

**EPS**

Escola Politècnica
Superior

Projecte/Treball Fi de Carrera

Estudi: Enginyeria Informàtica. Pla 1997

Títol: Desenvolupament d'una biblioteca per a la generació de diagrames vectorials basat en l'entorn gràfic de JAVA2D i aplicació d'aquesta biblioteca per a la implementació d'una eina per al desenvolupament de diagrames d'E/R

Document: Annexes

Alumne: CRISTINA CONSTANTINS ESGLEYES

Director/Tutor: JOSEP SUY FRANCH

Departament: Informàtica i Matemàtica Aplicada

Àrea: Llenguatges i Sistemes Informàtics

Convocatòria (mes/any): setembre 2008

INDEX

ANNEX A .-	MANUAL D'ÚS DE LA LLIBRERIA	2
ANNEX B .-	MANUAL D'INSTAL·LACIÓ I MANUAL D'ÚS DE L'EDITOR E/R.....	7
ANNEX C .-	INTRODUCCIÓ AL XML.....	22
ANNEX D .-	INTRODUCCIÓ AL DXF.....	31
ANNEX E .-	DIAGRAMES	43

ANNEX A .- MANUAL D'ÚS DE LA LLIBRERIA

A.1 .-	BÀSICS	3
A.2 .-	INCORPORAR LA LLIBRERIA A UN PROJECTE.....	3
A.2.1 .-	NETBEANS	3
A.3 .-	FORMES D'INTERACCIÓ PROGRAMA - LLIBRERIA	3
A.3.1 .-	OBJECTE MEDIADOR.....	4
A.3.2 .-	SISTEMA D'EVENTS.....	4
A.4 .-	ELS OBJECTES	5
A.4.1 .-	REPRESENTACIÓ VISUAL	5
A.4.2 .-	CREACIÓ D'OBJECTES VISUALS	6

A.1 .- BÀSICS

En el present projecte fi de carrera s'ha desenvolupat una llibreria (JavaDiagrammer), aquesta llibreria proporciona als programadors una interfície i unes funcionalitats bàsiques per a dissenyar i implementar programes que utilitzin una visualització basada en esquemes o diagrames.

La llibreria disposa d'un conjunt d'elements bàsics que es poden utilitzar com a representació d'un model de dades, però alhora permet que els programadors puguin definir elements propis i personalitzats a cada aplicació.

Es proporciona també un ampli ventall de funcionalitats que permeten entre altres coses manipular els elements de forma directa i interactuar amb l'entorn de treball.

La llibreria ha estat programada en Java, ja que es tracta d'un llenguatge multiplataforma i d'ús comú.

S'ha dotat la llibreria d'una petita aplicació en forma de test com a mostra del que permet la llibreria. També es pot utilitzar per a observar com funcionen els diferents mecanismes de què disposa la llibreria.

A.2 .- INCORPORAR LA LLIBRERIA A UN PROJECTE

La llibreria es pot trobar en forma d'arxiu .jar per així facilitar la seva distribució, utilització i incorporació en altres projectes. Molts IDE's incorporen mecanismes per a adjuntar de forma fàcil arxius .jar en els projectes que s'estan desenvolupant.

A.2.1 .- Netbeans

Primer de tot cal crear el projecte que utilitzarà la llibreria. Es pot crear com una "aplicació de Java".

Un cop s'ha creat el programa ja es pot incorporar la llibreria, per a fer això, és necessari situar-se en la subfinestra "Projects" i sobre l'etiqueta "Libraries" de l'arbre prémer el botó dret. Llavors s'escull l'opció "Add JAR/Folder ..." i es localitza la llibreria.

No és necessari que s'incorpori la llibreria just després d'haver creat el projecte, es pot afegir el .jar en el moment que el programador ho necessiti.

A.3 .- FORMES D'INTERACCIÓ PROGRAMA - LLIBRERIA

En la implementació de la llibreria s'ha seguit el disseny MVC (Model – Vista - Controlador) separant així el que és la visualització, del model o objectes i dels diferents

mecanismes que gestionen les diferents parts de la llibreria. Per altra part, l'ús d'aquest disseny facilita la interacció entre l'aplicació i la mateixa llibreria, ja que agrupa els punts de comunicació entre els dos sistemes.

La interacció entre el programa o aplicació que utilitza la llibreria i la mateixa llibreria es realitza principalment des de dos punts diferents: l'objecte mediador (*MediadorEntornVisual*) i els sistema d'events.

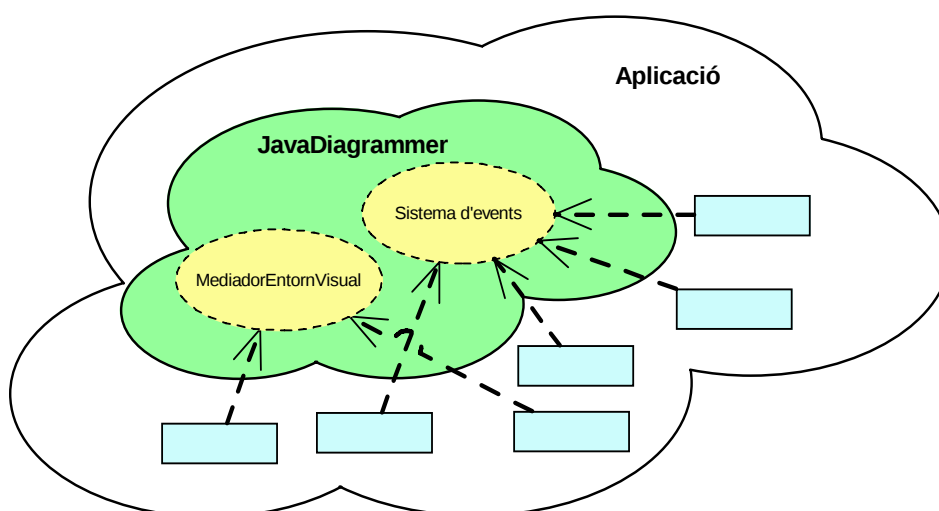


Figura 1.- Formes d'interacció entre l'aplicació i la llibreria

A.3.1 .- Objecte Mediador

S'ha definit un objecte mediador, el qual disposa de la gran majoria de funcionalitats que disposa la llibreria. Aquest mediador està implementat com una classe Singleton, és a dir, que es pot obtenir la seva única instància, sempre que sigui necessari, des de qualsevol punt del programa.

A.3.2 .- Sistema d'events

La llibreria implementa un sistema de gestió d'events basat en estats. Les operacions que necessiten certs events per a realitzar-se, es descomposen en estats que es van executant en funció dels events que es reben.

La classe Event que utilitza la llibreria és una implementació pròpia que permet definir els events sense d'haver-se de cenyir als events de què disposa Java. Aquest sistema permet també crear events no estrictament provinents de la interacció amb un element visual (jPanel, ElementRectangle, ...).

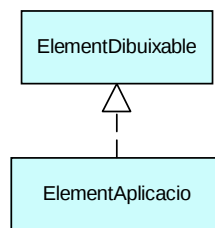
A.4 .- ELS OBJECTES

La llibreria disposa d'un conjunt limitat d'objectes bàsics, a més de la funcionalitat de convertir un conjunt d'elements en un element compost.

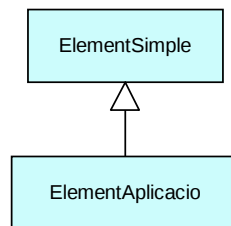
A.4.1 .- Representació visual

Els objectes creats han estat dissenyats per a ser la representació visual dels objectes model d'una aplicació. Una aplicació pot utilitzar els objectes de diverses formes, a continuació se'n descriuen algunes:

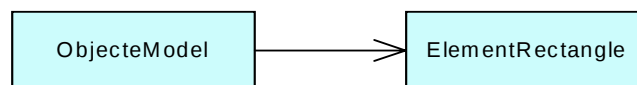
- Nous objectes que implementin alguna de les interfícies.



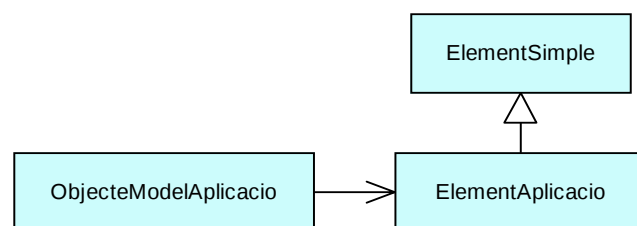
- Heretant dels objectes directament.



- Els objectes model del programa contenen un objecte visual de la llibreria com a vista.



- Els objectes model del programa contenen una herència o implementació dels objectes visuals de la llibreria, i l'utilitzen com la seva visualització.



A.4.2 .- Creació d'objectes visuals

A la hora de crear nous objectes visuals, tan si es tracta de la implementació d'una interfície com si es tracta d'herència, és necessari indicar a la llibreria sobre aquests nous objectes. Per afegir els nous objectes a la llibreria es pot fer de dues formes: utilitzant el mètode de l'objecte mediador, o bé, directament afegint l'objecte a la llista en qüestió (normalment *ColectorElementDibuixable*).

Els nous objectes es poden afegir a la vista de forma directa o mitjançant el ratolí utilitzant el sistema de gestió d'events. Per a fer-ho de forma directa implica donar les dimensions i la posició a l'element, per a afegir-lo a través del ratolí cal crear un event visual de tipus “*AFEGIR_ELEMENT*” i enviar-lo al “*GestorEstats*”. Un cop enviat l'event, el gestor d'estats canvia el seu estat i espera que l'usuari li doni les dades per afegir el nou element (posició i/o dimensions).

ANNEX B .- MANUAL D'INSTAL·LACIÓ I MANUAL D'ÚS DE L'EDITOR E/R

B.1 .-	MANUAL D'INSTAL·LACIÓ.....	8
B.1.1 .-	LLICÈNCIA	8
B.1.2 .-	PROGRAMARI.....	8
B.1.3 .-	INSTAL·LACIÓ	8
B.2 .-	MANUAL D'USUARI	11
B.2.1 .-	BARRA DE MENÚS	11
B.2.2 .-	BARRA D'ICONES	14
B.2.3 .-	ZONA DE DIBUIX	15
B.2.4 .-	ARBRES D'OBJECTES.....	15
B.2.5 .-	SISTEMES D'INFORMACIÓ DE L' APLICACIÓ	16
B.2.6 .-	FORMULARIS DELS OBJECTES.....	18
B.2.7 .-	ALTRES FORMULARIS.....	20

B.1 .- MANUAL D'INSTAL·LACIÓ

BDDDiagrammer és una aplicació que permet als programadors dissenyar diagrames Entitat-Relació de forma fàcil i dinàmica. Ha estat programat utilitzant la llibreria per a dissenyar diagrames: JavaDiagrammer.

El disseny de l'aplicació ofereix un accés ràpid a les funcionalitats més importants i una manipulació directa dels elements, proporcionant així un entorn simple i amigable.

BDDDiagrammer disposa d'un conjunt d'eines per a poder exportar els dissenys a formats d'imatge, de DXF, a informe HTML i a script SQL per a bases de dades Oracle.

B.1.1 .- Llicència

El programa ha estat realitzat per a fins docents com a projecte fi de carrera. No es permet la distribució ni l'ús del programa en cap àmbit que no sigui el docent.

B.1.2 .- Programari

És necessari disposar d'un sistema amb els següents requisits mínims:

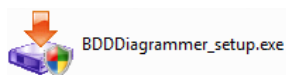
- Java 1.6
- Mínim 128MB de RAM, recomanat 512MB
- Mínim 3MB d'espai de disc
- Visualitzadors de formats PDF i HTML

B.1.3 .- Instal·lació

S'ha creat un fitxer instal·lable per a facilitar la instal·lació del programa. L'instal·lador funciona per a sistemes operatius Windows i s'encarrega de situar els fitxers en el directori del programa i crear enllaços en el menú inici i a l'escriptori.

Per a dur a terme la instal·lació del programa sols cal anar seguint els passos que indica l'instal·lador:

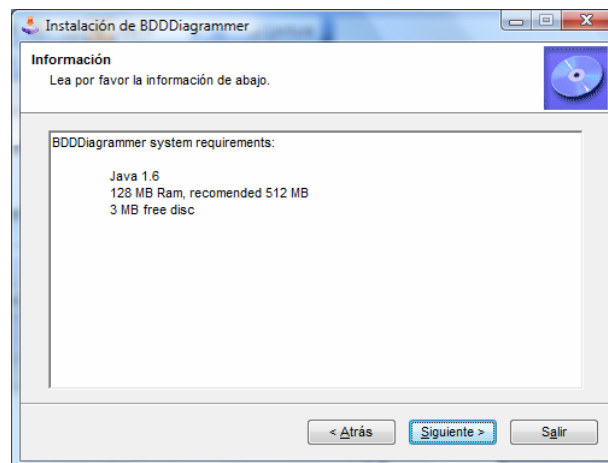
- Clicar sobre la icona del fitxer instal·lador.



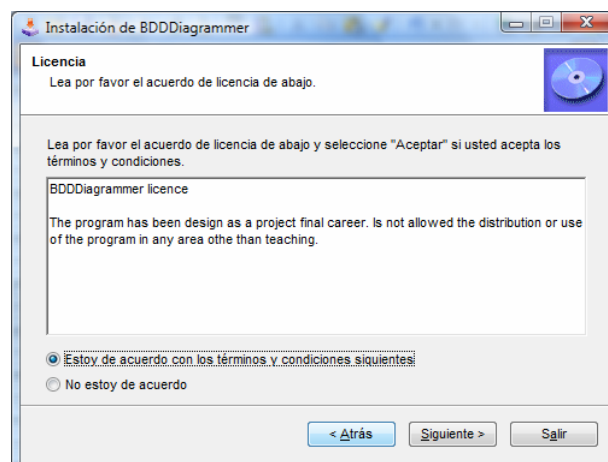
- L'instal·lador consta de diferents passos a seguir. Primer de tot es mostra una pantalla de presentació de la instal·lació.



