una mala nota. Así pues y junto con las consideraciones previas, podemos afirmar que la evaluación formativa anima a una mejor formación y desmitifica el afán de los alumnos por “simplemente aprobar los exámenes”.

Por otra parte la evaluación formativa desmotiva el plagio. Stefani y Carroll [SC01] consideran que el plagio es difícil de verificar con precisión y se podría resolver mediante una mayor difusión de las normas académicas. Dado que en la evaluación formativa se fomenta la legalidad, los estudiantes tienden a admitir sus deficiencias, considerando que la copia sólo les perjudica a ellos mismos. Así pues bajo un entorno de evaluación formativa, el plagio es menos probable.

La evaluación de los resultados de aprendizaje es compleja, especialmente en la educación universitaria. [Kni01] considera que muchos resultados de aprendizaje no pueden ser evaluados de forma fiable sólo a partir de una evaluación sumativa y sostiene que la evaluación formativa es el único método que proporciona información sobre estos resultados. Afirma que un aumento de la evaluación formativa, como parte de una estrategia que también incorpora en menor medida evaluación sumativa, es una buena solución para obtener la valoración final.

Así pues, las ventajas pedagógicas que aporta la evaluación formativa son:

- alienta a los estudiantes al aprendizaje activo
- se puede usar para evaluar una gama más amplia de los objetivos de aprendizaje
- ayuda a la hora de establecer puntuaciones
- desalienta el plagio

Algunos de los problemas que presenta la evaluación formativa son [Kni01]:

- Infravaloración de la evaluación formativa. Si los alumnos están bajo la presión de una evaluación sumativa es probable que no den importancia a la formativa. Para solventar este problema propone una estrategia de evaluación basada en dos partes: inicialmente se usa la evaluación formativa y posteriormente se va introduciendo la evaluación sumativa.
- Efectividad de la evaluación formativa. Es necesario asegurar que la evaluación formativa se realice con el máximo rigor posible para que sea lo más efectiva. Para ello es necesario proporcionar el feedback más adecuado en cada momento para mejorar el aprendizaje.
- Problemas de evaluación formativa asociados con grupos numerosos. Para solventar este problema [Rus01] considera que se pueden utilizar métodos para asegurar la calidad de la evaluación. Entre estos métodos podemos citar:
  - Concentrar al máximo los esfuerzos al inicio del curso. Se da mucha importancia al trabajo en las primeras semanas, donde el profesor debe inculcar a los alumnos cómo debe ser el trabajo a realizar. La creación y difusión de instrucciones muy detalladas,

con ejemplos de actividades y su evaluación, evitan malas interpretaciones, reducen el tiempo de corrección y las solicitudes de revisión.

- Realización de la evaluación en la clase. Realizando actividades de evaluación en la clase, junto con las exposiciones de la materia. Deben llevarse a cabo y/o ser corregidas en la clase y el profesor debe facilitar el feedback más adecuado.
- Reducción del número de actividades de evaluación.
- Autoevaluación, evaluación por pares y evaluaciones en grupo. Se pueden realizar estas técnicas para una evaluación más rápida y solucionar problemas generales.
- Automatización de la evaluación. Con la finalidad de ahorrar tiempo y recursos en el proceso de la evaluación.

El primer apartado puede ser un buen enfoque metodológico. Ahora bien la realización de la evaluación en clase supone una reducción de tiempo en el desarrollo del programa y la reducción de actividades de evaluación limita la calidad de la evaluación. Por otra parte la evaluación y la evaluación por pares pueden suponer un ahorro de tiempo al profesor, pero no son la solución óptima.

Tanto la evaluación formativa como la sumativa tienen sus ventajas y sus inconvenientes. Nosotros consideramos que la evaluación formativa se ajusta más al enfoque que se pretende dar a la formación universitaria dentro del marco del EEES. Ahora bien, a pesar de las ventajas pedagógicas asociadas a la evaluación formativa, su uso no es el más frecuente en la educación superior. El motivo principal se debe al sobreesfuerzo en tiempo y recursos que supone una evaluación formativa bien realizada. Así pues, no es de extrañar que la tendencia en el ámbito universitario sea la de utilizar una combinación de las dos.

Por otra parte consideramos que la automatización de la evaluación, a pesar de los inconvenientes que puede suponer, es una buena solución para asegurar la calidad del proceso de enseñanza/aprendizaje y al mismo tiempo ahorrar tiempo y recursos a los profesores. Tal como se verá en la sección 2.2, un sistema CBA nos proporciona las herramientas necesarias para llevar a cabo estas tareas.

Nuestro trabajo está en la línea de los anteriores comentarios y se centra en desarrollar herramientas para la automatización de la evaluación, tanto la formativa como la sumativa.

## 2.1.4 Evaluación y taxonomía de Bloom

El tipo de evaluación depende también de los objetivos de aprendizaje establecidos. Estos objetivos están clasificados según la taxonomía de Bloom en seis niveles cognitivos: conocimiento, comprensión, aplicación, análisis, síntesis y evaluación (ver Figura 2.2) [Blo56] [KBB73]. Estos niveles pueden expresarse de forma muy parecida según la taxonomía revisada de Bloom (ver Figura 2.3) [AK01] en que estos niveles cognitivos se describen a través de las siguientes acciones verbales: recordar, comprender, aplicar, analizar, evaluar y crear.

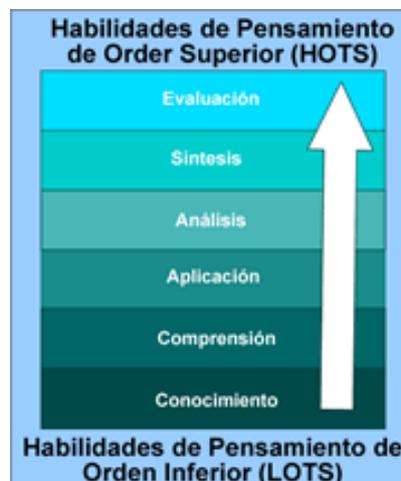


Figura 2.2 Taxonomía de Bloom.

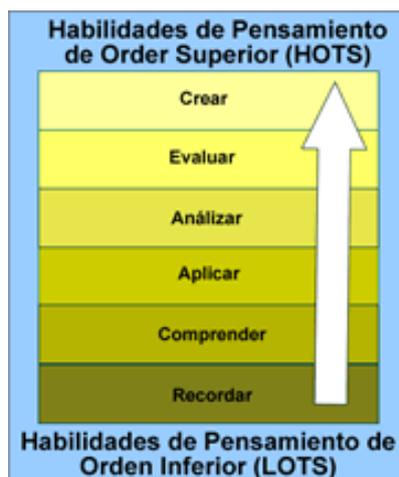


Figura 2.3 Taxonomía revisada de Bloom.

En la Tabla 2.1 presentamos de forma sintetizada la descripción de los niveles cognitivos de la taxonomía de Bloom. Cada fila corresponde a un nivel y en las columnas representamos las habilidades que debe adquirir el estudiante y posibles verbos que se pueden utilizar para expresar los objetivos de aprendizaje.

Teniendo en cuenta esta taxonomía, a la hora de evaluar debemos asignar a los estudiantes las actividades adecuadas según el nivel de objetivos de aprendizaje fijados. Por ejemplo, con ejercicios donde el alumno debe escoger una respuesta correcta entre varias podremos evaluar los dos primeros niveles de la taxonomía de Bloom, pero difícilmente los niveles superiores.

Diferentes trabajos examinan si la taxonomía de Bloom es apropiada para los estudios de Computer Science (CS). Todos ellos confirman su validez y aportan nuevas ideas, distintas interpretaciones y ejemplos de cómo se debe aplicar. En [JF06] se realiza un estudio sobre las evaluaciones de un primer curso de CS. Cinco profesores debían determinar qué nivel de conocimiento se estaba evaluando en las diferentes actividades. Hubo diferencias en sus consideraciones y la media obtenida fue que el 42% de las actividades evaluaban conocimiento, el 37% comprensión, el 36% aplicación y el 11% análisis. Los autores concluyen que los niveles más altos, síntesis y evaluación, sólo se pueden aplicar en el último curso. Por otra parte constatan las diferencias que tuvieron profesores y asesores en diferenciar los niveles de aplicación y análisis. Como estos estudios van encaminados al desarrollo de sistemas para aplicar en distintos entornos, sugieren un nuevo nivel que estaría por encima de los demás al que llaman “aplicación superior” para el cual es necesario saber analizar, evaluar y crear. [TLW+08] y [Sco03] proponen ejemplos de ejercicios para evaluar cada uno de los niveles en materia de programación y [Mac00] comenta la experiencia en la aplicación de la taxonomía de Bloom en tres asignaturas: estructuras de datos, inteligencia artificial y arquitectura de computadores. Por otra parte [JHR+09] agrupan los seis niveles en tres categorías: inferior (conocimiento y comprensión), intermedia (aplicación y análisis) y superior (síntesis y evaluación) y realizan un estudio detallado en 16 asignaturas, para determinar la relación entre las preguntas de examen y estas tres categorías. [HPV08] consideran que en el aprendizaje de la programación se deben realizar ejercicios que van desde los niveles de comprensión hasta los de síntesis. Entre estos ejercicios debe haber de más fáciles y de más difíciles y consideran como positivo que se puedan hacer a través de herramientas de evaluación automática.

En la enseñanza superior y especialmente en las carreras del ámbito científico/tecnológico, con un alto grado de experimentalidad y tecnología, las competencias que los alumnos deben adquirir nos indican que el alumno debe ser capaz de aplicar, analizar, diseñar, etc. diferentes temas en función de la materia. En los primeros cursos, la evaluación de los alumnos se basa en los niveles cognitivos inferiores de la taxonomía de Bloom: conocimiento, comprensión y aplicación. En los últimos cursos es donde el alumno tiene una visión más amplia de los diferentes temas y se le evaluarán niveles cognitivos superiores: aplicación, análisis y síntesis. Consideramos, tal como comentaremos en el Capítulo 3, que para adquirir las competencias de un curso de BD, los profesores deben evaluar los niveles de conocimiento, comprensión, aplicación y análisis.

CATEGORÍA	Descripción de las habilidades que se deben demostrar en este nivel	Que hace el Estudiante	Ejemplos de verbos que se pueden utilizar para expresar los objetivos de tipo cognitivo
<b>CONOCIMIENTO</b>	Se refiere a aprender información previamente aprendida. Reconocer informaciones, ideas, hechos, fechas, nombres, símbolos, definiciones, etc., de una manera aproximada a como se han aprendido.	El estudiante recuerda y reconoce información e ideas además de principios aproximadamente en la misma forma en que los aprendió	Decir, definir, describir, escribir, etiquetar, identificar, leer, listar, nombrar, numerar, recoger, reproducir, rotular, seleccionar
<b>COMPRENSIÓN</b>	Quiere decir entender (apropiarse, aferrar) lo que se ha aprendido. Se demuestra cuando se presenta la información de otra manera, se transforma, se buscan relaciones, se asocia a otro hecho, se interpreta o se saben decir las posibles causas y consecuencias.	El estudiante esclarece, comprende, o interpreta información en base a conocimiento previo	Asociar, citar, clasificar, comparar, convertir, dar ejemplos, describir, discutir, estimar, explicar, exponer, generalizar, ilustrar, resumir
<b>APLICACIÓN</b>	El alumno selecciona, transfiere y utiliza datos y leyes para completar un problema o tarea con un mínimo de supervisión. Utiliza lo que ha aprendido. Aplica las habilidades adquiridas a nuevas situaciones que se le presentan. Utiliza la información que ha recibido en situaciones nuevas y concretas para resolver problemas	El estudiante selecciona, transfiere, y utiliza datos y principios para completar una tarea o solucionar un problema	Aplicar, calcular, construir, demostrar, determinar, establecer, incluir, informar, producir, proporcionar, proyectar, relacionar, resolver, solucionar, usar, utilizar
<b>ANÁLISIS</b>	El alumno distingue, clasifica y relaciona evidencias o estructuras de un hecho o de una pregunta, se hace preguntas, elabora hipótesis. Descompone el todo en sus partes y puede solucionar problemas a partir del conocimiento adquirido, razona. Intenta entender la estructura de la organización del material informativo examinando las partes de las que se compone. La información que obtiene le sirve para desarrollar conclusiones. Identifica motivos y causas haciendo inferencias y/o halla evidencias que corroboran sus generalizaciones.	El estudiante diferencia, clasifica, y relaciona las conjeturas, hipótesis, evidencias, o estructuras de una pregunta o aseveración	Analizar, categorizar, comparar, construir diagramas, contrastar, discriminar, distinguir, inferir, limitar, precisar, priorizar, seleccionar, separar, subdividir
<b>SÍNTESIS</b>	El alumno crea, integra, combina ideas, planea, propone nuevas maneras de hacer. Crea aplicando el conocimiento y las habilidades anteriores para producir algo nuevo o original. Se adapta, prevé, se anticipa, categoriza, colabora, se comunica, compara ...	El estudiante genera, integra y combina ideas en un producto, plan o propuesta nuevos para él o ella.	Adaptar, codificar, combinar, compilar, componer, comunicar, contrastar, crear, elaborar hipótesis, estructurar, expresar, formular, generar, integrar, inventar, planear, reorganizar, revisar, validar
<b>EVALUACIÓN</b>	Emitir juicios sobre la base de criterios preestablecidos. Emitir juicios respecto al valor de un producto según las propias opiniones personales a partir de unos objetivos determinados.	El estudiante valora, evalúa o critica en base a estándares y criterios específicos.	Argumentar, ayudar, comparar, concluir, contrastar, criticar, decidir, evaluar, interpretar, justificar, juzgar, predecir, recomendar, valorar

Tabla 2.1 Descripción de los niveles de conocimiento según la taxonomía de Bloom.

## 2.1.5 Evaluación y tecnología

El aumento de la utilización de la tecnología en entornos de e-learning ha impulsado la evaluación basada en el ordenador y esto ha originado un rediseño de los sistemas de enseñanza tradicionales [Ell08]. El aprendizaje pasa a ser un proceso continuo [Jeg05], durante toda la vida, con lo cual la evaluación de este proceso permanente pasa a adquirir una gran importancia. Según [Ben02] la tecnología es un componente esencial del sistema de aprendizaje moderno y en consecuencia, la tecnología lo es también en el proceso de evaluación. Tal como veremos en la sección 2.2, dentro de las tecnologías del e-learning se distinguen el Computer Based Assessment (CBA) y el Computer Assisted Assessment (CAA), que a menudo se usan indistintamente y algunas veces de forma incoherente. Nos referimos a CBA cuando hay una interacción entre el estudiante y el ordenador durante el proceso de evaluación. En dicha evaluación, la entrega de pruebas, corrección y feedback al alumno es realizado de forma totalmente automática por el sistema, mientras que el CAA es más general y abarca todo el proceso de evaluación, pudiéndose realizar las tareas de evaluación fuera del sistema [CE98].

Por otra parte, se ha demostrado empíricamente que el sistema de enseñanza uno a uno (alumno-profesor) es mejor que la enseñanza en el aula tradicional, con mejoras de aprendizaje muy substanciales [Blo84] pero no es aplicable en un entorno universitario. Para complementar la enseñanza tradicional también se ofrecen tutorías, aunque la ratio de alumnos/profesor de éstas es bastante elevada y alejada del modelo de enseñanza individualizada. Es por este motivo que los sistemas de e-learning que permiten una evaluación totalmente automatizada y que ofrecen un feedback de calidad tienden a realizar las funciones de un tutor individual para cada alumno. Se ha demostrado [Cor01] que estos sistemas mejoran el nivel de aprendizaje en materias que requieren de la resolución de problemas.

Por estos motivos, la utilización de ordenadores para ayudar en las tareas de evaluación ha sido un tema de investigación durante décadas, aunque básicamente ha consistido en aplicar los métodos tradicionales de evaluación a un entorno informático, proliferando la evaluación a través de ejercicios tipo test, elección múltiple, etc. Sin embargo son pocos los trabajos referentes a la evaluación de actividades complejas en donde el alumno pueda introducir libremente su solución. Estas herramientas deben proporcionar al profesor funcionalidades para el seguimiento individual y de forma que garanticen la evaluación formativa del alumno. Por otra parte, las instituciones universitarias nos exigen una calificación numérica como resultado final de las actividades de evaluación. En esta línea es deseable que las herramientas dispongan de las funcionalidades oportunas para almacenar el resultado de todas las pruebas realizadas, a partir de las cuales el sistema de forma automática (o el profesor, en vista de los resultados) establecerá una calificación. Además, especialmente en la resolución de problemas complejos, es deseable que la corrección no sólo sea correcto/incorrecto sino que facilite una puntuación fiable, lo más parecida posible a la de un profesor.

## 2.1.6 Resumen

La evaluación de las competencias adquiridas por los alumnos en los estudios del nuevo EEES es una de las tareas relevantes que los profesores deben asumir. El sistema tradicional de evaluación basado únicamente en los exámenes finales ha quedado postergado a unas nuevas formas de evaluar. En la actualidad el sistema mayoritario de evaluación de nuestros universitarios se basa en la combinación de la evaluación formativa y la sumativa tradicional.

Uno de los inconvenientes de la evaluación formativa es que necesita más recursos que la evaluación sumativa debido a su frecuencia y al detalle del feedback a proporcionar. Teniendo en cuentas que las universidades disponen de pocos recursos y que el sistema universitario español favorece la investigación frente a la docencia en los curriculums de los profesores, difícilmente se puede llevar a cabo una evaluación formativa y menos todavía en grupos numerosos. Para paliar esta situación y en la línea de nuestro trabajo, se están desarrollando herramientas automatizadas que permiten la realización de actividades de evaluación, su corrección y facilitan el feedback oportuno al alumno.

## 2.2 COMPUTER BASED ASSESSMENT

Con el impulso de las TIC's las empresas han dedicado gran cantidad de recursos en la mejora de sus equipos y sistemas informáticos. Con ellos se han automatizado muchos de los procesos manuales y repetitivos que debían hacer. Esta influencia está llegando también al sector de la enseñanza, con el ánimo de automatizar las tareas más repetitivas. Dentro de estas tareas la corrección de trabajos, ejercicios, prácticas, exámenes, etc. consumen una gran parte del tiempo del profesor y más si tenemos en cuenta que los grupos suelen ser numerosos. Por este motivo, se están desarrollando plataformas educativas, cada vez más complejas, que permiten llevar a cabo todo el proceso de aprendizaje de una materia. Los sistemas CBA además facilitan las labores de corrección y evaluación del alumnado.

En el ámbito universitario cada vez se exige más a los profesores. Estos se deben implicar en tareas de investigación, de transferencia de tecnología, de gestión, etc. Desde un punto de vista docente y especialmente con la entrada del EEES se les exige un control más directo sobre el aprendizaje de los alumnos, con más actividades a evaluar y que sin duda alguna suponen una mayor dedicación. Por otra parte los ejercicios que se desarrollan en muchas asignaturas del ámbito científico/técnico son complejos, con múltiples soluciones correctas y suponen un tiempo considerable de corrección.

En esta sección nos centraremos en describir las bases del CBA, considerado el segundo pilar de nuestra investigación (ver Figura 2.4). Siguiendo la estructura del bloque anterior presentaremos una definición del concepto y la terminología asociada (sección 2.2.1). A continuación mostraremos una clasificación de los sistemas CBA (sección 2.2.2). Veremos también las ventajas (sección 2.2.3) e inconvenientes (sección 2.2.4) que presenta. Finalmente se presentará la evaluación de los niveles cognitivos superiores utilizando CBA (sección 2.2.5) así como su aplicación en Computer Science (sección 2.2.6).

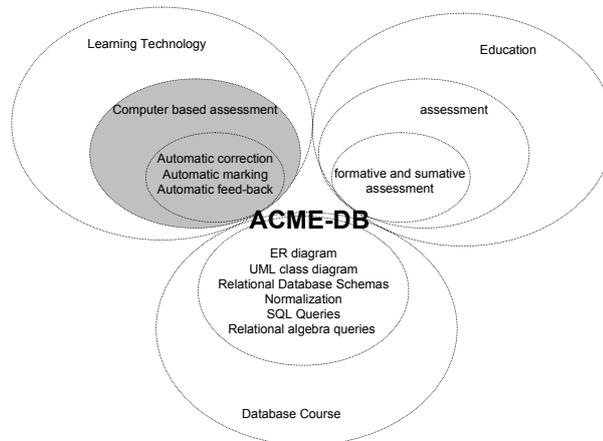


Figura 2.4 Marco de la tesis. CBA.

### 2.2.1 Definición

Actualmente las universidades disponen de entornos de e-learning desde donde el alumno puede consultar y descargarse los materiales del curso. También puede entregar sus trabajos y realizar simples cuestionarios de autoaprendizaje y/o evaluación a través de ejercicios de respuesta fija. Estos entornos son suficientes para muchos profesores de materias en que sólo haya que evaluar niveles bajos de conocimiento según la taxonomía de Bloom (sección 2.1.4). Ahora bien, en entornos/materias con un alto grado de experimentalidad y para evaluar niveles altos de conocimiento, estas herramientas no son suficientes y se requiere de entornos de e-learning más especializados.

Por otra parte la evaluación de los alumnos [Mca02] es una tarea difícil y más en grupos numerosos. En este contexto los profesores tienen pocas opciones a la hora de evaluar [Gib92], [BR96], [CEA+03]. La alternativa más aceptada es aplicar las TIC para desarrollar entornos y plataformas especializados que automaticen el proceso de evaluación. El conjunto de todos los procesos necesarios para la realización de

la evaluación se conoce como el ciclo de vida de la evaluación. En las últimas décadas se han propuesto diferentes entornos de e-learning que varían en función del grado de automatización de este ciclo de vida. Esta variabilidad ha dado lugar a la siguiente terminología:

- **Computer Assisted Learning (CAL)** es un término generalizado que se refiere a la utilización de la tecnología para facilitar el proceso de aprendizaje de forma virtual. Sólo se automatizan de forma superficial algunas de las tareas de enseñanza y aprendizaje. De esta forma, la entrega a los alumnos de materiales, como tutoriales, apuntes de clase, manuales, etc. mediante paquetes de software como Microsoft PowerPoint, ficheros .pdf, etc. constituye una forma básica de CAL.
- **Computer Based Learning (CBL)** se define como un subconjunto de la CAL en que los materiales de aprendizaje deben ser presentados al estudiante de forma on-line a través de un ordenador. El estudiante navega en su sistema y obtiene los materiales didácticos disponibles que va aprendiendo a su ritmo. Sólo las tareas de enseñanza y aprendizaje están automatizadas, no la evaluación.
- **Computer Assisted Assessment (CAA)** se refiere a la utilización de la tecnología para ofrecer cursos a los alumnos y calificar las respuestas a los ejercicios propuestos [CM03] [SM97]. En el CAA es probable que algunas tareas de evaluación, enseñanza y aprendizaje estén ya automatizadas, pero algunas etapas del proceso todavía se realizan a través de mecanismos externos al sistema (suelen ser las de obtención y evaluación de las soluciones). Por ejemplo, una práctica habitual es el uso de marcas ópticas (Optical Mark Reader) para leer las respuestas de los alumnos que posteriormente el sistema comparará con el modelo correcto. Los materiales generalmente se crean a partir de herramientas informáticas, con lo que el CAA se puede considerar una especialización del CAL que engloba la puntuación automática y el análisis y entrega de feedback a los envíos de los alumnos.
- **Computer Based Assessment (CBA)** se refiere a un sistema integrado y on-line que permite la entrega de materiales para la enseñanza y su evaluación, la entrada de soluciones por parte de los estudiantes, un proceso de evaluación automatizado y el feedback adecuado [CE98] [Tsi02] [Bli06]. Por sus características, el CBA puede considerarse una especialización del CAA en la que todo el proceso, incluyendo la entrada de soluciones, se produce on-line. Un sistema de este tipo debe tener las funcionalidades de corrección, evaluación y feedback totalmente automatizadas.

En la Figura 2.5 representamos la relación que existe entre estas tecnologías en función del grado de especialización en el proceso de automatización de las tareas de enseñanza y aprendizaje y su capacidad para soportar la evaluación. El CBA es la forma más especializada en la tecnología de e-learning actual, ya que proporciona el mayor nivel de automatización y actúa a lo largo de todo el ciclo de vida de la evaluación.

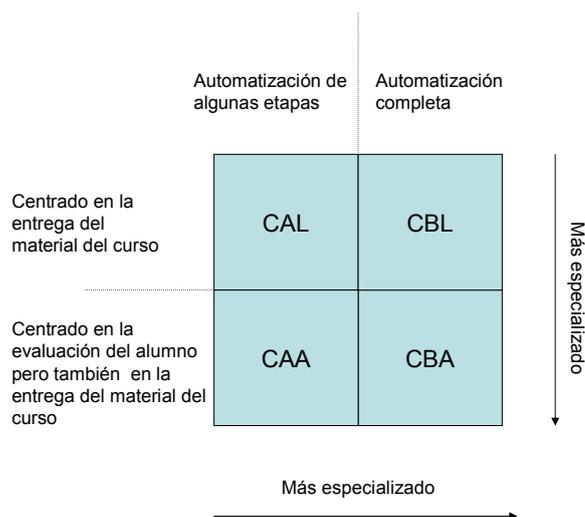


Figura 2.5 Relación entre tecnologías de e-learning.

Parte de nuestro trabajo se centrará en los entornos de CBA, por ello en las siguientes secciones los analizaremos con más detalle, presentando una clasificación de los mismos, sus ventajas e inconvenientes.

## 2.2.2 Clasificación de los sistemas Computer Based Assessment

La clasificación del CBA se puede realizar en función de la tipología de ejercicios que permite resolver. De esta forma se pueden considerar dos tipos de ejercicios: de *respuesta fija* y de *respuesta libre*. La diferencia principal entre ellos radica en el proceso por el cual el alumno construye la solución del problema.

### Evaluación automatizada a partir de respuestas fijas

En los sistemas de *respuesta fija*, el estudiante elige una solución entre un número fijo de alternativas definidas previamente. Una o más de las alternativas es la solución correcta, siendo las otras incorrectas. En este caso el sistema de evaluación simplemente registra si la solución enviada es correcta, proceso que puede realizarse con una simple comparación. A veces cuando nos referimos a este tipo de evaluación la llamamos *evaluación objetiva*.

Básicamente los ejercicios de respuesta fija incluyen:

- **Preguntas de elección múltiple (PEM).** En este caso se presenta al usuario una pregunta seguida de una serie de opciones de las que se debe seleccionar una o más que son las correctas. Frosini [FLM98] proporcionan una lista de variantes de las PEM: preguntas tipo verdadero/falso, elección múltiple, múltiple respuesta, relleno de blancos, etc. Los sistemas actuales permiten que tanto preguntas como respuestas sean textos, imágenes, sonidos o videos. La mayoría de sistemas comerciales ofrecen plataformas de evaluación que permiten PEM. Por ejemplo Questionmark [BSP+03] (<http://www.questionmark.com/esp/index.aspx>) permite la creación de PEM dentro de un entorno de edición interactivo e incluye funcionalidades para la presentación de preguntas y respuestas con una gran variedad de colores, imágenes, fuentes, etc. También permite abrir otras aplicaciones que se puedan necesitar en el momento de preparar la respuesta (hojas de cálculo, calculadoras, etc). Otras plataformas con prestaciones similares son LXR\*TEST (<http://www.lxrtest.com/site/home.aspx>), Quiz It (<http://www.chemware.co.nz/quizit.htm>), etc.
- **Ejercicios de respuesta corta.** En este caso el estudiante responde al ejercicio en forma de palabra, frase corta o un número. El sistema simplemente compara la respuesta dada a una o varias respuestas correctas. Por su parecido con las PEM, la mayoría de plataformas permiten la evaluación a través de este tipo de ejercicios.
- **Ejercicios con gráficos.** En este caso el alumno selecciona un gráfico, para relacionarlo con otros objetos a través de una línea, o para organizar gráficos en un recuadro, ordenarlos, etc. La respuesta correcta está definida de antemano. Questionmark dispone de este tipo de ejercicios que ya previamente se podían crear a través de distintas herramientas multimedia (Asymetrix Toolbook, Macromedia Authorware).

### Evaluación automatizada a partir de respuestas libres

En los sistemas de *respuesta libre* las soluciones son construidas libremente sin ninguna restricción. A diferencia de los ejercicios de respuesta fija en los que la corrección se realiza con una simple comparación, en este caso para la evaluación de la solución se requiere de sistemas específicos. Por un lado se deben realizar interfaces adecuadas para la entrada de las soluciones y los métodos de corrección y evaluación son más complejos ya que la solución del alumno no puede definirse de forma única. Debido a esta complejidad son pocas las plataformas existentes que permiten respuestas libres.

Los ejercicios de respuesta libre son adecuados para una gran variedad de actividades permitiendo, por ejemplo, evaluar ejercicios de programación, ejercicios con diagramas, ejercicios basados en un texto libre (ensayo) etc. Veamos una breve descripción de cada uno de ellos:

- **Ejercicios de programación.** Se realizan principalmente en estudios de Computer Science y otras carreras del ámbito científico/técnico. Históricamente la corrección de estas tareas ha sido uno de los temas pioneros en la evaluación automatizada. Con ellos se pueden evaluar las habilidades cognitivas de aplicación, análisis y síntesis dentro de la taxonomía de Bloom. En la sección 2.2.6 se realizará una descripción de los sistemas más relevantes que facilitan el aprendizaje y evaluación en ejercicios de programación
- **Ejercicios de diagramas.** Se basan en la realización de diagramas y no son muy frecuentes. Aunque son muchos los paquetes de software que permiten y facilitan la realización de diagramas, son muy pocos los que permiten su evaluación. Entre estos paquetes destacan Microsoft Visio (<http://office.microsoft.com/en-us/visio/default.aspx>) que permite la realización de gran variedad de diagramas y Rational Rose [Qt99] (<http://www-01.ibm.com/software/awdtools/developer/rose>) que permite la realización de diagramas UML. Estos paquetes están pensados para desarrolladores de software y no están orientados al aprendizaje y evaluación. También hay bastantes autores que en sus artículos describen herramientas que ayudan en el proceso de realización y aprendizaje de algunos tipos de diagramas. En la sección 2.2.6 también se realizará una descripción de los sistemas más relevantes que facilitan el aprendizaje y evaluación en este tema.
- **Ejercicios de texto libre.** Se basan en la corrección de un texto libre (ensayo) y son típicos ejercicios de evaluación con los que se pueden evaluar los niveles cognitivos superiores. En este caso se trata de interpretar el lenguaje natural, siendo este un tema amplio de investigación y que no se trata en esta tesis.

### 2.2.3 Beneficios del Computer Based Assessment

Los beneficios que aporta el CBA respecto a los sistemas de evaluación clásicos (es decir los realizados por un profesor) son varios. Según Charman y Elmes [CE98] estos se pueden clasificar en beneficios prácticos y pedagógicos:

- Desde un punto de vista práctico, el CBA nos permite reducir el tiempo dedicado a las tareas de evaluación y además nos facilita la diversificación de la evaluación, permitiendo la evaluación formativa y sumativa. Puesto que el tiempo de evaluación depende del número de estudiantes estos sistemas resultan especialmente beneficiosos en grupos numerosos.

Los beneficios del CBA se deben valorar en un periodo largo de tiempo ya que al principio, el desarrollo del sistema, el aprendizaje de su uso, la preparación y escritura de ejercicios y la sustitución de los métodos tradicionales de evaluación suponen un coste considerable de tiempo y esfuerzo por parte de los profesores. Sin embargo, una vez tenemos el sistema automatizado en funcionamiento y las colecciones de ejercicios desarrolladas, supone un ahorro de recursos y de tiempo muy considerable ya que se pueden reutilizar en repetidas ocasiones con un esfuerzo mínimo. Debe remarcarse que un análisis coste/beneficio de estos sistemas en un periodo corto de tiempo fácilmente puede ser negativo.

- Desde un punto de vista pedagógico y teniendo en cuenta diferentes estudios [CE98, BBF93, Tsi02, Bli06], la experiencia del uso de CBA:
  - Aumenta la participación de los estudiantes con más dificultad en el aprendizaje. En este mismo sentido, facilita la detección de las deficiencias de aprendizaje y permite actuar en consecuencia aplicando las estrategias más idóneas en cada caso.
  - Proporciona feedback inmediato a las respuestas de los alumnos. Se reducen los tiempos de espera con lo cual el alumno tiene en mente los razonamientos aplicados para resolver el problema. En caso de error le es más fácil detectar dónde se ha equivocado.

- Permite aumentar la confianza de los estudiantes. Aplicando estrategias en las que los primeros ejercicios propuestos suelen ser fáciles para que obtengan buenos resultados que sin duda motivan al trabajo y al esfuerzo continuado.
- Ayuda a la planificación del tiempo y de la eficacia en el trabajo. Saben que pueden ir enviando soluciones hasta la fecha límite y que con mejores soluciones obtienen mejores puntuaciones.
- Aumenta la conciencia de los estudiantes sobre el proceso de evaluación.

Mencionar también la importancia del CBA en la evaluación formativa en cualquier sistema de e-learning. Aunque permite diferentes enfoques, tal como se comenta en [FRB05], en este tipo de evaluación se produce un feedback continuo entre profesores y alumnos que permite mejorar las actividades de enseñanza /aprendizaje.

Finalmente, teniendo en cuenta los criterios propuestos por [BR96] que consideran que una buena evaluación debe ser: válida, fiable, imparcial, equitativa, formativa, oportuna, incremental, recuperable, exigente y eficiente, el CBA presenta ventajas pedagógicas sobre la evaluación tradicional en 7 de estos 10 criterios (véase Tabla 2.2).

<b>Criterio</b>	<b>Significado</b>	<b>Aplicación en el CBA</b>	<b>Ventajas</b>
Válida	Se debe evaluar lo que se ha planificado y enseñado, no otras cualidades.	Medirá los aspectos especificados del curso. Se supone un buen diseño de la evaluación inicial.	√
Fiable	Debe ser consistente y fiable, consensuado entre los evaluadores.	El mismo proceso de evaluación se llevará a cabo en cada prueba. La coherencia es absoluta.	√
Imparcial	Los estudiantes han de percibir el proceso como justo, con igualdad de oportunidades para todos ellos.	Depende del diseño. Puede ser que a un alumno se le asigne un ejercicio más fácil. CBA no ofrece ventajas.	X
Equitativa.	No debe haber ningún tipo de discriminación entre los estudiantes, todos han de ser tratados por igual.	El mismo proceso de evaluación se llevará a cabo en cada prueba. No existe ningún tipo de discriminación.	√
Formativa	Debe ofrecer amplias oportunidades de aprendizaje.	CBA proporciona la oportunidad de realizar actividades de evaluación con frecuencia, permitiendo varios envíos y ofreciendo feedback.	√
Oportuno	Las actividades de evaluación se deben hacer en el momento oportuno dentro del periodo de aprendizaje.	CBA permite realizar actividades de evaluación en cualquier momento dentro del proceso de aprendizaje.	√
Incremental	Debe ser un proceso gradual a través del cual el estudiante ve sus progresos.	Depende del diseño. CBA no ofrece ventajas.	X
Recuperable	Se deben poder recuperar los errores iniciales y proporcionar una segunda oportunidad.	CBA es adecuado para permitir, en caso de error, el envío de muchas soluciones.	√
Exigente	La evaluación no debe ser fácil y debe estar enfocada al nivel correcto según lo que se testea.	Depende del diseño. CBA no ofrece ventajas.	X
Eficiente.	Debe ser manejable y efectiva teniendo en cuenta las limitaciones de los recursos.	Es una de las motivaciones del CBA: el ahorro de tiempo y recursos.	√

Tabla 2.2 Criterios para una buena evaluación aplicados al CBA.

## 2.2.4 Limitaciones del Computer Based Assessment

Al igual que todo sistema de evaluación el CBA tiene ventajas e inconvenientes. En esta sección presentaremos las que se han identificado como limitaciones principales del CBA. Entre estas limitaciones podemos mencionar:

- Desde un punto de vista práctico, las limitaciones del CBA están relacionadas con distintos factores que intervienen en su implantación. Inoue en su informe [Ino01] identifica seis factores que son aplicables a cualquier tecnología educativa :
  - Conocimiento y habilidades en tecnología de los profesores.
  - Disponibilidad de hardware y software apropiado.
  - Compromiso en las partes involucradas. Los educadores deben tener la determinación de perseverar en la solución de los problemas relacionados con la aplicación de la tecnología educativa, en lugar de volver a las formas tradicionales de evaluación.
  - Disponibilidad de tiempo. La implantación de un sistema de CBA supone en la fase inicial una gran inversión en tiempo y esfuerzo.
  - Disponibilidad de soporte técnico. La tecnología educativa puede ser difícil de aplicar. Los desarrolladores deben proporcionar el apoyo técnico adecuado a los profesores.
  - Costes asumibles. El coste de todo el sistema no debe ser excesivo.

Según [CE98] las limitaciones se centran básicamente en la disponibilidad del hardware y software adecuado para la preparación del material, su distribución y la posterior recepción y evaluación de las actividades.

- Dificultad para asegurar la autoría de las soluciones. Cuando un sistema CBA es utilizado en la educación a distancia, otro de los problemas existentes es que desconocemos quien realmente ha resuelto los ejercicios y pruebas planteados, motivo suficiente para tener que realizar una prueba final presencial para evitar posibles fraudes. Esto mismo ocurre en los entornos donde se utiliza un sistema CBA como complemento a las clases presenciales. El sistema podría evaluar automáticamente al alumno en función del trabajo realizado. Ahora bien, desde el momento en que el alumno puede realizar su trabajo fuera de las aulas, no sabemos realmente quien hay detrás de su trabajo. Así pues se podría pensar que el papel de CBA es muy limitado ya que siempre se acaba realizando una prueba que deberá corregirse manualmente. Este problema, tal como comentaremos más adelante, se puede minimizar si el propio CBA dispone de funcionalidades que permiten la realización de exámenes. El profesor convoca en un aula a sus alumnos donde presencialmente realizarán la prueba a través del mismo CBA. En estos casos es deseable que el sistema disponga de mecanismos de puntuación automática de forma que el CBA obtenga directamente la puntuación del alumno evitando la corrección manual.
- Asumiendo la taxonomía de Bloom como la más aceptada en términos de evaluación automatizada, las limitaciones pedagógicas del CBA están relacionadas con el nivel cognitivo a evaluar y el tipo de respuesta que puede evaluar el sistema. De esta forma los ejercicios de respuesta fija básicamente permiten evaluar los conocimientos de nivel inferior de la taxonomía de Bloom (conocimiento y comprensión) y por lo tanto la evaluación es muy limitada. La situación se agrava si el sistema de evaluación es un CBA basado principalmente en este tipo de ejercicios. En [JA00] sus autores constatan que estos métodos de evaluación tienen las siguientes desventajas:
  - Los alumnos pueden adivinar la solución correcta de las alternativas que se presentan
  - Puntuaciones negativas a las respuestas incorrectas para evitar el “adivinar por casualidad” desanima a los alumnos.
  - Los alumnos pueden descartar varias opciones tan solo con un poco de sentido común.
  - A menudo es difícil saber porque un alumno escogió una respuesta, pudiendo obtener una respuesta correcta a partir de un razonamiento erróneo.
  - Los estudiantes pueden verse perjudicados de manera desproporcionada en función de la redacción de las preguntas y las posibles respuestas. La tasa de respuestas correctas puede verse reducido en un 20% por el simple hecho del uso de una palabra desconocida.

Por otra parte [BD04] señalan que el CBA está generalmente asociado con preguntas de opción múltiple, mientras que [Cul98] señala que la mayoría de ellos sólo tienen en cuenta ejercicios de respuesta fija. Este tipo de ejercicios tiene la ventaja de que los algoritmos de evaluación son muy sencillos, ya que sólo tienen que comprobar si la respuesta es correcta. Por esta misma razón otros sistemas mecanizados de evaluación, tales como los basados en el reconocimiento óptico de marcas (OMR) se basan en preguntas de respuesta fija. La creencia que los sistemas CAA y CBA se basan exclusivamente en este tipo de preguntas es mayoritaria entre los profesores que asocian la evaluación automatizada con preguntas de respuesta fija. En el informe [CEA+03] sólo un 36% de los profesores encuestados estaba de acuerdo en que se podía evaluar el aprendizaje de niveles altos a través de medios automatizados.

Con los ejercicios de respuesta libre los estudiantes responden sin ninguna restricción a las preguntas formuladas. Con este tipo de ejercicios estas limitaciones del CBA se reducen considerablemente. El problema es que hay pocos entornos capaces de aceptar este tipo de ejercicios debido a la complejidad de los algoritmos necesarios para llevar a cabo la corrección y puntuación automatizada.

Vemos pues que uno de los retos actuales está en cómo adaptar el CBA en la evaluación de los niveles cognitivos superiores. Dada la importancia en este tema, en la siguiente sección analizaremos este aspecto con más detalle.

## 2.2.5 Evaluación de los niveles cognitivos superiores utilizando CBA

Tal como hemos comentado en la sección precedente, la evaluación automatizada se ha asociado tradicionalmente a la evaluación de los niveles cognitivos más bajos de la taxonomía de Bloom. Ello es debido a que para este tipo de evaluación son suficientes preguntas de respuesta fija que fácilmente son corregidas de forma automática. Todavía se considera que la evaluación de los niveles cognitivos superiores, remarcados en la Figura 2.6, es difícilmente factible a través del CBA debido a su dependencia de los sistemas que la deben llevar a cabo [RJE02].



Figura 2.6 Niveles superiores Taxonomía de Bloom.

La investigación sobre la evaluación a través de ordenador de los niveles cognitivos más altos de la taxonomía de Bloom se divide en dos amplias categorías:

- El enfoque de la primera categoría consiste en diseñar cuidadosamente las preguntas de respuesta fija para forzar a los estudiantes a demostrar sus habilidades cognitivas superiores (análisis, síntesis y evaluación) minimizando la limitación pedagógica surgida por el tipo de respuesta. En este sentido [CE98] sugieren la construcción de preguntas de elección múltiple, en donde las respuestas se basan en afirmaciones/razonamientos y [DK01a] presentan el diseño técnico de preguntas de este tipo y de preguntas gráficas para garantizar la calidad de la evaluación automatizada en la enseñanza superior.

- El enfoque de la segunda categoría se basa en utilizar métodos más complejos de evaluación que permitan a los CBA utilizar los tipos de preguntas que se utilizan tradicionalmente para evaluar las capacidades cognitivas de orden superior [MB99].

Así pues, uno de los problemas con que se encuentra el CBA es el grado de especialización que se requiere para evaluar niveles cognitivos más altos dentro de la taxonomía de Bloom. Los sistemas se han de desarrollar a medida del tipo de contenido/materia que se pretende evaluar. Por ejemplo si se quiere evaluar si un alumno sabe programar se requiere un entorno especializado que le permite enviar sus programas y que sepa corregirlos y ofrecer el feedback oportuno [LSK05][MK04].

Aunque la tecnología actual es lo suficientemente amplia y potente para realizar estas tareas, a veces los profesores son reticentes a esta automatización. Por ejemplo, en [CS98] los autores sostienen que la puntuación final siempre debe ser realizada por personas, utilizando la automatización como una simple ayuda. En [MW98] y [MW99] se propone un enfoque que implica un sistema on-line para la visualización de los exámenes y para la recogida de las respuestas, pero sólo dan un papel limitado a la puntuación automática. Nosotros, al igual que otros trabajos [Tsi02][Bli06], consideramos que sí es posible un proceso de evaluación automatizado de los niveles cognitivos superiores. Por otra parte, al igual que [Swi06], consideramos que el CBA ofrece un gran potencial de ventajas para el alumno y para el profesor.

Nuestra línea de investigación forma parte de la segunda categoría. Nos centraremos en la automatización de la evaluación de los niveles cognitivos superiores a través de la utilización de mecanismos de evaluación más compleja. Las tareas de programación y la realización de diagramas [Tsi02] [Bli06] son adecuados para evaluar los niveles cognitivos superiores de la taxonomía de Bloom y por lo tanto se puede utilizar para reducir las limitaciones pedagógicas del CBA. En nuestro caso nos centraremos en el entorno del Computer Science y en particular en las asignaturas de bases de datos. Aunque en la sección 2.4 analizaremos en detalle las aportaciones realizadas en bases de datos, hemos considerado oportuno detallar en la sección 2.2.6 las principales aportaciones en Computer Science.

## 2.2.6 Entornos CBA en Computer Science

Históricamente, la investigación en la evaluación automatizada ha sido motivada por el deseo de evaluar a más estudiantes, de forma más rápida y con menos esfuerzo por parte del profesor. Los primeros sistemas de evaluación automatizada datan de los años sesenta en que algunos profesores de física, matemáticas y especialmente de Computer Science querían aprovechar el ahorro de tiempo que suponía el uso de ordenadores para realizar tareas simples y repetitivas en el proceso de evaluación.

El primer sistema de evaluación automatizado [Hol59] [Hol60] se realizó para evaluar los programas realizados en lenguaje máquina en un curso de programación. A partir de ahí [FW65] realizaron un sistema parecido para evaluar programas en Balgol, una variación de Algol-58. Los alumnos dejaban su trabajo en tarjetas perforadas que luego eran tratadas en un proceso batch. Posteriormente [HW69] describen un sistema parecido al anterior, en donde se automatizaban los procesos batch. Más tarde el proceso de evaluación automatizada se expandió a otras áreas, entre ellas la física [TD76], las matemáticas [RH83] y la química [Mye86]. Todos estos sistemas disponían de mecanismos simples de evaluación y utilizaban ejercicios cuidadosamente preparados para ser evaluables. Estos sistemas, más que mejorar la calidad pedagógica de la evaluación, fueron vistos como mecanismos para evaluar de forma más rápida y a la vez evaluar a más alumnos. A menudo, esta automatización condujo a un cambio en el proceso de evaluación que se vio restringido por las limitaciones de la tecnología disponible. Los estudiantes tenían que presentar sus soluciones en una forma simplificada, de manera que pudieran ser interpretadas y evaluadas por el sistema.

A partir de los años noventa ya se ven reconocidas las implicaciones pedagógicas de la evaluación automatizada y se van extendiendo los sistemas que evalúan a partir de respuestas fijas. En esta línea nos encontramos con todas las plataformas de e-learning comerciales tipo WebCT [WebCT], [PH03], WebMCQ [WebMCQ], QuestionMark [Qmark], y las plataformas open-source, como por ejemplo Moodle [Moodle], .LRN [LRN] y las específicas de algunas universidades como el caso de Openmark [OpenM] de la Open University, QUEST [CBF07] de la PUC-Rio, WEAS [HB07], etc. Además, con el

auge de las TIC, se han desarrollado una gran cantidad de aplicaciones que permiten este tipo de corrección.

Tal como se ha comentado en la sección 2.2.2, entre los tipos de ejercicios para la evaluación de niveles cognitivos altos a partir de respuestas libres nos encontramos, con relación a este trabajo, la corrección de tareas de programación y la realización de diagramas.

Para la evaluación de las tareas de programación se han desarrollado también muchas plataformas. Una de las herramientas pioneras en la evaluación automática fue Ceilidh, que con las nuevas versiones cambió de nombre pasando a llamarse CourseMaster con mejoras muy significativas [FHG+01] [HHS+03] [HGS+05]. El handicap de esta herramienta es que requiere instalar un software determinado por parte del estudiante. Otros sistemas, entre ellos RoboProf [Dai99], Schema-Robo [SMK01], TRAKLA [KM00], BOSS [JGB05], EduComponents [APR06], etc. proporcionan herramientas completas para la evaluación automática en el contexto concreto de la corrección de programas informáticos en distintos lenguajes. Otros sistemas como Assyst [JU97], [Jac00] prefieren una evaluación semi-automática, en donde el sistema realiza gran parte de las tareas de corrección y finalmente es el tutor quien realiza las valoraciones más difíciles. Sin embargo, en la mayoría de los casos la evaluación automática cubre sólo una parte de los requisitos de evaluación de un curso. Por este motivo, otros autores [ES00] [Dal01] comentan la necesidad de una plataforma que integre diferentes tipos de corrección y que almacene los resultados en una única base de datos.

En referencia a la realización de diagramas, las herramientas CASE (Computer Aided Software Engineering) permiten la realización de una amplia variedad de diagramas y se utilizan en la enseñanza para que los estudiantes puedan aplicar la teoría y los conceptos básicos, pero están destinadas a un uso comercial y carecen de las funcionalidades de cualquier herramienta de e-learning. De la misma forma el software comercial Rational Rose [Qt99] de IBM permite generar plantillas de código C++ y Java a partir de los diagramas de clases. Sin embargo sus funcionalidades van dirigidas a desarrolladores de software y no están pensadas para fines educativos.

Uno de los primeros trabajos en la realización de diagramas es [Hir88] que ya en el 1988 describe una herramienta que permite a los alumnos realizar diagramas de flujo de datos, Entidad/Relación, etc. Su evaluación, sin embargo, se realiza de forma tradicional y considera que su evaluación automatizada es difícil. En la misma línea Power [Pow99] presenta un entorno llamado Designer que permite al alumno interactuar con sus diagramas.

En la línea de la evaluación de diagramas, [HL98] diseñaron un sistema interactivo de CBA, el cual combina una herramienta CASE que proporciona las funcionalidades de diagramación con un sistema Computer Assisted Learning (CAL) que proporciona las funcionalidades de ayuda al alumno. El mecanismo de verificación de los diagramas de los alumnos se basa en dos partes. En primer lugar el alumno va apareando de forma manual los elementos (nodos) del diagrama realizado con los de la solución correcta. En segundo lugar el sistema compara los nodos y sus conexiones con la solución correcta y muestra las diferencias, proporcionando el feedback oportuno al alumno. DATsys [Tsi02] es un entorno CBA para la realización de diagramas que proporciona el ciclo de vida completo de la evaluación automatizada mediante la integración en la plataforma CourseMarker. Utiliza el editor Daidalos para la realización de los diagramas que son utilizados en la evaluación de diseño lógico, diagramas de flujo y diseño orientado a objetos en un curso de herramientas de software. La puntuación de los diagramas se realiza en función de cada tipo de diagrama.

Otro trabajo relevante es el de Thomas [Tho04b] que presenta una herramienta de dibujo simple para permitir a los estudiantes dibujar diagramas como parte de un examen online. El diagrama es representado como un conjunto de nodos y los enlaces correspondientes. La mayoría de estudiantes fueron capaces de utilizar la herramienta a pesar de no estar familiarizada con ella y estar bajo la presión de un examen. Más tarde en [TWS05], [TWS07] y [WTS07] describen el trabajo realizado sobre corrección y evaluación de diagramas entidad/relación y en general sobre el modelado de datos.

Entre estos sistemas no hay ningún entorno que esté especializado en bases de datos y que permita una evaluación totalmente automatizada. Para ello se necesitan correctores de respuesta libre capaces de evaluar diagramas (entidad/relación, diagramas de clase, etc.) y que además sepan interpretar y evaluar consultas en álgebra relacional y SQL, parecidas en cierto modo a un programa informático. En este entorno, se han desarrollado aplicaciones que sólo cubren de forma independiente algún tema de la materia.

## 2.2.7 Puntuación automática

Otro de los aspectos importantes dentro de un entorno CBA es la técnica utilizada para la puntuación automática de los ejercicios. Hay varios factores que justifican la necesidad de la puntuación automática:

- En ejercicios muy extensos es comprensible que el alumno cometa pequeños fallos. Por este motivo no podemos calificar como incorrecto ejercicios con pequeños errores y habrá que ajustar la nota en función de ellos.
- El sistema universitario nos exige que expresemos con una nota el logro de competencias alcanzadas satisfactoriamente.
- Para la evolución formativa nos interesa que el alumno resuelva los ejercicios correctamente y que vaya aprendiendo de sus errores, del feedback obtenido y de las tutorías realizadas. Ahora bien, al utilizar un sistema CBA en el que el alumno libremente nos envía sus soluciones a través de Internet, debemos asegurar la autoría de los problemas realizados. Para ello debemos realizar exámenes/pruebas de validación presenciales. Si en estas pruebas presenciales utilizamos el mismo sistema CBA, nos interesará que disponga de funcionalidades para la puntuación automática. De esta forma el alumno sabrá inmediatamente si supera la prueba y el profesor no tendrá que corregir ningún examen, como mucho alguna posible reclamación.

En este marco de la puntuación automática, tanto [CEA+03] como [Tsi02] establecen categorías para las herramientas de puntuación automática en función del tipo de preguntas, distinguiendo, de forma análoga a la que hemos visto, en preguntas de respuesta fija y de respuesta libre.

En el entorno de Computer Science, la mayoría de sistemas de e-learning soportan la corrección y puntuación de preguntas de respuesta fija que pueden ser usadas para evaluar varios niveles cognitivos, especialmente los de conocimiento y comprensión [Lis01] o incluso niveles más altos [DK01b]. Este tipo de preguntas se ofrecen en plataformas como WebCT [Webct] ahora fusionada con Blackboard [Black] para el aprendizaje de entornos de software. Tal como describen [KTT+93] [FLM98] [DG99] [Qmark] las preguntas se pueden presentar de varias formas: elección múltiple, relleno de espacios en blanco, escritura de un texto o de un valor numérico, combinación de cualquiera de las anteriores, etc. Algunos entornos [FJM98] [The00] permiten generar distintos test a partir de permutaciones de conjuntos de preguntas o preguntas variables a partir de diferentes valores con el objetivo de individualizar las pruebas, evitando la memorización y el plagio e incrementando su dificultad. Muchos sistemas permiten calcular una nota a partir de las respuestas. En algunos casos sólo se valoran las respuestas correctas, en otros casos además se penaliza por las incorrectas. En todos los casos se puede dar una nota muy precisa, aunque a veces no representa fielmente el nivel de conocimiento del alumno, ya que por lógica e intuición el alumno puede descartar varias respuestas. Por otra parte el factor suerte puede ser determinante para que un alumno apruebe o suspenda una prueba. Las ventajas que tienen son su sencillez y que se pueden aplicar a cualquier materia, por ese motivo son las más utilizadas.

Los sistemas CBA, dentro del Computer Science, también utilizan preguntas de respuesta libre. Ahora bien los sistemas de puntuación de ejercicios de este tipo de preguntas son mucho más complejos que los de respuesta fija. Dentro de este tipo de respuestas tanto [CEA+03] como [Tsi02] distinguen tres tipos de preguntas:

**Ejercicios de programación.** Dentro de la categoría de preguntas de respuesta libre, los entornos que permiten la evaluación de ejercicios de programación son los más usados y extendidos. Desde hace varias décadas se han venido desarrollando sistemas que permiten la corrección de ejercicios de programación. En este caso, si el objetivo es evaluar el nivel de aplicación de la taxonomía de Bloom, los entornos CBA pueden ser útiles para evaluar el correcto funcionamiento de clases, funciones, etc. sin necesidad de desarrollar el programa completo [Bar99][Buc00][SMK01]. El CBA también puede evaluar el nivel de análisis. Por ejemplo se puede proporcionar a los alumnos programas que contienen errores y el trabajo del alumno consiste en analizar dónde están los errores y proporcionar las modificaciones para que el programa se pueda compilar y funcionar correctamente [Eng02]. También existen diferentes entornos que permiten evaluar programas completos. Los parámetros que se pueden evaluar van desde sus funcionalidades, su complejidad, estilo y eficiencia [HST02b][TR97]. Algunos de estos sistemas disponen de test de pruebas [MKS02][BPS+04] que permiten evaluar su correcto funcionamiento y a partir de aquí establecer una puntuación.

**Respuestas visuales.** Consideramos respuestas visuales a aquellas en que la respuesta se facilita a través de un gráfico, un diagrama, una imagen, un esquema, etc. y son muy importantes en la materia de bases de datos ya que en diseño de bases de datos se utilizan distintos tipos de diagramas en diseño conceptual a partir de los cuales se obtiene el esquema de la base de datos. Se pueden utilizar diferentes tipos de visualizaciones para los niveles cognitivos de comprensión, aplicación y análisis. Por ejemplo, se puede evaluar estructuras de datos y algoritmos [KM00]. De todas formas, estos entornos se suelen utilizar en materias donde tradicionalmente se usan gráficos o diagramas para la enseñanza de la materia (diagramas de flujo, de clases, ERD, etc.), como es el caso de bases de datos, ingeniería del software y programación [HL98] [BSD00].

Hay muy pocos trabajos que hagan referencia a la puntuación automática de diagramas. Los entornos mencionados en la sección 2.2.6 tratan este tema de la siguiente manera:

- En el presentado en [HL98] no puntúa los ejercicios.
- El entorno de CBA diseñado por la Universidad de Nottingham se ha ido desarrollando en varias fases y ha ido cambiando su nombre, Ceilidh, CourseMaster y en la actualidad CourseMarker [FHG+01]. En principio esta herramienta fue diseñada para la enseñanza de la programación y era capaz de puntuar los programas realizados por los alumnos [HGS+05]. A partir de ahí [Tsi02], [HST+02a] y [HG06] describen la extensión del CourseMaster para evaluar respuestas en forma de diagramas. Los aspectos más relevantes de su trabajo son: i) el sistema es capaz de presentar y evaluar ejercicios de diseño lógico, concretamente diagramas de flujo y diseño orientado a objetos. ii) para puntuar el diseño lógico se utiliza un simulador de circuitos, que se configura con un test de datos de prueba y el resultado esperado de salida. iii) para puntuar diagramas de flujo, el sistema convierte el diagrama en un programa que se ejecuta con un test de datos de entrada, comparando la salida con los resultados esperados. iv) para el diseño orientado a objetos, la respuesta del alumno es tratada según las características predefinidas, tal como el uso de relaciones específicas entre las clases predefinidas. El handicap de esta herramienta es que no ofrece puntuaciones parciales de soluciones muy parecidas a la correcta, simplemente proporciona el feedback adecuado en caso de error. Las soluciones son correctas o incorrectas.
- El entorno de [BH07] trata de un sistema semi-automático de puntuación de diagramas. La principal funcionalidad de esta herramienta es agrupar las respuestas de los estudiantes de forma que segmentos idénticos de las respuestas queden identificados y agrupados. A partir de aquí el tutor sólo tiene que puntuar una sola vez estos segmentos iguales y completar la corrección de forma manual.
- Finalmente el sistema de la Open University es desde nuestro punto de vista el más completo en el apartado de la corrección y puntuación de diagramas. Desde hace mucho tiempo han investigado acerca de la puntuación automática de exámenes en los cuales los alumnos podían responder de forma similar a un texto libre y no de forma restringida tipo elección múltiple [TPP+01] [TPP+02]. Para ampliar la tipología de preguntas proponen un sistema de puntuación de diagramas [Tho04a]. A partir de ahí, [STW04] se centra en el tratamiento de la imprecisión de los diagramas y [Tho04b] comenta los aspectos prácticos de la utilización de una herramienta de dibujo en un examen, aunque no aportan mucho en el tema de la puntuación. En [WTS04] se describe el proceso de puntuación para el caso concreto de los diagramas ER. El sistema de puntuación se basa en el concepto de unidades mínimas significativas (minimal meaningful units MMU) definidas como las partes que tienen un significado específico. Por ejemplo en un diagrama Entidad/Relación tanto las entidades como las relaciones son MMU's. La puntuación se realiza en varias etapas, la primera dedicada a la identificación de las MMU's. Una vez identificadas se pasa a la fase de interpretación donde se compara la solución enviada con la ideal. El objetivo de estas comparaciones es intentar aparear las MMU's del estudiante con las de la solución ideal. Como no se restringe el nombre de las entidades ni de las relaciones el sistema intenta aparear los nombres de las entidades y relaciones, utilizando las ideas del procesamiento del lenguaje natural (NLP). Las entidades son identificadas con nombres y las relaciones como expresiones verbales, intentado la detección de sinónimos. Una vez identificadas y relacionadas con la solución modelo, el sistema dispone de un esquema de puntuación. Aunque sin precisar, los autores comentan que han ideado un algoritmo parametrizado de puntuación, de difícil comprensión, pero adecuado para las tareas de puntuación. Basado en medidas de similitud el algoritmo puntúa cada característica identificada. Finalmente en varios artículos realizan la evaluación del sistema desarrollado. En [Tho04c] se comparan las notas obtenidas por profesores y las obtenidas automáticamente por el sistema y en

[TWS05] se relatan los experimentos realizados que muestran unos buenos resultados. Otros artículos de los mismos autores [TSW06] [TSW07] [TSW08] amplían sus trabajos en la puntuación automática de diagramas. En estos trabajos confirman los buenos resultados obtenidos en las puntuaciones automáticas respecto a las manuales sobre una muestra de cerca de 400 diagramas. Por otra parte también exponen las herramientas para poder puntuar de forma independiente cada uno de los componentes del diagrama.

El handicap de esta herramienta es que sólo está pensada para evaluar diferentes tipos de diagramas y no ofrece ninguna prestación para la corrección de otros temas tan importantes en materia de BD como las consultas SQL o el álgebra relacional. Tampoco permite la evaluación de esquemas de bases de datos relacionales ni normalización. Por otra parte estos trabajos están encarados a la puntuación automática, despreocupándose de la evaluación formativa a llevar a cabo durante todo el curso.

**Respuestas textuales.** El alumno escribe un texto libre como respuesta a las preguntas formuladas. Según como estén diseñadas servirán para evaluar distintos niveles de conocimiento. Hay varios enfoques para soportar la evaluación y puntuación automática de respuestas libres en forma de texto. Para su evaluación, algunos entornos de aprendizaje se basan en una comparación directa de los textos o expresiones regulares. También hay enfoques más sofisticados y complejos como los propuestos en [Chr99] [BLS01] [MG02] en los que las técnicas de puntuación automáticas a las respuestas de este tipo de preguntas se basan generalmente en diferentes tipos de prototipos y generalmente no están totalmente automatizados. Una serie de técnicas sobre este tipo de ejercicios son descritas en [AB05].

Como valoración final a la puntuación automática, comentar que varios autores [Tsi02] [CEA+03] coinciden en que no todos los niveles cognitivos de la taxonomía de Bloom son fáciles de evaluar de forma automatizada. Las preguntas de respuesta fija nos pueden servir especialmente para los niveles inferiores, ahora bien para los niveles cognitivos superiores, en donde se requiere evaluar aplicación, análisis y síntesis, se requiere de herramientas muy especializadas, especialmente si se quiere evaluar habilidades de diseño, de resolución de problemas, de realización de programas y algoritmos, etc.

Las ventajas que ofrece la puntuación automática son:

- Se reduce considerablemente el tiempo de corrección.
- Es muy apropiado para la enseñanza a distancia.
- Las evaluaciones pueden hacerse más frecuentemente permitiendo evaluar fragmentos más pequeños de la materia impartida en el curso.
- La puntuación se realiza de forma rápida y consistente.
- Se puede proporcionar feedback consistente y de alta calidad.
- Puede ayudar en la detección y prevención del plagio. La detección puede ser implementada comparando las respuestas de los alumnos. La prevención puede ser implementada asignando ejercicios de forma aleatoria a los alumnos a partir de un amplio repositorio de problemas de forma que cada alumno disponga de ejercicios diferentes.

Los inconvenientes que presenta son:

- Se necesitan muchos recursos para su implementación y especialmente para la generación de ejercicios.
- La rigidez de una evaluación automatizada. La mayoría de entornos no saben distinguir entre respuestas totalmente incorrectas de las que casi son correctas. Un ejemplo de esta situación es cuando un determinado problema debe resolverse a través de una serie de pasos. En una corrección manual generalmente se aplica una pequeña penalización si se detecta un pequeño error en las primeras etapas, siempre y cuando las deducciones/razonamientos posteriores sean correctos, aunque el resultado final sea incorrecto. Es muy difícil aplicar este tipo de corrección de forma automática. Otro ejemplo lo tenemos en la corrección de diagramas. Un sistema automatizado puede considerar el ejercicio incorrecto, cuando simplemente hay un pequeño error respecto a la solución ideal entrada por el profesor.
- Son pocos los tipos de problemas de respuesta libre que se pueden puntuar de forma automática. La falta de herramientas que permiten la puntuación de preguntas de respuesta libre es un gran handicap que impide su extensión.

Finalmente comentar que la mayoría de los sistemas CBA en Computer Science están focalizados en el aprendizaje de la programación. Según el estudio realizado por [CEA+03] se desprende que son muchos los encuestados que afirman utilizar distintos entornos para la evaluación de los niveles cognitivos superiores en materia de programación. Con la puntuación obtenida se permite evaluar, al menos parcialmente, al alumno. Ahora bien, en los últimos años se han ido desarrollando técnicas que permiten tanto el aprendizaje como la puntuación de distintos tipos de diagramas con lo que el ámbito del CBA se puede extender a otras materias.

## 2.2.8 Resumen

En la sección 2.2.1 hemos definido la tecnología del CBA en términos de automatización de todos los procesos que intervienen en la evaluación. En teoría el CBA ofrece un gran potencial de ventajas para el alumno y para el profesor. El alumno sabe que puede utilizar libremente el sistema siempre que desee, obteniendo el feedback más adecuado en cada momento. Si el sistema está bien diseñado puede ser agradable y especialmente motivador para el alumno. Para el profesor le permite tutorizar y evaluar un gran número de alumnos sin un incremento considerable de trabajo ya que el sistema le facilita las funcionalidades para la monitorización del progreso de cada alumno. Consideramos que la investigación en CBA ofrece la posibilidad de ahorrar tiempo y esfuerzo a los profesores facilitando un sistema fiable y robusto para la evaluación de los alumnos.

Sin embargo, en la práctica el CBA se usa mayoritariamente en el entorno científico/técnico y casi siempre a través de preguntas de selección múltiple o tipo test y ofrece feedback a medida en relativamente pocos casos. Entre las razones de esta práctica limitada podemos citar que existe una cierta antipatía al CBA ya que se percibe como un sistema impersonal y aceptable sólo para la evaluación de ciertas tareas de un bajo nivel cognitivo y que requieran de respuestas fijas.

En estos momentos en que las plataformas de e-learning tienen un papel muy destacado en las universidades y que la evaluación automatizada va adquiriendo cada vez más importancia, los sistemas CBA se van especializando cada vez más para poder evaluar niveles cognitivos superiores. Como veremos en la sección 2.3 en un curso de bases de datos necesitamos herramientas capaces de evaluar los niveles cognitivos superiores y difícilmente podemos hacerlo con preguntas de elección múltiple. En esta línea se enmarca esta tesis, centrándonos en el desarrollo de un entorno web especializado en materia de bases de datos, que permita realizar una gran variedad de actividades de aprendizaje que serán evaluadas de forma automática.

## 2.3 DOCENCIA EN BASES DE DATOS

Analizando los conceptos de evaluación y de Computer Based Assessment (CBA) hemos visto que para evaluar de forma automatizada los niveles más altos de la taxonomía de Bloom se necesitan herramientas especializadas según la materia en que se van a aplicar. En nuestro caso nos centraremos en la materia de bases de datos (BD), materia que consideramos el tercer pilar de nuestra investigación (ver Figura 2.7). Así en este tercer bloque presentaremos el contenido y la metodología habitual de un curso básico (asignatura) de BD (sección 2.3.1). Posteriormente nos centraremos en los temas/competencias más importantes del curso (secciones 2.3.2 y 2.3.3), para finalmente hacer un análisis de las herramientas de e-learning más destacadas para el aprendizaje/enseñanza/evaluación de los temas más relevantes en esta materia (sección 2.3.4).

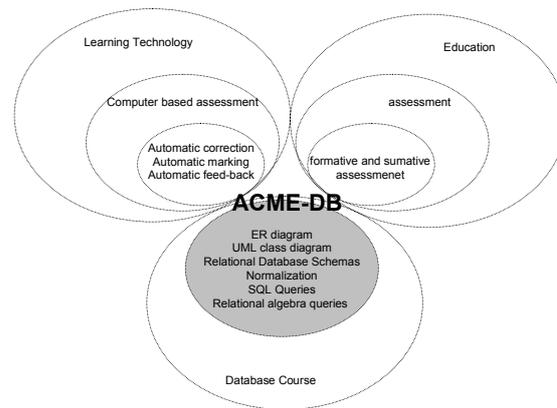


Figura 2.7 Marco de la tesis. Bases de datos.

### 2.3.1 Contenido y metodología de una asignatura de bases de datos

Los modelos curriculares más prestigiosos en computación especificados por la ACM/IEEE/AIS [CurACM] coinciden en la importancia de la gestión de la información en cada una de las cinco disciplinas:

- Computer Engineering [CCE04]
- Computer Science [CCS08]
- Information Systems [CIS02]
- Information Technology [CIT08]
- Software Engineering [CSE04]

En cada una de estas disciplinas se facilitan las directrices básicas de un curso de BD que permiten introducir las herramientas para la gestión de la información. Sin lugar a duda, podemos asegurar que las BD son una de las materias más importantes en las carreras de Informática. Independientemente de la disciplina, las asignaturas básicas de BD tienen un contenido común. Tal como se corrobora en el estudio [RR03] y en otros artículos en la misma línea [RWG+00][SRR00] en un curso básico de BD se imparten, entre otras, materias relacionadas con el diseño de BD, herramientas de consulta y conceptos de administración, gestión y funcionamiento de un Sistema de Gestión de Bases de Datos (SGBD). Por otro lado entre los docentes hay una coincidencia sobre los temas que consideran más relevantes en un curso de BD: SQL, diseño de bases de datos (modelo entidad/relación, modelo relacional y normalización) y también otros temas como bases de datos en Internet y gestión de transacciones [RR03] [MD07] [BM03]. Todos estos conceptos son los que encontramos en la bibliografía especializada en bases de datos, siendo [EN07], [CB09], [Dat04], [SKS05], [Pos05], [RG03],[Kro04] y [RC04] la más relevante.

En cursos avanzados de bases de datos también se vuelve a incidir en alguno de estos temas [Sey02] [UD01], especialmente en el modelado y diseño de bases de datos, incluyendo el modelado conceptual a través de diagramas de clase UML [BRJ99].

La metodología docente más tradicional usada en las distintas universidades para la impartición de esta asignatura suele ser muy parecida y está disponible en las páginas web de sus respectivas universidades, entre ellas [FIB][UPM][UPV]. Esta metodología consiste en unas sesiones teóricas, complementadas por unas sesiones de laboratorio. Generalmente los alumnos deben realizar trabajos individuales y en grupo. Esta metodología también es constatada por algunos responsables de estas asignaturas en diferentes artículos [Njo03][HD03][EN03]. El uso de herramientas informáticas es cada día más aceptado como complemento a las sesiones presenciales [NS04], ya sea simplemente para practicar o también para evaluar. También se fomenta el desarrollo de proyectos en grupo como un sistema de aprendizaje colaborativo, intentando integrar teoría y práctica [DU98]. Dentro de esta metodología, otros autores [Ull03] sugieren la automatización de los trabajos de casa y exámenes, de forma que su corrección y puntuación anime al trabajo y al aprendizaje de los alumnos.

A continuación describiremos el contenido y la metodología docente en diseño de bases de datos y en las consultas sobre éstas, por ser las materias principales en un curso de BD. También se describen diferentes propuestas para mejorar la docencia de BD. Comentar que no haremos mención a otros temas que se suelen impartir, como por ejemplo los módulos que componen un Sistema de Gestión de Bases de Datos (SGBD), estructuras de accesos a BD, transacciones, concurrencia, etc. Todos estos temas suelen ser muy descriptivos y a nivel de evaluación se pueden utilizar ejercicios de respuesta fija, de muy fácil corrección y que no se tratan en esta tesis.

### 2.3.2 Diseño de bases de datos

En una organización, el sistema de bases de datos suele ser una parte del llamado sistema de información, que incluye todos los recursos que participan en la obtención, administración, uso y distribución de los recursos de información de la organización. Estos recursos incluyen tanto el hardware como el software utilizado, pasando por todos los usuarios que de alguna manera u otra intervienen en el sistema de información (programadores, administrador de la base de datos, usuarios finales, etc.). El entorno software comprende las aplicaciones finales, el SGBD, los propios datos, etc. Así pues, el sistema de bases de datos es sólo una parte del sistema de información de una organización.

En este entorno uno de los puntos clave es el diseño de la BD, que podemos definir como las tareas a desarrollar para diseñar la estructura conceptual, lógica y física de la BD para atender a las necesidades de información de los usuarios de una organización según los requisitos definidos para sus aplicaciones.

Los objetivos del diseño de la base de datos son múltiples:

- Satisfacer los requisitos de los usuarios y aplicaciones.
- Proporcionar una estructuración de la información que sea fácil de entender.
- Soportar los requisitos de procesamiento y de rendimiento.

Generalmente el proceso de diseño de bases de datos suele ser laborioso y complicado. Este problema se agrava si tenemos en cuenta que a menudo se comienza con unos requisitos muy informales y poco definidos, mientras que el resultado del diseño de una BD debe terminar con un esquema perfectamente definido y que no se podrá modificar fácilmente una vez implementado. Es por este motivo que el proceso de diseño de una base de datos debe realizarse de forma rigurosa y dada su importancia es fundamental que el alumno adquiera unas buenas habilidades [Wat06]. En este proceso se siguen las etapas siguientes:

- **Obtención y análisis de requisitos.** Es la fase inicial y consiste en realizar el análisis de los requisitos que debe satisfacer la aplicación. En esta fase el diseñador debe comprender el dominio de la aplicación que le servirá de base para el desarrollo del diseño conceptual y debe analizar los requisitos para establecer relaciones entre los distintos elementos que conforman la base de datos.
- **Diseño conceptual de la base de datos.** El modelo conceptual de la BD es una descripción a alto nivel de la BD que posteriormente se transforma en un modelo lógico el cual puede ser

implementado en un SGBD. La calidad del diseño conceptual de la BD es de vital importancia y por este motivo los cursos dedican una buena parte de su tiempo a su enseñanza. Para ello se desarrolla el modelo conceptual utilizando generalmente diagramas basados en el modelo entidad/relación [Che76] o el modelo entidad/relación extendido [TYF86]. Estos modelos consisten en determinar las diferentes entidades que intervienen en nuestro sistema y como interactúan entre sí (relaciones). El resultado de esta fase es la obtención de un diagrama entidad/relación (ERD). También y con el uso de metodologías orientadas a objeto y a través del lenguaje UML (Unified Modeling Language) [BRJ99] se pueden utilizar los diagramas de clase para representar conceptos similares.

- **Elección de un SGBD.** En un curso básico se suele elegir cualquier SGBD que siga el modelo relacional, por ser éste el de uso más generalizado.
- **Transformación del modelo conceptual al modelo de datos elegido** (en nuestro caso al modelo relacional). El modelo relacional fue introducido por Codd [Cod70] y representa la base de datos como una colección de relaciones matemáticas. En términos más sencillos cada relación se asemeja a una tabla de valores. Para pasar del diseño conceptual (p.e. a partir de un ERD) al diseño lógico (p.e. el modelo relacional) simplemente se tendrán que aplicar varios pasos [EN07]. El resultado final de esta transformación (mapping) será el esquema de la base de datos en el modelo relacional, donde se detallan las diferentes tablas y atributos, especificando cuáles van a ser las claves principales y externas (foráneas).
- **Diseño físico de la base de datos.** Su objetivo es crear la estructura adecuada de los datos para garantizar un buen rendimiento de la BD.
- **Implementación y ajuste del sistema de bases de datos.** Se crea la BD y empieza su explotación. Periódicamente hay que revisar y ajustar su estructura si es necesario.

En la Figura 2.8 se muestra las diferentes fases del diseño de una base de datos y los resultados que se obtienen.

## DISEÑO DE BASES DE DATOS

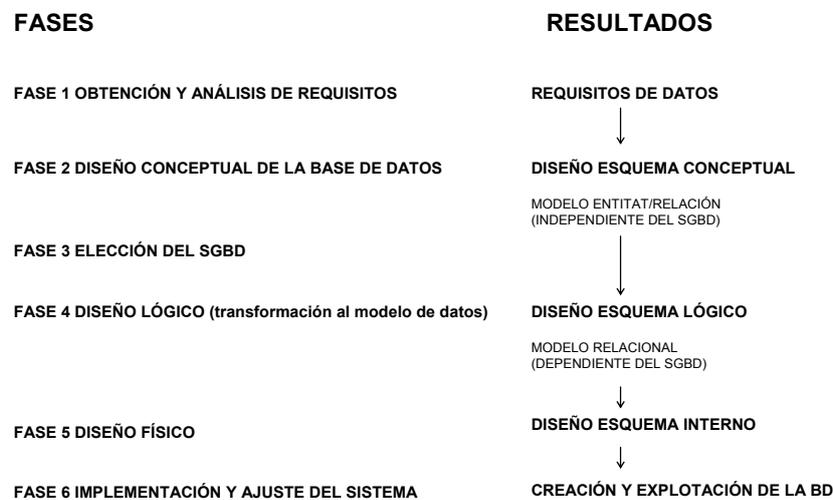


Figura 2.8 Fases y resultados obtenidos en el diseño de bases de datos.

Desde el punto de vista docente, para la enseñanza del desarrollo de sistemas de información, los planes de estudio de las Ingenierías Informáticas de las distintas universidades han elaborado planes de estudio en las que tradicionalmente han separado la parte referente a los datos (asignaturas de Bases de Datos) de las del desarrollo del sistema de información en sí (asignaturas de Ingeniería del Software). Así pues, en nuestro caso las competencias referentes al diseño de bases de datos, se imparten en la asignatura del mismo nombre.

El aprendizaje del diseño de bases de datos con un alto grado de calidad es una de las competencias de cualquier curriculum en Informática. Como hemos visto, el diseño de una base de datos comprende varias etapas, siendo la primera la obtención de los requisitos. Esta tarea puede ser un trabajo a realizar por el alumno o bien el profesor puede detallarlos en el enunciado del problema. El alumno debe comprender

perfectamente el entorno y por este motivo los enunciados de los supuestos sobre los que debe trabajar deben ser concisos y entendedores. La siguiente etapa consistirá en realizar el diseño conceptual, siendo la metodología más usada la realización de un ERD. Para una determinada situación pueden haber distintos diagramas válidos, la solución no es única. No hay ningún algoritmo que nos permita obtener un ERD a partir del listado de requisitos.

El método tradicional de enseñanza de ERD se basa en unas sesiones en la que se explica el modelo. Para que el alumno adquiera la competencia es necesario el desarrollo de un número considerable de supuestos parecidos a los que se encontrará en su entorno profesional. La tutorización de este trabajo requiere sin duda un esfuerzo considerable por parte de los profesores que difícilmente pueden atender de forma individual y continuada al alumno. De aquí surge la necesidad de desarrollar tutores virtuales que ayuden y ofrezcan feedback al alumno en el proceso de aprendizaje y entrenamiento del diseño conceptual y lógico de bases de datos

Consideramos que el aprendizaje del diseño conceptual es un paso crucial y que si el alumno sabe realizar los ERD, le será muy fácil desarrollar el diseño lógico asociado, en nuestro caso utilizando el modelo relacional. Una vez obtenido el esquema de la base de datos debe verificar que el sistema desarrollado cumpla unas normas de calidad. Para ello y aplicando la teoría de la normalización de bases de datos, se comprobará que esté normalizado hasta una determinada forma normal. Generalmente, es suficiente con comprobar que las relaciones (tablas obtenidas con el modelo relacional) cumplan hasta la tercera forma normal (3FN) [Cod71] o hasta la forma normal de Boyce-Codd (BCF) [Cod74]. Aunque las dos últimas fases (la realización del diseño físico y la implementación y ajuste del sistema) son también importantes, desde un punto de vista docente, se suelen ver de forma más rápida ya que al ser conocimientos menos requeridos, más especializados y más relacionados con el SGBD usado se da más preferencia a los otros temas.

### 2.3.3 Consultas en bases de datos

Un modelo de datos, como por ejemplo el modelo relacional, debe definir un conjunto de operaciones para manipular la BD junto con los conceptos necesarios para la definición de su estructura y restricciones. De esta forma, para la manipulación de una BD relacional y la realización de todo tipo de consultas se han definido diferentes tipos de lenguajes de consulta, como por ejemplo:

- **El álgebra relacional.** El conjunto de operaciones básicas del modelo relacional es el álgebra relacional que permite al usuario especificar las peticiones fundamentales de recuperación de datos. El resultado de una recuperación (consulta) es una nueva relación, por consiguiente las operaciones del álgebra relacional siempre producen nuevas relaciones que pueden ser manipuladas de nuevo usando las mismas operaciones del álgebra. Una secuencia de operaciones de álgebra relacional constituye una expresión de álgebra relacional. En esta secuencia se va detallando en que orden hay que aplicar las diferentes operaciones sobre cada tabla para llegar al resultado final. Se trata pues de un lenguaje procedural en el que hay que describir el orden de ejecución de las operaciones. El álgebra relacional es muy importante por varias razones:
  - Proporciona un fundamento formal para las operaciones del modelo relacional.
  - Se utiliza de base para la implementación y optimización de consultas en los SGBD relacionales.
  - Algunos de sus conceptos se han incorporado al lenguaje estándar de consultas SQL.

Aunque ninguno de los SGBD comerciales actuales proporcionan una interfaz para las consultas de álgebra relacional, las funciones y operaciones centrales de cualquier sistema relacional están basadas en estas operaciones.

- **El cálculo relacional.** Otro de los lenguajes formales de consulta es el cálculo relacional. Éste ofrece una notación declarativa de alto nivel para especificar las consultas relacionales. Una expresión de cálculo relacional crea una nueva relación, la cual está especificada en términos de variables que engloban filas o columnas de las relaciones almacenadas en la base de datos. A diferencia del álgebra relacional, en una expresión de cálculo relacional, no existe un orden en que aplicar las operaciones para obtener los resultados de la consulta ya que la expresión sólo especifica la información que el resultado debería contener. El cálculo relacional se basa en la lógica matemática y es un lenguaje declarativo.

- **Structured Query Language (SQL).** Una de las razones del éxito comercial de las bases de datos relacionales es el lenguaje de consulta estándar SQL (Structured Query Language). Este lenguaje se convirtió rápidamente en un estándar para este modelo. Este hecho propició la migración de otros modelos de datos (jerárquicos, en red) al modelo relacional. Aunque los distintos SGBD relacionales sean muy distintos, si todos utilizan las funciones que forman parte del estándar, el cambio y conversión de un SGBD relacional a otro no presenta problemas. A lo largo de los años se han ido realizando varias versiones de SQL, como SQL1 (o SQL86) y SQL2 (o SQL92). El siguiente estándar SQL99 es una versión mejorada del anterior SQL2 y ahora es el que más se utiliza. Se han propuesto otros estándares como SQL3, pero no han tenido el suficiente respaldo por parte de la industria.

A pesar de la importancia del álgebra relacional que ya hemos comentado, sus operaciones son consideradas muy técnicas por la mayoría de los usuarios de un SGBD comercial, ya que una consulta se escribe como una secuencia de operaciones que al ejecutarse, producen el resultado final. Por lo tanto, el usuario debe especificar el orden en que hay que ejecutar las operaciones de la consulta. En cambio, SQL proporciona una interfaz de lenguaje declarativo de alto nivel, en el que el usuario sólo especifica lo que quiere obtener, dejando para el SGBD la optimización y las decisiones de cómo ejecutar la consulta. Aunque SQL incluye algunas características del álgebra relacional, está basado en gran medida en el cálculo relacional de tuplas. Sin embargo, la sintaxis de SQL es mucho más amigable para el usuario que el álgebra y el cálculo relacional.