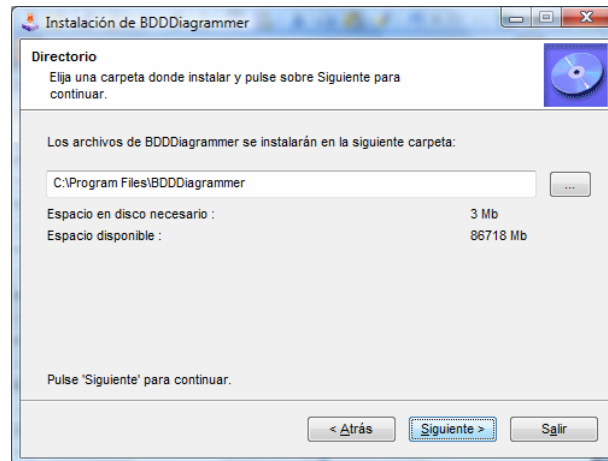
- La segona pantalla de l'instal·lador informa a l'usuari els requisits mínims del sistema, necessaris perquè el programa funcioni.



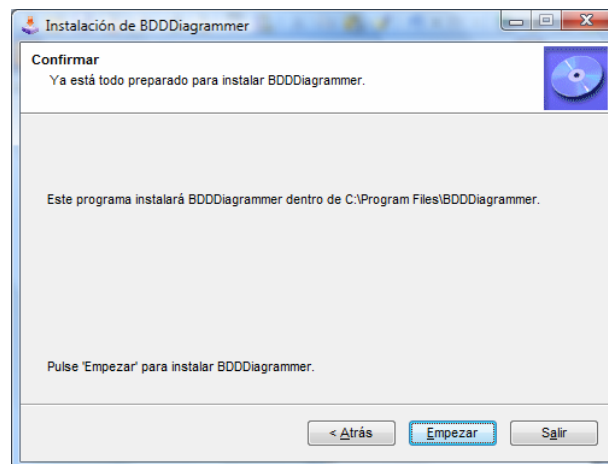
- La pantalla següent mostra la llicència del programa i es demana a l'usuari si accepta les condicions d'ús.



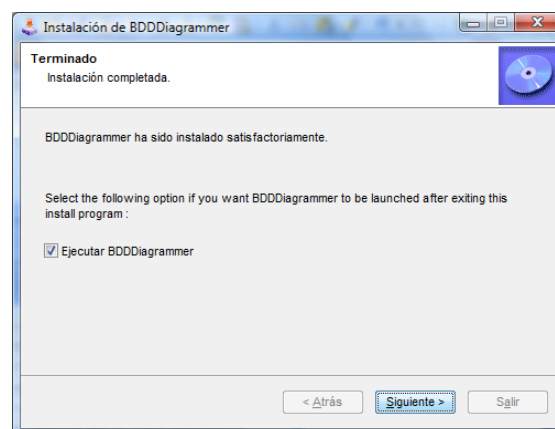
- Es proposa el directori on s'instal·larà el programa permetent canviar-lo pel que l'usuari vulgui.



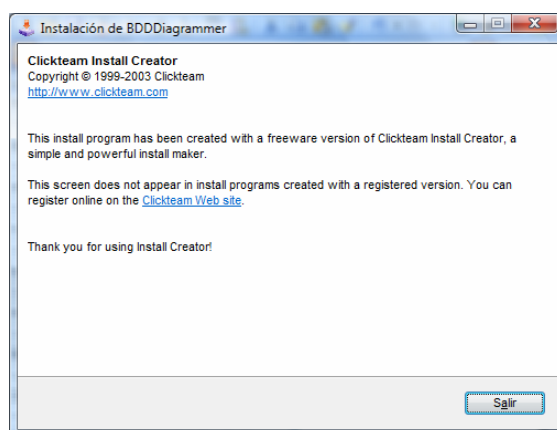
- S'informa a l'usuari que en el següent pas s'instal·larà l'aplicació, sent encara a temps per canviar alguns dels paràmetres de la instal·lació, o bé, cancel·lar-la.



- La següent pantalla informa a l'usuari s'ha dut a terme correctament i es demana si es vol executar l'aplicació.



- La última finestra que es mostra és de promoció de l'empresa programadora de l'instal·lador.



B.2 .- MANUAL D'USUARI

En el manual d'usuari es descriuen les diferents funcionalitats del programa. L'aplicació es basa en una pantalla principal i diverses pantalles secundàries. La pantalla principal conté les funcionalitats del programa.

Les diferents funcionalitats del programa han estat separades en zones per a proporcionar a l'usuari un espai de treball clar i senzill. A continuació es comenta les funcionalitats de cada una d'aquestes zones.

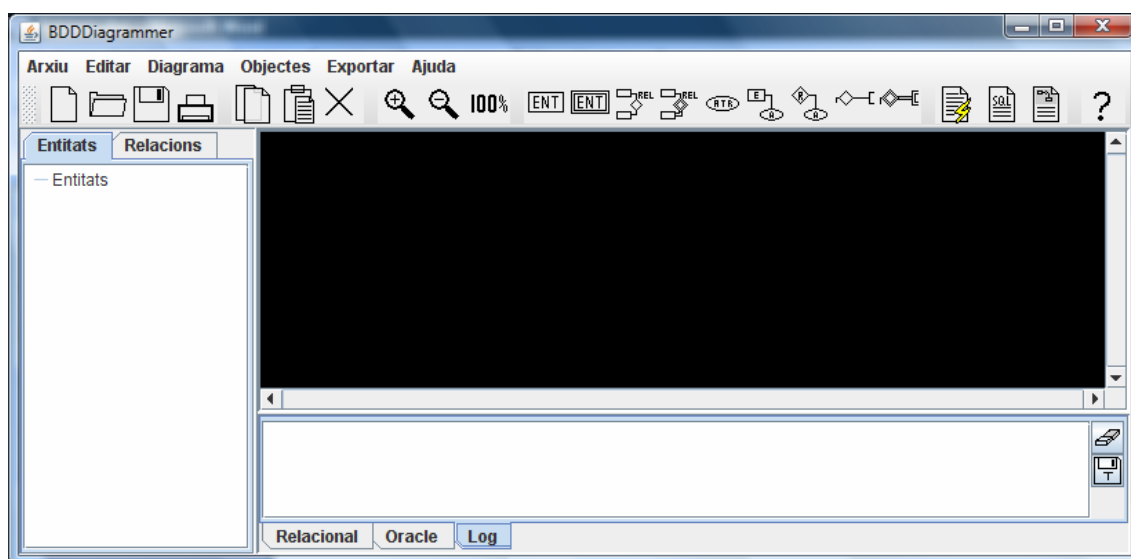


Figura 2.- Pantalla principal del programa BDDDiagrammer

B.2.1 .- Barra de menús

La barra de menús proporciona accés a totes les funcionalitats de què disposa l'aplicació. Aquesta barra està dividida en vàries parts:

- Arxiu. Engloba les funcionalitats bàsiques del tractament de projectes: crear un nou projecte; obrir un projecte guardat; guardar el projecte;

guardar el projecte en un altre path; imprimir el diagrama E/R creat i sortir de l'aplicació.

Arxiu	
Nou	Ctrl-N
Obrir	Ctrl-O
Guardar	Ctrl-G
Guardar com	
Imprimir	Ctrl-P
Sortir	Ctrl-S

Figura 3.- Menú d'arxiu

- Editar. Inclou les funcionalitats d'edició dels elements creats: copiar un o varis elements; enganxar els elements prèviament copiats; eliminar els elements seleccionats i editar el primer dels elements seleccionats (veure secció B.2.6).

Editar	
Copiar	Ctrl-C
Enganxar	Ctrl-V
Eliminar	Ctrl-D
Editar element	Enter

Figura 4.- Menú d'edició

- Diagrama. En el menú de diagrama es poden trobar aquelles funcionalitats que tenen efecte sobre l'entorn de treball, això inclou: apropar zoom; allunyar zoom; refrescar la visualització i obrir el formulari d'edició dels paràmetres generals del projecte (veure secció B.2.7).

Diagrama	
Zoom +	
Zoom -	
Refrescar	
Propietats generals	Ctrl-B

Figura 5.- Menú del diagrama

- Objectes. Proporciona els mètodes necessaris per a crear els objectes de l'aplicació. Permet crear entitats fortes i febles, atributs i connectors per

indicar de quina entitat o relació formen part, també permet crear relacions fortes, relacions febles i els seus connectors respectius cap a les entitats.

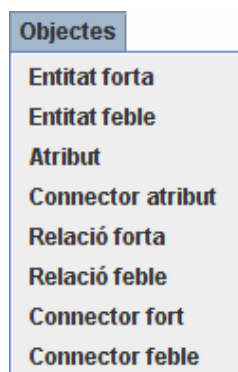


Figura 6.- Menú dels objectes

- Exportar. En el menú exportar es poden trobar les diferents formes d'exportar el projecte, també incorpora la funció d'anàlisi del diagrama. La funció d'anàlisi (check) avalua si l'usuari ha creat adequadament el diagrama entitat relació. El diagrama permet exportar-se en format script per una base de dades d'Oracle, també es pot exportar a informe en format HTML de la transcripció a Relacional del diagrama E/R. El diagrama creat també es pot guardar com a imatge o com a arxiu DXF.

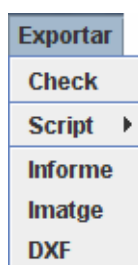


Figura 7.- Menú de l'exportar

- Ajuda. L'ajuda està formada per aquest manual d'ús en format pdf i per un formulari que conté les dades de l'aplicació.

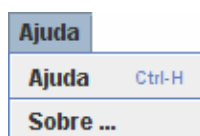


Figura 8.- Menú de l'ajuda

B.2.2 .- Barra d'icones

La barra d'icones és una agrupació de les funcionalitats més destacades de l'aplicació. Aquestes funcionalitats també es troben en la barra de menús. A continuació es mostra una taula amb la icona i la funcionalitat que representa.

	Crear un nou projecte
	Obrir un projecte guardat
	Guardar el projecte actual
	Imprimir el diagrama E/R
	Copiar els elements seleccionats
	Enganxar els elements copiats
	Aproximar zoom
	Allunyar zoom
	Zoom 100%
	Crear una entitat forta
	Crear una entitat feble
	Crear una relació forta
	Crear una relació feble
	Crear un atribut
	Connectar un atribut a una entitat
	Connectar un atribut a una relació
	Crear una connexió forta entre una entitat i una relació
	Crear una connexió feble entre una entitat i una relació
	Analitzar el diagrama E/R (check)
	Exportar a script per Oracle (en cas que el check sigui positiu)
	Exportar a informe HTML (en cas que el check sigui positiu)
	Obre el manual en pdf

B.2.3 .- Zona de dibuix

La zona de dibuix és una herència de la vista estàndard de la llibreria. Aquesta vista és la que rep els events de ratolí. La zona de dibuix permet les funcionalitats de seleccionar, editar, moure els elements,

Els elements es poden seleccionar fent un clic amb el ratolí sobre l'element. Una altra forma de seleccionar elements és dibuixant un rectangle amb el ratolí, llavors es seleccionen tots aquells elements que es troben absolutament dins el rectangle seleccionat.

Per a moure un element es pot fer un clic sobre l'element i, sense deixar anar el botó del ratolí, arrossegar l'element fins a la posició desitjada. També es poden moure diversos elements, sempre i quan s'hagin seleccionat anteriorment.

En realitzar un doble clic sobre un element, s'obre la finestra d'edició de les propietats de l'element (veure secció B.2.6).

La zona de dibuix també permet augmentar o disminuir el zoom mitjançant la rodeta del ratolí mentre es manté premuda la tecla "Control".

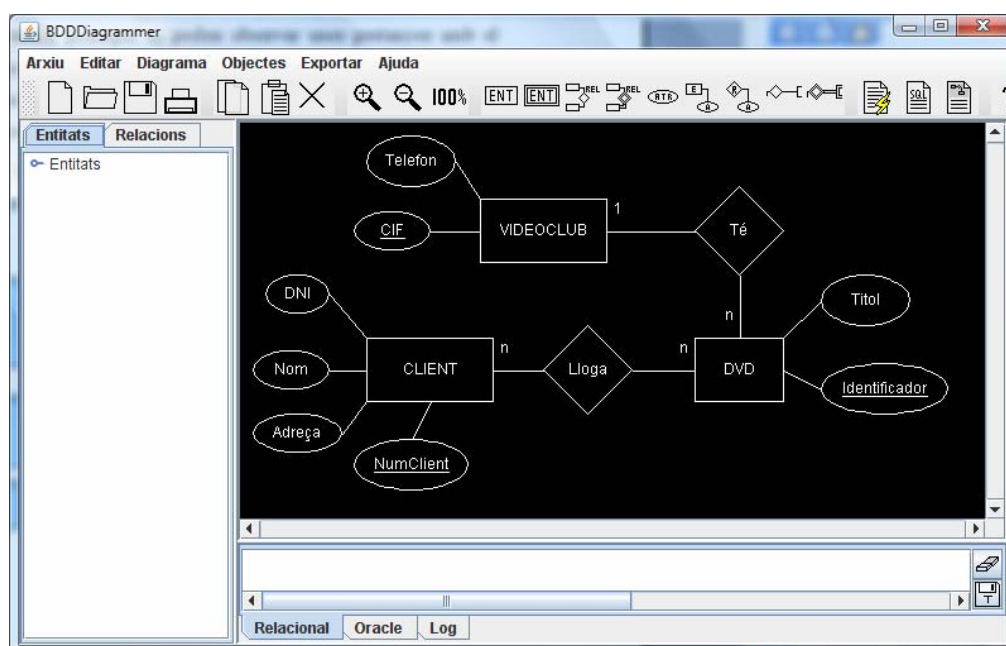


Figura 9.- Exemple del disseny d'una base de dades

B.2.4 .- Arbres d'objectes

A l'esquerra de la pantalla principal es poden observar unes pestanyes amb el nom: "Entitats" i "Relacions". En cada pestanya es pot observar un arbre amb les entitats o relacions que s'han anat creant en el diagrama, els atributs i en cas de l'arbre de relacions també les connexions cap a les entitats.

En realitzar un doble click sobre un dels nodes o fulles d'algun dels dos arbres (excepte el node arrel de l'arbre), s'obre la finestra d'edició de l'element (veure secció B.2.6).

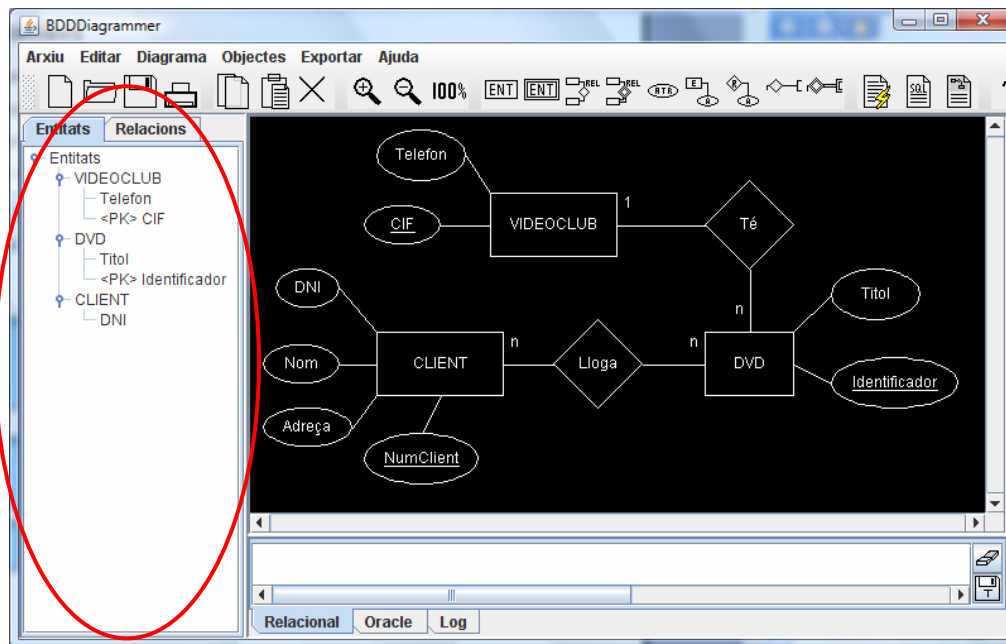


Figura 10.- Els arbres d'objectes en l'exemple del videoclub

B.2.5 .- Sistemes d'informació de l'aplicació

En la part inferior de la pantalla principal es poden observar unes pestanyes amb les etiquetes: "Relacional", "Oracle" i "Log".

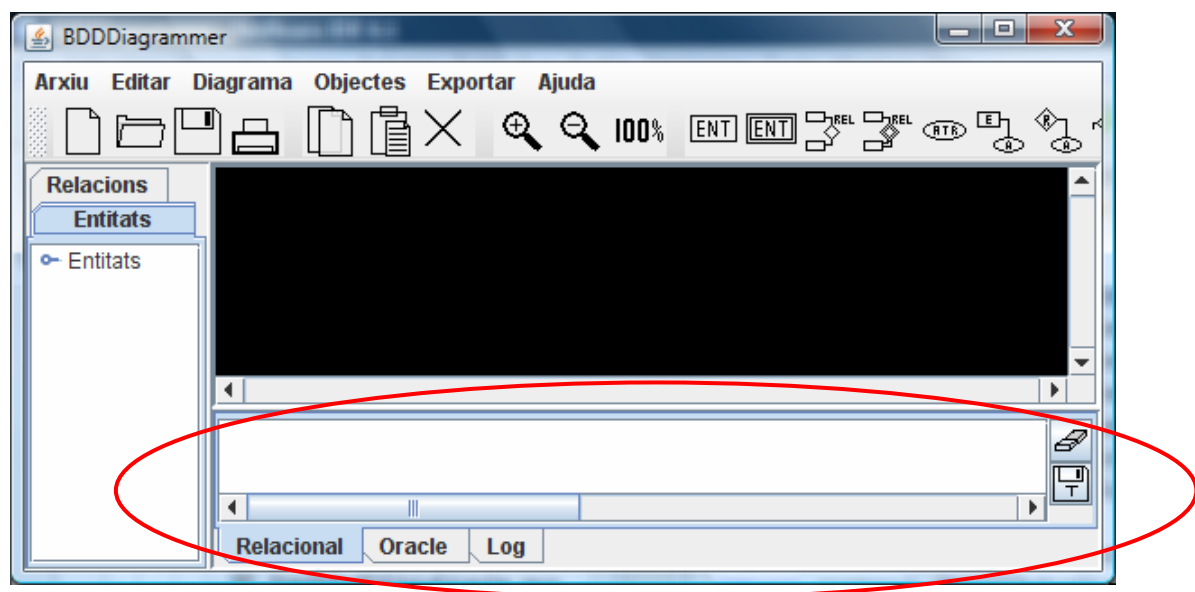


Figura 11.- Sistemes d'informació

Aquestes pestanyes mostren diferents texts informatius a l'usuari:

- Relacional. Mostra la transcripció a relacional del diagrama E/R creat. La transcripció realitzada es mostra en text pla. Per a poder realitzar aquesta transcripció és necessari que el diagrama de base de dades dissenyat sigui correcte.

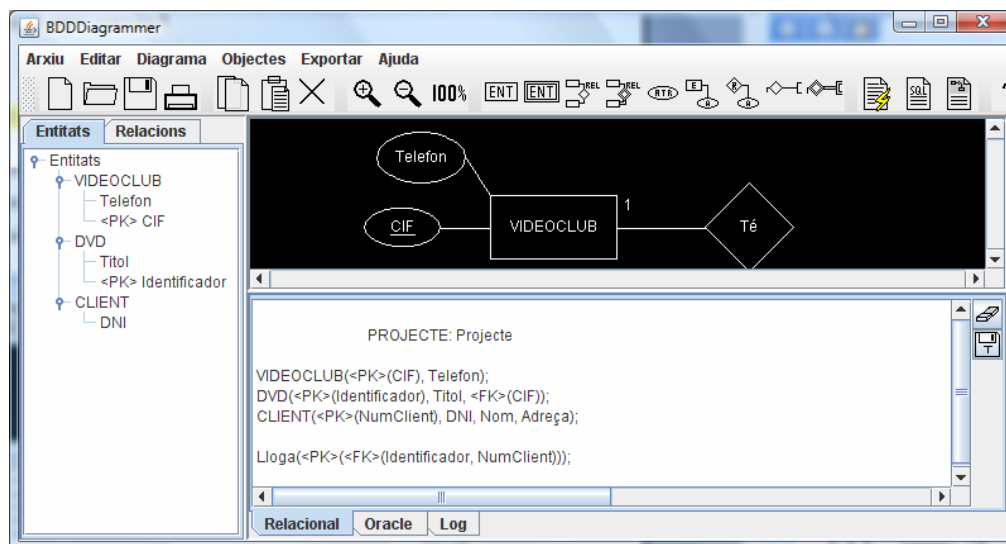


Figura 12.- Detall de la pestanya “Relacional”

- Oracle. Mostra la transcripció a script per a una base de dades Oracle. Per a poder realitzar aquesta transcripció és necessari que el diagrama de base de dades dissenyat sigui correcte.

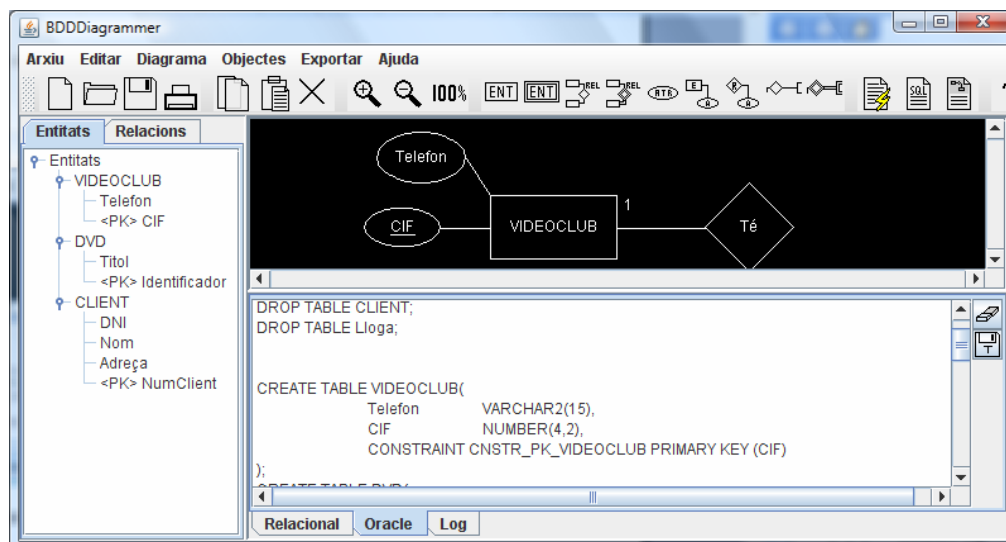


Figura 13.- Detall de la pestanya Oracle

- Log. En aquesta pestanya es descriu tot el que va succeint en el programa. Informa de: creacions d'objectes, errors detectats en durant l'execució, el resultat de l'anàlisi del diagrama E/R,

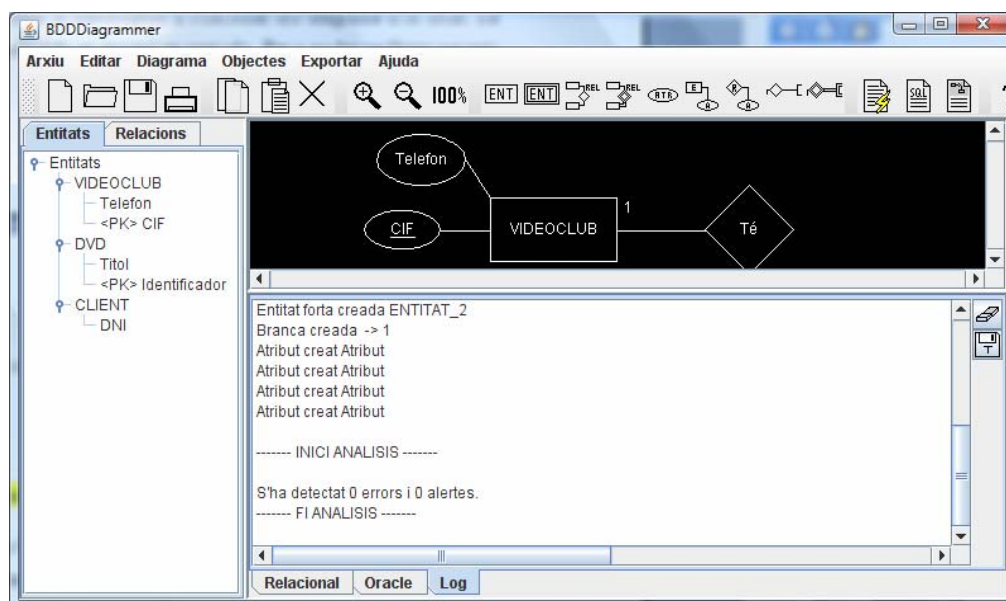


Figura 14.- Detall de la pestanya Log

Cada una de les pestanyes disposa de dos botons, el primer d'ells serveix per a eliminar el text mostrat, l'altre serveix per a guardar-lo a disc.

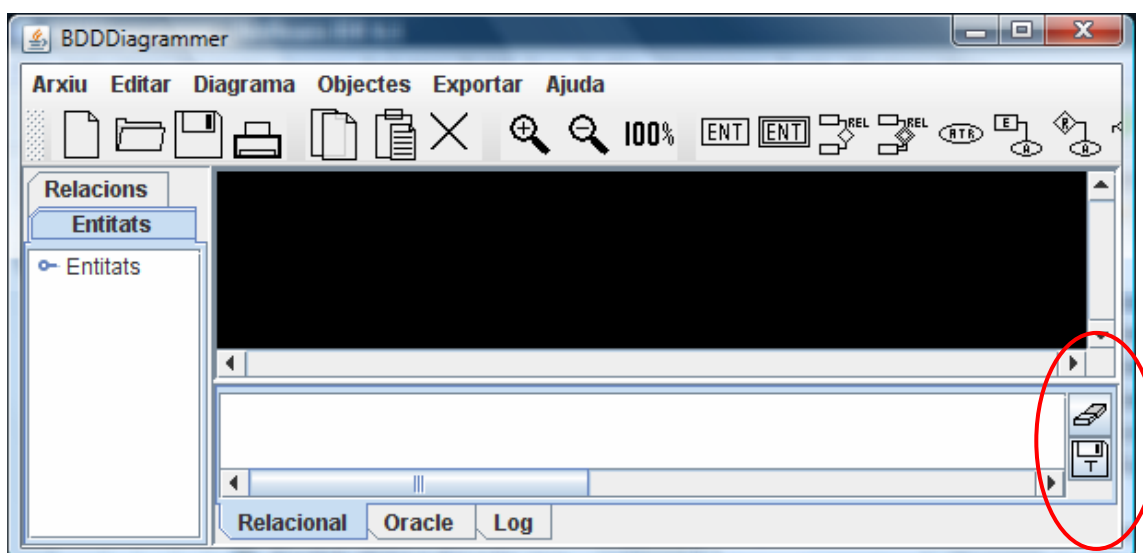


Figura 15.- Detall dels botons netejar i guardar

B.2.6 .- Formularis dels objectes

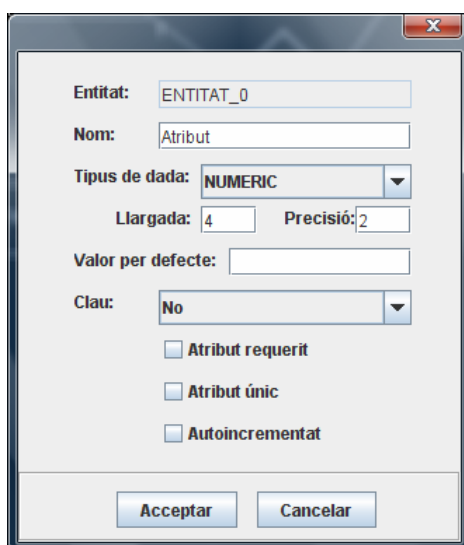
Cada tipus d'objecte té diferents propietats, per això cada tipus té el seu formulari d'edició particular.

- Entitat. Permet modificar el nom de l'entitat, una abreviació d'aquest nom i indicar si es tracta d'una entitat feble o forta. També permet visualitzar i editar els seus atributs.