Por otra parte, la creación, el mantenimiento (inserción, modificación, eliminación) y la obtención de resultados (consultas) de los datos de una base de datos a través de SQL, son tareas que el alumno debe aprender perfectamente, ya que éstas forman parte de una de las competencias básicas que debe adquirir un graduado en informática.

Desde un punto de vista docente, tal como queda reflejado en el informe [RR03], la mayoría de docentes de bases de datos imparte conocimientos de álgebra relacional y especialmente de SQL. Además coinciden que para una buena comprensión de SQL es importante tener conocimiento del álgebra relacional. Respecto a la introducción al cálculo relacional se le da menos importancia, por ser también un lenguaje declarativo, parecido en cierta manera a SQL.

### 2.3.4 Innovación y mejoras en la enseñanza de Bases de datos

En la sección 2.3.1 hemos visto la metodología y los contenidos típicos de un curso de BD. Ahora bien, con los avances de las TIC y en especial con el auge de Internet se han realizado muchas experiencias innovadoras y se han propuesto distintos métodos para mejorar y/o complementar la metodología tradicional.

De esta forma, diferentes autores proponen métodos de aprendizaje basados en proyectos. En esta línea [DU96] propone trabajar a partir de proyectos que son desarrollados por grupos en trabajo cooperativo. En [MP03] proponen complementar las sesiones teóricas con un entreno práctico basado en proyectos y utilizan WebCT como herramienta virtual de estudio. [Rob00] propone el desarrollo de proyectos que son divididos en varias partes, intentando simular las tareas realizadas en una empresa y [MAB05] propone el aprendizaje a partir del trabajo con mini-proyectos sobre bases de datos existentes y luego deducir y aprender a diseñar sus propias bases de datos, en lo que llama un ciclo de vida inverso. En la misma línea de motivar a los alumnos en temas que les sean más cercanos y de su interés, en [WDW+09] se propone a los alumnos el desarrollo de un proyecto “Metube” que es una versión simple y modificada del popular Youtube (<http://www.youtube.com>). El proyecto consiste en el desarrollo tanto de la base de datos como de la correspondiente programación del entorno web. Sus autores aseguran que los estudiantes alcanzan los objetivos de aprendizaje fijados de forma mucho más eficiente a través de este tipo de proyectos.

En [BO03] se mezclan los métodos tradicionales con el entorno virtual de aprendizaje Blackboard VLE y utilizan CBA para la evaluación, a través de preguntas tipo test, de los temas de teoría desarrollados en clase (50% de la nota). El resto de la nota se obtiene a partir de la evaluación de SQL, también a partir de diferentes tests. En [Har04] se utiliza el blended learning, para la evaluación de los distintos módulos. Para ello utilizan distintos tests bajo la plataforma WebCT. Estos tests son usados para la evaluación formativa, proporcionando un feedback detallado a los alumnos. Finalmente se usa una versión de estos

tests para la evaluación sumativa que cuenta un 50% del total de la evaluación. En [Bal03] además de las clases presenciales, proponen una serie de pruebas semanales de autoevaluación tipo test a través de WebCT. Además utilizan diferentes herramientas para dar soporte a las clases: EsqI [KSF97], Winrdbi [DEP97][Winrdbi], SQL-Tutor [Mit98a] [Mit03], Kermit [SM02] [SM04] and Normit [Mit02]. Comentan que el mayor inconveniente de estas herramientas es que son independientes y que los alumnos deben familiarizarse con cada una de ellas. Por otro lado comentan que alguna de estas herramientas, como SQL-Tutor desaniman a los estudiantes porque no permiten respuestas que difieran de las establecidas.

Otro sistema novedoso y pensado para evaluar el grado de comprensión de la materia de BD en las clases teóricas es el presentado en [PMO04] en donde los alumnos disponen de un mando a distancia con el que seleccionan el número de respuesta correcta entre las mostradas de un determinado ejercicio. A través de un PRS (Personal Response System) que recoge las respuestas seleccionadas, profesor y alumnos visualizan en forma de diagrama que repuestas han sido enviadas y los % correspondientes. Los alumnos saben si se han equivocado y pueden ver su nivel respecto al resto de la clase. En [PBK04] se presenta el sistema multimedia IDLE (Interactive Database Learning Environment) que consiste en varios recursos educacionales en forma de clases (en formato texto y audio), tutoriales (en formato de animaciones) y laboratorios virtuales (mediante distintos entornos interactivos) que permiten la realización de diagramas entidad relación, su transformación al modelo relacional, consultas SQL, etc. También permite ejercicios de auto-evaluación. Otros sistemas como OTC (On-line Testing Center) [Ull03] sugieren la automatización de los trabajos y exámenes. El método propuesto sólo puntúa soluciones de ejercicios de selección múltiple. En [RVV+07] se propone QUEST (Quest Environment for Self-managed Training) un entorno competitivo entre grupos de alumnos que van obteniendo puntos. Tanto alumnos como profesores puede proponer nuevas actividades que son corregidas y puntuadas por el profesor. Los alumnos ven las puntuaciones de cada grupo, con lo que se pretende motivarlos a través de la competición que se establece entre ellos. Finalmente en [JND+09] se recogen algunas ideas en la enseñanza, aprendizaje y evaluación de bases de datos.

En el aprendizaje del diseño de bases de datos juegan un papel importante las herramientas para la realización de diagramas (por ejemplo de diagramas de clase, entidad/relación, etc). Algunas de estas herramientas son productos comerciales de propósito general para el diseño de bases de datos, como por ejemplo Microsoft Visio (<http://office.microsoft.com/visio>). Esta herramienta está diseñada para los profesionales vinculados al proceso de desarrollo y uso de diferentes sistemas. Es una de las herramientas más usadas, pero si consideramos su complejidad seguramente no es la más adecuada para estudiantes sin experiencia. Por otro lado, también hay herramientas de software libre más reducidas, como por ejemplo DeZign for Databases disponible en <http://www.datanamic.com/dezing/index.html> o bien Toad Data Modeler disponible en <http://www.casestudio.com/enu/default.aspx>. Ambas soportan diagramas entidad/relación y proporcionan una integración total con los más populares SGBD (Oracle, MySQL, DB2, etc.).

En [MG09a][MG09b] se describe AdbC (Animated Database Courseware) un software con un conjunto de animaciones diseñado para dar soporte a la enseñanza del diseño de bases de datos, que incluye la descripción de modelo ER, diagramas ER, el mapeo al modelo relacional y normalización. Según los autores estas animaciones facilitan el aprendizaje de la materia. Para complementar los métodos tradicionales de enseñanza de diseño de bases de datos en [CSM06] sus autores proponen juegos de ordenador para proporcionar un entorno colaborativo de aprendizaje.

Con referencia al diseño de bases de datos, [CSM06] comenta que las estrategias pedagógicas tradicionales para la enseñanza del análisis y diseño de bases de datos son parecidas a otras del campo de las ciencias de la computación, como por ejemplo la enseñanza de la programación informática. Estas estrategias consisten en la impartición de muchos conocimientos teóricos por parte del profesor, convirtiendo a los alumnos en meros espectadores totalmente pasivos, mientras que los alumnos tienden a asimilar mejor unos conceptos básicos con un gran componente práctico. En esta misma línea [CB06] critica este tipo de metodología (modelo objetivista) frente al modelo constructivista que propone que la enseñanza es un proceso activo en donde los nuevos conocimientos son adquiridos a partir de los conocimientos previos y los problemas a solucionar. Así pues no es de extrañar que una de las áreas en que los alumnos tienen dificultades sea el análisis y diseño de bases de datos. Por estos motivos y según se corrobora en el informe europeo [CL04] la principal deficiencia en materia de BD que las empresas/organizaciones constatan en los egresados de Tecnologías de la Información es el diseño de BD. En esta misma línea [Phi07] comenta las áreas de confusión en la enseñanza del diseño de BD y cómo se

pueden solucionar y [Ven96] hace un estudio de las deficiencias de los diseñadores novatos frente a los expertos y las estrategias para superar estas dificultades, entre ellas la práctica y experiencia en entornos muy distintos.

En vista de todas estas deficiencias se proponen cambios en las metodologías docentes que sin duda pasan por las TIC. Se constata en [CSM06] que las generaciones actuales ya han crecido en este entorno en que parte de sus aprendizajes, juegos, ocio se ha realizado a través de dispositivos tecnológicos avanzados (Ordenadores con conexión a Internet en las aulas de primaria y secundaria, Nintendo, Xbox, teléfonos móviles, etc.) y que muestran una predisposición a aprender y trabajar con las TIC's.

Por otra parte y desde un punto de vista más teórico, en [MS00] se proponen las actividades más adecuadas según cada uno de los seis niveles de aprendizaje de la taxonomía de Bloom [Blo56] (conocimiento, comprensión, aplicación, análisis, síntesis y evaluación) para el diseño de bases de datos. Para las dos primeras etapas se proponen actividades tipos test, selección múltiple, etc. a través de cualquier herramienta de e-learning. Ahora bien a partir de las dos etapas iniciales, para los dominios de aplicación y análisis, se proponen actividades relacionadas con:

- el mapeo de diagramas ER al modelo relacional.
- miniproyectos en donde el alumno debe aplicar los conocimientos adquiridos.
- analizar situaciones en donde aplicar sus conocimientos.

En este aspecto, comenta las dificultades para disponer de herramientas que faciliten la evaluación en los niveles superiores de la taxonomía. En [Nee07] se comenta que el diseño de bases de datos es tanto un arte como una ciencia, situando el diseño de una base de datos en el dominio de análisis según la taxonomía de Bloom, ya que el alumno debe recordar los conocimientos adquiridos, debe comprender los modelos y debe saber aplicarlos a una nueva situación.

Como se ha visto previamente, el diseño de bases de datos no es una tarea fácil y por este motivo se han propuesto muchos sistemas de mejora [FDK05] y se han desarrollado muchos sistemas innovadores para mejorar su enseñanza. Entre ellos citar Relational Assistant (RA) que es uno de los sistemas expertos pioneros para el aprendizaje del diseño de bases de datos [MPW95]. En [BM99] se detallan los elementos que se requieren para evaluar la resolución de problemas en entornos tecnológicos.

### **2.3.5 Resumen**

La materia de bases de datos es una de las más importantes en cualquier curriculum de informática. Todas las universidades imparten este tipo de asignaturas siendo los temas del diseño de BD y la realización de consultas sobre éstas los más importantes.

El diseño de bases de datos se realiza a través de varias etapas, siendo el diseño conceptual y el diseño lógico las más importantes. En la docencia del diseño conceptual se suele utilizar el modelo Entidad/Relación o también diagramas de clases. Una vez el alumno ha realizado el diagrama correspondiente el siguiente paso es la realización del diseño lógico, siendo el modelo relacional el más usado. El objetivo final es que el alumno debe ser capaz de obtener el esquema de la BD y comprobar que esté normalizado.

Las competencias que el alumno debe adquirir acerca de las consultas sobre BD hacen referencia a que debe ser capaz de realizar cualquier sentencia SQL, especialmente la sentencia SELECT para obtener la información referente a cualquier consulta. Sin duda alguna en el ámbito profesional se le exigirá un buen conocimiento de SQL. Académicamente y para una mejor comprensión de SQL y del funcionamiento de las consultas se aconseja previamente la realización de consultas a través de álgebra relacional.

En los últimos años se han propuesto diferentes sistemas para mejorar la docencia en esta materia. Muchas de estas propuestas pasan por incorporar sistemas de e-learning con los que poder realizar diferentes actividades de aprendizaje. Nuestra propuesta va en esta línea, incorporando además las funcionalidades necesarias para poder realizar la evaluación de forma totalmente automatizada

## 2.4 ESTADO DEL ARTE EN HERRAMIENTAS PARA EL APRENDIZAJE Y EVALUACIÓN EN MATERIA DE BD

En esta sección nos centraremos en describir el estado del arte en materia de entornos CBA para el aprendizaje y evaluación de las materias más importantes de un curso de bases de datos. Tal como se ha comentado en la sección 2.3 estos temas son: diagramas ER, diagramas de clase, esquemas de BD relacionales, normalización, álgebra relacional y SQL (ver Figura 2.9).

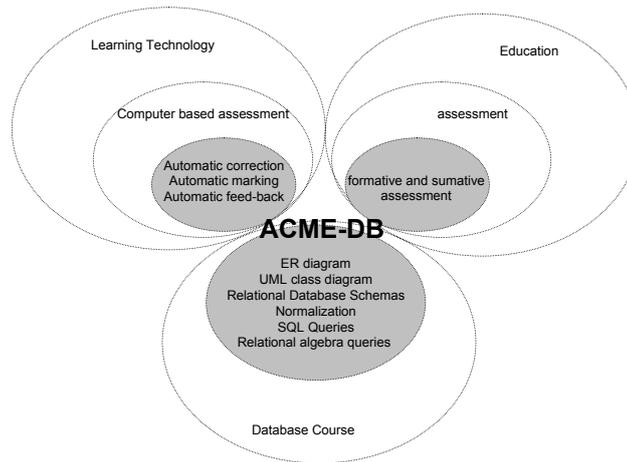


Figura 2.9 Entorno de la tesis.

Para cada uno de estos temas se describirá el estado del arte en herramientas que faciliten el aprendizaje y evaluación en estas materias. Puesto que los entornos CBA especializados en cada uno de estos temas son muy pocos, se extenderá este estado del arte a otras herramientas que por su interés en facilitar el aprendizaje del tema se consideran también importantes, aunque no las podamos clasificar como CBA. Con el fin de homogeneizar nuestro estudio para cada tema se realizará una tabla comparativa, en la que se considerarán los parámetros siguientes:

- **Entorno integrado para la docencia de bases de datos.** Valoramos muy positivamente que los distintos correctores se integren en un único entorno.
- **Web Based tool.** Para facilitar el trabajo al alumno es importante que sólo requiera de una conexión a Internet y un navegador, sin ninguna necesidad de descarga de ninguna aplicación ni de su instalación.
- **Corrección y feedback automáticos.** De caras a la evaluación formativa, es muy importante que el sistema corrija automáticamente y ofrezca el feedback oportuno al alumno.
- **Puntuación automática del ejercicio.** Pensando en la evaluación sumativa del alumno, es deseable que la plataforma ofrezca una puntuación de cada ejercicio lo más parecida posible a la que obtendría un profesor en una corrección no automatizada.
- **Evaluación automática del alumno.** El mismo entorno debe facilitar la evaluación final de todo el trabajo realizado por el alumno, facilitando la tarea del profesor.
- **Trabajo colaborativo.** El entorno debe ofrecer funcionalidades que permitan el trabajo en grupo y de forma colaborativa. Las nuevas metodologías docentes muestran los buenos resultados que se obtienen con este tipo de trabajo.
- **Necesidad de software adicional.** Es importante que para el funcionamiento de la herramienta no se necesite software adicional. Por ejemplo en algunas aplicaciones educativas se requiere además la compra/instalación de un SGBD, de software específico para la realización de diagramas, etc.

Veremos que la mayoría de estas herramientas sólo están pensadas como un entorno para practicar, para facilitar el aprendizaje, pero no ofrecen las prestaciones de un CBA.

## 2.4.1 Correctores de diagramas entidad/relación.

El modelo Entidad/Relación (ER) [Che76] es un modelo conceptual de alto nivel ampliamente utilizado y que se basa en identificar y representar en un diagrama las entidades que intervienen en el sistema, sus relaciones y las restricciones entre éstas. En los últimos años han aparecido distintas herramientas para el soporte de la enseñanza y el aprendizaje de los temas más relevantes de bases de datos. Uno de los primeros entornos fue DBTool [LH92] que creó un entorno sencillo para la representación de Diagramas Entidad/Relación (ERD) a partir del que se podía obtener el correspondiente esquema de la base de datos relacional. A partir de este trabajo se han ido desarrollando distintos entornos, la mayoría de los cuales se presentan como herramientas independientes, no integradas a ninguna plataforma de e-learning. Se trata de entornos especializados que se deben descargar e instalar antes de ser usados. Los trabajos más relevantes en herramientas para el aprendizaje del modelo ER son los que se citan a continuación:

- Hall y Gordon [HG98] proponen ERM-VLE (Entity Relationship Modelling in a Virtual Learning Environment). Es un entorno virtual de aprendizaje basado en comandos tipo texto. Utilizan el paradigma Multi User Dimensions (MUD) usado en muchos juegos como una forma de desarrollar un entorno virtual de aprendizaje basado en notaciones textuales. A partir de un problema dado, los alumnos realizan su modelo ER escribiendo los comandos textuales que permiten navegar y manipular objetos por diferentes compartimentos. Disponen de un restringido conjunto de comandos para interactuar con los diferentes objetos de este espacio virtual y que permiten crear, mover, desplazarse, ir, etc. A través de estos objetos y de los comandos para su manipulación se va construyendo el diagrama entidad relación. El alumno puede ir pidiendo verificaciones de lo que va realizando y en caso de error respecto a la solución propuesta, el sistema lo detecta e informa al estudiante que la asociación no es correcta. Los comentarios de los propios autores afirman que los inconvenientes de la plataforma son su rigidez y que es más apropiada para los alumnos que empiezan desde cero que para los que ya tienen conocimientos de modelado de bases de datos. El tener que utilizar comandos textuales es otro inconveniente ya que las herramientas actuales se basan en funcionalidades gráficas.
- Constantino y otros [CS00] proponen COLER (Collaborative Learning environment for Entity Relationship Modeling). Es un entorno colaborativo de aprendizaje en el que los alumnos construyen de forma individual un diagrama ER de un problema dado. A partir de aquí trabajan en pequeños grupos con el objetivo de consensuar una solución de grupo. Los alumnos disponen de un espacio de trabajo propio y uno de grupo. El sistema les anima a compartir y discutir sus soluciones y muestra las diferencias entre su solución y la final consensuada por el grupo. Se basa en el aprendizaje colaborativo y su finalidad es desarrollar estas habilidades. Una de las limitaciones que tiene es que no corrige de forma automática los diagramas ER desarrollados y requiere la participación de un tutor experto.
- Antony y Batra [AB02] presentan CODASYS. Se trata de un prototipo de herramienta de consulta para diseñadores de bases de datos inexpertos. Está pensada para ayudar a realizar diagramas ER que puedan ser transformados en esquemas de bases de datos normalizadas, libres de dependencias derivadas. Se basa en estudios previos de los mismos autores basados en analizar los errores más frecuentes que realizaban los diseñadores inexpertos y la mejora que se obtiene en el diseño conceptual a través del feedback [BA94] [BS93]. En este prototipo aplican una metodología propia de diseño de bases de datos basada en reglas [BZ94]. Tal como sus autores afirman, CODASYS es un prototipo para diseñadores inexpertos y no es apropiada para el diseño de bases de datos un poco complejas. Por ejemplo no soportan relaciones unarias, sólo permiten una relación entre dos entidades, etc.
- Suraweera y Mitrovic en [SM02] y en otros artículos de los mismos autores sobre el mismo tema, presentan el KERMIT, sistema de tutorización inteligente (Intelligent Tutor System) sobre el cual han desarrollado un entorno de resolución de ERD's que el alumno debe construir a partir de un conjunto de requisitos. Sus autores afirman que no es un corrector propiamente dicho ya que consideran que un corrector de este tipo es de extrema dificultad, especialmente en el tema de identificar el lenguaje natural con el que se han de identificar entidades y especialmente atributos. El sistema compara las soluciones de los alumnos con

la solución ideal entrada por el profesor, basada en el conocimiento del dominio y representada como un conjunto de reglas que sirven para comparar la solución del alumno con la ideal. Cada una de estas reglas está formada por un indicador de relevancia (más o menos importancia), la condición a cumplir y el mensaje asociado en caso de violación de la regla. Algunas de estas reglas permiten identificar errores de sintaxis y las hay más simples (por ejemplo se debe escribir el nombre de las entidades en Mayúsculas) y más complejas (la participación de una entidad débil en una relación con su entidad regular debe ser total). Dispone también de reglas semánticas capaces de hallar construcciones equivalentes a la solución ideal para interpretar la solución del alumno. El mayor handicap de esta herramienta es que para establecer las correspondencias entre los componentes del diagrama y las de la solución ideal, siempre que se añade un nuevo componente en el diagrama, el alumno se ve forzado a subrayar la palabra que representa una entidad o un atributo y en el caso de las relaciones la frase correspondiente. Estas correspondencias son usadas para activar la producción de reglas usadas en la corrección que son comparadas con las de la solución ideal. El feedback se genera si se viola alguna de estas restricciones.

- El sistema DATsys en [Tsi02] [HST02a] proporciona un entorno CBA para la realización de diagramas a través de CourseMarker. Actualmente se han extendido sus capacidades de corrección automática, con el proceso de corrección automático de ERD's. Han desarrollado comandos específicos para puntuar los diagramas ER y herramientas para su dibujo. El método de corrección se basa en identificar los distintos tipos de nodos, los textos asociados y los enlaces entre estos nodos. A esta estructura se pasan diferentes test (“exist”, “exact”, “connection”, “compositeRelationship”,...) para determinar el grado de corrección del diagrama y su puntuación.
- El proyecto DEAP (Diagrammatic Electronic Assessment Project) [WTS04] [TWS07] [WTS07] de la Open University busca soluciones para el aprendizaje a distancia y la evaluación automática de exámenes. En el ámbito del diseño de bases de datos han desarrollado un sistema formado básicamente por tres herramientas: Diagram Exerciser Revision, SimpleDraw y Exercice Builder. La primera herramienta Exerciser contiene las preguntas sobre modelado de datos que son presentadas como el guión a seguir. Los estudiantes dibujan el diagrama correspondiente mediante el editor Simple Draw. Finalmente Diagram Exerciser puede puntuar el diagrama y proporcionar distintos niveles de feedback. El sistema está diseñado como una herramienta para practicar siguiendo las técnicas de modelado de datos. Cuando un profesor entra un guión, debe a continuación entrar el diagrama y mapear cada elemento del diagrama con la parte relacionada de la pregunta. Esta información es guardada por el sistema y utilizada para la corrección. Para la puntuación automática se basa en la interpretación de diagramas imprecisos [STW04] y los resultados iniciales que han obtenido muestran que es posible obtener una puntuación automática de diagramas ER. No son entornos web ni tampoco están integrados en ninguna herramienta de e-learning. No permiten el trabajo en grupo.
- El proyecto ABC (Assess by Computer) [TSW05], al igual que el proyecto DEAP, está focalizado en la tecnología para la puntuación de cualquier tipo de diagramas (por ejemplo, diagramas ER, diagramas de clase, circuitos eléctricos, etc.) que se pueden representar mediante cajas y conectores, o sea como si se tratara de un grafo. Un grafo puede ser comparado con otro por similaridad usando el concepto de isomorfismo de grafos y una aproximación heurística para comparar y evaluar el grado de similaridad de dos diagramas. Esta idea ha sido desarrollada también en el proyecto DEAP de la Open University y también por el LTR Research Group de la Universidad de Nottingham. En este caso los autores no buscan una puntuación totalmente automática sino presentar la información de forma que el profesor visualiza las partes que son exactamente igual a la solución correcta y el profesor sólo debe corregir las partes que son distintas. Evidentemente esta técnica semi-automática facilita el trabajo del profesor. Se utiliza este método para evaluar la similaridad entre dos diagramas cualquiera, pero no pretende ser un método de aprendizaje de diseño conceptual de bases de datos, sino un método para la puntuación de los ejercicios. Los autores manifiestan que en pruebas realizadas sobre exámenes reales se han producido algunos resultados no esperados. No proporciona ningún tipo de feedback.

- Batmaz y Hinde [BH07] [BSH09] proponen un sistema semiautomático de evaluación de diagramas consistente en agrupar segmentos iguales de las soluciones de los alumnos y entonces el sistema pregunta al profesor que verifique su corrección. De este modo una vez verificado un segmento, de forma automática se puntúa de forma igual a todos los estudiantes que tienen igual el mismo segmento. Esto es importante para realizar una evaluación objetiva en los cursos con muchos alumnos y en los que hay que corregir y evaluar gran cantidad de ejercicios. En cierto modo es parecido al proyecto ABC, con la diferencia de que éste comprueba que los elementos del diagrama sean iguales al de referencia, mientras que el sistema de Batmaz, al igual que Kermit, se comprueba a través de las referencias que ligan el texto del problema con el diagrama y que son guardados por el sistema. Una vez verificados los distintos segmentos, el proceso de corrección queda simplificado considerablemente. Las agrupaciones automáticas de segmentos son calificadas de forma automática por el sistema. Estos segmentos pueden ser entidades, relaciones o agrupaciones de ellas. Las entidades son consideradas iguales si tienen el mismo nombre y el mismo número de atributos. Proponen un editor de diagramas para facilitar el proceso de corrección semiautomático.
- #EER [BSB+08] es una herramienta diseñada para facilitar el diseño de bases de datos a los alumnos, automatizando algunos de los procesos. Está basada en el modelo Entidad Relación Extendido (EER) y permite representar diagramas EER. Una vez dibujado el diagrama, el sistema realiza de forma automática la generación del esquema de BD relacional equivalente y también puede generar los comandos DDL (Data Definition Language) para generar la base de datos. Está pensada para que los alumnos que sólo realizan un breve curso introductorio de bases de datos puedan generar los comandos necesarios para la creación de la base de datos diseñada. No es una herramienta de corrección automática, ni tampoco ofrece ningún tipo de feedback al alumno.

La Tabla 2.3 ofrece una comparativa resumida de los principales aspectos que estamos evaluando de las distintas plataformas. De las herramientas citadas, ERM-VLE ha quedado desfasada, debido principalmente a su interfaz textual. COLER está pensada como una herramienta de aprendizaje colaborativo y su handicap es que no corrige automáticamente. Una de las más destacadas es KERMIT, que está en la línea de nuestro trabajo, aunque su función principal es más el aprendizaje/enseñanza que no la puntuación automática. Los proyectos ABC, DEAP y DATsys están más enmarcados en la línea de evaluación/puntuación automática (automatic marking) para facilitar las tareas de corrección al profesor y no están tan encaradas al autoaprendizaje o como herramientas para facilitar el blended learning. Aunque en todos estos casos las herramientas desarrolladas se han concebido con distintas finalidades, los problemas con que se encuentran son parecidos: identificar el lenguaje natural e identificar diagramas equivalentes a uno o varios de correctos.

MODELO ENTIDAD RELACIÓN								
	ERM-VLE	COLER	KERMIT	DATSYS	DEAP-EXER	ABC	Batmaz	#EER
CARACTERÍSTICAS / ENTORNOS								
Entorno integrado para la docencia de bases de datos	no	no	si	no	no	no	no	parcial
Web-based tool	no	si	si	no	no	no	no	no
Editor de diagramas propio	si text	si	no	si	si	si	si	si
Corrección automática	no	no	si	si	si	semi	semi	no
Puntuación automática del ejercicio	no	no	no	si	si	semi	semi	no
Evaluación automática del alumno	no	no	no	si	si	semi	si	no
Trabajo colaborativo	no	si	si	no	no	no	no	no
Necesidad de software adicional	no	no	si	no	no	no	no	no
Proporciona feed-back automático	si	si	si	si	si	no	no	no

Tabla 2.3 Comparativa de herramientas de aprendizaje de diagramas ER.

## 2.4.2 Correctores de diagramas de clase

Los diagramas de clase nos describen los distintos objetos de un sistema y las relaciones estáticas que hay entre ellos. Se utilizan para modelar el esquema conceptual de la base de datos. En los últimos años han aparecido distintas herramientas para el soporte de la enseñanza y el aprendizaje de los temas más relevantes en el uso y corrección de diagramas de clase. Tal como veremos, la mayoría de ellos se presentan como una herramienta independiente, no integrada a ninguna plataforma de e-learning. Se trata de entornos especializados que se deben descargar e instalar antes de ser usados. Los trabajos más relevantes en herramientas para el aprendizaje y corrección de diagramas de clase son los que se citan a continuación:

- Hoggarth y Lockyer en [HL98] proponen un sistema basándose en la integración de herramientas CASE (Computer Aided Software Engineering) y herramientas CAL (Computer Assisted Learning). Se propone un sistema de evaluación de diagramas (en general) en el que parte de la corrección se realiza de forma manual, de forma que una vez realizado el diagrama se presenta al alumno una relación de los objetos usados en su diagrama y los de la solución correcta con la que se debe comparar. El alumno debe aparear estos objetos de forma manual y una vez realizada esta operación el sistema de comparación genera la lista de mensajes con el feedback más apropiado. No se evalúan los diagramas con ninguna nota.
- Baghaei, Mitrovic y Irwin en distintos artículos [BM05] [BMI06] presentan COLLECT-UML. Es un sistema de tutorización inteligente (Intelligent Tutor System) sobre el cual han desarrollado un entorno de resolución de diagramas de clase. Inicialmente desarrollaron una versión monousuario en la que el alumno debe desarrollar un diagrama de clases a partir de unos requisitos iniciales. El sistema compara las soluciones de los alumnos con una única solución ideal entrada por el profesor. El sistema se basa en el conocimiento del dominio que es representado como un conjunto de reglas. En el proceso de comparación se utilizan estas reglas para determinar el feedback a enviar al alumno. Cada una de estas reglas está formada por un indicador de relevancia (más o menos importancia), la condición a cumplir y el mensaje asociado en caso de violación de la regla. Posteriormente desarrollaron la versión que soporta aprendizaje colaborativo. En esta versión la pantalla se divide en dos partes principales: en una el alumno puede realizar sus diagramas de forma individual y en la otra es donde se realiza el trabajo colaborativo y donde los miembros del grupo van aportando sus ideas. El moderador del grupo puede enviar el diagrama a corregir y el sistema devuelve el feedback oportuno. COLLECT-UML no permite a los alumnos definir sus propios identificadores para clases, atributos y métodos. Por ejemplo el nombre de una clase debe escogerlo de una frase subrayada en el enunciado del problema. La decisión de usar un determinado nombre ya viene determinada por el sistema. Una interfaz de este tipo restringe el concepto de identificación y de asignación de nombre a clases y atributos.
- Blank y otros en [BPW+05] presentan CIMEL-ITS (Collaborative, constructive, Inquiry-based Multimedia E-Learning) un método para dar soporte al aprendizaje del análisis y diseño orientado a objetos a través de mejorar sus habilidades con la resolución de problemas. El sistema dispone de un componente de evaluación que observa las entradas del alumno (como por ejemplo el nombre de las clases, de cada atributo y de cada método) y intenta mapearlas con la de una solución de las correctas entradas por el profesor y si es el caso con los posibles errores previstos por el profesor. Las soluciones correctas y la relación de posibles errores son entrados por el profesor. Si una entrada del alumno no coincide con ninguna de las correctas y puede mapearse con uno de los errores previstos, el módulo de evaluación entiende que la solución es incorrecta. Una de las limitaciones de este sistema es que se anticipa a las acciones de los estudiantes, determinando las posibles acciones correctoras y los posibles errores por anticipado.
- Virvou and Tourtoglou en [VT06] presentan ASSET (Adaptative Support to a Software Engineering Trainer) como un entorno adaptativo para la construcción de diagramas de UML. Se basa en un entorno colaborativo en donde hay dos tipos de usuarios, el entrenador (experto) y los alumnos. El sistema ayuda a los alumnos a localizar al experto que mejor puede ayudarle en cada momento. El sistema está constantemente monitorizando el trabajo

del alumno y recoge sus características para clasificar a los alumnos según su personalidad y su nivel de conocimiento. Cada usuario es asignado a un grupo de trabajo con un rol específico (desarrollador junior, desarrollador senior, etc.) según su nivel de conocimiento. No es un corrector automático propiamente dicho sino un entorno que proporciona ayuda al entrenador y facilita ayuda a los estudiantes.

- Nguyen-Thanh Le en [Ngu06] ha realizado una extensión de la herramienta ArgoUML, que permite la realización de diagramas UML, para conseguir un sistema de aprendizaje que permite diseñar diagramas de clase sin ninguna restricción en cuanto a nombre de las clases y atributos (free-form). El sistema se basa en un módulo de evaluación del diagrama de clases a través de unas pautas de diseño que el alumno debe seguir. Enviada una solución por parte del alumno, el sistema la mapea con la solución ideal facilitada. El sistema en primer lugar identifica las clases y los atributos según el sistema de identificación de nombres en las frases y con listas de categorías de conceptos según Larman [Lar01]. Para ello cada clase de la solución ideal dispone de una lista de posibles nombres a mapear con la solución del alumno. Si alguna clase del alumno no se puede mapear es considerada superflua y si una clase de la solución ideal no se puede aparear con ninguna del alumno se considera un error. Las clases superfluas, no identificadas por el sistema, se las intenta mapear con las clases de la solución ideal a través de sus asociaciones. Una vez identificadas las clases se identifican las asociaciones. Se consideran correctas si las clases relacionadas coinciden. El problema del método es que de momento sólo corrige el nombre de las clases y sus asociaciones, pero no evalúa ni atributos, ni métodos, ni navegación, con lo cual suponemos que no se está utilizando en un entorno real.
- Haji Ali. N. y otros [HSI07a] [HSI07b] presentan UCDA (UML Class Diagram Assessor). Este entorno no dispone de un editor propio de diagramas de clase y los alumnos deben utilizar Rational Rose [Rat] como herramienta de edición del diagrama. Así pues la entrada es siempre un fichero obtenido a partir de Rational Rose y la salida es una lista de comentarios como guía al estudiante. El sistema de comparación consiste en extraer la información del fichero .mdl generado por Rational Rose y compararla con la del profesor.

La Tabla 2.4 ofrece una comparativa de los principales parámetros que estamos evaluando. Desde nuestro punto de vista la herramienta que ofrece mejores prestaciones es COLLECT-UML aunque no permita la puntuación automática. CIMEL-ITS detecta los errores a medida que se van produciendo. Desde nuestro punto de vista, consideramos que guía en demasía al alumno y da muy poca independencia al alumno. ASSET no es un sistema automático y necesita de un tutor experto que es quien ofrece el feedback. Es un sistema pensado para dar formación a las personas que se integran en una empresa de desarrollo de software. UCDA tiene el inconveniente que se necesita disponer de Rational Rose para la realización de los diagramas. La propuesta de la extensión de ArgoUML es quizás la más innovadora ya que no tiene ninguna restricción sobre los nombres de los distintos elementos que conforman el diagrama. Ahora bien, de momento no permiten ciertas funcionalidades habituales de un diagrama de clase. Uno de los problemas de estas herramientas es que están encaradas al aprendizaje/enseñanza de los diagramas de clase, permitiendo una evaluación formativa, pero no realizan ningún tipo de puntuación automática de los ejercicios realizados por los alumnos.

Existen además otro tipo de herramientas que permiten validar esquemas UML. Son herramientas que no están pensadas para ser usadas en un entorno de aprendizaje y que no validan si los diagramas son correctos desde un punto de vista semántico sino más bien si son correctos desde un punto de vista sintáctico, es decir si se puede instanciar. Estas herramientas permiten que el alumno pueda ver si sus esquemas conceptuales están correctamente escritos. Entre este tipo de herramientas citar a [Use] y [CCR07].

UML	H O G G A R T H	C O L L E C T	C I M E L	A S S E T	A r g o U M L	U C D A
<b>CARACTERÍSTICAS / ENTORNOS</b>						
Entorno integrado para la docencia de bases de datos	no	no	no	no	no	no
Web-based tool	no	si	no	no	no	no
Editor de diagramas propio	no	si	no	si	no	no
Corrección automática	semi	si	si	no	parcial	si
Puntuación automática del ejercicio	no	no	no	no	no	no
Evaluación automática del alumno	no	no	no	no	no	no
Trabajo colaborativo	no	si	si	si	no	no
Necesidad de software adicional	si	no	si	no	si	si
Proporciona feed-back automático	si	si	si	no	no	si

Tabla 2.4 Comparativa de herramientas de aprendizaje de diagramas de clase.

### 2.4.3 Correctores de esquemas de bases de datos relacionales

Se han desarrollado muy pocas herramientas que permitan la corrección de esquemas de bases de datos relacionales. Uno de los motivos es que desde un punto de vista académico se suele dar más importancia al diseño conceptual que al diseño lógico. El paso de uno a otro consiste en mapear los diferentes elementos del diagrama e ir creando las tablas necesarias, especificando su clave principal, sus claves externas y sus posibles restricciones. Por otra parte muchas herramientas disponen de utilidades que automáticamente crean el esquema de la base de datos a partir del diagrama ER. Desde un punto de vista pedagógico consideramos que es muy importante que el alumno sepa construir el esquema de la BD a partir de un diagrama conceptual. El trabajo más relevante en este apartado es:

- El desarrollado por el grupo encabezado por A.Mitrovic [MM04] [MMM06] llamado ERM-Tutor a través del cual los alumnos practican los pasos para transformar un diagrama ER al correspondiente esquema de la base de datos. ERM-Tutor es una aplicación web enmarcada dentro de los sistemas de tutorización Inteligentes (ITS Intelligent Tutor System). El tutor dispone de un conjunto de problemas y el sistema se basa en un conjunto de 121 reglas que representan el dominio del conocimiento. El proceso de corrección se divide en varias tareas que se corresponden con los pasos de la transformación del modelo ER al modelo relacional. Un alumno puede escoger cualquier problema que se presenta como un diagrama entidad relación. Con este diagrama entidad relación el sistema permite ir aplicando los distintos pasos de la transformación. Por ejemplo el alumno puede escoger el paso 1 (transformación de las entidades regulares) y tendrá que ir escribiendo las tablas que se obtengan de cada entidad. En todo momento puede ir comprobando si el trabajo realizado es correcto. Dispone de una interfaz apropiada para seguir cada uno de los pasos y también dispone de un área de “feedback” y otra de ayuda. Internamente el sistema guarda tanto la información del diagrama ER como información textual basada en cuatro listas para guardar entidades, relaciones, atributos y conexiones. Los inconvenientes que vemos en el ERM-Tutor son:
  - Ofrece un conjunto igual de problemas a todos los estudiantes
  - Parte siempre de un diagrama entidad-relación, no de un enunciado de un supuesto dado.
  - Los alumnos pueden ir obteniendo feedback a cada momento. Consideramos que es mejor realizar por completo el problema y luego enviarlo a corregir.
  - Sólo es utilizado para el auto-aprendizaje, no para la evaluación, con lo que no ofrece ninguna puntuación del ejercicio.

Hay otros artículos referentes a este tema, pero en unos casos se trata de simples prototipos y en otros se hace referencia a los requisitos que debería tener un sistema de este tipo, pero no son herramientas desarrolladas, ni tampoco contemplan la corrección/evaluación automática. En la Tabla 2.5 se reflejan las características de ERM-Tutor.

<b>MODELO RELACIONAL</b>	
	<b>E R M - T U T O R</b>
<b>CARACTERÍSTICAS / ENTORNOS</b>	
Entorno integrado para la docencia de bases de datos	si parcial
Web-based tool	si
Editor de diagramas propio	si
Corrección automática	si
Puntuación automática del ejercicio	no
Evaluación automática del alumno	no
Trabajo colaborativo	no
Necesidad de software adicional	no
Proporciona feed-back automático	si

Tabla 2.5 Comparativa de herramientas de aprendizaje de esquemas de BD.

#### 2.4.4 Correctores para la normalización de esquemas de BD.

La normalización de bases de datos fue introducida por E.Codd [Cod71] a principios de los 70 poco después de crear el modelo relacional [Cod70]. La normalización pretendía añadir calidad a los diseños de BD que seguían el modelo relacional, evitando la aparición de datos redundantes y los problemas de actualización que conllevan. La normalización de las relaciones (tablas) de una BD supone comprobar una serie de normas, llamadas formas normales, y a continuación descomponer las relaciones que no las cumplan para obtener al final un esquema normalizado. La mayoría de herramientas que tratan el tema de normalización de bases de datos no han sido concebidas como entornos de aprendizaje y autoevaluación sino que están pensadas como aplicaciones para automatizar el proceso de normalización. Algunas de estas herramientas pueden también ser utilizadas para el aprendizaje de la normalización, aunque carecen de todas las funcionalidades de una plataforma de e-learning (gestión de usuarios, de actividades, de evaluación, etc). Entre estas herramientas podemos citar: Micro [DW99], Database Normalization tool [Sel03], Database Normalizer [Elm04] y Normalizer [Arm06].

Entre los entornos enfocados a la docencia y con algunas de las funcionalidades que estamos evaluando podemos citar:

- NOCAT (NORmalisation CAse Tool) [Pou95] ofrece tres partes completamente diferenciadas: el tutorial, el proceso de normalización y la normalización automática. El tutorial introduce los conceptos de normalización y proporciona preguntas de selección múltiple a los alumnos que cubren la parte más teórica de la normalización, suministrando, en caso de error, el feedback apropiado para cada pregunta. El proceso de normalización facilita a los alumnos sesiones interactivas que permiten normalizar los esquemas facilitados o los propios del alumno. Estos ejercicios siempre consisten en facilitar un supuesto y una relación universal. A partir de aquí y tanto a través de preguntas tipo selección múltiple, como a través de un pequeño editor para entrar las dependencias básicas se va guiando al alumno hasta obtener el esquema normalizado, facilitando el feedback oportuno en cada momento. Finalmente la normalización automática consiste en facilitar una relación universal, la clave principal y las dependencias funcionales básicas y a partir de aquí el alumno obtiene directamente las relaciones resultantes del proceso

de normalización. Como en otros casos se trata de herramientas que guían al alumno en el proceso de normalización.

- Mitrovic en [Mit02] y en otros artículos similares propone NORMIT, en la misma línea que el Kermit y el ERM-Tutor. NORMIT se basa en la filosofía de sistemas ITS que se fundamenta en el modelado basado en restricciones (Constraint-based Modelling CBM). En este entorno, el alumno selecciona un problema y a partir de aquí va avanzando por una serie de pasos para analizar la calidad del esquema de bases de datos presentado. El alumno va siguiendo una serie fija de acciones: determinar las claves candidatas, las dependencias funcionales, los conjuntos de atributos primos y no primos, etc. Cada una de estas tareas se realiza en una página web distinta. Las respuestas de los estudiantes son analizadas en cada paso y el sistema facilita el feedback correspondiente. En entorno de trabajo es un cuestionario en el cual el alumno va respondiendo a las preguntas y el sistema le notifica si la pregunta es correcta o no. El sistema está pensado para ser una herramienta de ayuda en el proceso de aprendizaje, no como una herramienta para puntuar y evaluar a un alumno. No incluye funcionalidades para ayudar al profesor en la evaluación de las actividades de aprendizaje.
- Fong y otros [FKN+02] presentan un sistema que se basa en ir navegando por distintas interfaces web en las que se van analizando cada una de las formas normales y en donde el alumno va aprendiendo el proceso de normalización. No es un corrector, sino una herramienta que va guiando al alumno en este proceso.
- Kung y otros en [KT02][KT06] presentan una herramienta web en la que los alumnos visualizan una relación universal y sus dependencias funcionales. Disponen básicamente de dos botones: normalizar y paso a paso. El primero sirve para normalizar la relación presentada y el resultado es la visualización de las relaciones resultantes ya normalizadas. El segundo botón va realizando el proceso de normalización paso a paso hasta llegar al resultado final, utilizando una técnica alternativa desarrollada en [KC04]. Tampoco corrige y simplemente visualiza el proceso de normalización aplicado en una relación universal.
- Zhang y otros en [ZKK05] presentan un prototipo basado en la idea de un entorno de gestión del conocimiento (Knowledge Management) adquirido con la experiencia en la enseñanza del tema. Las interfaces de usuario que presentan se dividen en dos partes. En una se seleccionan las tareas de normalización y en la otra se visualizan los conceptos de la normalización de bases de datos. Igual que la anterior es una herramienta para facilitar el aprendizaje de la materia y no un corrector para este tipo de materias.
- JMathNorm [YK07] es un entorno que permite definir atributos y establecer dependencias funcionales de forma que el alumno puede experimentar con las tareas de normalización. El sistema facilita las relaciones normalizadas según las dependencias funcionales entradas. JMathNorm usa el software Mathematica y no corrige los ejercicios, simplemente realiza la normalización para que el alumno pueda ver el resultado final.
- FDTutor [ECZ08] es un sistema inteligente de tutorización basado en la web para el aprendizaje y trabajo de dependencias funcionales que son la base de las tres primeras formas normales y la de Boyce-Codd (FNBC). Proporciona ejercicios de dependencias funcionales a los alumnos y determina cuando un alumno ha realizado un error, proporcionándole el feedback correspondiente. También ofrece al alumno una serie de estadísticas correspondientes al trabajo realizado y comparaciones con el resto del grupo. El sistema guarda también toda la información del trabajo realizado por el alumno de forma que el profesor puede obtener todo tipo de estadísticas. También puede detectar la tipología de errores más frecuentes para facilitar las indicaciones oportunas en las clases teóricas. De igual modo puede obtener toda la información de un alumno o las estadísticas de resolución de un problema. Como inconveniente observamos que sólo permite realizar ejercicios de dependencias funcionales, pero en ningún caso se pueden realizar tareas de normalización.
- LDBN (Learn Database Normalization) [Geo08] es un entorno web en el que el alumno selecciona los ejercicios de una lista establecida. Cada ejercicio consiste en un esquema de BD relacional en forma de una única relación universal, con los atributos y dependencias funcionales correspondientes. A partir de aquí debe obtener un esquema normalizado, especificando las

claves correspondientes. El alumno puede enviar una solución parcial o total y el sistema analiza la solución verificando la corrección de la descomposición. Al usuario se le muestra el resultado y en caso de error el sistema ofrece feedback en forma de pequeñas pistas, indicando donde está el error.

La Tabla 2.6 ofrece una comparativa resumida de los principales aspectos que estamos evaluando de las distintas plataformas. Como se puede observar son pocas las plataformas de e-learning que permitan la corrección automática de ejercicios de normalización y que faciliten el feedback necesario para ayudar al alumno. Excepto en NORMIT, FDTutor y LDBN, las demás son herramientas que ayudan y guían en el proceso de aprendizaje de la normalización de bases de datos. NORMIT va corrigiendo las preguntas que a modo de cuestionario se hacen al alumno y que le van guiando en el proceso de normalización. FDTutor corrige ejercicios de dependencias funcionales y LDBN corrige el resultado de un proceso de normalización.

NORMALIZACIÓN								
	N O C A T	N O R M I T	F O N G	K U N G	Z H A N G	J M a t h N o r m	F D T U T O R	L D B N
CARACTERÍSTICAS / ENTORNOS								
Entorno integrado para la docencia de bases de datos	no	si parcial	no	no	no	no	no	no
Web-based tool	no	si	si	si	si	si	si	si
Editor propio	si	si	si	si	si	si	si	si
Corrección automática	no	si	no	no	no	no	si	si
Puntuación automática del ejercicio	no	no	no	no	no	no	no	no
Evaluación automática del alumno	no	no	no	no	no	no	si	no
Trabajo colaborativo	no	no	no	no	no	no	no	no
Necesidad de software adicional	no	no	no	no	no	si	no	no
Proporciona feed-back automático	si	si	no	no	si	no	si	si

Tabla 2.6 Comparativa de herramientas de aprendizaje de la normalización de BD.

## 2.4.5 Correctores de consultas de álgebra relacional

El álgebra relacional es una parte integral del modelo relacional y está formada por un conjunto de operaciones para manipular relaciones y especificar consultas. La mayoría de herramientas que permiten trabajar en álgebra relacional no han sido concebidas como entornos de aprendizaje y autoevaluación sino que están pensadas como aplicaciones que permiten ejecutar expresiones de álgebra relacional sobre una base de datos y visualizar el resultado obtenido. Estas herramientas pueden ser utilizadas para el aprendizaje del álgebra relacional, aunque carecen de todas las funcionalidades de una plataforma de e-learning (gestión de usuarios, actividades, evaluación, etc.). Entre estas herramientas podemos citar:

- Dietrich y otros [DEP97] han propuesto WinRDBI [WinRDBI], una herramienta educativa, desarrollada por Arizona State University, para aprender los lenguajes relacionales formales. Para ello dispone de una interfaz amigable en entorno Windows que permite la creación de bases de datos relacionales, la inserción de registros en las distintas tablas y la formulación de consultas en álgebra relacional, SQL y cálculo relacional de tuplas y de dominio. Para utilizar WinRDBI se debe descargar de su página original e instalarla. El objetivo de esta herramienta es que los alumnos exploren y trabajen los lenguajes de consulta relacionales y obtengan un feedback inmediato, viendo las soluciones a las consultas propuestas. Es el propio alumno quien entra los registros y luego entra una consulta en cualquier de los lenguajes. Finalmente el alumno visualiza el resultado de la ejecución de la consulta en el lenguaje seleccionado. No es un corrector automático sino una herramienta educativa para entrar consultas y ver el resultado de su ejecución. Como inconveniente, la notación que debe usarse en las consultas de álgebra relacional difiere bastante de la notación matemática estándar utilizada normalmente.

- LEAP [Leap] es un SGBD relacional y está diseñado como una herramienta educativa para ayudar a estudiantes y profesores. Es una aplicación para trabajar con álgebra relacional y no es compatible con SQL. Permite interrogar a una base de datos mediante una serie de comandos de álgebra relacional, no soportando la notación matemática habitual. LEAP es distribuido bajo los términos de las licencias GPL. La distribución contiene todo el código fuente en C, simplemente hace falta compilarlo y se ejecuta sin problemas, tanto en Linux como en Windows. Está pensada como una herramienta para que el alumno se familiarice con el álgebra relacional, para que se ejercite y vaya visualizando el resultado de la aplicación de las diferentes operaciones. No corrige automáticamente, sino que ejecuta las operaciones indicadas por el alumno.
- RelationalQuery [RQuery] es otra herramienta que permite el trabajo con álgebra relacional. El funcionamiento de esta herramienta consiste en escribir la expresión en álgebra relacional y el sistema la traduce a una sentencia SQL equivalente y a continuación la ejecuta para que el alumno pueda visualizar el resultado. El alumno visualiza la sentencia SQL equivalente. Dispone de dos tipos de interfase: una de líneas de comando y otra una sencilla interficie gráfica. No permite expresiones en la notación matemática habitual.
- Relational [Rela] es otra herramienta educativa desarrollada con el objetivo de ejecutar una expresión de álgebra relacional y mostrar el resultado. Permite al alumno definir sus propias tablas, entrar los valores de los distintos registros y comprobar si el resultado obtenido en una consulta es el esperado según los valores que el mismo ha entrado. A diferencia de LEAP permite utilizar la notación matemática habitual. Para utilizar este software es necesario descargarlo, instalarlo y definir las relaciones con las que se va a trabajar.
- Hernández y otros [HCP+02] desarrollan una herramienta formada por un editor y un intérprete que traduce las expresiones de álgebra relacional a SQL y las ejecuta sobre Microsoft Jet 3.0. Sólo está disponible en Windows y se necesita tener instalado Microsoft Access en la máquina donde se ejecuta. Permite la notación matemática habitual y como las demás herramientas sirve para que el alumno realice la consulta y compruebe si el resultado es el esperado.
- Davis y otros proponen Virtura [DF03]. Se trata de un tutor virtual para álgebra relacional diseñado específicamente para el aprendizaje a distancia. Las limitaciones más importantes de esta herramienta es que se ha diseñado para una sola base de datos, no es una herramienta web, ni tampoco permite escribir expresiones de álgebra relacional en la notación matemática habitual. Dispone de un intérprete que construye la sentencia SQL equivalente y la ejecuta para poder mostrar los resultados al alumno.
- Apple y otros proponen iDFQL [AQT+04] un entorno interactivo para ayudar a los estudiantes a resolver consultas de álgebra relacional a través de unos diagramas de flujo de datos. Para la construcción de estos diagramas el alumno va seleccionando las distintas relaciones y las operaciones a efectuar entre ellas. El resultado final es un esquema en forma de árbol, en donde las relaciones son nodos terminales y las operaciones son los nodos intermedios. Una vez efectuadas todas las operaciones en la raíz del árbol se obtiene el resultado. Cuando se ha dibujado el esquema y todos los nodos están conectados, se puede ejecutar cada operación paso a paso para ir viendo el resultado obtenido o bien ejecutar todo el árbol. En este caso el alumno visualiza el resultado final y también la sentencia SQL equivalente. No es una herramienta web, no corrige la solución del alumno, simplemente se limita a visualizar el resultado de la ejecución parcial o total del diagrama.
- Beynon y otros proponen EDDI [BBR+03], un entorno que permite a los alumnos explorar las relaciones entre SQL y álgebra relacional. Permite escribir expresiones de álgebra relacional y de SQL. Las expresiones de álgebra relacional se tienen que escribir en un determinado formato (no admite la notación matemática). No formula consultas, sino que simplemente muestra el resultado a las consultas efectuadas. No es una herramienta web y simplemente sirve para ejercitarse en la resolución de este tipo de consultas.
- McMaster y otros [MAB06] proponen un sistema basado en una librería de funciones, con una función para cada una de las operaciones del álgebra relacional. Han implementado esta librería de funciones en Visual FoxPro bajo el nombre de RALGPROC. Usando esta aproximación, cada

consulta de álgebra relacional es escrita como una secuencia de llamadas a funciones de la librería. Cada función dispone de sus parámetros que pueden ser tanto relaciones, operaciones del álgebra relacional, expresiones, etc. Evidentemente no permite expresiones escritas en notación matemática, no es un entorno web y tampoco corrige de forma automática.

Una evaluación más detallada de alguna de estas herramientas se puede obtener de [GEV+05]. La Tabla 2.7 ofrece una comparativa resumida de los principales aspectos que estamos evaluando de las distintas plataformas. Tal como se puede observar en la tabla y según nuestro conocimiento no hay plataformas de e-learning que permitan la corrección automática de ejercicios de álgebra relacional y que faciliten el feedback necesario para ayudar al alumno en caso de que cometa errores. Ninguna de las herramientas examinadas permite la corrección de forma automática. La mayoría no están desarrolladas en un entorno web y es necesario descargar e instalar el software para su utilización. Todas ellas están pensadas para que el alumno escriba una expresión en álgebra relacional (en diferentes formatos, según la herramienta) y el sistema interprete, ejecute y visualice los resultados de las distintas operaciones. Son pocas las que permiten la notación matemática habitual.

ALGEBRA RELACIONAL	W i n R D B I	L E A P	R e l i e n t o	R e l a t i o n a l	H C P + 0 2	V I R T U R A	i D F Q L	E D D I	R A L G P R O C
CARACTERÍSTICAS / ENTORNOS									
Entorno integrado para la docencia de bases de datos	si parcial	no	no	no	no	no	no	no	no
Web-based tool	no	no	no	no	no	no	no	no	no
Editor propio	si	si	si	si	si	si	si	si	si
Corrección automática	no	no	no	no	no	no	no	no	no
Puntuación automática del ejercicio	no	no	no	no	no	no	no	no	no
Evaluación automática del alumno	no	no	no	no	no	no	no	no	no
Trabajo colaborativo	no	no	no	no	no	no	no	no	no
Necesidad de software adicional	no	no	no	no	si	no	no	no	si
Proporciona feed-back automático	no	no	no	no	no	no	no	no	no
Notación matemática	no	no	no	si	si	no	no	no	no

Tabla 2.7 Comparativa de herramientas de aprendizaje de álgebra relacional.

## 2.4.6 Correctores de sentencias SQL

El lenguaje de consulta estándar SQL (Structured Query Language) tiene una amplia aceptación y es el utilizado en el trabajo diario con una base de datos relacional. Dada la importancia del lenguaje SQL se han desarrollado distintas herramientas que ofrecen soporte a su aprendizaje. En muchos casos su objetivo es simplemente que el alumno practique y se ejercite en la resolución de consultas. Tal como veremos, muchos de ellos se presentan como una herramienta independiente, no integrada a ninguna plataforma de e-learning. A menudo se tiene que descargar la herramienta e instalarla para poder utilizarla. Los trabajos más relevantes en herramientas para el aprendizaje de SQL son los que se citan a continuación:

- Dietrich y otros [DEP97] han propuesto WinRDBI [WinRDBI], tal como hemos comentado en la sección 2.4.5. El alumno entra una consulta en SQL y el sistema visualiza el resultado de la ejecución de la consulta. No es un corrector automático sino una herramienta educativa para entrar consultas y ver el resultado de su ejecución.
- Kearns y otros [KSF97] presentan eSQL, una herramienta tipo SGBD en la que el alumno dispone de un subconjunto de sentencias SQL con las que puede trabajar. No pretende ser un SGBD más, sino una herramienta meramente educacional. No es una herramienta web y no corrige las soluciones enviadas, simplemente permite al alumno ver las tablas de la consulta con sus valores, escribir una sentencia SQL y visualizar el resultado obtenido. Dispone de un modo de ayuda en el que va ejecutando, paso a paso, la consulta entrada. Los valores de las tablas se

limitan a unos pocos registros para que puedan visualizarse por pantalla. No es una herramienta que pueda usarse para la evaluación de los alumnos.

- Mitrovic en [Mit98a] [Mit98b] [Mit03] y otros artículos similares de su equipo, proponen SQLTutor, un sistema de tutorización inteligente en la misma línea que los trabajos [SM02][MMM06][Mit02]. El objetivo de estos trabajos es implementar sistemas que permitan la tutorización del alumno de forma virtual. SQLTutor está diseñado como un entorno de prácticas en donde se supone que los alumnos ya conocen la materia y se ofrece como un complemento a las clases presenciales. Se ha adaptado la versión inicial para que sea una herramienta web. Sólo permite trabajar con sentencias SELECT. El sistema dispone de un conjunto de problemas para distintas bases de datos y la solución ideal para cada caso. La corrección se realiza a través de la comparación con una solución ideal. No ejecuta la sentencia, con lo cual el alumno no puede visualizar ningún resultado, sólo recibe el feedback adecuado a la solución enviada. SQL-tutor se basa en Constraint-Based Modeling (CBM) en donde el conocimiento del dominio se expresa a través de una serie de restricciones. La corrección de la solución de los alumnos se basa en verificar si cumple las reglas establecidas y en caso contrario se da el feedback necesario al alumno a través de los mensajes de error. Por ejemplo una regla específica es que después de una SELECT debe aparecer un \* o un campo. Además de este tipo de reglas SQLTutor también dispone de reglas que comparan la solución del alumno con la ideal entrada por el profesor y en función de las diferencias se mandan los mensajes de error oportunos. Uno de los inconvenientes de esta herramienta es que el alumno no visualiza el resultado de la ejecución de las sentencias SELECT. Esta herramienta se ofrece junto con NORMIT (herramienta de tutorización de ejercicios de normalización) en [www.aw-bc.com/databaseplace](http://www.aw-bc.com/databaseplace)
- Allen en [All00] propone WebSQL, un entorno interactivo desarrollado en la Universidad de Minnesota para la ejecución de sentencias SQL vía Internet. Al igual que en los sistemas anteriores, el sistema simplemente se limita a la ejecución de la consulta y visualización del resultado, y es el alumno quien debe verificar si el resultado es correcto. WebSQL acepta una instrucción SQL a través de un navegador cualquiera, la envía al servidor y éste la manda a un SGBD que la ejecuta. Con el resultado devuelto por el SGBD se construye un fichero HTML que es enviado al navegador para su visualización. Los estudiantes pueden seleccionar entre tres bases de datos y a continuación pueden escoger una consulta. Además WebSQL ofrece un tutorial para las posibles dudas que puedan surgir al alumno. No corrige de forma automática. Disponible en [www.websql.org](http://www.websql.org).
- Laine en [Lai01] presenta SQL-trainer una herramienta web para el entrenamiento con comandos SQL. Dispone de una sola base de datos a partir de la cual se realizan las distintas consultas. El sistema realiza un doble análisis, pre y post ejecución. De entrada valida que los comandos SQL no modifiquen la base de datos y a continuación envían la sentencia a ORACLE para que la analice y si no hay errores la ejecute. Sólo soporta la corrección de sentencias SELECT. Disponible en [http://www.cs.helsinki.fi/u/ronkaine/tikape/Sp-2008/trainer/trainerstart\\_sp08.html](http://www.cs.helsinki.fi/u/ronkaine/tikape/Sp-2008/trainer/trainerstart_sp08.html)
- Bhagat y otros en [BBK+02] desarrollan Acharya, un sistema de tutorización inteligente (ITS) para el aprendizaje de SQL. Soporta cualquier comando SQL y en cada momento va determinando el siguiente problema a realizar en función de las consultas resueltas correctamente. Para cada problema se guardan los enunciados y las soluciones. La interfase de entrada es específica para cada cláusula (select, from, where, group by, having y order by). El proceso de corrección se realiza analizando lo que el alumno ha enviado en cada una de estas cláusulas, siguiendo tres procesos. En un preproceso inicial se eliminan palabras clave como between, in, not in, exist y except y se expresan en función de or y and. Un segundo proceso atomiza las condiciones de la cláusula where que serán evaluadas en un tercer proceso como si se trataran de tablas de verdad para identificar las expresiones equivalentes a las de la solución correcta.
- Coleman en [Col03] propone AsseSQL un sistema para evaluar (tipo examen) a los alumnos en la materia de SQL. Propone un sistema en que el alumno debe contestar un conjunto de sentencias en un determinado tiempo. Cuando escoge una consulta le aparece el resultado que deberá obtener. El alumno escribe su sentencia SELECT y la manda a ejecutar. Si tiene errores sintácticos, se comunican al alumno. En caso de poder ejecutarse, se compara el resultado con el

esperado y se facilitan los mensajes correspondientes. Sólo permite ejecutar sentencias SELECT y su función es la de auto-evaluación o la realización de exámenes para los estudiantes.

- Sadiq y otros [SOS+04] proponen SQLator una herramienta web interactiva para el aprendizaje de SQL. La principal aportación de esta herramienta es que dispone de una función para la corrección automática de SELECTS basada en complejos algoritmos heurísticos. Actualmente disponen de dos bases de datos con más de 300 consultas divididas en tres grupos, según su dificultad. No aporta feedback alguno, solamente se limita a indicar si son correctas o incorrectas las soluciones enviadas por el alumno. Sólo corrige sentencias SELECT en las que el alumno visualiza el resultado de la corrección. Disponible en <http://flexel.itee.uq.edu.au/sqlator/>
- Kenny y otros en [KP05] proponen un sistema automatizado de tutorización para fomentar las habilidades en el trabajo con bases de datos. Tipifica las sentencias SELECT en varios niveles según el grado de dificultad. Su sistema facilita feedback de tres formas diferentes: mensajes de error, pistas o soluciones parciales a elección del alumno. Ofrece una valoración del trabajo realizado por el alumno y según esta valoración muestra mensajes detallando el grado de dificultad para cada nivel de consultas y aconseja y selecciona un grupo de consultas que sería interesante que intentara resolver. Sólo trabaja con sentencias SELECT. La corrección se realiza mediante la comparación de la solución del alumno con la almacenada a través de una serie de pasos.
- Dekeyser y otros en [DRL06] [DRL07] [RDL07] proponen SQLify un sistema de corrección mixto (semiautomático). El sistema evalúa la solución enviada por el alumno y le asigna un nivel de varios posibles. El sistema determina este nivel en función de cómo ha podido evaluar la solución en comparación de la sentencia correcta facilitada por el profesor. El nivel más bajo indica que la solución es sintácticamente incorrecta y el más alto indica que la solución es correcta. En medio se establecen varios niveles que indican que el sistema no ha sido capaz de determinar si la solución es con toda seguridad correcta o incorrecta. En este caso se establece un sistema de revisión por pares entre los propios alumnos y con la moderación de un instructor. A partir de la nota facilitada por el sistema más la de la revisión se obtiene la nota final.
- Abelló y otros en [ARU+08] [ABC+09] y [BQA+08] proponen LEARN-SQL, un sistema basado en la web y conforme a la especificación IMS QTI que permite el aprendizaje y evaluación de SQL. Diseñado por la Universidad Politécnica de Catalunya y la UOC, permite la corrección automática de cualquier comando SQL. El sistema está integrado a la plataforma Moodle. Se definen varios juegos de pruebas para cada pregunta, con lo cual el tiempo de preparación de un ejercicio es considerable. La corrección de una solución enviada se realiza mediante la ejecución de varios tests de pruebas y según el número de pruebas que haya superado se asigna una puntuación a cada ejercicio. Los alumnos no visualizan el resultado de las sentencias SELECT. Los autores afirman que el uso de esta herramienta fuera del aula no ha tenido mucho éxito y es usada principalmente para la realización de ejercicios en el aula de prácticas.

Como comentarios generales podemos decir que la mayoría sólo soportan la sentencia SELECT y que cada una de ellas se ha desarrollado para un entorno y contexto determinado, con sus particularidades específicas. De todos estos trabajos, los más conocidos son el WinRDBI y el SQLTutor. WinRDBI se ha utilizado en muchas universidades como herramienta de entrenamiento. El alumno construye su base de datos, entra sus datos y a partir de aquí puede practicar en la resolución de consultas. No es un corrector y hay que descargar e instalar el software. Por su parte SQLTutor y LEARN-SQL son las más interesantes. La Tabla 2.8 ofrece una comparativa resumida de los principales aspectos que estamos evaluando de las distintas plataformas.

SQL	WinRDBI	esQL	SQL-Tutor	WebSQL	SQL-trainer	Accharya	Assesqil	SQLator	KP05	SQLify	LEARN-SQL
<b>CARACTERÍSTICAS / ENTORNOS</b>											
Entorno integrado para la docencia de bases de datos	si parcial	no	si parcial	no	no	no	no	no	no	no	no
Web-based tool	no	no	si	si	si	si	si	si	si	si	si
Editor propio	si	si	si	si	si	si	si	si	si	si	si
Corrección automática	no	no	si	no	si	si	si	si	si	semi	si
Puntuación automática del ejercicio	no	no	no	no	no	no	no	no	no	semi	no
Evaluación automática del alumno	no	no	no	no	si	no	si	si	no	si	si
Trabajo colaborativo	no	no	si	no	no	no	no	no	no	no	no
Necesidad de software adicional	no	no	no	si	si	no	no	no	no	no	si
Proporciona feed-back automático	no	no	si	no	si	si	si	no	si	si	si
Cualquier comando Sql	no	no	no	no	si	si	no	si	no	no	si
Visualización resultado SELECT's	si	si	no	si	no	no	si	si	si	si	no
Visualización del esquema de la base de datos	si	no	si	no	si	si	no	si	no	si	si

Tabla 2.8 Comparativa de herramientas de aprendizaje de SQL.

Por ser SQL un lenguaje estándar utilizado ampliamente en el sector de desarrollo de software, se han desarrollado muchos tutoriales y herramientas para su aprendizaje y práctica. La mayor parte de ellos están orientados a un determinado SGBD (Oracle, SQLServer, MySQL, etc. ) aunque también hay entornos web que ofrecen buenos cursos online. Algunos de estos productos son software libre y otros se ofrecen como productos comerciales. Entre estos productos podemos citar:

- SQLCourse. Es un entorno interactivo online que además de ofrecer un tutorial permite practicar en línea utilizando un interprete de SQL. Se reciben los resultados de los comandos SQL de forma inmediata. El entorno no está pensado para la corrección automática ni tampoco facilita ningún tipo de feedback, solamente visualiza el resultado de la ejecución de la sentencia. Disponible en [www.sqlcourse.com](http://www.sqlcourse.com)
- SQL Training. Además de tutoriales, ofrece ejemplos prácticos, ilustraciones y detalladas explicaciones. Se asignan ejercicios a los alumnos para que puedan practicar y ver sus progresos. Disponible en <http://www.sqltraining.org>

## 2.4.7 Integración de entornos de aprendizaje en materia de BD.

En el estado del arte en entornos para el aprendizaje de BD que hemos desarrollado en las secciones anteriores hemos visto que la mayoría de ellos se presentan como herramientas independientes. Consideramos que para un buen aprendizaje/evaluación es mejor disponer de una única herramienta que integre los diferentes entornos, de forma que el alumno disponga de una única plataforma. En este aspecto de los entornos evaluados son muy pocos los que presentan un entorno integrado.

En esta línea de integración, uno de los entornos más conocidos, y que ya se ha comentado previamente, es WinRDBI. Dietrich y otros [DEP97] desarrollaron WinRDBI [WinRDBI] que es una herramienta educativa para aprender los lenguajes relacionales formales. Para ello dispone de una interfaz amigable en entorno Windows que permite la creación de bases de datos relacionales, la inserción de registros en las distintas tablas y la formulación de consultas en álgebra relacional, SQL y cálculo relacional de tuplas y de dominio. Para utilizar WinRDBI se debe descargar el programa de <http://www.eas.asu.edu/~winrdbi/> e instalarlo. El objetivo de esta herramienta es que los alumnos exploren y trabajen los lenguajes de consulta relacionales y obtengan un inmediato feedback, viendo las soluciones a las consultas propuestas. Es el propio alumno quien entra los registros y luego entra una consulta en cualquiera de los lenguajes. Finalmente el alumno visualiza el resultado de la ejecución de la consulta en el lenguaje seleccionado. No

es un corrector automático ni tampoco puntúa los ejercicios, simplemente es una herramienta educativa para entrar consultas y ver su resultado. No permite ni la realización de ERD, ni esquemas de BD, ni tampoco la normalización.

Por otro lado el equipo liderado por A. Mitrovic ha desarrollado diferentes entornos en materia de bases de datos:

- Kermit: para el aprendizaje modelo ER [SM02] [SM04]
- Collect-UML: para el aprendizaje de diagramas de clase [BM05] [BMI06]
- ERM-Tutor: para la obtención de esquemas de bases de datos relacionales [MM04][MMM06]
- Normit: para el aprendizaje de la normalización [Mit02]
- SQLTutor: para el trabajo con sentencias SQL [Mit98a] [Mit98b] [Mit03]

Aunque son entornos diferentes, el conjunto de todos ellos permite el trabajo con los temas más importantes en BD (exceptuando álgebra relacional). Ahora bien estos entornos están pensados para el aprendizaje de estas materias y una limitación importante es que ninguno de ellos ofrece una puntuación automática a los ejercicios. Kermit, SQLTutor y Normit [MSM+04][Mit06] están disponibles en <http://www.aw-bc.com/databaseplacdemo/>

## 2.4.8 Resumen

Se han desarrollado muchos entornos que ayudan en el aprendizaje de los temas principales de un curso de bases de datos. Como se ha visto en las tablas comparativas de las secciones precedentes, la mayoría de estos sistemas consisten en un software que sólo permiten el trabajo en un área concreta y no forman parte de un sistema integral para la docencia en bases de datos. Generalmente no están integrados en ninguna plataforma de e-learning con lo que carecen de las funcionalidades básicas de gestión de cursos, actividades, alumnos, grupos, problemas, etc. ni tampoco de las funcionalidades para el seguimiento del trabajo del alumno que permite llevar a cabo una evaluación formativa. Aunque algunos de ellos realizan tareas de corrección automática y ofrecen el feedback más adecuado en cada momento, son muy pocos los que obtienen una puntuación automática a las respuestas de los alumnos.

A pesar de que los más recientes ya suelen ser herramientas web, todavía hay muchos entornos que necesitan ser descargados e instalados en un ordenador para su utilización o bien necesitan de un software adicional para su funcionamiento.

## **2.5 LA PLATAFORMA ACME**

Uno de los objetivos de esta tesis es evaluar si el uso de correctores desarrollados en materia de BD aporta beneficios en su docencia. Para ello necesitamos de una plataforma de e-learning en la que integrar nuestras propuestas. La plataforma sobre la que vamos a trabajar es ACME (Avaluació Continuada i Millora de l'Ensenyament) que surgió de la iniciativa de varios profesores del Departamento de Informática y Matemática Aplicada de la Universidad de Girona con el ánimo de ofrecer cuadernos de problemas personalizados a los alumnos. La idea inicial era simple: la plataforma proporcionaría problemas a los alumnos, estos los resolverían y enviarían las soluciones a corregir. ACME corregiría las soluciones de forma automática y enviaría información acerca del resultado de la corrección. A pesar de la sencillez de la idea, ACME se centró en la corrección de problemas de respuesta libre de complejidad avanzada, para poder evaluar los niveles cognitivos superiores de la taxonomía de Bloom que se necesitan en el ámbito universitario. Esta corrección automatizada con el correspondiente feedback y con toda la información almacenada del trabajo del alumno, nos permite llevar a cabo una buena evaluación formativa. Por otro lado, si los módulos correctores son capaces de puntuar las actividades también se podría llevar a cabo una evaluación sumativa. Con toda esta información ACME actúa como un sistema CBA completo obteniendo la evaluación automatizada de cada alumno.

En comparación con otras plataformas de e-learning tipo Moodle, Sakai, etc., que sólo permiten la corrección de ejercicios de respuesta fija (elección múltiple, tipo test, relleno de espacios en blanco, etc.), la clave de la plataforma ACME radica en la corrección automática de diferentes tipos de problemas, tanto simples como por ejemplo los de respuesta fija, como complejos de respuesta libre (matemáticos, programación, etc.)

La primera versión de ACME data del año 1998 y estaba centrada en la resolución de problemas matemáticos. El objetivo inicial era que los alumnos se ejercitasen en la resolución de este tipo de ejercicios. Posteriormente la plataforma ha ido evolucionando y se ha ampliado considerablemente, permitiendo la corrección de diferentes tipos de problemas (ver sección 2.5.4) y siendo muy útil para el desarrollo de gran variedad de actividades formativas.

La plataforma está pensada para ser un soporte a las clases presenciales (blended learning), aunque dispone de las funcionalidades de un Campus Virtual (forum, chat, agendas, gestión de cursos, alumnos, asignaturas, etc.) y puede ser utilizada para muy distintas actividades. ACME se ha especializado en la corrección automática de problemas, aunque también permite poner a disposición del alumno todo tipo de contenidos.

Hemos estructurado esta sección en cinco apartados. En el primero se presentan las características generales de la plataforma. En el segundo se describe la tipología de usuarios que soporta ACME. En el tercero se presenta la estructura y funcionamiento de la plataforma ACME. En el siguiente apartado se detalla la tipología de problemas y actividades soportadas. Finalmente se comentan las aportaciones que en innovación docente presenta ACME.

### **2.5.1 Características generales de la plataforma ACME**

A partir de la primera versión, la plataforma ha ido evolucionando y se han incorporado mejoras en cada una de las siguientes versiones. También se ha ido incrementando la variedad de actividades a desarrollar así como la tipología de problemas soportados. La versión actual de ACME se caracteriza por:

- Disponer de un repositorio único de problemas base/actividades de tipología muy distinta, por ejemplo problemas de matemáticas, de física, de programación informática, etc. Los profesores autorizados pueden introducir nuevos ejercicios, así como compartir los ya existentes. Este material está catalogado por temáticas, categorías, asignaturas, nivel de dificultad, etc.
- En según qué tipo de ejercicios, cada problema base del repositorio puede estar formado por varios enunciados y parámetros variables, de forma que con la combinación de ellos se pueden generar muchas versiones distintas del mismo problema. Además cada problema/actividad base

lleva asociado las reglas/sistema para su corrección automática. Por ejemplo, un problema base de matemáticas lleva asociado el código del lenguaje “Mathematica” correspondiente para su corrección y un programa informático lleva asociados varios test de pruebas con las entradas y las salidas esperadas para cada entrada. De esta forma cada problema base se definirá de la forma adecuada para que el corrector de esta tipología de problema sepa llevar a cabo la corrección.

- Para cada asignatura en la que se utilice la plataforma ACME, el sistema proporciona a cada alumno un cuaderno de problemas personalizado con enunciados diferentes para evitar posibles copias. El profesor responsable de cada asignatura organiza el cuaderno por temas/bloques. Cada tema puede contener muchos problemas/actividades. En cualquier momento del curso el profesor puede añadir más problemas.
- El alumno visualiza sus problemas/actividades a través de ACME, los resuelve y envía la solución al sistema que los corrige automáticamente y de forma inmediata. En caso de que la solución sea errónea el sistema facilita el feedback oportuno para ayudar al alumno a detectar sus errores.
- La plataforma ACME guarda todas las soluciones enviadas por los alumnos y la información acerca de su corrección. También ofrece diferentes funcionalidades para que el profesor pueda consultar y seguir el trabajo realizado por los alumnos. Todo ello permite realizar tanto una evaluación formativa como sumativa.
- Para el uso de la plataforma sólo se requiere de una conexión a Internet y un navegador. Todas las funcionalidades de la plataforma se realizan desde cualquier navegador y la plataforma es multilingüe.

## 2.5.2 Tipología de usuarios

La plataforma dispone de los mecanismos de autenticación necesarios para permitir la entrada exclusivamente a los usuarios autorizados. Estos usuarios están clasificados en cuatro perfiles distintos:

- **Administrador.** Tiene acceso a todas las funcionalidades del sistema y es el encargado de gestionar (altas, bajas, modificaciones) los cursos, las asignaturas, los profesores responsables y tutores, la creación de usuarios y sus permisos, asignación masiva de alumnos a una asignatura, etc. En general puede realizar todas las gestiones típicas de un administrador de una plataforma de e-learning.
- **Profesor responsable.** Será la persona encargada de gestionar una asignatura y de definir los temas/bloques de que estará formada. Deberá verificar los problemas creados y guardarlos en el repositorio común. Seleccionará los diferentes ejercicios a asignar a cada tema, pudiendo seleccionar muchos y especificar al sistema que asigne un número determinado de ellos a cada alumno. En cualquier momento puede ir añadiendo más ejercicios y actividades a cualquier tema. Como profesor responsable puede realizar todas las funciones de un profesor tutor.
- **Profesor tutor.** Profesor que dentro del contexto de sus asignaturas tendrá asignado un grupo de alumnos, atenderá sus posibles dudas y realizará el seguimiento y control de cada uno de ellos. Puede visualizar todas las respuestas enviadas y puede facilitar indicaciones si lo cree oportuno. Obtiene todo tipo de estadísticas, resultados y una evaluación de cada alumno.
- **Alumno.** Al entrar en la plataforma visualizará las distintas asignaturas que utilizan ACME. Seleccionará la que desea trabajar y entrará a los distintos temas/bloques de la asignatura. Al seleccionar un tema, verá los distintos problemas/actividades asignados y procederá a su resolución. Una vez resuelto un ejercicio enviará la solución a través de la interfaz correspondiente en función de la tipología del problema. La plataforma ACME realiza de forma instantánea y automática la corrección de la solución enviada. Si es incorrecta le facilita el feedback oportuno que le sirve de guía para corregir los errores. Puede seguir enviando soluciones hasta llegar a la correcta. Cuando utiliza la plataforma para la realización de exámenes/pruebas de validación, visualiza la nota obtenida.

### 2.5.3 Estructura y funcionamiento de la plataforma ACME

La plataforma ACME ha sido desarrollada siguiendo una estructura modular, y la conexión funcional de estos módulos para las tareas del profesor y del alumno es la que se muestra en la Figura 2.10, siendo sus principales módulos los que se describen a continuación:

1. **Repositorio.** El sistema mantiene en un repositorio único los problemas base con el método, las reglas y si es necesario los datos para su corrección. Opcionalmente puede llevar asociadas las pautas de ayuda para guiar al alumno en el proceso de resolución. Todos los contenidos de la plataforma se guardan en este repositorio único.
2. **Módulo de Introducción de Problemas.** Permite al profesor introducir los problemas en el repositorio. Los problemas son creados con un editor cualquiera siguiendo las especificaciones establecidas en función de la tipología del mismo. Cada tipología de problema tiene su estructura determinada. Una vez creado el problema, la plataforma dispone de las funcionalidades para su validación, catalogación y entrada en el repositorio.

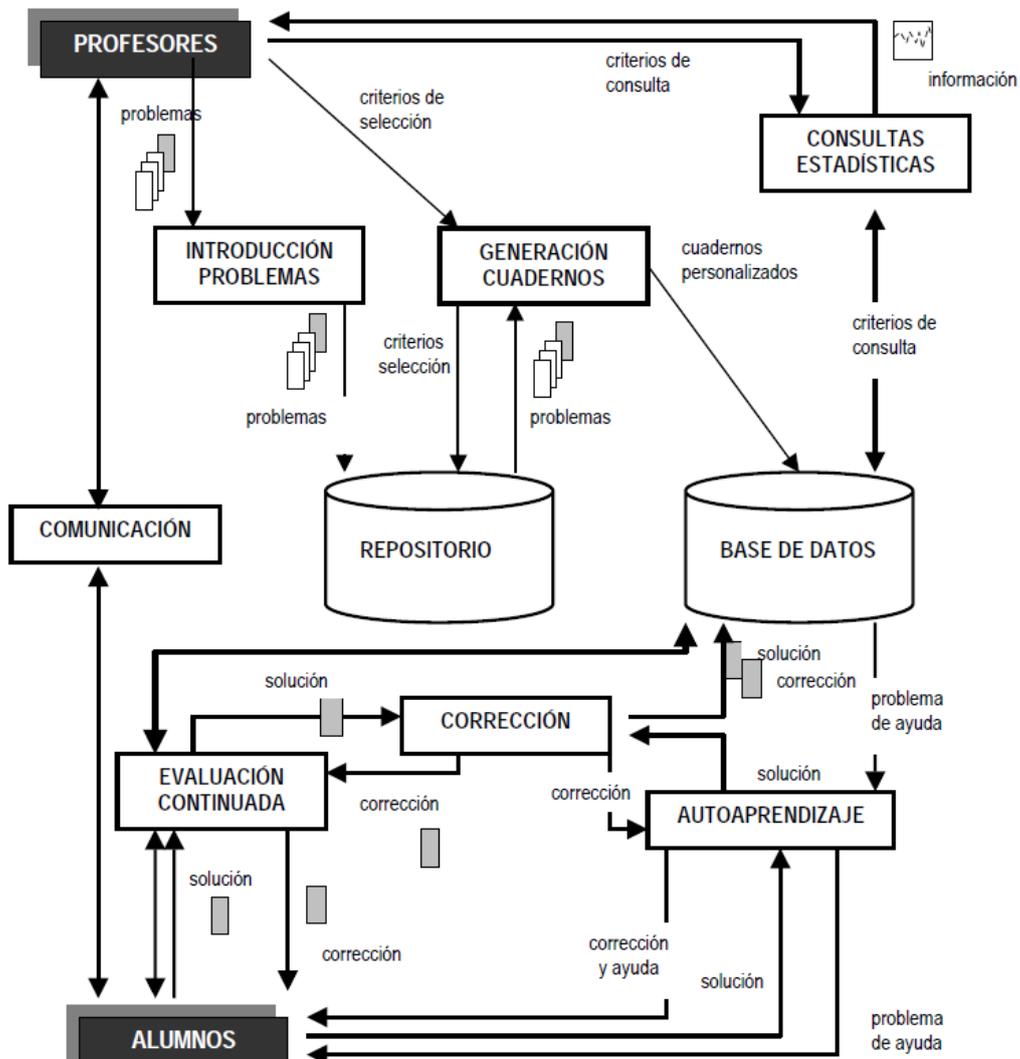


Figura 2.10 Estructura modular de la plataforma ACME.