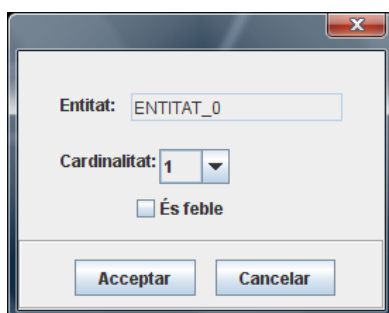
- Relació. El formulari d'edició de les relacions permet modificar el nom de la relació. També permet visualitzar i editar els atributs de la relació i els enllaços que té amb les diferents entitats que relaciona.

Entitat	Cardinalitat	Tipus
ENTITAT_0	1	Fora
ENTITAT_1	1	Feble

- Atribut. El formulari d'edició dels atributs permet modificar moltes propietats: el nom de l'atribut, el tipus de dades amb què es guardarà (String, número enter, real, ...), la precisió i la llargada del tipus de dada escollit, el valor per defecte, si es tracta d'un atribut clau primària o no, si l'atribut és requerit, únic o autoincrementat (pas = 1). També mostra a l'usuari de quina entitat o relació forma part.

Aquesta imatge mostra un formulari de propietats d'entitat. Els camps inclouen: 'Entitat:' amb el valor 'ENTITAT_0', 'Nom:' amb el valor 'Atribut', 'Tipus de dada:' amb un menú desplegable seleccionant 'NUMERIC', 'Llargada:' amb el valor '4' i 'Precisió:' amb el valor '2'. També hi ha un camp 'Valor per defecte:' buit i un menú desplegable 'Clau:' seleccionant 'No'. A sota hi ha tres caselles de selecció: 'Atribut requerit', 'Atribut únic' i 'Autoincrementat', totes desmarcades. Al final hi ha dos botons: 'Acceptar' i 'Cancelar'.

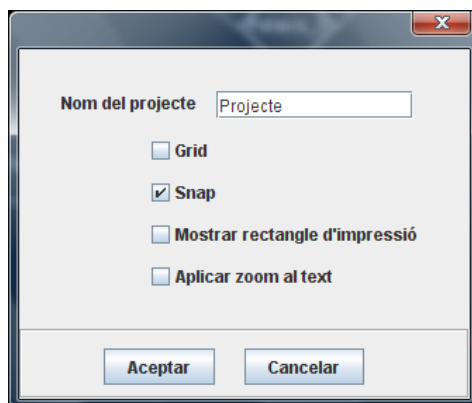
- Connector relació-entitat. El connector entre una relació i una entitat permet definir-ne la cardinalitat i si es tracta d'una part feble. En cas que el connector canviï de fort a feble o viceversa, aquest canvi també afecta a la relació. El formulari de propietats del connector relació-entitat també mostra a l'usuari quina entitat relaciona.

Aquesta imatge mostra un formulari de propietats de connector relació-entitat. Els camps inclouen: 'Entitat:' amb el valor 'ENTITAT_0' i 'Cardinalitat:' amb un menú desplegable seleccionant '1'. També hi ha una casella de selecció 'És feble' desmarcada. Al final hi ha dos botons: 'Acceptar' i 'Cancelar'.

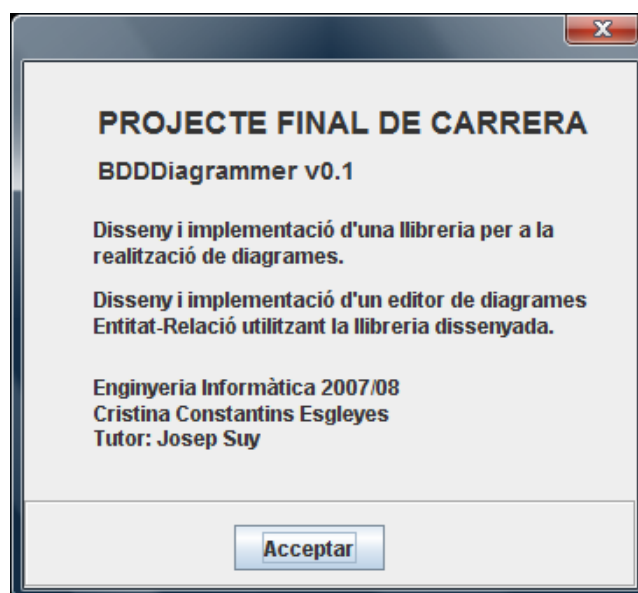
B.2.7 .- Altres formularis

El formulari de propietats generals permet definir el nom del projecte. Aquest nom de projecte s'utilitza com a nom per defecte a la hora de guardar o obrir qualsevol fitxer del programa.

En el formulari de propietats generals també: es pot definir si es vol utilitzar el grid o l'snap definits en l'aplicació, per així alinear els objectes; es pot definir que el text que es mostra se li apliqui el zoom i mostrar el rectangle de la zona d'impressió.



La finestra “Sobre ...” mostra les dades principals del projecte.



ANNEX C .- INTRODUCCIÓ AL XML

C.1 .-	CONCEPTES GENERALS.....	23
C.2 .-	SINTAXI.....	23
C.2.1 .-	COS PRINCIPAL.....	23
C.2.2 .-	ETIQUETES I ATRIBUTS.....	23
C.3 .-	NAMESPACES	26
C.4 .-	CSS	27
C.5 .-	PARSERS.....	29

C.1 .- CONCEPTES GENERALS

L'XML és un llenguatge que s'utilitza per a descriure dades mitjançant etiquetes no predefinides.

Conjuntament amb l'XML s'utilitza DTD (Document Type Declaration) [DTD03] o XML Schema (eXtensible Markup Language Schema) [XSD05] per descriure les dades o la informació representada en l'XML.

L'XML i l'HTML (HiperText Multimedia Language) [HTM05], tot i assemblar-se, han estat dissenyats amb dos propòsits molt diferents, el primer ha estat creat per a descriure dades i el segon per a mostrar dades.

Els usos de l'XML són molts i molt diversos, com per exemple:

- Separar dades o informació que es mostra en un HTML.
- Intercanviar informació entre sistemes incompatibles.
- Compartir o guardar informació.
- Crear nous llenguatges.
- ...

C.2 .- SINTAXI

La sintaxi de l'XML és simple però molt estricta. Sols cal conèixer unes quantes directrius per poder entendre com s'estructura un fitxer en XML.

C.2.1 .- Cos principal

En la primera línia del document es defineix la versió del XML i del sistema de codificació de caràcters utilitzat en el document. En l'exemple de la figura 16 observem que utilitza el sistema de codificació *ISO-8859-1*, però podria utilitzar altres com per exemple *windows-1252* o *UTF-8*.

La següent línia descriu l'element principal (root). En la última línia es tanca aquest element. En l'exemple de la figura 16 l'element principal és *amics*, que s'obre i es tanca amb *<amics>* i *</amics>* respectivament.

C.2.2 .- Etiquetes i atributs

Entre les etiquetes d'inici i fi de l'element principal hi poden haver altres elements, cadascun d'ells amb les respectives etiquetes d'inici i fi, en l'exemple observem que hi ha dos elements *persona*.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<amics>
  <persona edat="21">
    <nom>Maria</nom>
    <cognom>Pi</cognom>
    <ADRESA>C/Juliana 32 2n, 1a</ADRESA>
    <pobl>Girona</pobl>
    <telf>972 221 279</telf>
  </persona>
  <persona edat="25">
    <nom>Agnès</nom>
    <cognom>Llorenç</cognom>
    <ADRESA>C/Jaume I 7 1r, 1a</ADRESA>
    <pobl>Girona</pobl>
    <telf>972 205 341</telf>
  </persona>
</amics>

```

Figura 16.- Exemple de fitxer en XML.

Els comentaris s'escriuen entre els símbols `<!--` i `-->`.

Totes les etiquetes són “case sensitive”, és a dir, distingeixen entre majúscules i minúscules. En la figura 17 tenim un exemple de dos etiquetes que són diferents perquè una utilitza majúscules.

```

<aliment>
  <Tipus>Patata</Tipus>
  <tipus>Tubercle</tipus>
</aliment>

```

Figura 17.- Exemple de etiquetes diferents amb la mateixa paraula.

En cas d'utilitzar més d'una etiqueta per definir un *element*, cal que l'ordre de tancament de les etiquetes sigui invers al de la seva obertura. En l'exemple de la figura 18 es mostra un cas erroni d'etiquetes que no respecten l'ordre de tancament, en la figura 19 hi ha el mateix exemple amb les etiquetes en l'ordre correcte.

```

<persona>
<nom>
Anna
</persona>
</nom>

```

Figura 18.- Exemple incorrecte d'ordre d'etiquetes.

```

<persona>
<nom>
Anna
</nom>
</persona>

```

Figura 19.- Exemple correcte d'ordre d'etiquetes.

L'XML té dos maneres per definir les dades, mitjançant els *elements* i els *atributs*. Els *elements* són aquelles dades que estan definides mitjançant una etiqueta d'inici i una de fi. Els *atributs* són valors que estan inclosos dins les etiquetes i es defineixen amb un nom d'atribut, seguit d'un símbol d'igualtat, i del valor de l'atribut escrit entre cometes simples o dobles (accepta els dos casos).

Si recuperem el primer exemple (figura 16) observem que tenim elements *persona*, *nom*, *cognom*, *ADRESA*, *pobl*, *telf* i *amics*, i *edat* defineix l'únic atribut de l'exemple.

```

<amics>
  <persona edat="21">
    <nom>Maria</nom>
    <cognom>Pi</cognom>
    <ADRESA>C/Juliana 32 2n, 1a</ADRESA>
    <pobl>Girona</pobl>
    <telf>972 221 279</telf>
  </persona>
  ...
</amics>

```

Figura 20.- Fragment del primer exemple.

Els **elements** d'un XML mantenen relacions de parentesc de forma similar als arbres (estructures de dades).

Els **elements** poden tenir diferents continguts, variant aquests entre:

- Elements: l'element conté sols altres elements. Per exemple l'element *persona* de la següent figura.
- Mixta: l'element conté dades i elements. Per exemple l'element *nom* de la següent figura.
- Simple: sols conté dades. Per exemple l'element *dataNaix* de la següent figura.
- Buit: no conté res. Per exemple l'element *dataDefuncio* de la figura.

L'**element** buit (contingui o no **atributs** en la seva etiqueta), es pot abreviar tal com es mostra en l'exemple.

```
<persona>
  <nom>Jose
    <cognom>Saramago</cognom>
  </nom>
  <dataNaix>1922</dataNaix>
  <dataDefuncio></dataDefuncio>
</persona>
<persona>
  <nom>Ken
    <cognom>Follett</cognom>
  </nom>
  <dataNaix>1949 </dataNaix>
  <dataDefuncio />
</persona>
```

Figura 21.- Exemple de tipus d'elements.

Perquè el nom d'un **element** sigui vàlid, cal que no comenci per les caràcters XML, ni per números o signes de puntuació, tampoc poden contenir espais.

C.3 .- NAMESPACES

Per a poder llegir, crear i manipular documents XML, tan es pot fer mitjançant un editor de text, com d'un analitzador o parser, sense necessitat d'instal·lar cap tipus de control o eina. El mateix Internet Explorer porta incorporat l'analitzador de Microsoft. Els analitzadors ofereixen sistemes per a una visualització més ràpida dels elements (joc de colors,...).

A vegades hi pot haver conflictes amb els noms dels **elements**, ja que un mateix nom pot fer referència a dos **elements** totalment diferents. Per solucionar això existeixen els **namespaces**. Els **namespaces** són paraules que s'afegeix al nom de l'element per diferenciar-los en cas d'haver-ne dos iguals.

El prefix que porten els elements indiquen amb quin **namespace** estan relacionats. L'especificació de la W3C sobre **namespaces** estableix que els **namespaces** han de ser URI (Uniform Resource Identifier [URI05]).

En la següent figura es mostra com s'utilitzen els **namespaces**.

```
<p:persona xmlns:p="http://amics">
  <p:nom>African Coffee Table</p:nom>
  <p:ADRESA>80</p:ADRESA>
  <p:telf>120</p:telf>
```

```

</p:persona>
<e:persona xmlns:e="http://escriptors">
  <e:nom>African Coffee Table</e:nom>
  <e:ADRESA>80</e:ADRESA>
  <e:telf>120</e:telf>
</e:persona>

```

Figura 22.- Exemple de namespaces.

C.4 .- CSS

Els arxius XML es poden veure en el Internet Explorer 5, Netscape 6 o versions superiors.

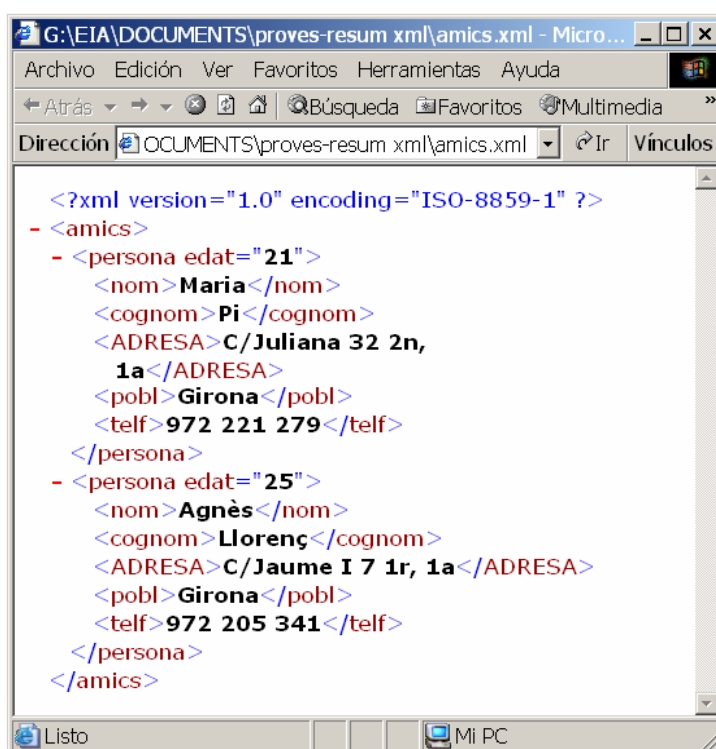


Figura 23.- Visualització des del Internet Explorer de l'exemple dels amics.

Un document XML en si sol no incorpora informació sobre com mostrar-se a pantalla, però existeixen diverses formes per definir com ho ha de fer. Una d'elles és utilitzant un arxiu CSS (Cascading Style Sheets) [CSS05]. Es tracta d'un fitxer enllaçat al XML que defineix com es mostra cada element. En la figura 24 s'observa l'instrucció continguda en el document XML per enllaçar amb el CSS.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/css" href="amics.css"?>
<amics>
  <persona edat="21">

```

```

    <nom>Maria</nom>
    <cognom>Pi</cognom>
    <ADRESA>C/Juliana 32 2n, 1a</ADRESA>
    <pobl>Girona</pobl>
    <telf>972 221 279</telf>
  </persona>
  <persona edat="25">
    <nom>Agnès</nom>
    <cognom>Llorenç</cognom>
    <ADRESA>C/Jaume I 7 1r, 1a</ADRESA>
    <pobl>Girona</pobl>
    <telf>972 205 341</telf>
  </persona>
</amics>

```

Figura 24.- Sentència d'enllaç en l'XML que l'indica com mostrar-se a pantalla.

En la figura 25 es mostra el contingut del fitxer CSS corresponent al XML anterior.

```

amics
{
background-color: #eeffbb;
width: 100%;
}
persona
{
display: block;
margin-bottom: 30pt;
margin-left: 0;
}
cognom, nom
{
color: #8800FF;
font-size: 20pt;
}
ADRESA
{
display: block;
color: #555555;
font-size: 15pt;
margin-left: 20pt;
}
pobl
{
display: block;
color: #555555;

```

```
font-size: 13pt;
margin-left: 20pt;
}
telf
{
display: block;
color: #000000;
}
```

Figura 25.- Sentències CSS que indiquen com es mostra l'exemple a pantalla.

Si obrim el fitxer anterior amb l'Internet Explorer, comprovem que ja no es visualitza el llenguatge XML amb les dades, sinó que es mostren sols les dades d'una forma bastant més clara.

En la figura 26 es pot observar l'exemple anterior (figures 24 i 25), vist des de l'Internet Explorer.

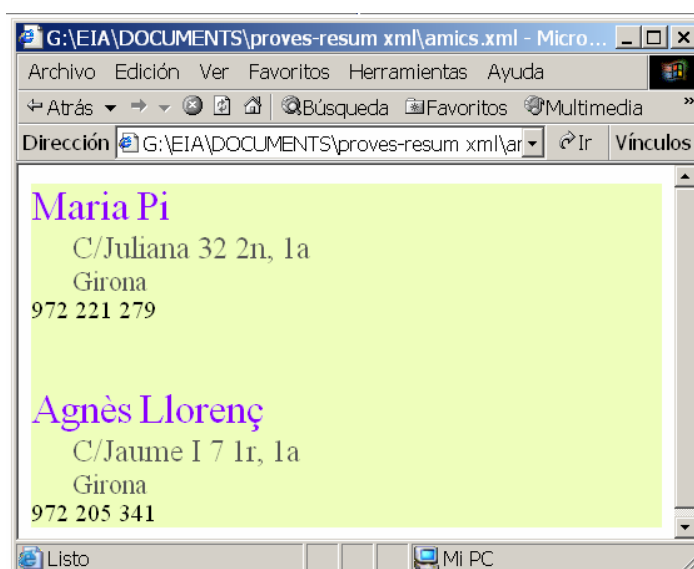


Figura 26.- Resultat mostrat a través de l'Internet Explorer.

C.5 .- PARSERS

Els analitzadors o parsers analitzen tot el text que forma un document XML. Per fer que un fragment del document no s'analitzi cal incloure'l entre una secció **CDATA**, identificada amb `<![CDATA[` com a marca d'inici i amb `]]>` com a marca de fi. Cal tenir present que en el contingut del **CDATA** no hi hagi la marca de fi.

```
<script>
<![CDATA[
function matchwo(a,b)
{
```

```
if (a<b && a<0) then
{
    return 1
}
else
{
    return 0
}
}]>
</script>
```

Figura 27.- Exemple de secció CDATA.

Amb una mica de creativitat es poden fer grans coses, utilitzant XML. Per exemple una pàgina web que mostri mitjançant uns botons la informació de cada un dels elements que conté l'XML.

ANNEX D .- INTRODUCCIÓ AL DXF

D.1 .-	INTRODUCCIÓ.....	32
D.2 .-	INTRODUCCIÓ ALS GRÀFICS VECTORIALS.....	32
D.3 .-	ESTRUCTURA DELS FITXERS R12 DXF	33
D.3.1 .-	SECCIÓ HEADER.....	34
D.3.2 .-	SECCIÓ TABLES	34
D.3.3 .-	SECCIÓ BLOCKS	34
D.3.4 .-	SECCIÓ ENTITIES	34
D.3.5 .-	MARCA DE EOF.....	34
D.4 .-	CODIS DE GRUP	34
D.5 .-	SECCIÓ ENTITIES.....	35
D.6 .-	ENTITAT POINT	35
D.7 .-	ENTITAT LINE	36
D.8 .-	ENTITAT POLYLINE	37
D.9 .-	ENTITAT CIRCLE	38
D.10 .-	ENTITAT ARC	38
D.11 .-	ENTITAT TEXT	39
D.12 .-	ESCRIURE/LLEGIR UN DXF	40
D.13 .-	LLEGIR UN FITXER DXF	40
D.14 .-	ESCRIURE UN FITXER DXF	41

D.1 .- INTRODUCCIÓ

El format de dibuix DXF és un format molt utilitzat en aplicacions industrials, inicialment utilitzat per a intercanviar informació entre diferents aplicacions. S'ha inclòs aquest format en la llibreria perquè es pugui incloure en qualsevol aplicació sigui del tipus que sigui.

Els fonaments teòrics sobre el format DXF i els gràfics vectorials per a traduir els gràfics, que es desenvolupin amb la llibreria, a DXF, han estat extrets del projecte “Eina per al disseny i dimensionat d'instal·lacions elèctriques de baixa tensió” d'Artur Juvé Vidal [AJV06]. Per a poder fer ús d'aquests coneixements s'ha demanat el consentiment explícit al seu autor.

D.2 .- INTRODUCCIÓ ALS GRÀFICS VECTORIALS

Els gràfics vectorials permeten modelar objectes en 2D i 3D, la informació d'aquest tipus de gràfics es basa en la combinació de figures geomètriques bàsiques (punt, recta, arc, cercle, etc.). La utilitat dels gràfics vectorials per computador dins l'enginyeria industrial és evident, gairebé en tots els camps són necessaris, per exemple en el disseny per MEF (Mètode dels Elements Finit), en el disseny de circuits electrònics, en el dibuix arquitectònic, etc.

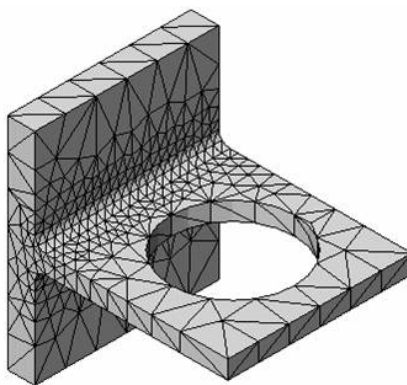


Figura 28.- Gràfics vectorials per computador, aplicació en MEF

Els gràfics vectorials per computador s'han estandarditzat en diversos formats, els més coneguts són els DXF, EPS, DWG, PICT, WMF. Aquesta varietat de formats és deguda a una extensa llista d'aplicacions que treballen amb aquesta tecnologia i que la majoria tenen el seu propi format de treball.

DXF (Drawing eXchange Format) és un fitxer molt usat dins l'entorn CAD/CAM/CAE i és suportat per la majoria de les aplicacions 2D/3D, permet

intercanviar dibuixos entre diferents usuaris, s'utilitza molt per importar/exportar els dibuixos entre diferents aplicacions.

El format de fitxer DXF és originari de l'aplicació AutoCAD de Autodesk, disposa de diverses versions (Release 10, Release 12, Release 14, DFX2000, etc.), però la versió R12 DXF és la més estesa en aplicacions CAD/CAM/CAE. En la figura D.2 es mostren els diferents formats de fitxers que permet guardar l'aplicació AutoCAD 2002, aquesta aplicació treballa amb dos tipus d'arxius: DWG i DXF, però de cada arxiu hi ha múltiples versions disponibles.

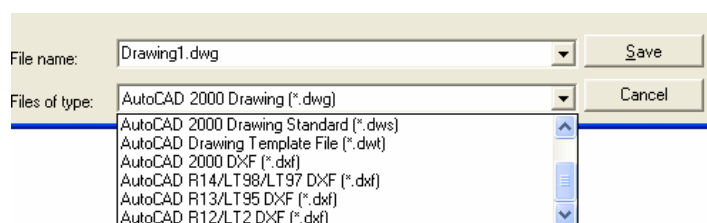


Figura 29.- Diferents formats de fitxer en AutoCAD 2002

D.3 .- ESTRUCTURA DELS FITXERS R12 DXF

L'estructura dels fitxers DXF, a diferència d'altres formats com per exemple el DWG, no conté informació comprimida, llavors és possible visualitzar-ne el contingut amb qualsevol editor de text que accepti la codificació en ASCII.

L'organització d'un fitxer DXF es divideix en quatre seccions: *HEADER* (capçalera), *TABLES* (taules), *BLOCKS* (blocs) i *ENTITIES* (entitats). Cada secció conté informació específica referent al dibuix.

SECCIÓ	CONTINGUTS
HEADER	Informació general de l'arxiu DXF.
TABLES	Defineix en taula: -Linetype table (LTYPE) -Layer table (LAYER) -Text style table (STYLE) -View table (VIEW) -User Coordinate System table (UCS) -Viewport configuration table (VPORT) -Dimension Style table (DIMSTYLE) -Application Identification table (APPID)
BLOCKS	Defineix grup d'entitats
ENTITIES	Describeix els elements dels dibuix: punts, línies, cercles, arcs, etc.
END OF FILE	Indica final d'arxiu.

Figura 30.- Organització en seccions, d'un fitxer DXF.

D.3.1 .- Secció HEADER

Aquesta secció correspon a l'encapsula't del format DXF. La informació d'aquesta secció és de caràcter general. Conté un número corresponent a la base de dades de l'AutoCAD i una sèrie de variables de sistema. Cadascun dels paràmetres conté una variable i els seu valor associat.

D.3.2 .- Secció TABLES

Aquesta secció conté les definicions de les taules de símbols:

APPID Taula de identificació de les aplicacions.
 BLOCK_RECORD Taula de referència de blocs.
 DIMSTYLE Taula d'estils d'acotació.
 LAYER Taula de capes.
 LTYPE Taula de tipus de línia.
 STYLE Taula d'estils de línia
 UCS Taula del sistema de coordenades personals.
 VIEW Taula de visualització.
 VPORT Taula configuració de la finestra gràfica.

D.3.3 .- Secció BLOCKS

Aquesta secció conté la definició dels objectes que componen els blocs usats en el dibuix, amb els seus noms y punts d'inserció.

D.3.4 .- Secció ENTITIES

Aquesta secció conté els objectes gràfics (entitats) del dibuix, inclòs les referències a blocs (entitats incrustades).

D.3.5 .- Marca de EOF

Marca que indica el final de fitxer.

D.4 .- CODIS DE GRUP

Cada secció agrupa un conjunt de valors, cadascun dels quals ocupa dues línies consecutives dins del fitxer, la primera és el codi de grup i la segona és el valor. El tipus de valor el determina el codi de grup. En la taula D.2 es mostra els diferents tipus de valors que hi ha en funció del número de codi de grup.

Interval de codis de grup	Tipus de valor
0 - 9	String
10 - 59	Floating-point
60 - 79	Integer

140 - 147	Floating-point
170 - 175	Integer
210 - 239	Floating-point
999	Comentari (string)
1000 - 1009	String
1010 - 1059	Floating-point
1060 - 1079	Integer

Figura 31.- Tipus de valor per intervals de codis de grup.

Existeixen múltiples codis de grup per a definir un valor en la següent línia de fitxer, aquest valor té un significat donat pel codi de grup. Per exemple, el codi de grup 999 representa un comentari, dins d'un fitxer DXF podem realitzar els comentaris de manera similar al següent fragment.

```
...
999
Això és un comentari.
999
Això és un altre comentari.
...
```

D.5 .- SECCIÓ ENTITIES

Aquesta és la secció més important del DXF, permet definir els objectes gràfics bàsics, és a dir, el punt, la línia, el text, el cercle i l'arc, entre d'altres. Depenent de la versió de DXF, la varietat d'entitats disponibles varia, per exemple la versió DXF R12 no admet segments ("splines"). La versió DXF R12 té un ventall suficient d'entitats per a la majoria de les necessitats. En aplicacions 3D per a elements finits resultaria limitat.

Per a comprendre millor els objectes gràfics elementals en DXF, es mostren exemples en els quals se'n descriuen els codis i valors que els defineixen.

D.6 .- ENTITAT POINT

Un punt és una entitat definida per tres coordenades: X, Y, Z.

Group codes	Description
100	Subclass marker (AcDbPoint)
10	Point location (in WCS) DXF: X value; APP: 3D point
20, 30	DXF: Y and Z values of point location (in WCS)
39	Thickness (optional; default = 0)
210	Extrusion direction (optional; default = 0, 0, 1)

	DXF: X value; APP: 3D vector
220, 230	DXF: Y and Z values of extrusion direction (optional)
50	Angle of the X axis for the UCS in effect when the point was drawn (optional, default = 0); used when PDMODE is nonzero

Figura 32.- Grup de codis per a definir un punt

Un exemple de punt:

0 Codi d'inici de la entitat.
POINT Valor de la entitat del punt.
8 Codi de la capa.
0 Número de capa.
10 Codi de la coordenada X.
300.0 Valor de la coordenada X.
20 Codi de coordenada Y.
100.0 Valor de la coordenada Y.
30 Codi de coordenada Z.
0.0 Valor de la coordenada Z.

D.7 .- ENTITAT LINE

Una línia és una entitat definida per dos punts, l'inicia'l i el final.