3. **Módulo Generador de Cuadernos de Problemas.** El profesor a partir de las funcionalidades que dispone la plataforma selecciona un conjunto de problemas base a incorporar en un tema de su

asignatura. Esta selección se hace en base a los criterios de catalogación asignados en la entrada del problema (temática, categoría, nivel de dificultad,...) fijados por el profesor. También determina cuantos de estos problemas deberá asignar a cada alumno. A partir de estos problemas seleccionados, el módulo generador crea las distintas versiones (combinando enunciados, parámetros y valores) y los va asignando automáticamente a los alumnos de la asignatura. El objetivo final es que cada alumno disponga de su cuaderno de problemas y que éstos sean parecidos, pero distintos para evitar posibles copias.

4. **Módulos de Corrección.** Una vez el alumno ha resuelto su problema envía la solución a la plataforma ACME, que a su vez la envía al módulo corrector correspondiente en función del tipo de problema. La plataforma dispone de un módulo corrector para cada tipo de problema. La corrección es efectuada según las especificaciones del problema. El alumno será informado en tiempo real del resultado de la corrección y en el caso de que la solución sea incorrecta, el módulo facilita el feedback oportuno para ir guiando al alumno en la resolución del problema. El alumno tiene la posibilidad de enviar nuevas soluciones y todas ellas quedan registradas en la base de datos del sistema.
5. **Módulo de Evaluación Continuada.** La base de datos del sistema mantiene toda la información relativa a un alumno (problemas asignados, soluciones enviadas,..). De esta forma el módulo de evaluación continuada permite a partir de las soluciones enviadas por el alumno y siguiendo los criterios fijados por el profesor, calcular de forma automática la calificación final que le corresponde.
6. **Módulo de Auto-aprendizaje.** Facilita al alumno un conjunto de problemas con distintos niveles de ayuda. Dichos niveles se activan de forma automática en función del número de errores que cometa el alumno al intentar resolverlos. En el caso extremo y después de varias soluciones fallidas el sistema le proporciona la solución correcta.
7. **Módulo de Consulta de Estadísticas.** Permite al profesor consultar la información que mantiene la base de datos del sistema. Puede obtenerse información del número de problemas asignados a los alumnos, de las soluciones enviadas a un determinado problema, de los intentos hasta llegar a la solución correcta, de la puntuación obtenida, etc.
8. **Módulo de Comunicación.** Este módulo proporciona canales de comunicación entre el profesor y el alumno permitiendo el envío de e-mails entre profesor y alumnos así como la participación en el foro asociado a la asignatura. También permite al profesor adjuntar notas con comentarios informativos a las soluciones enviadas por los alumnos.

Además, para facilitar la generación de problemas, se ha desarrollado un editor específico para ACME. La principal característica de este editor es que permite seleccionar un tipo de problema y ya muestra la plantilla adecuada con todas las marcas y sólo se debe rellenar con la información específica del problema. También se dispone de un manual donde se detalla la estructura de cada tipo de ejercicio.

## 2.5.4 Tipología de problemas y actividades soportadas por ACME

Tal como se ha comentado anteriormente ACME soporta la corrección automática de una gran variedad de problemas/actividades siendo ésta su aportación más innovadora en el campo del e-learning. Concretamente y sin tener en cuenta los módulos correctores de bases de datos que se presentan en esta tesis, las distintas tipologías de problema que soporta son:

- **Ejercicios de respuesta fija.** Dentro de esta categoría ACME soporta los problemas actividades estándar de cualquier plataforma de e-learning: elección múltiple, tipo test, relleno de blancos, cierto/falso, etc.
- **Ejercicios de respuesta libre.** ACME se ha especializado en este tipo de ejercicios, entre ellos:
  - **Problemas/actividades que requieran un planteamiento matemático.** De esta forma se corrigen gran variedad de problemas como: derivadas, integrales, ecuaciones diferenciales, geometría, física, economía y en general cualquier problema que se pueda

resolver con un planteamiento matemático. Información más detallada en [SPB+02], [PBJ+03] y [PBS+05a].

- **Programas informáticos.** Soporta la corrección de programas informáticos escritos en C, C++, Pascal, Java, ... y en general en cualquier lenguaje de programación. El código fuente del programa informático puede ser enviado en uno o más ficheros. Información más detallada en [BPS+04] y [BSP+04]. También soporta la corrección de algoritmos escritos en pseudocódigo imperativo y también en el pseudocódigo orientado a objetos POODI [PCP06].
- **Otros.** Corrección de formulación química (orgánica e inorgánica), de circuitos eléctricos, autómatas deterministas, etc.

También permite la recepción de trabajos que el profesor corregirá de forma manual, no automática. En este caso la plataforma sólo sirve para recibir/organizar el trabajo que realizan los alumnos. La corrección no se efectúa de forma automática y es el profesor quien controla y corrige el trabajo del alumno. Entre estos problemas/actividades soporta: wiki's, portafolios, glosarios y textos a corregir (ya sean escritos directamente en el entorno o en un fichero adjunto).

En la Figura 2.11 mostramos un esquema de los diferentes problemas que actualmente soporta ACME, en los que no se ha incluido la tipología de problemas que se presenta en este trabajo.

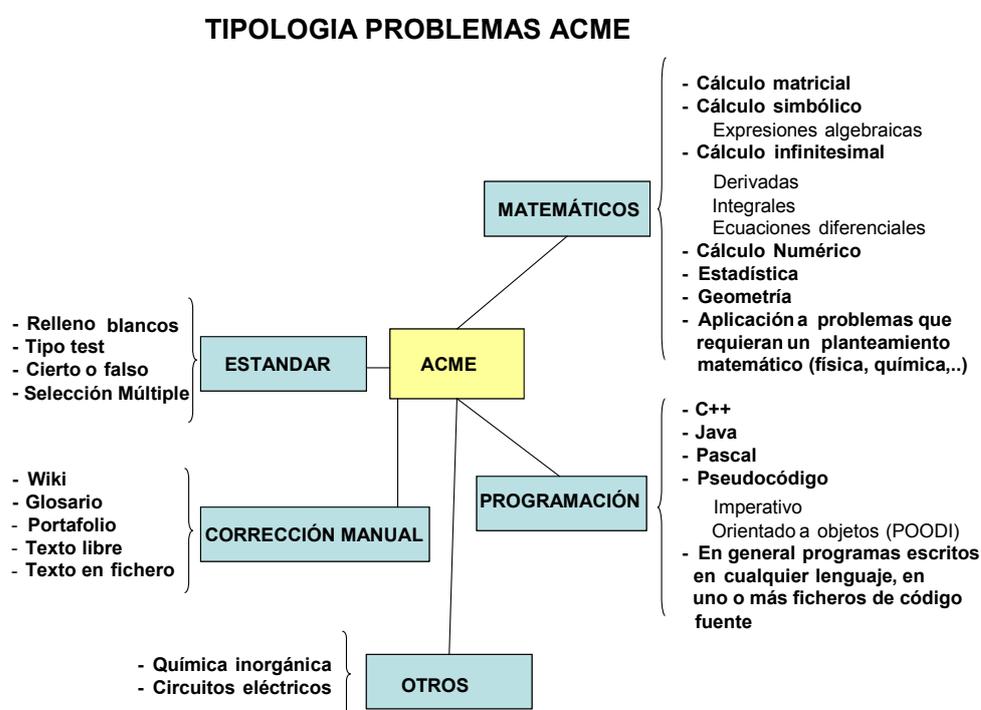


Figura 2.11 Tipología de problemas soportados por la plataforma ACME.

Otra de las características principales de la plataforma ACME es la amplia variedad de actividades formativas, tanto presenciales como no presenciales, que se pueden desarrollar a través de ella [SPB+06a]. Estas actividades son las que nos permiten realizar la evaluación formativa a lo largo de todo el curso. Además, junto con la amplia variedad de problemas soportados, hace que la plataforma se pueda utilizar en cualquier tipo de asignatura. Las distintas actividades que se pueden realizar con la plataforma las clasificamos como presenciales, no presenciales y actividades que se pueden desarrollar de ambas formas:

- **No presenciales.** Estas actividades formativas las podrá realizar el alumno desde cualquier lugar donde pueda establecer una conexión a Internet. Entre estas actividades podemos destacar:
  - **Actividades para la evaluación formativa.** Para que la evaluación de un alumno no dependa exclusivamente de un examen, sino de su trabajo diario y de su evolución

durante el curso, es necesario disponer de un sistema informático, en este caso de ACME, a través del cual tanto profesor como alumno sepan el trabajo realizado. De esta forma el alumno tiene la suficiente información para saber en todo momento su evolución en el curso y sus deficiencias se pueden corregir a tiempo, sin necesidad de esperar el resultado del examen final. En nuestro caso y para ello podemos usar cualquier ejercicio del repositorio. Podemos hacer una evaluación específica de un tipo de problema/actividad o bien de todo un tema a través de los ejercicios asignados. La plataforma ACME permite al profesor ir añadiendo temas y ejercicios en el cuaderno de cada alumno a medida que avanza el curso. Cada actividad tendrá una fecha límite de entrega. El alumno podrá enviar varias soluciones hasta llegar a la correcta. El profesor puede visualizar en todo momento el trabajo realizado por el alumno y llevar un control exhaustivo de su evolución. La plataforma puede calcular notas a partir de las soluciones enviadas por el alumno, ya sea en función del número de intentos llegar a la solución correcta o bien calculada por el propio corrector. El profesor podrá modificar esta nota. En este tipo de actividades marcamos al alumno la importancia de enviar a corregir una solución una vez la haya comprobado y esté lo más seguro posible que la solución sea correcta.

- **Aprendizaje autónomo.** La idea es exactamente la misma del apartado anterior, con las salvedades que el profesor no establece una fecha límite y tampoco se le evalúa automáticamente. Simplemente se trata de ejercicios a partir de los cuales el alumno podrá realizar un aprendizaje autónomo. Este tipo de actividad nos sirve también para asignar *problemas de refuerzo* a un alumno. Si detectamos que un alumno tiene dificultades de resolución de un tipo determinado de problemas de evaluación le podemos asignar más problemas de este tipo que no servirán para su evaluación, simplemente para que practique. En este caso, el profesor de forma manual o la plataforma de forma automática, en función de los errores cometidos, pueden asignar problemas de esta tipología como refuerzo. Estos problemas de refuerzo se corrigen siguiendo la misma filosofía ACME y su objetivo final es el autoaprendizaje.

En este mismo apartado podemos mencionar las *auto-evaluaciones*. En este caso el profesor agrupa diferentes problemas/actividades en un bloque de autoevaluación y el alumno puede observar su progreso en el curso.

Finalmente comentar que ACME soporta y ofrece también “*problemas de ayuda*” o “*problemas guiados*”. La finalidad de este tipo de problemas es guiar al alumno en el desarrollo de la actividad. De esta forma un “problema guiado” dispone de varios niveles de ayuda. El primer nivel de ayuda consiste en la resolución del problema por pasos, donde el alumno debe ir enviando respuestas a las preguntas formuladas. Si las soluciones enviadas van siendo correctas el alumno va avanzando hasta llegar a la solución final. En caso de que el alumno no sepa contestar correctamente a las preguntas formuladas se activa el segundo nivel de ayuda que consiste en facilitar alguna explicación detallada de cómo afrontar la resolución de los distintos pasos. Si a pesar de la información facilitada el alumno no sabe como avanzar se activa el tercer nivel de ayuda en el cual visualiza el desarrollo completo del problema. El último nivel de ayuda, consiste en que el alumno puede descargarse un video que muestra al profesor desarrollando y explicando la resolución del problema.

- **Trabajo en grupo.** Una de las competencias transversales que todo alumno debería adquirir es el trabajo en grupo. La plataforma ACME ofrece la posibilidad de trabajar en grupo. En este caso se asigna un problema/actividad a un grupo de alumnos. Estos dispondrán de un foro específico tutelado por el profesor y al que podrán hacer sus aportaciones. Consensuada una solución final, ésta se envía a corregir y sigue el proceso habitual de ACME.
- **Participación en foros.** Los foros permiten un intercambio de información entre un determinado grupo de alumnos y sus profesores. Suelen ser actividades complementarias a las anteriores.

- **Presenciales.** En estas actividades el alumno usará la plataforma ACME desde un aula que disponga de PC's o portátiles con conexión a Internet. En este caso será una actividad presencial tutelada por un profesor que explica una determinada materia/problema y a continuación el alumno pasa a resolver ejercicios con la plataforma ACME. Si el alumno no resuelve correctamente los ejercicios propuestos, el profesor le irá indicando sus errores y le ayudará a resolver sus dudas. El profesor podrá dedicar más tiempo a los alumnos que tengan más

dificultades, mientras que los alumnos que vayan resolviendo los problemas de forma correcta no necesitarán ayuda del profesor. Al final de la sesión y a través de la plataforma visualizará el trabajo realizado por cada alumno.

- **Presenciales/no presenciales.** Este es el caso de las actividades tipo examen que se pueden realizar de forma presencial desde una aula o bien de forma no presencial, a elección del profesor. Sirven para la evaluación sumativa. En el primer caso, el profesor convoca a los alumnos en un aula informática a una hora determinada (tipo examen). El profesor facilita una clave que necesitará el alumno para poder realizar el examen con la finalidad que nadie desde fuera pueda suplantarle. El alumno va resolviendo el examen y enviando soluciones. Inmediatamente va sabiendo si realiza correctamente cada apartado. Llegada la hora final ya no puede enviar más soluciones y visualizará la nota obtenida. En el segundo caso, el alumno podrá realizar el examen dónde y cuándo quiera, ahora bien, una vez entrada en la sesión de examen dispondrá del tiempo estipulado y no podrá salir de la sesión y completarlo más tarde. La Figura 2.12 muestra el resumen de las actividades que se puedan desarrollar con ACME.

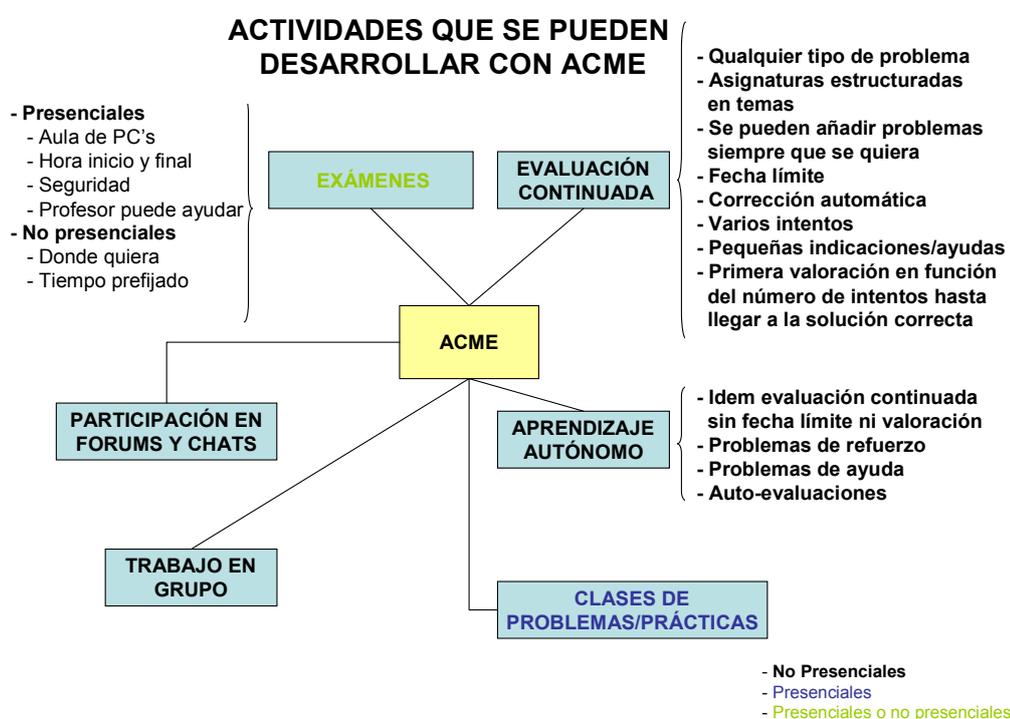


Figura 2.12 Actividades soportadas por ACME.

## 2.5.5 Aportaciones a la innovación docente de la plataforma ACME

Las aportaciones en materia de innovación docente y mejoras que hemos detectado con el uso de la plataforma ACME son:

- La plataforma ofrece al profesor las herramientas necesarias para realizar un seguimiento personalizado del trabajo del alumno. Esta información puede ser usada en el proceso de evaluación formativa. El profesor sabe en todo momento cuáles son los alumnos que están enviando soluciones y puede visualizar su trabajo. También le permite al profesor detectar e ir corrigiendo deficiencias de aprendizaje, sin tener que esperar al examen final.
- Permite compartir material entre los profesores ya que se dispone de un repositorio único de problemas.

- Menor dedicación del profesor en la corrección de ejercicios, por el simple hecho de que el sistema corrige automáticamente las soluciones enviadas. Por el contrario el profesor deberá preparar los problemas a incorporar a la plataforma y dedicar más tiempo a las tutorías presenciales ya que los alumnos que tienen dificultades en la resolución de los ejercicios acuden a ellas. También aumentan considerablemente las consultas planteadas por correo electrónico. Consideramos que este tipo de consultas son siempre positivas.
- Permite la planificación de distintas actividades con diferentes usos: práctica, auto-evaluación, realización de exámenes, etc. aunque el uso mayoritario que se le da es para la evaluación formativa.
- Estimula al alumno en su trabajo diario. El alumno sabe inmediatamente y en cualquier momento si el ejercicio realizado es correcto. Para incentivar el trabajo diario del alumno, proponemos que la resolución correcta de ejercicios ACME se tenga en cuenta en la evaluación sumativa. Evidentemente se corre el riesgo que el alumno nos engañe y que en realidad no sea él quien nos mande las soluciones o que simplemente alguien le resuelva el problema. Para evitar posibles fraudes, aconsejamos que se realicen pruebas de validación en las que se exija una nota mínima no inferior a 4. El alumno es consciente que el trabajo realizado es controlado por el profesor.
- Aumenta el número de tutorías, tanto presenciales como virtuales vía e-mail, motivo por el cual se potencia la relación alumno-profesor.
- Generalmente los alumnos que realizan correctamente las actividades ACME, aprueban la asignatura.

Finalmente comentar que se puede encontrar información más detallada de la plataforma ACME en las publicaciones [SPB+02], [PBJ+03], [PBS+05a], [SPB+06a] y [SPB+06b] en donde se explica su funcionamiento y se justifican los comentarios mencionados.

## 2.5.6 Resumen

El hecho de disponer de la plataforma de e-learning ACME, construida a medida de nuestras necesidades, nos permite desarrollar entornos muy especializados en distintas materias. Su estructura modular nos permite fácilmente incorporar nuevos correctores en función de la tipología de problemas a corregir. Concretamente en este trabajo, y para alcanzar nuestro objetivo, se deberán desarrollar módulos que hacen referencia a temas de bases de datos con los que queremos conseguir un sistema CBA en esta materia. Para cada nueva tipología de problema a incorporar a la plataforma será necesario definir:

- la estructura que tendrá el problema (enunciados, parámetros, valores, pautas y reglas para la corrección, mensajes de error, datos para testear las soluciones, parámetros para la puntuación, etc.)
- el diseño e implementación de las interfaces adecuadas
- el diseño e implementación del módulo corrector

Desde un punto de vista docente la plataforma soporta una gran variedad de actividades que permiten al profesor utilizarla tanto para la evaluación formativa como sumativa del alumno.

En este trabajo que se presenta pretendemos que las aportaciones de ACME se hagan extensibles también a la materia de bases de datos. El problema radica en la complejidad de desarrollar correctores que de forma automática sepan interpretar la semántica asociada a un problema de diseño de bases de datos o a una consulta cualquiera.

## **2.6 VALORACIÓN FINAL**

En este capítulo hemos presentado el estado del arte de los pilares de nuestra investigación: evaluación, CBA y docencia en bases de datos. Dado que nuestra línea de investigación abarca estos tres temas y siendo nuestro objetivo el desarrollar un entorno CBA en materia de bases de datos, también se ha presentado el estado del arte en herramientas para el aprendizaje y evaluación en materia de BD. Finalmente se ha descrito la plataforma ACME sobre la cual se desarrollará nuestro trabajo.

El objetivo principal de los sistemas CBA es realizar la evaluación del alumno de forma totalmente automatizada a través de la evaluación de un conjunto de actividades realizadas. La mayoría de estos sistemas se basan principalmente en ejercicios de respuesta fija, donde el alumno simplemente debe escoger una respuesta entre varias o entrar una palabra o un número como respuesta a las actividades planteadas. Estos sistemas son válidos para evaluar los niveles cognitivos inferiores de la taxonomía de Bloom (conocimiento y comprensión). Ahora bien, para lograr las competencias de una asignatura de Bases de Datos es necesario alcanzar niveles intermedios/superiores de conocimiento. Es necesario que el alumno sepa aplicar, analizar, diseñar sobre una serie de temas concretos. Para evaluar estos niveles cognitivos a través de un sistema CBA son necesarias herramientas mucho más especializadas que permitan al alumno desarrollar sus propias respuestas.

Estos sistemas nos permiten llevar a cabo la evaluación formativa controlando en todo momento el trabajo desarrollado por el alumno. A partir de esta evaluación continuada el profesor puede modificar el desarrollo del curso y subsanar posibles deficiencias. El tiempo que el profesor dedica a la corrección de ejercicios se reduce muy considerablemente. Por otra parte, el alumno conoce en cada momento cual es situación, qué sabe y qué no sabe hacer. El sistema facilita el feedback oportuno al alumno que le va guiando hacia la solución correcta.

Los CBA que permiten el trabajo a través de Internet tienen la ventaja de que el alumno puede realizar sus actividades en cualquier sitio y a cualquier hora. Ahora bien, tienen el inconveniente de que no sabemos si realmente quien envía las soluciones es quien ha desarrollado el ejercicio. Para solventar este problema son necesarias pruebas de validación/exámenes para asegurarnos de que no haya ningún fraude. En este tipo de actividades es necesario que el sistema nos facilite una puntuación, más teniendo en cuenta que en problemas complejos y extensos son aceptables pequeños errores.

Las herramientas existentes en materia de BD son muy limitadas y específicas de un tema. Según nuestro conocimiento no hay ninguna herramienta web que pueda actuar como un sistema CBA que integre las principales actividades que se desarrollan en un curso de Bases de Datos y que además las corrija, ofrezca el feedback correspondiente y calcule la puntuación de forma totalmente automática.

Dadas estas limitaciones, nuestro objetivo en este trabajo es desarrollar un entorno integrado en materia de BD que facilite todas las prestaciones de un CBA, desde la entrega de materiales y tutoriales, hasta la asignación de ejercicios que, una vez resueltos, serán corregidos de forma automática, retornando el feedback adecuado y calculando la puntuación pertinente.



## **CAPÍTULO 3. LÍNEAS GENERALES DE NUESTRA PROPUESTA**

Nuestro objetivo es desarrollar un entorno de CBA de soporte a la docencia en materia de bases de datos. Para alcanzar este objetivo deben analizarse los diferentes temas que se tratan en un curso de BD. Si bien estos temas son diferentes todos ellos tienen puntos en común: son complejos y requieren de ejercicios con respuestas libres para su evaluación. Estas similitudes nos han permitido definir una metodología de trabajo común aplicable a cada uno de los problemas que queremos resolver. En este capítulo queremos presentar este marco de trabajo común y dar una visión global del trabajo. En primer lugar presentamos nuestro marco de trabajo situando el entorno en que se realiza la tesis. A continuación planteamos las líneas generales de nuestro trabajo y finalmente presentamos la metodología de trabajo que hemos seguido.

### **3.1 MARCO DE TRABAJO**

Tal como se ha comentado en el capítulo 1, esta tesis se enmarca dentro del Computer Based Assessment (CBA) en el ámbito de la docencia de bases de datos en un entorno universitario. Las competencias básicas que el alumno debe adquirir en esta materia y relacionadas con este trabajo son:

- Ser capaz de realizar el diseño conceptual y lógico de una base de datos.
- Crear una base de datos relacional y realizar consultas con SQL y álgebra relacional sobre ella.

Estas dos competencias van a ser sin duda las que tendrá que realizar cuando se incorpore al mundo laboral. Para lograrlas es necesario que el alumno se ejercite con la resolución de supuestos análogos a situaciones reales. Estas dos competencias tienen en común un elevado componente práctico, por esta razón, en la metodología tradicional de enseñanza el profesor propone problemas al alumno para que así consolide los conocimientos teóricos. Ahora bien, de todos es conocido el fracaso de este sistema donde el profesor, después de explicar una materia, facilita un listado de problemas a los alumnos, despreocupándose tanto de si los saben hacer como de si realmente los hacen. Por otro lado, y para que el alumno aprenda esta materia, es necesario que se ejercite con la resolución de varios (a ser posibles muchos) supuestos, consultas, ejercicios, etc. con la correspondiente corrección e indicaciones por parte del profesor. En este contexto la evaluación formativa es la más adecuada a pesar del esfuerzo que debe realizar el profesor para corregir estos trabajos, más si tenemos en cuenta el elevado número de alumnos y que la corrección no es trivial ni de respuesta única. Por ejemplo, para que el alumno adquiriera la primera competencia es necesario que realice diagramas ER o diagramas de clases. Si tenemos en cuenta que cada uno de ellos puede tener muchos elementos (entidades, relaciones, atributos, etc.) y que la solución correcta no tiene por qué ser única, nos encontramos que la corrección de un ejercicio y las anotaciones oportunas pueden llevar varios minutos. Si a esto le añadimos que el alumno deberá realizar muchos ejercicios para su aprendizaje y que los grupos suelen ser numerosos, el tiempo dedicado por el profesor para la corrección y seguimiento del trabajo de los alumnos es considerable. A menudo, esta mayor dedicación docente no es asumible por parte del profesor.

En la última década para ayudar a los profesores en sus labores docentes han aparecido muchas plataformas de e-learning que permiten tanto la gestión de las tareas académicas (gestión de cursos, de alumnos, de actividades, de material, etc.) como de la evaluación automatizada. Como hemos visto en el capítulo precedente, en un entorno CBA el propio sistema es quien realiza las tareas de evaluación normalmente a través de preguntas de respuesta fija por su sencillez tanto en la preparación como en los sistemas que las deben corregir. El principal handicap de estos sistemas es que sólo permiten la corrección de los niveles inferiores de conocimiento de la taxonomía de Bloom.

En el contexto de una asignatura de BD, las competencias que el alumno debe adquirir en esta materia exigen al profesor evaluar niveles de conocimiento intermedios/altos de la taxonomía de Bloom. Tal como veremos en los siguientes capítulos, se necesita evaluar niveles cognitivos de comprensión y especialmente de aplicación y análisis. Para poder realizar una evaluación de estos niveles de conocimiento a través de un entorno CBA, se necesitan herramientas muy especializadas no disponibles en las plataformas de e-learning habituales. Por ejemplo, en el entorno de BD y para el logro de las

competencias mencionadas, es necesario que el sistema permita la realización de diagramas ER y/o diagramas de clase para la realización del diseño conceptual. Por otra parte el sistema debe ser capaz de validar un esquema de BD y su normalización. Finalmente también debe permitir la entrada de diferentes consultas, tanto en SQL como en álgebra relacional, de las que el sistema debe verificar su corrección.

Además, como estas herramientas deben facilitar la evaluación formativa de los estudiantes, tienen que disponer de las funcionalidades oportunas para realizar la corrección y evaluación automática de los distintos ejercicios. También deben facilitar de forma automática el feedback oportuno a cada ejercicio y el profesor debe poder realizar un preciso seguimiento del trabajo realizado. Por otra parte nos interesa que las herramientas se desarrollen en un entorno web, sencillo de utilizar y que simplemente se necesite una conexión a Internet y de cualquier navegador, sin tener que descargar e instalar software alguno. De esta forma el sistema debe actuar como un tutor virtual que ayude al alumno en el desarrollo de su trabajo en cualquier sitio y a cualquier hora. En este contexto, y tal como se ha comentado en la sección 2.2.4, una de las limitaciones de los sistemas CBA es que no sabemos quién realmente resuelve los ejercicios enviados al sistema, siendo necesaria la realización de exámenes/pruebas de validación presenciales. Aunque el peso que se asigna a estas pruebas puede ser mínimo, su realización es necesaria para asegurar que el alumno no nos haya engañado y que ha alcanzado las competencias planificadas. Con el objetivo de conseguir un auténtico CBA el sistema debe ser capaz de puntuar estas pruebas presenciales con una nota que debe ser muy parecida a la que obtendría un profesor y facilitando la evaluación sumativa tradicional. De esta manera el profesor conseguirá un ahorro de tiempo considerable.

El trabajo de esta tesis se enmarca en el estudio, análisis, desarrollo y utilización de estas herramientas para valorar las mejoras que aportan. Para ello ha sido necesario integrarlas en una plataforma de e-learning. La plataforma de e-learning considerada en esta tesis es ACME (“Avaluació Continuada i Millora de l’Ensenyament”).

### 3.2 NUESTRA PROPUESTA

Nuestra propuesta se centra en el desarrollo de un CBA que facilita el aprendizaje y evaluación de los principales temas de una asignatura de BD dentro del marco de trabajo que se ha comentado en la sección anterior. Tal como se ha comentado en la sección 2.3, entre estos temas destacan:

- **Diseño conceptual:** El alumno debe ser capaz de realizar diagramas ER y de clases a un supuesto dado. Dentro de la plataforma ACME será necesario desarrollar módulos que permitan la corrección/evaluación de estos diagramas.
- **Diseño lógico:** El alumno debe obtener el esquema de una BD relacional. Habrá que desarrollar un módulo específico para la verificación de esquemas de BD.
- **Normalización:** Los esquemas de bases de datos deben estar normalizados. Podremos aprovechar el mismo entorno anterior para comprobar que los esquemas estén normalizados.
- **Consultas en álgebra relacional:** Para una mejor comprensión del modelo relacional y de SQL es importante el aprendizaje de álgebra relacional. Desarrollaremos un entorno para la corrección de consultas expresadas en álgebra relacional.
- **SQL:** Como lenguaje estándar de uso del modelo relacional, el alumno debe conocer perfectamente sus comandos. Será necesario desarrollar un módulo específico para su corrección.

La Figura 3.1 muestra las líneas generales de trabajo de esta tesis. Este conjunto de módulos se denominan ACME-DB y tal como se muestra en la Figura 3.2, amplían la tipología de problemas que se podrán evaluar desde la plataforma ACME.

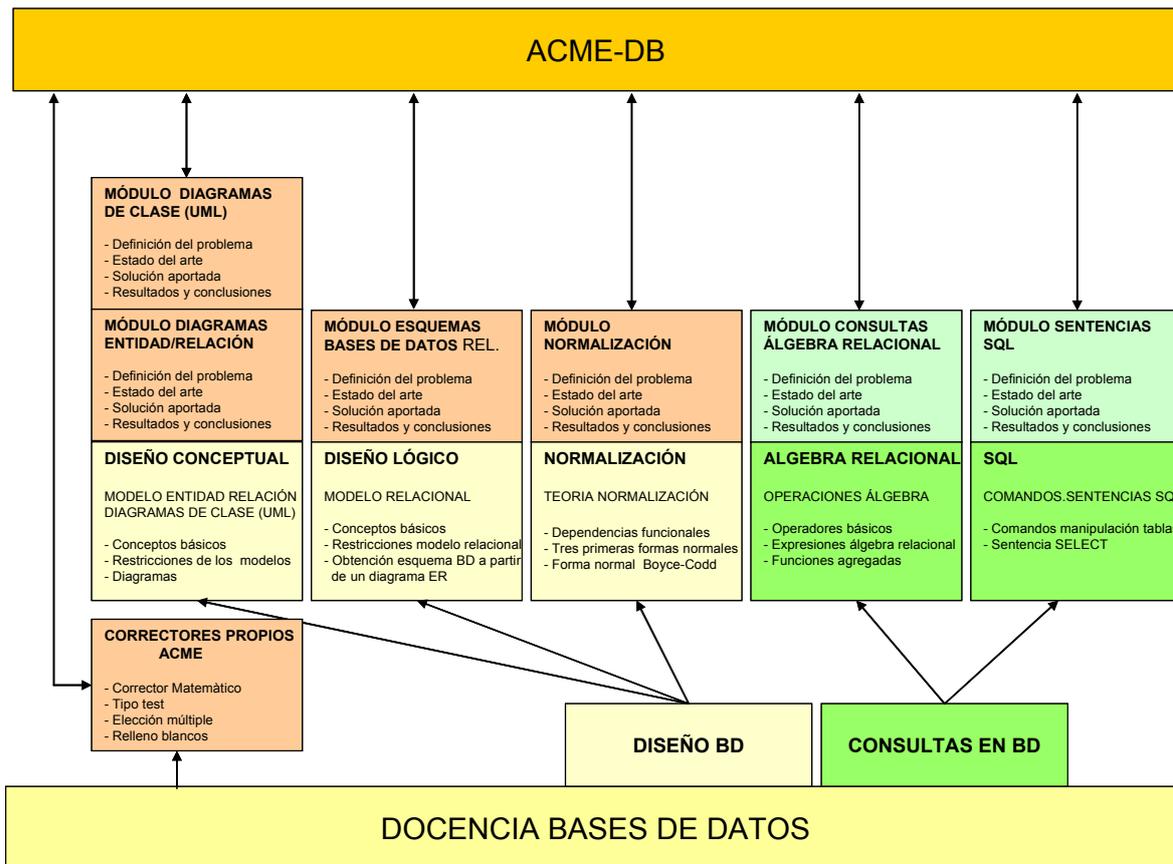


Figura 3.1 Líneas generales de trabajo.

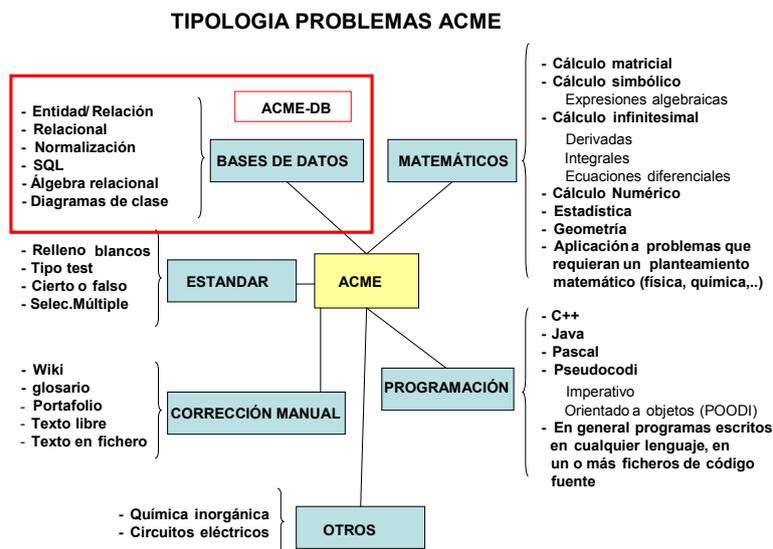


Figura 3.2 Tipología problemas ACME.

Los otros temas que se desarrollan en una asignatura de BD no requieren de módulos específicos y se podrán evaluar a partir de las funcionalidades/correctores propios de ACME, por ejemplo del corrector matemático, de preguntas de elección múltiple, etc. y no son tratados en esta tesis.

Finalmente remarcar que en el desarrollo de estos módulos se ha querido aportar mejoras a los sistemas existentes, buscando siempre la simplicidad y una fácil utilización. Así pues, las características de partida han sido:

- **Entorno integrado para la docencia de bases de datos.** El mismo entorno debe facilitar el aprendizaje de todos los temas, con la misma metodología de trabajo para el alumno y con interfaces adecuadas a cada tema. Valoramos muy positivamente que los distintos correctores se integren en una única plataforma ya que facilita el aprendizaje a los alumnos y nos permite seguir todo su trabajo y evolución. Este entorno integrado es el que llamamos ACME-DB.
- **Web Based tool.** Para facilitar el trabajo al alumno es importante que sólo requiera de una conexión a Internet y un navegador, sin ninguna necesidad de descarga de software ni instalación alguna. Además también valoramos positivamente que no se requiera de software adicional.
- **Corrección y feedback automáticos.** De caras a la evaluación formativa, es muy importante que el sistema corrija automáticamente y ofrezca el feedback oportuno al alumno.
- **Puntuación automática del ejercicio.** Pensando en la evaluación sumativa del alumno es deseable que la plataforma ofrezca una puntuación de cada ejercicio lo más parecida posible a la que obtendría un profesor en una corrección no automatizada.
- **Evaluación automática del alumno.** El mismo entorno debe proporcionar la evaluación final de todo el trabajo realizado por el alumno, facilitando la tarea al profesor. Además valoramos positivamente que estas notas se puedan exportar a formatos típicos, como por ejemplo Microsoft Excel.
- **Trabajo colaborativo.** El entorno debe ofrecer funcionalidades que permitan el trabajo en grupo. Las nuevas metodologías docentes muestran los buenos resultados que se obtienen con este tipo de trabajo.

### 3.3 METODOLOGÍA DE TRABAJO

El desarrollo de un entorno CBA capaz de llevar a cabo la corrección de este tipo de problemas no es una tarea trivial. Como se ha comentado en la sección 2.2.5, para la evaluación de los niveles cognitivos intermedios/superiores se deben utilizar ejercicios de respuesta libre. Para ello es preciso disponer de entornos especializados según el tema a tratar.

De forma simplificada, para el desarrollo de un entorno CBA que requiera la resolución de ejercicios de respuesta libre será necesario integrar varios apartados básicos:

- **Creación de ejercicios.** Cada ejercicio consistirá como mínimo de dos partes:
  - **Enunciado.** Es la parte del ejercicio que visualizará el alumno
  - **Elementos de corrección.** Que se utilizarán para la corrección automática y que dependen del tipo de ejercicio.
- **Entorno de e-learning.** Que nos facilite las funcionalidades típicas de una plataforma de e-learning: gestión de cursos, de alumnos, asignación de problemas, etc. En nuestro caso la plataforma ACME nos facilita todas estas funcionalidades.
- **Interfaces para la entrada de soluciones.** Son diseñadas específicamente para cada tipo de ejercicios.
- **Módulo de corrección.** Nos permite la corrección automática del ejercicio, el cálculo de la nota y el retorno del feedback oportuno.

En el caso de ejercicios de respuesta fija, por su simplicidad, la definición de estos apartados no presenta ningún tipo de dificultad. Ahora bien, en los ejercicios de respuesta libre estos apartados están muy ligados al tipo de ejercicio que se trata, por lo cual deben diseñarse específicamente para cada caso. Este es uno de los factores que limitan su aplicación, especialmente en el ámbito científico/técnico donde se pueden plantear muy distintas tipologías de problemas.

A pesar de esta dificultad, la creación de un entorno de CBA que soporte ejercicios complejos, como es nuestro caso, puede abordarse aplicando una metodología común. La metodología de trabajo que se ha llevado a cabo ha consistido en:

- Realizar el estado del arte en cuanto a entornos existentes con objetivos parecidos a los nuestros (véase Capítulo 2).
- Estudiar y analizar el problema a resolver y definir los requisitos del módulo. Para ello es necesario definir:
  - Analizar las características del problema a resolver, qué información se debe presentar al alumno y cómo.
  - Analizar las características de las soluciones y qué herramientas/interfaces son las más adecuadas para poderlas entrar al sistema.
  - Definir la estructura de los ejercicios.
  - Diseñar las interfaces más adecuadas para la entrada de las respuestas.
  - Definir la estrategia de corrección, de retorno de feedback y de puntuación.
- Implementar el módulo e integrarlo a la plataforma ACME.
- Crear una amplia colección de problemas. Para saber si el sistema realmente aporta mejoras a la docencia de bases de datos, se ha creado una amplia colección de ejercicios de cada tema. Estos problemas han sido asignados a los alumnos de bases de datos en los últimos cursos y con ellos se ha realizado un amplio abanico de actividades.
- Analizar los datos obtenidos, tanto de la resolución/evaluación de los problemas como de las encuestas realizadas a alumnos y profesores.
- Obtener conclusiones. A partir del análisis de los datos obtenidos por la plataforma y de los resultados académicos obtenidos después de varios años de utilización.

Así pues, en nuestro trabajo y para cada uno de los temas a desarrollar se ha realizado un análisis de la tipología de ejercicio a resolver, se han evaluado los sistemas existentes, se ha definido el problema a resolver, se ha desarrollado una solución y se ha puesto en marcha en un entorno real para comprobar su validez. En los siguientes capítulos se detalla cada uno de estos seis módulos.



## **CAPÍTULO 4. MÓDULO DE DIAGRAMAS ENTIDAD/RELACIÓN**

En este capítulo presentaremos el primero de los módulos correctores desarrollados: el módulo de diagramas entidad/relación (ERD). Este módulo dará soporte al aprendizaje y docencia del diseño conceptual, correspondiente a la primera etapa del diseño de una base de datos.

Este capítulo se ha estructurado de la forma siguiente. En la sección 4.1 se hace una introducción al tema. En la sección 4.2 se describe el modelo ER y a continuación en la sección 4.3 se detallan las necesidades y la problemática en la enseñanza del diseño conceptual. En la sección 4.4 se hace un breve resumen de entornos de corrección de diagramas ER. Posteriormente en la sección 4.5 plantearemos el problema que nos proponemos resolver, describiremos detalladamente la solución aportada y su integración a la plataforma ACME. En la sección 4.6 se realiza una evaluación del módulo desarrollado, presentando las actividades y resultados obtenidos. Finalmente en la sección 4.7 se describen las conclusiones a las que hemos llegado.

### **4.1 INTRODUCCIÓN**

La primera fase para el diseño de una base de datos es la obtención y análisis de requisitos que deberá satisfacer la aplicación a desarrollar. Una vez analizados estos requisitos la segunda fase es el diseño conceptual de la BD. Para ello se han propuesto diferentes metodologías, siendo la basada en el modelo Entidad/Relación (ER: Entity-Relationship) [Che76] la más utilizada.

El modelo ER es un modelo conceptual de datos de alto nivel que se basa en identificar y representar en un Diagrama Entidad/Relación (ERD) las entidades que intervienen, sus relaciones y las restricciones entre éstas. En cualquier curso básico de BD el diseño conceptual es una de las competencias que el alumno debe adquirir. Con este fin, el profesor plantea supuestos parecidos a un entorno real a partir de los cuales el alumno deberá realizar el diagrama, identificando entidades, relaciones, atributos, etc. Si la asignatura de Bases de Datos precede a las de Ingeniería del Software es el profesor quien especifica los requisitos de datos a modelar ya que todavía el alumno desconoce cómo obtenerlos.

Generalmente, la metodología docente que se aplica consiste en plantear ejercicios que requieran la aplicación de los conceptos teóricos del modelo. El alumno deberá analizarlos y realizar el correspondiente ERD. Para poder evaluar si ha adquirido la competencia, será necesario que se enfrente a diferentes supuestos reales. En estos supuestos el número de entidades, relaciones, atributos, etc. es considerable y además suelen haber varias soluciones que se pueden considerar correctas.

El proceso de evaluación formativa consiste en la revisión de los diagramas de cada alumno y en la realización de las anotaciones correspondientes para que el alumno pueda aprender de sus propios errores. Con el feedback obtenido en la corrección, el profesor adapta las diferentes sesiones, haciendo hincapié en los puntos donde detecte deficiencias de aprendizaje. Como se puede apreciar estas tareas de revisión son laboriosas y conllevan una gran dedicación difícilmente asumible por el profesor, dificultando la aplicación del sistema de evaluación formativa.

Por otra parte las actividades de diagramación las podemos enmarcar en los niveles cognitivos intermedios (aplicación y análisis) dentro de la taxonomía de Bloom. Tal como se ha comentado en el capítulo 2, en un entorno de CBA es muy difícil evaluar estos niveles cognitivos a través de preguntas de respuesta fija y se requiere de entornos mucho más especializados.

Nuestro objetivo es solventar estas limitaciones diseñando y desarrollando un entorno CBA especializado que nos permita la evaluación automatizada de los niveles cognitivos intermedios de la taxonomía de Bloom en materia de ERD. El sistema debe facilitar la evaluación tanto formativa como sumativa del modelo ER. Para ello debe permitir definir problemas ERD, su corrección y generación de feedback automáticos. El sistema debe soportar la evaluación formativa de manera que el profesor pueda controlar el trabajo del grupo y no pierda tiempo en la tediosa labor de corrección. Además el sistema

proporcionará una puntuación automática de cada ejercicio para poder realizar la evaluación sumativa. Con las puntuaciones obtenidas el sistema facilitará una evaluación final del alumno.

Este entorno para el aprendizaje y evaluación de ERD debe integrarse en la plataforma ACME actuando conjuntamente como un sistema CBA y con el que se realizarán las correspondientes pruebas sobre los alumnos de la asignatura de BD. El objetivo es valorar si su utilización aporta mejoras en el aprendizaje de la realización de ERD's. En este capítulo se detalla ACME-ER, el primer módulo de los seis que conforman ACME-DB.

## 4.2 DESCRIPCIÓN DEL MODELO ENTIDAD/RELACIÓN

El modelo ER [Che76] es un modelo conceptual de datos de alto nivel muy utilizado que se basa en identificar las entidades que intervienen en el sistema, sus relaciones y las restricciones entre éstas. Aunque la representación gráfica del modelo varía según los autores, nosotros hemos escogido la notación original de Chen. Veamos a continuación de forma resumida los conceptos del modelo ER.

El modelo ER se basa en un conjunto de conceptos básicos que se representan a través de una notación determinada. Para la visualización de la notación ver Figura 4.1. Los conceptos básicos son:

- **Entidad:** Es un objeto (o un hecho) que existe, que es diferenciable de los otros y del que es necesario que el sistema guarde información de sus propiedades (características).
- **Atributos:** Son las propiedades o características específicas de cada entidad. Para cada entidad sus atributos tendrán un determinado valor. En el modelo ER los atributos se representan con una elipse con su nombre en el interior. Los atributos derivados se representan con una elipse de puntos, los multivaluados con una doble elipse y los compuestos con sus atributos simples.
- **Conjunto o tipos de entidades:** Un tipo de entidades define a un conjunto de entidades con los mismos atributos. En el modelo ER se representan a través de un rectángulo con las diferentes elipses que son los atributos de este tipo de entidad. A menudo, y para simplificar, a los tipos de entidades se les llama entidades sin más, sobreentendiendo que ya se hace referencia a todas las entidades individuales.
- **Dominio de un atributo:** Representa el conjunto de valores permitidos para este atributo.
- **Atributo clave primaria:** Es el atributo o conjunto de atributos tal que sus valores identifican de manera única una determinada entidad. En el modelo ER se representan subrayando su nombre dentro de la elipse. Una restricción de la clave primaria es que no podrá tener un valor nulo.

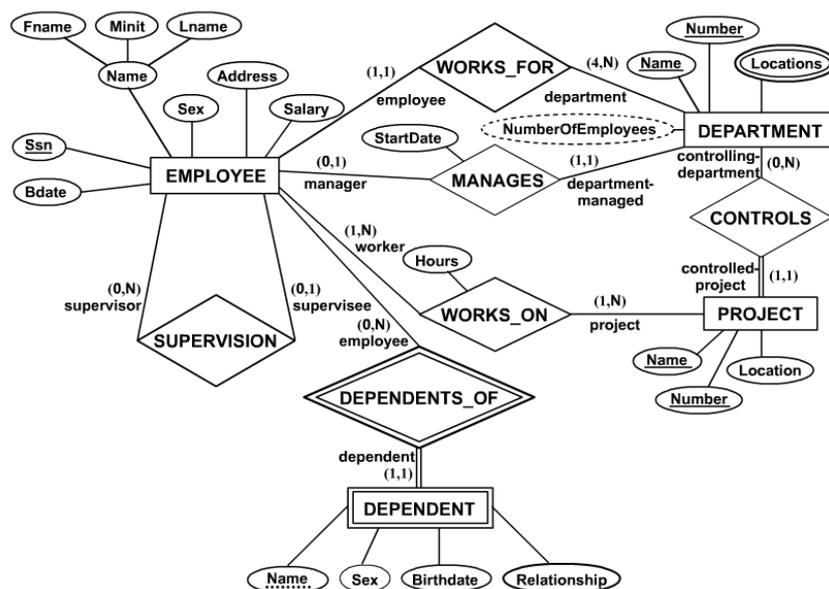


Figura 4.1 Notación modelo ER.

- **Relación:** Representa una asociación entre diversas entidades.
- **Conjunto o tipos de relaciones:** Representan al conjunto que contiene todas las relaciones del mismo tipo. Normalmente el conjunto de relaciones implica a dos conjuntos de entidades (relaciones binarias), pero pueden implicar a más. En el modelo Entidad/Relación los conjuntos de relaciones se representan con un rombo que une a los tipos de entidades relacionadas y debajo el nombre. A menudo, y para simplificar, a los conjuntos de relaciones los llamamos relaciones, sin más, entendiendo que ya se hace referencia a todas las relaciones individuales del conjunto. Los tipos de relaciones suelen tener restricciones que limitan las posibles combinaciones de las entidades que intervienen en la relación. Podemos distinguir dos tipos de restricciones: por razón de cardinalidad y por participación.
  - **Restricciones por razón de cardinalidad.** Especifican el número de relaciones en las que puede intervenir una entidad. En el caso de relaciones binarias podemos tener los siguientes casos:
    - **Relaciones una a una (1:1).** Una entidad del tipo A está asociada a una entidad del tipo B y viceversa. La representación en el modelo ER se realiza situando un número 1 a cada lado del rombo representativo de la relación.
    - **Relaciones una a muchas (1:n).** Una entidad del tipo A está asociada a un número cualquiera de entidades del tipo B, mientras que una entidad de B, como mucho, sólo puede estar asociada a una entidad de A. Se representa de la misma forma anterior pero colocando un 1 al lado de la entidad A y un n al lado de la relación B.
    - **Relaciones muchas a muchas (n:m).** Una entidad del tipo A está asociada a un número cualquiera de entidades del tipo B, y una entidad de B también puede estar asociada con muchas de A.

Otra manera de representar la cardinalidad es a través de la notación (min,max) en la que se especifica el número mínimo y máximo de entidades que pueden intervenir en cada relación.

- **Restricción de participación.** Especifica si todas las entidades de los dos conjuntos participan en la relación o si sólo intervienen unas cuantas. Con relación a la participación podemos distinguir dos tipos de restricciones:
  - **Total.** En que toda entidad de A está relacionada con alguna entidad de B.
  - **Parcial.** En que no todas las entidades de A han de estar relacionadas con alguna entidad de B. Se pueden representar colocando un (0,1) o un (0,n) en el lado correspondiente de la relación.
- **Relaciones reflexivas.** Son relaciones de un tipo de entidad consigo mismo.
- **Atributos de las relaciones:** Una relación puede tener atributos propios. En el modelo Entidad/Relación se representan con una elipse conectada al rombo que representa el conjunto de relaciones.
- **Grado de un tipo de relación:** Es el número de tipos de entidades que participan en la relación. Generalmente se suele trabajar con relaciones binarias (entre dos tipos de entidades), pero se puede extender a relaciones de orden superior, ternarias o en general n-arias.
- **Entidades débiles.** Los tipos de entidad que no tienen atributos clave propios se denominan tipos de entidad débiles, en contraste con los tipos de entidad regulares (fuertes). Las entidades que pertenecen a un tipo de entidad débil se identifican por su relación con entidades de otro tipo de entidad, en combinación con alguno de sus atributos. Un tipo de entidad débil siempre tiene una restricción de participación total con respecto al tipo de entidad fuerte del que depende. Su representación en el modelo ER se realiza mediante un doble rectángulo.

Una vez descritos los conceptos básicos del modelo ER veamos un ejemplo que nos servirá de muestra y ejemplo para los correctores. El ejemplo a considerar es el típico supuesto de una empresa [EN07] y que podemos resumir su enunciado como:

*Realizar el diagrama ER que represente el modelo conceptual de una base de datos para una empresa con los siguientes requisitos.*

- *La empresa está organizada en departamentos. De cada departamento nos interesa saber su número, el nombre y el teléfono. Cada departamento tiene asignados empleados de los que nos interesa saber su nombre completo, NIF, dirección completa, sueldo, sexo y fecha*

nacimiento. El trabajo de cada empleado está supervisado por otro empleado. Uno de los empleados del departamento desempeña la función de director del departamento.

- Cada departamento controla un cierto número de proyectos. De cada proyecto nos interesa saber su código, su nombre y la ciudad donde se realiza. Cada proyecto es controlado por un único departamento.
- Un empleado puede trabajar en varios proyectos. Nos interesa saber el número de horas que cada empleado dedica a cada proyecto.
- Un empleado puede tener varias personas dependientes que figuran en su cartilla de la seguridad social. De cada una de estas personas nos interesa saber su número de orden dentro de la cartilla, su parentesco con el empleado, su nombre y su fecha de nacimiento.

El sistema diseñado nos debe permitir:

- A partir de un departamento saber todos los empleados que tiene asignados, saber quien es su director y los proyectos que controla.
- A partir de un empleado saber sus datos personales, los datos de su supervisor, en que proyectos ha trabajado y cuantas horas ha dedicado a cada proyecto. También nos interesa saber las personas dependiente que tiene a su cargo.
- A partir de un proyecto saber los datos de este, qué departamento lo controla, los empleados que están trabajando en él y las horas dedicadas.

Un diagrama que representa esta situación sería el mostrado en la Figura 4.2. Contiene todos los elementos básicos descritos anteriormente. Para su simplicidad solamente se muestran los atributos clave.

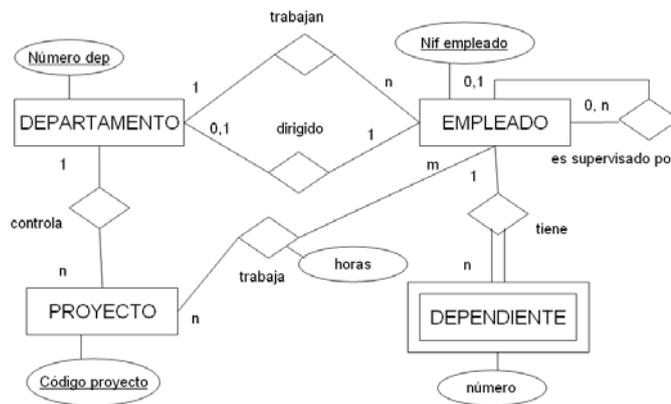


Figura 4.2 Diagrama ER-1 del supuesto planteado.

La Figura 4.3 representa una segunda solución correcta en la que se ha considerado la entidad DEPENDIENTE fuerte en vez de débil. Aunque varía poco de la anterior debemos considerarla correcta. En general puede haber varias soluciones correctas aunque especificando claramente los enunciados podemos reducirlas considerablemente. Este hecho dificulta la tarea de corrección ya que por un lado es necesario comparar la solución enviada con todas las correctas y por otro lado, en caso de tratarse de una solución incorrecta, se deberá retornar el feedback según la solución correcta más parecida.

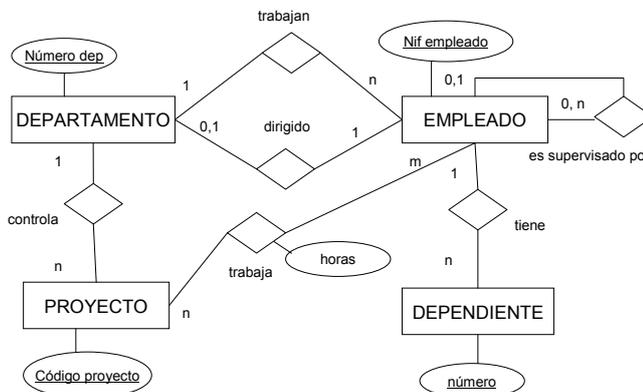


Figura 4.3 Diagrama ER-2 del supuesto planteado.

### **4.3 NECESIDADES Y PROBLEMÁTICA EN LA ENSEÑANZA DEL DISEÑO CONCEPTUAL**

Es muy importante que el alumno sepa realizar correctamente el diseño conceptual de una BD ya que una de las tareas típicas de los ingenieros informáticos es el diseño de BD. En las empresas se encontrarán con herramientas CASE (Computer Aided Software Engineering) que permiten el diseño conceptual y que a partir de los diagramas realizados ya van a obtener el código correspondiente para la generación de las tablas en el modelo relacional. En el ámbito de la educación universitaria en materia de BD, una de las competencias que el alumno debe adquirir es la realización del diseño de una BD.

Después de muchos años impartiendo docencia en cursos de BD hemos observado y constatado las dificultades en el aprendizaje del diseño de bases de datos, especialmente en el diseño conceptual. Los estudios sobre el aprendizaje del modelo ER, llevados a cabo en [BS93] [BA94], muestran que aunque los alumnos no suelen tener muchos problemas en la detección de las entidades si suelen tener problemas en las relaciones y su cardinalidad, en las entidades débiles y en general en las relaciones n-arias.

La enseñanza tradicional del diseño conceptual pasa por unas sesiones teórico/prácticas donde el profesor explica los conceptos básicos del modelo ER y en qué consiste un ERD. A partir de ahí realiza diferentes ejemplos para aplicar y consolidar los conocimientos explicados. Puesto que en la realización de un ERD se integran todos los conocimientos del modelo ER y se plasma la capacidad del alumno para la realización del diseño conceptual, el profesor evalúa al alumno en función de si es capaz de realizar el correspondiente ERD a partir de un supuesto planteado. La adquisición de esta habilidad por parte del alumno pasa por un proceso de aprendizaje que siguiendo la taxonomía de Bloom podemos sintetizar en los siguientes apartados:

- **Conocimiento.** El profesor define y describe en clase los diferentes componentes que conforman el modelo ER. El alumno debe ser capaz de recordar, definir y describir los conceptos de entidad, atributo, relación, cardinalidad, etc.
- **Comprensión.** A través de los ERD el alumno asocia entidades y relaciones, da ejemplos de diferentes tipos de relaciones, clasifica los atributos según las entidades, expone cuáles van a ser las claves, etc.
- **Aplicación.** El alumno aplica los conceptos del modelo a un ejemplo sencillo perfectamente detallado. Construye diagramas sencillos, identifica los elementos, determina las cardinalidades, etc.
- **Análisis.** El alumno debe ser capaz de analizar un supuesto real que se le plantea, identificar las entidades existentes y como están relacionadas, identificando sus atributos, las claves y la cardinalidad de las relaciones. El alumno debe realizar una abstracción del problema que se plantea y plasmarlo en un diagrama.

Ahora bien, para la adquisición de estas habilidades, especialmente la de análisis, es necesario que el alumno se ejercite realizando muchos ERD sobre diferentes situaciones reales. Aquí es donde aparece el punto débil de la metodología tradicional: corregir muchos diagramas a muchos alumnos ofreciendo las anotaciones oportunas, conlleva una gran dedicación del profesor que difícilmente puede llevar a cabo. En este aspecto los sistemas de e-learning especializados en corrección de diagramas juegan un papel importante.

### **4.4 CORRECCIÓN AUTOMÁTICA DE DIAGRAMAS ER**

Debido a la importancia del modelo ER han sido muchos los entornos realizados para su aprendizaje (ver sección 2.4.1 donde se comentan los más relevantes). El problema visto desde un punto de vista general se puede tratar como la interpretación y corrección de cualquier tipo de diagramas. En este tipo de problemas, uno de los principales retos es interpretar la semántica de la solución enviada por el alumno para identificar entidades, atributos y relaciones. Otro de los problemas es que se puede aceptar más de una solución como correcta (véase ejemplo sección 4.2).

Los entornos evaluados, DEAP [WTS04,TWS07,WTS07] y ABC [TSW05], se centran principalmente en la puntuación automática de diagramas, pero no están pensados para llevar a cabo una evaluación

formativa. El equipo liderado por A. Mitrovic dispone de Kermit [SM02] para el aprendizaje del modelo ER que corrige pero no puntúa. Por otra parte [CS00] ofrece una herramienta para el aprendizaje colaborativo pero no corrige ni puntúa automáticamente. Datsys, [Tsi02][HST02][HB06], sí que ofrece todas las ventajas de los CBA permitiendo tanto una evaluación formativa como sumativa. La limitación de este trabajo es que está pensado para la corrección de cualquier tipo de diagramas, entre ellos ERD y diagramas de clase, pero no tratan el resto de ejercicios que se ofrecen en un curso de BD.

Tal como se ha comentado en la sección 2.4.7 otra de las limitaciones de estos entornos (con la salvedad de los trabajos liderados por A. Mitrovic) es que sólo permiten el trabajo en un tema concreto, en este caso los ERD, pero no abordan problemas de diseño lógico (esquemas de BD y normalización) ni tampoco ejercicios de consultas sobre BD.

Dadas estas limitaciones, nuestro objetivo en este trabajo es desarrollar un entorno integrado en materia de BD, que facilite todas las prestaciones de un CBA, desde la entrega de materiales y tutoriales, hasta la asignación de ejercicios, que una vez resueltos serán corregidos de forma automática, retornando el feedback adecuado y obteniendo la puntuación pertinente. En este capítulo se detalla el módulo correspondiente al modelo ER.

## **4.5 ACME-ER. Módulo de diagramas ER**

En esta sección se detalla el primer módulo referente a la evaluación de diagramas ER que denotamos como ACME-ER. En la sección 4.5.1 se determinan los requisitos que deberá satisfacer el sistema. En la sección 4.5.2 se presenta una visión general de la solución desarrollada. En las cuatro siguientes secciones se detallan los aspectos más técnicos del módulo: la estructura de un problema ERD (sección 4.5.3), su interfaz (sección 4.5.4), el módulo de corrección y feedback (sección 4.5.5) y finalmente la puntuación del problema (sección 4.5.6).

### **4.5.1 Requisitos del sistema ACME-ER**

El problema que se plantea es desarrollar un sistema que de forma totalmente automática sea capaz de evaluar un diagrama ER de un supuesto planteado (véase ejemplo sección 4.2). Debe tenerse en cuenta que un mismo supuesto puede tener varias soluciones correctas, por lo que el sistema deberá contemplar que la solución puede no ser única (véase Figuras 4.2 y 4.3). Los requisitos que definiremos para diseñar nuestro módulo corrector son los siguientes:

- El sistema deberá contemplar todos los elementos que integran un diagrama ER, es decir los descritos en la sección 4.2
  - Las entidades que intervienen, ya sean regulares (fuertes) o débiles.
  - Los atributos de cada entidad, especificando los que forman la clave principal.
  - Las relaciones existentes y entre qué entidades.
  - Las restricciones, tanto de cardinalidad (1:1, 1:n, n:m) como de participación (total o parcial).
  - Los atributos de las relaciones.
  
- El sistema deberá facilitar las funcionalidades siguientes:
  - Disponer de una interfase apropiada para la entrada de diagramas ER que deberá permitir:
    - Dibujar (y modificar/borrar) entidades fuertes, permitiendo la asignación de atributos y atributos clave.
    - Dibujar (y modificar/borrar) entidades débiles, permitiendo la asignación de atributos, atributos clave parcial y especificar de qué entidad fuerte depende.
    - Permitir la asignación de un nombre cualquiera a las entidades y relaciones.
    - Dibujar (y modificar/borrar) las relaciones, señalando entre qué entidades están definidas y su cardinalidad.

- El sistema deberá facilitar la evaluación formativa. Cuando el alumno envíe una respuesta, el sistema recogerá la información asociada a los distintos objetos dibujados y los enviará al servidor. Esta información será guardada por ACME como una solución enviada por el alumno a uno de sus ejercicios. El módulo desarrollado comparará la solución enviada con las soluciones correctas asociadas al problema. En caso que sea correcta se lo notificará al alumno mediante un “Correcto”. En caso de que el corrector determine que es incorrecta facilitará pequeñas notificaciones (feedback) que se determinan en función del error cometido y con la finalidad de ayudar al alumno a encontrar el fallo y poder enviar una nueva solución.
- El sistema deberá facilitar una puntuación a cada ejercicio. En las actividades de evaluación sumativa el sistema debe facilitar la nota obtenida.
- El sistema deberá integrarse en la plataforma ACME por lo que se compartirán todas las funcionalidades del ACME, como por ejemplo la gestión de cursos, asignaturas, temas, actividades, alumnos, envío de mensajes, foros, chats, etc.

## 4.5.2 Análisis inicial de ACME-ER

En el momento de analizar el módulo ACME-ER, el primer problema a resolver es cómo identificar el nombre de las entidades y de los atributos, ya que el alumno puede asociarles el nombre que quiera. Por ejemplo, un alumno puede asignar el nombre EMPLEADO a una entidad pero otros preferirán EMPLEAT, EMPLOYEE, TRABAJADOR, EMPL o simplemente E, T, o cualquier otra identificación. Igual pasa con los atributos. La identificación del lenguaje natural es un problema complejo y difícilmente podemos contemplar toda la casuística. Algunos trabajos como [GSD99], [AP00] y [OHM04] intentan obtener diagramas ER a partir de especificaciones de lenguaje natural. Ninguno de estos trabajos tiene finalidades académicas y están enmarcados en líneas de investigación de ingeniería del conocimiento. En los entornos de corrección de diagramas este problema se suele solucionar de varias formas:

- Relacionando cada parte del diagrama con la parte del enunciado que hace referencia. A través de esta asociación se van identificando los diferentes elementos del diagrama. Los sistemas Kermit descritos en [SM02] y [BH07] utilizan esta técnica.
- Utilización de una lista de sinónimos. Se identifican los distintos elementos a partir de una lista de posibles sinónimos. Los sistemas descritos por DEAP [WTS07] y DATSys [Tsi02] utilizan esta técnica.

Para solventar este problema, la solución que nosotros hemos adoptado ha sido facilitar el nombre de los diferentes atributos especificándolos en la descripción del enunciado. Simplemente mostramos los nombres de los atributos entre paréntesis. Facilitando esta información no ayudamos apenas al alumno, en todo caso le especificamos de forma más concisa los atributos que debe guardar el sistema. Por el contrario no ponemos ninguna limitación respecto al nombre de las entidades y de las relaciones y el alumno puede asignarles el nombre que desee. Tampoco le especificamos el atributo clave principal que debe identificar. A partir de las claves principales el sistema identificará las entidades y a partir de éstas las relaciones entre ellas.

Otra restricción de nuestro sistema es que no permitimos las relaciones n-arias. Consideramos que el alumno puede descomponer una relación n-aria en otras binarias. Tampoco aceptamos atributos derivados, compuestos y multivaluados, trabajando solamente con atributos simples que serán los que en definitiva se almacenarán en la BD. Estas mismas limitaciones se presentan también en la mayoría de herramientas CASE comerciales que permiten la realización de este tipo de diagramas y en todos los entornos evaluados en la sección 2.4.1.

Fijadas estas restricciones ya estamos en condiciones de diseñar el modelo. Este proceso irá ligado a la estructura de la plataforma ACME e implicará la definición de:

- **La estructura del problema.** Cada problema se guarda en un fichero plano en el repositorio con una determinada estructura y con la información para su visualización y corrección. En el caso de los ejercicios ER este fichero está formado por un enunciado y las

posibles soluciones correctas. El profesor escribirá cada problema mediante un editor cualquiera. A continuación y con las funcionalidades específicas de ACME lo incorporará al repositorio y validará que su estructura sea la correcta. A partir de aquí ya estará disponible para ser utilizado.

- **Una interfaz apropiada.** A través de ella el alumno visualizará el enunciado, introducirá las soluciones dibujando el ERD y recibirá el feedback.
- **Un módulo corrector.** Interpretará la solución entrada por el alumno y la comparará con las soluciones correctas descritas en la estructura del problema y que están guardadas en el repositorio. Este módulo corrector determina el feedback a retornar al alumno. El propio sistema ACME se ocupa de guardar toda esta información y a partir de ella se facilita, tanto a profesor como alumno, la evaluación formativa en esta materia.
- **Un sistema de puntuación** que facilita una nota entre 0 y 10 en función de los errores cometidos.

A continuación analizaremos en detalle cada uno de estos puntos.

### 4.5.3 Estructura de un problema tipo ERD

La estructura de un problema debe contener la información que debe mostrarse al alumno y la información necesaria para poder corregir automáticamente la solución enviada. Esta información se mantiene en un fichero que queda guardado en el repositorio del sistema. En el caso de los ejercicios ERD la estructura del fichero es la que se muestra a continuación:

```
<identificador de tipo de problema>
<E> enunciado </E>
<SOLUCIO1_ER> Solución 1 </SOLUCIO1_ER>
<SOLUCIO2_ER> Solución 2 </SOLUCIO2_ER>
.....
```

Como puede observarse las diferentes partes se diferencian a partir de marcas (tags). Inicialmente tenemos un identificador del tipo de problema. A continuación el enunciado y posteriormente las diferentes soluciones propuestas por el profesor. En el enunciado se incluye el nombre de los atributos que figura entre paréntesis. Para cada solución se especifican los siguientes ítems:

- **Todas las entidades regulares (fuertes).** Para cada una de ellas le asignamos un nombre, la lista de todos los atributos y cual es el atributo clave.
- **Todas las entidades débiles.** Para cada una de ellas debe especificarse su nombre, la entidad regular de que depende, la lista de atributos y la clave parcial.
- **Todas las relaciones.** Para cada una de ellas se debe especificar las dos entidades que intervienen, su cardinalidad y los posibles atributos de la relación.

Teniendo en cuenta estas consideraciones cada solución tiene la estructura siguiente:

```
NOMBRE ENTIDAD           <nombre entidad regular >
CLAVE PRINCIPAL          < atributo/s clave principal>
OTROS ATRIBUTOS          <otros atributos>
..... (** tantos ítems de este tipo como entidades regulares haya **)
```

```
NOMBRE ENTIDAD DEBIL     < nombre entidad débil>
CLAVE PARCIAL            < atributo/s clave parcial>
OTROS ATRIBUTOS          <otros atributos>
..... (** tantos ítems de este tipo como entidades débiles haya **)
```

```
TIPO RELACION            <cardinalidad>
ENTIDADES                 <entidades que intervienen>
ATRIBUTOS RELACIÓN       <atributos de la relación>
..... (** tantos ítems de este tipo como relaciones haya **)
```

Una vez el profesor ha creado un problema con esta estructura y como paso previo a su utilización se debe incorporar al repositorio de contenidos ACME. Según el identificador de tipo de problema, el módulo de validación comprueba que la estructura del fichero sea la correcta y a partir de aquí ya se guarda definitivamente en el repositorio. En caso que el módulo de verificación detecte que no se ajusta a la estructura esperada, el sistema dará el mensaje de error correspondiente y no se guardará en el repositorio.

Una vez guardado el problema en el repositorio ya estará disponible para ser asignado a los cuadernos de los alumnos. Siguiendo las especificaciones de la plataforma ACME, el profesor ya podrá seleccionar los ejercicios y el módulo de generación de cuadernos los asignará a los alumnos. Una vez asignado el problema, el alumno visualizará el enunciado general para este tipo de problemas, que es fijo:

*Utilizando el modelo Entidad-Relación, proponer un diagrama Entidad/Relación al supuesto desarrollado a continuación. Se deben definir las diferentes entidades y las relaciones entre ellas. Las entidades han de tener todos los atributos que en el enunciado figuran entre paréntesis. Para las relaciones se debe especificar su cardinalidad. Finalmente, se debe introducir el diagrama entidad/relación obtenido, siguiendo las especificaciones de ACME (ver manual de usuario)*

y a continuación el enunciado específico del supuesto guardado en el repositorio.

Una vez el alumno ha resuelto el problema ya podrá introducir el diagrama obtenido en el área de trabajo diseñada. El corrector y tal como se detalla más adelante, utiliza el resto de información del problema para su verificación y puntuación.

#### 4.5.4 Diseño de la interfaz ERD

Otro apartado básico en el desarrollo de este módulo es el referente al diseño de la interfaz para la visualización del enunciado y la posterior entrada del diagrama ER por parte del alumno. En la Figura 4.4 se muestran los distintos apartados de la interfaz. Como se puede apreciar, la ventana está dispuesta en varias zonas:

1. **Zona de visualización del enunciado.** El alumno visualiza el enunciado. Tal como se ha comentado el nombre de los atributos figura entre paréntesis.
2. **Zona de botones con los objetos a dibujar.** Formado por tres elementos: entidades, entidades débiles y relaciones.
3. **Zona de propiedades.** Sirve para especificar las propiedades de los objetos seleccionados Si el elemento seleccionado es una entidad, el alumno deberá asociarle los atributos y determinar cual (o cuáles) forman la clave principal. Si el elemento seleccionado es una entidad débil, el alumno deberá indicar además de los atributos, la entidad fuerte de que depende y la clave parcial. Si el elemento seleccionado es una relación, el alumno deberá indicar su cardinalidad y los posibles atributos de la relación.
4. **Herramienta zoom.** Muy útil en casos de diagramas ER muy extensos.
5. **Área de trabajo del diagrama.** Área donde se dibujará el diagrama. El objeto seleccionado en el área de trabajo debe quedar perfectamente identificado y al seleccionarlo se deben visualizar en la zona 3 sus propiedades que deben poderse modificar y eliminar. En el momento de dibujar el diagrama, todos los objetos se podrán desplazar dentro del área de trabajo. Al mover una entidad se deben también desplazar las relaciones asociadas. Las relaciones deben ir ligadas a las entidades y su desplazamiento en el área de trabajo simplemente separa o acerca el dibujo de la relación a las entidades asociadas. Se deberá evitar, siempre que sea posible, el cruce de las líneas que representan las relaciones.
6. **Botón para enviar la solución a corregir.**

Como puede verse en la Figura 4.4 la interfaz de dibujo es muy intuitiva y simplemente es necesario ir seleccionando los objetos a dibujar en la zona 2 e ir asignando sus propiedades a través de los desplegables de la zona 3. Para establecer las relaciones se escogen dos entidades y se pulsa el botón “relación”. Por defecto se asigna una relación 1:n modificable a través de las propiedades que aparecen en

la zona 3. Una vez completado el diagrama ya se puede enviar a corregir a través del botón correspondiente.

Una vez enviada la solución, la interfaz de recepción de mensajes de ACME es muy simple, simplemente se visualiza la solución enviada y una zona para indicar si la solución es correcta o indicar los mensajes de error si hubieran (véase Figura 4.5).

Esta interfaz se ha implementado con un Applet de Java. Para la correcta utilización de esta herramienta de diseño y dibujo el alumno dispone de un manual de usuario.

Estàs a » ACME » Inici » Bases de Dades » PROBLEMES ENTITAT RELACIO » Problem: 10

---

**EXERCICI: 10 de l'activitat PROBLEMES ENTITAT RELACIO** ← →

Utilitzant el model Entitat-Relació proposeu un diagrama E/R al supòsit desenvolupat a continuació. Cal que definiu les diferents entitats i les relacions entre elles. Les entitats han de tenir COM A MÍNIM TOTS els atributs que en l'enunciat estan entre parèntesi. Per les relacions especifiqueu el seu tipus (1:1, 1:n, n:m) i si són parcials o totals. Finalment cal que des d'aquesta mateixa plana introduïu el diagrama entitat/relació obtingut, seguint les especificacions de l'ACME.

Una empresa està organitzada en departaments. De cada departament nos interessa saber su número (numero\_dep), el nombre (nombre\_dep) y el teléfono (tel\_dep).

Cada departamento tiene asignados empleados de los que nos interesa saber su nombre completo (nombre, apellido1, apellido2), NIF (nif), dirección completa (direccion, codigo\_postal, ciudad), sueldo(sueldo), sexo(sexo) y fecha nacimiento(fecha\_nac). El trabajo de cada empleado está supervisado por otro empleado (nif\_super). Uno de los empleados del departamento desempeña la función de director del departamento.

Cada departamento controla un cierto número de proyectos. De cada proyecto nos interesa saber su código (codigo\_proyecto), su nombre (nombre\_proyecto) y la ciudad (ciudad\_proyecto) donde se realiza. Cada proyecto es controlado por un único departamento.

Un empleado puede trabajar en varios proyectos. Nos interesa saber el número de horas (horas) que cada empleado dedica a cada proyecto.

Un empleado puede tener varias personas dependientes que figuran en su cartilla de la seguridad social. De cada una de estas personas nos interesa saber su número dentro de la cartilla (numero), su parentesco (parentesco) con el empleado, su nombre completo ( nombre\_depe, apellido1\_depe, apellido2\_depe) y su fecha de nacimiento (fecha\_nac\_depe).

El sistema diseñado nos tiene que permitir:

- A partir de un departamento saber todos los empleados que tiene asignados, saber quien es su director y los proyectos que controla.
- A partir de un empleado saber sus datos personales, los datos de su supervisor, en que proyectos ha trabajado y cuantas horas ha dedicado a cada proyecto.
- También nos interesa saber las personas dependientes que tiene a su cargo.
- A partir de un proyecto saber los datos de éste, que departamento lo controla, los empleados que están trabajando en él y las horas dedicadas.

ZOOM 4

+

-

ENTITAT 2

RELACIÓ

ENTITAT FEBLE

5

Opcions de Relació

Nom:  OK

Cardinalitat A:B:  Eliminar

Atributs de la Relació: 3

Nou Atribut

Modificar Atribut

Eliminar Atribut

HORES

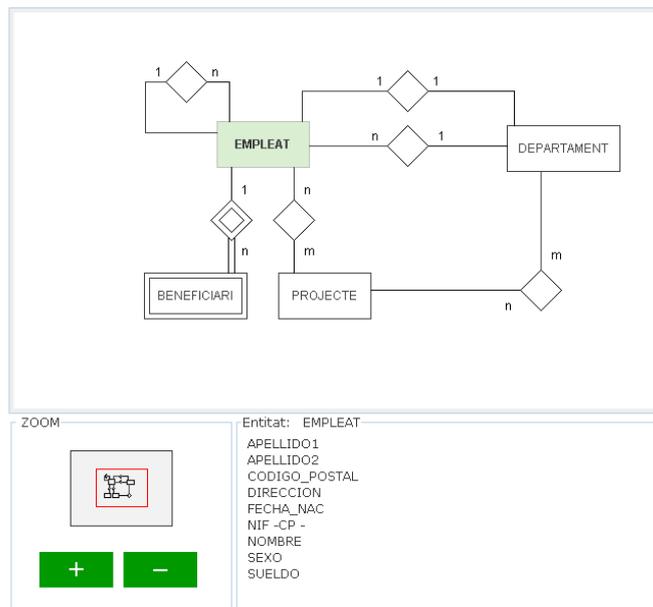
Atributs d'Entrada

←

Corregir 6

Figura 4.4 Enunciado e interfaz de entrada de soluciones a un problema ER.

La solució enviada és:



El resultat de la correcció és:

L'entitat EMPLEAT és incorrecta, repassa-la.  
Hi han RELACIONS incorrectes, repassa-les.

**Continuar**

Figura 4.5 Mensajes de error ACME-ER.

### 4.5.5 El módulo corrector ACME-ER

El tercer punto a considerar es el diseño e implementación del módulo corrector. Cuando un alumno envía una solución a corregir, ésta se guarda con la misma estructura que las soluciones detalladas en la sección 4.5.3, junto con la posición de las entidades y relaciones dentro del área de trabajo. De esta forma, en posteriores consultas se puede reconstruir (ya sea por parte del alumno o del profesor) exactamente el mismo diagrama enviado. Esto también permite enviar una nueva solución a partir de la anterior, de forma que el diagrama visualizado será exactamente el mismo, con las entidades y relaciones situadas en la misma posición y solamente tendrá que realizar las modificaciones oportunas.

La estrategia de corrección se basa en utilizar los nombres de atributos que figuran entre paréntesis en el enunciado del problema. Como el alumno tiene que utilizar estos nombres, identificamos las entidades de la solución del alumno según el campo clave utilizado. En caso de que el alumno asigne un campo no clave como campo clave de una entidad, ésta será considerada incorrecta.

El objetivo de este módulo es comprobar si hay alguna solución correcta asociada a este problema que sea igual a la enviada por el alumno. Si es así se envía el mensaje de solución correcta al alumno. Si la solución del alumno no es correcta hay que determinar cuál de las soluciones correctas almacenadas en el repositorio es la más parecida a la enviada. Tomando de referencia la solución más parecida se envían mensajes al alumno que le ayudarán a detectar sus errores. Así pues este módulo está formado por dos apartados: verificación de la solución y retorno de feedback.

#### Verificación de la solución.

Este apartado tiene como objetivo detectar si la solución es correcta y en caso contrario retornar la solución correcta más parecida a la enviada por el alumno. A partir del resultado de este apartado se retornará el feedback al alumno.

Tal como se muestra en la Figura 4.6, se van accediendo a las diferentes soluciones correctas descritas en el fichero del problema y se van comparando con la enviada por el alumno. En caso de encontrar una de

igual se envía el mensaje de “correcto” y finaliza el proceso de corrección. En caso contrario, se guarda el grado de diferencia entre la solución enviada por el alumno y cada una de las correctas. Para la determinación de las diferencias se dispone de tres procesos de verificación que van comprobando las diferencias entre las entidades regulares, las débiles y las relaciones. Al final del proceso se escoge la que el grado de diferencia es menor como la solución correcta más parecida a la enviada por el alumno y respecto de la cual se enviarán los mensajes de error al alumno.

En este proceso las entidades de la solución del alumno se van identificando según el campo clave utilizado. Una vez identificada una entidad se comprueba que el resto de atributos sean correctos. Las relaciones se van identificando a partir de las entidades que intervienen en la relación y se comprueba que su cardinalidad y atributos sean correctos.

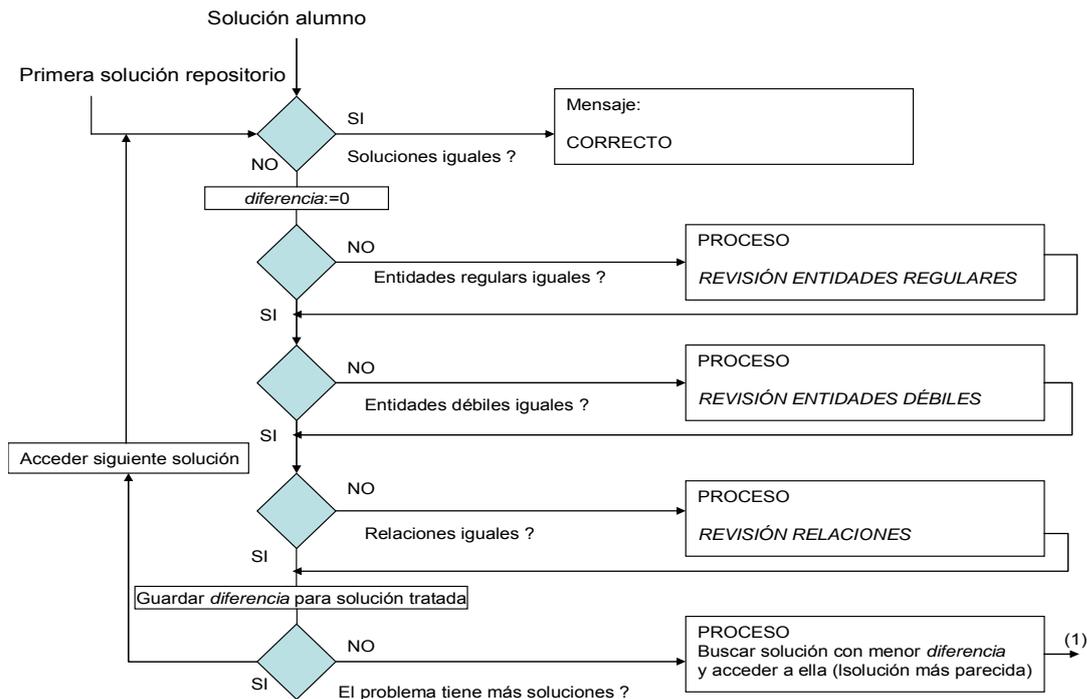


Figura 4.6 Proceso de verificación de la solución en ACME-ER.