Group codes	Description
100	Subclass marker (AcDbLine)
39	Thickness (optional; default = 0)
10	Start point (in WCS) DXF: X value; APP: 3D point
20, 30	DXF: Y and Z values of start point (in WCS)
11	End point (in WCS) DXF: X value; APP: 3D point
21, 31	DXF: Y and Z values of end point (in WCS)
210	Extrusion direction (optional; default = 0, 0, 1) DXF: X value; APP: 3D vector
220, 230	DXF: Y and Z values of extrusion direction (optional)

Figura 33.- Grup de codis per a definir una línia

Un exemple de línia:

0 Codi d'inici de la entitat.
LINE Valor de la entitat.
8 Codi de la capa.
0 Número de capa.
10 Codi de la coordenada X del punt inicial.

100.0 Valor de coordenada X. 20 Codi de la coordenada Y del punt inicial. 100.0 Valor de coordenada Y. 30 Codi de la coordenada Z del punt inicial. 0.0 Valor de coordenada Z. 11 Codi de la coordenada X del punt final. 200.0 Valor de coordenada X. 21 Codi de la coordenada Y del punt final. 100.0 Valor de coordenada Y. 31 Codi de la coordenada Z del punt final. 0.0 Valor de coordenada Z.

D.8 .- ENTITAT POLYLINE

La polilínia és una seqüència de línies o de segments d'arc connectats.

100	Subclass marker (AcDb2dPolyline or AcDb3dPolyline)
10	DXF: always 0 APP: a "dummy" point; the X and Y values are always 0, and the Z value is the polyline's elevation (in OCS when 2D, WCS when 3D)
20	DXF: always 0
30	DXF: polyline's elevation (in OCS when 2D, WCS when 3D)
39	Thickness (optional; default = 0)
70	Polyline flag (bit-coded); default is 0: 1 = This is a closed polyline (or a polygon mesh closed in the M direction). 2 = Curve-fit vertices have been added. 4 = Spline-fit vertices have been added. 8 = This is a 3D polyline. 16 = This is a 3D polygon mesh. 32 = The polygon mesh is closed in the N direction. 64 = The polyline is a polyface mesh. 128 = The linetype pattern is generated continuously around the vertices of this polyline.
40	Default start width (optional; default = 0)
41	Default end width (optional; default = 0)
71	Polygon mesh M vertex count (optional; default = 0)
72	Polygon mesh N vertex count (optional; default = 0)
73	Smooth surface M density (optional; default = 0)
74	Smooth surface N density (optional; default = 0)
75	Curves and smooth surface type (optional; default = 0); integer codes, not bit-coded: 0 = No smooth surface fitted 5 = Quadratic B-spline surface 6 = Cubic B-spline surface 8 = Bezier surface

210	Extrusion direction (optional; default = 0, 0, 1) DXF: X value; APP: 3D vector
220, 230	DXF: Y and Z values of extrusion direction (optional)

Figura 34.- Grup de codis per a definir una polilínia.

D.9 .- ENTITAT CIRCLE

Un cercle es defineix per un centre i un radi.

Group codes	Description
100	Subclass marker (AcDbCircle)
39	Thickness (optional; default = 0)
10	Center point (in OCS) DXF: X value; APP: 3D point
20, 30	DXF: Y and Z values of center point (in OCS)
40	Radius
210	Extrusion direction (optional; default = 0, 0, 1)
220, 230	DXF: Y and Z values of extrusion direction (optional)

Figura 35.- Grup de codis per a definir un cercle.

D.10 .- ENTITAT ARC

Un arc és un tros de cercle, es defineix per un centre, un radi i dos angles.

Group codes	Description
100	Subclass marker (AcDbCircle)
39	Thickness (optional; default = 0)
10	Center point (in OCS) DXF: X value; APP: 3D point
20, 30	DXF: Y and Z values of center point (in OCS)
40	Radius
100	Subclass marker (AcDbArc)
50	Start angle
51	End angle
210	Extrusion direction. (optional; default = 0, 0, 1)
220, 230	DXF: Y and Z values of extrusion direction (optional)

Figura 36.- Grup de codis per a definir un arc.

D.11 .- ENTITAT TEXT

Group codes	Description
100	Subclass marker (AcDbText)
39	Thickness (optional; default = 0)
10	First alignment point (in OCS) DXF: X value; APP: 3D point
20, 30	DXF: Y and Z values of first alignment point (in OCS)
40	Text height
1	Default value (the string itself)
50	Text rotation (optional; default = 0)
41	Relative X scale factor-width (optional; default = 1) This value is also adjusted when fit-type text is used.
51	Oblique angle (optional; default = 0)
7	Text style name (optional, default = STANDARD)
71	Text generation flags (optional, default = 0): 2 = Text is backward (mirrored in X). 4 = Text is upside down (mirrored in Y).
72	Horizontal text justification type (optional, default = 0) integer codes (not bit-coded) 0 = Left; 1 = Center; 2 = Right 3 = Aligned (if vertical alignment = 0) 4 = Middle (if vertical alignment = 0) 5 = Fit (if vertical alignment = 0) See the Group 72 and 73 integer codes table for clarification.
11	Second alignment point (in OCS) (optional) DXF: X value; APP: 3D point This value is meaningful only if the value of a 72 or 73 group is nonzero (if the justification is anything other than baseline/left).
21, 31	DXF: Y and Z values of second alignment point (in OCS) (optional)
210	Extrusion direction (optional; default = 0, 0, 1) DXF: X value; APP: 3D vector
220, 230	DXF: Y and Z values of extrusion direction (optional)
100	Subclass marker (AcDbText)
73	Vertical text justification type (optional, default = 0): integer codes (not bit-coded): 0 = Baseline; 1 = Bottom; 2 = Middle; 3 = Top See the Group 72 and 73 integer codes table for clarification.

Figura 37.- Grup de codis per a definir un text.

Un exemple de text:

0 Codi d'inici de la entitat.
TEXT Valor de la entitat.
8 Codi de la capa.

0 Número de capa.
10 Codi de la coordenada X.
300.00 Valor de coordenada X.
20 Codi de la coordenada Y.
100.00 Valor de coordenada Y.
30 Codi de la coordenada Z.
0.0 Valor de coordenada Z.
40 Codi d'altura de text.
2.5 Valor de l'altura del text.
1 Codi del text.
Girona té color El text.

D.12 .- ESCRIURE/LLEGIR UN DXF

La lectura d'un DXF generat per una eina de dibuix és complicada, les eines de dibuix acostumen a posar-hi informació redundant que en dificulta l'anàlisi.

Autodesk exposa dos codis d'exemple de com llegir i escriure un fitxer DXF en llenguatge Basic.

D.13 .- LLEGIR UN FITXER DXF

Exemple de lectura d'un fitxer DXF.

```

1000 REM
1010 REM Extract lines from DXF file
1020 REM
1030 G1% = 0
1040 LINE INPUT "DXF file name: "; A$
1050 OPEN "i", 1, A$ + ".dxf"
1060 REM
1070 REM Ignore until section start encountered
1080 REM
1090 GOSUB 2000
1100 IF G% <> 0 THEN 1090
1110 IF S$ <> "SECTION" THEN 1090
1120 GOSUB 2000
1130 REM
1140 REM Skip unless ENTITIES section
1150 REM
1160 IF S$ <> "ENTITIES" THEN 1090
1170 REM
1180 REM Scan until end of section, processing LINEs
1190 REM
1200 GOSUB 2000
1210 IF G% = 0 AND S$ = "ENDSEC" THEN 2200
1220 IF G% = 0 AND S$ = "LINE" THEN GOSUB 1400 : GOTO 1210
  
```

```

1230 GOTO 1200
1400 REM
1410 REM Accumulate LINE entity groups
1420 REM
1430 GOSUB 2000
1440 IF G% = 10 THEN X1 = X : Y1 = Y : Z1 = Z
1450 IF G% = 11 THEN X2 = X : Y2 = Y : Z2 = Z
1460 IF G% = 0 THEN PRINT "Line from (";X1;" ";Y1;" ";Z1;) to
      (";X2;" ";Y2;" ";Z2;)" : RETURN
1470 GOTO 1430
2000 REM
2010 REM Read group code and following value
2020 REM For X coordinates, read Y and possibly Z also
2030 REM
2040 IF G1% < 0 THEN G% = -G1% : G1% = 0 ELSE INPUT #1, G%
2050 IF G% < 10 OR G% = 999 THEN LINE INPUT #1, S$ : RETURN
2060 IF G% >= 38 AND G% <= 49 THEN INPUT #1, V : RETURN
2080 IF G% >= 50 AND G% <= 59 THEN INPUT #1, A : RETURN
2090 IF G% >= 60 AND G% <= 69 THEN INPUT #1, P% : RETURN
2100 IF G% >= 70 AND G% <= 79 THEN INPUT #1, F% : RETURN
2110 IF G% >= 210 AND G% <= 219 THEN 2130
2115 IF G% >= 1000 THEN LINE INPUT #1, T$ : RETURN
2120 IF G% >= 20 THEN PRINT "Invalid group code";G% : STOP
2130 INPUT #1, X
2140 INPUT #1, G1%
2150 IF G1% <> (G%+10) THEN PRINT "Invalid Y coord code"; G1% :
      STOP
2160 INPUT #1, Y
2170 INPUT #1, G1%
2180 IF G1% <> (G%+20) THEN G1% = -G1% ELSE INPUT #1, Z
2190 RETURN
2200 CLOSE 1

```

D.14 .- ESCRIURE UN FITXER DXF

Exemple d'escriptura d'un fitxer DXF.

```

1000 REM
1010 REM Polygon generator
1020 REM
1030 LINE INPUT "Drawing (DXF) file name: "; A$
1040 OPEN "o", 1, A$ + ".dxf"
1050 PRINT #1, 0
1060 PRINT #1, "SECTION"
1070 PRINT #1, 2
1080 PRINT #1, "ENTITIES"
1090 PI = ATN(1) * 4

```

```
1100 INPUT "Number of sides for polygon: "; S%
1110 INPUT "Starting point (X,Y): "; X, Y
1120 INPUT "Polygon side: "; D
1130 A1 = (2 * PI) / S%
1140 A = PI / 2
1150 FOR I% = 1 TO S%
1160 PRINT #1, 0
1170 PRINT #1, "LINE"
1180 PRINT #1, 8
1190 PRINT #1, "0"
1200 PRINT #1, 10
1210 PRINT #1, X
1220 PRINT #1, 20
1230 PRINT #1, Y
1240 PRINT #1, 30
1250 PRINT #1, 0.0
1260 NX = D * COS(A) + X
1270 NY = D * SIN(A) + Y
1280 PRINT #1, 11
1290 PRINT #1, NX
1300 PRINT #1, 21
1310 PRINT #1, NY
1320 PRINT #1, 31
1330 PRINT #1, 0.0
1340 X = NX
1350 Y = NY
1360 A = A + A1
1370 NEXT I%
1380 PRINT #1, 0
1390 PRINT #1, "ENDSEC"
1400 PRINT #1, 0
1410 PRINT #1, "EOF"
1420 CLOSE 1
```

ANNEX E .- DIAGRAMES

E.1 .-	INTRODUCCIÓ.....	44
E.2 .-	FITXES DE CASOS D'ÚS	44
E.2.1 .-	GESTIONAR PROJECTES	44
E.2.2 .-	GESTIONAR FIGURES	47
E.2.3 .-	GESTIONAR UN ENTORN DE TREBALL	49
E.3 .-	DIAGRAMA D'ESTATS.....	52
E.4 .-	DIAGRAMES DE CLASSES.....	53
E.4.1 .-	DIAGRAMA DE CLASSES DE LA LLIBRERIA.....	53
E.4.2 .-	DIAGRAMA DE CLASSES DE L'APLICACIÓ BDDDIAGRAMMER	54
E.5 .-	CODI DEL PARSER XML	55

E.1 .-INTRODUCCIÓ

En aquest annex s'agrupen els diferents diagrames UML i taules desenvolupades durant el projecte.

S'ha realitzat els diagrames de casos d'ús i les corresponents fitxes de casos d'ús. En la memòria s'ha inclòs les tres fitxes de casos d'ús més destacades.

El següent diagrama que es pot observar és el diagrama d'estats de les operacions que permet la llibreria. En aquest diagrama s'inclouen totes les operacions que es realitzen mitjançant aquesta màquina d'estats implementada.

Posteriorment es troben els diagrames de classes, el de la llibreria i el de l'aplicació.

Per últim s'ha inclòs el codi desenvolupat per al parser XML. En la memòria es pot observar la gramàtica que s'ha desenvolupat per a reconèixer un fitxer XML. Aquest codi ha estat desenvolupat mitjançant l'editor ANTLWorks per a l'ANTLR 3.

E.2 .-FITXES DE CASOS D'ÚS

En aquest annex s'ha posat aquelles fitxes de casos d'ús que s'ha considerat més comunes o no tan destacades. Aquestes s'han separat en tres grups: Gestionar projectes, Gestionar figures i Gestionar un entorn de treball.

E.2.1 .- Gestionar projectes

Nom del cas d'ús	CARREGAR		
Autor	Cristina Constantins	Data	09-feb-08
Descripció	Carregar l'estructura de figures guardat anteriorment		
Actors	Usuari		
Precondició	Disposar d'un projecte guardat		
Flux normal	1 .- L'usuari indica que vol carregar un projecte guardat 2 .- Es demana el path del fitxer que conté el projecte 3 .- S'obre el fitxer i s'analitza 4 .- Es creen els objectes que descriu el fitxer 5 .- Es tanca el fitxer		
Flux alternatiu	En cas que l'usuari indiqui un path que no sigui vàlid, no es realitzarà cap acció En cas que el fitxer sigui corrupte, no es realitzarà cap acció En cas que en demanar el path es cancel·li, no s'obrirà el projecte		
Poscondició	S'ha obert un projecte que s'havia guardat anteriorment		

Nom del cas d'ús	GUARDAR		
Autor	Cristina Constantins	Data	09-feb-08
Descripció	Guardar les figures i l'entorn de manera que es pugui carregar posteriorment		
Actors	Usuari		
Precondició	Disposar d'un entorn de treball		
Flux normal	6.- L'usuari indica que vol guardar el projecte 7.- Es demana a l'usuari que indiqui on vol guardar el fitxer 8.- S'estructura el fitxer on es guardaran les dades 9.- Es guarden les propietats de l'entorn de treball 10.- Cada un dels elements guarda les seves propietats al fitxer 11.- El sistema guarda el fitxer a disc		
Flux alternatiu	En cas que l'usuari indiqui un path que no sigui vàlid, no es crearà ni es guardarà el fitxer En cas que en demanar el path es cancel·li, no es guardarà el projecte		
Poscondició	S'han guardat a disc les propietats de la llibreria i dels objectes creats		

Nom del cas d'ús	EXPORTAR A DXF		
Autor	Cristina Constantins	Data	09-feb-08
Descripció	Guardar els gràfics de la llibreria en format DXF		
Actors	Usuari		
Precondició	L'entorn de treball ha d'estar instanciat		
Flux normal	1.- L'usuari indica que vol exportar a format DXF 2.- El sistema demana a l'usuari en quin path vol guardar el fitxer resultant 3.- El sistema guarda l'entorn de treball i demana a cada objecte que també es dibuixi 4.- El sistema guarda el fitxer a disc		
Flux alternatiu	En cas que l'usuari no indiqui un path vàlid, el fitxer no es crearà ni guardarà En cas que en demanar el path es cancel·li la operació, no es guardarà el projecte En cas que el path donat correspongui a un fitxer existent, es demanarà a l'usuari si vol sobre escriure'l		
Poscondició	La representació gràfica de la vista es guarda a disc, en un fitxer en format DXF		

Nom del cas d'ús	EXPORTAR A XML		
Autor	Cristina Constantins	Data	09-feb-08
Descripció	Guardar les propietats de les figures en format XML		
Actors	Usuari		
Precondició	L'entorn de treball ha d'estar instanciat		
Flux normal	1.- L'usuari indica que vol exportar a XML 2.- El sistema demana a l'usuari en quin path vol crear el fitxer resultant 3.- El sistema guarda les propietats de l'entorn de treball i seguidament demana a cada objecte que també guardi les seves propietats 4.- El sistema guarda el fitxer a disc		
Flux alternatiu	En cas que l'usuari no indiqui un path vàlid, el fitxer no es crearà ni guardarà En cas que en demanar el path es cancel·li la operació, no es guardarà el projecte En cas que el path donat correspongui a un fitxer existent, es demanarà a l'usuari si vol sobre escriure'l		
Poscondició	Es guarda a disc un fitxer que conté una descripció estructurada de les propietats de les figures creades amb la llibreria.		

Nom del cas d'ús	EXPORTAR A UN FORMAT GRÀFIC		
Autor	Cristina Constantins	Data	09-feb-08
Descripció	Guardar les figures de la llibreria en un format gràfic		
Actors	Usuari		
Precondició	L'entorn de treball ha d'estar instanciat		
Flux normal	1.- L'usuari indica que vol exportar a un format gràfic 2.- El sistema demana en quin path crear el fitxer 3.- Es guarda l'entorn de treball i es demana a cada objecte que també es dibuixi 4.- El sistema guarda el fitxer a disc		
Flux alternatiu	En cas que l'usuari no indiqui un path vàlid, el fitxer no es crearà ni guardarà En cas que en demanar el path es cancel·li, no es guardarà el projecte En cas que el path donat correspongui a un fitxer existent, es demanarà a l'usuari si vol sobre escriure'l		
Poscondició	Es guarda a disc un fitxer gràfic que conté una representació de les figures creades amb la llibreria.		

E.2.2 .- Gestionar figures

Nom del cas d'ús	CREAR UNA FIGURA		
Autor	Cristina Constantins	Data	09-feb-08
Descripció	Afegir una nova figura a l'escenari		
Actors	Usuari		
Precondició	L'entorn de treball ha d'estar instanciat		
Flux normal	1.- L'usuari indica el tipus de figura que vol crear d'entre els tipus de figures possibles 2.- El sistema instancia una nova figura 3.- S'afegeix la nova figura en la llista de figures instanciades 4.- L'usuari indica on vol situar la figura mitjançant el ratolí 5.- El sistema assigna a la figura, la posició definida per l'usuari 6.- Es repinten totes les vistes del sistema		
Flux alternatiu	En cas que es realitzi una acció no prevista es cancel·larà la operació		
Poscondició	S'ha creat una nova figura		

Nom del cas d'ús	ELIMINAR UNA FIGURA		
Autor	Cristina Constantins	Data	09-feb-08
Descripció	Eliminar un objecte creat		
Actors	Usuari		
Precondició	L'entorn de treball ha d'estar instanciat		
Flux normal	1.- L'usuari indica que vol eliminar els objectes seleccionats 2.- Es desseleccionen els objectes a eliminar i es treuen de la llista d'objectes instanciats 3.- Es repinten totes les vistes del sistema		
Flux alternatiu	En cas que no hi hagi cap element seleccionat, no s'elimina res		
Poscondició	S'han eliminat els objectes que s'havien seleccionat		

Nom del cas d'ús	COPIAR I ENGANXAR		
Autor	Cristina Constantins	Data	09-feb-08
Descripció	Es dupliquen els objectes seleccionats		
Actors	Usuari		
Precondició	S'ha d'haver creat almenys un objecte visual		
Flux normal	1.- L'usuari indica que vol copiar els objectes seleccionats 2.- El sistema crea còpies de cada un dels objectes a seleccionats 3.- L'usuari indica que vol enganxar els elements copiats 4.- S'afegeixen els nous objectes a la llista d'objectes instanciats 5.- Es repinten totes les vistes del sistema		
Flux alternatiu	En cas que no s'hagin seleccionat elements a copiar, no es realitza cap operació		
Poscondició			

Nom del cas d'ús	SELECCIONAR FIGURES		
Autor	Cristina Constantins	Data	09-feb-08
Descripció	Seleccionar un o més elements		
Actors	Usuari		
Precondició	És necessari haver creat almenys un element		
Flux normal	1.- L'usuari dibuixa un rectangle amb el ratolí en la zona de treball 2.- Per cada element creat es marquen com a seleccionats tots els elements que estiguin completament dins el rectangle		
Flux alternatiu	En cas que el rectangle no inclogui cap element, no es selecciona res En cas que enlloc de dibuixar un rectangle, sols es dibuixi un punt, es selecciona l'element que hi hagi sota el ratolí. Si no hi ha cap element no es selecciona res		
Poscondició	Un o més elements s'han marcat com a seleccionats		

Nom del cas d'ús	MOURE FIGURES		
Autor	Cristina Constantins	Data	09-feb-08
Descripció	Es desplacen els objectes seleccionats		
Actors	Usuari		
Precondició	És necessari que s'hagi creat almenys un element		
Flux normal	1.- L'usuari prem el botó del ratolí sobre un dels elements seleccionats i l'arrossega fins a un altre punt dins l'entorn 2.- El sistema modifica la posició dels elements seleccionats segons la distancia indicada amb el ratolí		
Flux alternatiu	En cas que no s'hagi seleccionat cap element, no es fa res		
Poscondició	Els elements han canviat de posició		

Nom del cas d'ús	AGRUPAR/DESAGRUPAR FIGURES		
Autor	Cristina Constantins	Data	09-feb-08
Descripció	Crear un nou objecte a partir de diversos elements bàsics		
Actors	Usuari		
Precondició	És necessari haver creat com a mínim un element bàsic		
Flux normal	1.- L'usuari selecciona els elements que vol agrupar 2.- Llavors indica que vol agrupar els elements seleccionats 3.- El sistema crea un objecte compost i li indica quins elements el formen 4.- Es visualitza el nou objecte com un sol element		
Flux alternatiu	En cas que no s'hagi seleccionat cap a agrupar, no es farà res En cas que sols es seleccioni un element a agrupar, els nou element compost sol tindrà un element simple		
Poscondició	S'obté un nou element fruit de la unió de diversos elements		

Nom del cas d'ús	MODIFICAR PROPIETATS D'UNA FIGURA		
Autor	Cristina Constantins	Data	09-feb-08
Descripció	Canviar les propietats d'un dels objectes		
Actors	Usuari		
Precondició	És necessari que s'hagi creat almenys un objecte		
Flux normal	1.- L'usuari indica que vol visualitzar les propietats de l'element seleccionat, ja sigui per menús o mitjançant un doble click sobre l'element 2.- El sistema mostra un formulari amb les propietats de l'element 3.- L'usuari modifica les propietats que desitja canviar 4.- El sistema actualitza les propietats de l'element 5.- Es mostren els canvis necessaris en la visualització dels elements		
Flux alternatiu	En cas que no s'hagi seleccionat cap element, no es poden visualitzar les seves propietats En cas d'haver seleccionat diversos elements, tampoc es poden visualitzar les seves propietats		
Poscondició	L'element seleccionat s'ha actualitzat d'acord amb les indicacions de l'usuari		

E.2.3 .- Gestionar un entorn de treball

Nom del cas d'ús	MODIFICAR ZOOM DE LA VISTA		
Autor	Cristina Constantins	Data	09-feb-08
Descripció	Canvia el zoom de l'entorn de treball		
Actors	Usuari		
Precondició	L'entorn de treball ha d'estar instanciat		
Flux normal	6.- L'usuari canvia indica que vol canviar el zoom mitjançant els botons de zoom o bé amb el teclat i el ratolí 7.- El sistema canvia el zoom de la vista actual 8.- Es repinten els objectes per a visualitzar-los amb el nou zoom		
Flux alternatiu	En afegir nous objectes aquests ja es visualitzen amb el nou zoom		
Poscondició	S'observen els objectes amb el nou zoom		
Comentaris	El zoom és particular de cada vista.		

Nom del cas d'ús	MODIFICAR SNAP DE LA VISTA		
Autor	Cristina Constantins	Data	09-feb-08
Descripció	Aplica o treu l'snap de l'entorn de treball		
Actors	Usuari		
Precondició	L'entorn de treball ha d'estar instanciat		
Flux normal	1.- L'usuari indica que vol aplicar o treure l'snap R 1.1) El sistema actualitza la vista indicant que s'utilitza o no l'snap 2.- L'usuari indica que vol canviar la dimensió de l'snap R 1.2) El sistema actualitza la vista indicant les noves dimensions de l'snap 3.- Es repinta la vista per actualitzar es canvis realitzats		
Flux alternatiu	En afegir nous objectes aquests ja es visualitzen amb el nou zoom		
Poscondició	S'observen els objectes amb el nou zoom		
Comentaris	El zoom és particular de cada vista.		

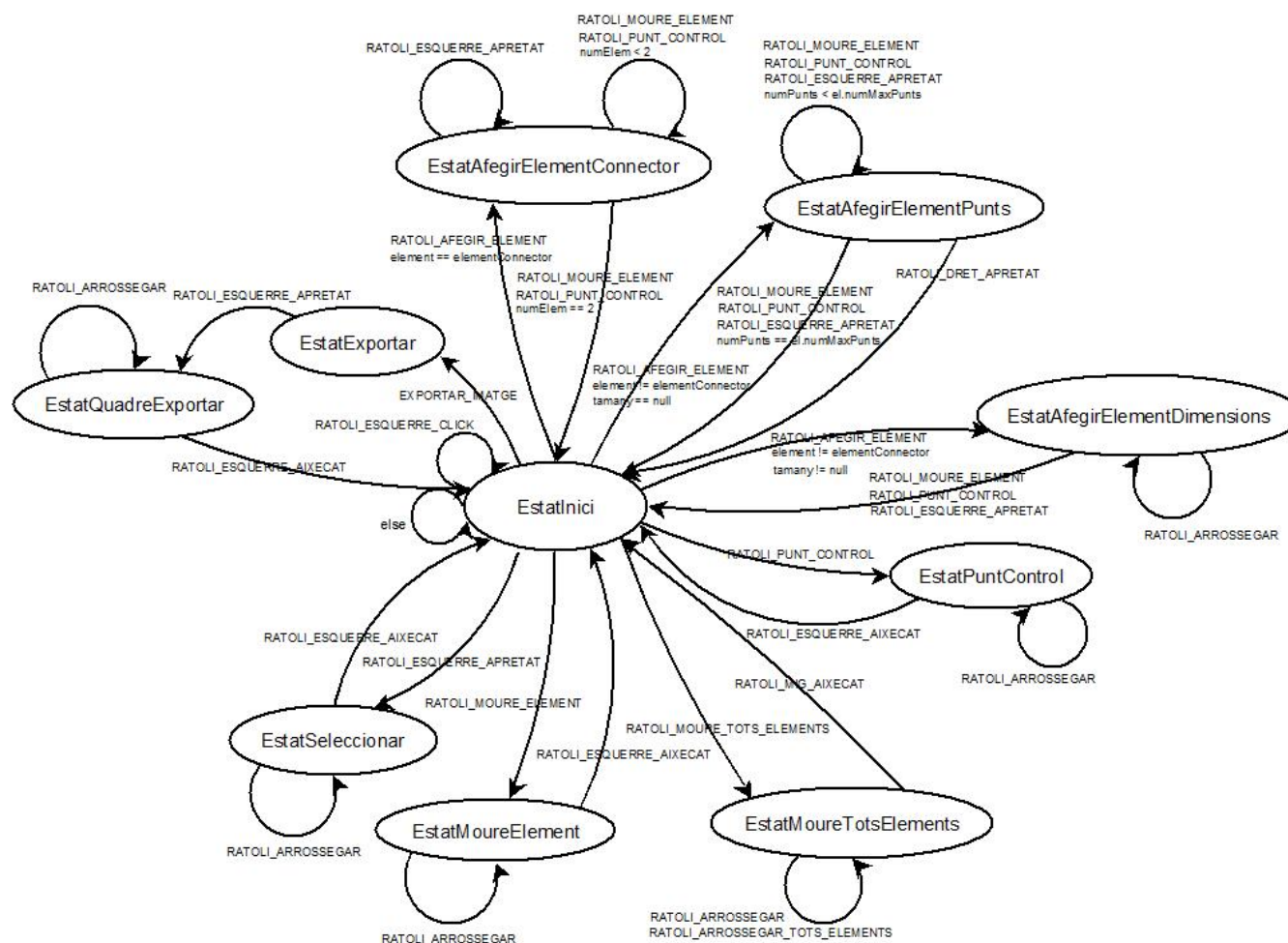
Nom del cas d'ús	GESTIONAR CAPES		
Autor	Cristina Constantins	Data	09-feb-08
Descripció	Administrar les capes		
Actors	Usuari		
Precondició	L'entorn de treball ha d'estar instanciat		
Flux normal	1.- L'usuari indica que vol crear una nova capa 1.1.- El sistema mostra el formulari de les propietats de la capa 1.2.- L'usuari escriu les dades pertinents 1.3.- El sistema guarda les dades i crea la nova capa 2.- L'usuari decideix modificar una capa 2.1.- El sistema mostra el formulari de les propietats de la capa 2.2.- L'usuari canvia la informació que vol 2.3.- El sistema guarda els valors i actualitza la capa segons la informació 3.- L'usuari decideix eliminar una capa 3.1.- El sistema demana si també es volen eliminar els elements que formen part de la capa 3.1.1.- En cas que l'usuari decideixi eliminar-los, s'eliminen els elements un a un, tal com s'ha indicat en la fitxa de casos d'ús "ELIMINAR UNA FIGURA" 3.1.2.- En cas que l'usuari no vulgui eliminar els elements, aquests passen a formar part de la capa base 3.2.- S'elimina la capa indicada		
Flux alternatiu	En cas que es vulgui eliminar la primera capa (la que crea la llibreria), s'informa a l'usuari que no és una capa vàlida per eliminar		
Poscondició			
Comentaris	En crear un projecte amb la llibreria es crea una capa per defecte		

Nom del cas d'ús	MODIFICAR GRID DE LA VISTA		
Autor	Cristina Constantins	Data	09-feb-08
Descripció	Canvia el grid de l'entorn de treball		
Actors	Usuari		
Precondició	L'entorn de treball ha d'estar instanciat		
Flux normal	1.- L'usuari canvia indica que vol canviar el tipus de grid 2.- El sistema canvia el tipus de grid de la vista actual 3.- Es repinta la vista per actualitzar la visualització del grid		
Flux alternatiu			
Poscondició	S'observa el grid escollit darrera els elements creats		
Comentaris	El tipus de grid és particular de cada vista.		