Los procesos de revisión de entidades y de relaciones utilizan la estrategia siguiente:

**Proceso de revisión entidades regulares.** Se accede a él ya sea porque el número de entidades regulares sea distinto al del alumno, porque el campo clave no lo sea o porque los atributos esperados sean distintos. En función de las diferencias encontradas se van sumando a *diferencia* los valores parametrizables siguientes:

- Si no coincide el número de entidades regulares se suma  $V_1$  por cada una de diferencia.
- Si no se puede identificar una entidad ya que el campo clave asignado por el alumno no coincide con ningún campo clave de la solución correcta se suma  $V_2$  para cada entidad no identificada.
- Para cada entidad, si los atributos no coinciden con los esperados para aquella entidad se suma  $V_3$  para cada atributo distinto.

De esta forma cuando más diferentes sean las entidades enviadas por el alumno respecto a la solución correcta comparada, mayor será el valor de *diferencia*.

**Proceso de revisión entidades débiles.** Se accede a él ya sea porque el número de entidades débiles sea distinto al del alumno, porque el campo clave parcial no sea correcto o porque los atributos esperados sean distintos. En función de las diferencias encontradas se van sumando a *diferencia* los valores parametrizables siguientes:

- Si no coincide el número de entidades débiles se suma  $V_4$  por cada una de diferencia.
- Si no se puede identificar una entidad débil ya que el campo clave parcial asignado por el alumno no coincide con el de la solución correcta se suma  $V_5$  para cada entidad no identificada.
- Para cada entidad débil, si los atributos no coinciden con los esperados para aquella entidad débil se suma  $V_6$  para cada atributo distinto.

**Proceso de revisión de relaciones.** Se accede a él ya sea porque el número de relaciones sea distinto en las soluciones comparadas, ya sea porque las entidades participantes o la cardinalidad de alguna de las relaciones sean incorrectas o porque los atributos de alguna relación sean incorrectos. En función de las diferencias encontradas se van sumando a *diferencia* los valores parametrizables siguientes:

- Si no coincide el número de relaciones se suma  $V_7$  por cada una de diferencia.
- Si las entidades participantes en la relación no coinciden se suma  $V_8$  para cada relación incorrecta.
- Si la cardinalidad de alguna relación no coincide se suma  $V_9$  para cada relación incorrecta.
- Para cada relación, si los atributos de relación de alguna relación no coinciden con los esperados se suma  $V_{10}$ .

Los valores asignados a los distintos  $V_i$  se han fijado después de varias pruebas y ajustes. Son valores diferentes en función de la relevancia de cada uno de ellos, por ejemplo tiene más importancia la falta de una entidad que la falta de un atributo o la falta de una relación que la de una cardinalidad errónea. De esta forma cuando más diferente sea la solución enviada por el alumno respecto a la solución correcta comparada, mayor será el valor de *diferencia*. Seleccionamos la solución con valor de *diferencia* menor, entendiendo que es la solución correcta más parecida a la del alumno. Esta solución será la que utilizaremos en el apartado siguiente.

## Retorno de feedback.

Una vez detectada la solución correcta más parecida a la enviada por el alumno, tenemos que enviarle los mensajes de error con respecto a ésta que le conduzcan a detectar los errores y le faciliten enviar una nueva solución revisada. Para una buena evaluación formativa consideramos que es necesario facilitar mensajes de error que no deben ser muy detallados, ya que no se pretende dar la solución correcta ni decirle exactamente los errores cometidos, sino facilitar mensajes que le sirvan de orientación. En la Figura 4.7 se muestra la estrategia seguida y el mensaje de error que se visualiza. En la Tabla 4.1 se muestra la causa y el mensaje de error facilitado:

CAUSA		MENSAJE ERROR
1	Si el número total entidades no coincide	A la solución enviada le faltan entidades o A la solución enviada le sobran entidades
2	Si número entidades regulares o débiles no coincide	Tipo de entidades erróneas
3	Si atributos no coinciden	La entidad <nombre entidad> es incorrecta
4	Si número total de relaciones no coincide	A la solución enviada le faltan relaciones o a la solución enviada le sobran relaciones
5	Si las entidades, la cardinalidad o los atributos de la relación no coinciden	Hay relaciones incorrectas

Tabla 4.1 Causas y mensajes de error del módulo ACME-ER.

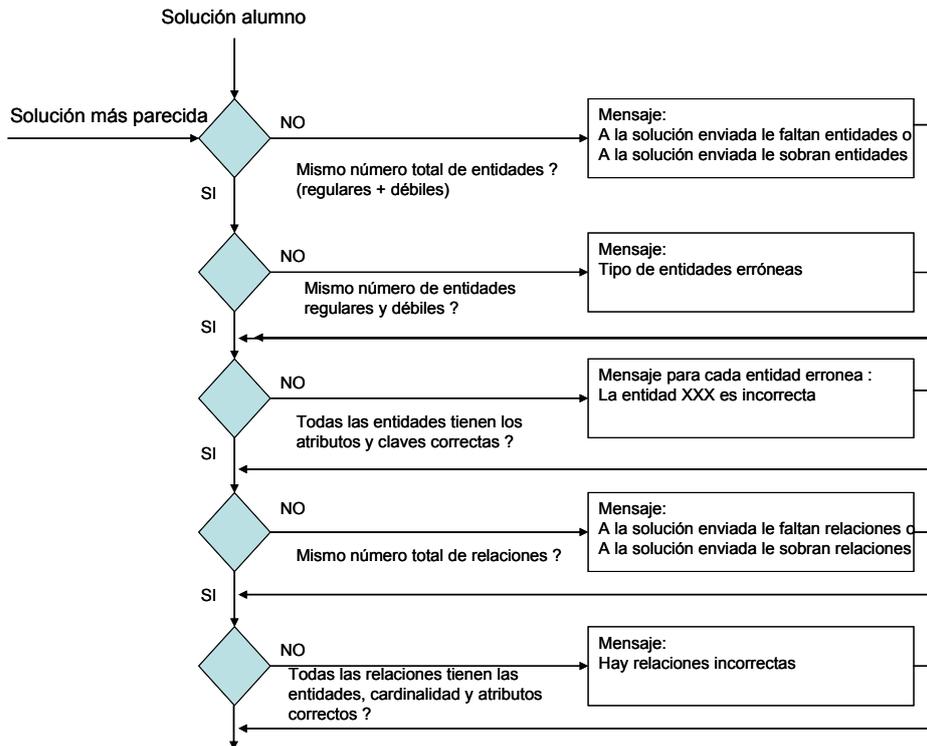


Figura 4.7 Proceso de retorno de feedback en ACME-ER.

#### 4.5.6 Puntuación automática de los diagramas ERD

Tal como se ha comentado en la sección 2.2.7, un tema importante dentro de los entornos CBA es el referente al sistema de puntuación. En muchos casos, y especialmente para la realización de pruebas de validación/exámenes, nos interesa que el sistema, además del correcto/incorrecto, nos facilite una nota sobre el ejercicio realizado por el alumno. Es habitual que en ERD's muy grandes el alumno realice pequeños fallos y no sería justa una mala nota.

Tal como hemos visto se han desarrollado distintos sistemas que permiten la corrección automática de diagramas. Todos estos diagramas se caracterizan por estar formados por diferentes partes, algunas de ellas relacionadas directamente. De esta forma un ERD está formado por entidades y relaciones. Cada entidad vendrá determinada por un conjunto de atributos, donde uno o varios formarán la clave principal que identificará la entidad. Las relaciones vendrán identificadas por su cardinalidad, las dos entidades relacionadas y posibles atributos de la relación.

La puntuación la obtendremos de forma similar a [WTS04]. En este trabajo se calcula a partir de la comparación de unidades mínimas significativas (MMU) con la solución ideal. En nuestro caso el sistema que proponemos se basa en asignar unos pesos a cada grupo de elementos del diagrama de forma que con la suma total de ellos se obtenga la puntuación final. Consideramos que la nota final estará entre 0 y 10. Así pues si consideramos  $n$  grupos de elementos, la suma de los pesos  $P_i$  asignados a cada grupo será 10.

$$\sum_{i=1}^n P_i = 10$$

Como que cada grupo  $i$  está formado por  $m_i$  elementos y considerando que a cada uno de ellos le queremos dar el mismo valor, cada elemento de este grupo tendrá un valor  $e_i = P_i / m_i$ . Por ejemplo, si se asigna un peso 5 puntos al grupo de las entidades y la solución correcta tienen 10 entidades, cada entidad valdrá 0,5 puntos. Por otra parte, consideramos que un error en un grupo de elementos no es tan grave que en otro y por este motivo se debe poder aplicar distintos factores penalizadores  $cp_i$  para cada error en función del grupo. De esta forma el sistema propuesto asigna la nota de la forma:

$$Nota = 10 - \sum_{i=1}^n (e_i * ne_i * cp_i) \text{ y } Nota \geq 0$$

Donde  $e_i$  representa el valor de cada elemento del grupo  $i$  y  $ne_i$  representa el número de errores cometidos en el grupo  $i$  respecto a la solución correcta más parecida a la enviada por el alumno (véase sección 4.5.5) y que el sistema ha detectado en el proceso de corrección automática.

Con la finalidad de ajustar el método es necesario que las notas obtenidas de forma automática se ajusten al máximo posible a las obtenidas manualmente por el profesor. En el sistema planteado el profesor es quien decide los valores  $P_i$  (puntuación de cada grupo de elementos) y es necesario calcular los valores  $Cp_i$  (coeficiente penalizador para los errores del grupo) que minimicen la diferencia de notas obtenidas de forma manual frente a las obtenidas de forma automática. La metodología seguida ha sido la siguiente:

1. A través de la plataforma los alumnos envían sus diagramas como respuesta a los ejercicios asignados.
2. De estos diagramas enviados se ha seleccionado una muestra significativa que ha sido corregida y puntuada de forma manual por tres profesores expertos en la materia y desconocedores de nuestra propuesta de puntuación automática.
3. Hemos considerado como puntuación manual final la nota media de las puntuaciones facilitadas por los tres profesores. A partir de esta puntuación manual final, de los valores de  $P_i$  establecidos por el profesor y del número de errores de cada grupo detectado por el sistema, se han realizado los cálculos oportunos para determinar los valores  $cp_i$  que consiguen que la nota automática se acerque más a la puntuación manual final.

Para aplicar esta metodología se han seleccionado tres problemas de entre 5 y 8 entidades y entre 6 y 8 relaciones. De cada problema se han seleccionado 13 ERD significativos. Los 39 problemas han sido corregidos de forma manual por tres profesores y con la nota media obtenida, con los valores  $P_i$  y el número de errores cometidos se ha realizado los cálculos de  $cp_i$

En primera instancia se pensó en tres grupos de elementos: entidades, relaciones y atributos con pesos de 4 puntos para entidades y relaciones y 2 puntos para atributos. Cada entidad, relación o atributo erróneo sería penalizado con un error. A partir de aquí se calcularon los valores  $cp_i$  que debían minimizar la diferencia con las correcciones manuales. Con este sistema los resultados que se obtuvieron no fueron muy buenos, con diferencias de más de un punto entre las calculadas manualmente y las calculadas de forma automática.

Entonces se pensó que el sistema debía calcular el número de errores cometidos de forma más parecida a la forma de actuar de un experto. El criterio establecido ha sido el siguiente:

- Se han establecido dos grupos, uno para las entidades y el otro para las relaciones. Los valores establecidos para las  $P_i$  de cada grupo son cinco puntos (cinco puntos para las entidades y cinco puntos para las relaciones).
- La forma de contabilizar errores por parte del sistema ha sido la siguiente
  - Errores en las entidades
    - Un error por cada entidad incorrecta. Por entidad incorrecta entendemos entidades que sobran o que faltan y entidades con atributo identificador incorrecto.
    - 0,25 errores por cada entidad identificada correctamente pero en la que hay algún atributo, distinto al identificador, incorrecto.
  - Errores en el grupo relaciones
    - Un error por cada relación incorrecta. Por relación incorrecta entendemos relaciones que sobran, relaciones que faltan y relaciones entre entidades distintas.
    - 0,5 errores por cada relación bien identificada pero con los atributos de la relación incorrectos (en caso de haberlos).
    - 0,75 errores por cada relación bien identificada pero con la cardinalidad incorrecta

Un vez establecidos los valores  $P_i$  y con el número de errores detectados por el sistema, se ha calculado los valores  $cp_i$  que minimicen las diferencias entre las notas manuales y las calculadas.

Para el cálculo de los  $cp_i$  se ha utilizado un modelo de regresión. Para ello hemos partido de que el cálculo de la nota se realiza a partir de  $\text{Nota} = 10 - cp_e * (e_e * ne_e) - cp_r * (e_r * ne_r)$  donde conocemos los valores de  $e_e$  (valor de una entidad), de  $ne_e$  (número de entidades erróneas), de  $e_r$  (valor de una relación) y de  $ne_r$  (número de relaciones erróneas). A partir de ahí se han calculado los valores de  $cp_e$  y de  $cp_r$  de forma que se ajusten a  $\text{Nota} - 10 = -cp_e * x_1 - cp_r * x_2$ . Los valores obtenidos para  $cp_e$  y  $cp_r$  son respectivamente 1,45 y 1,62. Estos valores son significativos ya que el p-valor es 0,00. El coeficiente de determinación  $R^2$  es de 0,993 con lo cual el modelo ofrece una calidad muy buena.

En la Figura 4.8 se puede ver en primer lugar (gráfica Normal Probability Plot) que el modelo obtenido se ajusta a la normalidad. Si observamos los valores de los residuos en las otras tres gráficas vemos que en ningún caso hay una diferencia superior/inferior a los 0,5 puntos respecto a las notas calculadas manualmente.

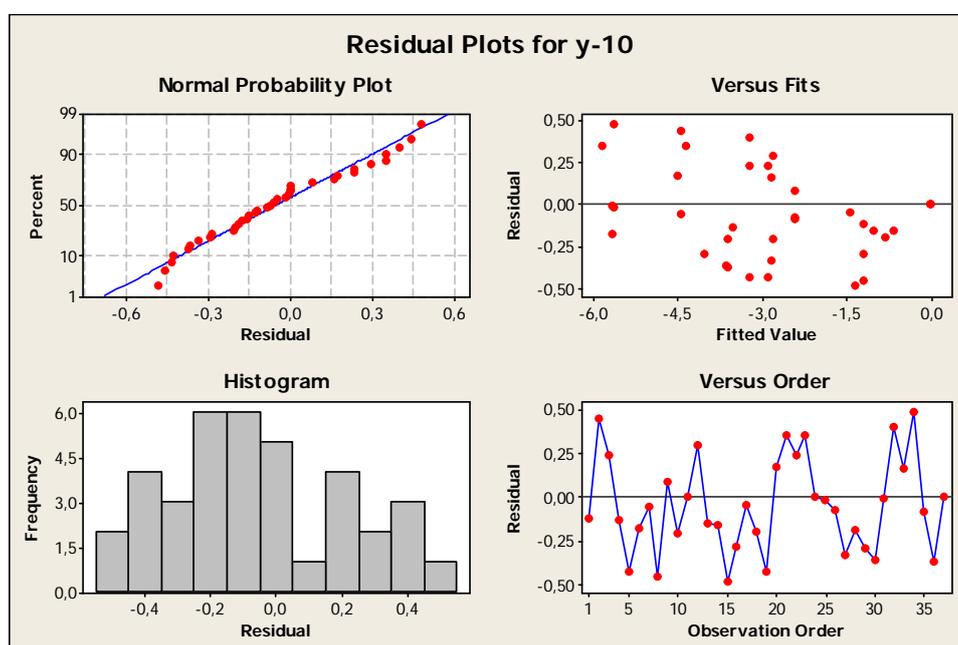


Figura 4.8 Gráficas del modelo de puntuación obtenido.

Comentar que para estos cálculos se han omitido los casos con una nota media inferior a 3. Hemos observado que aunque en un problema haya gran cantidad de errores la nota que asigna el profesor suele estar entre 2 y 3 independientemente de si le correspondería un cero. En estos casos, las notas calculadas según el modelo eran siempre inferiores a la media de la corrección manual. Por otra parte, no consideramos importante esta omisión ya que para notas inferiores a 3, el sistema responderá simplemente “prueba no superada”. Por otra parte vemos que no considerando estos valores el modelo se ajusta muy bien al resto de casos que son los que realmente nos interesa evaluar.

## 4.6 EVALUACIÓN DEL MÓDULO

En este apartado se comenta la preparación de los ejercicios, la tarea previa a su utilización y a continuación se detallan las actividades realizadas y los resultados obtenidos.

### 4.6.1 Preparación de ejercicios ER

Antes de poder utilizar el módulo y poder asignar problemas a los alumnos era necesario el desarrollo de una amplia colección de problemas de forma que, en la medida de lo posible, fueran distintos los ejercicios de cada alumno o por lo menos que no fueran exactamente iguales. Para ello se ha creado una colección formada por tres grupos de problemas:

- **Grupo 1.** 50 problemas de inicio, de poca dificultad. Son problemas con soluciones entre 3 y 4 entidades y entre 3 y 4 relaciones.
- **Grupo 2.** 50 problemas de nivel medio con un poco más de dificultad. Su solución tiene entre 5 y 8 entidades y otras tantas relaciones.
- **Grupo 3.** 50 problemas ya más complicados en los que su solución tiene entre 9 y 15 entidades y las consiguientes relaciones.

De cada problema se han considerado las posibles soluciones correctas y se ha intentado que los enunciados sean claros, concisos y que no conduzcan a confusiones innecesarias. Con ello se ha conseguido que las soluciones estuvieran muy acotadas. El primer grupo de problemas suele tener muy pocas soluciones posibles, mientras que los del tercer grupo suelen tener varias soluciones.

Una vez escrito cada problema el sistema ha verificado que su sintaxis fuese la establecida para este tipo de problema. Una vez la sintaxis es la correcta se ha testeado el problema para que no hubiera ningún error en la/s solución/es. Como se puede apreciar el tiempo dedicado a la creación y testeo de estos problemas es considerable, tal como se ha comentado en la sección referente a los inconvenientes de los sistemas CBA, pero remarcar que una vez verificado el sistema y los ejercicios preparados, se dispone de un entorno y de una colección de problema para ser utilizado siempre que queramos sin ningún esfuerzo adicional.

### 4.6.2 Actividades realizadas y resultados obtenidos

Este entorno corrector de diagramas entidad relación se ha utilizado principalmente como complemento a las clases presenciales (Blended Learning) durante los cinco últimos cursos de la asignatura de bases de datos de la carrera de Ingeniería Técnica en Informática de Gestión (ITIG). Una vez se ha explicado el modelo ER en las sesiones teóricas se asignan varios problemas a través de la plataforma ACME con el objetivo de evaluar el nivel cognitivo de aplicación y de análisis de la taxonomía de Bloom. El profesor elige el nivel de dificultad de los problemas seleccionándolos del repositorio y también establece el número de problemas y la valoración de cada uno de ellos de caras a la evaluación sumativa. Mencionar que el grado de dificultad de los ejercicios va creciendo, siendo los últimos de mayor dificultad. A cada alumno se le asignaran ejercicios de dificultad fácil, media y difícil. Para evitar el plagio, el profesor selecciona todos los problemas de un determinado tipo de dificultad y el sistema, de forma aleatoria los distribuye a los cuadernos de los alumnos. De esta forma, aunque haya alguna repetición, la mayoría de ejercicios de un alumno son distintos de los de sus compañeros, aunque el nivel de dificultad es muy parecido. La realización de estos ejercicios se tiene en cuenta para la valoración final de la asignatura. Para que los alumnos trabajen de forma precisa en los diagramas se les insiste que tendrán penalizaciones en función del número de intentos hasta llegar a la solución correcta. Los alumnos disponen de un manual de usuario de ACME-ER, aunque debido a la sencillez de uso, raramente es consultado. Se insiste a los alumnos en la importancia del trabajo continuado.

Los datos de utilización de este entorno están reflejados en la Tabla 4.2. Para cada curso se muestra el número de alumnos, los problemas asignados, el % de ejercicios que han sido leídos, el % de ejercicios a los cuales se han enviado soluciones y el % que han sido resueltos correctamente. Finalmente en las cuatro últimas columnas se especifica el % de intentos con los que se ha alcanzado la solución correcta (1, 2, 3 o más intentos).

CURSO	ALUMNOS	PROBLEMAS	% EJERCICIOS LEIDOS	% SOLUCIONES ENVIADAS	% SOLUCIONES CORRECTAS	% INTENTOS PARA LLEGAR A LA SOLUCIÓN CORRECTA			
						1	2	3	>3
2005/06	61	6	92,5%	90,5%	81,5%	45,3%	26,4%	16,0%	12,3%
2006/07	50	8	94,0%	90,0%	76,4%	39,8%	25,1%	15,7%	19,4%
2007/08	46	8	96,2%	94,0%	81,0%	57,4%	23,8%	9,8%	9,0%
2008/09	36	8	91,0%	90,5%	80,0%	42,3%	36,9%	10,1%	10,7%
2009/10	32	6	91,6%	88,0%	82,2%	39,8%	23,8%	12,0%	24,4%

Tabla 4.2 Utilización del entorno ACME-ER.

Como se puede apreciar en esta tabla más de un 90% leen los enunciados y más de un 85% envían soluciones. Alrededor de un 80% consiguen llegar a una solución correcta y en más del 75% de los casos la obtienen en los tres primeros intentos. Por otra parte si descartamos los ejercicios asignados que ni tan siquiera han sido visualizados por el alumno (alrededor de un 10% y que hacen referencia a alumnos matriculados pero que no asisten a clase o abandonan la asignatura) y sólo nos fijamos en las soluciones enviadas por los alumnos realmente interesados (Tabla 4.3) vemos que prácticamente siempre envían las soluciones y entre un 80 y un 90% de los casos obtienen la solución correcta.

TEMA	CURSO	ALUMNOS	PROBLEMAS	% EJERCICIOS LEIDOS	% SOLUCIONES ENVIADAS	% SOLUCIONES ENVIADAS/LEIDAS	% SOLUCIONES CORRECTAS	% SOLUCIONES CORRECTAS/LEIDAS
ENTIDAD RELACIÓN	2005/06	61	6	92,5%	90,5%	97,8%	81,5%	88,1%
	2006/07	50	8	94,0%	90,0%	95,7%	76,4%	81,3%
	2007/08	46	8	96,2%	94,0%	97,7%	81,0%	84,2%
	2008/09	36	8	91,0%	90,5%	99,5%	80,0%	88,0%
	2009/10	32	6	91,6%	88,0%	96,1%	82,2%	89,7%

Tabla 4.3 Valoración sobre los ejercicios ACME-ER leídos.

Por otro lado el porcentaje de soluciones correctas varía en función del nivel de dificultad del problema, variando desde casi un 100% en los problemas más fáciles a una horquilla entre el 65% y un 85% en los más difíciles. En los primeros problemas, los más fáciles, se llega a la solución correcta en uno o dos intentos, mientras que en los más difíciles el número de intentos crece considerablemente. Como puede apreciarse el grado de participación suele ser muy alto. Este hecho, por si solo, ya mejora el anterior sistema, en que facilitábamos una lista escrita de problemas y eran muy pocos los alumnos que los resolvían.

En el capítulo 10 se realiza una valoración más exhaustiva de las mejoras en el aprendizaje del diseño de bases de datos que comporta la utilización de ACME-ER, juntamente con el entorno ACME-EBD para el desarrollo de esquemas de bases de datos y ACME-NOR para la normalización de bases de datos.

Para la evaluación formativa del tema esta actividad es francamente positiva ya que desde la plataforma y de manera muy fácil se consigue:

- **Por parte del profesor.** Disponer de herramientas para el seguimiento de la actividad el grupo (Figura 4.9) que le permiten fácilmente determinar qué alumnos están resolviendo correctamente los problemas y los que tienen dificultades. Accediendo al número de ejercicio el profesor visualiza las soluciones enviadas por el alumno. Fácilmente se detectan los errores más frecuentes que realizan que son comentados en la próxima sesión de clase. En todo momento el profesor tiene una visión global del aprendizaje del modelo ER.
- **Por parte del alumno.** Con la resolución de los ejercicios asignados rápidamente se dan cuenta de si sabe aplicar los conocimientos adquiridos. Si un alumno tiene dudas o problemas en la resolución de un ejercicio lo consulta al profesor vía correo electrónico o a través de las tutorías presenciales. El alumno es consciente de que el profesor controla y visualiza su trabajo. Finalmente remarcar que el alumno puede en cualquier momento y desde cualquier sitio enviar las soluciones a la plataforma y de inmediato sabe si son o no correctas. Simplemente necesita de una conexión Internet y un navegador cualquiera.

Esta actividad se tiene en cuenta para la evaluación continua. Para ello damos un valor a cada ejercicio en función de su dificultad. Como que el objetivo final es que el alumno resuelva correctamente el ejercicio planteado y dispone de tiempo para hacerlo, no ponemos ninguna limitación respecto al número de intentos que puede hacer. Aunque el sistema es parametrizable, no ponemos ninguna penalización si se resuelve dentro de los tres primeros intentos. A partir de aquí el profesor establece las penalizaciones que considera oportunas.

Actualmente la forma en que obtenemos la nota final se basa en la evaluación formativa (continua) que vamos realizando a lo largo del curso y en la evaluación sumativa a través de un examen, a modo de prueba de validación. Uno de los problemas del CBA es que no podemos saber quién realmente ha resuelto los ejercicios y pruebas planteados, motivo suficiente para realizar una prueba final. Esta prueba consiste en la resolución de ejercicios de dificultad parecida a la planteada en los ejercicios de ACME-DB y si se obtiene una nota superior a cuatro valida la prueba. Para el próximo curso 2010/2011 con el desarrollo del sistema de puntuación automático descrito en la sección 4.5.6 y a través del entorno para exámenes de la plataforma ACME descrito en la sección 2.5.4 estas pruebas de validación se realizarán en un aula informática. Se convocará a los alumnos en estas aulas equipadas con ordenadores personales y conexión a Internet y deberán ir solucionando la prueba de validación. Al final de la sesión enviarán las respuestas entradas a corregir y automáticamente ACME-DB les facilitará la nota obtenida.

The interface is divided into three main sections on the left sidebar: 'QUADERN D'EXERCICIS' (Exercise Book) with options like 'Gestionar els exercicis', 'Cercar exercicis', and 'Dossier dels alumnes'; 'RESPONSABLE' (Responsible) with options like 'Administrar l'assignatura', 'Administrar els usuaris', 'Gestió de grups d'alumnes', 'Avaluació de l'assignatura', 'Enviar e-mails als usuaris', 'Motlle', and 'Gestió enquestes'; and 'UTILITATS' (Utilities) with options like 'Suggeriments', 'Canviar password', 'Canviar dades', and 'Ajuda'. Below the sidebar is a calendar for February 2010, with the 9th highlighted in red.

The main table, titled 'ACTIVITAT 1 - MODEL ENTITAT/RELACIÓ', tracks student performance across six exercises. The columns are 'Alumne' (Student), 'N. Exercici' (Exercise Number), 'Estat' (Status), 'Errors de Resultat' (Result Errors), and 'Lectures' (Readings). The table lists five students: Albaráñez, César; Alonso Cozar, Adrian; Alonso Muñoz, Kenneth; Aza Puig, Pau; and Baltrons Fabregas, Oriol. Each student's performance is recorded for exercises 1 through 6, with status indicators (Resolved, No Result) and corresponding error counts and readings.

Alumne	N. Exercici	Estat	Errors de Resultat	Lectures
Albaráñez, César	1	Resolt	0	4
	2	Resolt	2	13
	3	Resolt	2	11
	4	Resolt	3	9
	5	No Resolt	5	16
	6	Resolt	3	7
Alonso Cozar, Adrian	1	Resolt	0	1
	2	Resolt	1	2
	3	Resolt	1	4
	4	Resolt	1	4
	5	Resolt	2	10
	6	Resolt	1	2
Alonso Muñoz, Kenneth	1	Resolt	0	1
	2	Resolt	1	2
	3	Resolt	0	1
	4	Resolt	2	4
	5	Resolt	2	4
	6	Resolt	0	2
Aza Puig, Pau	1	Resolt	0	5
	2	Resolt	1	6
	3	Resolt	0	4
	4	Resolt	0	6
	5	Resolt	6	14
	6	Resolt	3	9
Baltrons Fabregas, Oriol	1	No Resolt	0	0
	2	No Resolt	0	0
	3	No Resolt	0	0
	4	No Resolt	0	0
	5	No Resolt	0	0

Figura 4.9 Interfaz de seguimiento del profesor.

Además de este tipo de actividades, en los dos últimos cursos se ha realizado otra actividad para paliar una serie de deficiencias detectadas. A partir de varias reuniones que tuvimos con empresarios del sector informático para el desarrollo del nuevo plan de estudios de grado en Ingeniería Informática, detectamos varias deficiencias en la formación de los actuales egresados de ITIG. Por una parte los empresarios se quejaban de que los alumnos no estaban familiarizados con la estructura, organización y funcionamiento empresarial y que tampoco conocían a fondo la terminología habitual de una empresa. Por otra parte, también se nos hizo notar que los alumnos están acostumbrados a proyectos/prácticas de un tamaño reducido, en contraposición con los proyectos empresariales que a menudo son de gran envergadura. Otro de los puntos débiles radicaba en el trabajo en grupo, comentándonos que a menudo no saben compartir ideas, proponer soluciones al grupo y evaluar entre todos cuál es la mejor solución. Para mejorar estas deficiencias nos propusimos, en una primera fase, desarrollar una práctica extensa, que se desarrollaría en grupo y que intentaría subsanar las deficiencias mencionadas.

Dado que una de las habilidades que el alumno debe adquirir en la carrera es la realización del diseño conceptual a través de diagramas entidad/relación y a continuación obtener el esquema de la base de datos en el modelo relacional, se creyó oportuno realizar esta experiencia en la asignatura de Bases de Datos y se realizaría en grupos de cuatro estudiantes, para fomentar y potenciar el trabajo en grupo. Por otra parte no se quería que esta experiencia supusiera una carga considerable de trabajo al profesor. Para facilitar la

corrección tanto de los diagramas entidad/relación como de los esquemas diseñados utilizaríamos los entornos de ACME-DB desarrollados. Se planificó realizar una práctica extensa en donde el esquema final de la base de datos diseñada tuviera más de 80 tablas. El diseño se realizaría sobre el supuesto de una empresa real y sus distintos departamentos para consolidar los conocimientos en organización y gestión empresarial.

Se preparó un dossier estructurado en diferentes partes en donde se explica la organización de una empresa. En el primer apartado se detalló el funcionamiento general de la empresa y en los cuatro siguientes la información referente a los departamentos de compras, ventas, recursos humanos y producción. Uno de los motivos para realizar cada parte de forma independiente era porque consideramos que para el alumno era mejor abordar cada una de las áreas, de unas veinte entidades, de forma separada y al final ya realizarían la integración de cada una de las vistas. En este dossier se detalla toda la información que deberá guardarse en la base de datos. Además del dossier se preparó cada una de las partes con la información necesaria para que la plataforma pudiera corregirlas de forma totalmente automática.

Para mejorar la evaluación formativa también se ha realizado un análisis de los errores que cometen los alumnos con el objetivo de incidir en las sesiones teóricas en los puntos débiles detectados. Para ello hemos analizado la tipología de errores en función del número de intento. La muestra se ha obtenido a partir de todos los errores cometidos en los ejercicios propuestos de este tema durante el último curso. La información obtenida se ha agrupado en tres bloques tal como muestra la Tabla 4.4. El primer bloque refleja el % de errores debidos exclusivamente a relaciones, ya sea porque sean incorrectas o porque sobren/falten relaciones. Este bloque varía desde el 35% en el primer intento hasta el 65% a partir del cuarto. En contraposición observamos en el tercer bloque que los errores debidos exclusivamente a la identificación de las entidades son sólo del 12% en el primer intento y apenas significativos en los demás intentos. El segundo bloque refleja los errores en soluciones donde aparecen errores tanto en las entidades como en las relaciones. Este tipo de errores va disminuyendo gradualmente desde un 53% en el primer intento hasta un 35% a partir del cuarto intento. En vista de estos datos vemos que en los errores que cometen los alumnos, casi siempre está involucrada una relación (bloque 1 más bloque 2), mientras que los errores en las entidades suelen ser mucho menores (bloque 2 más bloque 3). Estos datos concuerdan con los obtenidos por [BS93] [BA94] que afirman que los alumnos no tienen muchos problemas en la detección de las entidades pero si en las relaciones y su cardinalidad. Este hecho, que ya habíamos detectado en otros cursos pero que no habíamos cuantificado, hace que en la enseñanza del modelo ER nos centremos principalmente en detectar las relaciones entre las diferentes entidades y determinar su cardinalidad.

TIPO DE ERROR	INTENTOS			
	PRIMER	SEGUNDO	TERCER	CUARTO y sig.
<b>BLOQUE 1 - Errores en relaciones</b>				
Relaciones incorrectas	20,2%	41,5%	49,3%	60,7%
Faltan relaciones	11,0%	4,6%	3,8%	2,3%
Demasiadas relaciones	3,7%	1,5%	0,0%	2,3%
<b>TOTAL</b>	<b>34,9%</b>	<b>47,7%</b>	<b>53,1%</b>	<b>65,2%</b>
<b>BLOQUE 2 - Errores en relaciones y entidades</b>				
Relaciones y entidades incorrectas	34,9%	36,9%	31,8%	30,2%
Demasiadas relaciones y demasiadas entidades	2,8%	1,5%	1,9%	2,3%
Demasiadas relaciones y entidades incorrectas	2,8%	0,0%	5,7%	0,0%
Demasiadas entidades y relaciones incorrectas	1,8%	1,5%	0,0%	0,0%
Faltan relaciones y faltan entidades	7,3%	1,5%	0,0%	0,0%
Faltan relaciones y entidades incorrectas	3,7%	3,1%	3,8%	2,3%
<b>TOTAL</b>	<b>53,2%</b>	<b>44,6%</b>	<b>43,2%</b>	<b>34,8%</b>
<b>BLOQUE 3 - Errores en entidades</b>				
Entidades incorrectas	10,1%	7,7%	3,7%	0,0%
Faltan entidades	0,9%	0,0%	0,0%	0,0%
Demasiadas entidades	0,9%	0,0%	0,0%	0,0%
<b>TOTAL</b>	<b>11,9%</b>	<b>7,7%</b>	<b>3,7%</b>	<b>0,0%</b>

Tabla 4.4 Tipología de errores en función del número de intento.

## 4.7 APORTACIONES Y CONCLUSIONES

Se ha propuesto, definido y desarrollado el entorno ACME-ER, un entorno CBA para la corrección/evaluación de diagramas entidad/relación, se ha integrado en la plataforma ACME y se ha creado una extensa colección de problemas. Se ha presentado la experiencia de uso del módulo en la asignatura de “Bases de datos” desde el curso 2005/2006. Las principales ventajas que presenta son:

- Permite llevar a cabo tanto la evaluación formativa como sumativa de diferentes actividades de diseño conceptual de BD a través de ERD. Actúa como un sistema CBA.
- Disminución del trabajo del profesor en cuanto a horas dedicadas a la corrección de este tipo de ejercicios, aunque el esfuerzo de la puesta a punto y diseño de ejercicios es considerable.
- Muchos más alumnos se ejercitan en el diseño conceptual de bases de datos y obtienen un resultado inmediato de la corrección.
- Mayor feedback alumno/profesor. Se aclaran muchas dudas que surgen en el momento de la resolución de los ejercicios.
- Permite un seguimiento muy estricto del trabajo del alumno.
- Evaluación automática de todas las actividades realizadas por el alumno.

La única limitación a considerar es que el alumno no puede asignar nombres a los atributos y debe escoger los marcados en el enunciado del problema.

El trabajo realizado referente al desarrollo de este corrector ha dado lugar al artículo “**A web based-tool for Entity-Relationship Modeling**” publicado en LNCS 3980, Pág. 364-372, referenciado en [PBS+06].

La utilización de parte de ACME-DB como un complemento a las clases presenciales en la docencia de bases de datos ha sido presentada en las XII Jornadas de la Enseñanza Universitaria de la Informática (JENUI) en el 2006 en el artículo “**Utilización de una plataforma de e-learning en la docencia de bases de datos**”, referenciado por [SPB+06b].

En el artículo “**Database design using a web-based e-learning tool**” presentado en el 2nd Workshop on Methods and Cases in Computing Education 2009 (MCCE) se expone la experiencia en la utilización de ACME-DB en el aprendizaje de diseño de bases de datos. Este artículo está referenciado en [SBP+09a].

El detalle de la experiencia llevada a cabo en el diseño de una base de datos extensa a través de la plataforma ACME-DB y los resultados obtenidos están disponibles en la ponencia “**Experiencia docente en diseño de bases de datos con la ayuda de herramientas de e-learning**” ha sido presentada en las XV Jornadas de la Enseñanza Universitaria de la Informática (JENUI) en el 2009, referenciado en [SBP+09b].



## CAPÍTULO 5. MÓDULO CORRECTOR DE DIAGRAMAS DE CLASE

En este capítulo presentaremos el segundo de los módulos correctores desarrollados: el módulo de diagramas de clase. Este módulo dará soporte al aprendizaje y docencia del modelado conceptual de datos utilizando UML.

Este capítulo se ha estructurado de la siguiente forma: en la sección 5.1 se realiza una introducción al capítulo. En la sección 5.2 se describen los diferentes componentes de un diagrama de clases y a continuación en la sección 5.3 describimos las necesidades y la problemática del modelado conceptual de datos mediante diagramas de clase. En la sección 5.4 se realiza un breve resumen de entornos para el aprendizaje de diagramas de clase. Posteriormente en la sección 5.5 planteamos el problema a resolver, describiremos detalladamente la solución aportada y su integración a la plataforma ACME. En la sección 5.6 se describe la evaluación del módulo, describiendo las actividades realizadas y los resultados académicos conseguidos. Finalmente en la sección 5.7 presentamos las conclusiones.

### 5.1 INTRODUCCIÓN

La primera fase para el diseño de una base de datos es la obtención y análisis de requisitos que deberá satisfacer la aplicación a desarrollar. Una vez analizados estos requisitos, la segunda fase es el diseño conceptual de la base de datos. Para ello se han propuesto diferentes metodologías, entre ellas el modelo Entidad/Relación (ER: *Entity-Relationship*) [Che76] y el modelo Entidad /Relación extendido (EER: *Enhanced Entity-Relationship*) [TYF86]. Estas metodologías son las más utilizadas para el modelado de datos.

Debido al auge en el análisis y diseño orientados a objetos, el Lenguaje Unificado de Modelado (UML) [BRJ99] es la notación más usada en el área de desarrollo de software. Dentro de UML, los diagramas de clase se pueden utilizar para modelar el esquema conceptual de la base de datos. Ahora bien, los diagramas de clase no han sido ampliamente adoptados para el diseño conceptual [Rum06] y se sigue utilizando el modelo ER. La tendencia actual es que se vaya pasando de los diagramas ER a los diagramas de clase.

Los diagramas de clase nos describen los distintos objetos de un sistema y las relaciones estáticas que hay entre ellos. Desde un punto de vista docente, tal como se detalla en la sección 5.3, hay partidarios de enseñar el diseño conceptual de BD a partir de los tradicionales diagramas ER y quien prefiere hacerlo a través de diagramas de clase. Lo importante es que, de una forma u otra, el alumno sepa realizar el diseño conceptual de una base de datos.

Los comentarios acerca del diseño conceptual de BD que hemos realizado en la introducción del Capítulo 4, referentes a la metodología docente, al proceso de evaluación, a las actividades de diagramación y a los entornos CBA, son perfectamente válidos para los diagramas de clase en vez de los ERD.

Nuestro objetivo es desarrollar un sistema CBA especializado que nos permita la evaluación automatizada de diagramas de clase. Para ello, el entorno debe permitir definir problemas, su corrección y generación de feedback automáticos en materia de diagramas de clases. El sistema debe soportar la evaluación formativa, de manera que el profesor pueda controlar el trabajo de los alumnos. Además el sistema puede proporcionar una puntuación automática de cada ejercicio para poder realizar la evaluación sumativa. Con las puntuaciones obtenidas el sistema facilitará una evaluación final del alumno. Este entorno para el aprendizaje y evaluación de diagramas de clase debe integrarse en la plataforma ACME, actuando conjuntamente como un sistema CBA y con el que se realizarán las correspondientes pruebas sobre los alumnos de la asignatura de BD.

## 5.2 DESCRIPCIÓN DE UN DIAGRAMA DE CLASES

En este apartado se comentará la descripción e importancia de los diagramas de clase en el modelado conceptual de objetos. Los diagramas de clase nos describen los distintos objetos de un sistema, las relaciones estáticas que hay entre ellos y se utilizan para modelar el esquema conceptual de la base de datos.

El modelado conceptual se basa en un conjunto de conceptos básicos que se representan a través de una notación determinada tal y como mostramos en el ejemplo de la Figura 5.2. Los conceptos básicos [BRJ99] del modelado conceptual usados en los diagramas de clase son:

- **Clase.** Una clase describe un grupo de objetos con propiedades comunes (atributos), comportamiento común (operaciones), relaciones comunes con otros objetos y la misma semántica. Mediante UML una clase se representa por medio de un rectángulo dividido en tres partes que contiene:
  - En la parte superior el nombre de la clase
  - En la parte central los atributos de la clase, con su visibilidad, tipo, posibles valores iniciales y propiedades.
  - En la parte inferior las operaciones de la clase
- **Relación.** Existen tres tipos distintos de relaciones:
  - Asociaciones (conexión entre clases)
  - Dependencias (relación de uso)
  - Especialización/Generalización (relaciones de herencia)
- **Asociación:** Modela una conexión semántica entre clases. Es una relación estructural que especifica que los objetos de una clase están conectados a los objetos de otra clase. Se puede navegar desde un objeto de una clase a uno de la otra. Se representa mediante una línea continua entre las clases. Como subtipos de las asociaciones se definen las agregaciones y composiciones.
  - **Agregación:** Es una forma especial de asociación que modela una relación “todo/parte” o de “continente/contenido” entre un agregado (el todo) y sus partes. Se representa mediante un rombo blanco en la parte del todo.
  - **Composición:** Es una forma de agregación con un propietario fuerte y tiempo de vida coincidentes. Las partes no tienen sentido sin el todo. Las partes sólo pueden crearse después del agregado (el todo) al cual pertenecen, pero una vez creadas viven y mueren con el agregado. Las partes sólo pueden formar parte de un agregado. Se representa mediante un rombo negro en la parte del todo.
- **Multiplicidad:** La multiplicidad define cuántos objetos pueden participar en una relación, es decir el número de instancias de una clase relacionadas con una instancia de la otra clase. Se especifican mediante un rango en cada final de asociación y los valores más habituales que suelen tener son: 1..\*, 1..1 (1), 0..1, 0..\* (\*), n..m, aunque también están permitidas otras expresiones como 4..\*, 3..5, ...
- **Navegación:** Las asociaciones son bidireccionales por defecto pero a veces es deseable restringir la navegación en una sola dirección. En este caso a la línea de la asociación se le añade una flecha que indica el sentido de la navegación.
- **Asociación cualificada:** Denota un atributo o lista de atributos en los cuales el valor de los atributos seleccionan uno o varios objetos relacionados en la clase destino. Entre la clase y la asociación se especifica el atributo calificador de la asociación
- **Clases asociativas:** Si una asociación tiene que tener atributos y/o operaciones propias entonces se tiene que definir como una clase. A este tipo de clases las llamamos clases asociativas o clases asociación. Representan propiedades y operaciones de los objetos implicados en una asociación. Se representa como una clase con una línea que la une a la asociación.
- **Asociaciones n-arias:** Son asociaciones entre tres o más clases.
- **Generalización:** Es una relación entre clases donde la clase especializada comparte la estructura y el comportamiento de la clase general.

Una vez descritos los conceptos básicos de los diagramas de clases y para ver la analogía con un diagrama ER, mostramos a continuación el ejemplo de la sección 4.2 en el que se desarrollaba el ERD al supuesto típico de una empresa. La Figura 5.1 muestra el diagrama ER y la Figura 5.2 el diagrama de clases equivalente, utilizando el modelado conceptual de objetos mediante diagramas de clases UML.

Aunque los enunciados estén dirigidos hacia una única solución, generalmente se pueden aceptar varias soluciones como correctas. En este caso podemos aceptar la solución mostrada en la Figura 5.2 como correcta. Otra posible solución correcta pasaría por considerar la asociación cualificada entre EMPLOYEES y DEPENDENT como una asociación normal en vez de ser una asociación cualificada. Evidentemente se pueden considerar otros diagramas de clase como correctos por considerar que son distintas interpretaciones del enunciado original.

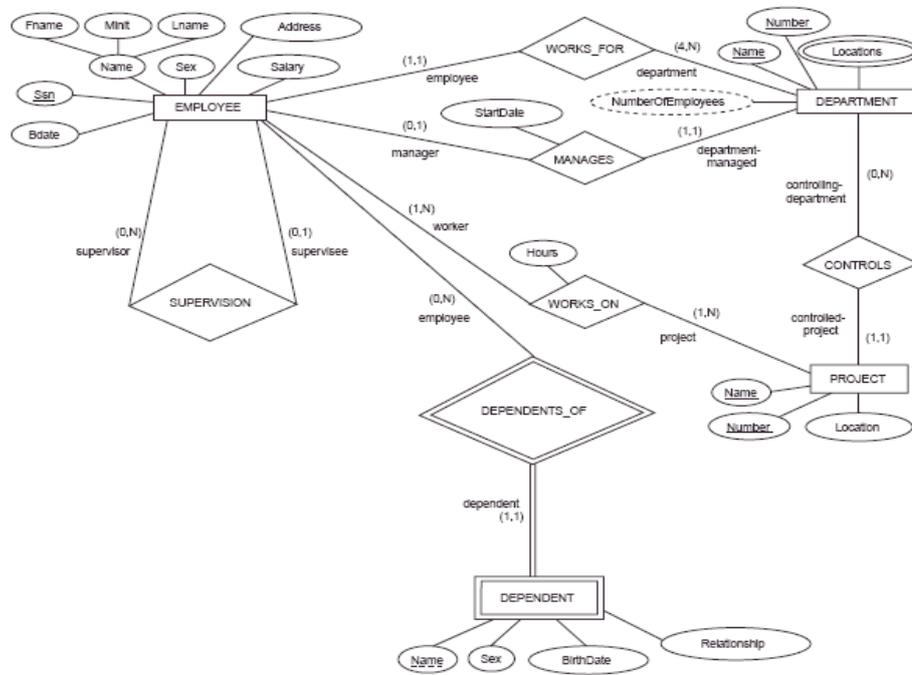


Figura 5.1 Diagrama ER al supuesto típico de una empresa.

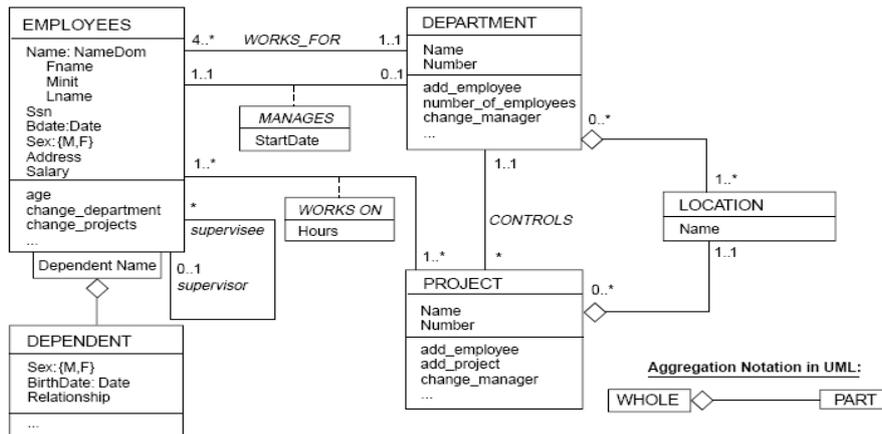


Figura 5.2 Diagrama de clases UML del mismo supuesto anterior.

Al igual que el módulo corrector ER descrito en el Capítulo 4, y siguiendo la metodología descrita en la sección 3.3, para lograr la implementación definitiva del módulo se deberán realizar las siguientes fases:

- Definir la estructura de los ejercicios.
- Desarrollar las interfaces para la entrada de diagramas y recepción de mensajes.
- Desarrollo del módulo corrector y su integración a la plataforma ACME.
- Creación de una colección de problemas para su posterior uso.

### **5.3 NECESIDADES Y PROBLEMÁTICA DE LA ENSEÑANZA DEL MODELADO CONCEPTUAL CON DIAGRAMAS DE CLASE.**

Las metodologías de modelado de objetos, tales como UML (Universal Modeling Language) [BRJ99] y OMT (Object Modeling Technique) [RBP+96], se han popularizado muy rápidamente. Aunque estas metodologías se desarrollaron principalmente para el diseño de software, una parte muy importante de este diseño consiste en el diseño de la base de datos a la que se accederá mediante módulos de software. Por lo tanto una parte importante de estas metodologías, principalmente los diagramas de clases, son similares en muchos aspectos a los diagramas Entidad Relación Extendido (EER) aunque la terminología sea distinta. El concepto de *tipo de entidad* del modelo ER se representa como clases y una *entidad ER* se corresponde a un objeto en UML. Los *tipos de relaciones* se denominan asociaciones en UML y como un caso particular de éstas se definen las agregaciones y las composiciones. Las restricciones de *cardinalidad* y participación pasan a llamarse *multiplicidades*. Los conceptos de *especialización /generalización* (herencia) introducidas en el modelo EER también están contemplados en la notación UML.

El punto de vista predominante es que UML es un entorno para construir aplicaciones que usan lenguajes orientados a objetos, pero no es el modelo más apropiado en el diseño de bases de datos. De todas formas según [Rum06], los autores de UML se inspiraron en los diagramas ER en lo que concierne a los diagramas de clases. Los diagramas ER ofrecen una visión abstracta de los datos, agrupándolos por entidades, describiendo sus atributos y estableciendo las relaciones entre ellas. Los diagramas de clases también ofrecen esta misma visión con las clases y sus atributos y las correspondientes asociaciones, pero con la ventaja que los diagramas de clase especifican los métodos asociados a cada clase. Ahora bien en la fase de modelado de datos, no hay necesidad de especificar los métodos de cada clase y es por este motivo, y también por su sencillez, que el modelo ER siga siendo válido para este menester.

Al imponerse la notación UML como un estándar en el desarrollo de software, la tendencia natural será, cada vez más, la utilización de diagramas de clase para el diseño conceptual de bases de datos para representar la estructura estática del sistema, aportando además los detalles necesarios de las clases persistentes para su posterior almacenamiento en la base de datos [NM01]

Actualmente diversos autores proponen incorporar el modelado conceptual mediante diagramas de clases en los cursos de bases de datos [Neu02]. Otros autores prefieren usar ambas notaciones pero que sean impartidas en cursos diferentes [Man06], enseñando el modelo ER en un curso de bases de datos y los diagramas de clase, junto con el resto de la notación UML, en una asignatura de análisis y diseño de sistemas. En esta línea los autores de [CJC06] consideran que ambas notaciones son necesarias y que en el caso que la asignatura de bases de datos preceda a la de análisis y diseño, es conveniente empezar con el modelo ER en bases de datos y UML en análisis y diseño, separando claramente datos y procesos. También hay quien ha sustituido la enseñanza del modelo ER argumentado que los diagramas de clase sirven para realizar el diseño conceptual tal como los diagramas ER, pero que además aportan información adicional que es usada en otras partes del desarrollo del sistema [GM06]

En un estudio del año 2006 realizado en [SG06] los autores se preguntan si es el momento oportuno para introducir el modelado de datos mediante UML. Para ello analizan el software de diagramación, los libros de referencia y que sistema de modelado de datos utilizan las más prestigiosas universidades americanas en los cursos de bases de datos. En relación con los paquetes de software más utilizados en representación de diagramas (Microsoft Visio, Oracle Designer, ERWin Data Modeler, SmartDraw, etc.) todos ellos soportan los diagramas UML, mientras que otros tipos de notación (Chen, Idefix, Bachman, Information Engineering) sólo son soportados por algunos de ellos. Esta tendencia al uso de UML todavía no se ha visto ampliamente reflejada en las últimas ediciones de los libros de referencia de bases de datos y sólo [EN07] trata con profundidad el tema de modelado de datos utilizando los diagramas de clases. Aunque también es cierta la aparición de bibliografía específica [NM01] y [Mul03] de UML para el diseño de bases de datos. En este estudio sólo 3 de las 19 universidades americanas más prestigiosas utilizaban UML en los cursos de introducción a las bases de datos, mientras que las otras utilizan modelado conceptual mediante diagramas ER. Sin embargo, la mayoría de estas universidades introducían UML cuando explicaban bases de datos orientadas a objetos y evidentemente en otras asignaturas de análisis y diseño.

Otros autores consideran que ni un modelo ni otro son los más adecuados para el modelado de datos [KG06]. Entre otras características consideran que un modelo de datos debe ser robusto, muy simple e independiente del modelo de bases de datos. Analizan los pros y contras de cada uno de ellos y consideran que se necesita un nuevo modelo de datos que sea más moderno que el ER y más práctico que UML. Por otra parte, en el estudio realizado en [WS06] se analiza la demanda de empleo según un conjunto de palabras clave que deben tener los candidatos. Entre estas palabras clave figuran “ERD”, “UML”, “Data Modeling”, etc. y al final los autores afirman que hay una cierta confusión entre estos términos.

Desde un punto de vista pedagógico las necesidades del desarrollo de un buen diseño conceptual, expuestas en la sección 4.3 son totalmente válidas, pero aplicables a los diagramas de clase en vez de a los ERD’s.

Después de analizar toda esta información y aunque haya una cierta indefinición con partidarios de las dos notaciones, en este trabajo hemos considerado la necesidad de desarrollar un corrector de diagramas de clase, en la línea de ofrecer ambos correctores para el aprendizaje del diseño conceptual de bases de datos.

## **5.4 CORRECCIÓN AUTOMÁTICA DE DIAGRAMAS DE CLASE**

Al ser UML el lenguaje de modelado de sistemas de software más conocido y ampliamente utilizado en la actualidad, van surgiendo diferentes entornos para el aprendizaje de la realización de los distintos tipos de diagramas que se utilizan. En el ámbito concreto del modelado conceptual utilizando diagramas de clases se han desarrollado distintos entornos que se han detallado en la sección 2.4.2. El problema, visto desde un punto de vista general, se puede tratar como la interpretación y corrección de un tipo de diagrama. En este tipo de problemas, uno de los principales retos es interpretar la semántica de la solución enviada por el alumno, en nuestro caso identificar clases, atributos, asociaciones y sus características. Otro de los problemas es que se puede aceptar más de una solución correcta.

De los entornos evaluados, la herramienta que ofrece mejores prestaciones es COLLECT-UML aunque no permite la puntuación automática. CIMEL-ITS detecta los errores a medida que se van produciendo. Desde nuestro punto de vista, consideramos que guía en demasía al alumno y le da muy poca independencia. ASSET no es un sistema automático y necesita de un tutor experto que es quien ofrece el feedback. Es un sistema pensado para dar formación a las personas que se integran en una empresa de desarrollo de software. UCDA tiene el inconveniente que se necesita disponer de Rational Rose para la realización de los diagramas. La propuesta de la extensión de ArgoUML es quizás la más innovadora ya que no tiene ninguna restricción sobre los nombres de los distintos elementos que conforman el diagrama. Ahora bien, de momento no permiten ciertas funcionalidades habituales de un diagrama de clase. Uno de los problemas de estas herramientas es que están orientadas al aprendizaje/enseñanza de los diagramas de clase, permitiendo una evaluación formativa, pero no realizan ningún tipo de puntuación automática de los ejercicios realizados por los alumnos.

Tal como se ha comentado en el apartado 2.4.7 otra de las limitaciones de estos entornos (con la salvedad de los trabajos liderados por A.Mitrovic) es que sólo permiten el trabajo en un tema concreto, en este caso los diagramas de clase, pero no abordan problemas de diseño lógico (esquemas de BD y normalización) ni tampoco ejercicios de consultas sobre BD.

Dadas estas limitaciones, nuestro objetivo en este trabajo es desarrollar un entorno integrado en materia de BD, que facilite todas las prestaciones de un CBA, desde la entrega de materiales y tutoriales, hasta la asignación de ejercicios, que una vez resueltos serán corregidos de forma automática, retornando el feedback adecuado y obteniendo la puntuación pertinente.

## 5.5 ACME-DC: MÓDULO DE DIAGRAMAS DE CLASE

En esta sección se detalla el módulo referente a la evaluación de diagramas de clase que denotamos como ACME-DC. En la sección 5.5.1 se determinan los requisitos que deberá satisfacer el sistema. En la sección 5.5.2 se presenta una visión general de la solución desarrollada. En las cuatro siguientes secciones se detallan los aspectos más técnicos del módulo: la estructura de un problema de diagramas de clase, su interfaz, el módulo de corrección y finalmente la puntuación del problema.

### 5.5.1 Requisitos del sistema ACME-DC

El problema que se plantea es desarrollar un sistema que de forma totalmente automática sea capaz de evaluar un diagrama de clases de un supuesto planteado (véase ejemplo de la sección 5.2). Debe tenerse en cuenta que un mismo supuesto puede tener varias soluciones correctas, por lo que el sistema deberá tener en cuenta que la solución puede no ser única. Los requisitos que definiremos para diseñar este módulo corrector son los siguientes:

- El sistema deberá contemplar los elementos que forman un diagrama de clases y que han sido descritos en la sección 5.2. No se contemplará ni la entrada de los métodos de la clase ni su corrección, ya que el modelado conceptual mediante diagramas de clases ofrece una visión estática del sistema y no es necesaria la definición de los métodos que se utilizarán. Concretamente los elementos que se utilizarán son:
  - Clases, clases asociativas y sus atributos. De los atributos sólo se utilizará su nombre.
  - Las asociaciones existentes, entre que clases, su multiplicidad y su navegación. Los valores permitidos para representar la multiplicidad serán 1..\*, 1, 0..1, \*, n..m.
  - Las agregaciones, composiciones y asociaciones calificadas.
  - Las generalizaciones y dependencias.
- El sistema deberá disponer de una interfaz apropiada para el dibujo de diagramas de clase que debe permitir:
  - Dibujar (y modificar/borrar) clases y clases asociativas.
  - Permitir la asignación de atributos.
  - Dibujar (y modificar/borrar) las asociaciones, agregaciones y composiciones, señalando entre que clases están definidas, su multiplicidad y navegación.
  - Dibujar (y modificar/borrar) generalizaciones y dependencias, señalando entre que clases están definidas.
  - Permitir la asignación de un nombre cualquiera a las clases y un nombre de rol a cualquier tipo de asociación.
- El sistema deberá facilitar la evaluación formativa. Cuando el alumno envíe una respuesta, el sistema recogerá la información asociada a los distintos objetos dibujados y los enviará al servidor. Esta información será guardada por ACME como una solución enviada por el alumno a uno de sus ejercicios. El módulo desarrollado comparará la solución enviada con las soluciones correctas asociadas al problema. En caso que sea correcta lo notificará al alumno mediante un “Correcto”. En caso de que el corrector determine que es incorrecta facilitará pequeñas notificaciones (feedback) para ayudar al alumno a encontrar el error y poder enviar una nueva solución.
- El sistema deberá facilitar una puntuación a cada ejercicio. En las actividades que se determinen (pruebas, exámenes, etc.) el sistema debe facilitar la nota obtenida, de forma que nos pueda servir como un elemento para su evaluación sumativa.
- El sistema deberá integrarse en la plataforma ACME, con lo cual se compartirán todas las funcionalidades ACME, como por ejemplo la gestión de cursos, asignaturas, temas, actividades, alumnos, envío de mensajes, foros, chats, etc.

Descritos los requisitos que debe cumplir el módulo de corrección de diagramas de clase, analizamos y desarrollamos el módulo.

## 5.5.2 Análisis inicial de ACME-DC

El problema, visto desde un punto de vista general, se puede tratar como la interpretación y corrección de un determinado tipo de diagrama. En este tipo de ejercicios, al igual que en los ERD (véase sección 4.5.2), uno de los principales problemas es interpretar la semántica de la solución enviada por el alumno, en nuestro caso identificar los diferentes componentes del diagrama de clases. Otro de los problemas es que se puede aceptar más de una solución correcta, con lo que los mensajes de error enviados tendrán que referirse a la solución correcta más parecida a la enviada por el alumno.

Para identificar los componentes del diagrama hemos optado por el mismo método que los diagramas ER. Para ello especificamos el nombre de los diferentes atributos que se colocan entre paréntesis a continuación de su descripción en el enunciado. Facilitando esta información no ayudamos apenas al alumno, en todo caso le especificamos de forma más concisa los atributos que debe guardar el sistema. Por el contrario no ponemos ninguna limitación respecto al nombre de las clases ni al nombre del rol de las asociaciones y el alumno puede asignarles el nombre que desee. A partir de la agrupación de los atributos, identificaremos las distintas clases y a partir de estas a las asociaciones entre ellas.

Otra restricción de nuestro sistema es que no permitimos asociaciones n-arias. Consideramos que el alumno puede descomponer una asociación n-aria en otras binarias. Esta misma limitación figura también en la mayoría de herramientas CASE comerciales que permiten la realización de este tipo de diagramas y también en los entornos evaluados en la sección 2.4.2. Tampoco vamos a corregir los métodos de las diferentes clases ya que en el diseño conceptual de bases de datos, los diagramas de clases son utilizados como diagramas estáticos, en donde nos interesa mostrar las diferentes clases y sus asociaciones y no los métodos de cada clase.

Fijadas estas restricciones ya estamos en condiciones de diseñar el modelo. Este proceso irá ligado a la estructura de la plataforma ACME e implicará la definición de:

- **La estructura del problema.** Cada problema se guarda en un fichero en el repositorio. Debe estar formada por un enunciado y las posibles soluciones correctas. El profesor mediante un editor cualquiera, o el específico de ACME, editará la estructura del problema. A continuación, y con las funcionalidades específicas de ACME, se validará que la estructura del problema sea correcta y en este caso se incorporará al repositorio. A partir de aquí ya estará disponible para ser utilizado.
- **Una interfaz apropiada** a través de la cual el alumno visualizará el enunciado del problema, dibujará los diagramas de clase correspondientes y finalmente recibirá el feedback oportuno.
- **Un módulo corrector** que interpretará la solución enviada y la comparará con las soluciones correctas descritas en la estructura del problema y que están guardadas en el repositorio. Este módulo corrector determina el feedback a retornar al alumno. El propio sistema ACME se ocupa de guardar todas las soluciones enviadas. Con toda esta información, el sistema facilita, tanto a profesor como alumno, la evaluación formativa en esta materia.
- **Un sistema de puntuación** que facilita una nota entre 0 y 10, en función de los errores cometidos.

A continuación analizaremos en detalle cada uno de estos puntos.

## 5.5.3 Estructura de un problema diagrama de clases

La estructura de un problema debe contener la información que debe mostrarse al alumno y la información necesaria para poder corregir automáticamente la solución que éste proponga. Esta información se mantiene en un fichero que queda guardado en el repositorio del sistema. En el caso de los ejercicios de diagramas de clase la estructura del fichero es la que se muestra a continuación

```
<identificador de tipo de problema>  
<E> enunciado </E>  
<SOLUCIO1_DC> Solución 1 </SOLUCIO1_DC>  
<SOLUCIO2_DC> Solución 2 </SOLUCIO2_DC>  
.....
```

Como puede observarse las diferentes partes se diferencian a partir de marcas (tags). Inicialmente tenemos un identificador del tipo de problema. A continuación el enunciado y posteriormente las diferentes soluciones propuestas por el profesor. En el enunciado se incluye el nombre de los atributos que figura entre paréntesis. Para cada solución se especifican los siguientes ítems:

- **Las clases.** Para cada una de ellas le asignamos un nombre y la lista de todos los atributos. El primer atributo será el identificador que usaremos para identificar la clase.
- **Las relaciones.** Tal como se ha comentado anteriormente, sólo se permiten relaciones binarias. Para cada una de ellas se debe especificar un nombre que representa el rol, las dos clases que intervienen, su multiplicidad y su navegación. También especificaremos si se trata de una asociación normal, de una agregación o composición, dependencia o generalización. Para las asociaciones calificadas deberemos saber la clase calificada, el atributo calificador y la clase destino.
- **Las clases asociativas.** Para las clases asociativas, necesitamos un nombre cualquiera que puede ser el mismo de la asociación, los atributos y el nombre de la asociación con la que está relacionada.

Teniendo en cuenta estas consideraciones cada solución tiene la estructura siguiente:

```

CLASE                <nombre clase normal>
ID                   <atributo identificador>
ATRIBUTOS            < atributo1, atributo2, ..>
..... (** tantos ítems de este tipo como clases haya **)

RELACIÓN             <nombre relación>
CLASES               <clase1, clase2>
MULTIPLICIDAD        <multiplicidad relación> (** permite valores 0..1, 1, *, 1..*, n:m **)
NAVEGACIÓN           <navegación sentido de la flecha de la asociación> (** <, >, - **)
TIPO RELACIÓN        <tipo>
                    (** valores ASS= asociación, AGR= agregación, COM=composición,
                    ASQ= asociación calificada (atributo calificador), GEN= generalización,
                    DEP= dependencia ****)
..... (** tantos ítems de este tipo como relaciones haya **)

CLASE ASOCIATIVA     < nombre clase asociativa>
ATRIBUTOS            <atributos>
ASOCIACIÓN           <nombre asociación>
..... (** tantos ítems de este tipo como clases asociativas haya **)

```

Una vez creado el fichero con esta estructura y con las funcionalidades específicas de ACME, el profesor lo incorpora al repositorio. Antes de su incorporación se comprueba a través del módulo de validación de problemas de diagramas de clases desarrollado. Según el identificador de tipo de problema, el módulo de validación comprueba que la estructura del fichero sea la correcta y a partir de aquí ya se guarda en el repositorio y está disponible para ser utilizado. En caso que este módulo detecte que no se ajusta a la estructura esperada, el sistema dará el mensaje de error correspondiente y no se guardará en el repositorio.

Una vez guardado ya estará disponible para ser asignado a los cuadernos de los alumnos. Siguiendo las especificaciones de la plataforma ACME, el profesor ya podrá seleccionar los ejercicios y el módulo de generación de cuadernos los asignará a los alumnos. Una vez asignado el problema, el alumno visualizará el enunciado general para este tipo de problemas, que es fijo:

*Proponer un diagrama de clases al supuesto desarrollado a continuación. Se deben definir las diferentes clases y las relaciones entre ellas (asociaciones, agregaciones, composiciones, generalizaciones y dependencias). Las clases han de tener como mínimo todos los atributos que en el enunciado están entre paréntesis. Para las asociaciones se debe especificar su multiplicidad. Finalmente, se debe introducir el diagrama de clases obtenido, siguiendo las especificaciones de ACME (ver manual de usuario)*

y a continuación el enunciado específico del supuesto y guardado en el repositorio.

El alumno visualiza el problema, lo resuelve y ya podrá dibujar el diagrama de clases con el editor específico que presentamos en la siguiente sección. Una vez dibujado el diagrama ya lo podrá enviar a corregir. El corrector utiliza el resto de información del problema para su verificación.

## 5.5.4 Diseño de la interfaz ACME-DC

Otro punto a tratar es el diseño de la interfaz para la visualización y posterior entrada del diagrama de clases. En la Figura 5.3 se muestran los distintos apartados del editor. En la parte superior del editor el alumno visualiza el enunciado del problema y a continuación ya dispone de la interfaz que le permite la entrada del diagrama de clases.

**BASES DE DADES**

**JOSEP SOLER MASO**

Estás a » ACME » Inici » Bases de dades » DIAGRAMAS DE CLASSE » Exercici 1

**ALUMNE: JORDI OLIVERAS ROVIRA**

**EXERCICI: 1 de l'activitat DIAGRAMAS DE CLASSE**

Una empresa està organitzada en departaments. De cada departament nos interessa saber su número (numero\_dep), el nombre (nombre\_dep) y el teléfono (tel\_dep).

Cada departamento tiene asignados empleados de los que nos interesa saber su nombre completo (nombre, apellido1, apellido2), NIF (nif), dirección completa (direccion, codigo\_postal, ciudad), sueldo(sueldo), sexo(sexo) y fecha nacimiento(fecha\_nac). El trabajo de cada empleado está supervisado por otro empleado (nif\_super). Uno de los empleados del departamento desempeña la función de director del departamento.

Cada departamento controla un cierto número de proyectos. De cada proyecto nos interesa saber su código (codigo\_proyecto), su nombre (nombre\_proyecto) y la ciudad (ciudad\_proyecto) donde se realiza. Cada proyecto es controlado por un único departamento.

Un empleado puede trabajar en varios proyectos. Nos interesa saber el número de horas (horas) que cada empleado dedica a cada proyecto.

Un empleado puede tener varias personas dependientes que figuran en su cartilla de la seguridad social. De cada una de estas personas nos interesa saber su número dentro de la cartilla (numero), su parentesco (parentesco) con el empleado, su nombre completo ( nombre\_depe, apellido1\_depe, apellido2\_depe) y su fecha de nacimiento (fecha\_nac\_depe).

El sistema diseñado nos tiene que permitir:

- A partir de un departamento saber todos los empleados que tiene asignados, saber quien es su director y los proyectos que controla.
- A partir de un empleado saber sus datos personales, los datos de su supervisor, en que proyectos ha trabajado y cuantas horas ha dedicado a cada proyecto.
- También nos interesa saber las personas dependientes que tiene a su cargo.
- A partir de un proyecto saber los datos de éste, que departamento lo controla, los empleados que están trabajando en él y las horas dedicadas.

**Data** | **Solució Enviada** | **Resultat**

**4**

**3**

**6**

**2**

**5**

**7**

**Corregir**

**ALUMNE: JORDI OLIVERAS ROVIRA**

Figura 5.3 Interfaz de entrada del diagrama de clases.

Como se puede apreciar, la ventana está dispuesta en varias zonas:

- 1) **Enunciado del problema.** Zona donde el alumno visualiza el enunciado. El nombre de los atributos figura entre paréntesis.
- 2) **Botones de control.** Estos botones tienen las siguientes funcionalidades:
  - Reset: elimina todo el dibujo.
  - Borrar: elimina el elemento seleccionado.
  - Cancelar: cancela la acción de la última tarea realizada.
  - Deshacer: va deshaciendo los últimos cambios realizados en el dibujo.
  - Rehacer: va rehaciendo los cambios deshechos.
- 3) **Área de dibujo.** Área donde el alumno va dibujando su diagrama. El objeto seleccionado en el área de dibujo debe quedar perfectamente identificado y al seleccionarlo se deben visualizar sus propiedades en la pestaña correspondiente para su edición. Al dibujar el diagrama, las clases se pueden desplazar dentro del área de trabajo. Al mover una clase se deben también desplazar las asociaciones relacionadas. Se debe evitar, siempre que sea posible, el cruce de las líneas que representan las asociaciones.
- 4) **Botones de dibujo.** Estos botones tienen las siguientes funcionalidades:
  - Clases: dibuja una clase.
  - Clases asociativas: dibuja una clase asociativa seleccionando previamente una clase y una asociación.
  - Relaciones: los siguientes botones permiten establecer distintos tipos de relaciones entre las dos clases seleccionadas: una asociación, una asociación direccional, una agregación, una composición, una asociación cualificada, una generalización y una dependencia.
- 5) **Zoom.** Esta área muestra una miniatura del diagrama que se está haciendo. Es posible desplazarse para ver las distintas zonas del diagrama. El área de dibujo visualiza el área seleccionada por la miniatura. Es útil en casos de diagramas de clase muy extensos.
- 6) **Pestañas de propiedades.** Permiten la entrada, modificación y eliminación de las propiedades del elemento seleccionado. Cada una de ellas se activa según el elemento seleccionado. En la Figura 5.4 se muestran las pestañas para las clases, las asociaciones, las asociaciones cualificadas y las clases asociativas.
- 7) **Botón de corrección.** Envía el diagrama diseñado al módulo corrector ACME-DC.

La interfaz de dibujo es muy intuitiva y simplemente es necesario ir seleccionando los objetos a dibujar del área de botones de dibujo e ir asignando sus propiedades a través de la pestaña correspondiente. Para el establecimiento de las relaciones simplemente se pulsa el botón de la relación correspondiente (asociación, composición, agregación, etc.) y a continuación se pulsa sobre las clases a relacionar. Por defecto se asigna una multiplicidad (1..\*), modificable a través de la pestaña correspondiente. Una vez completado el diagrama ya se puede enviar a corregir a través del botón corregir.

Esta interfaz se ha implementado con un Applet de Java. Para la correcta utilización de esta herramienta de diseño y dibujo el alumno dispone de un manual de usuario.

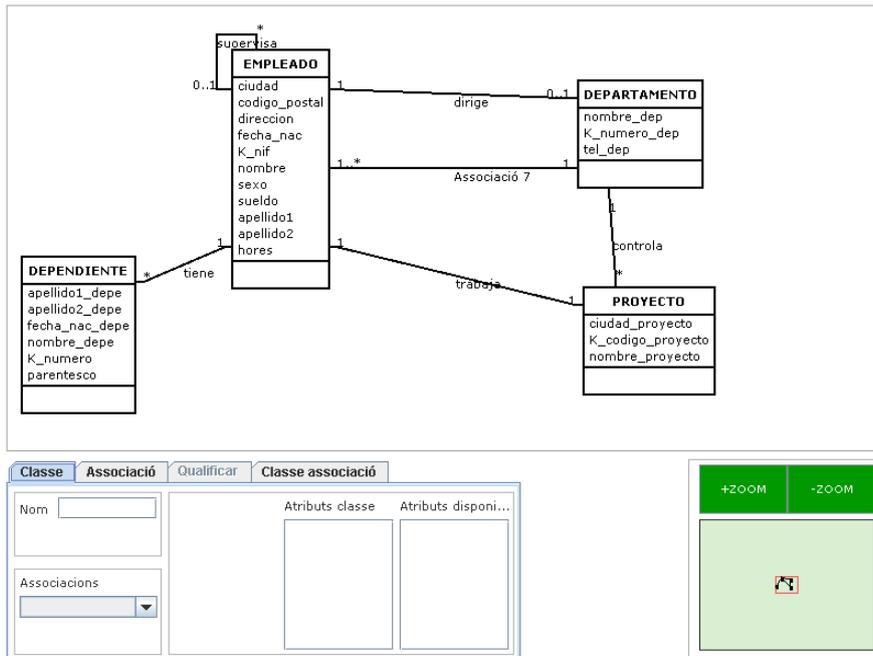


Figura 5.4 Pestañas de interfaz de entrada diagramas de clase

Una vez enviada la solución, la interfaz de recepción de mensajes de ACME es muy sencilla, simplemente se visualiza la solución enviada y una zona para indicar si la solución es correcta o indicar los mensajes de error si es el caso (véase Figura 5.5). Los mensajes de error que se facilitan son:

- A la solución enviada le faltan <objeto>. Donde <objeto> pueden ser clases, relaciones o clases asociativas.
- A la solución enviada le sobran <objeto>. Donde <objeto> pueden ser clases, relaciones o clases asociativas.
- La clase <nombre clase> es incorrecta.
- La relación <nombre relación> es incorrecta.
- La clase asociativa <nombre clase asociativa> es incorrecta.

La solució enviada és:



El resultat de la correcció és:

A la solució enviada li falten Classes associació.  
 La classe "EMPLEADO" es incorrecte; repassa-la.  
 L'associació "trabaja" és incorrecta; repassa-la.

Continuar

Figura 5.5 Mensajes de error ACME-DC.

La Figura 5.6 visualiza la resolución del problema ejemplo planteado en la sección 4.2. En ella observamos todos los elementos importantes de un diagrama de clases: clases con sus atributos, clases asociativas, asociaciones con su multiplicidad, etc. Al seleccionar un elemento, en este caso una clase, automáticamente queda seleccionada la pestaña a través de la cual podemos modificar sus propiedades.

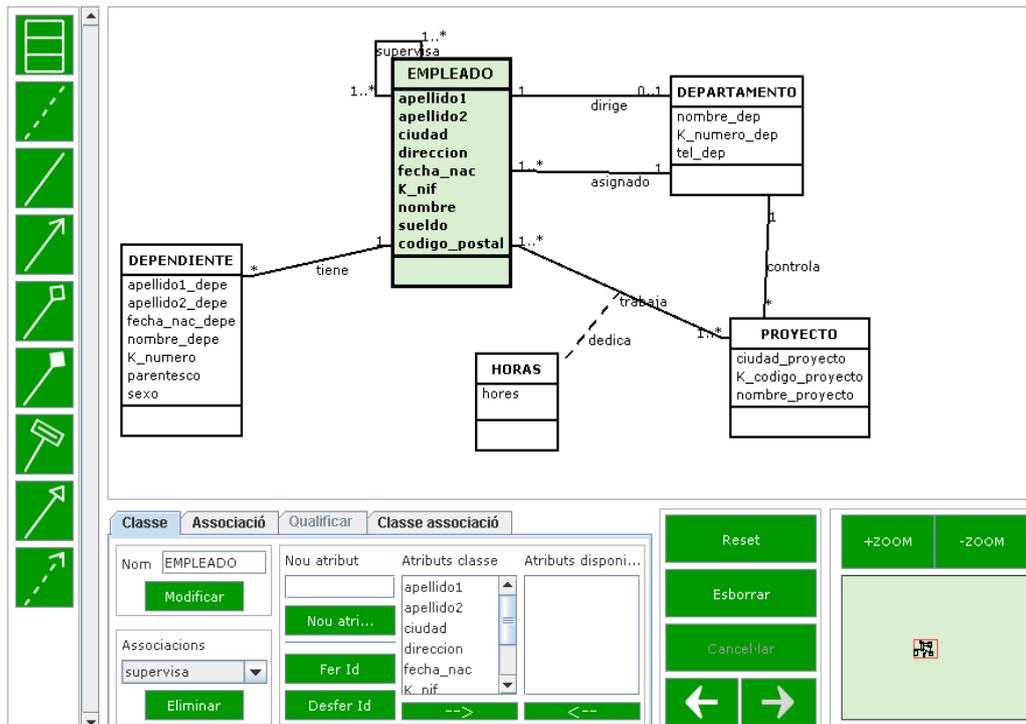


Figura 5.6 Resolución del problema ejemplo.

### 5.5.5 El módulo corrector ACME-DC

El tercer punto a considerar en el proceso de desarrollo de nuestro entorno es el diseño del módulo corrector. Cuando un alumno envía una solución a corregir, ésta se guarda con la misma estructura que las soluciones detalladas en el apartado 5.5.3 y con la posición de las clases y asociaciones dentro del área de trabajo. De esta forma en posteriores consultas, por parte del alumno o del profesor, se puede reconstruir exactamente el mismo diagrama enviado. Esto también permite dibujar una nueva solución a partir de la anterior, de forma que el diagrama visualizado inicialmente será exactamente el mismo, con las clases y asociaciones situadas en la misma posición y el alumno solamente tendrá que realizar las modificaciones oportunas.

La estrategia de corrección se basa en utilizar los nombres de atributos que figuran entre paréntesis en el enunciado del problema. Como que el alumno tiene que utilizar estos nombres, identificamos las clases de la solución del alumno a partir del campo identificador utilizado.

El objetivo de este módulo es comprobar si hay alguna solución correcta asociada a este problema que sea igual a la enviada por el alumno. Si es así se envía el mensaje de “solución correcta” al alumno. Si la solución del alumno no es correcta hay que determinar cuál de las soluciones correctas almacenadas en el repositorio es la más parecida a la enviada. Tomando de referencia la solución más parecida se envían mensajes al alumno que le ayudarán a detectar sus errores. Así pues este módulo está formado por dos apartados: verificación de la solución y retorno de feedback.

#### Verificación de la solución

Este apartado tiene como objetivo detectar si la solución es correcta y en caso contrario retornar la solución correcta más parecida a la del alumno. A partir del resultado de este apartado se retornará el feedback al alumno.

Tal como se muestra en la Figura 5.7, se van accediendo a las diferentes soluciones correctas descritas en el fichero del problema y se van comparando con la enviada por el alumno.

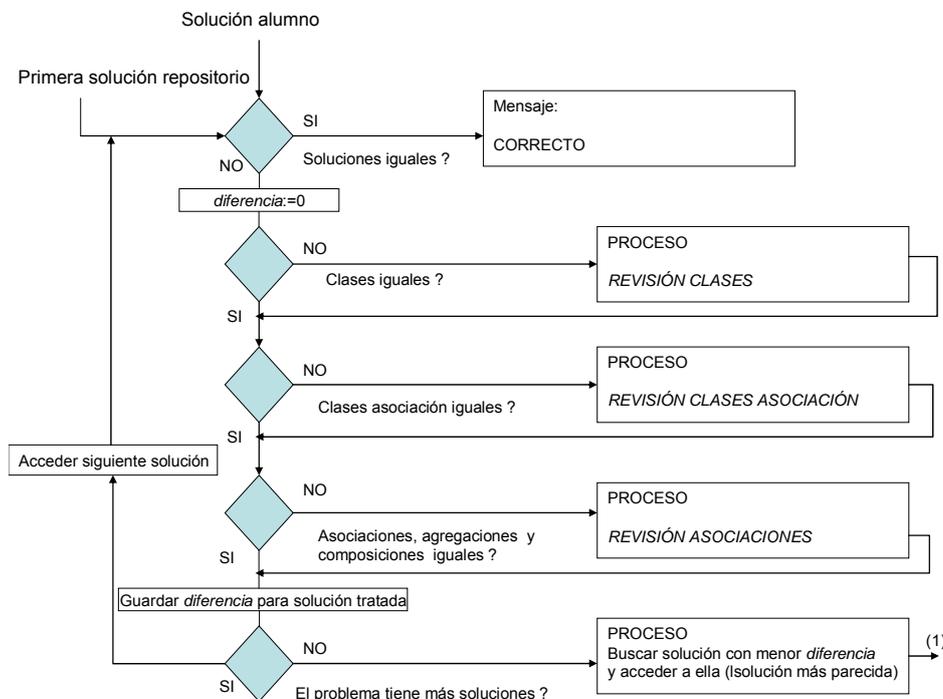


Figura 5.7 Proceso de verificación de la solución en ACME-DC.

En este proceso si se encuentra una de igual se envía el mensaje de “correcto” y finaliza el proceso de corrección. En caso contrario, se guarda el grado de diferencia entre la solución enviada por el alumno y cada una de las correctas. Para la determinación de las diferencias se dispone de varios procesos de verificación que van comprobando las diferencias entre las clases (normales y asociación), y las asociaciones, agregaciones o composiciones. Al final del proceso se escoge la que el grado de diferencia es menor como la solución correcta más parecida a la enviada por el alumno y respecto de la cual se enviarán los mensajes de error al alumno.

Los procesos de revisión de clases, de clases asociativas y de asociaciones realizan las siguientes funciones:

**Proceso de revisión de clases normales.** Se accede a él ya sea porque el número de clases normales sea distinto al del alumno, o bien porque los atributos esperados sean distintos. En función de las diferencias encontradas se van sumando a *diferencia* los valores parametrizables siguientes:

- Si no coincide el número de clases totales se suma  $V_1$  por cada una de las diferencias.
- Si no se puede identificar una clase por no tener el atributo identificador esperado se suma  $V_2$  para cada entidad no identificada.
- Si los atributos no coinciden con los esperados para aquella clase se suma  $V_3$  para cada atributo distinto. Este punto se realiza para cada clase.

De esta forma cuando más diferentes sea las clases enviadas por el alumno respecto a la solución correcta comparada, mayor será el valor de *diferencia*.

**Proceso de revisión de clases asociativas.** Se accede a él ya sea porque el número de clases asociativas sea distinto al del alumno, porque la clase asociativa no dependa de la asociación esperada, o bien porque los atributos esperados sean distintos. En función de las diferencias encontradas se van sumando a *diferencia* los valores parametrizables siguientes:

- Si no coincide el número de clases asociativas se suma  $V_4$  por cada una de las diferencias.
- Si no se puede identificar una clase asociativa porque la asociación de la que dependen no coincide con la de la solución correcta se suma  $V_5$  para cada clase asociativa no identificada.
- Si los atributos no coinciden con los esperados para aquella clase asociativa se suma  $V_6$  para cada atributo distinto. Este punto se realiza para cada clase asociativa.

**Proceso de revisión de relaciones.** Se accede a él ya sea porque el número de relaciones sea distinto en las soluciones comparadas, o porque las clases participantes, la multiplicidad, la navegación o el tipo de asociación de alguna de las relaciones sean incorrectas. En función de las diferencias encontradas se van sumando a *diferencia* los valores parametrizables siguientes:

- Si no coincide el número de relaciones se suma  $V_7$  por cada una de diferencia.
- Si las clases participantes en la relación no coinciden se suma  $V_8$  para cada relación incorrecta.
- Si la multiplicidad de alguna relación no coincide. Se suma  $V_9$  para cada relación incorrecta.
- Si la navegación de alguna relación no coincide se suma  $V_{10}$  para cada relación incorrecta.
- Si el tipo de relación (asociación normal, agregación, composición o asociación cualificada) no coinciden con los esperados se suma  $V_{11}$  para cada relación.

Los valores asignados a los distintos  $V_i$  se han fijado después de varias pruebas y ajustes. Son valores diferentes en función de la relevancia de cada uno de ellos, por ejemplo tiene más importancia la falta de una clase que la falta de un atributo o la falta de una asociación que la de una multiplicidad errónea. De esta forma cuando más diferente sea la solución enviada por el alumno respecto a la solución correcta comparada, mayor será el valor de *diferencia*. La solución con valor de *diferencia* menor, consideraremos que es la solución correcta más parecida a la del alumno. Esta solución será la que utilizaremos para proporcionar el feedback al alumno.

## Retorno de feedback.

Una vez detectada la solución correcta más parecida a la enviada por el alumno, tenemos que enviarle los mensajes de error con respecto a ésta para ayudarle a detectar los errores y facilitarle la obtención de una solución revisada. Los mensajes de error no deben ser muy detallados ya que no se pretende dar la solución correcta ni decirle exactamente los errores cometidos, sino orientarle en la resolución del problema. En la Figura 5.8 se muestra la estrategia seguida y el mensaje de error que se visualiza.

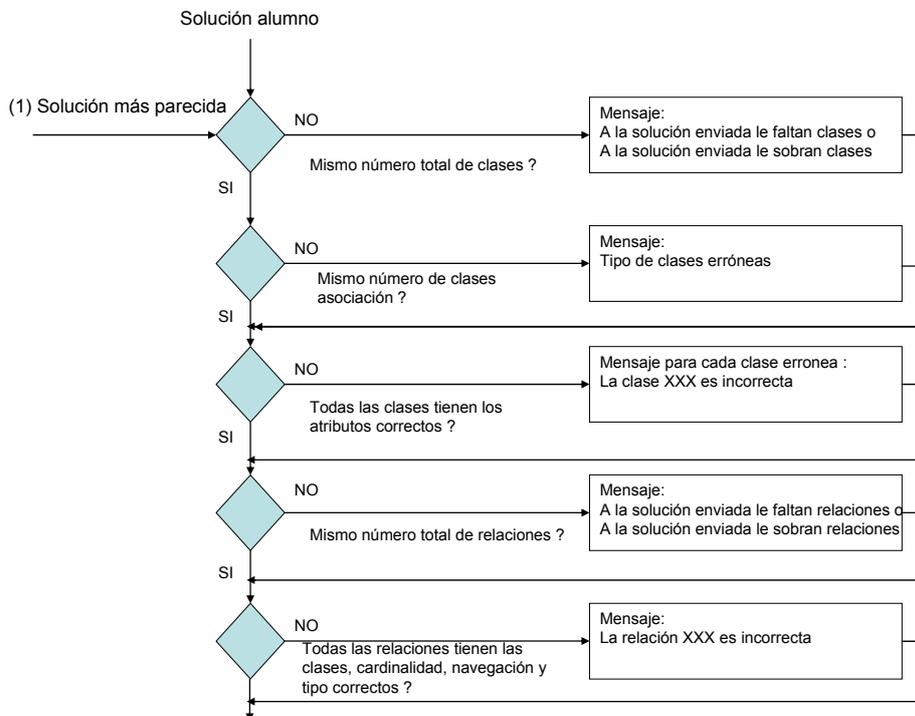


Figura 5.8 Proceso de retorno de feedback en ACME-DC.

En la siguiente Tabla 5.1 se muestra de forma más clara la causa y el mensaje de error facilitado:

	CAUSA	MENSAJE ERROR
1	Si el número total de clases no coincide	A la solución enviada le faltan clases o A la solución enviada le sobran clases
2	Si número clases o clases asociativas no coincide	Tipo de clases erróneas
3	Si atributos no coinciden	La clase <nombre clase> es incorrecta
4	Si número total de relaciones no coincide	A la solución enviada le faltan relaciones o A la solución enviada le sobran relaciones
5	Si las características de la relación son incorrectas	La relación <nombre relación> es incorrecta

Tabla 5.1 Causas y mensajes de error del módulo ACME-DC.

## 5.5.6 Puntuación automática de los diagramas de clase

Un tema importante dentro de los entornos CBA es el referente al sistema de puntuación (ver sección 2.2.7). En muchos casos, y especialmente para la evaluación sumativa, nos interesa que el sistema, además del correcto/incorrecto, nos facilite una nota sobre el ejercicio realizado por el alumno. Es habitual que en diagramas de clases extensos el alumno realice pequeños fallos y no sería justa una mala nota.

Los comentarios realizados en la sección 4.5.6 son perfectamente válidos para los diagramas de clase y la propuesta presentada es totalmente válida. De esta forma la puntuación la podremos calcular utilizando la misma fórmula

$$Nota = 10 - \sum_{i=1}^n (e_i * ne_i * cp_i) \text{ y } Nota \geq 0$$

Donde  $ne_i$  representa el número de errores cometidos en el grupo  $i$  respecto a la solución que el sistema ha detectado como la solución más parecida a la enviada por el alumno (véase sección 5.5.5) en el proceso de corrección automática.

Como que la analogía entre los ERD y los diagramas de clase es considerable y los razonamientos que se siguen para su creación son muy parecidos, hemos establecido que el sistema calcule el número de errores siguiendo las mismas pautas establecidas en la sección 4.5.6. Por lo tanto también hemos establecido dos grupos de elementos: clases y asociaciones equivalentes a entidades y relaciones de los ERD's. Los valores establecidos para cada grupo son también cinco puntos (cinco puntos para las clases y cinco puntos para las asociaciones) y son configurables por parte del profesor.

Para el cálculo del número de errores  $ne_i$  que se cometen en cada grupo de elementos también se han seguido unas pautas muy parecidas. El criterio seguido es:

- Errores en las clases:
  - Se contabiliza un error por cada clase/clase asociativa incorrecta. Por clases incorrectas entendemos aquellas que sobran o que faltan y clases con atributo identificador incorrecto. Por clases asociativas incorrectas entendemos aquellas que sobran o que faltan y las que no dependen de la asociación prevista.
  - 0,25 errores por cada clase identificada correctamente pero en la que hay algún atributo incorrecto distinto al identificador.
- Errores en el grupo relaciones:
  - Se contabiliza un error por cada asociación incorrecta. Por asociación incorrecta entendemos asociaciones que sobran o que faltan y asociaciones entre clases distintas a las previstas.
  - 0,5 errores por cada asociación bien identificada pero con el tipo de asociación erróneo (asociación normal, agregación, composición, etc.).
  - 0,75 errores por cada asociación bien identificada pero con la multiplicidad incorrecta.

Para los factores penalizadores  $cp_i$  de los errores hemos asignado inicialmente los valores obtenidos en las pruebas realizadas para los diagramas ER y detallados en la sección 4.5.6. Así pues, y a falta de un estudio más riguroso, el coeficiente penalizador utilizado para las clases es 1,45 (el equivalente de las entidades del módulo ER) y el utilizado para las asociaciones es 1,62 (el equivalente de las relaciones del módulo ER).

## 5.6 EVALUACIÓN DEL MÓDULO

En este apartado se comenta la preparación de los ejercicios y a continuación se detallan las actividades realizadas y los resultados obtenidos.

### 5.6.1 Preparación de ejercicios de diagramas de clases.

La implementación del módulo ACME-DC finalizó en junio 2009 y entre julio y setiembre se preparó una colección inicial de problemas para ser utilizada en la asignatura de Bases de Datos de la carrera de Ingeniería Técnica en Informática de Gestión que se imparte en el primer semestre del curso. Esta colección está formada por 30 problemas de entre 5 y 8 clases y otras tantas relaciones (asociaciones, asociaciones cualificadas, agregaciones y composiciones). No se han utilizado ni dependencias ni generalizaciones, aunque ACME-DC permite su corrección. El enunciado de estos problemas es idéntico a problemas del Grupo 2 del módulo Entidad/Relación.

Una vez revisado cada problema el sistema verifica que su sintaxis sea la establecida para este tipo de ejercicios. Una vez la sintaxis es correcta se ha testado que la/s solución/es fuesen correctas. El tiempo para la creación y testeo de estos problemas es considerable pero una vez probados disponemos de un problema para ser utilizado siempre que queramos.

### 5.6.2 Actividades realizadas y resultados obtenidos

Este entorno corrector se ha utilizado en la asignatura de Bases de Datos de ITIG impartida en el primer semestre del curso 2009/10. Al ser la primera vez que se utilizaba y de disponer de una colección limitada de problemas se ha realizado una prueba piloto, para comprobar su perfecto funcionamiento y subsanar posibles incidencias.

La metodología seguida ha consistido en impartir una sesión presencial en la que se ha explicado el modelado conceptual utilizando diagramas de clase. Una vez concluido el tema se ha propuesto una lista de ejercicios formada por los 30 problemas que se han entrado en el repositorio ACME. Se dejaba escoger a los alumnos si preferían resolver los ejercicios mediante el nuevo entorno de resolución de ejercicios ACME-DC o hacerlos a mano. Esta actividad se ha planificado como una actividad voluntaria en la que debían resolver 4 ejercicios sin repercusión en la nota. En ella han participado 24 alumnos, el 75% de los matriculados.

Los primeros resultados obtenidos nos confirman la buena aceptación que ha tenido por parte de los estudiantes. Nos confirman la facilidad de uso y que no han tenido problemas en su utilización. Las incidencias han sido mínimas. Los resultados obtenidos en esta prueba piloto son los mostrados en la Tabla 5.2. De los 24 alumnos que han participado en la experiencia, todos han leído todos los problemas. Han enviado soluciones al 95,8% de los ejercicios propuestos y han alcanzado la solución correcta en el 87,6% de los casos, aunque la mayoría de ellos necesitó varios intentos.

ALUMNOS PARTICIPADO	PROBLEMAS	% EJERCICIOS LEIDOS	% SOLUCIONES ENVIADAS	% SOLUCIONES CORRECTAS	% INTENTOS PARA LLEGAR A LA SOLUCIÓN CORRECTA			
					1	2	3	>3
24	4	100,0%	95,8%	87,6%	19,0%	34,5%	16,6%	29,9%

Tabla 5.2 Resultados obtenidos con el entorno ACME-DC.

La mayoría de los participantes (17 de los 24) resolvió correctamente todos los problemas, tal como refleja la Tabla 5.3 aunque los siete restantes no pudieron solucionarlos todos.

ACME-CD	PROBLEMAS RESUELTOS CORRECTAMENTE				
	4	3	2	1	0
NUM. ALUMNOS	17	4	2	1	0
% ALUMNOS	70,8%	16,7%	8,3%	4,2%	0,0%

Tabla 5.3 Problemas resueltos con ACME-DC.

Del análisis de estos primeros datos también observamos que el 34,2% de los errores se deben a errores de multiplicidad, el 31,3% son debido a errores en los tipos de asociación, el 17,5% son debidos a errores en la definición de las clases y el 17,0% restante es debido a los otros errores.

Somos conscientes que disponemos de muy pocos datos para hacer un estudio más exhaustivo, pero las primeras impresiones son positivas. A partir de ahí, estamos preparando una colección mucho más amplia de problemas para utilizarla en los próximos cursos.

## 5.7 APORTACIONES Y CONCLUSIONES

Se ha propuesto, definido y desarrollado el entorno ACME-DC para la corrección/evaluación de diagramas de clases. Se ha integrado a la plataforma ACME y se ha creado una colección inicial de problemas. Se ha utilizado en este curso en la asignatura de “Bases de datos” y las principales ventajas que presenta este módulo son:

- Permite llevar a cabo tanto la evaluación formativa como sumativa de diferentes actividades de diseño conceptual de BD a través de diagramas de clase. Actúa como un sistema CBA.
- Disminución del trabajo del profesor en cuanto a horas dedicadas a la corrección de este tipo de ejercicios, aunque el esfuerzo de la puesta a punto y diseño de ejercicios es considerable.
- Muchos más alumnos se ejercitan en el diseño de bases de datos mediante diagramas de clase y obtienen un resultado inmediato de la corrección.
- Mayor feedback alumno/profesor. Se aclaran muchas dudas que surgen en el momento de la resolución de los ejercicios.
- Permite un seguimiento muy estricto del trabajo del alumno.
- Evaluación automática del trabajo realizado por el alumno.

Las limitaciones de este sistema son básicamente dos:

- El alumno no puede asignar un nombre cualquiera a los atributos, debe escoger el nombre asignado por el profesor.
- El sistema no permite la entrada ni corrección de los métodos de la clase. Tal como se ha comentado en la sección 5.5.1, en el diseño conceptual mediante diagramas de clase, no se definen los métodos de la clase.

El trabajo realizado en el desarrollo y primeros resultados de este entorno han dado lugar al artículo “**A web based e-learning tool for UML class diagrams**” publicado en los Proceedings de la IEEE Engineering Education Conference EDUCON 2010 y referenciado por [SBP+10a].

Una ampliación del anterior artículo y en el que se describe el sistema de evaluación formativa que ofrece ACME-DC ha sido publicado en la revista International Journal of Emerging Technologies in Learning (iJET) con el título “**A formative assessment tool for conceptual database design using UML class diagram**” referenciado por [SBP+10c].

# CAPÍTULO 6. MÓDULO DE ESQUEMAS BASES DE DATOS RELACIONALES

En este capítulo presentaremos el tercer módulo corrector desarrollado: el módulo de esquemas de datos relacionales. Este módulo dará soporte al aprendizaje y docencia del diseño lógico en BD, evaluando si los esquemas de BD realizados por los alumnos son correctos. De esta forma se pretende mejorar la evaluación formativa. El sistema también facilitará la evaluación sumativa, facilitando una nota en aquellos ejercicios que se realicen dentro de una actividad de examen/prueba de validación.

Este capítulo se ha estructurado de la siguiente forma. En la sección 6.1 se realiza una introducción al capítulo. En la sección 6.2 se realiza una breve descripción del modelo relacional y cómo transformar un ERD a un esquema de bases de datos. A continuación, en la sección 6.3 se comenta la necesidad y problemática de este tipo de correctores. En la sección 6.4 se hace un breve resumen de los entornos específicos de corrección de esquemas de bases de datos relacionales. En la sección 6.5 planteamos el problema a resolver, detallamos la solución aportada y su integración a la plataforma ACME. Seguidamente en la sección 6.6 se describen las actividades y resultados obtenidos. Finalmente, en la sección 6.7 se describen las aportaciones y conclusiones alcanzadas.

## 6.1 INTRODUCCIÓN

Las bases de datos relacionales son muy comunes en la sociedad actual. Por este motivo la habilidad de diseñar buenas bases de datos es una de las competencias que se debe adquirir en un curso de Bases de Datos de las carreras de Informática. Distintas técnicas y modelos han sido desarrollados para ayudar en el proceso de creación de bases de datos. Tal como hemos visto, el modelo más utilizado para el diseño conceptual es el Entidad/Relación (ER) que proporciona una visión general de alto nivel, independiente del modelo de datos en que se implementará en la base de datos. El diseño de un ERD suele ser el primer paso en el aprendizaje del diseño de bases de datos. El siguiente paso consiste en la obtención del esquema de bases de datos, que ya nos indicará la estructura de las tablas a crear y utilizar en un SGBD. Para ello, el modelo relacional [Cod70] es el más ampliamente aceptado y su enseñanza se lleva a cabo en todos los cursos de bases de datos. Un esquema de bases de datos relacional describe todas las tablas que formarán la base de datos, con sus atributos, clave principal y claves foráneas.

En la enseñanza del modelo relacional se persiguen varios objetivos:

- Aprendizaje de los conceptos básicos.
- Mapeo y obtención de un esquema de bases de datos relacional a partir de un ERD.
- Normalización de las bases de datos.
- Aprendizaje de lenguajes de consulta sobre una BD relacional. Aunque el SQL es el mayoritariamente empleado, a nivel académico su enseñanza suele estar precedida por la enseñanza del álgebra relacional.

Debido a la importancia del tema es fundamental realizar un buen aprendizaje de los conceptos básicos del modelo relacional. Estos conceptos se suelen impartir en sesiones presenciales. Con los conocimientos adquiridos de diseño conceptual más los que ahora adquiere del modelo relacional, el siguiente paso consiste en el diseño lógico de la base de datos. Para ello se enseñan los pasos a aplicar para transformar un ERD a un esquema de bases de datos. El mapeo y obtención de un esquema de BD a partir de un ERD es una competencia que el alumno debe adquirir, por lo que será muy importante que el alumno adquiera soltura en este tema. En la aplicación de estos pasos, a menudo, el alumno suele cometer errores y por este motivo es importante que se ejercite con diferentes supuestos parecidos a situaciones reales.

El proceso de evaluación formativa del diseño lógico consiste en la revisión de los esquemas diseñados por cada alumno y la realización de las anotaciones correspondientes para que el alumno pueda darse cuenta de los errores cometidos. A partir de los resultados obtenidos en la corrección, el profesor adapta las diferentes sesiones, haciendo hincapié en los puntos donde detecte deficiencias de aprendizaje. Estas tareas de revisión son laboriosas y conllevan una gran dedicación difícilmente asumible por el profesor, dificultando la aplicación del sistema de evaluación formativa. Un entorno CBA especializado en la corrección de este tipo de ejercicios es una buena ayuda al profesor.

En la construcción del esquema el alumno tiene que poner de manifiesto los conceptos básicos del modelo y saberlos aplicar a un supuesto dado que deberá ser analizado previamente. Así pues, la obtención de un esquema de bases de datos la podemos enmarcar en los niveles cognitivos intermedios (aplicación y análisis) dentro de la taxonomía de Bloom. Tal como se ha comentado en el capítulo precedente, en un entorno de CBA es difícil evaluar estos niveles cognitivos a través de preguntas de respuesta fija y se requiere de entornos más especializados.

Nuestro objetivo es solventar estas limitaciones, diseñando y desarrollando un entorno CBA especializado que nos permita la evaluación automatizada de los niveles cognitivos intermedios de la taxonomía de Bloom en materia de esquemas de bases de datos. Para ello, el profesor preparará ejercicios que serán almacenados en el repositorio, que posteriormente serán asignados a los alumnos que deberán obtener el esquema de la base de datos correspondiente y enviarlo al sistema que lo corregirá automáticamente y retornará el feedback apropiado. El sistema debe soportar la evaluación formativa de manera que el profesor pueda controlar el trabajo del grupo y no deba perder tiempo en la tediosa labor de corrección. Además el sistema proporcionará una puntuación automática de cada ejercicio para poder realizar la evaluación sumativa. Con las puntuaciones obtenidas el sistema facilitará una evaluación final del alumno.

Este entorno para el aprendizaje y evaluación de esquemas de bases de datos relacionales se debe integrar en la plataforma ACME, actuando conjuntamente como un sistema CBA, con la que se realizarán las correspondientes pruebas sobre los alumnos de la asignatura de BD. El objetivo es corregir y evaluar las soluciones enviadas por el alumno, facilitando los mensajes más adecuados que le guíen hacia la solución correcta. Este módulo es el tercero de los seis módulos que conforman ACME-DB.

Comentar finalmente que los otros apartados referentes al aprendizaje de la normalización y de los lenguajes de consulta relacionales se describirán en los próximos capítulos.

## 6.2 DESCRIPCIÓN DEL MODELO RELACIONAL

El modelo de datos relacional fue introducido por Codd [Cod70] y pronto acaparó gran atención debido a su simplicidad y a sus fundamentos matemáticos. El modelo utiliza el concepto de *relación matemática* (que tiene una apariencia similar a una tabla de valores a la que coloquialmente llamamos “tabla”) como su bloque de construcción básico. De esta forma una base de datos se representará como una colección de relaciones. Este modelo tiene sus bases teóricas en la teoría de conjuntos y la lógica de predicados de primer orden. De forma coloquial decimos que una base de datos está formada por un conjunto de tablas interrelacionadas entre sí, de forma que en cada fila representamos una entidad o algún tipo de relación del modelo ER. En la terminología del modelo relacional a cada fila de la tabla la llamamos **registro o tupla** y a cada columna la llamamos **campo o atributo**. Los conceptos básicos del modelo relacional son:

- **Dominio.** Es un conjunto de valores indivisibles que representa los posibles valores que puede adquirir un campo. El modelo relacional no acepta valores compuestos ni multivaluados.
- **Esquema de relación** (o esquema de la tabla). Un esquema de relación  $R(A_1, A_2, \dots, A_n)$  se compone de un nombre de relación  $R$  y una lista de atributos  $A_1, A_2, \dots, A_n$ . Los posibles valores que puede tener cada atributo  $A_i$  forman el dominio de  $A_i$  expresado por  $\text{dom}(A_i)$ . Un esquema de relación sirve para describir una relación. El grado de una relación (de una tabla) es el número de atributos que tiene la relación. Todos los esquemas de relación han de tener un atributo o conjunto de atributos que formen la clave primaria que se acostumbra a subrayar.
- **Relación** (o también ejemplar, instancia de relación). Una relación  $r$  del esquema de relación  $R(A_1, A_2, \dots, A_n)$  es un conjunto de  $m$  tuplas  $r = \{t_1, t_2, \dots, t_m\}$  donde cada tupla  $t$  es una lista ordenada de  $n$  valores  $t = \langle v_1, v_2, \dots, v_n \rangle$ , donde cada valor  $v_i$  con  $1 \leq i \leq n$  es un valor del dominio  $\text{dom}(A_i)$  o bien el valor especial “null”.
- **Esquema de una base de datos.** Un esquema de una base de datos relacional (EBD) está formado por el conjunto de esquemas de relación  $\text{EBD} = \{R_1, R_2, \dots, R_n\}$  más las posibles restricciones de integridad.
- **Ejemplar de una base de datos.** Un ejemplar BD de un esquema de base de datos relacional EBD, es un conjunto de instancias de relaciones  $\text{BD} = \{r_1, r_2, \dots, r_n\}$  tal que cada  $r_i$  es un ejemplar de  $R_i$  y tal que las relaciones  $r_i$  satisfacen las restricciones de integridad especificadas.

Durante el diseño lógico de un sistema de información se obtienen los esquemas de relación de cada una de las relaciones que forman la base de datos, o sea el esquema de la base de datos. El modelo relacional impone varios tipos de restricciones:

**Restricciones de dominio.** Especifican que el valor de cada atributo  $A_i$  tiene que ser un valor simple del dominio de  $A_i$ .

**Restricciones de clave.** Especifican que todos los registros de una tabla han de ser distintos ya que no podemos tener dos registros iguales. Como mínimo el valor del campo clave tiene que ser distinto. En la notación utilizada para representar un esquema de relación se suele subrayar el campo(s) clave(s).

**Restricción de integridad de entidades.** Establece que ningún valor de la clave primaria puede ser nulo pues la clave sirve para identificar los registros de una relación. Las restricciones de clave y de integridad de entidades se establecen para cada relación a nivel individual.

**Restricción de integridad referencial.** Se establece entre dos relaciones y sirve para mantener la consistencia entre los registros de ambas relaciones. De manera informal, la restricción de integridad referencial establece que si un registro  $p$  de una tabla  $A$  hace referencia a otro registro  $q$  de una tabla  $B$ , es obligatorio que en la tabla  $B$  exista este registro  $q$  referenciado. Para introducir estos conceptos de una manera formal definimos el concepto de clave externa o foránea. Un atributo o un conjunto de atributos en el esquema de relación  $R_1$  es una **clave foránea (o externa)** de  $R_1$  si se cumple que:

- El atributo (o atributos) que forman la clave foránea en  $R_1$  tienen el mismo dominio que la clave primaria de otro esquema de relación  $R_2$ . Se dice que la clave externa hace referencia a la relación  $R_2$ .
- Un valor de la clave externa en un registro de  $R_1$  tiene que existir como valor de la clave principal en algún registro de  $R_2$  o tener el valor nulo. En el primer caso tenemos que  $t_i[CF] = t_j[CP]$  y decimos que la tupla(registro)  $t_i$  de la relación  $R_1$  hace referencia a la tupla  $t_j$  de la relación  $R_2$ .

Estas condiciones que definen una clave externa determinan una **restricción de integridad referencial** entre dos esquemas de relación  $R_1$  y  $R_2$ . En una base de datos con muchas tablas habrá en general muchas restricciones de integridad referencial que deben cumplir todos los ejemplares de este esquema.

Cuando el alumno ya conoce el modelo ER y el modelo relacional, el siguiente paso es seguir una metodología para la obtención de un esquema de bases de datos relacional a partir de los ERD's. Para ello seguimos la metodología de [EN07] consistente en aplicar un proceso de mapeo siguiendo los pasos siguientes:

**Paso 1)** Para cada tipo de entidad regular (fuerte) del diagrama ER se crea una relación (tabla)  $R$  que contiene todos los atributos simples de la entidad. Se debe escoger un atributo o un conjunto de atributos como clave primaria de  $R$ .

**Paso 2)** Para cada tipo de entidad débil del diagrama E/R se crea una relación (tabla)  $R$  que contiene todos los atributos simples de la entidad. La clave primaria de esta relación será la combinación de la clave primaria de la entidad fuerte de la que depende más la clave parcial de la entidad débil. Además la clave principal de la entidad fuerte será también clave foránea de esta nueva relación.

**Paso 3)** Para cada tipo de relación binaria 1:1 del diagrama ER se identifican las relaciones (tablas)  $S$  y  $T$  que intervienen en la relación 1:1. Se elige una de las tablas, por ejemplo la que representa la entidad  $S$  y se añade como clave externa en  $S$  la clave principal de  $T$ . Si una de las dos entidades tiene participación total, escogeremos esta en el papel de  $S$ . Añadiremos también los posibles atributos de la relación 1:1, como atributos de  $S$ .

**Paso 4)** Para cada tipo de relación binaria 1:n del diagrama ER, identificamos la relación (tabla)  $S$  que representa la entidad del lado  $n$  de la relación 1:n. Añadimos como clave externa en  $S$  la clave primaria de la relación  $T$  que representa la entidad del lado 1. Si la relación tiene atributos, estos se añaden también a la relación  $S$ .

**Paso 5)** Para cada tipo de relación binaria n:m del diagrama ER, se crea una nueva relación (tabla) T. La clave principal de esta nueva tabla estará formada por la combinación de las claves principales de las entidades que intervienen en la relación n:m que a su vez serán también claves externas. Todos los atributos de la relación n:m se incluirán en esta nueva tabla.

Comentar que siempre es posible transformar las relaciones 1:1 y 1:n de un modo similar a cómo se hace con las relaciones n:m. Esta alternativa resulta especialmente útil para evitar muchos valores nulos en claves externas cuando se dan pocos ejemplares de la relación. En este caso la clave primaria de la relación será sólo una de las claves externas que en el caso de una relación 1:n será la del lado n y en el caso de una relación 1:1 se escogerá, si la hay, la de la entidad con participación total. Estas posibilidades hacen que se puedan construir distintos esquemas de bases de datos y que todos ellos sean correctos.

En la bibliografía de bases de datos también se suelen comentar más pasos para determinar como se transforman las relaciones n-arias y los atributos multivaluados. En nuestro caso no vamos a permitir relaciones n-arias y sólo vamos a utilizar atributos simples que es lo habitual.

Una vez descritos los conceptos básicos del modelo relacional, la Figura 6.1 muestra el diagrama ER del mismo ejemplo referenciado en el apartado 4.2. que nos servirá también de ejemplo para este módulo corrector de esquemas de bases de datos relacionales.

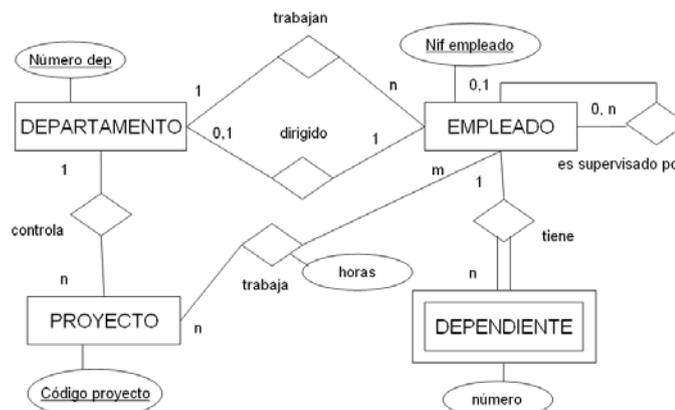


Figura 6.1 Ejemplo de diagrama ER del supuesto planteado.

Una vez obtenido el diseño conceptual, el siguiente paso es realizar el diseño lógico a partir del cual ya obtendremos el esquema de la base de datos y las diferentes tablas en un SGBD cualquiera. La Figura 6.2 muestra el correspondiente esquema de la base de datos diseñada siguiendo los pasos descritos.

**DEPARTAMENTO** (Número dep, nombre\_dep, tel\_dep, *nif\_director*)  
**EMPLEADO** (Nif, nombre, apellido1, apellido2, direccion, codigo\_postal, ciudad, sueldo, sexo, fecha\_nac, *nif\_super*, *numero\_dep*)  
**PROYECTO** (Codigo proyecto, nombre\_proyecto, ciudad\_proyecto, *numero\_dep*)  
**DEPENDIENTE** (Nif, numero, fecha\_nac\_depe, nombre\_depe, apellido1\_depe, apellido2\_depe, parentesco)  
**TRABAJA EN** (Nif, Codigo proyecto, horas)

Figura 6.2 Esquema de la base de datos del supuesto planteado.

En este esquema podemos ver cada relación (tabla) con sus campos. En negrita/subrayado figuran las claves principales, en negrita/cursiva las claves foráneas y negrita/cursiva/subrayado las claves principales que además son claves foráneas.

Como que a un supuesto determinado podemos especificar varios ERD's correctos, para cada uno de ellos se obtendrá un esquema de bases de datos distinto que deberemos considerar correcto. Así pues, al igual que con los ERD's y los diagramas de clase, dado un supuesto consideraremos que puede tener varias respuestas correctas.

Una vez se dispone del esquema de la base de datos, otro de los objetivos que se persiguen es valorar la calidad del esquema obtenido. Para ello la normalización de BD establece un conjunto de formas normales que aseguran la no aparición de problemas. En el capítulo 7 se tratará el módulo de normalización de bases de datos.

A partir del esquema de bases de datos normalizado ya se podrán crear las tablas obtenidas en un SGBD relacional cualquiera. Para ello en todos los cursos de Bases de datos se enseña el lenguaje de consulta estándar SQL. Esta es otra de las competencias básicas que el alumno debe adquirir y que en el Capítulo 9 se detallará cómo se ha desarrollado en ACME-DB. Previamente a la enseñanza de SQL, suele ser habitual impartir algunas sesiones de álgebra relacional para facilitar la posterior comprensión de éstas. En el Capítulo 8 se tratará el módulo de consultas de álgebra relacional.

### **6.3 NECESIDADES Y PROBLEMÁTICA EN LA OBTENCIÓN DE ESQUEMAS DE BASES DE DATOS**

La metodología tradicional que se aplica consiste en desarrollar los conceptos teóricos en sesiones de clase presenciales. Se explican los conceptos básicos del modelo relacional y los pasos necesarios para pasar un ERD a un esquema de BD relacional. A partir de ahí se realizan diferentes ejercicios para aplicar y consolidar los conocimientos del modelo relacional. Una de las competencias a adquirir por parte del alumno es la de obtener el esquema de una base de datos a partir de su diseño conceptual. Puesto que en la realización de un esquema de bases de datos se integran gran parte de los conocimientos que debe adquirir del modelo, el profesor debe evaluar si ha adquirido esta habilidad. La adquisición de esta habilidad pasa por un proceso de aprendizaje que siguiendo la taxonomía de Bloom podemos resumir en los siguientes apartados.

- **Conocimiento.** El profesor define y describe en clase los conceptos básicos del modelo relacional. El alumno debe ser capaz de recordar, definir y describir los conceptos expuestos en la sección 6.2.
- **Compresión.** El alumno asocia claves principales a las correspondientes tablas, convierte una entidad y/o relación del modelo ER a las correspondientes tablas, asocia claves foráneas a cada relación, etc. Identifica los pasos necesarios para el mapeo de un ERD a un esquema de relación.
- **Aplicación.** A partir de un ERD sencillo, aplica los pasos y obtiene el esquema de la BD, identificando las claves principales y las externas.
- **Análisis.** El alumno debe ser capaz de analizar un supuesto real, identificar las entidades existentes y sus relaciones y construir el correspondiente ERD. Una vez obtenido, transforma el ERD al esquema de la BD, analizando cada una de las relaciones resultantes y verificando que sean correctas.

Después de impartir docencia en cursos de bases de datos durante bastantes años hemos constatado que a pesar de ser un proceso secuencial, el mapeo de un ERD a un esquema de base de datos es una actividad en que el estudiante suele cometer bastantes errores. De hecho las sesiones teóricas no son el mejor marco para la enseñanza de esta actividad ya que no permiten una práctica guiada. Para la adquisición de esta habilidad, especialmente para evaluar el nivel de análisis, es necesario que el alumno obtenga el esquema de la base de datos de diferentes y amplios supuestos. En este apartado es donde falla la metodología tradicional ya que es muy laborioso corregir muchos esquemas a muchos alumnos y facilitarles el feedback oportuno a sus errores. Los sistemas de e-learning especializados en la corrección de este tipo de ejercicios facilitan la labor de los profesores.

## **6.4 CORRECCIÓN AUTOMÁTICA DE ESQUEMAS DE BASES DE DATOS RELACIONALES**

Tal como se ha descrito en la sección 2.4.3, se han desarrollado muy pocas herramientas que permitan la corrección de esquemas de bases de datos relacionales. Uno de los motivos es que desde un punto de vista académico se suele dar más importancia al diseño conceptual que al diseño lógico. El paso de uno a otro consiste en analizar los diferentes elementos de un ERD e ir creando las tablas necesarias, especificando su clave principal, sus claves externas y sus posibles restricciones. Por otra parte, muchas herramientas CASE disponen de utilidades que automáticamente crean el esquema de la base de datos a partir del diagrama ER. Desde un punto de vista pedagógico consideramos que es muy importante que el alumno sepa construir el esquema de la BD a partir de un diagrama conceptual. El trabajo más relevante en este apartado es el desarrollado por el grupo encabezado por A.Mitrovic [MM04] [MMM06] llamado ERM-Tutor, a través del cual los alumnos practican los pasos para transformar un diagrama ER al correspondiente esquema de la base de datos. ERM-Tutor es una aplicación web enmarcada dentro de los sistemas de tutorización Inteligentes (ITS). El tutor dispone de un conjunto de problemas y el sistema se basa en un conjunto de reglas que representan el dominio del conocimiento. El proceso de corrección se divide en varias tareas que se corresponden con los pasos de la transformación del modelo ER al modelo relacional. Los problemas se presentan como un diagrama ER y el alumno puede escoger cualquiera. Con este diagrama el sistema permite al alumno ir aplicando los distintos pasos de la transformación. Por ejemplo, el alumno puede escoger el paso 1 (transformación de las entidades regulares) y tendrá que ir escribiendo las tablas de cada entidad. En todo momento puede ir comprobando si el trabajo realizado es correcto. Dispone de una interfaz apropiada para seguir cada uno de los pasos y también dispone de un área de “feedback” y otra de ayuda. Internamente el sistema guarda tanto la información del diagrama ER como cuatro listas con información textual para guardar entidades, relaciones, atributos y conexiones. Los inconvenientes que vemos en el ERM-Tutor son:

- Ofrece un conjunto igual de problemas a todos los estudiantes.
- Parte siempre de un diagrama entidad-relación, no del enunciado de un supuesto.
- Los alumnos pueden ir obteniendo feedback en cada momento. Consideramos que es mejor realizar por completo el problema y luego enviarlo a corregir.
- Sólo es utilizado para el auto-aprendizaje, no para la evaluación, por lo que no ofrece ninguna puntuación del ejercicio.

Nuestro objetivo en este trabajo es desarrollar un entorno CBA para el aprendizaje del mapeo de un ERD al correspondiente esquema de la base de datos. Los ejercicios asignados a los alumnos serán corregidos por el sistema que ofrecerá los comentarios más adecuados en cada momento y facilitará una puntuación en las actividades de evaluación sumativa.

## **6.5 ACME-EBD: MÓDULO DE ESQUEMAS DE BASES DE DATOS RELACIONALES**

En esta sección se detalla el módulo referente a la evaluación de esquemas de bases de datos relacionales que denotamos como ACME-EBD. En la sección 6.4.1 se determinan los requisitos que debe satisfacer el sistema. En la sección 6.4.2 se presenta una visión general de la solución desarrollada. En las cuatro siguientes secciones se detallan los aspectos más técnicos del módulo: la estructura de un problema de esquemas de bases de datos relacionales, su interfaz, el módulo de corrección y feedback y finalmente la puntuación del problema.

## 6.5.1 Requisitos del sistema ACME-EBD

El problema que planteamos consiste en desarrollar un sistema que de forma totalmente automática sea capaz de corregir un esquema de bases de datos relacional de un supuesto planteado, como por ejemplo el mostrado en la Figura 6.2. Un mismo supuesto puede tener varias soluciones correctas, por lo que el sistema deberá tener en cuenta que la solución puede no ser única. Los requisitos establecidos son:

- El sistema debe contemplar todos los elementos descritos anteriormente y que conforman el esquema de una base de datos:
  - Un conjunto de relaciones (tablas) con su nombre.
  - Cada una de estas tablas estará formada por un conjunto de campos (atributos) de los que debemos especificar el atributo(s) clave principal y las claves foráneas (externas)
- El sistema debería facilitar las funcionalidades siguientes:
  - Disponer de una interfaz apropiada para la entrada de las tablas. De esta manera, tiene que permitir poder asignar cualquier nombre a las tablas. A continuación, y para cada tabla, entrar los nombres de los campos, y especificar las claves principales y foráneas.
  - El sistema debe facilitar la evaluación formativa. Cuando el alumno envíe una respuesta, el sistema recogerá la información asociada a los distintos objetos dibujados y los enviará al servidor. Esta información será guardada por ACME-EBD como una solución enviada por el alumno a uno de sus ejercicios. El módulo desarrollado comparará la solución enviada con las soluciones correctas asociadas al problema. En caso que sea correcta lo notificará al alumno mediante un “Correcto”. En caso de que el corrector determine que es incorrecta facilitará pequeñas notificaciones (feedback) para ayudar al alumno a encontrar el error y poder enviar una nueva solución.
  - El sistema debe facilitar una puntuación a cada ejercicio. En las actividades de evaluación el sistema debe facilitar la nota obtenida de forma que nos pueda servir como un elemento para su evaluación sumativa.
  - El sistema se debe integrar en la plataforma ACME, con lo cual se compartirán todas las funcionalidades ACME, como por ejemplo la gestión de cursos, asignaturas, temas, actividades, alumnos, envío de mensajes, foros, chats, etc.

## 6.5.2 Análisis inicial de ACME-EBD

El problema visto desde un punto de vista general se puede tratar como la interpretación y corrección de una solución. En este tipo de ejercicios uno de los principales problemas es interpretar la semántica de la solución enviada por el alumno, en nuestro caso el identificar entidades y atributos. Otro de los problemas es que se puede aceptar más de una solución correcta, con lo que los mensajes de error enviados tendrán que referirse a la solución correcta más parecida a la enviada por el alumno.

Para la identificación de los nombres de atributos y claves se ha adoptado el mismo sistema que el módulo corrector de diagramas ER (ver sección 4.5). O sea, se facilita el nombre de los diferentes atributos que figurarán entre paréntesis en el enunciado del problema. Este sistema tiene la ventaja de que nos permite reutilizar el mismo enunciado que los mismos ejercicios planteados para los ERD's a los que sólo habrá que añadir las soluciones de los esquemas de base de datos que aceptamos como correctos. Facilitando esta información no ayudamos apenas al alumno, en todo caso le especificamos de forma más concisa los atributos que debe guardar el sistema. En cambio no pondremos ninguna limitación respecto al nombre de las tablas y el alumno podría asignarles el nombre que desease. Tampoco le especificamos la clave principal que debe ser capaz de identificar.

La solución aportada se tiene que integrar a la plataforma ACME. Para ello es necesario desarrollar:

- **La estructura del problema.** Para este tipo de problemas debe estar formada por un enunciado y las posibles soluciones correctas. El profesor mediante un editor cualquiera o el específico de ACME editará la estructura del problema. A continuación con las funcionalidades de ACME, lo incorporará al repositorio y a partir de aquí ya estará disponible para ser utilizado.

- **Una interfaz apropiada** a través de la cual el alumno entrará las relaciones del esquema de las bases de datos.
- **Un módulo corrector** que interpretará la solución enviada y la comparará con las soluciones correctas, descritas en la estructura del problema guardado en el repositorio. Este módulo corrector determina el feedback a retornar al alumno. El propio sistema ACME se ocupa de guardar toda esta información y con ella facilita, tanto a profesor como alumno, la evaluación formativa en esta materia.
- **Un sistema de puntuación** que asigna una nota entre 0 y 10 en función de los errores cometidos.

Veamos con más detalle cada uno de estos apartados.

### 6.5.3 Estructura de un problema de esquemas de BD

La estructura general de un problema es la que utiliza ACME para visualizar el enunciado del problema y es la que utilizará el módulo corrector para la corrección de las soluciones enviadas por los alumnos. Esta estructura se guarda en un fichero que se puede editar con cualquier editor y que se guarda en el repositorio. Para este tipo de problemas, en los que puede haber varias soluciones correctas, la estructura del problema está formada por las siguientes partes:

```

<identificador de tipo de problema>
<E> enunciado </E>
<SOLUCIO1_EBD> Solución 1 </SOLUCIO1_EBD>
<SOLUCIO2_EBD> Solución 2 </SOLUCIO2_EBD>
.....

```

Como puede observarse las diferentes partes se diferencian a partir de marcas (tags). Inicialmente tenemos un identificador del tipo de problema. A continuación el enunciado y posteriormente las diferentes soluciones propuestas por el profesor. En el enunciado se incluye el nombre de los atributos que figura entre paréntesis. Para cada solución se especifican los siguientes ítems:

- **Todas las tablas.** Para cada una de ellas le asignamos un nombre y la lista de todos los campos.
- **Para cada tabla debemos especificar sus campos clave principal y claves foráneas.** Generalmente para estas últimas, utilizaremos el mismo nombre que el de la clave principal de la tabla a que hace referencia la clave foránea. En el caso que una clave sea principal y al mismo tiempo foránea lo deberemos especificar adecuadamente.

Así pues cada una de las soluciones tiene la estructura:

```

NOMBRE TABLA           < nombre tabla >
CLAVE PRINCIPAL        < campo/s clave principal>
CLAVE FORANEA          < campo/s clave foranea>
OTROS CAMPOS           < otros campos>

..... (** tantos ítems de este tipo como tablas haya **)

```

Una vez creado el fichero con esta estructura, ACME dispone de las funcionalidades específicas para incorporarlo al repositorio. Antes de su incorporación, se valida a través del módulo de validación de problemas de esquemas de bases de datos desarrollado. Según el identificador de tipo de problema, el módulo de validación comprueba que la estructura del fichero sea la correcta y a partir de aquí ya se guarda en el repositorio y en el caso que este módulo detecte que no se ajusta a la estructura esperada, el sistema dará el mensaje de error correspondiente y no se guardará en el repositorio.

Una vez guardado el problema en el repositorio ya estará disponible para ser asignado a los cuadernos de los alumnos. Siguiendo las especificaciones de la plataforma ACME, el profesor ya podrá seleccionar los ejercicios y el módulo de generación de cuadernos los asignará a los alumnos. Una vez asignado el problema, el alumno visualizará el enunciado general y fijo para este tipo de problemas:

Diseñar un esquema de bases de datos relacionales al supuesto desarrollado a continuación. Utilizando el modelo Entidad/Relación definir las diferentes entidades y las relaciones entre ellas. Las entidades han de tener como mínimo los atributos que en el enunciado figuran entre paréntesis. Para las relaciones especificar su tipo (1:1, 1:n, n:m) y si son parciales o totales. A continuación proponer un esquema de bases de datos relacional equivalente al diagrama ER definido, especificando cuáles son las claves principales y las foráneas. Finalmente y desde esta misma página introducir el esquema de bases de datos relacional obtenido, siguiendo las especificaciones de ACME (tablas, clave principal, claves foráneas, otros atributos).

y a continuación el enunciado específico del supuesto y guardado en el repositorio.

Una vez el alumno ha resuelto el problema podrá introducir el esquema obtenido en el área de trabajo. El corrector, tal como se detalla más adelante, utiliza el resto de información del problema para su verificación.

Comentar que como el enunciado es el mismo que para los ejercicios del modelo ER y que la única cosa que varía es la descripción de las soluciones, en este tipo de problemas podemos definir para un mismo enunciado los dos tipos de soluciones y, en el momento de asignar los problemas en el cuaderno del alumno, el profesor determina si se presentan como problemas ER en los que el alumno debe realizar el ERD o como problemas de esquema de bases de datos. En estos casos la definición del problema es:

```
<identificador de tipo de problema>
<E> enunciado</E>
<SOLUCIO1_EBD> Solución 1 </SOLUCIO1_EBD>
<SOLUCIO2_EBD> Solución 2 </SOLUCIO2_EBD>
.....
<SOLUCIO1_ER> Solución_ER 1</SOLUCIO1_ER>
<SOLUCIO2_ER> Solución_ER 2</SOLUCIO2_ER>
.....
```

#### 6.5.4 Diseño de la interfaz ACME-EBD

Otro apartado importante es el diseño de la interfaz para la visualización del enunciado y posteriormente la entrada del esquema de la base de datos. En las Figuras 6.3 y 6.4 podemos ver los distintos apartados. Como se puede apreciar en la Figura 6.3, la ventana está dividida en varias zonas:

1. **Zona de visualización del enunciado.** Tal como se aprecia los nombres de los atributos figuran entre paréntesis.
2. **Entrada de nombre de tabla.** El alumno puede dar el nombre que quiera a cada tabla.
3. **Lista desplegable con todos los atributos.** Se puede seleccionar más de un campo a la vez pulsando la tecla de CTRL. De esta forma la entrada de una solución es muy rápida.
4. **Elección de tipo de atributo.** Deberá especificar si los atributos seleccionados forman la clave principal, si son claves foráneas o si son atributos normales.
5. **Aceptación de valores.** Al pulsar este botón se visualizarán los campos entrados en la zona 7 de visualización de las tablas (Figura 6.4)
6. **Botón de corrección.** Envía a corregir la solución entrada.
7. **Zona de visualización de las tablas entradas.** A medida que se entra cada tabla se va visualizando en el espacio que hay entre el enunciado y la zona de introducción de tablas (Figura 6.4).

ALUMNE: ADRIAN ALONSO COZAR



## EXERCICI: 8 de l'activitat MODEL RELACIONAL

Dissenyeu un esquema de base de dades relacional al supòsit desenvolupat a continuació. Utilitzant el model Entitat/Relació o el model Entitat/Relació Estès, cal que definiu les diferents entitats i les relacions entre elles. Les entitats han de tenir **COM A MÍNIM TOTS** els atributs que en l'enunciat estan entre parèntesi. Per les relacions especifiqueu el seu tipus (1:1, 1:N, N:M) i si són parcials o totals. A continuació, proposeu un esquema relacional equivalent al diagrama Entitat/Relació definit, especificant quines són les claus principals i quines les foranes. Finalment cal que des d'aquesta mateixa plana introduïu l'esquema de bases de dades relacional obtingut, seguint les especificacions de l'ACME (taules, clau principal, claus foranes, altres atributs).

Una empresa està organitzada en departaments. De cada departament nos interessa saber su número (numero\_dep), el nombre (nombre\_dep) y el teléfono (tel\_dep).

Cada departamento tiene asignados empleados de los que nos interesa saber su nombre completo (nombre, apellido1, apellido2), NIF (nif), dirección completa (direccion, codigo\_postal, ciudad), sueldo(sueldo), sexo(sexo) y fecha nacimiento(fecha\_nac). El trabajo de cada empleado está supervisado por otro empleado (nif\_super). Uno de los empleados del departamento desempeña la función de director del departamento.

Cada departamento controla un cierto número de proyectos. De cada proyecto nos interesa saber su código (codigo\_proyecto), su nombre (nombre\_proyecto) y la ciudad (ciudad\_proyecto) donde se realiza. Cada proyecto es controlado por un único departamento.

Un empleado puede trabajar en varios proyectos. Nos interesa saber el número de horas (horas) que cada empleado dedica a cada proyecto.

Un empleado puede tener varias personas dependientes que figuran en su cartilla de la seguridad social. De cada una de estas personas nos interesa saber su número dentro de la cartilla (numero), su parentesco (parentesco) con el empleado, su nombre completo ( nombre\_depe, apellido1\_depe, apellido2\_depe) y su fecha de nacimiento (fecha\_nac\_depe).

El sistema diseñado nos tiene que permitir:

- A partir de un departamento saber todos los empleados que tiene asignados, saber quien es su director y los proyectos que controla.
- A partir de un empleado saber sus datos personales, los datos de su supervisor, en que proyectos ha trabajado y cuantas horas ha dedicado a cada proyecto.
- También nos interesa saber las personas dependientes que tiene a su cargo.
- A partir de un proyecto saber los datos de éste, que departamento lo controla, los empleados que están trabajando en él y las horas dedicadas.

Data	Solució Enviada	Resultat
	No s'ha enviat cap solució	

Nom de la taula:

2

Nom del camp:

▲

APELLIDO1,  
APELLIDO1\_DEPE,  
APELLIDO2,  
APELLIDO2\_DEPE,  
CIUDAD,  
CIUDAD\_PROYECTC  
CODIGO\_POSTAL,  
CODIGO\_PROYECTC  
DIRECCION,  
FECHA\_NAC

▼

3

Tipus del camp:

Clau principal

Clau forana

Altres camps

4

Afegir

5

Corregir

6

ALUMNE: ADRIAN ALONSO COZAR



Figura 6.3 Interfaz entrada esquemas BD relacionales.

La interfaz de entrada es muy intuitiva y sólo es necesario asignar un nombre a cada tabla y para cada una de ellas asignar los campos, especificando cuáles son las claves primarias, las claves foráneas y el resto de campos.

La interfaz de recepción de mensajes de ACME es muy simple, sólo se visualiza la solución enviada y una zona para indicar si es correcta o mostrar los mensajes de error si es el caso (véase Figura 6.5). Los mensajes de error que se facilitan son:

- A la solución enviada le faltan tablas
- A la solución enviada le sobran tablas
- Campo clave incorrecto
- La tabla <nombre tabla> es incorrecta

- A partir de un proyecto saber los datos de éste, que departamento lo controla, los empleados que están trabajando en él y las horas dedicadas.

Data	Solució Enviada	Resultat
	No s'ha enviat cap solució	

<b>Nom de la taula:</b>	EMPLEADO
<b>Clau principal:</b>	NIF,
<b>Clau forana:</b>	NIF_SUPER,, NUMERO_DEP,
<b>Altres camps:</b>	APELLIDO1,, APELLIDO2,, CIUDAD,, CODIGO_POSTAL,, DIRECCION,, FECHA_NAC, NOMBRE,, SEXO,, SUELDO,,

<b>Nom de la taula:</b>	DEPARTAMENTO
<b>Clau principal:</b>	NUMERO_DEP,
<b>Clau forana:</b>	NIF,,
<b>Altres camps:</b>	NOMBRE_DEP,, TEL_DEP,

<b>Nom de la taula:</b>	PROYECTO
<b>Clau principal:</b>	CODIGO_PROYECTO,
<b>Clau forana:</b>	NUMERO_DEP,
<b>Altres camps:</b>	CIUDAD_PROYECTO, NOMBRE_PROYECTO,,

<b>Nom de la taula:</b>	TRABAJA
<b>Clau principal:</b>	CODIGO_PROYECTO, NIF,
<b>Clau forana:</b>	CODIGO_PROYECTO, NIF,
<b>Altres camps:</b>	HORAS,

<b>Nom de la taula:</b>	<b>Nom del camp:</b>	<b>Tipus del camp:</b>
DEPENDIENTE	<input type="text"/> APELLIDO1, APELLIDO1_DEPE, APELLIDOS, APELLIDO2_DEPE, CIUDAD, CIUDAD_PROYECT, CODIGO_POSTAL, CODIGO_PROYECT, DIRECCION, FECHA_NAC	<input type="radio"/> Clau principal <input type="radio"/> Clau forana <input checked="" type="radio"/> Altres camps

**Afegir**

**Corregir**

Figura 6.4 Visualización de tablas entradas.

Estás a > ACME > Inid > Bases de dades > MODEL RELACIONAL > Exerciđ: 10

La solució enviada és:

<b>Nom de la taula:</b>	EMPLEADO
<b>Clau principal:</b>	NIF,
<b>Clau forana:</b>	NUMERO_DEP,
<b>Altres camps:</b>	APELLIDO1, APELLIDO2, CIUDAD, CODIGO_POSTAL, DIRECCION, FECHA_NAC, NOMBRE, SEXO, SUELDO,

<b>Nom de la taula:</b>	DEPARTAMENTO
<b>Clau principal:</b>	NUMERO_DEP,
<b>Clau forana:</b>	NIF,
<b>Altres camps:</b>	NOMBRE_DEP, TEL_DEP,

<b>Nom de la taula:</b>	PROYECTO
<b>Clau principal:</b>	CODIGO_PROYECTO,
<b>Clau forana:</b>	NUMERO_DEP,
<b>Altres camps:</b>	CIUDAD_PROYECTO, NOMBRE_PROYECTO,

<b>Nom de la taula:</b>	TRABAJA
<b>Clau principal:</b>	CODIGO_PROYECTO, NIF,
<b>Clau forana:</b>	CODIGO_PROYECTO, NIF,
<b>Altres camps:</b>	HORAS,

El resultat de la correcció és:

A la solució enviada li falten taules.  
La taula EMPLEADO és incorrecta; repassa-la.

**Continuar**

Figura 6.5 Visualización mensajes de error ACME-EBD.

## 6.5.5 El módulo corrector ACME-EBD

El siguiente punto a tratar es el diseño e implementación del módulo corrector. Cuando el alumno envía una solución a corregir, ésta se guarda con la misma estructura que las soluciones detalladas en la sección 6.5.3. De esta forma, en posteriores consultas siempre mostramos la información completa enviada por el alumno. Esto también permite enviar una nueva solución a partir de la anterior. Este hecho se da muy a menudo ya que el alumno se suele equivocar en pocas cosas y con pequeñas modificaciones dispone ya de una nueva solución lista para ser enviada.

La estrategia de corrección se basa en utilizar los nombres de atributos que figuran entre paréntesis en el enunciado del problema. Como que el alumno tiene que utilizar estos nombres, identificamos las tablas de la solución del alumno según el campo clave utilizado. En caso de que el alumno asigne un campo no clave como campo clave de una tabla, ésta será considerada incorrecta.

El objetivo de este módulo corrector es comprobar si hay alguna solución correcta asociada a este ejercicio que sea igual a la enviada por el alumno. En este caso se envía el mensaje de solución correcta al alumno. Si la solución del alumno no es correcta hay que determinar cuál de las soluciones correctas almacenadas en el repositorio es la más parecida a la enviada. Tomando como referencia la solución más parecida se envían mensajes al alumno que le ayudarán a detectar sus errores. Así pues este módulo está formado por dos apartados: verificación de la solución y retorno de feedback.

### Verificación de la solución.

Este apartado tiene como objetivo detectar si la solución es correcta y en caso contrario retornar la solución correcta más parecida a la del alumno. A partir del resultado de este apartado se retornará el feedback al alumno.

Tal como se muestra en la Figura 6.6, se van accediendo a las diferentes soluciones correctas descritas en el fichero del problema y se van comparando con la enviada por el alumno. Las tablas del alumno se van identificando según el campo clave utilizado. En caso de que todas las tablas de la solución enviada sean iguales a las de una solución del repositorio, se envía el mensaje de “correcto” y finaliza el proceso de corrección. En caso contrario, se guarda el grado de diferencia entre la solución enviada por el alumno y cada una de las correctas. Para la determinación de las diferencias se dispone de procesos de verificación que van comprobando las diferencias entre las tablas, sus claves (primarias y foráneas) y el resto de campos. Al final del proceso se escoge la que el grado de diferencia es menor como la solución correcta más parecida a la enviada por el alumno y respecto de la cual se enviarán los mensajes de error al alumno.

Los procesos de revisión de tablas y de revisión de campos utilizan la estrategia siguiente:

**Proceso de revisión del número de tablas.** Se accede a él cuando el número de tablas de la solución es distinto a la del alumno. Por cada tabla de más o de menos se suma a *diferencia* el valor parametrizable  $V_1$ .

**Proceso de revisión de campos.** Para cada una de las tablas de la solución enviada por el alumno se realizan las siguientes verificaciones:

- Identificar el campo clave primaria enviado por el alumno. Si no coincide con ninguna de las tablas de la solución que se compara se suma a *diferencia* el valor parametrizado  $V_2$ .
- Si hemos podido identificar la tabla (clave primaria correcta) se comprueba que tanto las claves foráneas como los otros atributos sean correctos. En caso de que los campos clave foráneas sean incorrectos, para cada uno de incorrecto se suma el valor  $V_3$  y en caso de otros atributos incorrectos, para cada uno que haya incorrecto se suma el valor  $V_4$ .

Los valores asignados a los distintos  $V_i$  se han fijado después de varias pruebas y ajustes. Son valores diferentes en función de la relevancia de cada uno de ellos. De esta forma cuando más diferente sea la solución enviada por el alumno respecto a la solución correcta comparada, mayor será el valor de la *diferencia*. Seleccionamos la solución con valor de la *diferencia* menor, entendiendo que es la solución correcta más parecida a la del alumno. Esta solución será la que utilizaremos en la siguiente fase.

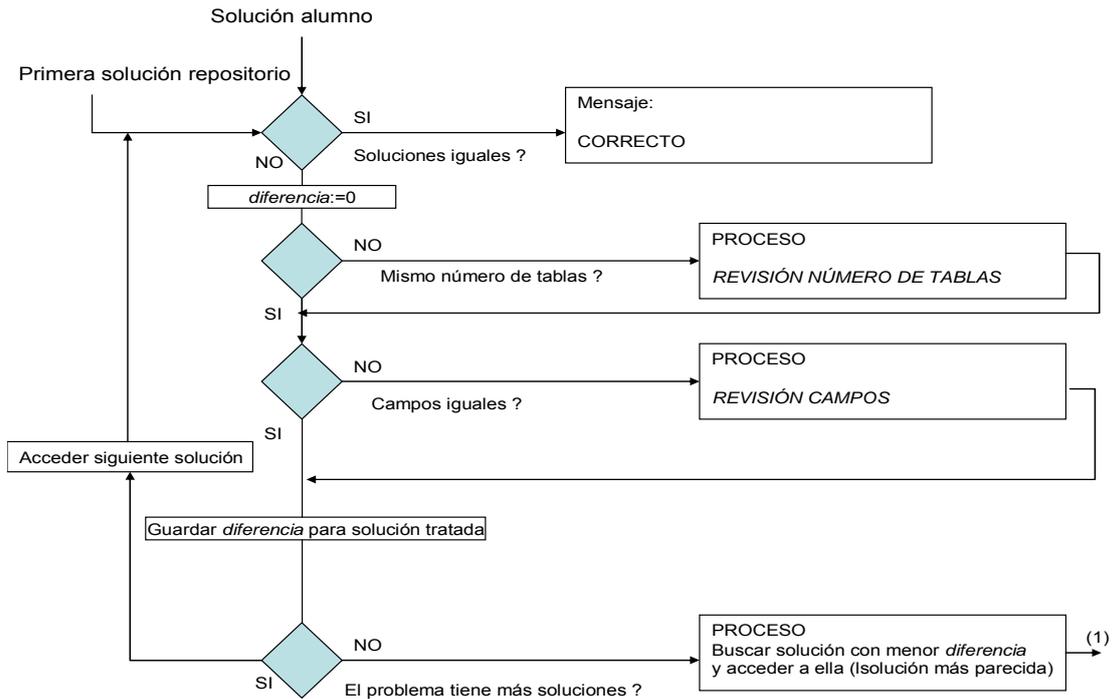


Figura 6.6 Proceso de verificación de la solución en ACME-EBD.

### Retorno de feedback.

Una vez detectada la solución correcta más parecida a la enviada por el alumno tenemos que enviarle los mensajes de error con respecto a ésta, que le conduzcan a detectar los errores y le faciliten enviar una nueva solución revisada. Los mensajes de error no deben ser muy detallados ya que no se pretende dar la solución correcta ni decirle exactamente los errores cometidos, sino facilitar mensajes que le sirvan de orientación. En la Figura 6.7 se muestra la estrategia seguida y el mensaje de error que se visualiza.

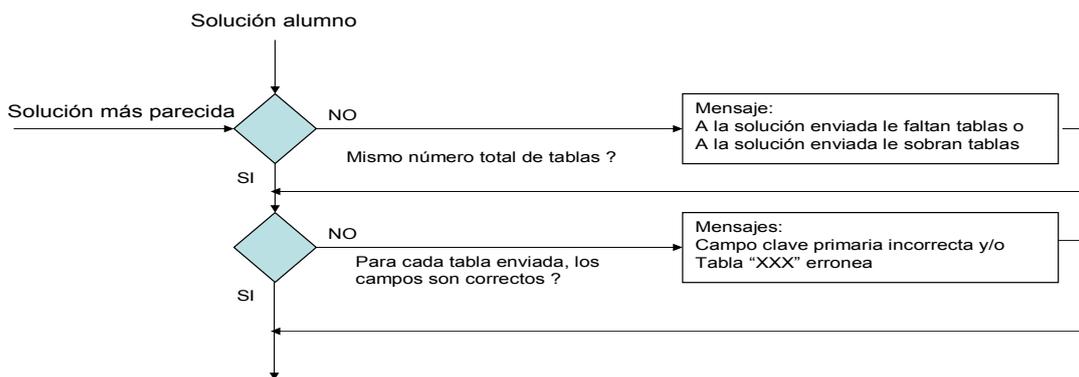


Figura 6.7 Proceso de retorno de feedback en ACME-EBD.

En caso de no coincidir el número de tablas del alumno con la solución más parecida se da el mensaje de que faltan o sobran tablas. En caso de que no se puede identificar la tabla a partir de su campo clave este se da el mensaje de “campo clave incorrecto” y en caso de que las claves foráneas o los otros atributos no coinciden, el mensaje será de “Tabla <nombre tabla> errónea”.

## 6.5.6 Puntuación automática de los esquemas de bases de datos

Tal como se ha comentado en la sección 2.2.7 un punto importante en los sistemas CBA es el referente al sistema de puntuación. En muchos casos, especialmente para la realización de pruebas de validación/exámenes, nos interesa que el sistema nos facilite una nota del ejercicio realizado por el alumno. Es habitual que en un esquema de BD muy grande el alumno cometa pequeños fallos que hacen que sea incorrecto, aunque sería injusta una mala nota. En la puntuación se tienen en cuenta: que se hayan identificado correctamente las tablas, las claves foráneas y de los atributos.

Para la puntuación automática de los esquemas de bases de datos relacionales hemos utilizado el mismo sistema que se ha comentado en la sección 4.5.6, consistente en asignar unos pesos  $P_i$  a cada grupo de elementos del esquema, de forma que con la suma total de ellos se obtenga la puntuación final. La nota final se obtiene a partir de

$$Nota = 10 - \sum_{i=1}^n (e_i * ne_i * cp_i) \text{ y } Nota \geq 0$$

Donde  $e_i$  representa el valor de cada elemento del grupo  $i$  y  $ne_i$  representa el número de errores cometidos en el grupo  $i$  respecto a la solución correcta más parecida a la enviada por el alumno y que el sistema ha detectado en el proceso de corrección automática. El valor  $cp_i$  representa el coeficiente penalizador que se aplica a cada error del grupo  $i$  y que debe calcularse para ajustar la nota calculada automáticamente con la manual. La metodología usada también ha sido la misma. En este caso se han seleccionado tres problemas de entre 6 y 9 tablas, entre 5 y 8 claves foráneas y con más de veinte atributos. De cada problema se han seleccionado 13 esquemas significativos. Los 39 problemas han sido corregidos de forma manual por tres profesores y con la nota media obtenida, con los valores  $e_i$  y el número de errores cometidos  $ne_i$  se han calculado los  $cp_i$ . Comentar que en las puntuaciones de los profesores había hasta un máximo de dos puntos de diferencia en un mismo ejercicio.

En primera instancia se pensó en tres grupos de elementos que representarían a las tablas, a las claves foráneas y al resto de atributos. Los dos primeros grupos tendrían un peso de 4 puntos y el tercer grupo de 2 puntos. Cada identificación incorrecta de cada uno de estos elementos sería penalizado con un error. A partir de aquí se calcularon los valores  $cp_i$  que debían minimizar la diferencia con las correcciones manuales. Al igual que con la corrección de ERD, con este sistema los resultados que se obtuvieron no fueron buenos. Habían diferencias significativas entre las calculadas manualmente y las calculadas de forma automática. Las notas calculadas automáticamente variaban respecto a la nota media manual desde -1,84 hasta 1,72 puntos. Consideramos que la imprecisión era demasiada y desestimamos el método.

Se pensó en reducir los tres grupos, pasando a dos grupos que representarían a las tablas (con todos sus atributos, excepto los que eran claves foráneas) y a las claves foráneas. El primer grupo tendría un peso de 6 puntos y el segundo de 4 puntos. El número de errores se contabilizaba igual que en el sistema anterior, con la excepción de que si una tabla sólo tenía atributos no claves erróneos se penalizaba con 0,5 errores. Los resultados obtenidos fueron parecidos a los anteriores, variando de -1,96 a 1,82 puntos. Por consiguiente también descartamos este método.

Finalmente se pasó a considerar un solo grupo de elementos: las tablas. Evidentemente el peso que tendría este grupo sería de 10 puntos y los criterios para contabilizar errores han sido:

- Se contabiliza un error por cada tabla errónea. Una tabla es errónea si sobra o falta o está mal identificada (atributo clave principal erróneo).
- Se contabiliza 0,75 errores por cada tabla identificada correctamente pero con claves foráneas erróneas.
- Se contabiliza 0,5 errores por cada tabla identificada correctamente pero con errores en atributos no clave.
- Se contabiliza un error por cada tabla identificada correctamente pero con errores en las claves foráneas y en el resto de los atributos.

Al considerar un solo grupo la nota final se obtendrá a partir de  $Nota = 10 - cp_t * (e_t * ne_t)$  con  $e_t$  (valor de una tabla) y  $ne_t$  (número de errores en las tablas). Para el cálculo de  $cp_t$  se ha utilizado un modelo de regresión que nos ha dado un valor de 1,28 para  $cp_t$ . Este valor es significativo ya que el p-valor es 0,00. El coeficiente de determinación  $R^2$  es de 0,972 con lo cual el modelo ofrece una buena calidad

En la Figura 6.8 puede verse en primer lugar (gráfica Normal Probability Plot) que el modelo obtenido se ajusta bastante bien a la normalidad. Si observamos los valores de los residuos en las otras tres gráficas vemos que en ningún caso hay una diferencia superior/inferior a los 1,3 puntos respecto a las notas calculadas manualmente. En el histograma de frecuencias podemos ver que la mayor parte de las notas obtenidos con el modelo están en un rango más/menos 0,7 puntos.

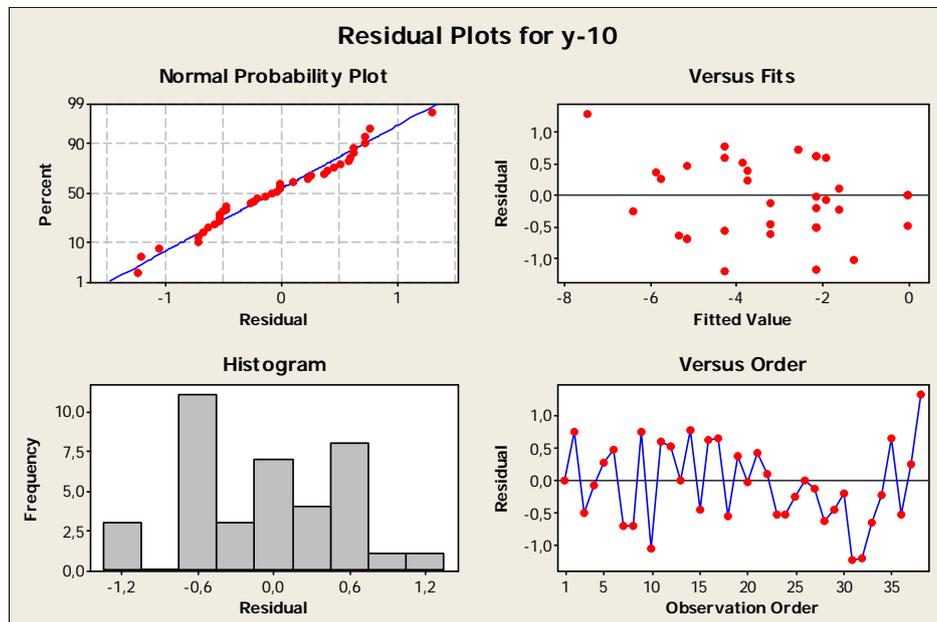


Figura 6.8 Gráficas del modelo obtenido.

Para verificar que el modelo se ajusta a la normalidad (Figura 6.9) se ha realizado un contraste de normalidad obteniendo un p-valor de 0,368 que nos lleva a aceptar la normalidad.

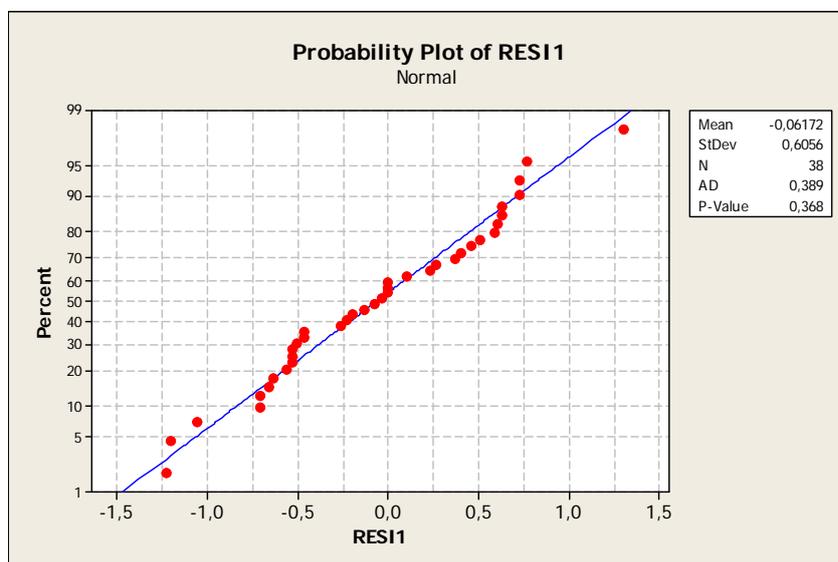


Figura 6.9 Contraste de normalidad.

Comentar que al igual que en la puntuación de ERD, se han omitido los casos con una nota media inferior a 3 ya que la puntuación del profesor no refleja con exactitud los errores existentes. En estos casos, las notas calculadas según el modelo eran siempre bastante más inferiores a la media de la corrección manual. No consideramos importante esta omisión ya que para notas inferiores a 3, el sistema responderá simplemente “prueba no superada”. Por otra parte vemos que no considerando estos valores el modelo se ajusta muy bien al resto de casos que son los que realmente nos interesa evaluar.

## 6.6 EVALUACIÓN DEL MÓDULO

En esta sección se comenta la preparación de los ejercicios y a continuación se detallan las actividades realizadas y los resultados obtenidos.

### 6.6.1 Preparación de ejercicios de EBD relacionales

Antes de poder utilizar el módulo y poder asignar problemas a los alumnos era necesario el desarrollo de una amplia colección de problemas de forma que, en la medida de lo posible, fueran distintos los ejercicios de cada cuaderno o por lo menos que no fueran exactamente los mismos. Para ello se han reutilizado los mismos problemas preparados para el modelo ER. A cada problema simplemente se le han añadido las soluciones correspondientes a los esquemas de bases de datos que podemos aceptar como válidas. Los mismos grupos de problemas ahora están formados por:

- **Grupo 1.** 50 problemas de inicio, de poca dificultad. Son problemas con soluciones entre 3 y 5 tablas
- **Grupo 2.** 50 problemas de nivel medio con un poco más de dificultad. Su solución tiene entre 6 y 9 tablas.
- **Grupo 3.** 50 problemas más complicados en los que su solución tiene entre 10 y 18 tablas.

Cada problema se ha resuelto y se han considerado las posibles soluciones válidas. Como se comentó en la sección 4.6.1, se ha intentado que los enunciados sean claros, concisos y que no conduzcan a confusiones innecesarias. Con ello se ha conseguido que las posibles soluciones estuvieran muy acotadas. El primer grupo de problemas suele tener muy pocas soluciones posibles, mientras que los del tercer grupo suelen tener varias soluciones.

Una vez escrito cada problema el sistema verifica que su sintaxis fuese la establecida para este tipo de problema. Una vez la sintaxis es la correcta se testea el problema para que no haya ningún error en la/s solución/es. Como se puede apreciar el tiempo dedicado a la creación y testeo de estos problemas es considerable, tal como se ha comentado en la sección referente a los inconvenientes de los sistemas CBA. Pero hay que remarcar que una vez verificado el sistema y los ejercicios preparados, se dispone de un entorno y de una colección de problemas para ser utilizado siempre que queramos, sin ningún esfuerzo adicional.

### 6.6.2 Actividades realizadas y resultados obtenidos

Este entorno corrector de esquemas de bases de datos relacionales se ha utilizado principalmente como complemento a las clases presenciales (Blended learning) durante los cinco últimos cursos en la asignatura de Bases de Datos de la carrera de Ingeniería Técnica en Informática de Gestión (ITIG). Tal como se ha comentado en el capítulo 4, en primer lugar el alumno aprende diseño conceptual mediante el modelo ER y a continuación el modelo relacional. Una vez se ha explicado el modelo relacional en las sesiones teóricas y se ha mostrado como pasar un ERD a un esquema de bases de datos relacional, se asignan varios problemas a través de la plataforma ACME con el objetivo de evaluar si el alumno sabe realizar la transformación. Los niveles cognitivos que pretendemos evaluar corresponden a los de aplicación y de análisis de la taxonomía de Bloom. El profesor establece el número de problemas que deberá resolver cada alumno y la valoración de cada uno de ellos. A continuación escoge del repositorio los problemas a ser asignados al cuaderno de cada uno de los alumnos. Generalmente se eligen problemas de poca dificultad (del Grupo 1) para ser asignados a los primeros problemas de cada tema, problemas de dificultad media (del Grupo 2) para los problemas intermedios de cada tema y finalmente se eligen problemas ya más extensos y difíciles (del Grupo 3) que serán asignados a los últimos problemas de cada tema. Para evitar el plagio, el profesor selecciona todos los problemas de un determinado tipo de dificultad y el sistema, de forma aleatoria los distribuye a los cuadernos de los alumnos. De esta forma, aunque haya alguna repetición, la mayoría de ejercicios de un alumno son distintos de los de sus compañeros, aunque el nivel de dificultad es muy parecido. La realización de estos ejercicios se tiene en cuenta para la valoración final de la asignatura. Para que los alumnos trabajen de forma precisa en la realización de los esquemas se les insiste que tendrán penalizaciones en función del número de intentos

hasta llegar a la solución correcta. Los alumnos disponen de un manual de usuario de ACME-EBD, aunque debido a la sencillez de uso, apenas lo consultan. Se insiste a los alumnos en la importancia del trabajo continuado.

Los datos de utilización de este entorno están reflejados en la Tabla 6.1. Para cada curso se muestra el número de alumnos, los problemas asignados, el % de ejercicios que han sido leídos, el % de ejercicios a los cuales se han enviado soluciones y el % que han sido resueltos correctamente. Finalmente en las cuatro últimas columnas se especifica el % de intentos con los que se ha alcanzado la solución correcta (1, 2, 3 o más intentos).

CURSO	ALUMNOS	PROBLEMAS	% EJERCICIOS LEIDOS	% SOLUCIONES ENVIADAS	% SOLUCIONES CORRECTAS	% INTENTOS PARA LLEGAR A LA SOLUCIÓN CORRECTA			
						1	2	3	>3
2005/06	61	6	92,5%	84,6%	70,5%	33,0%	30,2%	16,2%	20,7%
2006/07	50	8	92,8%	84,4%	74,4%	46,8%	27,4%	12,9%	12,9%
2007/08	46	8	96,0%	90,9%	75,4%	36,5%	33,2%	13,9%	16,3%
2008/09	36	8	86,2%	82,4%	78,1%	43,3%	28,7%	13,4%	14,6%
2009/10	32	7	92,4%	83,9%	77,6%	40,2%	32,1%	9,7%	18,0%

Tabla 6.1 Utilización del entorno ACME-EBD.

Como se puede apreciar en esta tabla alrededor del 90% de los problemas son leídos y se envían soluciones a más de un 80% de ellos. Alrededor de un 80% de los problemas son resueltos de forma correcta y cerca del 80% de los casos son resueltos en los tres primeros intentos. Por otra parte si descartamos los ejercicios asignados que ni tan siquiera han sido leídos por el alumno (alrededor de un 10% y que hacen referencia a alumnos matriculados pero que no asisten a clase regularmente o que abandonan la asignatura) y sólo nos fijamos en las soluciones enviadas por los alumnos realmente interesados vemos (Tabla 6.2) que prácticamente siempre envían las soluciones (a más del 90% de los problemas) y entre un 75% y un 90% de los casos obtienen la solución correcta.

CURSO	ALUMNOS	PROBLEMAS	% EJERCICIOS LEIDOS	% SOLUCIONES ENVIADAS	% SOLUCIONES ENVIADAS/LEIDAS	% SOLUCIONES CORRECTAS/LEIDAS	
						% SOLUCIONES CORRECTAS	% SOLUCIONES CORRECTAS/LEIDAS
2005/06	61	6	92,5%	84,6%	91,5%	70,5%	76,3%
2006/07	50	8	92,8%	84,4%	90,9%	74,4%	80,2%
2007/08	46	8	96,0%	90,9%	94,7%	75,4%	78,5%
2008/09	36	8	86,2%	82,4%	95,6%	78,1%	90,6%
2009/10	32	7	92,4%	83,9%	90,8%	77,6%	84,0%

Tabla 6.2 Valoración sobre los ejercicios ACME-EBD leídos.

Por otra parte para tener información de los tipos de errores que de forma más frecuente realizan los alumnos y para ver las dificultades que tienen hasta llegar a la solución correcta, hemos evaluado la tipología de errores cometidos en los diferentes intentos. Los datos mostrados en la Tabla 6.3 hacen referencia al último curso y observamos que en el caso de problemas sencillos del Grupo 1 (hasta cinco tablas) al tercer intento ya prácticamente sólo tienen una tabla incorrecta.

<b>EJERCICIOS GRUPO 1. ENTRE 3 Y 5 TABLAS</b>				
<b>TIPO DE ERROR</b>	<b>1ER INTENTO</b>	<b>2O INTENTO</b>	<b>3ER INTENTO</b>	<b>MÁS DE 3 INT.</b>
1 TABLA INCORRECTA	55,3%	69,2%	87,5%	100,0%
2 TABLAS INCORRECTAS	21,1%	23,1%	12,5%	
3 TABLAS INCORRECTAS	5,3%	7,7%		
FALTAN TABLAS	2,6%			
SOBRAN TABLAS	2,6%			
1 TABLA INCORRECTA Y FALTAN TABLAS	2,6%			
2 TABLAS INCORRECTAS Y FALTAN TABLAS	2,6%			
1 TABLA INCORRECTA Y SOBRAN TABLAS	5,3%			
2 TABLAS INCORRECTAS Y SOBRAN TABLAS	2,6%			

Tabla 6.3 Tipología de errores. Problemas Grupo 1.