Nom del cas d'ús	MODIFICAR PROPIETATS		
Autor	Cristina Constantins	Data	09-feb-08
Descripció	Canviar les propietats de l'entorn de treball		
Actors	Usuari		
Precondició	L'entorn de treball ha d'estar instanciat		
Flux normal	1.- L'usuari indica que vol canviar les propietats de l'entorn 2.- El sistema mostra un formulari amb les propietats 3.- L'usuari modifica les propietats que vol 4.- El sistema guarda els nous valors 5.- S'actualitza l'entorn de treball, d'acord amb les noves propietats		
Flux alternatiu			
Poscondició	S'ha modificat l'entorn de treball segons les indicacions de l'usuari		

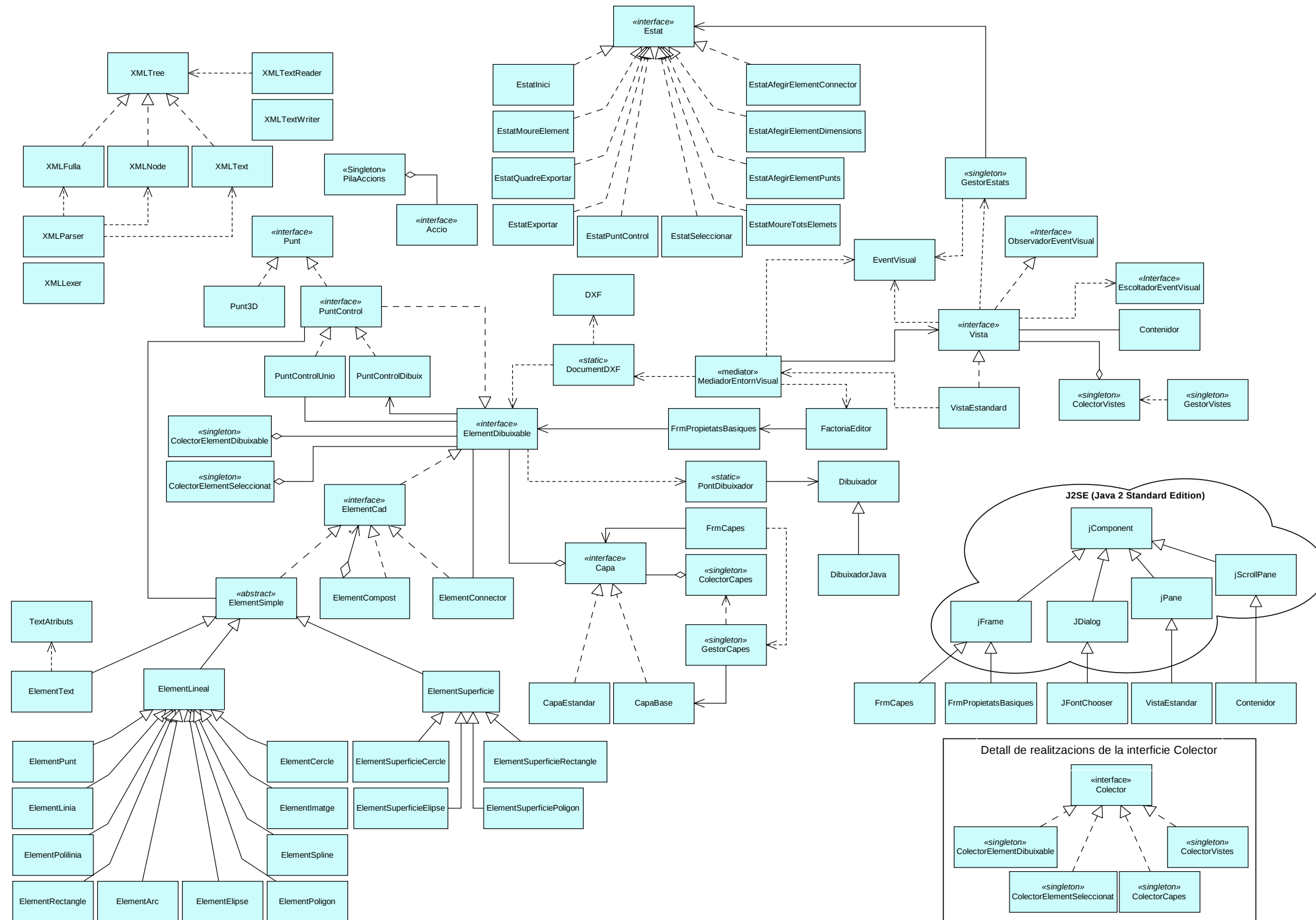
Nom del cas d'ús	NOTIFICAR EVENTS		
Autor	Cristina Constantins	Data	09-feb-08
Descripció	La llibreria informa sobre els events enviats al sistema o succeïts a la capa superior		
Actors	Usuari		
Precondició	L'entorn de treball ha d'estar instanciat		
Flux normal	1.- L'usuari realitza una operació i es produeix un event 2.- El sistema envia l'event al sistema que gestiona els events, per a resoldre la operació 3.- Si l'event és un dels esperats, es genera un canvi d'estat en el sistema d'events de manera que es realitza un pas en l'execució de l'operació		
Flux alternatiu	Els events també es poden enviar des de l'exterior, els quals es tracten igual que si fossin rebuts directament en la capa superior En cas que l'event rebut no sigui un dels esperats es pot desestimar l'event o, fins i tot, cancel·lar l'operació en curs		
Poscondició	La capa superior està informada dels events que es produeixen		

E.3 .-DIAGRAMA D'ESTATS

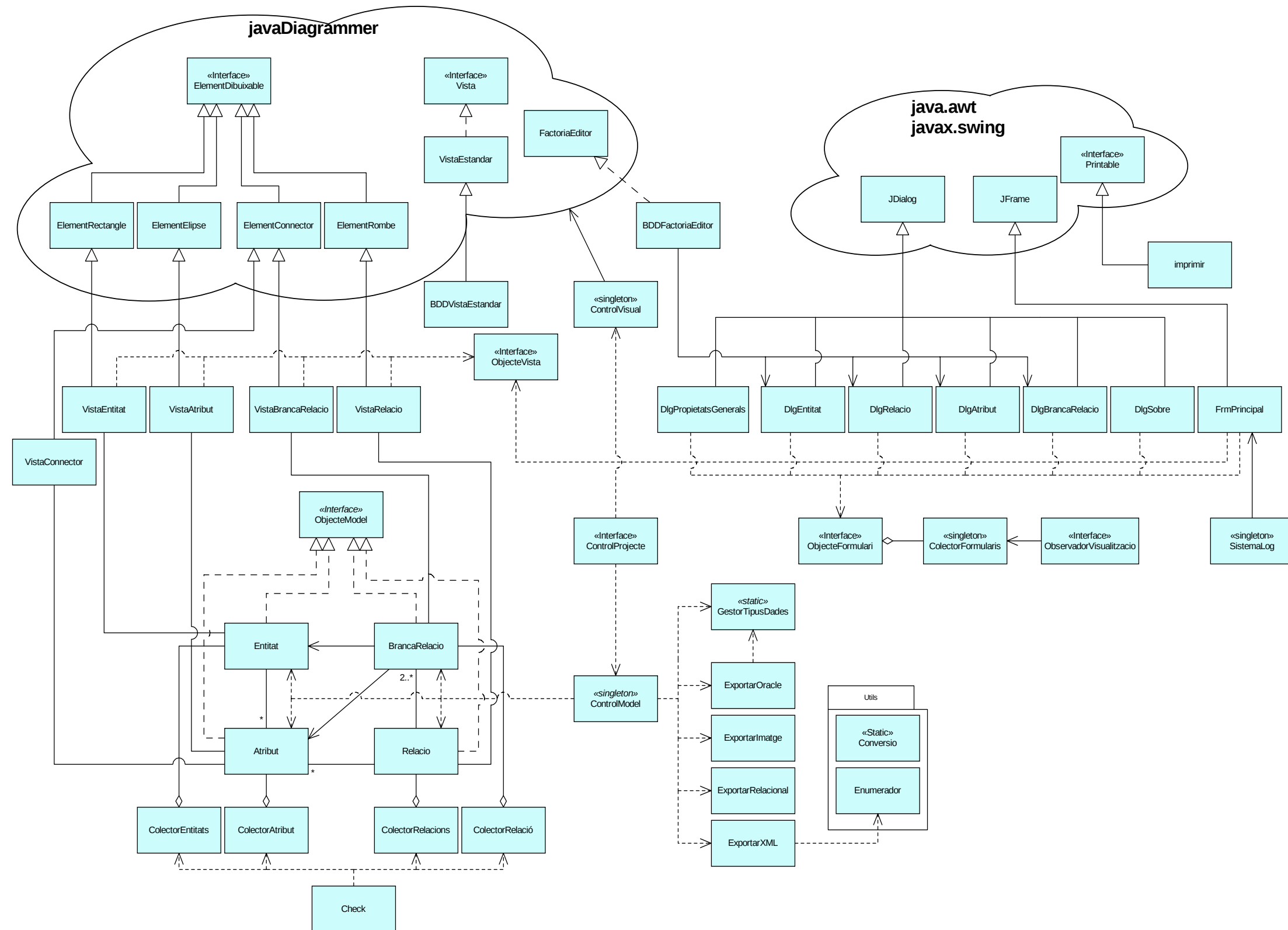


E.4 .-DIAGRAMES DE CLASSES

E.4.1 .- Diagrama de classes de la llibreria



E.4.2 .- Diagrama de classes de l'aplicació BDDDiagrammer



E.5 .-CODI DEL PARSER XML

En el següent requadre es pot observar el codi per a generar el parser XML de la llibreria. En la memòria s'ha descrit la gramàtica que reconeix aquest codi. Per a compilar el codi és necessari l'ANTLR v3.

```
grammar XML;

@header {
package view.guardar;

import java.util.Hashtable;
}

@lexer::header {
package view.guardar;
}

@lexer::members {
//variables global lexic
    boolean tagMode = false;
}

/*
 * Sintàctic/Atributs
 */

document returns [ArrayList nodes]
    :
        a=xmltree EOF
        { nodes = a; }
    ;

xmltree returns [ArrayList nodes]
    :
        { nodes = new ArrayList(); }
        (( a=node { nodes.add((Object)a); })+
        | b=fulla { nodes.add((Object)b); })
    ;

node returns [XMLNode node]
    :
        { node = new XMLNode(); }
        a=tag_ini b=atribut TK_mes_gran {System.out.println("Node
trobat");} c=xmltree d=tag_fi
        {
            if((a.trim()).equals(d.trim())){
```

```

        node.assignarEtiqueta(a.trim());
        node.assignarNodes(c);
        node.afegirAtributs(b);
    }else{
        System.out.println("ERROR semàntic: etiquetes no
coincidents.");
    }
}
;

fulla returns [XMLFulla fulla]
:
{ System.out.println("fulla trobada");
  String texte = "";
  fulla = new XMLFulla();
  a=TK_text {texte = a.getText();
    texte.trim();
    fulla.assignarText(texte);}
;

tag_ini returns [String etiqueta]
:
    TK_mes_petit a=TK_ident { etiqueta = a.getText();}
;

tag_fi returns [String etiqueta]
:
    TK_tancar a=TK_ident { etiqueta = a.getText(); } TK_mes_gran
;

atribut returns [Hashtable atrib]
:
    {atrib = new Hashtable();}
    (a=TK_ident TK_igual TK_cometes b=TK_text TK_cometes
      {atrib.put(a.getText(), b.getText());})
)*
;

/*
 * Lèxic
 */
TK_mes_gran : '>' { tagMode = false; }
;

TK_tancar : '</' { tagMode = true; }

```

```
    ;

TK_mes_petit : '<' { tagMode = true; }
    ;

TK_cometes : ""
    ;

TK_igual : '='
    ;

TK_ident : { tagMode }?=> (LLETRA|'_') (LLETRA|DIGIT|'_'|'.')*
    ;

TK_text : { !tagMode }?=> (~('<'>'|''))*
    ;

fragment DIGIT : '0'..'9'
    ;

fragment LLETRA : 'a'..'z'
    | 'A'..'Z'
    ;

WS : { tagMode }?=>
    ( ' '
    | '\t'
    | '\r'
    | '\n')
    { skip();}
    ;
```