En el caso de problemas del Grupo 2 (entre seis y hasta nueve tablas) tal como muestra la Tabla 6.4, los alumnos que necesitan más de 3 intentos ya prácticamente sólo les falta arreglar una tabla para alcanzar la solución correcta. Por otra vemos que a partir de los primeros intentos ya solamente les falta corregir una o dos tablas.

<b>EJERCICIOS GRUPO 2. ENTRE 6 Y 9 TABLAS</b>				
<b>TIPO DE ERROR</b>	<b>1ER INTENTO</b>	<b>2O INTENTO</b>	<b>3ER INTENTO</b>	<b>MÁS DE 3 INT.</b>
1 TABLA INCORRECTA	31,7%	40,9%	63,6%	83,3%
2 TABLAS INCORRECTAS	7,3%	18,2%	27,3%	16,7%
3 TABLAS INCORRECTAS	7,3%	4,5%		
MÁS DE TRES TABLAS INCORRECTAS	9,8%	4,5%		
1 TABLA INCORRECTA Y FALTAN TABLAS	9,8%	4,5%	9,1%	
2 TABLAS INCORRECTAS Y FALTAN TABLAS	2,4%	4,5%		
3 TABLAS INCORRECTAS Y FALTAN TABLAS	4,9%	9,1%		
MÁS DE TRES TABLAS INCORRECTAS Y FALTAN TABLAS	4,9%	4,5%		
1 TABLA INCORRECTA Y SOBRAN TABLAS	4,9%			
2 TABLAS INCORRECTAS Y SOBRAN TABLAS	2,4%			
3 TABLAS INCORRECTAS Y SOBRAN TABLAS	4,9%			
MÁS DE TRES TABLAS INCORRECTAS Y SOBRAN TABLAS	4,9%			
FALTAN TABLAS	2,4%	9,1%		
SOBRAN TABLAS	2,4%			

Tabla 6.4 Tipología de errores. Problemas Grupo 2.

En el caso de problemas del Grupo 3 (entre diez y hasta dieciocho tablas), y tal como muestra la Tabla 6.5, los resultados obtenidos son muy parecidos a los obtenidos con los problemas del Grupo 2. A partir del 3 intento ya prácticamente tienen resuelto el problema y solamente les queda una tabla por corregir. Esto nos hace pensar que el sistema de feedback es adecuado y que realmente ayuda a los alumnos

<b>EJERCICIOS GRUPO 3. ENTRE 10 Y 18 TABLAS</b>				
<b>TIPO DE ERROR</b>	<b>1ER INTENTO</b>	<b>2O INTENTO</b>	<b>3ER INTENTO</b>	<b>MÁS DE 3 INT.</b>
1 TABLA INCORRECTA	16,2%	37,0%	69,7%	82,6%
2 TABLAS INCORRECTAS	9,5%	19,6%	15,2%	17,4%
3 TABLAS INCORRECTAS	13,5%	4,3%	3,0%	
MÁS DE TRES TABLAS INCORRECTAS	13,5%	8,7%		
1 TABLA INCORRECTA Y FALTAN TABLAS	13,5%	10,9%	6,1%	
2 TABLAS INCORRECTAS Y FALTAN TABLAS	5,4%	4,3%	3,0%	
3 TABLAS INCORRECTAS Y FALTAN TABLAS	4,1%	2,2%		
MÁS DE TRES TABLAS INCORRECTAS Y FALTAN TABLAS	2,7%			
1 TABLA INCORRECTA Y SOBRAN TABLAS	1,4%	2,2%	3,0%	
2 TABLAS INCORRECTAS Y SOBRAN TABLAS	4,1%	4,3%		
3 TABLAS INCORRECTAS Y SOBRAN TABLAS	5,4%	2,2%		
MÁS DE TRES TABLAS INCORRECTAS Y SOBRAN TABLAS	8,1%	2,2%		
FALTAN TABLAS	1,4%	2,2%		
SOBRAN TABLAS	1,4%			

Tabla 6.5 Tipología de errores. Problemas Grupo 3.

Comentar que de cara a la evaluación formativa y más concretamente al seguimiento del trabajo del alumno, el entorno facilita considerablemente las tareas del profesor (véase Figura 4.7). Los comentarios respecto a este tema realizados en la sección 4.6.2 son perfectamente válidos para esta sección.

Finalmente comentar que este módulo también ha sido utilizado en la experiencia llevada a cabo en los dos últimos cursos para desarrollar el diseño de una base de datos extensa referente al funcionamiento de una empresa (se ha comentado en la sección 4.6.2). El alumno debía realizar el ERD correspondiente a cada una de las cinco partes en que dividimos el supuesto y a continuación realizar el esquema correspondiente a través del módulo ACME-EBD.

En el capítulo 10 se realiza una valoración más exhaustiva de las mejoras en el aprendizaje del diseño de bases de datos que comporta la utilización de ACME-EBD, juntamente con el entorno ACME-ER y ACNE-NOR.

## 6.7 APORTACIONES Y CONCLUSIONES

Se ha desarrollado el entorno ACME-EBD que es un entorno CBA para la corrección/evaluación de esquemas de bases de datos relacionales. Se ha integrado a la plataforma ACME y se ha creado una extensa colección de problemas que se están utilizando en la asignatura de “Bases de datos” desde el curso 2005-2006. Las principales ventajas que presenta este entorno son:

- Permite llevar a cabo tanto la evaluación formativa como sumativa en el proceso de transformación de un ERD a un esquema de bases de BD relacional. Actúa como un sistema CBA.
- Disminución del trabajo del profesor en cuanto a horas dedicadas a la corrección de este tipo de ejercicios, aunque el esfuerzo de la puesta a punto y diseño de ejercicios es considerable.
- Muchos más alumnos se ejercitan en el diseño lógico de bases de datos y obtienen un resultado inmediato de la corrección.
- Mayor feedback alumno/profesor. Se aclaran muchas dudas que surgen en el momento de la resolución de los ejercicios.
- Permite un seguimiento muy estricto del trabajo del alumno.
- Sistema de puntuación automático que obtiene notas muy parecidas a las manuales.
- Evaluación automática del trabajo realizado por el alumno.

La única limitación a considerar es que el alumno no puede asignar nombres a los atributos y debe escoger los marcados en el enunciado del problema.

El trabajo realizado en el desarrollo de este corrector ha dado lugar al artículo “**An Automatic Correction Tool for relational Database Schemas**” presentado en la International Conference on Information Technology based Higher Education and Training, ITHET 2005, S3C pp 9-14, 2005, referenciado en [PBS+05b].

La utilización de parte de ACME-DB como un complemento a las clases presenciales en la docencia de bases de datos ha sido presentada en las XII Jornadas de la Enseñanza Universitaria de la Informática (JENUI) en el 2006 en el artículo “**Utilización de una plataforma de e-learning en la docencia de bases de datos**”, referenciado por [SPB+06b].

En el artículo “**Database design using a web-based e-learning tool**” presentado en el 2nd Workshop on Methods and Cases in Computing Education 2009 (MCCE) se expone la experiencia en la utilización de ACME-DB en el aprendizaje de diseño de bases de datos. Este artículo está referenciado en [SBP+09a].

El detalle de la experiencia llevada a cabo en el diseño de una base de datos extensa a través de la plataforma ACME-DB y los resultados obtenidos están disponibles en la ponencia “**Experiencia docente en diseño de bases de datos con la ayuda de herramientas de e-learning**” ha sido presentada en las XV Jornadas de la Enseñanza Universitaria de la Informática (JENUI) en el 2009, referenciado en [SBP+09b].



## **CAPÍTULO 7. MÓDULO PARA LA NORMALIZACIÓN DE BASES DE DATOS.**

En este capítulo presentaremos el módulo de normalización de bases de datos. Se describe como se ha utilizado el módulo del capítulo anterior para la realización de ejercicios de normalización de bases de datos relacionales.

Este capítulo se ha estructurado de la forma siguiente. En la sección 7.1 se hace una introducción al tema. En la sección 7.2 se describirá brevemente la teoría de la normalización y a continuación en la sección 7.3 se detallan las necesidades y la problemática en su enseñanza. En la sección 7.4 se hace un breve resumen de los entornos para el aprendizaje de la normalización. Posteriormente en la sección 7.5 plantearemos el problema que nos proponemos resolver, describiremos detalladamente la solución aportada y su integración a la plataforma ACME. En la sección 7.6 se realiza una evaluación del módulo desarrollado, presentando las actividades y resultados obtenidos. Finalmente en la sección 7.7 se describen las conclusiones a las que hemos llegado.

### **7.1 INTRODUCCIÓN**

La normalización de bases de datos fue introducida por E.Codd [Cod71] después de crear el modelo relacional [Cod70]. La normalización pretende añadir calidad a los diseños de BD que siguen el modelo relacional, evitando la aparición de datos redundantes y los problemas de actualización que conllevan. Tal como hemos visto en el capítulo anterior, para un mismo entorno se pueden diseñar distintos esquemas de bases de datos para representar su información. Algunos de ellos pueden ser aceptables y correctos para su implementación y otros no. La teoría de la normalización de bases de datos relacionales establece las pautas para determinar por qué unos esquemas son mejores que otros. Esta teoría establece una serie de pautas, llamadas formas normales, que deben cumplir las relaciones. La normalización de las relaciones (tablas) de una BD supone comprobar las formas normales y en caso que alguna relación no cumpla alguna de ellas, se descompondrá en otras relaciones que sí las cumplan, hasta obtener al final un esquema normalizado. Hoy en día, y tras casi 40 años desde su aparición, el modelo relacional es ampliamente usado, aunque no podemos afirmar lo mismo de la normalización.

El objetivo final de la enseñanza de la normalización es que el alumno debe ser capaz de normalizar hasta una determinada forma normal un esquema de bases de datos. La docencia de esta materia, tal como propone [Dat04], se centra en las tres primeras formas normales y como mucho hasta la forma normal de Boyce-Codd. No se suele tratar ni la cuarta ni la quinta forma normal por tener menos importancia y tratarse de situaciones muy poco frecuentes. Para conseguir este objetivo, es necesario que el alumno adquiriera conocimientos de:

- Directrices de diseño de esquemas de relación.
- Dependencias funcionales. Reglas de inferencia para las dependencias funcionales.
- Formas normales basadas en las dependencias funcionales.

Aunque todos estos conceptos son importantes, pretendemos dar un enfoque muy práctico a este tema valorando especialmente que el alumno sepa normalizar hasta forma normal de Boyce-Codd. Para ello centramos el tema en la comprensión de los conceptos básicos, las dependencias funcionales, las formas normales y en aplicar estos conceptos normalizando un esquema de base de datos. La tipología más habitual de ejercicio a realizar por el alumno consiste en facilitar un esquema de bases de datos y un conjunto de dependencias funcionales. A partir de ahí, el alumno debe analizar si se cumplen las formas normales y obtener un esquema normalizado. No damos tanta importancia a los formalismos más teóricos de la teoría de la normalización.

El resultado final del proceso de normalización es la obtención de un conjunto de tablas normalizadas, por lo tanto consideramos oportuno reutilizar el módulo de esquemas de bases de datos (véase capítulo 6) para plantear ejercicios de normalización. Ahora el objetivo no sería la obtención del esquema de bases de

datos a partir de un supuesto dado, sino que a partir de un esquema no normalizado y un conjunto de dependencias funcionales, el alumno obtenga como solución un conjunto de tablas normalizadas que entraría al sistema ACME de la forma vista en el capítulo anterior. De esta forma reutilizamos el módulo para la actividad de normalización. Aunque utilizamos el sistema para la normalización hasta la forma normal de Boyce-Codd también se puede utilizar para la cuarta y la quinta forma normal.

Finalmente comentar que otras preguntas o ejercicios del tema de normalización también se pueden formular a través de las típicas preguntas de elección múltiple. Evidentemente ACME facilita también la corrección de este tipo de ejercicios.

## **7.2 NORMALIZACIÓN DE BD RELACIONALES**

La técnica que hemos explicado para obtener esquemas de bases de datos relacionales, consistente en realizar el diagrama entidad/relación y a continuación transformarlo al modelo relacional, nos lleva generalmente a diseñar buenos esquemas de bases de datos.

Por otro lado, para un mismo supuesto se pueden diseñar distintos esquemas de bases de datos, todos ellos más o menos aceptables. La teoría de la normalización se ha desarrollado para medir formalmente la calidad de un esquema de una base de datos verificando si cumple las formas normales. El proceso de normalización consiste en someter a una tabla a una serie de pruebas para comprobar si está en una determinada forma normal. Codd [Cod71] propuso tres formas normales (conocidas respectivamente como la primera, la segunda y la tercera). Más adelante Boyce y Codd [Hea71] propusieron una definición más estricta de la tercera forma normal: la llamada forma normal de Boyce-Codd. Posteriormente se desarrollaron una cuarta y quinta forma normal, basadas en los conceptos de dependencias multi-valoradas y dependencias de reunión.

La normalización consiste en un proceso en que los esquemas de relación (tabla) que no cumplen una forma normal, se descomponen en esquemas de relación más pequeños que sí las cumplan. Un esquema de relación se dice que está en una determinada forma normal si cumple un conjunto de restricciones. No es necesario que los esquemas de bases de datos estén normalizados hasta la quinta forma normal. A menudo se comprueban sólo las tres primeras formas normales. En general, el diseño de bases de datos a partir del modelo Entidad/Relación y su transformación al modelo relacional, nos suele llevar a esquemas de Bases de datos que ya cumplen las tres primeras formas normales. Estas tres primeras formas normales y la forma normal de Boyce-Codd se basan en el concepto de dependencia funcional.

Una dependencia funcional es una restricción entre dos conjuntos de atributos de la bases de datos. Dado un esquema de base de datos formado por un conjunto de esquemas de relación  $EBD=\{R_1,R_2,\dots,R_m\}$  supongamos que en total hayan  $n$  atributos  $A_1,A_2,\dots,A_n$ . Para facilitar el aprendizaje de la teoría de la normalización y los conceptos de dependencias funcionales, vamos a suponer que se trabaja con un único esquema de relación  $R$  ficticio y que contiene todos los atributos, o sea  $R=\{A_1,A_2,\dots,A_n\}$ . Supongamos ahora dos subconjuntos de estos atributos que llamamos  $X$  e  $Y$ . Se dice que el subconjunto  $Y$  depende funcionalmente de  $X$  si para cualquier dos tuplas  $t_1$  y  $t_2$  tales que  $t_1[X]=t_2[X]$  entonces se cumple  $t_1[Y]=t_2[Y]$ . Eso quiere decir que los valores del atributo  $Y$  de una tupla de  $R$ , dependen de los valores de  $X$ , para cualquier tupla de  $R$ . Se dice que  $X$  es un determinante de  $Y$ , o que los valores de  $Y$  vienen determinados por los de  $X$ .

Una dependencia funcional entre  $X$  y  $Y$  la representamos por  $X \rightarrow Y$ . Obviamente se deduce que si  $X$  es una clave candidata de  $R$ , entonces  $X \rightarrow Y$  para cualquier subconjunto de atributos de  $R$ . Una dependencia funcional básica se deduce de la semántica de los atributos que forman  $X$  e  $Y$ .

La normalización de los datos hasta la forma normal de Boyce-Codd puede considerarse como un proceso de análisis de los esquemas de relación, basado en sus dependencias funcionales y claves primarias para conseguir una minimización de las redundancias y también de las anomalías de inserción, modificación y eliminación. Las formas normales basadas en dependencias funcionales son las tres primeras y la de Boyce-Codd. Diremos que un esquema de relación está en una determinada forma normal cuando ésta sea la más alta a la que ha sido normalizada. Los conceptos básicos que se utilizan en la definición de las formas normales son:

**Superclave.** Una superclave de un esquema de relación es una colección de atributos que identifica de forma única una tupla de la relación.

**Clave.** Es una superclave con la propiedad adicional que no se puede eliminar ningún atributo sin que deje de ser una superclave.

**Clave candidata.** En un esquema de relación podemos tener más de una superclave. En este caso diremos que tenemos claves candidatas.

**Clave primaria.** Es la escogida para identificar la relación.

**Atributo primo.** Es un atributo que forma parte de una clave candidata

**Atributo no primo.** Es un atributo que no forma parte de ninguna clave candidata

**Dependencia funcional total.** Una dependencia funcional  $X \rightarrow Y$  es una dependencia funcional total si la eliminación de cualquier atributo A de X hace que la dependencia deje de ser válida.

**Dependencia funcional parcial.** Una dependencia funcional  $X \rightarrow Y$  es una dependencia funcional parcial si es posible eliminar un atributo A de X y la dependencia funcional sigue siendo válida.

A partir de estos conceptos básicos, las formas normales se definen de la siguiente manera:

**Primera forma normal (1FN).** Actualmente la 1FN ya se considera como una parte de la definición formal de relación en el modelo relacional, aunque históricamente se definió para prohibir los atributos multivaluados, los atributos compuestos y sus combinaciones. Establece que el dominio de un atributo debe incluir sólo valores atómicos (simples, indivisibles) y que el valor de un atributo en una tupla debe ser un valor individual del dominio del atributo.

**Segunda forma normal (2FN).** La 2FN se basa en el concepto de dependencia funcional total. Un esquema de relación R está en segunda forma normal si todo atributo no primo A de R depende funcionalmente de manera total de la clave primaria de R (o de forma más general de alguna clave de R). En caso de que un esquema de relación no esté en segunda forma normal, se le puede normalizar dando lugar a varias relaciones que si lo estarán. Para ello crearemos nuevas relaciones donde los atributos no primos estarán asociados sólo a la parte de la clave primaria de la que dependen funcionalmente de manera total.

**Tercera forma normal (3FN).** La 3FN se basa en el concepto de dependencia transitiva. Una dependencia funcional  $X \rightarrow Y$  es una dependencia transitiva si existe un conjunto de atributos Z que no sea un subconjunto de cualquier clave de R y se cumplen tanto  $X \rightarrow Z$  y  $Z \rightarrow Y$ . Entonces decimos que un esquema de relación R está en tercera forma normal 3FN si está en 2FN y ningún atributo no primo de R depende transitivamente de la clave primaria de R. O sea, que todos los atributos no primos han de ser funcionalmente dependientes de la clave primaria y no lo pueden ser de otro atributo no primo. Detectamos que una relación no está en 3FN si vemos que un atributo no clave viene determinado por otro atributo no clave. El proceso de normalización consistirá en descomponer la relación que no está en 3FN en otras relaciones, que manteniendo las mismas dependencias funcionales, estén en 3FN. El proceso a seguir consistirá en crear nuevas relaciones donde el atributo no primo que depende de otro no primo, pasará a ser la clave primaria de una nueva relación.

Otra definición de la tercera forma normal es la siguiente: un esquema de relación R está en tercera forma normal 3FN si siempre que una dependencia funcional  $X \rightarrow A$  se cumple en R tenemos que:

- X es una superclave de R, o bien
- A es un atributo primo de R

**Forma normal de Boyce–Codd (FNBC).** Fue propuesta como una forma más estricta que la 3FN ya que toda relación que está en FNBC está también en 3FN, pero no al revés. Se dice que un esquema de relación está en la FNBC si siempre que una dependencia funcional  $X \rightarrow A$  es válida en R, entonces X es una superclave de R.

Una vez descritos los conceptos básicos para la normalización de esquemas de bases de datos relacionales, veamos un ejercicio sencillo que servirá de muestra para ver el proceso de normalización y que utilizaremos de ejemplo para el módulo desarrollado.

Dada la relación **NOTAS** con clave primaria formada por **Num\_alumno Codi\_asig** y formada por los siguientes atributos:

**NOTAS** (*Num\_alumno*, **Codi\_Asig**, *Nota*, *Dni*, *Nom\_Alu*, *Apel1\_alu*, *Apel2\_alu*, *Nombre\_asig*, *Codi\_Pobla*, *Nombre\_Pobla*)

Y con las dependencias funcionales básicas siguientes:

$\{Num\_alumno, Codi\_Asig\} \rightarrow \{Nota\}$   
 $\{Num\_alumno\} \rightarrow \{Dni, Nom\_Alu, Apel1\_alu, Apel2\_alu, Codi\_Pobla, Nombre\_Pobla\}$   
 $\{Dni\} \rightarrow \{Num\_alumno\}$   
 $\{Codi\_Asig\} \rightarrow \{Nombre\_asig\}$   
 $\{Codi\_Pobla\} \rightarrow \{Nombre\_Pobla\}$       *se pide*

- ¿La relación NOTAS está en 2FN? En caso que no lo esté, normalizar a 2FN.
- ¿La relación NOTAS o las relaciones resultantes de normalizar a 2FN (caso que no hubiese estado a 2FN y hayas normalizado a 2FN) están en 3FN? En caso que no lo estén normalizar a 3FN.
- ¿La relación NOTAS o las relaciones resultantes de normalizar a 3FN (caso que no hubiese estado a 3FN y hayas normalizado a 3FN) están en FNBC? En caso que no lo estén normalizar a FNBC.

Como que no tenemos atributos compuestos ni multi-valorados podemos afirmar que la relación está en 1FN. Vemos también que no está en 2FN, ya que hay atributos no primos que sólo dependen de una parte de la clave principal. Normalizando a 2FN, obtenemos las relaciones siguientes: (negrita → clave principal, cursiva → clave foránea, negrita cursiva → clave principal y foránea a la vez)

**NOTAS** (*Num\_alumno*, **Codi\_Asig**, *Nota*)  
**ALUMNO** (*Num\_alumno*, *Dni*, *Nom\_Alu*, *Apel1\_alu*, *Apel2\_alu*, *Codi\_Pobla*, *Nombre\_Pobla*)  
**ASIGNATURA** (**Codi\_Asig**, *Nombre\_asig*)

De estas tres relaciones resultantes, observamos que la relación ALUMNO, no está en 3FN, ya que hay dependencias funcionales transitivas. Normalizando a 3FN obtenemos el siguiente esquema normalizado.

**NOTAS** (*Num\_alumno*, **Codi\_Asig**, *Nota*)  
**ALUMNO** (*Num\_alumno*, *Dni*, *Nom\_Alu*, *Apel1\_alu*, *Apel2\_alu*, *Codi\_Pobla*)  
**POBLACION** (**Codi\_Pobla**, *Nombre\_Pobla*)  
**ASIGNATURA** (**Codi\_Asig**, *Nombre\_asig*)

Y comprobamos que las relaciones resultantes ya están normalizadas hasta FNBC.

Esta es una solución correcta, ahora bien, podemos considerar otras soluciones correctas. Observando la dependencia funcional  $\{Dni\} \rightarrow \{Num\_alumno\}$  nos indica que Dni es una clave candidata, con lo que también sería correcta la solución:

**NOTAS** (*Dni*, **Codi\_Asig**, *Nota*)  
**ALUMNO** (**Dni**, *Num\_alumno*, *Nom\_Alu*, *Apel1\_alu*, *Apel2\_alu*, *Codi\_Pobla*)  
**POBLACION** (**Codi\_Pobla**, *Nombre\_Pobla*)  
**ASIGNATURA** (**Codi\_Asig**, *Nombre\_asig*)

## **7.3 NECESIDADES Y PROBLEMÁTICA EN LA ENSEÑANZA DE LA NORMALIZACIÓN DE BASES DE DATOS.**

El interés por la normalización figura de forma explícita en las recomendaciones de la ACM/IEEE sobre el currículum en informática [CurACM]. Toda la bibliografía especializada en BD trata el tema de la normalización y en todos los programas de las titulaciones informáticas de las universidades españolas encontramos un tema dedicada a ella. Por todo lo anterior, da la sensación de ser un tema vigente, sin embargo en las empresas se da más importancia a realizar buenos diseños conceptuales de BD y terminar las aplicaciones cuanto antes que en comprobar si sus esquemas de bases de datos están normalizados.

El diseño conceptual [Che76], que apareció históricamente después de la normalización, concentra el mayor esfuerzo del diseño de una BD. El diseño lógico posterior es una transformación del esquema

conceptual al esquema lógico que se realiza siguiendo unos pasos determinados [TYF86] [EN07]. Si estos apartados se realizan de forma rigurosa y se parte de un buen diseño conceptual, el diseño lógico resultante estará normalizado, en la mayoría de los casos, hasta FNBC [Dat04] [Byr03]. ¿Quiere esto decir que podemos prescindir de normalizar? Fotache [Fot05] [Fot06b] considera que a pesar de que la normalización es un buen conjunto de conceptos y metodologías, el excesivo grado de formalización hace que los diseñadores desarrollen sus bases de datos a partir del diseño conceptual (y también, porque no decirlo, de su intuición y experiencia) y luego en algunos casos intenten validar su esquema utilizando la normalización. Otros autores consideran que la normalización es un elemento valioso para fines académicos [DJP09] [Fot06a] aunque luego no se utilizará en las aplicaciones reales.

En general los alumnos consideran que el proceso de normalización es complicado, abstracto y difícil de asimilar. Por este motivo no es de extrañar que se desarrollen técnicas alternativas al proceso habitual de normalización como por ejemplo [KC04] [KT07] en la que los autores desarrollan una “receta” para obtener las relaciones normalizadas simplemente siguiendo sus pasos. Aunque según los autores se consiguen buenos resultados, en este caso los alumnos desconocen el sentido de la normalización y se convierten en meros autómatas de la receta. Otros autores [FK05] desarrollan juegos para intentar motivar a los alumnos en este aprendizaje.

Desde un punto de vista docente es muy importante que el alumno sepa valorar si un esquema de bases de datos cumple las formas normales, especialmente las tres primeras y la de Boyce-Codd. De esta forma, descubrirá errores en sus esquemas, sabrá como resolverlos y será capaz de decidir que esquema de bases de datos es el más apropiado. La cuarta y quinta forma normal, desde un punto de vista docente, no tienen tanta importancia.

## **7.4 CORRECCIÓN AUTOMÁTICA DE EJERCICIOS DE NORMALIZACIÓN**

Tal como se ha comentado en la sección 2.4.4, la mayor parte de herramientas desarrolladas actúan como una calculadora, que muestra el resultado de la normalización, pero no el proceso ni los pasos intermedios hasta conseguirlo. Aunque se han desarrollado bastantes entornos para la ayuda en el aprendizaje de la normalización, son pocas las plataformas de e-learning que permiten la corrección automática de este tipo de ejercicios y que faciliten el feedback necesario para ayudar al alumno. Los entornos que corrigen este tipo de ejercicios son NORMIT [Mit02], FDTutor [ECZ08] y LDBN [Geo08]. NORMIT va corrigiendo las preguntas que a modo de cuestionario se hacen al alumno y que le van guiando en el proceso de normalización. FDTutor corrige ejercicios de dependencias funcionales y LDBN corrige el resultado de un proceso de normalización, en la línea de nuestro trabajo. Las demás son herramientas que ayudan y guían en el proceso de aprendizaje de la normalización de bases de datos.

Otra de las limitaciones de estos trabajos, con excepción de NORMIT, es que se presentan como entornos aislados y que sólo facilitan el aprendizaje de un tema concreto, en este caso la normalización.

Nuestro objetivo en la corrección automática de ejercicios de normalización es reutilizar el entorno ACME-EBD visto en el capítulo anterior para la entrada de esquemas normalizados, que el sistema corregirá.

## **7.5 ACME-NOR: MÓDULO DE NORMALIZACIÓN**

En esta sección se detalla el módulo referente a la normalización de bases de datos relacionales que denotamos como ACME-NOR. A diferencia de las herramientas comentadas en el punto anterior nosotros no pretendemos construir una herramienta que vaya guiando paso a paso al alumno en el proceso de aprendizaje, sino que partimos de entrada que el alumno ya dispone del material apropiado y que en las clases presenciales ya se le ha explicado el tema. Nuestro objetivo es que el alumno se ejercite en el proceso de normalización, que el sistema le corrija las soluciones enviadas de los problemas asignados y finalmente que el sistema pueda evaluar esta actividad de forma automática.

El problema que planteamos consiste en desarrollar un sistema que de forma totalmente automática sea capaz de corregir un problema típico de normalización de un esquema de bases de datos relacional, como por ejemplo el mostrado en la sección 7.2. Un mismo ejercicio puede tener varias soluciones correctas, por lo que el sistema deberá tener en cuenta que la solución puede no ser única. Tal como se ha descrito anteriormente, el resultado final de un proceso de normalización pasa por la obtención de un conjunto de relaciones (tablas) normalizadas hasta una determinada forma normal. La información a introducir al sistema, la corrección a efectuar y el feedback a retornar son los mismos que en el tipo de ejercicios de corrección de esquemas de bases de datos relacionales vista en el capítulo anterior, por lo tanto los requisitos a cumplir por el sistema serán los mismos que se han descritos en la sección 6.5.1.

La solución aportada es exactamente la misma que en el módulo de esquemas de bases de datos (capítulo 6). La estructura del problema vista en la sección 6.5.3 es exactamente igual, pero en el momento de la visualización no se visualiza ningún enunciado general y sólo se visualiza el enunciado propio del problema. La interfaz, el sistema de corrección, los mensajes retornados y el sistema de puntuación son exactamente los mismos que los de esquemas de bases de datos relacionales vistos en las secciones 6.5.4, 6.5.5 y 6.5.6 respectivamente. En la Figura 7.1 se muestra la visualización de un ejercicio de este tipo. El sistema de puntuación adoptado también es idéntico al descrito en la sección 6.5.6.

Estàs a » ACME » Inici » Bases de dades » NORMALITZACIÓ » Exercici 6

ALUMNE: JOSEP MARIA DEL VAL TORRA

EXERCICI: 6 de l'activitat NORMALITZACIÓ

Donada la relació R formada pels atributs R(A,B,C,D,E,F,G,H,I,J,K) i amb les dependències funcionals

A,B,F --> C,D,E,G,H,I,J,K  
 E --> G,H,I,J,K  
 B --> C  
 J --> K

Entreu les taules resultants del procés de normalització fins a la Forma normal de Boyce-Codd.

Data	Solució Enviada	Resultat
No s'ha enviat cap solució		

Nom de la taula:

Nom del camp:

Tipus del camp:

Clau principal  
 Clau forana  
 Altres camps

Figura 7.1 Visualización ejercicio normalización e interfaz de entrada.

## 7.6 EVALUACIÓN DEL MÓDULO

La misma metodología explicada en la sección 6.6 fue utilizada para este tipo de ejercicios. Para ello se creó una colección formada por 50 problemas. Cada problema partía de una única tabla y un conjunto de dependencias funcionales. A partir de ahí el alumno debía realizar la normalización hasta la forma normal indicada e introducir las tablas normalizadas a través de la misma interfaz diseñada para los ejercicios de EBD. El resultado de la normalización de la tabla universal propuesta suele dar lugar a entre cuatro y diez tablas normalizadas.

A partir de la colección creada, y al igual que los correctores anteriores, ACME-NOR se ha utilizado principalmente como complemento a las clases presenciales durante los cinco últimos cursos de la asignatura de “Bases de Datos” de la carrera de Ingeniería Técnica en Informática de Gestión (ITIG). Una

vez se ha explicado el modelo relacional, se introducen los conceptos básicos para la normalización de Bases de Datos y se realizan diferentes ejemplos en las sesiones presenciales. A continuación se asignan varios problemas a través de la plataforma ACME con el objetivo de evaluar el nivel cognitivo de aplicación y de análisis de la taxonomía de Bloom. El profesor elige el nivel de dificultad de los problemas seleccionándolos del repositorio y también establece el número de problemas y la valoración de cada uno de ellos de caras a la evaluación sumativa. Mencionar que el grado de dificultad de los ejercicios va creciendo, siendo los últimos de mayor dificultad. La realización de estos ejercicios se tiene en cuenta para la valoración final de la asignatura. Para que los alumnos trabajen de forma precisa se les insiste que tendrán penalizaciones en función del número de intentos hasta llegar a la solución correcta. Se insiste a los alumnos en la importancia del trabajo continuado.

Los datos de utilización de este entorno están reflejados en la Tabla 7.1. Para cada curso se muestra el número de alumnos, los problemas asignados, el % de ejercicios que han sido leídos, el % de ejercicios a los cuales se han enviado soluciones y el % que han sido resueltos correctamente. Finalmente en las cuatro últimas columnas se especifica el % de intentos con los que se ha alcanzado la solución correcta (1, 2, 3 o más intentos).

CURSO	ALUMNOS	PROBLEMAS	% EJERCICIOS LEIDOS	% SOLUCIONES ENVIADAS	% SOLUCIONES CORRECTAS	% INTENTOS PARA LLEGAR A LA SOLUCIÓN CORRECTA			
						1	2	3	>3
2005/06	61	6	89,6%	86,9%	82,0%	48,0%	26,8%	11,4%	13,8%
2006/07	50	8	91,5%	88,5%	82,5%	50,9%	23,6%	17,6%	7,9%
2007/08	46	8	84,2%	77,8%	72,3%	57,4%	23,8%	9,8%	9,0%
2008/09	36	8	81,4%	78,6%	75,0%	51,4%	28,6%	11,4%	8,6%
2009/10	32	6	80,2%	77,0%	72,9%	54,2%	26,4%	8,5%	10,9%

Tabla 7.1 Utilización del entorno ACME-NOR.

Como se puede apreciar en esta tabla más de un 80% de problemas son leídos y a más de un 77% de problemas se envían soluciones. Alrededor de un 75% son resueltos correctamente y en más del 85% de los casos dentro de los tres primeros intentos. Por otra parte si descartamos los ejercicios asignados que ni tan siquiera han sido visualizados por el alumno (entre un 9% y un 20% y que hacen referencia a alumnos matriculados pero que no asisten a clase o abandonan la asignatura) y sólo nos fijamos en las soluciones enviadas por los alumnos realmente interesados (Tabla 7.2), vemos que prácticamente siempre envían las soluciones y en más del 85% de los casos obtienen la solución correcta.

CURSO	ALUMNOS	PROBLEMAS	% EJERCICIOS LEIDOS	% SOLUCIONES ENVIADAS	% SOLUCIONES ENVIADAS/LEIDAS	% SOLUCIONES CORRECTAS	
						CORRECTAS	CORRECTAS/LEIDAS
2005/06	61	6	89,6%	86,9%	97,0%	82,0%	91,5%
2006/07	50	8	91,5%	88,5%	96,7%	82,5%	90,2%
2007/08	46	8	84,2%	77,8%	92,4%	72,3%	85,8%
2008/09	36	8	81,4%	78,6%	96,5%	75,0%	92,1%
2009/10	32	6	80,2%	77,0%	96,0%	72,9%	90,9%

Tabla 7.2 Valoración sobre los ejercicios ACME-NOR leídos.

El porcentaje de alumnos que envían soluciones correctas varía en función del nivel de dificultad del problema, variando desde alrededor de un 90% en los problemas más fáciles a una horquilla entre el 58% y un 70% en los más difíciles. En los primeros problemas, los más fáciles, se llega a la solución correcta en uno o dos intentos, mientras que en los más difíciles el número de intentos crece considerablemente. Como puede apreciarse el grado de participación suele ser muy alto. Este hecho, por si solo, ya mejora el anterior sistema, en que facilitábamos una lista escrita de problemas y eran muy pocos los alumnos que los resolvían.

En el capítulo 10 se realiza una valoración más exhaustiva de las mejoras en el aprendizaje del diseño de bases de datos que comporta la utilización de ACME-NOR, juntamente con el entorno ACME-ER para el desarrollo de ERD y ACME-EBD para el diseño lógico de bases de datos.

## **7.7 APORTACIONES Y CONCLUSIONES**

Se ha reutilizado el corrector de esquemas de bases de datos relacionales para la normalización de éstas. Se ha creado una colección de problema y se está utilizando en la asignatura de “Bases de datos”. Las principales ventajas que presenta este módulo son:

- Disminución del trabajo del profesor, en cuanto a horas dedicadas a la corrección de este tipo de ejercicios.
- Muchos más alumnos se ejercitan en la resolución de este tipo de problemas que en las metodologías clásicas. Además obtienen un resultado inmediato de la corrección.
- Mayor feedback alumno/profesor. Se aclaran muchas dudas que surgen en el momento de la resolución de los ejercicios.
- Seguimiento del trabajo del alumno.
- Evaluación automática del trabajo realizado por el alumno.

El trabajo realizado en el desarrollo y experiencia del uso de este corrector ha dado lugar al artículo “**A web-based Problem-Solving Environment for database Normalization**” presentado en el 8avo International Symposium on Computers in Education SIIE 06, 2006, referenciado en [SBP+06]

La utilización de parte de ACME-DB como un complemento a las clases presenciales en la docencia de bases de datos ha sido presentada en las XII Jornadas de la Enseñanza Universitaria de la Informática (JENUI) en el 2006 en el artículo “**Utilización de una plataforma de e-learning en la docencia de bases de datos**”, referenciado por [SPB+06b].

En el artículo “**Database design using a web-based e-learning tool**” presentado en el 2nd Workshop on Methods and Cases in Computing Education 2009 (MCCE) se expone la experiencia en la utilización de ACME-DB en el aprendizaje de diseño de bases de datos. Este artículo está referenciado en [SBP+09a].

# CAPÍTULO 8. MÓDULO DE CONSULTAS EN ÁLGEBRA RELACIONAL

En este capítulo presentaremos el quinto de los módulos correctores desarrollados: el módulo de álgebra relacional. Este módulo dará soporte al aprendizaje de la resolución de consultas mediante expresiones del álgebra relacional.

Este capítulo se ha estructurado de la siguiente forma. En la sección 8.1 se hace una introducción al tema. En la sección 8.2 se describen las operaciones básicas del álgebra relacional y un ejemplo ilustrativo. En la sección 8.3 describimos las necesidades y la problemática de la enseñanza del álgebra relacional. En la sección 8.4 se realiza un breve resumen de los entornos existentes. En la sección 8.5 se plantea el módulo, se detalla la solución aportada y su integración en la plataforma ACME. Finalmente en las secciones 8.6 y 8.7 se describen los resultados académicos conseguidos y las aportaciones y conclusiones a las que hemos llegado.

## 8.1 INTRODUCCIÓN

Un modelo de datos, como por ejemplo el modelo relacional, debe definir un conjunto de operaciones para definir su estructura junto con los conceptos necesarios para manipular los datos. De esta forma, para la manipulación de una base de datos relacional y la realización de todo tipo de consultas se han definido diferentes tipos de lenguajes de consulta (véase sección 2.3.3), como por ejemplo:

- Lenguajes formales: el álgebra relacional y el cálculo relacional
- Structured Query Language (SQL)

El álgebra relacional es una parte integral del modelo relacional y está formada por un conjunto de operaciones para manipular las tablas, con el objetivo de realizar consultas seleccionando la información a obtener. El resultado de una consulta es una nueva relación. Por consiguiente, las operaciones del álgebra relacional siempre producen nuevas relaciones que pueden ser manipuladas más adelante usando de nuevo operaciones del álgebra. Una secuencia de operaciones de álgebra relacional constituye una expresión de álgebra relacional. En esta secuencia se va detallando en qué orden hay que aplicar las diferentes operaciones sobre cada tabla para llegar al resultado final. Se trata pues de un lenguaje procedural en el que hay que describir el orden de ejecución de las operaciones. El álgebra relacional es importante por varias razones:

- Proporciona un fundamento formal para las operaciones del modelo relacional.
- Se utiliza como base para la implementación y optimización de consultas en los SGBD relacionales
- Algunos de sus conceptos se han incorporado al lenguaje estándar de consultas SQL.

Aunque ninguno de los SGBD comerciales actuales proporcionan una interfaz para las consultas de álgebra relacional, las funciones y operaciones centrales de cualquier sistema relacional están basadas en estas operaciones.

A pesar de la importancia del álgebra relacional, sus operaciones son consideradas muy técnicas por la mayoría de los usuarios de un SGBD comercial, ya que una consulta en álgebra relacional se escribe como una secuencia de operaciones que al ejecutarse, producen el resultado final. Por lo tanto, el usuario debe especificar el orden en que hay que ejecutar las operaciones de la consulta, pareciéndose, en cierta medida, a la realización de un programa en que todas sus operaciones se realizan de forma secuencial. En cambio SQL proporciona una interfaz de lenguaje declarativo de alto nivel, en el que el usuario sólo especifica lo que quiere obtener, dejando para el SGBD la optimización y las decisiones de cómo ejecutar la consulta. Aunque SQL incluye algunas características del álgebra relacional, está basado en gran medida en el cálculo relacional de tuplas. La sintaxis de SQL es mucho más amigable para el usuario que los otros dos lenguajes formales.

## 8.2 EL ÁLGEBRA RELACIONAL

Además de definir la estructura y restricciones de la base de datos, un modelo de datos debe incluir un conjunto de operaciones para la manipulación de los datos. El álgebra relacional está formada por un conjunto básico de operaciones que permiten al usuario especificar las distintas consultas. El resultado de una recuperación (consulta) es una nueva relación que se ha formado a partir de una o más relaciones. La nueva relación creada también la podremos manipular con las mismas operaciones del álgebra. Una secuencia de operaciones del álgebra relacional forma una expresión de álgebra relacional, cuyo resultado siempre es una relación. La importancia del álgebra relacional se basa en que la utilizan los SGBD para el proceso de optimización y creación del código asociado a una consulta SQL, tal como se muestra en la Figura 8.1.

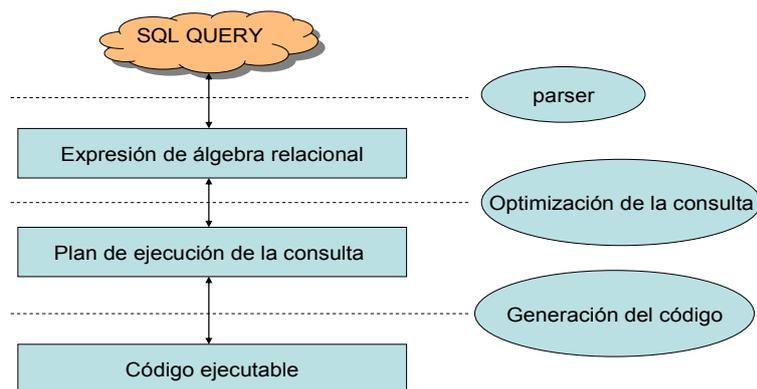


Figura 8.1 Proceso de optimización y generación de código de una sentencia SQL.

Las operaciones básicas del álgebra relacional suelen clasificarse en dos grupos: las operaciones de la teoría matemática de conjuntos (unión, intersección, diferencia y producto cartesiano) y las operaciones creadas específicamente para bases de datos relacionales (selección, proyección, reunión, funciones de agregación, ...). Veamos las más importantes:

### **Operación Seleccionar.**

Sirve para seleccionar un subconjunto de tuplas de una relación que satisfacen una condición de selección. Se representa mediante la letra sigma ( $\sigma$ ). La estructura y sintaxis de la operación es:

$$\begin{array}{c} \sigma \langle \text{condición de selección} \rangle \langle \text{nombre de la relación} \rangle \\ \downarrow \\ \langle \text{nombre atributo} \rangle \langle \text{operador relacional} \rangle \langle \text{valor constante} \rangle \\ \langle \text{nombre atributo} \rangle \langle \text{operador relacional} \rangle \langle \text{nom atributo} \rangle \\ \downarrow \\ = \# < > >= <= \end{array}$$

Además, podemos incorporar los operadores booleanos y, o, no

Se trata de una operación unaria ya que se aplica a una sola relación. El grado de la relación resultante es el mismo de la relación inicial, no se pierden atributos. El número de tuplas de la relación resultante será menor o igual que la relación original. Seleccionar es una operación conmutativa ya que:

$$\sigma \langle \text{cond}_1 \rangle (\sigma \langle \text{cond}_2 \rangle (R)) = \sigma \langle \text{cond}_2 \rangle (\sigma \langle \text{cond}_1 \rangle (R))$$

Un conjunto de operaciones seleccionar siempre se puede simplificar a través del operador y, de forma

$$\sigma \langle \text{cond}_1 \rangle (\sigma \langle \text{cond}_2 \rangle \dots (\sigma \langle \text{cond}_n \rangle (R))) = \sigma \langle \text{cond}_1 \rangle \text{ y } \langle \text{cond}_2 \rangle \text{ y } \dots \langle \text{cond}_n \rangle (R)$$

### **Operación proyectar.**

La operación proyectar selecciona ciertas columnas de una relación y elimina las demás. Se representa mediante la letra pi ( $\Pi$ ). La estructura y sintaxis de la operación es:

$$\Pi\langle\text{Lista atributos}\rangle(\langle\text{nombre relación}\rangle)$$

El grado de la nueva relación será el número de atributos especificado. Si la lista de atributos solamente contiene atributos no clave, es posible que haya tuplas duplicadas, en este caso se eliminan las tuplas duplicadas (por la restricción de entidades del modelo relacional). El número de tuplas resultantes es menor o igual a las de la relación inicial. Si la lista de atributos, contiene un campo clave, el número de tuplas será el mismo que la relación original. No es una operación conmutativa.

### **Secuencia de operaciones**

Se pueden expresar varias operaciones del álgebra relacional, una detrás de otra, formando una expresión de operaciones de álgebra relacional, como por ejemplo:

$$\text{RESULTADO} \leftarrow \Pi \text{ dni, nombre, apellido1 } (\sigma_{\text{población}=\text{“GIRONA”}}(\text{ALUMNOS}))$$

o bien se pueden ir creando tablas intermedias y operar sobre estas tablas

$$\begin{aligned} \text{ALUMNOS\_GIRONA} &\leftarrow \sigma_{\text{población}=\text{“GIRONA”}}(\text{ALUMNOS}) \\ \text{RESULTADO} &\leftarrow \Pi \text{ dni, nombre, apellido1 } (\text{ALUMNOS\_GIRONA}) \end{aligned}$$

Desde un punto de vista estrictamente docente es mejor esta segunda forma ya que se ven más claras las operaciones del álgebra relacional y los pasos que hay que ir realizando hasta llegar a la solución final.

### **Operación renombrar**

Esta operación nos permite cambiar (renombrar) el nombre de los atributos resultantes en la relación final. Por ejemplo  $\text{RESULTADO}(\text{Identidad, nombre, apel1}) \leftarrow \Pi \text{ dni, nom, cognom1}(\text{ALUMNOS})$

### **Operaciones unión, intersección y diferencia**

Son las operaciones típicas de la teoría de conjuntos y siempre se aplican a relaciones con el mismo tipo de tuplas. Se representan con los típicos símbolos  $U$ ,  $\cap$  y  $-$ .

### **Operación de reunión**

Sirve para combinar tuplas relacionadas de dos relaciones. Esta operación es sin duda la más importante y permite reunir tuplas de dos tablas. Combina las tuplas de una relación con las de otra, siempre que cumplan con la condición especificada. Se representa con el símbolo  $\bowtie$ . Así pues si tenemos dos relaciones  $R(A_1, A_2, \dots, A_n)$  y  $S(B_1, B_2, \dots, B_m)$  la reunión de estas dos relaciones se expresa como:

$$Q \leftarrow R \bowtie \langle\text{condición de reunión}\rangle S$$

donde la condición de reunión se puede expresar como:

$$\langle\text{atributo de R}\rangle \langle\text{operador relacional}\rangle \langle\text{atributo de S}\rangle$$

La nueva relación  $Q(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$  estará formada por todos los atributos de la relación R y todos los de la relación S, y sus tuplas serán las que satisfagan la condición establecida. Cada tupla tendrá todos los atributos de R y todos los de S.

### **Operación de reunión natural**

La mayor parte de operaciones de reunión son equireuniones (condición con operador  $=$ ). En este caso en la relación Q resultante hay un par de atributos con los mismos valores. Para evitar esta duplicidad se ha creado la operación reunión natural, representada por el signo  $*$ , la cual elimina el atributo redundante. Así pues la reunión natural  $*$  es una equireunión que elimina el atributo redundante. De manera formal, con la reunión natural de dos relaciones R y S se obtendrá una nueva relación Q fruto de la combinación de las tuplas que tengan el mismo valor del atributo que tenga el mismo nombre en las dos relaciones R y S. El atributo con el mismo nombre no se repetirá en la relación Q resultante. Se representa por  $Q \leftarrow R * S$ .

### Funciones agregadas

A través de las operaciones que hemos definido hasta ahora no se pueden expresar ciertas funciones matemáticas que afectan a un conjunto de tuplas. Por eso se definen un conjunto de funciones agregadas para aplicar a un conjunto de valores. Estas funciones agregadas son: **suma, promedio, máximo, mínimo y cuenta**. Su sintaxis viene determinada de la siguiente forma:

$\mathbf{F}$  <lista de funciones / atributos> (Nombre tabla)

A menudo las funciones agregadas se aplican después de agrupar las tuplas según alguno/s de sus atributo/s. En este caso la sintaxis nos viene determinada por:

<atributos de agrupación>  $\mathbf{F}$  <lista de funciones/atributos> (nombre relación)

### Otras Operaciones

El álgebra relacional dispone además de otras operaciones de menor importancia desde un punto de vista académico como son: producto cartesiano, unión externa, división, reunión externa derecha, reunión externa izquierda y reunión externa completa.

### Ejemplo

Una vez descritas las operaciones básicas del álgebra relacional, veamos un ejemplo que nos servirá de muestra para los correctores desarrollados. El ejemplo a considerar es el típico supuesto de una empresa a la que ya hemos realizado el diseño conceptual (diagrama ER) en el capítulo 4 y hemos obtenido el esquema de la base de datos en el capítulo 6. El punto de partida será pues el esquema de la base de datos obtenido:

**DEPARTAMENTO** (Número\_dep, nombre\_dep, tel\_dep, *nif\_director*)

**EMPLEADO** (Nif, nombre, apellido1, apellido2, direccion, codigo\_postal, ciudad, sueldo, sexo, fecha\_nac, *nif\_super*, *num\_dep*)

**PROYECTO** (Codigo\_proyecto, nombre\_proyecto, ciudad\_proyecto, *numero\_dep*)

**DEPENDIENTE** (Nif\_numero, fecha\_nac\_depe, nombre\_depe, apellido1\_depe, apellido2\_depe, parentesco)

**TRABAJA EN** (Nif, Codigo\_proyecto, horas)

A partir de aquí realizaremos las diferentes consultas a través de las operaciones del álgebra relacional hasta obtener el resultado final. Como ejemplo vamos a realizar la siguiente consulta: "Obtener el nombre y la dirección de todos los empleados que trabajan en el Departamento de Ventas". Para ello las operaciones del álgebra relacional necesarias para la obtención del resultado son:

$DEP\_VENDAS \leftarrow \sigma_{nombre\_dep="Ventas"}(DEPARTAMENTO)$

$EMPL\_DEP\_VENDAS \leftarrow DEP\_VENDAS \bowtie_{Número\_dep=num\_dep}(EMPLEADO)$

$RESULTADO \leftarrow \Pi_{nombre, apellido1, direccion}(EMPL\_DEP\_VENDAS)$

Ahora bien, una misma consulta puede tener muchas soluciones correctas, a modo de ejemplo, esta misma consulta tan sencilla podríamos formularla de formas muy distintas, entre ellas las tres siguientes:

$EMPL\_DEPARTAMENTO \leftarrow DEP\_VENDAS \bowtie_{Número\_dep=num\_dep}(EMPLEADO)$

$EMP\_VENDAS \leftarrow \sigma_{nombre\_dep="Ventas"}(EMPL\_DEPARTAMENTO)$

$RESULTADO \leftarrow \Pi_{nombre, apellido1, direccion}(EMP\_VENDAS)$

$DEP\_VENDAS \leftarrow \Pi_{numero\_dep}(\sigma_{nombre\_dep="Ventas"}(DEPARTAMENTO))$

$EMPL\_DEP\_VENDAS \leftarrow DEP\_VENDAS \bowtie_{Número\_dep=num\_dep}(EMPLEADO)$

$RESULTADO \leftarrow \Pi_{nombre, apellido1, direccion}(EMPL\_DEP\_VENDAS)$

$DEP\_VENDAS(num\_dep) \leftarrow \Pi_{numero\_dep}(\sigma_{nombre\_dep="Ventas"}(DEPARTAMENTO))$

$EMPL\_DEP\_VENDAS \leftarrow DEP\_VENDAS * EMPLEADO$

$RESULTADO \leftarrow \Pi_{nombre, apellido1, direccion}(EMPL\_DEP\_VENDAS)$

## **8.3 NECESIDADES Y PROBLEMÁTICA EN LA ENSEÑANZA DEL ÁLGEBRA RELACIONAL**

La creación, el mantenimiento (inserción, modificación, eliminación) y la obtención de resultados (consultas) de una base de datos son tareas que el alumno debe aprender perfectamente, ya que éstas forman parte de una de las competencias básicas que debe adquirir un graduado en informática. Desde un punto de vista docente, y tal como queda reflejado en el informe [RR03], la mayoría de docentes de bases de datos imparte conocimientos de álgebra relacional y especialmente de SQL. Además coinciden que para una buena comprensión de SQL es importante tener conocimiento de álgebra relacional. Respecto a la introducción al cálculo relacional se le da menos importancia, por ser también un lenguaje declarativo parecido en cierta manera a SQL.

Generalmente, la metodología docente que se aplica en la enseñanza del álgebra relacional consiste en unas sesiones teóricas donde se explican las diferentes operaciones del álgebra. A partir de ahí se plantean inicialmente pequeños ejercicios de consulta sobre una base de datos que requieran la aplicación de algunas de las operaciones explicadas. Finalmente se van planteando consultas más complejas en las que intervengan todas las operaciones del álgebra. El alumno deberá analizar la consulta y detallar la secuencia de operaciones a realizar para la obtención del resultado final. Para poder evaluarlo será necesario que realice diferentes consultas. En estos supuestos finales el número de operaciones a realizar es considerable y además suele haber muchas e imprevisibles soluciones que se pueden considerar correctas.

El proceso de evaluación formativa consiste en la revisión de estas consultas formadas por varias expresiones del álgebra relacional y la realización de las anotaciones correspondientes para que el alumno pueda aprender de sus propios errores. Con el feedback obtenido en la corrección, el profesor adapta las diferentes sesiones presenciales haciendo hincapié en los puntos donde detecte deficiencias de aprendizaje. Como se puede apreciar estas tareas de revisión son laboriosas y conllevan una gran dedicación difícilmente asumible por el profesor, lo que dificulta la aplicación del sistema de evaluación formativa.

Por otra parte la realización de consultas de álgebra relacional la podemos enmarcar en los niveles cognitivos intermedios (aplicación y análisis) dentro de la taxonomía de Bloom. Tal como se ha comentado en el capítulo 2, en un entorno de CBA es muy difícil evaluar estos niveles cognitivos a través de preguntas de respuesta fija y se requiere de entornos mucho más especializados. La resolución de una consulta en álgebra relacional es parecida en cierta manera a la realización de un programa secuencial, en el que intervienen tablas, atributos y las operaciones del álgebra. Tal como se ha comentado en la sección 2.2.2, en la evaluación automatizada a partir de respuestas libres la corrección de programas informáticos ha sido un tema en el que se han desarrollado muchos trabajos. En cambio son pocos los entornos especializados en la corrección de expresiones de álgebra relacional.

Comentar que aunque no sea una tarea extremadamente difícil, a menudo la realización de consultas en álgebra relacional resulta costosa ya que supone un conocimiento a fondo del modelo relacional y de las operaciones del álgebra.

Nuestro objetivo es solventar estas limitaciones diseñando y desarrollando un entorno CBA especializado que nos permita la evaluación automatizada de los niveles cognitivos intermedios de la taxonomía de Bloom en materia de consultas de álgebra relacional. Para ello debe permitir definir consultas en álgebra relacional, su corrección y generación de feedback automático. El sistema debe soportar la evaluación formativa de manera que el profesor pueda controlar el trabajo del grupo y no deba perder tiempo en la corrección. Este entorno debe integrarse en la plataforma ACME actuando conjuntamente como un sistema CBA con el que se realizarán las correspondientes pruebas sobre los alumnos de la asignatura de BD, con el objetivo de valorar si su utilización aporta mejoras en su aprendizaje.

## **8.4 CORRECCIÓN AUTOMÁTICA DE CONSULTAS DE ÁLGEBRA RELACIONAL**

Tal como hemos comentado en la sección 2.4.5 según nuestro conocimiento no hay plataformas de e-learning que permitan la corrección automática de ejercicios de álgebra relacional y que faciliten el feedback necesario para ayudar al alumno en caso de que cometa errores. Ninguna de las herramientas examinadas permite la corrección de forma automática. La mayoría no están desarrolladas en un entorno web y es necesario descargar e instalar el software para su utilización. Todas ellas están pensadas para que el alumno escriba una expresión en álgebra relacional (en diferentes formatos, según la herramienta) y el sistema interprete, ejecute y visualice los resultados de las distintas operaciones. Son pocas las que permiten la notación matemática habitual.

Dada esta falta de entornos, conscientes de la importancia del álgebra relacional y siempre avalados por los diseños curriculares más prestigiosos en Computer Science [AcmCur] y por la bibliografía más relevante en la materia [EN07], [Dat04], [RC04], [CB09] nos propusimos desarrollar, dentro de la plataforma ACME, un entorno que permitiera la corrección de consultas de álgebra relacional.

## **8.5 ACME-AR: MÓDULO CORRECTOR DE CONSULTAS DE ÁLGEBRA RELACIONAL**

En esta sección se detalla el módulo referente a la evaluación de consultas de álgebra relacional que denotamos como ACME-AR. En la sección 8.5.1 se determinan los requisitos que deberá satisfacer el sistema. En la sección 8.5.2 se presenta una visión general de la solución desarrollada. En las siguientes secciones se detallan los aspectos más técnicos del módulo: la estructura de una consulta de álgebra relacional, su interfaz y el módulo de corrección y feedback retornado.

### **8.5.1 Requisitos del sistema ACME-AR**

El problema que planteamos es desarrollar un sistema que de forma totalmente automática sea capaz de corregir las distintas expresiones de álgebra relacional que nos llevan a la obtención del resultado final de una consulta dada. Una misma consulta puede tener muchas e imprevisibles respuestas válidas, con lo cual no nos sirven los métodos de comparación respecto a soluciones ideales y será necesario ejecutar las soluciones recibidas sobre unos valores de prueba y comprobar el resultado obtenido. Los requisitos establecidos para este módulo son:

- Disponer de una interfaz apropiada para la entrada de expresiones de álgebra relacional que debe contemplar todas las operaciones descritas en el apartado 8.2:
  - Operaciones: Seleccionar, proyectar, reunión, reunión natural, reunión externa derecha, reunión externa izquierda, reunión externa completa, unión, unión externa, intersección, diferencia, producto cartesiano y división.
  - Operadores relacionales: Igual, mayor, menor, diferente, mayor o igual, menor o igual.
  - Operadores booleanos: y, o, no.
  - Funciones agregadas: Suma, promedio, contar, máximo y mínimo.
  - Función renombrar.
  - Funcionalidades de: borrar relación, borrar todo, corregir.
  - Visualizador de relaciones iniciales.
  - Visualizador de relaciones intermedias resultantes.
  - Visualizador de los atributos de cada relación.
  - Visualizador de mensajes del sistema.
- Disponer de un corrector apropiado. Este corrector deberá tener tres partes diferenciadas:
  - Disponer de un analizador sintáctico. Cuando el alumno envía la solución, el sistema recoge la información asociada a las distintas expresiones de álgebra

relacional escritas por el alumno, las analiza y comprueba que su sintaxis sea correcta.

- Disponer de un módulo ejecutor de operaciones. Una misma consulta se puede solucionar de muchas formas distintas y todas ellas correctas, tal como hemos visto en el ejemplo de la sección 8.2. En este caso, a diferencia de los módulos correctores anteriores, no tenemos un acotado número de soluciones que podemos afirmar que son correctas, sino que las soluciones correctas son muchas y no predecibles. Así pues, para determinar si una solución es correcta, será necesario ejecutar cada una de las operaciones de la solución del alumno sobre una serie de tests de valores concretos de la base de datos y verificar si los resultados obtenidos coinciden con los esperados.
  - Disponer de un módulo comparador. Al final de cada ejecución de las operaciones sobre los distintos tests de pruebas obtendremos una relación que compararemos con la solución correcta esperada para este test de pruebas que tenemos almacenada en la descripción del problema. Debemos tener en cuenta que a lo mejor el orden de los atributos es distinto al que consideramos correcto y también puede darse el caso que el orden de las tuplas no sea el mismo, pero los valores retornados han de ser los esperados.
- El sistema debe facilitar la evaluación formativa. Cuando el alumno envíe una respuesta, el sistema recibirá la información de cada una de las expresiones del álgebra relacional con que ha resuelto la consulta. El módulo desarrollado ejecutará cada una de las sentencias enviadas sobre un test de prueba. Finalmente se comparará si la solución obtenida es la esperada y se facilitará el feedback apropiado a cada caso.
  - El sistema debe integrarse a la plataforma ACME para compartir las funcionalidades de ésta.

## 8.5.2 Análisis inicial de ACME-AR

El problema visto desde un punto de vista general se puede tratar como la validación sintáctica de expresiones de álgebra relacional, la ejecución de las distintas operaciones del álgebra sobre un conjunto de valores y la posterior comprobación del resultado obtenido. Evidentemente todo ello debe estar integrado en la plataforma ACME, de forma que tanto profesores como alumnos dispongan de las funcionalidades de la plataforma.

La consulta entrada por el profesor deberá tener una estructura determinada, en la que deben aparecer el enunciado de la consulta a realizar, el esquema de las tablas a las que se aplicará la consulta, el conjunto de valores de los atributos sobre los que se testeará la solución enviada y el resultado esperado al ejecutar la consulta sobre el conjunto de valores.

Como en los módulos anteriores, la plataforma guardará cada solución enviada por el alumno de un determinado ejercicio. Siguiendo la filosofía ACME, si la solución es correcta el sistema lo notificará al alumno mediante un “Correcto”. En caso de que el corrector determine que la solución es incorrecta facilitará pequeñas notificaciones para ayudar al alumno a encontrar el error y poder enviar una nueva solución. Entre estas indicaciones tendremos que distinguir entre las que sean producidas por expresiones sintácticamente mal construidas y las que lleven a resultados incorrectos. En este segundo caso se mostrará al alumno los valores retornados en la ejecución de la consulta y el mensaje de error apropiado.

Al igual que en los correctores anteriores el módulo desarrollado debe estar ligado a la plataforma ACME a la que se tiene que integrar. Para ello es necesario definir:

- **La estructura del problema.** Cada problema se define en un fichero plano y se guarda en el repositorio. Debe estar formada por los distintos enunciados de consultas que se realizan sobre una base de datos determinada. Para cada base de datos se realizará un único problema con tantos enunciados distintos como consultas distintas queramos formular. El profesor mediante un editor cualquiera o el específico de ACME editará la estructura del problema. A continuación y con las funcionalidades específicas de ACME, lo incorporará al repositorio y a partir de ese momento ya estará disponible para ser utilizado.

- Una interfaz apropiada a través de la cual el alumno escribirá las expresiones del álgebra relacional que resuelven la consulta formulada. Una vez entrada, enviará la solución a corregir y el sistema le mandará los mensajes de corrección a través de la interfaz de salida.
- Un módulo corrector que validará e interpretará la expresión, ejecutará las operaciones del álgebra relacional y comparará el resultado obtenido con la solución correcta para la consulta descrita en la estructura del problema.

Veamos con más detalle cada una de estas tres partes.

### 8.5.3 Estructura de un problema de álgebra relacional

La primera parte a tratar es la estructura general del problema que se utiliza para visualizar el enunciado del problema y que también utilizará el módulo corrector para la corrección de las soluciones enviadas por los alumnos. Esta estructura se guarda en un fichero que se puede editar con cualquier editor y que se guarda en el repositorio. Para este tipo de problemas podemos incluir muchas consultas que se ejecutarán sobre los distintos test de pruebas. Cada test de pruebas está formado por la descripción de las relaciones, valores para los distintos atributos que se utilizarán para ejecutar las operaciones y finalmente la solución correcta a cada una de las consultas. La estructura es la siguiente:

```

<identificador de tipo de problema>
<E1> enunciado 1 </E1>
<E2> enunciado 2 </E2>
....
<TABLAS TEST 1>
descripción tablas, atributos y valores
</TABLAS TEST 1>
<SOLUCION1> Solución 1 </SOLUCIO1>
<SOLUCION2> Solución 2 </SOLUCIO2>
....
(***) tantos ítems tablas test y soluciones como test de prueba se quieran usar(***)

```

Las diferentes partes están separadas por marcas. De entrada hay un identificador de problema a través del cual ACME reconoce el tipo de problema. A continuación vienen los enunciados de las consultas que queremos que se asignen a los cuadernos de los alumnos. La siguiente parte describe los test de pruebas. Para cada test de pruebas se determinan las tablas, los atributos y los valores sobre los que se van a ejecutar las expresiones de álgebra relacional recibidas. Finalmente se describen las soluciones que se deberían obtener para cada test y enunciado formadas por los valores que se deben obtener al ejecutar la consulta. Para cada tabla se debe especificar:

<b>NOMBRE TABLA</b>	<i>nombre tabla</i>
<b>CLAVE PRINCIPAL</b>	<i>atributo/s clave principal</i>
<b>CLAVES FORANEAS</b>	<i>atributo/s clave foránea</i>
<lista atributos>	(** $A_1, A_2, A_3, \dots, A_n$ **)
<tuplas con valores>	(** $v_{11}, v_{12}, v_{13}, \dots, v_{1n}$ **)
	(** $v_{21}, v_{22}, v_{23}, \dots, v_{2n}$ **)
	....
	(** $v_{m1}, v_{m2}, v_{m3}, \dots, v_{mn}$ **)

Comentar que estos valores no son visibles ni accesibles a los alumnos de forma que nunca saben sobre qué valores se ejecutarán sus expresiones.

Para cada solución se debe especificar los valores obtenidos en la ejecución de la consulta sobre los datos de las tablas. Estos valores se guardan como un conjunto de tuplas.

```

v11, v12, v13, ... v1j
v21, v22, v23, ... v2j
....
vi1, vi2, vi3, ... vij

```

Una vez creado el fichero con esta estructura, ACME dispone de las funcionalidades específicas para incorporarlo al repositorio. Antes de su incorporación se valida a través del módulo de validación de problemas de álgebra relacional desarrollado. Según el identificador de tipo de problema, el módulo de validación comprueba que la estructura del fichero sea la correcta y a partir de aquí se guarda en el repositorio y pasa a estar disponible para ser utilizado. En caso que este módulo detecte que no se ajusta a la estructura esperada, el sistema dará el mensaje de error correspondiente y no se guardará en el repositorio.

Una vez guardado el problema en el repositorio ya estará disponible para ser asignado a los cuadernos de los alumnos. Siguiendo las especificaciones de la plataforma ACME, el profesor ya podrá seleccionar los ejercicios y el módulo de generación de cuadernos los asignará a los alumnos. Para este tipo de problemas el profesor crea problemas base en que cada uno contiene muchas consultas. Cuando quiere asignar consultas de álgebra relacional a los alumnos, el profesor selecciona unos cuantos problemas base de este tipo e indica a la plataforma cuantas consultas de cada problema base se deben asignar a los cuadernos. El sistema va asignando aleatoriamente consultas de cada problema base a los cuadernos. De esta forma, los cuadernos de los alumnos son similares, pero todos ellos distintos. La estrategia seguida por el módulo es la mostrada en la Figura 8.2

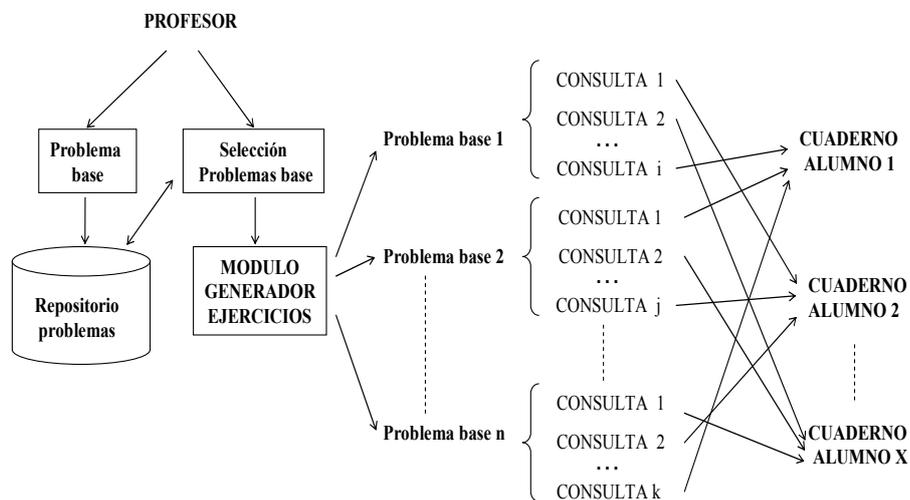


Figura 8.2 Estrategia de asignación de consultas de álgebra relacional.

Una vez asignadas las consultas, el alumno las visualiza y las podrá resolver. Una vez resueltas podrá introducir las expresiones obtenidas a través de la interfaz descrita en la siguiente sección. El corrector, tal como se detalla en la sección 8.5.5, utiliza el resto de información del problema para su verificación.

### 8.5.4 Diseño de la interfaz ACME-AR

El segundo punto a tratar es el referente a la interfaz para la visualización del problema, la entrada de las expresiones de álgebra relacional y la posterior recepción de los mensajes de feedback. En la Figura 8.3 se muestran las distintas zonas en que se ha dividido la interfaz.

JOSEP SOLER MASCO

Estàs a » ACME » Inici » Bases de dades » ÀLGEBRA RELACIONAL » Exercici 1

ALUMNE: MARC HURTOS CORDONET

EXERCICI: 1 de l'activitat ÀLGEBRA RELACIONAL

Escriure les operacions de l'àlgebra relacional necessàries per obtenir el nom, el cognom1 i l'adreça de tots els empleats que treballen en el Departament de VENDES

1

Data	Solució Enviada	Resultat
	No s'ha enviat cap solució	

R1 ←  $\sigma_{\text{NOMDEPART}=\text{"VENDES"}}(\text{DEPARTAMENT})$   
R2 ←  $R1 \bowtie_{\text{NUMD}=\text{NUMDEP}}(\text{EMPLEAT})$   
R3 ←  $\pi_{\text{NOM},\text{COG1},\text{ADREÇA}}(R2)$

6

4

Neteja    Eliminar Relació

Relacions

R1  
R2  
R3

7

Relacions inicials

EMPLEAT  
DEPARTAMENT  
PROJECTES  
EMPLEAT\_TREBALLEN  
DEPENDENTS

2

Camps

DNISUPER  
NOM  
COG1  
COG2  
DATANEIX  
ADREÇA  
POBLACIO  
CODIPOSTAL

3

Corregir 8

Relació : R4-    Enter 5

ALUMNE: MARC HURTOS CORDONET

Figura 8.3 Interfaz de entrada de soluciones a problemas de álgebra relacional.

Las funcionalidades de cada una de estas zonas son:

1. **Enunciado de la consulta a resolver.** Zona de visualización del enunciado.
2. **Relaciones iniciales.** A partir de estas relaciones el alumno tiene que resolver la consulta.
3. **Atributos de la tabla seleccionada.** Se visualizan los atributos de la tabla seleccionada en la zona 2.
4. **Zona de botones.** Contiene las operaciones, operadores y funciones del álgebra relacional. También dispone de botones de borrado total y de eliminar una relación creada por el alumno.
5. **Área de entrada.** Zona de edición de cada una de las expresiones de álgebra relacional.
6. **Área de expresiones entradas.** Zona donde se visualizan las expresiones entradas.
7. **Área de visualización.** Con el nombre de cada una de las relaciones intermedias creadas.
8. **Botón de envío a corregir.** Una vez entradas todas las expresiones de una consulta sirve para enviarla a corregir.

La información de las áreas 1, 2 y 3 es obtenida a partir del problema base. A partir de esta información y con los botones de la zona 4 el alumno va editando sus expresiones en el área 5. Una vez entrada cada una de las expresiones que conforman su solución, al pulsar “enter” aparece en el área 6 de expresiones entradas y se muestra la nueva relación en el área 7 de visualización. Finalmente, una vez entradas cada una de las expresiones que conforman la solución, con el botón “corregir” se envía la solución a la plataforma ACME.

Una vez enviada la solución, ACME la corrige y nos manda el feedback correspondiente. En la Figura 8.4 se visualiza la interfaz de recepción del mensaje de corrección que está formada por tres partes:

1. **Visualización de la solución enviada.** Aparece las expresiones que el alumno ha enviado a corregir.
2. **Resultado obtenido.** Esta parte sólo aparecerá si la solución enviada es sintácticamente correcta y las operaciones se han podido ejecutar sobre los valores de las tablas especificados en el problema. Se visualiza el resultado obtenido sobre el último test de pruebas ejecutado.
3. **Resultado de la corrección.** Con el mensaje de error en color amarillo si se trata de un error sintáctico y color teja si se trata de un error de resultado. Si el problema es correcto, de color azul visualizaremos el mensaje de correcto.

Estàs a » ACME » Inici » Bases de dades » ÀLGEBRA RELACIONAL » Exercici 1

La solució enviada és:

```
R1 ← σNOMDEPART='VENDES'(DEPARTAMENT)
R2 ← R1 ⋈NUMD=NUMDEP(EMPLEAT)
R3 ← πDNI,NOM(R2)
```

1

El resultat obtingut és:

DNI	NOM
555	ANNA
666	POL
777	PAU
888	MARIA

2

El resultat de la correcció és:

LA RELACIÓ RESULTANT TÉ DIFERENT NOMBRE ATRIBUTS

3

Continuar

Figura 8.4 Interfaz de mensajes del módulo ACME-AR

### 8.5.5 El módulo corrector ACME-AR

El tercer apartado a tratar es el diseño e implementación del módulo corrector. Una vez el alumno ha enviado su solución, el sistema se ocupa de su corrección. El objetivo de este módulo es comprobar si el resultado de la ejecución de las expresiones del alumno es el esperado según el juego de pruebas especificado en el enunciado del problema. El corrector actúa en tres fases: análisis sintáctico, ejecución de operaciones y comprobación del resultado. La Figura 8.5 muestra las tres fases.

## PROCESO CORRECCIÓN CONSULTAS ÁLGEBRA RELACIONAL

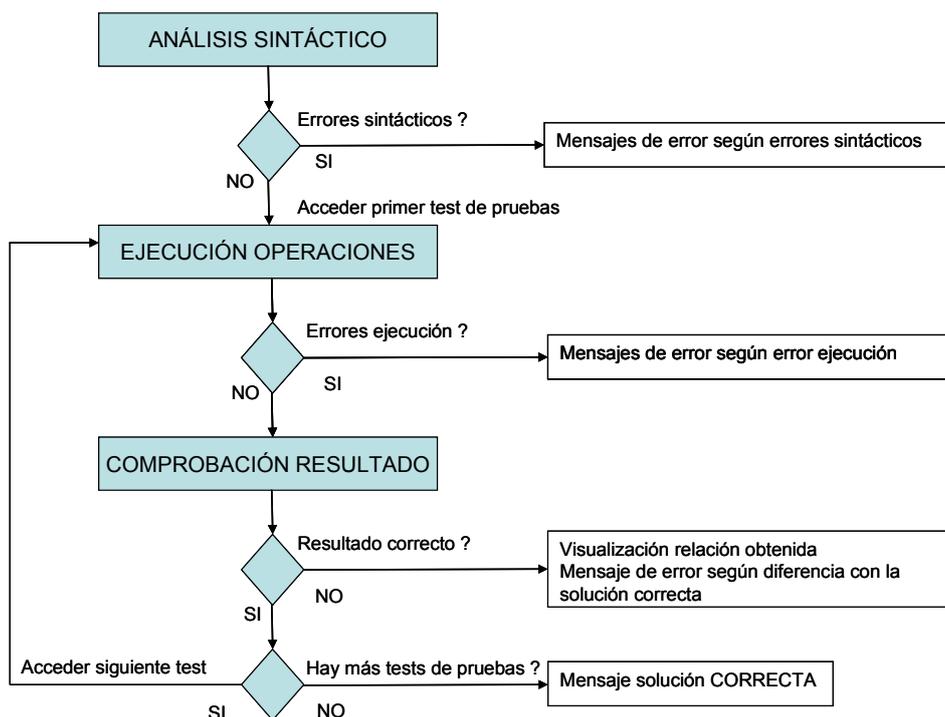


Figura 8.5 Proceso de corrección en ACME-AR.

### Análisis sintáctico

El sistema recoge la información asociada a las distintas expresiones de álgebra relacional enviadas por el alumno y las analiza y comprueba que sean correctas según la notación usada. La notación que se ha utilizado es la definida en [EN07], que es la más utilizada y que se ha descrito en el apartado 8.2 junto con la descripción de las operaciones. Así pues en este módulo se comprueba que cada una de las expresiones formuladas por el alumno sean sintácticamente correctas. Para ello se identifican la/s operaciones que forman parte de la expresión y se verifica su validez. En caso de error se envía el mensaje con el error asociado que se visualiza en color amarillo en la interfaz de salida. Para no mostrar la larga lista de mensajes que se pueden producir, mostramos los más relevantes:

- Tabla inexistente
- Falta relación en la expresión
- Falta el campo de la derecha/izquierda en la operación de reunión
- Error en operador lógico
- Error en operador relacional
- Falta paréntesis
- Falta función
- Falta operación
- Falta campo en función agregada
- Error faltan campos para la operación
- ....

En estos casos, el alumno a partir de la misma solución enviada, y sin necesidad de reescribirla toda de nuevo, podrá corregir el error detectado. En caso que sintácticamente las expresiones enviadas sean correctas se pasa al siguiente proceso, consistente en ir ejecutando las expresiones en el orden que figura en la solución.

## Ejecución de operaciones

Una misma consulta de álgebra relacional se puede solucionar de muchas formas distintas y todas ellas correctas, tal como hemos visto en el ejemplo de la sección 8.2. En este caso, a diferencia de los módulos correctores vistos anteriormente, no tenemos un acotado número de soluciones correctas, sino que las soluciones correctas son muchas y no predecibles. Así pues, para determinar si una solución es correcta, la estrategia que hemos seguido consiste en ejecutar cada una de las operaciones indicadas en la solución del alumno sobre unos valores concretos de diferentes tests de pruebas, especificados en la estructura del problema base guardada en el repositorio.

Para desarrollar este módulo la primera tarea a realizar consiste en la implementación de cada una de las operaciones y funciones del álgebra relacional. Cada uno de estas operaciones realiza la función indicada en el apartado 8.2. De esta forma se han implementado los algoritmos que nos permiten realizar las operaciones de selección, proyección, reunión, reunión natural, unión, intersección, diferencia, renombrar y también los que implementan las funciones agregadas (máximo, mínimo, contar y promedio) con o sin agrupación previa.

El módulo de ejecución va ejecutando cada una de las operaciones sobre los valores de cada uno de los test de pruebas utilizando los valores especificados de cada tabla del problema base. Partiendo de la primera expresión se van ejecutando sus operaciones en el orden correspondiente y se van creando las relaciones intermedias (que por defecto se van llamando R1, R2, ...). Estas nuevas relaciones serán utilizadas en la ejecución de las operaciones siguientes. De esta forma se van obteniendo nuevas relaciones hasta llegar a la relación final que representa el resultado de la consulta.

En este proceso de ejecución también nos podemos encontrar con errores que no nos permitan llevar a cabo la operación requerida. En este caso se envía el error con el mensaje correspondiente que se visualiza de color rojo en la interfaz de salida. Algunos de los mensajes que se facilitan en caso de error son:

- Error en la operación de reunión
- Error en la operación de reunión natural
- Error en la operación intersección
- Error en la operación proyección
- Error en la operación de selección
- ....

En el caso en que no se haya producido ningún error en la ejecución de las distintas operaciones se habrá obtenido una relación final con  $n$  tuplas y  $m$  campos. Para saber si la relación final obtenida con la solución del alumno es correcta ya sólo nos faltará comprobar si se corresponde con la guardada en el repositorio.

## Comprobación del resultado

Una vez se dispone de la relación resultante obtenida con la ejecución de las operaciones ya sólo nos queda comprobar si es correcta. Para ello se compara esta relación con la guardada en el repositorio del problema. En esta comprobación nos podemos encontrar con la casuística siguiente:

- Las dos tablas comparadas tienen distinto número de filas i/o columnas. En este caso la solución es incorrecta y se da el mensaje de error correspondiente:
  - Tabla resultante con distinto número de filas
  - Tabla resultante con distinto número de columnas
  - Tabla resultante con distinto número de filas y columnas
- Las tablas comparadas tienen el mismo número de filas y columnas. En este caso iremos comprobando si los valores de cada una de los campos son coincidentes. Si efectivamente es así la solución será correcta para este test de pruebas.
- Las tablas comparadas tienen el mismo número de filas y columnas pero los valores sean distintos. En este caso se hará la comprobación de que los valores no estén en distinto orden, ya

sea porque los campos estén desordenados o porque las diferentes tuplas estén en distinto orden. En este caso se comprueba para cada tupla si los valores coinciden con los esperados a pesar del orden distinto. En caso de no coincidir se dará el mensaje de ‘Valores resultantes distintos’ y en caso de coincidir la solución será correcta para este test de pruebas.

- Para evitar posibles soluciones correctas “por casualidad” se pasan los diferentes tests de pruebas especificados en el problema base. Si con la ejecución de todos los test de pruebas se obtienen soluciones correctas consideramos la solución correcta y así lo notificamos al alumno. En caso de que un test de pruebas nos dé un resultado incorrecto lo notificamos como tal al alumno.

### **8.5.6 Puntuación automática de consultas de álgebra relacional**

En el caso de las consultas de álgebra relacional creemos que su puntuación sólo pasa por considerar si se ha resuelto de forma correcta o no, de forma parecida a un programa de ordenador. Para la realización de pruebas de validación/exámenes, nos interesa comprobar si el alumno es capaz de llegar a la solución correcta. Ahora bien es frecuente que al escribir una consulta el alumno cometa pequeños fallos que el sistema detectará y comunicará al alumno. Para solventar este tipo de errores facilitaremos al alumno un número restringido de intentos con los que llegar a la solución correcta. En ningún caso este número de intentos tiene que permitir al alumno llegar a la solución correcta a base de “prueba y error”.

ACME ya nos permite establecer el número de intentos a un determinado tipo de actividades, con lo cual este sistema de evaluación correcto/incorrecto ya queda garantizado por la plataforma.

## **8.6 EVALUACIÓN DEL MÓDULO**

En este apartado se comenta la preparación de ejercicios y a continuación se detallan las actividades realizadas y los resultados obtenidos.

### **8.6.1 Preparación de ejercicios de álgebra relacional**

La implementación de este módulo finalizó en el 2007 y fue probado en un curso piloto durante este mismo curso. A partir del curso 2009/2010 ya ha sido utilizado en la asignatura de Bases de Datos que se imparte en el primer semestre del curso en la carrera de ITIG. Para ello se han realizado 3 problemas base. El primer problema estaba formado por 10 consultas y los dos restantes por 25 consultas cada uno. Las diez consultas del primer problema son muy parecidas a las realizadas en clase, mientras que las consultas de los otros dos problemas base son ya más complejas.

Cada uno de estos ejercicios sigue la estructura definida en 8.5.3. Cada problema está formado por un conjunto de consultas que se han escrito de forma clara y concisa, evitando todo tipo de ambigüedades. Los datos entrados para la ejecución de las consultas son considerables y consistentes, para que la ejecución de las consultas pueda realizarse sin ninguna dificultad. Finalmente comentar que sólo se ha utilizado un solo test de pruebas para la ejecución de las consultas, conscientes de que debemos especificar más tests de pruebas para evitar posibles resultados correctos por casualidad.

Una vez escrito cada problema el sistema ha verificado que su sintaxis fuese la establecida para este tipo de ejercicios. Una vez la sintaxis es la correcta se ha probado para que las soluciones fuesen correctas. Como se puede apreciar el tiempo para la creación y testeo de estos problemas es considerable, siendo este uno de los problemas de los sistemas CBA, pero hay que mencionar que una vez en funcionamiento disponemos de un problema para ser utilizado siempre que queramos.

## 8.6.2 Actividades realizadas y resultados obtenidos

El entorno de álgebra relacional se ha utilizado en la asignatura de Bases de Datos de ITIG impartida en el primer semestre del curso 2009/10.

La metodología seguida ha consistido en impartir unas sesiones presenciales en las que se ha explicado el álgebra relacional y se han realizado diferentes ejemplos de consultas para afianzar los conceptos teóricos. Una vez concluido el tema se ha asignado a cada alumno una lista de seis consultas de álgebra relacional. La primera consulta era la más fácil y correspondía al primer problema base. Además de esta consulta se le asignaban dos más del segundo problema y tres más del tercero. Así pues, se ha utilizado como un entorno de blended learning, con el objetivo de evaluar los niveles cognitivos de aplicación y análisis de la taxonomía de Bloom. El sistema de asignación de las consultas a los cuadernos de los alumnos (véase Figura 8.2) hace que las consultas asignadas a los alumnos sean distintas, aunque al no disponer de una muy extensa colección de consultas se repiten algunas. La realización de estos ejercicios se tiene en cuenta para la nota final y está programada como una actividad más de evaluación continuada.

Los primeros resultados obtenidos nos confirman la buena aceptación que ha tenido por parte de los estudiantes. Nos confirman la facilidad de uso y que no han tenido problemas en su utilización. Para una mayor comprensión de lo que hace cada operación se sugirió a los alumnos que fueran enviando las expresiones a corregir a medida que las iban entrando. El sistema les diría que no eran correctas pero ellos visualizarían los campos retornados y tendrían una referencia de si lo estaban haciendo bien. Sólo tendríamos en cuenta si al final se llegaba a la solución correcta, independientemente del número de intentos. Los resultados obtenidos se muestran en la Tabla 8.1. Podemos observar que el 87,5% de los enunciados fueron leídos por los alumnos y se enviaron soluciones al 71,8% de ellos. Las consultas contestadas correctamente fueron el 62,5% del total. Comentar que de los 32 alumnos, 8 no enviaron soluciones.

ALUMNOS	PROBLEMAS	% EJERCICIOS LEIDOS	% SOLUCIONES ENVIADAS	% SOLUCIONES CORRECTAS	% INTENTOS PARA LLEGAR A LA SOLUCIÓN CORRECTA			
					1	2	3	>3
32	6	87,5%	71,8%	62,5%	19,0%	34,5%	16,6%	29,9%

Tabla 8.1 Resultados obtenidos con ACME-AR.

Por otra parte, y debido a la dificultad de algunas consultas, no todos los alumnos las resolvieron correctamente. Catorce participantes resolvieron correctamente todos los problemas, tal como refleja la Tabla 8.2.

ACME-AR	PROBLEMAS RESUELTOS CORRECTAMENTE						
	6	5	4	3	2	1	0
NUM. ALUMNOS	14	4	2	2	0	2	8
% ALUMNOS	43,75%	12,50%	6,25%	6,25%	0,00%	6,25%	25,00%

Tabla 8.2 Problemas resueltos con ACME-AR.

Si no tenemos en cuenta estos alumnos que ni tan siquiera leyeron los enunciados, los resultados obtenidos son muy positivos. Somos conscientes que disponemos de pocos datos para hacer un estudio más extenso, pero las primeras impresiones son buenas.

## 8.7 APORTACIONES Y CONCLUSIONES

Se ha propuesto, definido y desarrollado el entorno ACME-AR, un entorno CBA para la corrección/ evaluación de consultas de álgebra relacional, se ha integrado en la plataforma ACME y se ha creado una colección de problemas. Hasta donde llega nuestro conocimiento no hay otros trabajos que hayan

desarrollado correctores con las características de ACME-AR. Este corrector se está utilizando en la asignatura de “Bases de datos” desde el curso 2009/2010. Las principales ventajas que presenta son:

- Permite llevar a cabo la evaluación formativa del aprendizaje de consultas de álgebra relacional, actuando como un sistema CBA.
- Disminución del trabajo del profesor en cuanto a horas dedicadas a la corrección de este tipo de ejercicios, aunque el esfuerzo de la puesta a punto y diseño de ejercicios es considerable.
- Muchos más alumnos se ejercitan en la realización de consultas de álgebra relacional y obtienen el resultado inmediato de la corrección.
- Mayor feedback alumno/profesor y se aclaran muchas dudas que surgen en el momento de la resolución de los ejercicios.
- Permite un seguimiento muy estricto del trabajo del alumno.
- Evaluación automática del trabajo realizado por el alumno.

Este corrector ha dado lugar al artículo “**An automatic correction tool for relational Algebra Queries**” presentado en la International Conference on Computational Science and its Applications (ICCSA 2007) y publicado en LNCS 4706 pág. 861-872, referenciado en [SBP+07].

## CAPÍTULO 9. MÓDULO DE SENTENCIAS SQL

En este capítulo presentaremos el último de los módulos correctores desarrollados referente a la corrección de sentencias SQL. Este módulo dará soporte al aprendizaje de cualquier comando del lenguaje SQL.

Este capítulo se ha estructurado de la siguiente forma. En la sección 9.1 se realiza una breve introducción al tema. En la sección 9.2 se describen los comandos básicos de SQL. En la sección 9.3 describimos las necesidades y la problemática de la enseñanza del SQL. En la sección 9.4 se realiza un breve resumen de los entornos existentes para este tipo de problemas. En la sección 9.5 se plantea el módulo, se detalla la solución aportada y su integración a la plataforma ACME. Finalmente en las secciones 9.6 y 9.7 se describen los resultados académicos conseguidos y las aportaciones y conclusiones a las que hemos llegado.

### 9.1 INTRODUCCIÓN

El lenguaje de consulta estándar **SQL (Structured Query Language)** es un lenguaje estándar de bases de datos relacionales que ha tenido una amplia aceptación, siendo el más utilizado en el trabajo diario con una base de datos relacional.

Los orígenes del SQL están ligados a los de las bases de datos relacionales. E.Codd [Cod70] propone el modelo relacional y asociado a éste un sub-lenguaje de acceso a los datos basado en el cálculo de predicados. Basándose en estas ideas, los laboratorios de IBM definen el lenguaje SEQUEL (Structured English QUERy Language) que más tarde sería ampliamente implementado por el Sistema de Gestión de Bases de Datos (SGBD) experimental System R desarrollado en 1977 también por IBM. Sin embargo, fue Oracle quien lo introdujo por primera vez en 1979 en un software comercial. El SEQUEL terminaría siendo el predecesor de SQL, siendo éste una versión evolucionada del primero. El SQL pasa a ser el lenguaje por excelencia de los distintos SGBD relacionales surgidos en los años siguientes y es por fin estandarizado en 1986 por el ANSI y en 1987 por la ISO, dando lugar a la primera versión estándar de este lenguaje, el SQL-86 o SQL1. Sin embargo este primer estándar no cubre todas las necesidades de los desarrolladores e incluye funcionalidades de definición de almacenamiento que posteriormente se suprimieron. En 1992 sale un nuevo estándar ampliado y revisado del SQL llamado "SQL-92" o "SQL2".

En la actualidad el SQL es el estándar de facto de la inmensa mayoría de los SGBD comerciales. Y, aunque la diversidad de añadidos particulares que incluyen las distintas implementaciones comerciales del lenguaje es amplia, el soporte al estándar SQL-92 es general y muy amplio.

Un lenguaje de base de datos como es el SQL debe permitir al usuario:

- La creación y mantenimiento de la base de datos y de todas sus relaciones (tablas) y restricciones.
- Gestión básica de los datos de la relación (inserción, modificación y borrado).
- Realización de consultas, tanto simples como complejas.

El éxito de SQL se basa en la sencillez y potencia para la obtención de los datos que cumplan unas determinadas condiciones. En este aspecto SQL es un lenguaje declarativo, en el que especificamos qué queremos obtener y el sistema ya se ocupa de interpretar la sentencia, optimizarla y devolvernos el resultado. En ningún caso se debe especificar el cómo hacerlo, sino simplemente qué queremos obtener. SQL utiliza los términos tabla, fila y columna para los términos habituales de relación, tupla y atributo del modelo relacional formal.

El aprendizaje de SQL es una de las competencias básicas que en materia de bases de datos debe adquirir el alumno. La creación, el mantenimiento (inserción, modificación, eliminación) y la obtención de resultados (consultas) de los datos de una base de datos a través de SQL son tareas que el alumno debe aprender perfectamente. Todas las asignaturas de bases de datos cuentan con un apartado dedicado al trabajo con SQL. Esta materia se imparte a menudo en aulas informáticas, donde se simultanean explicaciones teóricas y de sintaxis de las diferentes sentencias con la práctica y la ejecución en el ordenador de sentencias propuestas por el profesor.

## 9.2 DESCRIPCIÓN DEL LENGUAJE SQL

En este capítulo se describen los conceptos más básicos de SQL. En la bibliografía específica de SQL [GW00] y [RMR+02] o de fundamentos de bases de datos [EN07] [Dat04] [Kro04] se desarrollan detalladamente todos los aspectos de SQL. En una clasificación muy general de los comandos de SQL podemos distinguir:

- Comandos que permiten la creación, modificación y borrado de tablas.
- Comandos para la actualización de tablas que nos permiten insertar, modificar y eliminar filas de una tabla.
- Sentencia SELECT para la realización de todo tipo de consultas.

En los siguientes tres apartados se hace un resumen muy breve de las sentencias más importantes.

### 9.2.1 Creación, modificación y borrado de tablas

Los comandos básicos para la gestión del esquema de la base de datos son:

**CREATE TABLE.** El comando CREATE TABLE se utiliza para especificar una nueva relación, asignándole un nombre y sus atributos con sus restricciones iniciales. Primero se especifican sus atributos, a cada uno de los cuales se le asigna un nombre, un tipo de datos y cualquier restricción de atributo como por ejemplo NOT NULL. Las restricciones de clave, integridad de entidad e integridad referencial pueden especificarse en el mismo CREATE TABLE, después de haber declarado los atributos, o pueden añadirse más tarde con la sentencia ALTER TABLE. Como ejemplo:

```
CREATE TABLE DEPARTAMENTO
  (NombreDpto VARCHAR (15) NOT NULL,
  NumeroDpto INT NOT NULL,
  Nifdirector CHAR (9) NOT NULL,
  Fechaaltadirector DATE,
  PRIMARY KEY (NumeroDpto),
  FOREIGN KEY (Nifdirector) REFERENCES EMPLEADO(Nif));
```

**ALTER TABLE.** El comando ALTER TABLE permite modificar la definición de una tabla. Permite por ejemplo añadir o eliminar un atributo, cambiar la definición de una columna o añadir o eliminar restricciones. Como ejemplos: ALTER TABLE EMPLEADO DROP COLUMN comarca. Otro ejemplo:

```
ALTER TABLE EMPLEADO
  ADD CONSTRAINT fk_empl_dep_num
  FOREIGN KEY (empl_dep_num)
  REFERENCES DEPARTMENT (dep_number);
```

**DROP TABLE.** El comando DROP se utiliza para eliminar elementos del esquema, como tablas o restricciones. Como ejemplo: DROP TABLE EMPLEADO

### 9.2.2 Inserción, actualización y borrado de filas

Los comandos básicos para la gestión de los valores de las filas dentro de una tabla son:

**INSERT.** En la forma más básica se utiliza para añadir una fila (tupla) a una relación. Se debe especificar el nombre de la relación y la lista de valores que tendrá la fila.

Ejemplo: INSERT INTO EMPLEADO VALUES ('Pedro', 'García', 'Vidal', .....)

**UPDATE.** El comando UPDATE se utiliza para modificar los valores de una o más tuplas seleccionadas. Puede incluir una cláusula WHERE que selecciona las tuplas que se van a modificar.

Ejemplo: UPDATE PROYECTO SET Presupuesto=50000 WHERE NumeroProyecto= 10

**DELETE.** El comando DELETE elimina tuplas de una relación. Puede incluir una cláusula WHERE que determina que tuplas se van a eliminar.

Ejemplo: DELETE FROM EMPLEADO WHERE Nif='12345678N'

### 9.2.3 Consultas

SELECT es la sentencia básica de SQL para recuperar la información de una base de datos. La sentencia tiene muchas opciones y versiones para la recuperación de la información. La estructura básica de una sentencia SQL es la siguiente:

```
SELECT      <lista de atributos>
FROM        <lista de tablas>
WHERE       <condiciones>
```

Aunque se puede completar con otras muchas cláusulas como GROUP BY, HAVING, ORDER BY, DESC, ASC, DISTINCT, LIKE, BETWEEN, IN, AND, OR, EXISTS, COUNT, SUM, MAX, MIN, AVG, INNER JOIN, etc. Además se pueden renombrar las tablas (asignación de alias), se pueden realizar consultas anidadas, etc. Con todo ello se pueden realizar consultas muy complejas y de difícil resolución. Además una misma consulta se puede realizar de formas muy distintas y todas ellas de forma correcta. En los siguientes tres ejemplos se muestra una misma consulta SQL para obtener los empleados del Departamento de 'Vendas' ordenados por Apellido1 y Nombre.

Ejemplo1:

```
SELECT      Nombre, Apellido1, Dirección
FROM        EMPLEADO, DEPARTAMENTO
WHERE       NombreDep = 'Vendas' and NumDptoempl = Numdep
ORDER BY   Apellido1, Nombre
```

Ejemplo2:

```
SELECT      E.Nombre, E.Apellido1, E.Dirección
FROM        EMPLEADO AS E, DEPARTAMENTO AS D
WHERE       D.NombreDep = 'Vendas' and E.NumDptoempl = D.Numdep
ORDER BY   E.Apellido1, E.Nombre
```

Ejemplo3:

```
SELECT      Nombre, Apellido1, Dirección
FROM        (EMPLEADO JOIN DEPARTAMENTO ON NumDptoempl = Numdep)
WHERE       NombreDep = 'Vendas'
ORDER BY   Apellido1, Nombre
```

Precisamente el hecho que una misma consulta se pueda solucionar de muchas formas distintas será uno de los factores a tener en cuenta a la hora de desarrollar el método de corrección de esta tipología de problemas.

## 9.3 NECESIDADES Y PROBLEMÁTICA DE LA ENSEÑANZA DE SQL

El dominio del lenguaje SQL es una de las competencias que el alumno debe adquirir en materia de bases de datos. En el ámbito del desarrollo de software en que se mueven los egresados de las carreras informáticas es indispensable el conocimiento en profundidad de este lenguaje. De aquí, la gran importancia de un buen aprendizaje de SQL.

Para la enseñanza de SQL es necesario que el alumno tenga unos sólidos conocimientos del modelo relacional. Para la realización de consultas mediante el comando SELECT es recomendable que haya formulado consultas en álgebra relacional que le permiten entender mejor como actúan. La enseñanza tradicional del lenguaje SQL consiste generalmente de sesiones prácticas en donde el profesor va explicando las diferentes acciones a realizar sobre una BD junto con los comandos para ejecutarlas. Se explican los comandos necesarios para la creación de tablas y el alumno crea un esquema de base de datos. Se explican los comandos de actualización y el alumno inserta, modifica y elimina registros. Ahora bien, en el apartado de consultas se dedica mucho más tiempo debido a su importancia y a la variedad de consultas que se pueden resolver. El profesor suele disponer de una base de datos con muchos registros para poder formular todo tipo de consultas. Sobre esta base de datos el profesor va explicando las diferentes posibilidades que ofrece la sentencia, que va simultaneando con consultas que deben realizar los alumnos. En esta fase los alumnos van simplemente aplicando los comandos en ejemplos muy parecidos a los explicados por el profesor. En referencia a la taxonomía de Bloom estaríamos centrados en el nivel de aplicación de conocimientos. A medida que se va desarrollando el tema, el profesor va introduciendo consultas más complejas en las que el alumno ya debe analizar cuales son las cláusulas que deberá aplicar para la resolución de una consulta determinada. En esta fase el alumno deberá tanto analizar como aplicar los conocimientos adquiridos. Tal como se ha comentado, en un entorno CBA en el cual se pretenda evaluar niveles de conocimiento de aplicación y análisis de la taxonomía de Bloom, se requieren de herramientas mucho más especializadas ya que con las típicas preguntas de respuesta fija, difícilmente podemos evaluar estos niveles.

Después de varios años impartiendo docencia en cursos de BD hemos observado y constatado las dificultades que tienen los alumnos en el aprendizaje de SQL. Durante las sesiones prácticas después de que el profesor haya explicado un determinado comando el alumno sabe aplicarlo sin demasiada dificultad. Ahora bien, a medida que se van introduciendo nuevos conceptos que se deben utilizar conjuntamente con los explicados en sesiones anteriores, tienen serias dificultades en formular correctamente las sentencias. Ello nos lleva a pensar que, al igual que con la programación, se requiere de un tiempo de aprendizaje que pasa por la resolución de muchas consultas y la revisión oportuna por parte del profesor. En muchas universidades, el profesor facilita un listado de consultas que el alumno debe resolver junto con el resultado que debe obtener. Sobre la base de datos facilitada el alumno va resolviendo las consultas y preguntando al profesor en caso de duda. Entre las deficiencias que presenta este método podemos citar:

- Los alumnos con mayores dificultades de aprendizaje no suelen terminar los ejercicios propuestos. Simplemente realizan los ejercicios más sencillos.
- Mientras el profesor atiende una consulta, otros alumnos se copian los ejercicios de los compañeros que los han resuelto.
- En la mayoría de ocasiones el alumno no dispone ni de la base de datos ni del SGBD para poder practicar fuera del aula.
- Desconoce si son correctas las consultas realizadas en lápiz y papel resueltas fuera del aula.
- Sólo se trabaja con una base de datos.

Para una buena evaluación formativa sería deseable que el alumno pudiera disponer de muchas sesiones de laboratorio, con un profesor que le pudiera resolver dudas y orientarle en las consultas que no supiera realizar. En muchas ocasiones esta situación no se produce por la limitación de las sesiones, lo extenso de programa, etc. y porque el alumno tampoco suele acudir a tutorías.

Para solventar estas deficiencias sería interesante disponer de un entorno de soporte a las sesiones presenciales en donde el alumno pudiera ejercitarse en la resolución de cualquier comando SQL.

## 9.4 CORRECCIÓN AUTOMÁTICA DE SENTENCIAS SQL

El SQL es el lenguaje utilizado en todos los sistemas de BD relacionales. Desde que se desarrollaron los primeros SGBD que soportaban el modelo hasta nuestros días ha sido necesario formar a gran cantidad de personas en SQL. En los últimos años con el auge del e-learning e Internet se han desarrollado muchos entornos para su aprendizaje tal como hemos visto en la sección 2.4.6. La mayoría de estos entornos sólo soportan la sentencia SELECT y cada uno de ellos se ha desarrollado para un contexto determinado, con sus particularidades específicas. Desde el punto de vista académico, los trabajos más interesantes son el WinRDBI [WinRDBI], el SQLTutor [Mit98a] [Mit98b] [Mit03] y LEARN-SQL [ARU+08] [ABC+09]. WinRDBI se ha utilizado en muchas universidades como herramienta de entrenamiento. El alumno construye su base de datos, entra sus datos y a partir de aquí puede practicar en la resolución de consultas. No es un corrector y hay que descargar e instalar el software. Por su parte SQLTutor y LEARN-SQL son las más parecidas a la que hemos desarrollado.

Además de estas herramientas, al ser SQL un lenguaje estándar utilizado ampliamente en el sector de desarrollo de software, se han desarrollado muchos tutoriales y herramientas para su aprendizaje y práctica. La mayor parte de ellos están orientados a un determinado SGBD (Oracle, SQLServer, MySQL, etc. ) aunque también hay entornos web que ofrecen buenos cursos online. Algunos de estos productos son software libre y otros se ofrecen como productos comerciales. En la sección 2.4.6 se detallan algunos de estos productos.

Nuestro objetivo en este apartado ha sido desarrollar un entorno que permitiera la corrección automatizada de sentencias SQL, en la misma línea que los correctores vistos anteriormente. Dada la complejidad e importancia del SQL, se quería facilitar al máximo su utilización, sin necesidad de instalar ningún software ni tan siquiera la necesidad de disponer de un SGBD.

## 9.5 ACME-SQL. MÓDULO DE SENTENCIAS SQL

En esta sección se detalla el módulo referente a la evaluación de sentencias SQL que denotamos como ACME-SQL. En la sección 9.5.1 se determinan los requisitos que deberá satisfacer el sistema. En la sección 9.5.2 se presenta una visión general de la solución desarrollada. En las cuatro siguientes secciones se detallan los aspectos más técnicos del módulo: la estructura de un problema SQL, su interfaz, el módulo de corrección y feedback y finalmente en la sección 9.5.6 la puntuación del problema.

### 9.5.1 Requisitos del sistema ACME-SQL

El problema que planteamos consiste en desarrollar un sistema que de forma totalmente automática sea capaz de corregir los distintos comandos de SQL. Aunque el sistema debe soportar cualquier comando SQL, siempre en la enseñanza de SQL se da mucha más importancia a la realización de consultas mediante la sentencia SELECT y a la gran cantidad de posibilidades que ofrece. Al igual que con las expresiones de álgebra relacional, una misma consulta puede tener muchas e imprevisibles respuestas válidas. El sistema deberá cumplir con los requisitos siguientes:

- El sistema deberá poder corregir cualquier comando SQL. En la sección 9.2 se han mostrado unos ejemplos muy sencillos, pero el entorno a desarrollar tiene que soportar cualquier comando y consulta por compleja que sea.
- El sistema no requerirá tener instalado ningún SGBD por parte del alumno. Sólo se requerirá un navegador y una conexión Internet.
- Disponer de una interfaz apropiada para la entrada de cualquier comando SQL. Debe permitir:
  - Visualizar la consulta a realizar.
  - Visualizar cada tabla, facilitando información de sus campos y propiedades.
  - Poder editar el comando.
  - Disponer de una interfaz para el retorno de feedback.
- Disponer de un corrector apropiado. Este corrector deberá tener tres partes diferenciadas:

- Disponer de un módulo de verificación, de forma que cuando el alumno envíe la solución, este módulo recogerá la sentencia SQL y comparará si la solución enviada es igual o equivalente a una de las contempladas como correctas en el enunciado del problema. Si no son iguales o equivalentes el sistema considerará la sentencia SQL como incorrecta, a no ser que sea una SELECT.
  - Módulo de ejecución de SELECT. Tal como hemos visto en el ejemplo de la sección 9.2, una misma consulta se puede solucionar de muchas formas distintas y todas ellas de forma correcta. En este caso, no tenemos un acotado número de soluciones que podemos afirmar son correctas, sino que las soluciones correctas son muchas y no predecibles. Así pues, para determinar si una solución es correcta será necesario ejecutar, sobre varias bases de datos de test, la consulta correcta almacenada en el repositorio y la del alumno.
  - Disponer de un módulo comparador. Verificará si los resultados obtenidos en las ejecuciones previas son iguales. Tendremos en cuenta que a lo mejor el orden de los atributos es distinto y también puede darse el caso que el orden de las tuplas no sea el mismo, pero los valores retornados han de ser los esperados.
- Aunque es necesario crear bases de datos en el servidor, no se quiere depender del administrador del sistema para la creación de nuevas bases de datos. Para ello dentro de la estructura del problema se incluirán todas las sentencias necesarias para su creación. Si al tener que ejecutar una consulta el sistema no dispone de la base de datos pertinente (la primera vez) ésta se creará automáticamente con la información asociada en la estructura del problema. Si la base de datos ya está creada, se ejecutará la consulta directamente. De esta forma se quiere tener total independencia del administrador del sistema.
  - La comparación de la solución del alumno con las del profesor sería la única prueba que se realizaría para aquellas sentencias que pueden modificar la estructura de la base de datos, las tablas o los registros. O sea, para todas las sentencias SQL distintas a una SELECT sólo se utilizaría este método ya que si por ejemplo permitiéramos la modificación de la estructura de la tabla o el borrado/modificación de las filas, entonces la base de datos podría quedar inutilizable muy rápidamente.
  - El sistema deberá integrarse a la plataforma ACME, con lo cual se compartirán todas las funcionalidades ACME, como por ejemplo la gestión de cursos, asignaturas, temas, actividades, alumnos, envío de mensajes, foros, chats, etc.

### 9.5.2 Análisis inicial de ACME-SQL

Dada la complejidad de SQL y la gran variedad de comandos y cláusulas que pueden llevar asociadas, el sistema de corrección diseñado se ha de basar en un sistema mixto comparación/ejecución. En una primera fase de la corrección se ha optado por la comparación del comando enviado por el alumno con los comandos correctos desarrollados por el profesor y almacenados en el repositorio. Si se detecta que son iguales o equivalentes se considerará la solución como correcta sin necesidad de ejecutarlos.

Ahora bien, en un entorno de aprendizaje de SQL, la resolución de todo tipo de consultas con la sentencia SELECT suele ser el aspecto al que se dedican más esfuerzos. Por otra parte la gran variedad de consultas que permite la sentencia SELECT junto con la gran cantidad de cláusulas que podemos utilizar hace de la SELECT la sentencia que requiere mayor tiempo de aprendizaje. Además, el hecho que una misma consulta se puede realizar de muchas maneras distintas y no predecibles (véase sección 9.2), hace mucho más complejo el proceso de corrección. Así pues, en el caso de la sentencia SELECT puede no ser suficiente la corrección por comparación y necesitamos la ejecución y posterior verificación de los resultados para determinar si es correcta. En este caso, se ejecutará la sentencia SELECT correcta entrada por el profesor y que está guardada en el repositorio sobre varias bases de datos de pruebas con valores distintos. Sobre las mismas bases de datos se ejecutará la sentencia SELECT enviada por el alumno. Finalmente se compararán los resultados obtenidos y de ser iguales se considerará la solución enviada como correcta.

Evidentemente todo el sistema debe estar integrado en la plataforma ACME de forma que tanto profesores como alumnos dispongan de las funcionalidades de la plataforma. Como en los módulos anteriores, la plataforma guardará cada solución enviada por el alumno a un determinado ejercicio.

Seguindo la filosofía ACME, si la solución es correcta, el sistema lo notificará al alumno mediante un “Correcto”. En caso de que el corrector determine que es incorrecta facilitará pequeñas notificaciones para ayudar al alumno a encontrar el error y poder enviar una nueva solución. Entre estas indicaciones tendremos que distinguir entre las que sean producidas por comandos sintácticamente mal contruidos de los que llevan a resultados incorrectos. En este segundo caso, si se trata de una SELECT, se mostrará al alumno los valores retornados en la ejecución de su consulta junto con el mensaje de error apropiado.

Al igual que en los correctores anteriores el módulo desarrollado debe estar ligado a la plataforma ACME a la que se tiene que integrar. Para ello es necesario desarrollar:

- **La estructura el problema.** Cada problema se guarda en un fichero en el repositorio con una determinada estructura y con la información para su visualización y corrección. En el caso de los ejercicios SQL debe estar formado por los distintos enunciados que se deben resolver mediante un comando SQL. Para cada base de datos se realizará un único problema con los distintos enunciados. El profesor mediante un editor cualquiera, o el específico de ACME, editará la estructura del problema.
- **Una interfaz apropiada.** A través de la cual el alumno escribirá el comando SQL que resuelve el ejercicio formulado. Una vez entrado, lo enviará a corregir y el sistema le mandará los mensajes de feedback a través de la interfaz de salida.
- **Un módulo corrector.** Este módulo comprobará si la solución enviada coincide con una de las correctas entradas por el profesor. En este caso se tratará de una solución correcta. En caso de que la solución no coincida y se trate de una SELECT, se ejecutarán tanto la solución del alumno como la del profesor en un SGBD dado sobre varias bases de datos de pruebas y se comprobará si los datos obtenidos son los mismos. En caso afirmativo consideramos que se trata de una solución correcta. El sistema notificará al alumno que su solución es correcta y mostrará la solución óptima entrada por el profesor. Si no coinciden el sistema mostrará la información retornada por la consulta del alumno y la del profesor. Si el SGBD no puede ejecutar la sentencia del alumno por estar mal construida se le notificará mediante un error de sintaxis.

Veamos con más detalle cada una de estas tres partes.

### 9.5.3 Estructura problema SQL

La estructura general de un problema es la que utiliza ACME para visualizar el enunciado de una consulta y que utilizará el módulo corrector para la corrección de las soluciones enviadas por los alumnos. Esta estructura se guarda en un fichero que se puede editar con cualquier editor y se guarda en el repositorio. En este tipo de problemas podemos incluir muchas preguntas/consultas a ejecutarse con la misma base de datos. Así pues cada problema SQL estará formado por las siguientes partes:

```

<identificador de tipo de problema>
<E1> enunciado general </E1>
<P1> enunciado consulta1 </P1>
<S1> solución1 consulta1 </S1>
<S2> solución2 consulta1 </S2>
...
<P2> enunciado consulta2 </P2>
<S1> solución1 consulta2 </S1>
<S2> solución2 consulta2 </S2>
...
.....(***) todos los enunciados con una o más soluciones óptimas)

<BD pruebas 1>
(***) Toda la información para la creación de la BD pruebas 1
....
CREATE's TABLE con el nombre de todas las tablas y de los campos a crear
....
INSERT's que sean necesarios para entrar los valores en los distintas tablas
....
</BD pruebas 1>

```

*(\*\*\* tantos ítems “BD pruebas” como test de prueba se quieran usar\*\*\*)*

En esta estructura vemos que en primer lugar aparece el identificador de problema. En segundo lugar un enunciado general único para el problema. A continuación el enunciado de cada ejercicio/consulta seguido de las posibles soluciones consideradas correctas. Cada problema base de este tipo puede contener muchas consultas. Una vez especificados todos los enunciados de las preguntas/consultas con sus soluciones óptimas se deben especificar todas las sentencias para la creación de las bases de datos de pruebas y los INSERT con los datos necesarios para construir las tablas sobre las cuales poder ejecutar las consultas. Cada base de datos de pruebas estará formada por un conjunto de valores distintos, para evitar al máximo el “acierto por casualidad”. Comentar que estos valores no son visibles ni accesibles a los alumnos, de forma que nunca saben sobre que valores se ejecutarán sus consultas.

Una vez creado el fichero con esta estructura y con las funcionalidades específicas de ACME, el profesor lo incorpora al repositorio. Antes de su incorporación se valida a través del módulo de validación de ejercicios SQL desarrollado. Según el identificador de tipo de problema, el módulo de validación comprueba que la estructura del fichero sea la correcta y a partir de aquí ya se guarda en el repositorio y está disponible para ser utilizado. En caso que este módulo detecte que no se ajusta a la estructura esperada, el sistema dará el mensaje de error correspondiente y no se guardará en el repositorio.

Una vez guardado el problema base en el repositorio ya estará disponible para ser asignado a los cuadernos de los alumnos. Siguiendo las especificaciones de la plataforma ACME, el profesor ya podrá seleccionar los ejercicios y el módulo de generación de cuadernos los asignará a los alumnos. Para este tipo de problemas en que cada uno puede contener muchas consultas, la estrategia seguida por el módulo es la misma que se ha comentado en la sección 8.5.3 y mostrada en la Figura 8.2. De esta forma, los cuadernos de los alumnos son similares, pero todos ellos distintos.

Una vez asignadas las consultas, el alumno las visualizará y las podrá resolver. Una vez resueltas podrá introducir las sentencias obtenidas en el área de edición diseñada. El corrector, tal como se detalla más adelante, utiliza el resto de información del problema para su corrección.

## 9.5.4 Diseño de las interfaces SQL

Para la visualización de los ejercicios y la entrada las sentencias SQL se ha diseñado la interfaz que se muestra en la Figura 9.1. Esta interfaz está formada por 4 zonas:

The screenshot shows a web interface for solving SQL problems. At the top, a green navigation bar contains the text "You are in > ACME > Home > Editor > Exemple SQL > Problem: 1". Below this is a table with four columns: DEPARTMENT, DEPENDENT, DEP\_LOCATIONS, and EMPLOYEE. The DEPARTMENT column has a sub-row with the value PROJECT, and the DEPENDENT column has a sub-row with the value WORKS\_ON. A red box labeled '1' highlights the DEPENDENT and DEP\_LOCATIONS columns. Below the table, the text "Write the SQL sentence that solves the following statement:" is followed by the question in Spanish: "Feu una select que resolgui el següent enunciat". Below this, a red box labeled '2' contains the question in English: "Retrieve the name (first name and last name) of all employees who work for the 'Research' department .". Below the question, the text "Solution:" is followed by a red box labeled '3' containing the SQL query: "SELECT empl\_fname, empl\_lname FROM EMPLOYEE, DEPARTMENT WHERE empl\_dep\_num=dep\_number and dep\_name='Research'". At the bottom left, a green box labeled 'Correct' is followed by a red box labeled '4'.

Figura 9.1 Interfaz entrada sentencias SQL.

1. **Zona de visualización de las tablas de la base de datos.** Seleccionando una tabla se verá toda la información asociada a la tabla que es necesaria para que el alumno pueda resolver la sentencia. La Figura 9.2 muestra la información que se visualiza.
2. **Enunciado de la pregunta a resolver.** Generalmente se tratará de una consulta que el alumno deberá resolver a través de una sentencia SELECT. De todas formas se puede

formular cualquier ejercicio en el que sea necesario responder con cualquier comando SQL.

3. **Área de escritura de las soluciones.** El alumno escribirá libremente la sentencia SQL
4. **Botón de envío a corregir.**

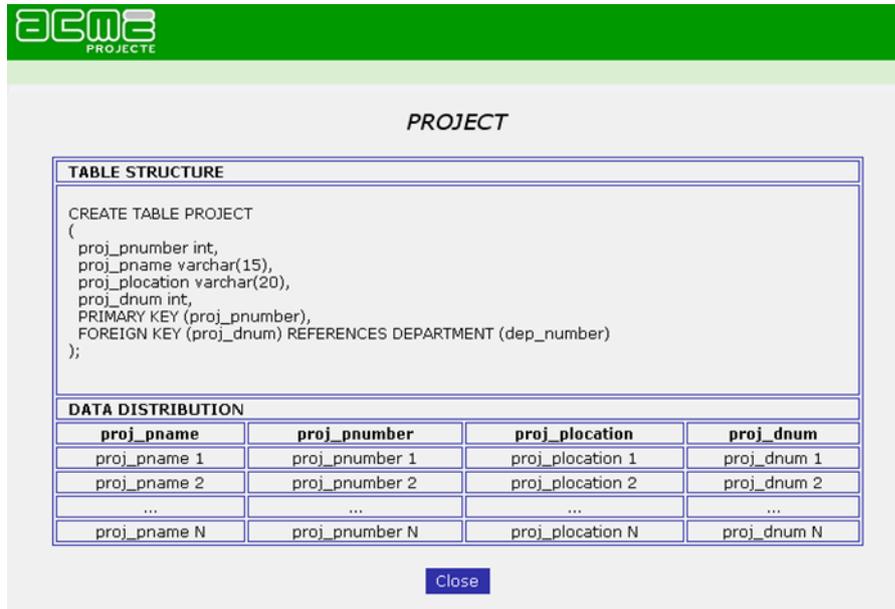


Figura 9.2 Interfaz visualización tablas en ACME-SQL.

La información de las zonas 1 y 2 y la mostrada en la Figura 9.2, se obtiene a partir del problema base almacenado en el repositorio. A partir de esta información el alumno escribe su sentencia SQL en la zona 3. Con el botón “corregir” se envía la solución a la plataforma ACME. Una vez enviada la solución, ACME la corrige automáticamente y nos manda el mensaje correspondiente. La interfaz de recepción del mensaje de corrección (véase Figura 9.3) está formada por tres partes:

1. **Visualización de la solución enviada.**
2. **Resultado de la corrección.** Con el mensaje de error en color amarillo si se trata de un error sintáctico y color teja si se trata de un error de resultado. Si el problema es correcto aparece el mensaje correspondiente.
3. **Resultado obtenido.** Esta parte sólo aparecerá en consultas SELECT sintácticamente correctas que se hayan tenido que ejecutar para verificar el resultado. En caso que la solución sea correcta se visualizará también la solución óptima entrada por el profesor.

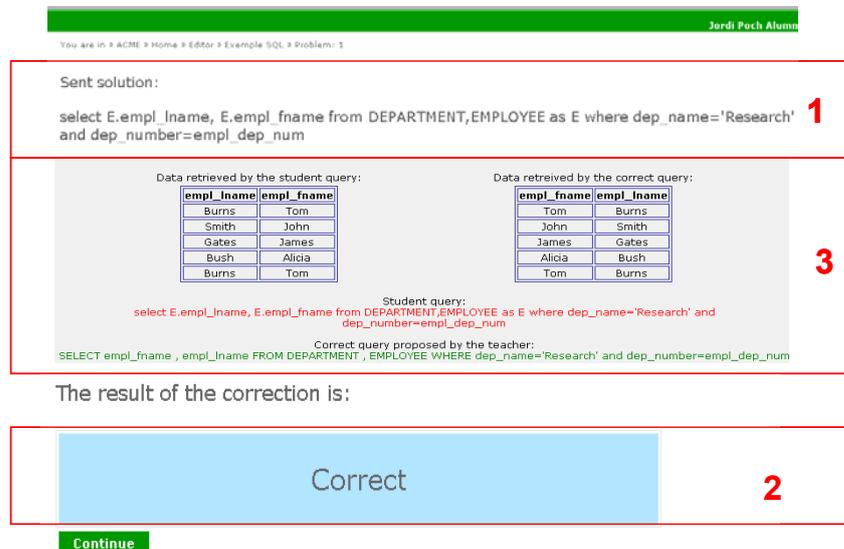


Figura 9.3 Zonas interfaz visualización resultados sentencias SQL.

### 9.5.5 El módulo corrector ACME-SQL

El sistema elegido para el módulo corrector es un sistema híbrido en el que en primer lugar se compara la solución enviada por el alumno con la del profesor. Este sistema se utiliza para todos los comandos SQL, ahora bien, en las consultas SELECT como que una misma consulta se puede formular de muchas maneras distintas, si la solución enviada no es ninguna de las contempladas por el profesor, ejecutaremos las sentencias sobre varias bases de datos de pruebas y compararemos los resultados obtenidos. De esta forma, una vez el alumno ha enviado su solución a corregir, el sistema corrector actúa en tres fases, tal como muestra la Figura 9.4

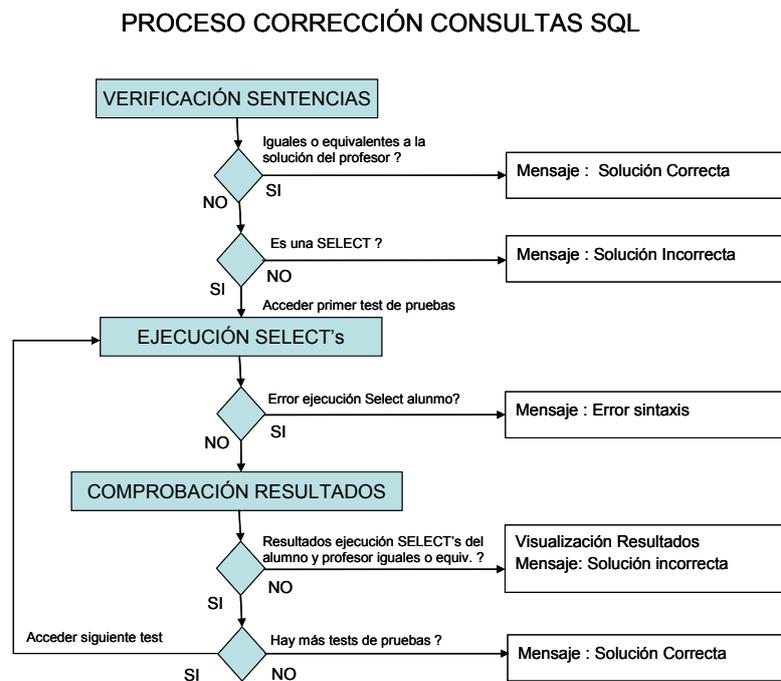


Figura 9.4 Proceso de corrección en ACME-SQL.

#### Verificación de las sentencias

El sistema recibe la sentencia SQL enviada por el alumno al pulsar el botón de corregir. De entrada se comprueba si la sentencia es exactamente igual a alguna de las definidas en la estructura del problema para este ejercicio. Si es así la consideramos correcta. Si no es exactamente igual se comprueba si es equivalente. Se entiende por equivalente si el orden de los campos es distinto pero hay la misma información. Por ejemplo, en una consulta SELECT, se entiende por equivalente si los campos escritos a continuación del SELECT, si el nombre de las tablas después del FROM o las distintas expresiones que vienen a continuación del WHERE están en distinto orden. Si el sistema detecta que son equivalentes, envía el mensaje de “correcto”. De esta forma y sin necesidad de ejecutar la sentencia ya detectamos las soluciones correctas. Esta situación se da en respuestas de ejercicios sencillos y en los que generalmente las soluciones enviadas suelen ser correctas. Si la solución no es igual o equivalente se envía el mensaje de solución incorrecta a no ser que sea una sentencia SELECT. En este caso puede haber muchas soluciones correctas por lo que para comprobar si es correcta ejecutaremos la SELECT enviada por el alumno y la correcta disponible en el repositorio. Una vez ejecutadas las dos compararemos los resultados obtenidos.

## Ejecución de las SELECT's

En el caso que la solución sea una SELECT y que sea distinta (o no equivalente) a las correctas guardadas en la estructura del problema tendremos que ejecutar ambas en un SGBD (Postgres en nuestro caso) y comparar los resultados obtenidos. En el caso que el SGBD nos dé un error en la ejecución de la SELECT del alumno, el sistema mandará el mensaje de “error de sintaxis” al alumno. Para dar una mayor independencia al sistema y no tener que estar vinculado a un único SGBD, el acceso a éste se hace a través de las librerías Pear. La Figura 9.5 muestra la arquitectura del sistema de ejecución de sentencias SELECT.

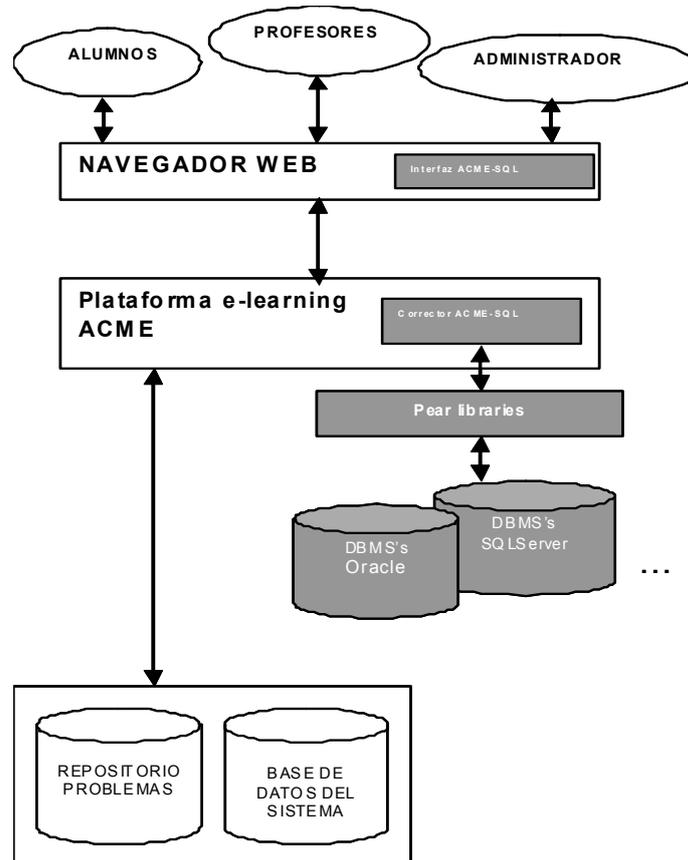


Figura 9.5 Arquitectura del sistema ACME-SQL.

Comentar que la primera vez que se ejecuta una consulta el sistema todavía no dispone de la base de datos creada. En este caso, de forma totalmente automática, se crean cada una de las tablas a partir de las especificaciones del problema.

Este proceso se repetirá sobre varias bases de datos con la misma estructura pero con distintos valores para evitar al máximo posibles aciertos por casualidad.

## Comprobación de los resultados

Una vez se dispone de los resultados obtenidos con la ejecución de las dos SELECT sólo nos queda comprobar si son iguales. En esta comprobación nos podemos encontrar con la casuística siguiente:

- Los dos resultados obtenidos tengan distinto número de filas y/o columnas. En este caso la solución es incorrecta y se da el mensaje de error correspondiente.

- Los dos resultados obtenidos tengan el mismo número de filas y columnas. En este caso iremos comprobando si los valores de cada una de los campos son coincidentes. Si efectivamente es así la solución será correcta.
- Los dos resultados obtenidos tengan el mismo número de filas y columnas pero los valores sean distintos. En este caso se hará la comprobación de que los valores no estén en distinto orden (los campos estén en distinto orden). En este caso se comprueba para cada tupla si los valores coinciden con los esperados a pesar del orden distinto. En caso de coincidir se dará el mensaje de correcto, al igual que en el apartado anterior y en caso de no coincidir, el mensaje de incorrecto.

En caso de que la respuesta sea incorrecta el alumno visualizará en la zona 3 de la Figura 9.3 el resultado obtenido con su consulta y el obtenido con la SELECT correcta del profesor, obtenidos según los valores del test de pruebas en el que se han detectado las diferencias. Para evitar posibles soluciones correctas “por casualidad” se pasan los diferentes tests de pruebas especificados en el problema base. Si con la ejecución de todos los test de pruebas se obtienen soluciones correctas consideramos la solución correcta y así lo notificamos al alumno. En la misma zona anterior el alumno visualizará los dos resultados y además el sistema le mostrará la sentencia del profesor, como solución óptima, para que el alumno la pueda conocer.

### 9.5.6 Puntuación automática de sentencias SQL

Al igual que en el caso de las consultas en álgebra relacional, creemos que las sentencias SQL se deben realizar de forma correcta y por lo tanto no dan lugar a puntuaciones intermedias. Para la evaluación de las sentencias SQL sólo consideramos si el alumno las ha resuelto de forma correcta o no. Para la realización de pruebas de validación/exámenes nos interesa comprobar si el alumno es capaz de llegar a la solución correcta. Ahora bien es frecuente que al escribir una consulta el alumno cometa pequeños fallos y debe poder enmendarlos. Para solventar este tipo de errores, en este tipo de actividades, facilitaremos al alumno un número restringido de intentos con los que llegar a la solución correcta. En ningún caso este número de intentos tiene que permitir al alumno llegar a la solución correcta a base de “prueba y error”.

ACME ya nos permite establecer el número de intentos a un determinado tipo de actividades, con lo cual este sistema de evaluación correcto/incorrecto ya queda garantizado por la plataforma.

## 9.6 EVALUACIÓN DEL MÓDULO

En este apartado se comenta la preparación de ejercicios de SQL y a continuación se detallan las actividades realizadas y los resultados obtenidos.

### 9.6.1 Preparación de ejercicios SQL

La implementación de este módulo finalizó en el 2005 y fue probado en un curso piloto durante este mismo año. Para ello se realizaron 3 problemas base. Los dos primeros problemas están desarrollados sobre la misma base de datos. El primer problema base está formado por 30 ejercicios considerados fáciles. Entre estos problemas nos encontramos con ejercicios para la creación de tablas y también para la inserción, modificación y borrado de registros. También dispone de consultas sencillas parecidas a las desarrolladas durante las sesiones presenciales. El segundo problema estaba formado por 30 consultas ya mucho más complejas. El tercer problema es sobre una base de datos distinta y estaba formado por 25 consultas, entre las que había de sencillas y también de más complejas.

Cada uno de estos problemas base sigue la estructura definida en 9.5.3. Cada problema está formado por su enunciado y a continuación las consultas que se han escrito de forma clara y concisa, evitando todo tipo de ambigüedades. En la mayoría de los casos se ha definido una única solución correcta. Los datos de la BD que se utilizan para la ejecución de las consultas son considerables y consistentes. Finalmente

comentar que sólo diseñamos un test de pruebas para la ejecución de las consultas, conscientes de que debemos especificar más tests de pruebas para evitar posibles resultados correctos por casualidad

Una vez escrito cada problema el sistema ha verificado que su sintaxis fuese la establecida para este tipo de ejercicios. Cuando la sintaxis es la correcta se ha testeado el problema exhaustivamente. Como se puede apreciar el tiempo para la creación y testeo de estos problemas es considerable, siendo este uno de los problemas de los sistemas CBA, pero hay que mencionar que una vez en funcionamiento disponemos de un problema para ser utilizado siempre que queramos.

## 9.6.2 Actividades realizadas y resultados obtenidos

Con este corrector se ha realizado una única experiencia. Durante el curso 2005/06 se pidió de forma voluntaria a los alumnos de la asignatura “Introducción a los ficheros y a las bases de datos” si querían participar en una prueba piloto para evaluar este corrector. De los 61 alumnos matriculados, 31 estuvieron dispuestos a participar. Se formaron dos grupos que compartían las mismas aulas de laboratorio donde se enseñaba SQL. En ellas el profesor explicaba los conceptos y sentencias de SQL y el alumno iba siguiendo el desarrollo de la materia. Una vez finalizada la clase el profesor facilitaba una lista de sentencias SQL (mayoritariamente SELECT) para que los alumnos practicasen en casa. Los alumnos que quisieron participar en la prueba piloto tenían que resolver las mismas sentencias pero a través de ACME. Al final de las sesiones de laboratorio todos los alumnos debían hacer un examen de SQL. En el examen se les propusieron 10 preguntas. La valoración se fijó en A (perfecto o un error), B (dos o tres errores), C (cuatro o cinco errores) y D (más de cinco errores). Los resultados del grupo que utilizó ACME (grupo B) fueron ligeramente superiores a los que no lo utilizaron (grupo A). En la Tabla 9.1 pueden observarse los resultados obtenidos

CALIFICACIÓN	GRUPO A	GRUPO B
A	6	8
B	13	16
C	9	4
D	3	3

Tabla 9.1 Resultados obtenidos con ACME-SQL.

Evidentemente, consideramos que estos resultados no son significativos, pero sirvieron para comprobar el perfecto funcionamiento del corrector, la facilidad de uso y corroborar las ventajas que aporta al profesor.

Desde entonces no se ha realizado ninguna otra prueba, simplemente por razones de organización y disponibilidad del profesorado en esta asignatura.

## 9.7 APORTACIONES Y CONCLUSIONES

Se ha propuesto, definido y desarrollado el entorno ACME-SQL que es un entorno CBA para la corrección/evaluación de cualquier comando SQL. Se ha integrado en la plataforma ACME y se ha creado una colección de problemas. Las principales ventajas que presenta son:

- Permite la corrección de cualquier sentencia SQL. Muchas de las herramientas existentes sólo permiten la validación de consultas a través de una SELECT.
- Permite llevar a cabo la evaluación formativa en el aprendizaje de SQL. Actúa como un sistema CBA.

El trabajo desarrollado en este corrector ha dado lugar al artículo “**A web based tool for teaching and learning SQL**” presentado en la International Conference on Information Technology based higher Education and Training, ITHET 2006, referenciado en [SPB+06c].



## CAPÍTULO 10. RESULTADOS GENERALES

En los capítulos precedentes hemos analizado de forma independiente los resultados obtenidos en cada uno de los módulos desarrollados referentes a la enseñanza y evaluación de diseño conceptual (diagramas ER y diagramas de clase), del diseño lógico (esquemas de bases de datos relacionales y normalización) y de consultas (álgebra relacional y SQL). Estos módulos conforman el entorno ACME-DB, un sistema CBA integrado a la plataforma ACME. En este capítulo queremos analizar el entorno ACME-DB como un todo. Para ello analizaremos los resultados obtenidos desde una perspectiva más general y amplia. En la sección 10.1 se describe cómo se ha utilizado la plataforma para mejorar la evaluación formativa durante los últimos cursos en materia de diseño de bases de datos. A continuación exponemos cómo se han evaluado los resultados obtenidos, tanto desde el punto de vista de los resultados académicos obtenidos, como del grado de satisfacción de los alumnos. Así pues, en la sección 10.2, con los datos obtenidos de ACME-DB, se realiza un análisis de la influencia de la utilización de ACME-DB y la mejora de resultados académicos. En la sección 10.3 se muestra un resumen de las encuestas que los alumnos han contestado de su experiencia con el uso de ACME-DB.

### 10.1 USO DE ACME\_DB EN LA ASIGNATURA DE BASES DE DATOS

Los distintos módulos que forman ACME\_DB se han utilizado principalmente como complemento a las clases presenciales (Blended learning) durante los últimos cursos en la asignatura de bases de datos de la carrera de Ingeniería Técnica en Informática de Gestión (ITIG). En el plan de estudios actual la materia de bases de datos está repartida en dos asignaturas: “Introducción a los ficheros y bases de datos (IFBD)” y “Bases de Datos (BD)”.

“Introducción a los ficheros y a las bases de datos” es una asignatura de 4,5 créditos que se imparte en el segundo semestre de primer curso y en la que no se profundiza en la parte de bases de datos. Consiste en dos partes perfectamente diferenciadas. En la primera se realiza el aprendizaje de los conceptos básicos de ficheros y el trabajo con distintos tipos de ellos. En la segunda parte se realiza una introducción a las bases de datos, explicando los conceptos más básicos del modelo ER y del modelo relacional, trabajando en las clases de prácticas con el lenguaje SQL. Se utilizan ejemplos sencillos a los cuales se les debe realizar el diagrama ER correspondiente y su mapeo al esquema de BD. Los alumnos utilizan el ACME-ER y el ACME-EBD con supuestos muy sencillos y de forma opcional. Tal como se ha comentado en el capítulo anterior no se utiliza ACME-SQL por razones de organización y de disponibilidad del profesorado de prácticas de esta asignatura. La Tabla 10.1 muestra de forma resumida la utilización actual de ACME en esta asignatura.

USO ACTUAL DE ACME EN IFBD					
TEMA		ENTORNO ACME	EJERCICIOS	CARÁCTER	VALORACIÓN
Ficheros	Dispositivos físicos	ACME-MAT(*)	4	optativo	0,25
	Índices sobre ficheros	ACME-MAT(*)	4	optativo	0,25
Introducción BD	Diseño conceptual: Diagramas ER	ACME-ER	6	optativo	0,25
	Diseño lógico: Modlo relacional	ACME-EBD	6	optativo	0,25
	SQL	no utilizado			
(*) Corrector de expresiones matemáticas de ACME, no desarrollado en esta tesis					

Tabla 10.1 Tabla resumen utilización actual de ACME-DB en la asignatura de IFBD.

Una vez el alumno ha cursado esta asignatura introductoria, en el siguiente semestre cursará la asignatura de “Bases de Datos” de 9 créditos con el objetivo de adquirir las competencias en la materia. Para ello

esta asignatura cuenta con tres partes. La primera parte consiste en el diseño de bases de datos, dando especial importancia al diseño conceptual y lógico, impartiendo en profundidad los modelos ER y relacional. Es en esta primera parte donde se ha utilizado ACME-DB. La segunda parte es más descriptiva y hace referencia a la estructura y funcionamiento de los SGBD. En esta parte los correctores habituales de preguntas tipo test, elección de múltiples respuestas e incluso el corrector específico de ejercicios matemáticos ACME se pueden utilizar para la realización, corrección y evaluación de los ejercicios correspondientes. La tercera parte hace referencia a las prácticas en las que se trabaja con distintos SGBD utilizando diferentes lenguajes de programación. La Tabla 10.2 muestra de forma resumida la utilización actual de ACME en esta asignatura.

USO ACTUAL DE ACME EN BD						
TEMA		ENTORNO ACME	EJERCICIOS	CARÁCTER	VALORACIÓN	Observaciones
Diseño de BD	Diseño conceptual: Diagramas ER	ACME-ER	entre 6 y 8	Obligatorio	0,5	Se realiza una prueba de validación (2 puntos)
	Diseño conceptual: Diagramas de clase	ACME-DC	4	Optativo	0,25	
	Diseño lógico: Modelo relacional	ACME-EBD	entre 6 y 8	Obligatorio	0,5	
	Normalización de BD	ACME-NOR	entre 6 y 8	Obligatorio	0,5	
	Consultas en Álgebra Relacional	ACME-AR	7	Obligatorio	0,5	
SGBD	Estructuras de acceso a datos	ACME-MAT(*)	3 con múltiples preguntas	Obligatorio	0,5	Se evalúa con un examen final (2,5 puntos)
	Otros temas	no utilizado				
Prácticas	Prácticas en diferentes entornos	no utilizado				Se evalúa conjuntamente (3 puntos)
				TOTAL	2,75	7,5

(\*) Corrector de expresiones matemáticas de ACME, no desarrollado en esta tesis

Tabla 10.2 Tabla resumen utilización actual de ACME-DB en la asignatura de BD.

La metodología de trabajo que se ha aplicado ha sido siempre la misma: una vez se ha explicado el tema en las sesiones teóricas se asignan varios problemas a través de la plataforma ACME con el objetivo de evaluar el nivel cognitivo de aplicación y de análisis de la taxonomía de Bloom. El profesor elige el nivel de dificultad de los problemas seleccionándolos del repositorio y también establece el número de problemas y la valoración de cada uno de ellos de cara a la evaluación sumativa. Mencionar que el grado de dificultad de los ejercicios va creciendo, siendo los últimos de mayor dificultad. A cada alumno se le asignarán ejercicios de dificultad fácil, media y difícil. Para evitar el plagio, el profesor selecciona todos los problemas de un determinado tipo de dificultad y el sistema, de forma aleatoria los distribuye a los cuadernos de los alumnos. De esta forma, aunque haya alguna repetición, la mayoría de ejercicios de un alumno son distintos de los de sus compañeros, aunque el nivel de dificultad es muy parecido. La realización de estos ejercicios se tiene en cuenta para la valoración final de la asignatura. Para que los alumnos trabajen de forma precisa en la resolución de los ejercicios se les insiste que tendrán penalizaciones en función del número de intentos hasta llegar a la solución correcta. Los alumnos disponen de un manual de usuario de cada uno de los módulos, aunque debido a la sencillez de uso, raramente son consultados. Se insiste a los alumnos en la importancia del trabajo continuado.

Para la evaluación formativa estas actividades son francamente positivas ya que desde la plataforma de manera muy fácil se consigue:

- **Por parte del profesor.** Disponer de herramientas para el seguimiento de la actividad del grupo (Figura 10.1) que le permiten fácilmente determinar qué alumnos están resolviendo correctamente los problemas y los que tienen dificultades. Pulsando encima del número de ejercicio el profesor visualiza las soluciones enviadas por el alumno. Fácilmente se detectan apartados en que varios alumnos se equivocan en lo mismo. En la próxima sesión de clase se comentan los errores. En todo momento el profesor tiene una visión global del aprendizaje del alumno y puede ir adaptando niveles y contenidos según el grupo. En todo momento puede añadir nuevas actividades, eliminarlas, etc.
- **Por parte del alumno.** Con la resolución de los ejercicios asignados rápidamente se dan cuenta de si saben aplicar los conocimientos adquiridos. Si un alumno tiene dudas o problemas en la resolución de un ejercicio lo consulta al profesor vía correo electrónico o a través de las tutorías

presenciales. El alumno es consciente de que el profesor controla y visualiza su trabajo. Finalmente hay que remarcar que el alumno puede en cualquier momento y desde cualquier sitio enviar las soluciones a la plataforma y de inmediato sabe si son o no correctas. Simplemente necesita de una conexión Internet y un navegador cualquiera.

The interface is divided into three main sections: a sidebar on the left, a header for the activity, and a main data table.

**Sidebar (Left):**

- QUADERN D'EXERCICIS**
  - Gestionar els exercicis
  - Cercar exercicis
  - Dossier dels alumnes
- RESPONSABLE**
  - Administrar l'assignatura
  - Administrar els usuaris
  - Gestió de grups d'alumnes
  - Avaluació de l'assignatura
  - Enviar e-mails als usuaris
  - Motlle
  - Gestió enquestes
- UTILITATS**
  - Suggeriments
  - Canviar password
  - Canviar dades
  - Ajuda

Below the sidebar is a calendar for February 2010, with the 9th highlighted in red.

**Header (Top Right):** ACTIVITAT 1 - MODEL ENTITAT/RELACIÓ

**Main Table:**

Alumne	N. Exercici	Estat	Error de Resultat	Lectures
Albaráñez, César	1	Resolt	0	4
	2	Resolt	2	13
	3	Resolt	2	11
	4	Resolt	3	9
	5	No Resolt	5	16
	6	Resolt	3	7
Alonso Cozar, Adrian	1	Resolt	0	1
	2	Resolt	1	2
	3	Resolt	1	4
	4	Resolt	1	4
	5	Resolt	2	10
	6	Resolt	1	2
Alonso Muñoz, Kenneth	1	Resolt	0	1
	2	Resolt	1	2
	3	Resolt	0	1
	4	Resolt	2	4
	5	Resolt	2	4
	6	Resolt	0	2
Aza Puig, Pau	1	Resolt	0	5
	2	Resolt	1	6
	3	Resolt	0	4
	4	Resolt	0	6
	5	Resolt	6	14
	6	Resolt	3	9
Baltrons Fabregas, Oriol	1	No Resolt	0	0
	2	No Resolt	0	0
	3	No Resolt	0	0
	4	No Resolt	0	0
	5	No Resolt	0	0

Figura 10.1 Interfaz de seguimiento del profesor.

Estas actividades se tienen en cuenta para la evaluación continua. Para ello damos un valor a cada ejercicio en función de su dificultad. Como el objetivo final es que el alumno lo resuelva correctamente y como dispone de tiempo para hacerlo, no ponemos ninguna limitación respecto al número de intentos que puede hacer ni tampoco ninguna penalización si se resuelve dentro de los tres primeros intentos. De todas formas, como que el sistema es parametrizable, el profesor puede establecer las penalizaciones que considere oportunas.

Actualmente, la forma en que obtenemos la nota final se basa en parte en la evaluación formativa (continua) que vamos realizando a lo largo del curso y en la evaluación sumativa a través de un examen final a modo de prueba de validación. Uno de los problemas existentes en los CBA es que no podemos saber quién realmente ha resuelto los ejercicios y pruebas planteados, motivo suficiente para realizar una prueba final de validación. Esta prueba consiste en la resolución de ejercicios de dificultad parecida a la planteada en los ejercicios de ACME-DB y si se obtiene una nota superior a cuatro valida la prueba. A través del entorno para exámenes de la plataforma ACME descrito en la sección 2.5.4, en la nueva asignatura de “Bases de Datos” de los estudios de Grado en Ingeniería Informática ya se utilizará el sistema de puntuación automático descrito y las pruebas de validación se realizarán en un aula informática. Se convocará a los alumnos en estas aulas equipadas con ordenadores personales con conexión a Internet y deberán solucionar la prueba de validación. Al final de la sesión enviarán las respuestas entradas a corregir y automáticamente ACME-DB les facilitará la nota obtenida que en caso de validar la prueba puede ponderar la nota obtenida en la evaluación continua. De esta forma nos ahorramos la corrección de las pruebas presenciales y habremos logrado un sistema de evaluación totalmente automatizado. Comentar que siempre la nota podrá ser revisada y modificada por el profesor.

## 10.2 RESULTADOS OBTENIDOS EN EL DISEÑO DE BD

En los actuales planes de estudio, la asignatura de Bases de Datos se imparte en el primer semestre de segundo curso a los alumnos de ITIG. En esta asignatura es donde, en mayor o menor grado, se han utilizado todos los módulos que forman el entorno ACME-DB desarrollado en esta tesis. Destacaríamos el uso referente a la parte de diseño de bases de datos, tanto del diseño conceptual como del lógico, que cubre una parte importante de esta asignatura. Los módulos ACME-ER para la realización de ERD, el módulo ACME-EBD para la corrección de esquemas de bases de datos relacionales y el módulo ACME-NOR para la normalización de estos esquemas han sido utilizados desde el curso 2004-2005. En este curso se realizaron las pruebas piloto necesarias para observar el correcto funcionamiento. Su utilización fue voluntaria por parte del alumnado y nos sirvió para evaluar los módulos de forma independiente, obteniendo resultados muy prometedores en el aprendizaje del diseño de bases de datos.

Hasta entonces la metodología usada en la enseñanza del diseño de BD consistía en unas sesiones de teoría y unas sesiones de prácticas/problemas. En las primeras el profesor introducía los principales temas teóricos con la ayuda de algunos ejemplos y problemas sencillos. A continuación se facilitaba a los alumnos una lista de ejercicios problemas para que el alumno los resolviera. Algunos de estos problemas se resolvían en clase pero la mayoría quedaba sin resolver. La mayoría de alumnos resolvían muy pocos problemas. Algunos creían que ya los sabían hacer y que no tendrían ninguna dificultad en aprobar el examen. Profesor y alumnos desconocían el nivel de conocimiento adquirido hasta el día del examen. Las deficiencias en el aprendizaje no se detectaban hasta el final y ya no había tiempo para ninguna rectificación. La evaluación final de la asignatura se obtenía a partir de las tres partes que conforman la asignatura: evaluación de la parte de diseño de BD, de la parte de estructura y funcionamiento de SGBD y de las prácticas. En la parte referente al diseño de bases de datos se realizaba una prueba donde el alumno debía realizar varios ERD en respuesta a situaciones que se le planteaban. También debía obtener el ERD y el esquema de bases de datos a un supuesto planteado. La prueba finalizaba con un ejercicio de normalización y también con una consulta en álgebra relacional. En la parte referente a SGBD se le formulaban distintas preguntas de cada tema. A partir de este examen final con estas dos partes más la nota de prácticas se obtenía la nota final.

Para el siguiente curso (2005/06) teníamos previsto realizar distintas pruebas con el objetivo de valorar la incidencia que tenía la utilización de estos módulos en el aprendizaje de diseño de bases de datos. Para ello planificamos inicialmente realizar varios grupos, unos realizarían las distintas actividades con ACME-DB y los otros con lápiz y papel. Al final decidimos que todos ellos utilizarían los módulos ACME-DB. Los motivos que nos llevaron a tomar esta decisión fueron varios:

- Grado de satisfacción de la experiencia del curso anterior.
- Robustez de los módulos desarrollados.
- Malos resultados académicos en los cursos precedentes (tasas de aprobados inferior al 60%).
- Puesta en marcha de un plan piloto para la adaptación de los estudios de ITIG al EEES.
- Insistencia de los alumnos en querer utilizar ACME.
- Convencimiento de que se iban a obtener mejores resultados.

Así pues, en el curso 2005/2006 introducimos los tres módulos para mejorar la adquisición de las competencias en diseño de base de datos, con lo que ACME-DB tendría un papel muy importante. El objetivo era explotar todas las capacidades de la plataforma para asignar, resolver y corregir de forma automática las actividades asignadas a los estudiantes. En esta nueva metodología el sistema asigna un cuaderno diferente para cada alumno que contiene ejercicios de los temas principales: diagramas entidad-relación, esquemas de base de datos relacionales y normalización. Para cada tema se asignan ejercicios de diferente complejidad. Los ejercicios más fáciles contienen entre 5 y 10 entidades y otras tantas relaciones. Los más difíciles entre 15 y 20. Estos ejercicios se utilizan para la evaluación continua que contribuye en un 50% de la nota final y el resto del 50% se obtiene de una prueba de validación realizada al final de la parte de diseño. La nota de evaluación continua se obtiene automáticamente a partir de los ejercicios resueltos. El alumno puede ir enviando soluciones hasta obtener la correcta. Aunque el sistema permite parametrizar el número de intentos, por defecto los tres primeros intentos no penalizan la nota. La ventaja del sistema es que el profesor no debe corregir los ejercicios.

En la Tabla 10.3 se recogen los resultados obtenidos de la utilización de la plataforma en los últimos cuatro años. Para cada uno de los tres temas (diagramas ER, esquemas de bases de datos relacionales y normalización) se muestra el número de estudiantes, el número de problemas asignados, el % de problemas leídos, el % de problemas a los que se han enviado soluciones y el % de problemas resueltos correctamente. En general más del 80% de los problemas son leídos y más del 70% son resueltos correctamente. También se muestra el % de problemas resueltos al primer, segundo, tercer o más intentos. Entre el 63% y 80% de los problemas son resueltos en uno o dos intentos.

TEMA	CURSO	ALUMNOS	PROBLEMAS	% PROBLEMAS LEIDOS	%SOLUCIONES ENVIADAS	%SOLUCIONES CORRECTAS	% INTENTOS			
							1	2	3	>3
DIAGRAMAS ENTIDAD/RELACIÓN	2005/06	61	6	92,5%	90,5%	81,5%	45,3%	26,4%	16,0%	12,3%
	2006/07	50	8	94,0%	90,0%	76,4%	39,8%	25,1%	15,7%	19,4%
	2007/08	46	8	96,2%	94,0%	81,0%	57,4%	23,8%	9,8%	9,0%
	2008/09	36	8	91,0%	90,5%	80,0%	42,3%	36,9%	10,1%	10,7%
ESQUEMAS BASES DE DATOS RELACIONALES	2005/06	61	6	92,5%	84,6%	70,5%	33,0%	30,2%	16,2%	20,7%
	2006/07	50	8	92,8%	84,4%	74,4%	46,8%	27,4%	12,9%	12,9%
	2007/08	46	8	96,0%	90,9%	75,4%	36,5%	33,2%	13,9%	16,3%
	2008/09	36	8	86,2%	82,4%	78,1%	43,3%	28,7%	13,4%	14,6%
NORMALIZACIÓN	2005/06	61	6	89,6%	86,9%	82,0%	48,0%	26,8%	11,4%	13,8%
	2006/07	50	8	91,5%	88,5%	82,5%	50,9%	23,6%	17,6%	7,9%
	2007/08	46	8	84,2%	77,8%	72,3%	57,4%	23,8%	9,8%	9,0%
	2008/09	36	8	81,4%	78,6%	75,0%	51,4%	28,6%	11,4%	8,6%

Tabla 10.3 Información obtenida del uso de ACME-DB.

Somos conscientes que para un buen estudio estadístico se deberían haber definido grupos de control creados aleatoriamente, es decir, tendríamos unos grupos utilizando la plataforma ACME-DB y otros la metodología clásica con lápiz y papel. Por los motivos comentados anteriormente, optamos por usar directamente los módulos ACME-DB desarrollados. Por este motivo, el estudio estadístico se ha realizado comparando los resultados obtenidos con los de los cursos precedentes en los que no se había utilizado. Así pues, para evaluar la influencia del uso de ACME-DB en los resultados académicos, comparamos los resultados obtenidos en la prueba de diseño de bases de datos antes de la aplicación de ACME-BD con los resultados obtenidos en estos últimos cuatro años en los que se ha utilizado el entorno.

Estos datos se recogen en la Tabla 10.4. Para cada curso mostramos el número de estudiantes y también sus calificaciones en diseño de bases de datos. La columna NP representa a los estudiantes que no se han presentado a la prueba. En las dos últimas columnas se muestra el % de los estudiantes que han superado la prueba. Se puede observar que los datos obtenidos al utilizar ACME-BD (periodo 2005/06 a 2008/09) son mejores que los anteriores en los que no se utilizaba. Si analizamos el número de estudiantes que han aprobado el curso, vemos que hay un incremento considerable en comparación con los resultados obtenidos cuando no se utiliza la plataforma. Comentar que aunque el número de estudiantes de los últimos cursos se ha reducido considerablemente, la ratio estudiantes/profesor se ha mantenido. Consideramos que estos mejores resultados se deben al hecho de que el estudiante realiza más problemas, sabiendo inmediatamente si los ha resuelto de forma correcta.

CURSO	ALUMNOS	EXCELENTE	NOTABLE	APROBADO	SUSPENSO	NP	NO SUPERAN	SUPERAN
2002-03	87	2,3%	14,9%	26,4%	39,1%	17,2%	56,3%	43,7%
2003-04	111	1,8%	9,0%	49,5%	25,2%	14,4%	39,6%	60,4%
2004/05	104	2,9%	13,5%	43,3%	16,3%	24,0%	40,4%	59,6%
2005/06	61	3,3%	37,7%	23,0%	16,4%	19,7%	36,1%	63,9%
2006/07	50	4,0%	24,0%	36,0%	20,0%	16,0%	36,0%	64,0%
2007/08	46	2,2%	21,7%	47,8%	15,2%	13,0%	28,3%	71,7%
2008/09	36	2,8%	52,8%	22,2%	5,6%	16,7%	22,2%	77,8%

Tabla 10.4 Resultados académicos.

En la Tabla 10.5 se evalúa la relación entre el número de alumnos que han resuelto correctamente problemas ACME-BD y el número de alumnos que han aprobado. Con la finalidad de analizar los efectos de ACME-BD en el rendimiento académico hemos agrupado los alumnos en 4 grupos en función del número de problemas resueltos correctamente. Cada grupo corresponde a una fila de la tabla. El grupo 1 representa los estudiantes que han resuelto menos del 25%, el grupo 2 los que han resuelto entre 25% y 50%, el grupo 3 entre 50% y el 75% y el grupo 4 más del 75%. Para cada grupo hemos calculado la nota media y la correspondiente desviación estándar de los 4 cursos juntos y de cada curso por separado, como

se muestra en la Tabla 10.5 Para los cálculos estadísticos las notas se han establecido entre 0 y 1 equivalentes a 0 y 10. En la Figura 10.2 hemos representado la distribución de notas en un box-plot. Se puede observar que la nota de los alumnos (marks) del grupo 1 es mayoritariamente 0 y tiene una baja variabilidad. En el caso del grupo 2, los estudiantes se dividen casi por igual entre los que aprueban (nota > 0,5) y los que suspenden (nota < 0,5), de hecho, la mediana es de 0,45.

Los grupos 3 y 4, presentan el mínimo a 0,45, aunque la mediana y los cuartiles del grupo 4 (0,69, 0,6 y 0,76, respectivamente) son superiores a los del grupo 3 (0,57, 0,45 y 0,65) lo que muestra un efecto positivo del uso de ACME-BD. Los alumnos que resuelven más ejercicios ACME obtienen mejores resultados académicos.

Grupo	Total		2006		2007		2008		2009	
	Media	Desv. Estan.								
1	0.059	0.13	0.14	0.20	0	0	0.15	0.089	0	0
2	0.46	0.12	0.46	0.12	0.35	0.18	0.47	0.089	0.55	0.11
3	0.56	0.11	0.58	0.13	0.56	0.10	0.54	0.073	0.60	0.085
4	0.67	0.13	0.71	0.12	0.64	0.13	0.64	0.12	0.72	0.13
<b>Total</b>	<b>0.57</b>	<b>0.12</b>	<b>0.57</b>	<b>0.13</b>	<b>0.55</b>	<b>0.12</b>	<b>0.56</b>	<b>0.11</b>	<b>0.61</b>	<b>0.11</b>

Tabla 10.5 Medias y desviaciones estándar de los cuatro cursos juntos y por separado.

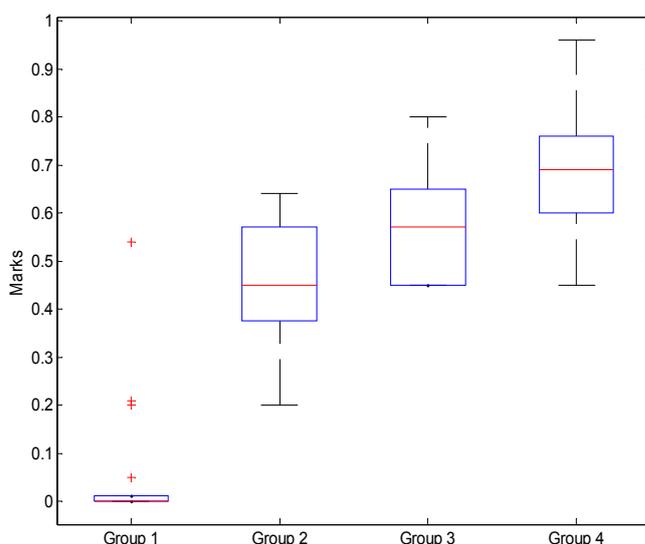


Figura 10.2 Box-plot por el total de los alumnos de los cuatro cursos.

Para corroborar este hecho, hemos llevamos a cabo un test de comparación de medias (ANOVA) para constatar que las diferencias observadas entre los grupos definidos según el % de ejercicios resueltos son significativas. En primer lugar para verificar si se puede utilizar el test ANOVA, se ha realizado un test de Levene para verificar la igualdad de la varianza en las distintas muestras. Para ello se ha utilizado la función “vartestn” de Matlab. Los resultados obtenidos se muestran en la Tabla 10.6. Se observa que todos los valores de p son mayores que un nivel de significación de 0,05, por lo tanto aceptamos que las varianzas son iguales y podemos aplicar el test ANOVA para comparar las medias.

	Total	2006	2007	2008	2009
<b>Levene's statistic</b>	0.73	1.12	1.91	0.94	0.71
<b>p-value</b>	0.53	0.35	0.14	0.43	0.55

Tabla 10.6 Test de Levene y valores de p.

Como se muestra en la Tabla 10.7, el test ANOVA nos permite afirmar que las medias de los cuatro grupos no son iguales debido a que el valor de p se encuentra por debajo de 0,05. En una comparación por pares, los valores de p para cada par son más bajos que 1.0E-3 (3,6 E-9, 0, 0, 9.0E-4, 2.0E-10, 5.3E-9).

Por lo tanto, podemos asegurar que existen diferencias significativas entre ellos, que confirman que los estudiantes que resuelven más ejercicios con ACME-BD obtienen mejores resultados académicos.

Source	SS	df	MS	F	p-value
Groups	5.886	3	1.962	130.2	0
Error	2.863	190	0.0151		
Total	8.749	193			

Tabla 10.7 Test ANOVA. Tabla de los cuatro cursos.

Comentar que hasta cierto punto eran de esperar estos resultados ya que generalmente los mejores alumnos son los que más participan en las clases y los que resuelven los ejercicios propuestos.

Estos resultados obtenidos se publicarán en el artículo “A web-based e-learning tool for database design courses” en el especial issue “Methods and Cases in Computing Education” de la revista *International Journal of Engineering Education* que hemos referenciado con [SBP+10b].

### 10.3 ENCUESTAS

Al final de cada semestre los alumnos pueden opcionalmente contestar a una encuesta para cada asignatura en la que han realizado actividades ACME. En esta encuesta se pide una valoración de ciertos aspectos de la plataforma ACME y de su uso. En el caso de la asignatura de Bases de Datos, donde se han utilizado los correctores presentados en esta tesis, en los últimos cuatro cursos hemos obtenido un total de 79 encuestas. En la Tabla 10.8 se muestran las preguntas más relevantes y el % de las respuestas facilitadas a cada ítem. Para poder visualizar más fácilmente estos resultados en la Figura 10.3 se muestran las gráficas correspondientes a cada pregunta, donde el número 1 indica “En desacuerdo total” y 5 nos indica “Totalmente de acuerdo”. Finalmente disponen de un apartado donde libremente pueden reseñar los aspectos que más les han gustado y los aspectos que se deberían mejorar.

RESUMEN ENCUESTA ACME-DB					
	En desacuerdo total	En desacuerdo	Ni de acuerdo ni en desacuerdo	De acuerdo	Totalmente de acuerdo
¿Es fácil aprender a utilizar ACME?	3,8%	7,6%	11,4%	36,7%	40,5%
¿El entorno de presentación de ACME es agradable para trabajar?	3,8%	10,1%	22,8%	35,4%	27,8%
¿La resolución de los ejercicios ACME me ha ayudado a aprender?	2,5%	2,5%	8,9%	44,3%	41,8%
¿El uso de ACME ayuda a programar mejor el estudio?	8,9%	8,9%	20,3%	38,0%	24,1%
¿La colección de ejercicios ACME es adecuada a la asignatura?	2,5%	7,6%	15,2%	43,0%	31,6%
¿Las actividades desarrolladas están suficientemente valoradas ?	16,5%	13,9%	27,8%	34,2%	7,6%
¿Los mensajes de error en las correcciones, se entienden ?	7,6%	10,1%	16,5%	41,8%	24,1%
¿El uso de ACME te ha ayudado a ver tus deficiencias?	2,5%	7,6%	17,7%	45,6%	26,6%
	casi nada	poca	la necesaria	No mucha	Mucha
¿Cuánto trabajo te han comportado las actividades ACME?	1,3%	6,3%	49,4%	24,1%	19,0%
¿Los ejercicios ACME los has desarrollado ?	Solo	En grupo	Otros		
	88,6%	11,4%	0,0%		

Tabla 10.8 Resumen encuesta ACME-DB.

Como se puede apreciar más del 77% está de acuerdo en que el entorno es fácil de utilizar. Respecto a si el entorno es agradable de trabajar sólo un 14% no están de acuerdo con esta afirmación. Hay un amplio consenso (más del 86%) en que ACME ayuda en el aprendizaje de los temas de BD. Son considerables los alumnos que opinan que ACME-DB ayuda a programar las actividades a desarrollar durante el curso mientras que más del 74% de los alumnos consideran que los ejercicios desarrollados son adecuados para la asignatura. La opinión de que las actividades ACME están suficientemente bien valoradas está muy dividida entre los alumnos, siendo muchos los que opinan que se podrían valorar más. Dos de cada tres alumnos consideran que los mensajes de feedback que se envían son adecuados y más del 72% está de acuerdo en que ACME-DB le ayuda a ver sus deficiencias de aprendizaje. En general consideran que ACME-DB les comporta un trabajo considerable pero tampoco excesivo. Finalmente el 88,6 % manifiesta que resuelve los ejercicios individualmente, mientras que el resto lo realiza en grupo.

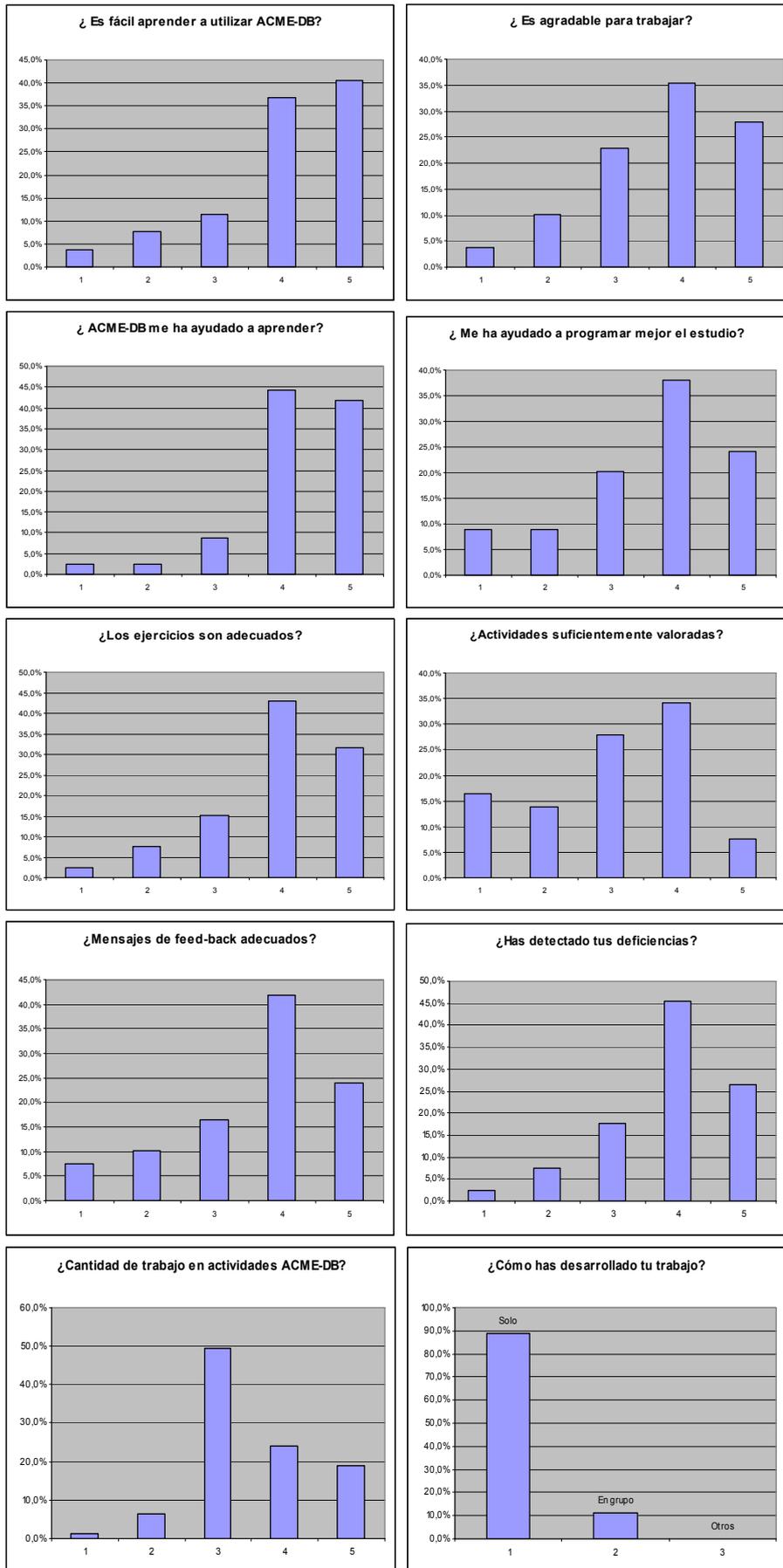


Figura 10.3 Gráficas encuestas ACME-DB.

En la Tabla 10.9 se muestran algunas de las respuestas de los alumnos a la pregunta ¿Qué es lo que más te ha gustado de ACME-DB?

<b>¿Qué es lo que más te gusta de ACME ?</b>
Te corrige al instante
Poder presentar los ejercicios desde casa
Que haya un seguimiento de la asignatura que te exige un trabajo constante
La facilidad y libertad de poderlo hacer en cualquier lugar y en cualquier momento
Sabes si lo que estás haciendo es correcto o no
Puedes trabajar desde casa y ser evaluado al momento
Nos hace aprender de los errores. Si entregas un trabajo en mano, cuando nos lo devuelven corregido ya no nos fijamos en los errores
ACME es una herramineta que nos ayuda muchísimo en hacer las cosas de una manera interactiva
Que especifica Donde se cometen los errores y esto ayuda para poder corregirlos
La manera de presentar el modelo ER y el modelo relacional
La agenda personal ya que nos avisa cuando has de entregar los ejercicios
En global todo

Tabla 10.9 Aspectos positivos de ACME-DB según encuestas.

En la Tabla 10.10 se muestran algunas de las respuestas de los alumnos a la pregunta ¿Qué aspectos crees que se deberían mejorar?

<b>¿Qué aspectos se deberían mejorar?</b>
Valorar más el trabajo realizado en ACME
Deberíamos podernos asignar problemas, aunque no puntuasen.
La redación de algunos problemas ER no es muy clara
Los mensajes de algunos errores no ayudan demasiado
El editor para la entrada de las tablas se podría mejorar
Periodos de entrega más largos
Las actividades comportan mucho trabajo
Poder acceder a las soluciones de los problemas que no hemos sabido resolver

Tabla 10.10 Aspectos a mejorar de ACME-DB según encuestas.



# CAPÍTULO 11. CONCLUSIONES FINALES Y TRABAJO FUTURO

En este capítulo se describen las conclusiones finales y el trabajo futuro. Se ha estructurado el tema en cinco secciones. En la sección 11.1 se describen las conclusiones obtenidas. En la sección 11.2 se enumeran las mejoras aportadas respecto a los entornos existentes y a continuación en la sección 11.3 se describe el trabajo futuro. En la sección 11.4 se realizan unas reflexiones finales.

## 11.1 CONCLUSIONES

En el Espacio Europeo de Educación Superior (EEES) en el que se da mucha importancia al aprendizaje del alumno y a la evaluación por competencias, se van a necesitar y utilizar cada vez más herramientas de e-learning para complementar las actividades presenciales que permitan:

- Disponer del material adecuado de la materia.
- Un control del aprendizaje del alumno.
- Una evaluación automática (tanto formativa como sumativa), ya sea para su autoevaluación, evaluación continuada, exámenes, etc. que suponga una menor dedicación del profesor a tareas muy repetitivas, como la corrección y evaluación de actividades.

Si bien existen entornos que satisfacen estas características, la mayoría están pensadas para la resolución de ejercicios a través de respuestas fijas. En el caso de la resolución de problemas complejos es necesario desarrollar entornos específicos con funcionalidades avanzadas que se ajusten a cada materia.

En esta tesis se han desarrollado un conjunto de herramientas web para dar soporte a la enseñanza y evaluación automatizada en bases de datos, integrándolas en un sistema CBA al que hemos llamado ACME-DB. Este entorno permite respuestas libres a los ejercicios planteados para poder evaluar niveles cognitivos intermedios (aplicación y análisis) de la taxonomía de Bloom. Para ello hemos desarrollado módulos capaces de interpretar, corregir y retornar el feedback apropiado a los ejercicios más habituales en materia de bases de datos, facilitando la evaluación formativa tanto a alumnos como profesores. Por otra parte algunos de estos módulos disponen de funcionalidades para la puntuación automática que permiten obtener una nota para cada ejercicio, facilitando de esta forma la evaluación final del alumno. Los módulos desarrollados en ACME-DB han sido:

- Módulo de diagramas Entidad/Relación (ACME-ER).
- Módulo de diagrama de clases (ACME-DC).
- Módulo de esquemas de bases de datos relacionales (ACME-EBD).
- Módulo para la normalización (ACME-NOR).
- Módulo de consultas en álgebra relacional (ACME-AR).
- Módulo de consultas SQL (ACME-SQL).

Todos estos módulos se han utilizado en entornos reales. Los módulos ACME-ER, ACME-EBD y ACME-NOR se están utilizando desde el curso 2005/06 y en esta tesis se han analizado los resultados obtenidos. Los módulos ACME-DC y ACME-AR se han utilizado por primera vez en el curso 2009/10. El módulo ACME-SQL también se ha utilizado en un curso piloto (curso 2005/06).

La utilización de los distintos módulos se ha analizado desde un punto de vista académico, evaluando los resultados obtenidos y el grado de satisfacción de los alumnos. Los resultados obtenidos muestran que los alumnos que realizan los problemas asignados a través de la plataforma ACME-DB obtienen mejores resultados académicos. Las encuestas realizadas también nos manifiestan una buena aceptación del entorno desarrollado.

Desde nuestro punto de vista, nuestra principal aportación en esta tesis es haber integrado en un único entorno (ACME-DB) un conjunto de herramientas para el aprendizaje/evaluación automatizados de los principales temas en materia de BD aportando mejoras a las existentes.

## 11.2 APORTACIONES RESPECTO A LOS ENTORNOS EXISTENTES

Uno de los objetivos propuestos en esta tesis era la de mejorar los entornos existentes en temas de corrección automática en materia de bases de datos. En la sección 2.4 se establecieron un conjunto de parámetros considerados importantes para estos entornos con los que se compararon las distintas herramientas descritas en el estado del arte de cada uno de los correctores.

A continuación para cada uno de estos parámetros se mencionan las mejoras de ACME-DB respecto a los entornos existentes.

- **Entorno integrado para la docencia de bases de datos.** Valoramos muy positivamente que los distintos correctores se integren en un único entorno. Los módulos desarrollados se han integrado en la plataforma ACME-DB de forma que el alumno sólo necesita conocer un único entorno. Los módulos desarrollados permiten el aprendizaje y la evaluación de:
  - El diseño conceptual de bases de datos. Para ello se ha creado el entorno de diagramas entidad/relación (ACME-ER) y el de diagramas de clase (ACME-DC).
  - El diseño lógico. Se ha desarrollado el entorno de esquemas de bases de datos (ACME-EBD).
  - La normalización de BD. Se ha reutilizado el módulo anterior para la programación de ejercicios de normalización (ACME-NOR).
  - Consultas de álgebra relacional. Se ha desarrollado un módulo corrector de expresiones de álgebra relacional (ACME-AR).
  - SQL. Se ha creado un módulo corrector de sentencias SQL (ACME-SQL).
- **Web Based tool.** Para facilitar el trabajo al alumno es importante que sólo requiera de una conexión a Internet y un navegador. En nuestro caso todos los módulos desarrollados funcionan correctamente en los navegadores habituales.
- **Corrección y feedback automáticos.** De caras a la evaluación formativa es muy importante que el sistema corrija automáticamente y ofrezca el feedback oportuno al alumno. En nuestro caso, cada uno de los módulos desarrollados corrige de forma automática el problema, ofreciendo el feedback apropiado.
- **Puntuación automática del ejercicio.** Pensando en la evaluación sumativa del alumno es deseable que la plataforma ofrezca una puntuación de cada ejercicio lo más parecida posible a la que obtendría un profesor en una corrección no automatizada. Con la excepción del módulo ACME-AR y el módulo ACME-SQL los demás módulos desarrollados ya incorporan funcionalidades de puntuación automática.
- **Evaluación automática del alumno.** El mismo entorno debe facilitar la evaluación final de todo el trabajo realizado por el alumno y facilitar también la tarea al profesor. Las propias funcionalidades de ACME recogen las puntuaciones de los distintos ejercicios y proporcionan la evaluación final del alumno. Estas funcionalidades permiten también exportar los resultados a una hoja de cálculo Excel.
- **Trabajo colaborativo.** El entorno debe ofrecer funcionalidades que permitan el trabajo en grupo y de forma colaborativa. Las nuevas metodologías docentes muestran los buenos resultados que se obtienen con este tipo de trabajo. Las propias funcionalidades de ACME permiten el trabajo en grupos.
- **Necesidad de software adicional.** Es importante que para el funcionamiento de la herramienta no se necesite software adicional. Ninguno de los módulos desarrollados necesita software adicional, simplemente el navegador. Tampoco es necesario disponer de ningún SGBD.

La Tabla 11.1 resume las características que tienen los entornos desarrollados en este trabajo. Como puede verse los módulos desarrollados satisfacen los requisitos que consideramos más importantes para este tipo de sistemas. En las Tablas 11.2, 11.3, 11.4, 11.5, 11.6 y 11.7 se muestran las tablas de características de las herramientas examinadas en el apartado del estado del arte a las que se ha añadido una última columna con las características de los distintos entornos ACME desarrollados en este trabajo. Como se puede apreciar las herramientas desarrolladas aportan mejoras respecto a las existentes.

Estas aportaciones juntamente con la experiencia en el uso del entorno ACME-DB han dado lugar a las publicaciones [PBS+05], [PBS+06], [SBP+06], [SPB+06a], [SPB+06b], [SPB+06c], [SBP+07], [SBP+09a], [SBP+09b], [SBP+10a], [SBP+10b], [SBP+10c].

CARACTERÍSTICAS / ENTORNOS ACME-DB	ACME-ER	ACME-DC	ACME-EBD	ACME-NOR	ACME-AR	ACME-SQL
Entorno integrado para la docencia de bases de datos	si	si	si	si	si	si
Web-based tool	si	si	si	si	si	si
Editor propio	si	si	si	si	si	si
Corrección automática	si	si	si	si	si	si
Puntuación automática del ejercicio	si	si	si	si	no*	no*
Evaluación automática del alumno	si	si	si	si	si	si
Trabajo colaborativo	si	si	si	si	si	si
Necesidad de software adicional	no	no	no	no	no	no
Proporciona feed-back automático	si	si	si	si	si	si
<b>CARACTERÍSTICAS ESPECÍFICAS</b>						
Notación matemática	-	-	-	-	si	-
Cualquier comando Sql	-	-	-	-	-	si
Visualización resultado SELECT's	-	-	-	-	-	si
Visualización del esquema de la base de datos	-	-	-	-	-	si
* No puntúa, sólo correcto/incorrecto						

Tabla 11.1 Características del entorno ACME-DB.

MODELO ENTIDAD RELACIÓN	ER M- V L E	CO L E R	KE R M I T	DA T S Y S	DE A P - E X E R	AB C	B a t m a z	# E E R	AC M E - E R
<b>CARACTERÍSTICAS / ENTORNOS</b>									
Entorno integrado para la docencia de bases de datos	no	no	si	no	no	no	no	parcial	<b>SI</b>
Web-based tool	no	si	si	no	no	no	no	no	<b>SI</b>
Editor de diagramas propio	si text	si	no	si	si	si	si	si	<b>SI</b>
Corrección automática	no	no	si	si	si	semi	semi	no	<b>SI</b>
Puntuación automática del ejercicio	no	no	no	si	si	semi	semi	no	<b>SI</b>
Evaluación automática del alumno	no	no	no	si	si	semi	si	no	<b>SI</b>
Trabajo colaborativo	no	si	si	no	no	no	no	no	<b>SI</b>
Necesidad de software adicional	no	no	si	no	no	no	no	no	<b>NO</b>
Proporciona feed-back automático	si	si	si	si	si	no	no	no	<b>SI</b>

Tabla 11.2 Características de las herramientas de diagramas ER.

<b>Diagrama de clases UML</b>							
	<b>H O G A R T H</b>	<b>C O L L E C T</b>	<b>C I M E L</b>	<b>A S S E T</b>	<b>A r g o U M L</b>	<b>U C D A</b>	<b>A C M E - D C</b>
<b>CARACTERÍSTICAS / ENTORNOS</b>							
Entorno integrado para la docencia de bases de datos	no	no	no	no	no	no	<b>SI</b>
Web-based tool	no	si	no	no	no	no	<b>SI</b>
Editor de diagramas propio	no	si	no	si	no	no	<b>SI</b>
Corrección automática	semi	si	si	no	parcial	si	<b>SI</b>
Puntuación automática del ejercicio	no	no	no	no	no	no	<b>SI</b>
Evaluación automática del alumno	no	no	no	no	no	no	<b>SI</b>
Trabajo colaborativo	no	si	si	si	no	no	<b>SI</b>
Necesidad de software adicional	si	no	si	no	si	si	<b>NO</b>
Proporciona feed-back automático	si	si	si	no	no	si	<b>SI</b>

Tabla 11.3 Características de las herramientas de diagramas de clase.

<b>MODELO RELACIONAL</b>		
	<b>E R M - T U T O R</b>	<b>A C M E - E B D</b>
<b>CARACTERÍSTICAS / ENTORNOS</b>		
Entorno integrado para la docencia de bases de datos	si parcial	<b>SI</b>
Web-based tool	si	<b>SI</b>
Editor de diagramas propio	si	<b>SI</b>
Corrección automática	si	<b>SI</b>
Puntuación automática del ejercicio	no	<b>SI</b>
Evaluación automática del alumno	no	<b>SI</b>
Trabajo colaborativo	no	<b>SI</b>
Necesidad de software adicional	no	<b>NO</b>
Proporciona feed-back automático	si	<b>SI</b>

Tabla 11.4 Características de las herramientas de esquemas de BD.



CARACTERÍSTICAS / ENTORNOS	W i n R D B I	e S Q L	S Q L - T u t o r	W e b S Q L	S Q L - t r a i n e r	A c h a r y a	A s s e s q l	S Q L a t o r	K P 0 5	S Q L i f y	L E A R N _ S Q L	A C M E - S Q L
Entorno integrado para la docencia de bases de datos	si parcial	no	si parcial	no	no	no	no	no	no	no	no	<b>SI</b>
Web-based tool	no	no	si	si	si	si	si	si	si	si	si	<b>SI</b>
Editor propio	si	si	si	si	si	si	si	si	si	si	si	<b>SI</b>
Corrección automática	no	no	si	no	si	si	si	si	si	semi	si	<b>SI</b>
Puntuación automática del ejercicio	no	no	no	no	no	no	no	no	no	semi	no	<b>NO*</b>
Evaluación automática del alumno	no	no	no	no	si	no	si	si	no	si	si	<b>SI</b>
Trabajo colaborativo	no	no	si	no	no	no	no	no	no	no	no	<b>SI</b>
Necesidad de software adicional	no	no	no	si	si	no	no	no	no	no	si	<b>NO</b>
Proporciona feed-back automático	no	no	si	no	si	si	si	no	si	si	si	<b>SI</b>
Cualquier comando Sql	no	no	no	no	si	si	no	si	no	no	si	<b>SI</b>
Visualización resultado SELECT's	si	si	no	si	no	no	si	si	si	si	no	<b>SI PARCIAL</b>
Visualización del esquema de la base de datos	si	no	si	no	si	si	no	si	no	si	si	<b>SI</b>
* No puntúa, sólo correcto/incorrecto												

Tabla 11.7 Características de las herramientas de SQL.

## 11.3 TRABAJO FUTURO

Nuestro trabajo futuro se centra en varias líneas de investigación y de uso de la plataforma:

### Selección adaptativa de los problemas

En los sistemas tradicionales de e-learning el profesor asigna un conjunto de problemas que el alumno debe resolver. Generalmente estos ejercicios están catalogados por distintos parámetros, entre ellos el grado de dificultad. En nuestro sistema el nivel de dificultad varía de 1 a 5 (muy fácil, fácil, medio, difícil, muy difícil) y en función de la asignatura (un mismo problema puede estar catalogado como difícil en un curso de introducción y fácil en otro curso). El profesor selecciona del repositorio los problemas de diferentes niveles de dificultad que considera más idóneos. El sistema los va asignando aleatoriamente a los cuadernos de los alumnos de forma que cada alumno dispone de un cuaderno distinto al de sus compañeros y de dificultad creciente. Todos los cuadernos tienen los mismos problemas de cada nivel de dificultad. Debido a la heterogeneidad de los grupos, a menudo los alumnos no saben resolver algunos de los problemas asignados y ya no siguen intentando la resolución del resto de ejercicios.

Para solventar esta problemática, la metodología que se propone consiste en adaptar los ejercicios a resolver al nivel de cada alumno. Para ello el sistema partirá del nivel de dificultad de cada problema. El profesor simplemente deberá escoger del repositorio los problemas de cada nivel que el sistema asignará automáticamente a cada cuaderno. La asignación de problemas se realizará de forma totalmente automática de la siguiente forma:

- 1) En primer lugar el sistema seleccionará y asignará a cada cuaderno un único ejercicio de dificultad media.
- 2) El alumno visualizará y deberá resolver el ejercicio asignado. Si el alumno resuelve el problema de forma correcta, el sistema le asignará otro de un nivel igual o superior.
- 3) En caso de error el sistema ofrecerá el feedback oportuno para ayudar al alumno. El alumno dispondrá de un número configurable de intentos para obtener la solución correcta. Si después de estos intentos no obtiene la solución correcta, el sistema automáticamente le asignará otro de nivel inferior.
- 4) Se irán repitiendo los pasos 2 y 3 hasta la fecha límite o hasta que el alumno soluciona de forma correcta los problemas del nivel superior.

El objetivo es que el sistema vaya asignando de forma automática los ejercicios a cada cuaderno en función del nivel de conocimiento del alumno.

Actualmente ya disponemos de una primera versión de selección adaptativa de problema que sigue las especificaciones comentadas.

## Puntuación de ejercicios SQL y de consultas en álgebra relacional

Hasta ahora hemos considerado que las consultas de álgebra relacional y de SQL se deben corregir considerando que son correctas o incorrectas, a diferencia de los diagramas ER (o de clases) y de los esquemas de bases de datos relacionales en los que podemos admitir pequeños errores. En estos casos asignamos una puntuación entre 0 y 10 en función de los errores cometidos. Un posible trabajo futuro sería la puntuación de los ejercicios de SQL y de las consultas en álgebra relacional de forma similar a la realizada en esta tesis en los otros módulos. En función del tipo de error cometido el sistema asignaría una puntuación que se intentaría ajustar a la corrección no automática. Dada la amplia casuística que se puede plantear, este es un tema complejo y más si se quieren ajustar las puntuaciones automáticas a las manuales calculadas por los profesores.

## Previsión de uso de ACME-DB en la asignatura de Bases de Datos

En el nuevo plan de estudios de Grado en Ingeniería Informática se impartirá en el segundo curso una única asignatura de “Bases de Datos”. Es ahí donde está previsto sacar el máximo provecho de ACME-DB, usando todos los entornos que lo conforman. Con la experiencia adquirida en estos últimos cinco cursos con los entornos ACME-ER, ACME-EBD y ACME-NOR, junto con los correctores ACME-DC, ACME-AR y ACME-SQL que ya están disponibles, se podrá realizar una asignatura donde se pretende un grado de evaluación automática muy elevado, por no decir total. La Tabla 11.8 muestra la previsión de utilización de ACME en esta nueva asignatura, en la cual además del entorno específico ACME-DB está prevista la utilización de otros correctores ACME.

PREVISIÓN DE USO DE ACME EN BD EN LOS NUEVOS ESTUDIOS DE GRADO						
TEMA	ENTORNO ACME	EJERCICIOS	CARÁCTER	VALORACIÓN	Observaciones	
Diseño de BD	Diseño conceptual: Diagramas ER	ACME-ER	8	Obligatorio	1,5	Se realizará una prueba de validación con la finalidad de validar la nota obtenida virtualmente. La puntuación se realizará automáticamente. Su nota puede ponderar la obtenida por ACME.
	Diseño conceptual: Diagramas de clase	ACME-DC	6	Obligatorio	0,5	
	Diseño lógico: Modelo relacional	ACME-EBD	8	Obligatorio	1	
	Normalización de BD	ACME-NOR	6	Obligatorio	0,5	
	Consultas en Álgebra Relacional	ACME-AR	8	Obligatorio	0,5	
SQL	Consultas en SQL	ACME-SQL	25	Obligatorio	1,5	idem anterior
SGBD	Estructuras de acceso a datos	ACME-MAT(*)	4 con múltiples preguntas	Obligatorio	0,5	idem anterior
	Otros temas	ACME-STD(**)	50	Obligatorio	2	
Prácticas	Prácticas en diferentes entornos	no se utilizará				Valoración 2 puntos
				TOTAL	8	2
(*) Corrector de expresiones matemáticas de ACME, no desarrollado en esta tesis						
(**) Corrector estandar de ACME (Preguntas de respuesta fija, elección múltiple, tipo test, relleno de espacios, etc.), no desarrollado en esta tesis						

Tabla 11.8 Tabla previsión de utilización de ACME-DB en la nueva asignatura de BD.

## **Integración de ACME-DB a Moodle**

Cada vez son más las universidades e instituciones educativas que utilizan la plataforma de e-learning Moodle. Con el objetivo de integrar todo el entorno ACME a Moodle, se desarrollará un plug-in que cumpla con los estándares de Moodle y que en una primera fase permita:

- Asignar actividades de ACME desde Moodle.
- Acceder a las actividades ACME desde un usuario Moodle.
- Exportar las notas de las actividades ACME a Moodle.
- Integrar las agendas ACME y Moodle.

Con el desarrollo de este plug-in se pretende que las universidades y centros interesados puedan utilizar fácilmente los distintos entornos y funcionalidades que ofrece ACME, así como las amplias colecciones de problemas existentes.

Comentar que ya se han implementado estas funcionalidades y que ya están disponibles para el curso 2010/11.

## **Evaluación de otros tipos de diagramas UML**

A partir de la experiencia en el desarrollo, interpretación y corrección de diferentes diagramas, otra línea de investigación consistirá en la realización de otros entornos para el aprendizaje y evaluación de diferentes tipos de diagramas. En concreto y para el aprendizaje de los distintos diagramas que se utilizan en UML, está previsto desarrollar entornos para la diagramación de casos de uso, de diagramas de secuencia, etc.

## **Utilización de tecnologías de la web semántica para la evaluación de los ejercicios**

Para las tareas de corrección, los módulos ACME-ER, ACME-DC, ACME-EBD y ACME-NOR utilizan el nombre prefijado de los atributos para la identificación de entidades/clases, relaciones/asociaciones y tablas. Uno de los trabajos futuros a realizar consiste en incorporar las tecnologías de la web semántica para la interpretación del nombre asignado a los distintos elementos de forma que no se tenga que prefijar el nombre de los atributos.

## **Realización de más experiencias**

Para una mejor evaluación de las herramientas desarrolladas y para poder determinar de forma más precisa si su utilización mejora el aprendizaje de la materia de bases de datos se van a realizar diferentes experiencias. En estas experiencias se pretende llevar a cabo un estudio riguroso a través del contraste de grupos, consistente en desdoblarse los alumnos de un curso en varios grupos, unos que utilizarán las herramientas desarrolladas y otros no. Al final de la experiencia se evalúan los resultados obtenidos en cada grupo.

## **Estandarización de la plataforma ACME**

La plataforma ACME actual no sigue ningún estándar de los muchos existentes en las tecnologías de e-learning. Uno de los trabajos futuros a realizar es su estandarización de forma que se puedan reutilizar distintos objetos de aprendizaje desarrollados según los estándares actuales. Por otro lado, los distintos ejercicios ACME no dejan de ser objetos de aprendizaje con lo cual se podrán encapsular de forma que puedan ser reutilizables en otros entornos.

## 11.4 REFLEXIONES FINALES

Finalmente, una vez finalizada la tesis y alcanzados los objetivos fijados, varias reflexiones a considerar:

¿Realmente los alumnos aprenden y asimilan mejor esta materia con la resolución de problemas a través de la plataforma ACME? La percepción que tenemos es que sí. El simple hecho que el alumno sabe que desde la plataforma se le controla el trabajo que realiza, le lleva a realizar el trabajo en las fechas marcadas, con lo que el alumno va llevando la asignatura al día. Este hecho, por si solo, ya lo consideramos muy positivo. En función del trabajo que va realizando, el alumno sabe en cada momento si va asimilando la materia y por otra parte va aprendiendo de los errores que comete. Estos son dos de los factores positivos de la evaluación formativa.

¿Los resultados académicos obtenidos son mejores? En vista de los resultados académicos que se muestran en la Tabla 10.3 podemos afirmar que sí, aunque con matices. Durante el curso 04/05 se empezó un plan piloto para los estudios de ITIG al Espacio Europeo de Educación Superior (EEES) que coincidió con la puesta en funcionamiento de los módulos ACME-ER, ACME-EBD y ACME-NOR que desde entonces se han venido usando. El cambio de la metodología usada juntamente con la utilización del entorno ACME-DB han mejorado los resultados. Por otra parte también se tiene que considerar el descenso del número de alumnos, que aunque la ratio profesor/alumno se haya mantenido prácticamente igual, permite coordinar mejor y trabajar en grupos no tan extensos. Por otro lado consideramos que la tipología de alumno que podemos calificar como “buenos y aplicados” aprueban exactamente igual, pero con notas un poco mejores debidas al cambio metodológico. Los alumnos con dificultades en el aprendizaje y en el seguimiento del curso siguen un poco mejor el desarrollo de las actividades, pero al final la mayoría suspende. De este grupo algunos consiguen aprobar, por lo que se mejora la metodología previa. Finalmente donde las mejoras son más significativas es en el grupo intermedio, en donde antes se iban descolgando muchos alumnos a medida que avanzaba el curso pero ahora con los cambios introducidos cada vez son más los alumnos de este grupo que consiguen aprobar.

¿Los alumnos se adaptan a este tipo de entornos? Muy rápidamente y de forma muy fácil. A menudo actúan de la misma forma que lo hacen con un videojuego, en el que no cejan en su empeño de pasar al siguiente nivel hasta que lo consiguen. De la misma forma cuando resuelven un problema de este tipo no cejan hasta conseguir el “Correcto”. Los mensajes ante los errores cometidos son las ayudas virtuales que el sistema ofrece a modo de tutor virtual disponible en todo momento.

¿Compensa el esfuerzo del desarrollo de los entornos y de la preparación de las colecciones de ejercicios? Consideramos que sí por varias razones. La primera y más importante es el seguimiento del trabajo y del aprendizaje del alumno, tanto desde un punto de vista individual como de grupo. Para los cambios metodológicos que conlleva el EEES consideramos de gran importancia el trabajo con este tipo de herramientas que facilitan la evaluación formativa. Otra de las razones es que la reutilización de los ejercicios del repositorio en diferentes cursos rentabiliza el esfuerzo inicial. Por otra parte el sistema permite evaluar muchas más actividades con mucha menor dedicación por parte del profesor. Es importante resaltar especialmente la drástica disminución del tiempo dedicado a la corrección de pruebas.



## BIBLIOGRAFÍA

- [AB02] Antony S. and Batra D. "CODASYS: a consulting tool for novice database designers" ACM SIGMIS Database. Vol. 33 (3), pp 54-68, 2002.
- [AB05] Attali Y. and Burstein J. "Automated Essay Scoring With E-rater® v.2.0", Report No. RR-04-45, Educational Testing Service, Princeton, NJ, USA. 2005.
- [ABC+09] Abelló A., Burgués X., Casany M.J., Martín C., Quer C., Rodríguez M.E. and Urpí T. "LEARN-SQL: Herramienta de gestión de ejercicios de SQL con autocorrección". XV Jornadas de Enseñanza Universitaria de la Informática (JENUI). pp 453-460, 2009.
- [AK01] Anderson L.W. and Krathwohl D. A Taxonomy for Learning, Teaching and Assessing: a Revision of Bloom's Taxonomy of Educational Objectives. Longman, New York, 2001.
- [All00] Allen G.N. "WebSQL: An interactive Web Tool for teaching Structured Query Language" Proc. 2000 Americas Conf. on Information System, pp. 2066-2071, 2000.
- [AP00] Atkins C. and Patrick J. "NaLer: a natural language method for interpreting entity-relationship models". Campus-wide Information System. Vol. 17(3), pp 85-93, 2000.
- [APR06] Amelung M., Piotrowski M. and Rosner D. "EduComponents: Experiences in e-assessment in Computer Science Education". ACM SIGCSE Bulletin. Vol. 38 (3), pp 88-92, 2006.
- [AQT+04] Appel P., Quintino Da Silva E., Traina C. and Machado A. "iDFQL - A query-based tool to help the teaching process of the relational algebra". World Congress on engineering and Technology Education (WCETE), 2004.
- [Arm06] Arman N. "Normalizer: A case tool to normalize relational database schemas" Information Technology Journal. Vol 5 (2), pp 329-331, 2006.
- [ARU+08] Abelló A., Rodríguez M.E., Urpí T, Burgués X. "LEARN-SQL: Automatic assessment of SQL Based on IMS QTI specification". 8th International Conference on Advanced Learning Technologies (ICALT). 2008.
- [BA94] Batra D. and Antony S.R. "Novice errors in conceptual design". European Journal of Information System. Vol. 3(1), pp 57-69, 1994.
- [Bal03] Ball L. "Online Learning and fun with databases submission". Proc. of Teaching, Learning and Assessment of Databases (TLAD). pp 1-6, 2003.
- [Bar99] Barshay O. "Online Programming Examinations using WebToTeach". Proc. of Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE). 1999.
- [BBF+93] Benford S., Burke E., Foxley E., Gutteridge N. and Zin A.M. "Experiences with the Ceilidh System". Proc. of the 1st International Conference on Computer Based Learning in Science (CBLIS). 1993.
- [BBK+02] Baghat S., Baghat L., Kavalan J. and Sasikumar S. "Acharya: An intelligent tutoring environment for learning SQL". Proc. of the Vidyakash 2002 International Conference on Online Learning. 2002.
- [BBR+03] Beynon M., Bhalerao A., Roe C. and Ward A. "A Computer-based Environment for the study of Relational query languages" Teaching, Learning and Assessment of Databases (TLAD). pp 104-108, 2003.

- [BD04] Bull J. and Danson M. "Computer-assisted Assessment (CAA)". LTSN Generic Centre: Assessment Series No. 14, 2004.
- [Ben02] Bennett R. "Inexorable and inevitable: The continuing story of technology and assessment". *Journal of Technology, Learning and Assessment*. Vol. 1 (1). 2002.
- [BH07] Batmaz F. and Hinde C.J., "A web-based Semi-automatic Assessment of Conceptual Database Diagrams". *Proc. of 6th Web-Based Education (WBE)*. 2007.
- [Black] Blackboard. [www.blackboard.com](http://www.blackboard.com).
- [Bli06] Bligh B. *Formative Computer Based Assessment in diagram based domains*. Ph.D. thesis, University of Nottingham. 2006.
- [Blo56] Bloom B.S. *Taxonomy of educational objectives, Handbook I: the cognitive domain*. David McKay Co Inc., New York. 1956.
- [Blo84] Bloom B. "The sigma 2 problem: the search for methods of group instruction as effective as one-to-one tutoring". *Educational Researcher*. Vol 13(6), pp 4-16, 1984.
- [BLS01] Burstein J., Leacock C. and Swartz R. "Automated Evaluation of Essays and Short Answers". *Proc. of 5th Annual CAA Conference*. 2001.
- [BM03] Baugh J.M. and Morris R. "A First Course in Database Management". *Proc. of Information Systems Education Conference (ISECON)*. 2003.
- [BM05] Baghaei N. and Mitrovic A. "COLLECT-UML: Supporting individual and collaborative learning of UML class diagrams in a constraint-based tutor". *Knowledge-Based Intelligent Information & Engineering Systems (KES)*. pp 458-464, 2005.
- [BM99] Baker E.L. and Mayer R.E. "Computer-based assessment of problem solving". *Computers in Human Behavior*. Vol 15, pp 269-282, 1999.
- [BMI06] Baghaei N., Mitrovic A. and Irwin W. "Problem-Solving Support in a Constraint-based Tutor for UML Class Diagrams". *Technology, Instruction, Cognition and Learning Journal (TICL)*. Vol 4(1-2), pp 113-137, 2006.
- [BO03] Bessis N. and Onley P. "Amalgamising traditional and virtual learning environment methods for delivery of databases in Higher Education". *Proc. of Teaching, Learning and Assessment of Databases (TLAD)*. pp 94-98, 2003.
- [BPS+04] Boada I., Prados F., Soler J. and Poch J. "A teaching/learning support tool for introductory programming courses" *Int'l Conf. on Information Technology Based Higher Education and Training (ITHET)*. pp 604-609, 2004.
- [BPW+05] Blank G., Parvez S., Wei F. and Moritz S. "A Web-based ITS for OO Design". *Proc. of 12th International Conference on Artificial Intelligence in Education*. 2005.
- [BQA+08] Burgués X., Quer C., Abelló A., Casany M.J., Martín C., Rodríguez M.E. and Urpí T. "Uso de LEARN-SQL en el aprendizaje cooperativo de Bases de Datos". *XIV Jornadas de Enseñanza Universitaria de la Informática (JENUI)*. 2008.
- [BR96] Brown S. and Race P. *500 Tips on assessment*. Cogan Page, London, UK, 1996.
- [BRJ99] Booch G., Rumbaugh J. and Jacobson I. *Unified Modeling Language (UML), User Guide*. Addison Wesley, 1999.
- [Bro01] Brown G. *Assessment: A Guide for Lecturers*. LTSN Generic Centre: Assessment Series, 2001.

- [BS93] Batra D. and Sein M. "Improving conceptual database design through feedback", International Journal of Man Machine Studies. Vol 40, pp 653-676, 1993.
- [BSB+08] Bogdanovic M., Stanimirovic A., Davidovic N. and Stoimenov L. "The Development and Usage of a Relational Database Design Tool for Educational Purposes". Proc. of the Informing Science & IT Education Conference (InSITE). 2008.
- [BSH09] Batmaz F., Stone S. and Hinde C.J. "Personalised feedback with semi-automatic assessment tool for conceptual database model". 10th Annual Conference of the Subject Centre for Information and Computer Sciences. pp 115-119. 2009.
- [BSP+03] Belton M., Shephard E., Phaup J., Fair K. and Kleeman J. "Questionmark's Holistic Approach: Assessments systems for the enterprise, Questionmark White Paper". 2003. Available from: <http://www.questionmark.com/>
- [BSP+04] Boada I., Soler J., Prados F. and Poch J. "Entorno virtual de ayuda a la docencia de un curso de programación básica". 3r Congrés Internacional Docència Universitària i Innovació (CIDUI). 2004.
- [Buc00] Buck D. and Stucki D.J. "Design Early considered Harmful: Graduated Exposure to Complexity and Structure Based on Levels of Cognitive Development". ACM SIGCSE Technical Symposium on Computer Science Education. (SIGCSE). 2000.
- [Byr03] Byrne B. "Good top-down design methodologies tend to produce fully normalised designs anyway". Teaching, learning and assessment in databases (TLAD). 2003.
- [BZ94] Batra D. and Zanakis S.H. "A conceptual database design approach based on rules and heuristics". European Journal of Information Systems. Vol 3(3), pp 228-239. 1994.
- [CB06] Connolly T.M. and Begg C.E. "A Constructivist-Based Approach to Teaching Database Analysis and Design". Journal of Information Systems Education. Vol 17(1) pp 43-53. 2006,
- [CB09] Connolly T. and Begg C. Database Systems: A Practical Approach to Design, implementation and Management. 5th Edition Addison Wesley. 2009.
- [CBF07] Choren R., Blois M. and Fuks H. "QUEST: an assessment tool for web-based learning". Proc. of World conference of the www, Internet and Intranet. 1998.
- [CCE04] ACM/IEEE Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering. Disponible en [http://www.acm.org/education/education/curric\\_vols/CE-Final-Report.pdf](http://www.acm.org/education/education/curric_vols/CE-Final-Report.pdf). 2004.
- [CCR07] Cabot J., Clarisó R., Riera D. "UMLtoCSP: a Tool for the Formal Verification of UML/OCL Models using Constraint Programming". Proc. of 22nd IEEE/ACM Int. Conf. on Automated Software Engineering (ASE). 2007.
- [CCS08] ACM/IEEE Computer Science Curriculum 2008. Disponible en <http://www.acm.org//education/curricula/ComputerScience2008.pdf>. 2008.
- [CE98] Charman D. and Elmes A. "Computer Based Assessment (Volume 1): A guide to good practice". SEED Publications, University of Plymouth. 1998.
- [CEA+03] Carter J., English J., Ala-Mutka K., Dick M., Fone W., Fuller U. and Sheard J. "How shall we assess this ?". ACM SIGCSE Bulletin. Vol 35 (4). pp 107-123. 2003.
- [Che76] Chen P., "The entity-Relationship model. Towards a Unified View of Data". ACM Transactions on Database Systems. Vol 1(1), pp 9-36, 1976.

- [Chr99] Christie J.R. "Automated Essay Marking for both Style and Content", Proc. of 3rd Annual Computer Assisted Assessment Conference. 1999.
- [CIS02] ACM/IEEE Curriculum Guidelines for Undergraduate Degree Programs in Information Systems. Disponible en [http://www.acm.org/education/education/curric\\_vols/is2002.pdf](http://www.acm.org/education/education/curric_vols/is2002.pdf). 2002.
- [CIT08] ACM/IEEE Curriculum Guidelines for Undergraduate Degree Programs in Information Technology. Disponible en <http://www.acm.org/education/curricula/IT2008Curriculum.pdf>. 2008.
- [CJC06] Carte T.A., Jasperson J. and Cornelius M.E. "Integrating ERD and UML Concepts When Teaching Data Modeling". Journal of Information Systems Education. Vol 17 (1) pp 55-64. 2006.
- [CL04] Connolly T.M. and Laiho M. "Database technology professional European survey analysis". Disponible en <http://www.dbtechnet.org/>. 2004.
- [CM03] Chalmers D. and McAusland D.M. "Computer-assisted assessment". The handbook for economics lecturers. Disponible en <http://www.economicsnetwork.ac.uk/handbook/caa/> 2003.
- [Cod70] Codd E.F. "A relational Model of Data for Large Shared Data Banks". Communications of the ACM. Vol 13(6), pp 377-387. 1970.
- [Cod71] Codd E.F. "Further normalization of the data base relational model". Database Systems: Courant computer Science Symposia Series 6, pp 33-64. Prentice Hall 1971.
- [Cod74] Codd E.F. "Recent Investigations into Relational Data Base Systems". IBM Research Report RJ1385. 1974.
- [Col03] Coleman J. "Online Assessment of SQL Query Formulations Skills". Proc. of fifth Australasian Computer Science Education Conference. pp 247-256, 2003.
- [Cor01] Corbett A. "Cognitive Computer Tutors: Solving the Two-Sigma Problem". User Modeling. LNAI 2109, pp 137-141. 2001.
- [CS00] Constantino-Gonzalez M. and Suthers D.D. "A coached collaborative learning environment for entity relationship modeling" Proc. of 5th Int.Conf. Intelligent Tutoring System (ITS). pp 1-10, 2000.
- [CS98] Canup M and Shackelford R. "Using software to solve problems in large computing courses". ACM SIGCSE Bulletin. Vol 30 (1), pp 135-139, 1998.
- [CSE04] ACM/IEEE Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. Disponible en <http://sites.computer.org/ccse/SE2004Volume.pdf> 2004.
- [CSM06] Connolly T.M., Stansfield M. and McLellan E. "Using an online Games-Based learning approach to teach database design Concepts". Electronic Journal of e-learning. Vol 4(1), pp 103-110, 2006.
- [Cul98] Culwin F. "Web hosted assessment: possibilities and policy". ACM SIGCSE Bulletin. Vol 30(3), pp 55-58. 1998.
- [CurACM] ACM/IEEE Curricula Recommendations. Disponible en <http://www.acm.org/education/curricula-recommendations>.
- [Dai99] Daly C. "RoboProf and an introductory computer programming course". Proc 4th Annual SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education. pp 155-158. 1999.

- [Dal01] Dalziel J. "Enhancing web-based learning with computer assisted assessment: Pedagogical and Technical considerations". Proc. of Intl. Computer Assisted Assessment Conference, 2001.
- [Dat04] Date C.J. An Introduction to Database Systems, 8th Edition. Addison Wesley 2004.
- [DEP97] Dietrich S.W., Eckert E. and Piscator K. "WinRDBI: a Windows-based Relational Database Educational Tool", Proc. of the 28th ACM SIGCSE Technical Symposium on Computer Science Education. pp 126-130, 1997.
- [DF03] Davis M. and Fitzpatrick M. "Virtura - A virtual tutor for relational algebra" Teaching, Learning and Assessment of Databases (TLAD). pp 25-29, 2003.
- [DG99] Dalziel J. and Gazzard S. "Next generation computer assisted assessment software: the design and implementation of WebMCQ". Proc. of 3rd Annual Computer Assisted Assessment Conference. 1999.
- [DJP09] Domínguez C., Jaime A. and Pérez T. "¿Todavía interesa normalizar bases de datos? Reflexionando sobre su enseñanza". XV Jornadas de la Enseñanza Universitaria de la Informática (JENUI). 2009.
- [DK01a] Duke-Williams E. and King T. "Testing Higher Learning Outcomes with CBA", Handbook of ILTAC 2001. The Institute for Learning and Teaching in Higher Education. Conference: Professionalism in Practice, University of York, 2001.
- [DK01b] Duke-Williams E. and King T. "Using Computer-Aided Assessment to Test Higher Level Learning Outcomes". Proc. of 5th Annual Computer Assisted Assessment Conference, 2001.
- [DRL06] Dekeyser S., de Raadt M. and Lee T. "Do students SQLify?. Improving learning outcomes with Peer review and enhanced computer assisted assessment of Querying Skills". Proc. of the 6th Baltic Sea conference on Computing Education Research. 2006.
- [DRL07] Dekeyser S., de Raadt M. and Lee T. "Computer Assisted Assessment of SQL Query Skills". Proc. of the 18th Australasian Database Conference (ADC). 2007.
- [DU96] Dietrich S. and Urban S. "Database theory in practice: learning from cooperative group projects". Proc. of 27th SIGCSE technical symposium on Computer Science Education. pp 112-116. 1996.
- [DU98] Dietrich S. and Urban S. "A Cooperative learning Approach to Database Group Projects: Integrating Theory and Practice". IEEE transaction on Education. Vol 41(4), pp 1-14, 1998.
- [DW99] Du H. and Wery L. "Micro: A normalization tool for relational database designers". Journal of Network and Computer applications. Vol 22(4), pp 215-232, 1999.
- [ECZ08] Enright J., Chu B. and Zaiane O.R. "Intelligent Functional Dependency Tutoring tool". Proc. of 16th International Conference on Computers in Education (ICCE). 2008.
- [Ell08] Elliot B. "Assessment 2.0: Modernising assessment in the age of web 2.0". Scottish Qualifications Authority, 2008.
- [Elm04] Jürgens Elmar. "Database Normalizer". Disponible en <http://home.in.tum.de/~juergens/DatabaseNormalizer/index.htm> . 2004.
- [EN03] Eaglestone B. and Nunes M.B. "Pragmatics and practicalities of teaching and learning in the Quicksand of database Syllabus" Proc. of Teaching, Learning and Assessment of Databases (TLAD). pp 4-12. 2003.

- [EN07] Elmasri R. and Navathe B. Fundamentals of Database Systems, 5th ed. Addison Wesley, 2007.
- [Eng02] English J, "Experience with a Computer-Assisted Formal Programming Examination". Proc. of Conference on Innovation and Technology in Computer Science Education (ITiCSE). pp 51-54. 2002.
- [ES00] English J. and Siviter P. "Experience with an automatically assessed course". Proc. of Conference on Innovation and Technology in Computer Science Education (ITiCSE). pp 168-171, 2000.
- [FDK05] Fessakis G., Dimitracopoulou A. and Komis V. "Improving database design teaching in secondary education : action research implementation for documentation of didactic requirements and strategies". Computers in Human Behavior. Vol 21, pp 159-194, 2005.
- [FHG+01] E. Foxley, C. Higgins, T. Hegazy, P. Symeonidis and A. Tsintsifas. "The CourseMaster CBA system: Improvements over Ceilidh". Proc. of Computer Assisted Assessment Conference. 2001.
- [FIB] Universidad Politécnic de Catalunya. Facultat de Informàtica. <http://www.fib.upc.edu/fib/infoAca/estudis/assignatures/BD.html>
- [FJM98] Farthing D.W., Jones D.M. and McPhee D. "Permutational Multiple-Choice Questions: An Objective and Efficient Alternative to Essay-Type Examination Questions". Proc. of Conference on Innovation and Technology in Computer Science Education (ITiCSE). pp 81-85, 1998.
- [FK05] Fanguy R. and Kleen B. "Normalization Shootout: a competitive game that impacts student learning". Issues in Information Systems. Vol 6(1), pp 21-27, 2005.
- [FKN+02] Fong J., Kwan I., Ng M. and Lau K.L. "Fifth Normal Form Made Easy with Novel Web-based CAI HCI". Proc. first Int. Conf. on web based learning, LNCS 2436, pp. 398-410, 2002.
- [FLM98] Frosini G., Lazzerini B. and Marcelloni F. "Performing automatic exams". Computers & Education Vol 31(3), pp 281-300, 1998.
- [Fot05] Fotache M. "Database Designers and Normalization: anatomy of a divorce". Proc. of 8th International Conference on Bussines Information Systems. pp 330-341, 2005
- [Fot06a] Fotache M. "A simpler and more practical-oriented methodology for database design". Proc. 6th Conference International Business Information Management Association (IBIMA). pp 237-243. 2006.
- [Fot06b] Fotache M. "Why normalization failed to become the ultimate guide for database Designers ?". Social Science Research Network Working Paper Series. 2006.
- [FRB05] Fantini A., Radice B. and Bocca E. "Enfoques para la evaluación formativa en e-learning". Congreso de Tecnologías de la Información y la Comunicación en la Enseñanza de las Ciencias (TICEC). pp 150-156, 2005.
- [FW65] Forsythe G. and Wirth N. "Automatic grading programs". Communications of ACM. Vol 8(5), pp 275-278, 1965.
- [Gar89] Garcia Ramos J.M. Bases pedagógicas de la evaluación. Ed. Síntesis. 1989
- [Geo08] Georgiev N. "A web based environment for learning normalization of relational database schemata". Master's thesis in Computer Science. 2008.

- [GEV+05] Gutiérrez J.J., Escalona M.J., Villadiego D. and Mejias M. "Comparativa de herramientas para la enseñanza de lenguajes relacionales". XI Jornadas de Enseñanza Universitaria de la Informática (JENUI). 2005.
- [Gib92] Gibbs G. "Assessing More Students". Oxford Centre for Staff Development. Oxford Brookes University, 1992.
- [GM06] Golden D. and Matos V. "Introducing the Unified Modeling Language into the information Systems curriculum". Journal of Information Systems Education. Vol 17(1), pp 83-92, 2006.
- [GSD99] Gomez F., Segami C. and Delaune C. "A system for the semiautomatic generation of ER models from natural language specifications". Data & Knowledge Engineering. Vol 29, pp 57-81, 1999.
- [GW00] Groff J.R. and Weinberg P. Guía LAN Times de SQL. Ed. Osborne Mc-Graw-Hill. 2000.
- [Har04] Harrison G. "Computer-Based Assessment Strategies in the Teaching of Databases at Honours Degree Level 1". 21st Annual British National Conference on Databases (BNCOD). LNCS 3112, pp 257-264, 2004.
- [HB07] He L. and Brandt P. "Weas: a web-based educational assessment system". ACM Southeast Regional Conference. pp 126-131, 2007.
- [HCP+02] Hernández C., Crespo Y., Romay P. and Laguna M.A. "Una Herramienta para el Aprendizaje del Álgebra Relacional". VIII Jornadas de enseñanza Universitaria de la Informática (JENUI). 2002.
- [HD03] Hameed K. and Dudley C. "A record of success – Database Teaching at Staffordshire University" Teaching, Learning and Assessment in Databases (TLAD). pp 89-93, 2003.
- [Hea71] Heath, I. "Unacceptable File Operations in a Relational Database" Proc. ACM SIGFIDET Workshop on Data Description, Access, and Control. 1971.
- [HG06] Higgins C.A. and Bligh B. "Formative Computer Based Assessment in Diagram Based Domains". Proc. 11th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE), pp 98-102, 2006.
- [HG98] Hall L. and Gordon A. "A Virtual learning Environment for Entity Relationship Modeling" Proc. of the 29th ACM SIGCSE Technical Symposium on Computer Science Education. pp. 345-353, 1998.
- [HGS+05] Higgins C., Gray G., Symeonidis P. and Tsintsifas A. "Automated assessment and experiences of teaching programming". ACM Journal of Educational Resources in Computing. Vol 5(3), pp 1-21, 2005.
- [HHS+03] Higgins C., Hegazy T., Symeonidis P. and Tsintsifas A. "The CourseMarker CBA system: Improvements over Ceilidh". Education and Information Technologies. Vol 8(3), pp 287-304, 2003.
- [Hir88] Hirmanpour I. "A student system development diagrammer". Proc. of the 19th SIGCSE Technical Symposium on Computer Science Education. pp 104-108, 1988.
- [HL98] Hoggarth G. and Lockyer M. "An Automated Student Diagram Assessment System". ACM SIGCSE Bulletin. Vol 30(3), pp 122-124, 1998.
- [Hol59] Hollingsworth J. "An educational program in computing". Communications of the ACM. Vol 2(8), pp 6, 1959.

- [Hol60] Hollingsworth J. "Automatic graders for programming classes". *Communications of the ACM*. Vol 3(10), pp 528-529. 1960.
- [HPV08] Hernán I., Pareja C. and Velázquez J.A. "Testing-Based Automatic Grading: A proposal from Bloom's Taxonomy" 8th IEEE International Conference on Advanced Technologies (ICALT). pp 847-849, 2008.
- [HSI07a] Haji Ali N., Shukur Z. and Idris S. "A design of an Assessment System for UML class diagram". *Proc. of International Conference on Computational Science and its Applications (ICCSA)*. pp 539-544, 2007.
- [HSI07b] Haji Ali. N., Shukur Z. and Idris S. "Assessment System for UML class diagram using notations extraction". *International Journal of Computer Science and Network Security (IJCSNS)*. Vol 7(8), 2007.
- [HST02a] Higgins C., Symeonidis P. and Tsintifas A. "Diagram-Based CBA using DATsys and Coursemaster". *International Conference on Computers in Education (ICCE)*. 2002.
- [HST02b] Higgins C, Symeonidis P. and Tsintifas A, "The Marking System for CourseMaster". *Proc. Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE)*. pp 46-50, 2002.
- [HW69] Hext J. and Winings J. "An automatic grading scheme for simple programming exercises". *Communications of ACM*. Vol 12(5), pp 272-275, 1969.
- [Ino01] Inoue Y. "Questionnaire Surveys: Four Survey Instruments in Educational Research". *Educational Resources Information Center (ERIC)*, 2001. Available from: <http://eric.ed.gov/>
- [JA00] Johnstone A.H. and Ambusaidi A., "Fixed Response: What are we testing?". *Chemistry Education: Research and Practice in Europe*. Vol 1(3), p323-328, 2000.
- [Jac00] Jackson D. "A semi-automated approach to online assessment". *Proc. of the 5th Annual SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education (ITiCSE)*. pp 164-167, 2000.
- [Jeg05] Jegede O. "Towards a New Paradigm for Assessment in Lifelong Learning", Presented at the International Association for Educational Assessment (IAEA). 2005.
- [JF06] Johson C. and Fuller U. "Is Bloom's taxonomy appropriate for Computer Science?". *Proc. 6th Baltic Sea conference on Computing Education Research*. pp 120-123. 2006.
- [JGB05] Joy M., Griffiths N. and Boyatt R. "The BOSS on-line submission and assessment system". *ACM Journal on Educational Resources in Computing*. Vol 5(3). 2005.
- [JHR+09] Jones K., Harland J., Reid J. and Barlett R. "Relationship between examination questions and Bloom's taxonomy" 39th ASEE/IEEE Frontiers in Education Conference. 2009.
- [JMM+04] Juwah C., Macfarlane-Dick D., Matthew B., Nicol D., Ross D. and Smith B. "Enhancing student learning through effective formative feedback". *The Higher Education Academy* 2004.
- [JND+09] James A., Nelson D., Davis K., Cooper R. and Monger A. "Ideas in Teaching, Learning and assessment of databases". *British National Conference on Databases (BNCOD)*. LNCS 5588, pp 213-224, 2009.
- [JU97] Jackson D. and Usher M. "Grading student programs using ASSYST". *Proc. 28th SIGCSE technical symposium on Computer Science Education*. pp 335-339, 1997.

- [KBB73] Krathwohl, D.R., Bloom, B.S. and Bertram, B.M. Taxonomy of educational objectives, the classification of educational goals. Handbook II: affective domain. David McKay Co. Inc., New York. 1973
- [KC04] Kung H. and Case T. "Traditional and alternative database normalization techniques: their impacts on IS/IT students perceptions and performance". International journal of information technology education. 2004.
- [KG06] Kroenke D.M. and Gray C.D. "Toward a next generation data modeling facility: Neither the Entity-Relationship Model nor UML Meet the Need". Journal of Information Systems Education. Vol 17(1), pp 29-38, 2006.
- [KM00] A. Korhonen and L. Malmi. "Algorithm simulation with automatic assessment". Proc. of Conference on Innovation and Technology in Computer Science Education (ITiCSE). pp 160-163, 2000.
- [Kni01] Knight P. "A Briefing on Key Concepts: Formative and summative, criterion and norm-referenced assessment". LTSN Generic Centre: Assessment Series. 2001.
- [KP05] Kenny C. and Pahl C. "Automated tutoring for a Database Skills Training Environment". ACM SIGCSE Technical Symposium on Computer Science Education 2005. pp 58-62, 2005.
- [Kro04] Kroenke D. Database Processing: Fundamentals, design and Implementation. 9th Edition. Prentice Hall. 2004.
- [KSF97] Kearns R., Shead S. and Fekete A. "A Teaching System for SQL". Proc. of Second Australasian Computing Education Conference (ACE). pp 224-231, 1997.
- [KT02] Kung H. and Tung H., "A Web-based Tool to enhance Teaching/Learning Database Normalization". Proc. Southern Association for Information System Conference, pp. 251-258, 2002.
- [KT06] Kung H. and Tung H. "An alternative approach to teaching database normalization: a simple algorithm and an interactive e-learning tool". Journal of information systems education. Vol 17(3). 2006.
- [KT07] Kung H. and Tung H. "Comparing two bottom-up database design methods". Proc. of the 2007 Southern Association for Information systems Conference. 2007.
- [KTT+93] Kashy E., Tsai Y., Thoenssen M. and Morrissey D. "Capa: An Integrated Computer Assisted Personal Assignment System". American Journal of Physics. 61(12). 1993.
- [Lai01] Laine H. "SQL-Trainer". Proc. First Annual Finnish/Baltic Sea Conference on Computer Science Education. pp 13-17, 2001.
- [Lar01] Larman C. Applying UML and Patterns. An Introduction to Object-oriented Analysis and Design. Prentice Hall 2001.
- [Leap] LEAP RDBMS: An educational relational database management system.  
<http://leap.sourceforge.net/>
- [LH92] Lim B. and Hunter R., "DBTool: A graphical Database Design Tool for an Introductory Database course" Proc. of the 23th ACM SIGCSE Technical Symposium on Computer Science Education, pp. 24-27, 1992.
- [Lis01] Lister R, "Objectives and Objective Assessment in CS1". ACM SIGCSE Bulletin. Vol 33(1), pp 292-296. 2001.
- [LRN] .LRN. <http://dotlrn.org/>

- [LSK05] Laakso M., Salakoski T. and Korhonen A. "The feasibility of automatic assessment and feedback". Proc. of Cognition and Exploratory Learning in Digital Age (CELDA). pp 113-122. 2005.
- [MAB05] McMaster K., Anderson N. and Bilyeu-Dittman D. "A Reverse Life-Cycle Database Course With Mini-Projects". Proc. of Information Systems Education Conference. (ISECON). 2005.
- [MAB06] McMaster K., Anderson N and Blake A., "Teaching Relational Algebra and Relational Calculus: A programming Approach". Proc. of 23rd Annual Information Systems Education Conference ISECON. pp 1-8. 2006.
- [Mac00] Machanick P. "Experience of applying Bloom's Taxonomy in Three Courses ". Proc. Southern African Computer Lecturers Association Conference. pp 135-144, 2000.
- [Man06] Mannino M. "Notation usage in Data Modeling Education". Journal of Information Systems Education. Vol 17 (1), pp 27-28, 2006.
- [MB99] McKenna C. and Bull J., "Designing effective objective test questions: an introductory workshop", Computer Assisted Assessment (CAA) Centre 1999. Disponible en <http://www.caacentre.ac.uk/dldocs/otghdout.pdf>
- [Mca02] McAlpine M. "Principles of assessment". Computer Assisted Assessment (CAA) Centre. 2002. Disponible en <http://www.caacentre.ac.uk/dldocs/Bluepaper1.pdf>
- [MD07] Martínez-González M.M. and Duffing G. "Teaching databases in compliance with the European dimension of higher education: Best practices for better competences." Education and Information Technologies. Vol 12 (4), pp 211-228. 2007.
- [MG02] Mason O. and Grove-Stephenson I. "Automated free text marking with Paperless School", Proc. of 6th Annual Computer Assisted Assessment Conference, 2002.
- [MG09a] Murray M. and Guimaraes M. "Animated database courseware (ADbC): interactive instructional materials to support the teaching of database concepts". Journal of Computing Sciences in Colleges. Vol 24(4), pp 267. 2009.
- [MG09b] Murray M. and Guimaraes M. "Animated courseware Support for Teaching Database design". Issues in Informing Science and Information Technology. Vol 6. pp 201-211. 2009.
- [Mit02] Mitrovic A., "Normit: a Web-enabled Tutor for database Normalization". Proc. of International Conference on Computers in Education (ICCE), pp 1276-1280. 2002.
- [Mit03] Mitrovic A., "An intelligent SQL tutor on the web", International Journal of Artificial Intelligence in Education. Vol 13(2), pp 173-197, 2003.
- [Mit06] Mitrovic A. and the ICTG team. "Large-Scale Deployment of three Intelligent web-based Database Tutors". 28th International Conference on Information Technology Interfaces (ITI), pp 135-140, 2006.
- [Mit98a] Mitrovic A., "Learning SQL with a computerized tutor," Proc. of the 29th ACM SIGCSE Technical Symposium on Computer Science Education, pp. 307-311, 1998.
- [Mit98b] Mitrovic A. "A Knowledge-based teaching system for SQL". Proc. of ED\_MEDIA. pp 1027-1032, 1998.
- [MK04] Malmi L. and Korhonen A. "Automatic feedback and resubmissions as learning Aid". 4th. IEEE International Conference on Advanced Learning Technologies (ICALT) pp.186-190, 2004.

- [MKS02] Malmi L, Korhonen A, Saikkonen R, “Experiences in Automatic Assessment on mass Courses and issues for designing virtual courses”, ACM SIGCSE Bulletin. Vol 34(3), pp 55-59, 2002.
- [MM04] Marshall M., Mitrovic A. “ERM-Tutor an ER-to-Relational Mapping Tutor”. BSc Honours Research Report. University of Canterbury. 2004.
- [MMM06] Milik N., Marshall M. and Mitrovic A. “Responding to free-form student questions in ERM-Tutor”. Proc. of 8th International Conference Intelligent Tutoring System (ITS). LNCS 4053, pp 707-709, 2006.
- [Moodle] Moodle. <http://moodle.org/>
- [MP03] Mikulecká J. and Poulová P. “The innovative teaching methods in the database courses on FIM UHK. Proc. of Teaching, Learning and Assessment of Databases (TLAD). pp 84-88. 2003.
- [MPW95] McIntyre D.R., Pu H. and Wolff F.G. “Use of software tools in teaching relational database design”. Computers & Education. Vol 24(4), pp. 279-286, 1995.
- [MS00] Mohtashami M. and Scher J.M. “Application of Bloom’s Cognitive Domain Taxonomy to Database Design”. Proc. of Information Systems Education Conference ISECON. 2000.
- [MSM+04] Mitrovic A., Suraweera P., Martin B. and Weerasinghe A. “DB-Suite: Experiences with Three Intelligent Web-Based Database Tutors”. Journal of Interactive Learning Research. Vol 15(4), pp 409-432, 2004.
- [Mul03] Muller R.B. Database Design for Smarties: Using UML for Data Modeling. The Morgan Kaufmann Publishers Inc. 2003.
- [MW98] Mason D. and Woit D. “Integrating technology into computer science examinations”, ACM SIGCSE Bulletin. Vol 30(1), pp 140-144, 1998.
- [MW99] Mason D. and Woit D. “Providing mark-up and feedback to students with online marking”, Proc. of the 30th SIGCSE Technical Symposium on Computer Science Education. pp 3-6. 1999.
- [Mye86] Myers R. "Computerized Grading of Freshman Chemistry Laboratory Experiments". Journal of Chemical Education Vol(63), pp 507-509, 1986.
- [Nee07] Neely M.P. “Mastery level learning and the art of database design”. Proc. 30th Americas Conference on Information Systems. pp 1-9, 2007.
- [Neu02] Neubauer B. “Data modeling in the undergraduate database course: adding UML and XML modeling to the traditional course content”. Journal of Computing Sciences in Colleges. Volume 17(5), pp.147-153, 2002.
- [Ngu06] Nguyen-Thinh L. “A constraint-based Assessment Approach for Free-Form design of class diagrams using UML”. Proc. of the Sixth International Conference on Intelligent Tutor Systems (ITS). pp 11-19, 2006.
- [Njo03] Njovu C. “Teaching, learning and assessment on Msc Database Units at University of Greenwich”. Teaching, Learning and Assessment in Databases (TLAD). pp. 89-93. 2003.
- [NM01] Naiburg J. and Maksimchuk R.A. UML for database design. Ed Addison-Wesley. 2001.
- [NS04] Naugler D. and Surendran K. “Simplicity first: Use of tools in Undergraduate Computer Science and Information Systems Teaching”. Information Systems Education Journal, Vol 2(5), pp 4-11, 2004.

- [OpenM] Openmark. <http://www.open.ac.uk/openmarkexamples/>
- [OHM04] Omar N., Hanna P. and McKeivitt P. "Heuristics-based entity-relationship modeling through natural language processing". 15th Irish Conference on Artificial Intelligence and Cognitive Science (AICS). pp 302-313. 2004.
- [PBJ+03] Poch J., Barrabés E., Juher D., Ripoll J., Soler J. and Calsina A. "ACME 2.0 un sistema de evaluación continuada y ayuda a la resolución de problemas". XI Congreso Universitario de Innovación Educativa en las Enseñanzas Técnicas. 2003.
- [PBK04] Pahl C., Barret R. and Kenny C. "Supporting active database learning and training through interactive multimedia". Proc. 9th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE). pp 27-31, 2004.
- [PBS+05a] Prados F., Boada I., Soler J. and Poch J. "Automatic generation and correction of technical exercises". International Conference on Engineering and Computer Education (ICECE). 2005. Disponible en <http://acme.udg.edu>
- [PBS+05b] Prados F., Boada I., Soler J. and Poch J. "An Automatic Correction Tool for relational Database Schemas," International Conference on Information Technology based higher Education and Training, (IHTET). S3C pp 9-14, 2005. Disponible en: <http://ieeexplore.ieee.org/Xplore/login.jsp?url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F10436%2F33131%2F01560331.pdf%3Farnumber%3D1560331&authDecision=-203>
- [PBS+06] Prados F., Boada I., Soler J. and Poch J., "A web based-tool for Entity-Relationship Modeling" Proc. of International conference on Computacional Science and its Applications (ICCSA). LNCS 3980, pp 364-372, 2006. Disponible en: <http://www.springerlink.com/content/1852j58546437256/>
- [PCP06] Prados F., Castro F. and Pujol J.J. "Use of the Information and Communication Technologies in a first course of computer programming". Proc 35th International IGIP Symposium. 2006.
- [PH03] Pain D. and Le Heron J. "Webct and online assessment: the best thing since SOAP?". Educational Technology & Society, Vol 6(2), pp 62-71, 2003.
- [Phi07] Philip G.C. "Teaching Database Modeling and Design: Areas of Confusion and Helpful Hints". Journal of Information Technology Education. Vol 6, pp- 481-496, 2007.
- [PMO04] Purchase H, Mitchell C. and Ounis I. "Gauging students' understanding through interactive lectures". British National Conference on Databases (BNCOD). LNCS 3112, pp 234-243, 2004.
- [Pos05] Post, G.V. Database Management System. 3th Edition McGraw-Hill 2005.
- [Pou95] Pouyioutas P. "NOCAT- a normalisation CASE tool" Transactions on Information and Communications Technologies. Vol 10, pp 277-285, 1995.
- [Pow99] Power C. "Designer: a logic diagram design tool". Proc. of the 4th Annual SIGCSE / SIGCUE on Innovation and Technology in Computer Science Education. p 211, 1999.
- [Qmark] Question Mark Computing Ltd. [www.qmark.com](http://www.qmark.com)
- [Qt99] Quatrani T. Visual Modeling with Rational Rose 2000 and UML, Addison-Wesley Professional, 1999.
- [Rat] IBM Rational Rose <http://www-01.ibm.com/software/rational/>

- [RBP+96] Rumbaugh J., Blaha M., Premerlani W., Hedí F. and Lorensen W. Modelado y diseño orientados a objetos. Metodología OMT. Editorial Prentice Hall 1996.
- [RC04] Rob P. and Coronel C. Database Systems: Design, Implementation and Implementation. 6th Edition. Ed. Thomson 2004.
- [RDL07] Raadt M., Dekeyser S. and Lee T. "A system employing peer review and enhanced computer assisted assessment of querying skills". Informatics in Education. Vol 6(1) pp 163-178. 2007.
- [Rela] Relational. Disponible en <http://galileo.dmi.unict.it/wiki/relational/doku.php>
- [RG03] Ramakrishnan R., Gehrke J. Database Management Systems. 3th Ed. McGraw Hill 2003.
- [RH83] Rottmann R.M. and Hudson H.T. "Computer Grading as an instructional Tool". Journal of College Science Teaching. Vol 12(3), pp 152-156, 1983.
- [RJE02] Rawles S., Joy M. and Evans M. "Computer Assisted Assessment Review Exercise". LTSN Centre for Information and Computer Sciences. 2002.
- [RMR+02] Rivero E, Martínez L, Reina L., Benavides J. and Olaizaola J.M. Introducción al SQL para usuarios y programadores. Ed. Thomson 2002.
- [Rob00] Robbert M.A. "Enhancing the value of a Project in the Database course". Proc. 31th SIGCSE Technical Symposium on Computer Science Education. pp 36-40. 2000
- [Rob08] Robling G. et al. "Enhancing learning Management Systems to better support computer Science Education". ACM SIGCSE Bulletin. Vol 40 (4), pp 142-166, 2008.
- [Row87] Rowntree D. "Assessing Students: How shall we know them?". Ed. Kogan Page, 1987.
- [RQuery] RelationalQuery. Disponible en <https://relationalquery.dev.java.net/>
- [RR03] Robbert M. and Ricardo C.M. "Trends in the evolution of the Database Curriculum" Proc. Innovation and Technology in Computer Science Education (ITiCSE). pp. 139-143, 2003.
- [Rum06] Rumbaugh J. "ER is UML". Journal of Information Systems Education. Vol 17(1), pp 21-26, 2006.
- [Rus01] Rust C. "A Briefing on Assessment of Large Groups". LTSN Generic Centre Assessment Series. 2001.
- [RVV+07] Regueras, L.M., Verdú E., Verdú M.J. Pérez M.A. and de Castro J.P. "E-learning Strategies to Support Databases Courses: a Case Study". First International Conference on Technology, Training and Communication. 2007.
- [RWG+00] Robbert, M.A., Wang, M., Guimaraes, M. and Myers, M.E. "The Database Course: What must be taught". ACM SIGCSE Bulletin, 32, pp 403-404, 2000.
- [SBP+06] Soler J., Boada I., Prados F. and Poch J. "A web-based Problem-Solving Environment for database Normalization". Proc. of 8th Internacional Symposium on Computers in Education (SIIE). 2006. Disponible en <http://acme.udg.edu>
- [SBP+07] Soler J., Boada I., Prados F., Poch J. and Fabregat R. "An automatic correction tool for relational Algebra Queries". Proc. of International Conference on Computational Science and its Applications (ICCSA). LNCS 4706, pp 861-872, 2007. Disponible en: <http://www.springerlink.com/content/u077163px8r66858/>

- [SBP+09a] Soler J., Boada I., Prados F., Poch J. and Fabregat R. "Database design using a web-based e-learning tool". Proc. 2nd Workshop on Methods and Cases in Computing Education (MCCE). pp 31-40, 2009. Disponible en: <http://www.scribd.com/doc/14226792/MCCE-2009-Proceedings>
- [SBP+09b] Soler J., Boada I., Prados F., Poch J. and Fabregat R. "Experiencia docente en diseño de bases de datos con la ayuda de herramientas de e-learning". Actas XV Jornadas de la Enseñanza Universitaria de la Informática (JENUI). pp 63-70, 2009. Disponible en: <http://upcommons.upc.edu/revistes/bitstream/2099/7885/6/p152.pdf>
- [SBP+10a] Soler J., Boada I., Prados F., Poch J. and Fabregat R. "A web based e-learning tool for UML class diagrams". Proc. of IEEE Engineering Education Conference (EDUCON). 2010. Disponible en breve en ieeexplore.
- [SBP+10b] Soler J., Boada I., Prados F., Poch J. and Fabregat R. "A web-based e-learning tool for database design courses". International Journal of Engineering Education". Pendiente publicación 2010.
- [SBP+10c] Soler J., Boada I., Prados F., Poch J., Fabregat R. "A formative assessment tool for conceptual database design using UML class diagrams". International Journal of Emerging Technologies in Learning (iJET). Vol 5 (3). pp 27-33. 2010.
- [SC01] Stefani L. and Carroll J. "A Briefing on Plagiarism". LTSN Generic Centre: Assessment Series. 2001.
- [Sco03] Scott T. "Bloom's taxonomy applied to testing in computer science classes" Journal of Computing Sciences in Colleges. Vol 19 (1), pp 267-274, 2003.
- [Sel03] Selikoff S. "The Database normalization tool". 2003. Disponible en [http://dbtools.cs.cornell.edu/norm\\_index.html](http://dbtools.cs.cornell.edu/norm_index.html)
- [Sey02] Seyed-Abbassi B. "The evolution of an Advanced Database Course in an Information Systems Curriculum". Proc. of the Information Systems Education Conference. 2002.
- [SG06] Suleiman J. and Garfield M.J. "Conceptual Data Modeling in the Introductory Database Course: Is it Time for UML?". Journal of Information Systems Education. Vol 17(1), pp 93-100, 2006.
- [SKS05] Silberschatz A., Korth H. and Sudarshan S., Database System Concepts. 5th Edition McGraw Hill 2005.
- [SM02] Suraweera P. and Mitrovic A. "KERMIT: A Constraint-Based Tutor for Database Modeling" Proc. of 6th International Conference Intelligent Tutoring System (ITS). LNCS 2363, pp 377-387, 2002.
- [SM04] Suraweera P. and Mitrovic A. "An Intelligent Tutoring System for Entity Relationship Modelling". International Journal of Artificial Intelligence in Education. Vol 14, pp 375-417, 2004.
- [SM97] Stephens D. and Mascia J. Results of a Survey into the use of Computer-Assisted Assessment in Institutions of Higher Education in the UK. Ed. Professional Development, Loughborough University. 1997.
- [SMK01] Saikkonen R., Malmi L. and Korhonen A. "Fully Automatic Assessment of Programming Exercises". Proc. of Conference on Innovation and Technology in Computer Science Education (ITiCSE). pp 133-136, 2001.
- [SOS+04] Sadiq S., Orłowska M., Sadiq W. and Lin J. "SQLator- An Online SQL Learning Workbench". Proc. of Conference on Innovation and Technology in Computer Science Education (ITiCSE). pp 223-227, 2004.

- [SPB+02] Soler J., Poch J., Barrabés E., Juher D. and Ripoll J. "A tool for the continuous assessment and improvement of the student's skills in a mathematics course". Proc. of International Symposium of Technology of Information and Communication in Education for engineering and industry (TICE). pp 105-110, 2002.
- [SPB+06a] Soler J., Prados F., Boada I. and Poch J. "Activitats formatives virtuals com a complement a les activitats presencials. Experiència de la seva utilització en un Pla Pilot d'Adaptació al EEES". Jornades en Xarxa sobre l'Espai Europeu d'Ensenyament Superior. pp 66-76. UOC 2006. Disponible en <http://acme.udg.edu>
- [SPB+06b] Soler J., Prados F., Boada I. and Poch J. "Utilización de una plataforma de e-learning en la docencia de bases de datos". XII Jornadas de la Enseñanza Universitaria de la Informática (JENUI). pp 581-588, 2006. Disponible en <http://acme.udg.edu>
- [SPB+06c] Soler J., Prados F., Boada I. and Poch J., "A web based tool for teaching and learning SQL" Proc. of 7th International Conference on Information Technology Based Higher Education and Training (ITHET), 2006. Disponible en <http://acme.udg.edu>
- [SRR00] Springsteel F., Robbert M.A. and Ricardo C.M. "The next Decade of the Database Course: Three Decades speak to the next". ACM SIGCSE Bulletin. Vol 32(1), pp 41-45, 2000.
- [STW04] Smith N., Thomas P. and Waugh K. "Interpreting Imprecise Diagrams". Proc. 3rd International Conference on the Theory and Application of Diagrams. LNCS 2980, pp 239-241. 2004.
- [Swi06] Swithenby S.J. "E-assessment for open learning". European Association of Distance Teaching Universities (EADTU) Annual Conference. 2006.
- [TD76] Taylor J. and Deever D. "Constructed-Response, computer-graded homework". American Journal of Physics. Vol 44(6), pp 598-599. 1976.
- [The00] Thelwall M. "Computer-based assessment: a versatile educational tool" Computers & Education. Vol 34(1), pp 37-49, 2000.
- [Tho04a] Thomas, P. "Grading Diagrams Automatically", Technical Report No 2004/01, Department of Computing Open University. 2004.
- [Tho04b] P. Thomas. "Drawing Diagrams in an Online Examination". Proc. of the 8th Computer Assisted Assessment Conference. 2004.
- [Tho04c] P.Thomas. "Comparing machine graded diagrams with human markers". Technical Report 2004/27. Department of Computing Open University. 2004.
- [TLW+08] Thompson E., Luxton-Reilly A., Whalley J.L., Hu M. and Robbins P. "Bloom's Taxonomy for Computer Science assessment". Proc. 10th conference on Australasian Computing Education (ACE). Vol 78, pp 155-161, 2008.
- [TPP+01] Thomas P., Price B., Paine C. and Richards M. "Experiments with Electronic Examinations over the Internet". Fifth International Computer Assisted Assessment Conference. pp 487-502. 2001.
- [TPP+02] Thomas P., Price B., Paine C. and Richards M. "Remote Electronic Examinations: an architecture for their production, presentation and grading". British Journal of Educational Technology. Vol 33(5), pp 539-552, 2002.
- [TR97] Thorburn G. and Rowe G. "PASS: An Automated System for Program Assessment". Computers & Education. Vol 29(4), pp 195-206. 1997.

- [Tsi02] Tsintsifas A. "A Framework for the Computer Based Assessment of Diagram Based Coursework". Ph.D. thesis, University of Nottingham 2002. Disponible en <http://www.cs.nott.ac.uk/~azt/research.htm>
- [TSW05] Tselonis C., Sargeant J. and Wood M.M. "Diagram matching for Human-Computer Collaborative assessment," Proc. of 9th International Computer Assisted Assessment CAA Conference. 2005.
- [TSW06] Thomas P., Smith N. and Waugh, K. "An approach to the automatic grading of imprecise diagrams". Technical Report No 2006/16, Department of Computing Open University. 2006.
- [TSW07] Thomas P.G., Waugh K. and Smith N. "Learning and automatically assessing graph-based diagrams". Proc. 14th International Conference of the Association for Learning Technology. pp 61-74, 2007.
- [TSW08] Thomas P., Waugh K. and Smith N. "Automatically assessing graph-based diagrams". Learning, Media and Technology. Vol 33(3), pp 249-267, 2008.
- [TWS05] Thomas P., Waugh K. and Smith N. "Experiments in the automatic marking of ER-Diagrams". 10th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE). pp 158-162, 2005.
- [TWS07] Thomas P., Waugh K. and Smith N. "Tools for supporting the teaching and learning of data modelling". World Conference on Educational Multimedia, Hypermedia and Telecommunications (EDMEDIA). 2007.
- [TYF86] Teorey T., Yang D. and Fry J. "A Logical design methodology for Relational databases using the extended Entity-Relationship Model". ACM Computing Surveys. Vol 18(2), pp 197-222, 1986.
- [UD01] Urban S.D. and Dietrich S.W. "Advanced Database Concepts for Undergraduates: Experience with Teaching a Second Course". Proc. of the 32th SIGCSE technical symposium on Computer Science Education. pp 357-361, 2001.
- [Ull03] Ullman J.D. "Improving the Efficiency of Database-System Teaching". Proc. of the 2003 ACM SIGMOD international conference on Management of data. pp 1-3, 2003.
- [UPM] Universidad Politécnica de Madrid. <https://srvoei.eui.upm.es/web/>
- [UPV] Universidad Politécnica de Valencia. [http://www.upv.es/pls/oalu/sic\\_gdoc2.get\\_content?P\\_OCW=&P\\_ASI=5660&P\\_IDIOM A=c&P\\_VISTA=&P\\_TIT=&P\\_CACA=&P\\_CONTENT=unidades](http://www.upv.es/pls/oalu/sic_gdoc2.get_content?P_OCW=&P_ASI=5660&P_IDIOM A=c&P_VISTA=&P_TIT=&P_CACA=&P_CONTENT=unidades)
- [Use] Use: A UML-based specification environment. Gogolla M. Disponible en <http://www.db.informatik.uni-bremen.de/projects/USE/#ref5>
- [Ven96] Venable J.R. "Teaching Novice Conceptual data Modelers to Become Experts". International Conference on Software Engineering: Education and Practice. pp 50-56, 1996.
- [VT06] Virvou M. and Tourtoglou K. "An Adaptive Training Environment for UML". Proc. of the Sixth International Conference on Advanced Learning Technologies (ICALT). pp 147-149, 2006.
- [Wat06] Watson R.T. "The essential skills of Data Modeling". Journal of Information Systems Education. Vol 17(1), pp 39-42, 2006.

- [WDW+09] Wang J., Davis T., Westall M. and Srimani K. "Work in progress – Metube: A novel way to teach database to undergraduates". 39th ASEE/IEEE Frontiers in Education Conference. 2009.
- [WebCT] WebCT Inc., [www.webct.com](http://www.webct.com)
- [WebMCQ] WebMCQ Pty Ltd., [www.webmcq.com](http://www.webmcq.com)
- [WinRDBI] WinRDBI Educational Tool, <http://www.eas.asu.edu/~winrdbi/>
- [WS06] Winkler H. and Seip H. "Employment analysis of ERD vs. UML for Data Modeling". Journal of Information Systems Education. Vol 17(1), pp 65-72, 2006.
- [WTS04] Waugh K., Thomas P. and Smith N. "Toward the automated assessment of Entity-Relationship diagrams". Teaching, learning and assessment in Databases (TLAD). 2004.
- [WTS07] Waugh K., Thomas P. and Smith N., "Teaching and Learning Applications Related to the Automated Interpretation of ERDs". 24th British National Conference on Databases. (BNCS). pp 39-47. 2007.
- [YK07] Yazici A. and Karakaya Z. "JMathNorm: A database Normalization tool using Mathematica". Lectures Notes in Computer Science. LNCS 4488, pp.186-193, 2007.
- [ZKK05] Zhang L., Kaschek R. and Kinshuk, "Developing a Knowledge Management Support Systems for Teaching Database Normalization" Proc. of the 5th IEEE International Conference on Advanced Learning Technologies. pp 344-348, 2005.