

# Índex

<b>1</b>	<b>Introducció, motivacions i objectius</b>	<b>4</b>
1.1	Introducció . . . . .	4
1.2	Motivacions . . . . .	5
1.3	Objectius . . . . .	5
<b>2</b>	<b>Estudi de viabilitat</b>	<b>6</b>
<b>3</b>	<b>Metodologia</b>	<b>7</b>
<b>4</b>	<b>Planificació</b>	<b>8</b>
<b>5</b>	<b>Marc de treball i conceptes previs</b>	<b>9</b>
5.1	Anàlisi de la plataforma ACME . . . . .	9
5.2	Arquitectura Client-Servidor . . . . .	9
5.3	Eines utilitzades . . . . .	11
5.3.1	Llenguatges de programació del costat del client . . . . .	11
5.3.1.1	HTML . . . . .	11
5.3.1.2	CSS . . . . .	12
5.3.1.3	Javascript . . . . .	13
5.3.2	Llenguatges de programació del costat del servidor . . . . .	13
5.3.2.1	PHP . . . . .	13
5.3.3	Frameworks de programació en Javascript . . . . .	14
5.3.3.1	JQuery . . . . .	17
5.3.3.2	JQueryUI . . . . .	18
5.3.4	Altres llenguatges utilitzats . . . . .	19
5.3.4.1	LaTeX . . . . .	19
5.3.4.2	Mathematica . . . . .	20
5.3.5	Entorn de desenvolupament . . . . .	21
5.3.5.1	Navegadors web . . . . .	21
5.3.5.2	SSH Secure Shell . . . . .	21
5.3.5.3	Notepad++ . . . . .	22
<b>6</b>	<b>Metodologia d'escriptura d'exercicis tradicional</b>	<b>23</b>
6.1	Anàlisi de l'escriptura d'un exercici . . . . .	23
6.1.1	Capçalera del fitxer . . . . .	24
6.1.2	Enunciat dels exercicis . . . . .	24
6.1.3	Definició dels paràmetres de l'exercici . . . . .	26
6.1.4	Codi Mathematica de resolució . . . . .	28
6.2	Exercici d'exemple . . . . .	29
6.3	Consideracions a la metodologia tradicional . . . . .	33
<b>7</b>	<b>Nova metodologia d'escriptura d'exercicis</b>	<b>34</b>
7.1	Estructura d'un exercici . . . . .	36
7.1.1	Tecnologia utilitzada . . . . .	36
7.1.2	Fitxer definició d'un exercici . . . . .	38
7.1.3	Altres fitxers definició . . . . .	43

7.1.4	Estructura d'arbre del fitxer definició . . . . .	43
7.2	Definició del model . . . . .	45
7.2.1	Preguntes . . . . .	46
7.2.1.1	Enunciats . . . . .	46
7.2.1.2	Fitxers . . . . .	48
7.2.1.3	Solucions . . . . .	48
7.2.2	Paràmetres . . . . .	51
7.2.2.1	Llistat de valors . . . . .	51
7.2.2.2	Paràmetre calculat . . . . .	52
7.3	Editor d'exercicis . . . . .	53
7.3.1	Editor antic . . . . .	53
7.3.2	Nou editor d'exercicis . . . . .	59
7.3.2.1	Pantalla inicial . . . . .	59
7.3.2.2	Visualització i correcció . . . . .	61
7.3.2.3	Preguntes . . . . .	61
7.3.2.4	Enunciats . . . . .	62
7.3.2.5	Enunciat general . . . . .	62
7.3.2.6	Fitxers . . . . .	63
7.3.2.7	Solucions . . . . .	63
7.3.2.8	Paràmetres . . . . .	66
7.3.2.9	Enviar dades . . . . .	69
7.3.2.10	Missatges d'error . . . . .	70
7.3.2.11	Baixar el fitxer XML . . . . .	70
7.4	Implantació del nou tipus d'exercici a l'ACME . . . . .	71
7.4.1	Procés de reconeixement del nou tipus de fitxer XML . . . . .	71
7.4.2	Procés de sorteig d'exercicis . . . . .	72
7.4.3	Visualització dels enunciats dels exercicis . . . . .	77
7.4.4	Procés de correcció dels exercicis . . . . .	79
<b>8</b>	<b>Implantació i resultats</b>	<b>81</b>
<b>9</b>	<b>Conclusions</b>	<b>82</b>
<b>10</b>	<b>Ampliacions i millores</b>	<b>83</b>
<b>11</b>	<b>Bibliografia</b>	<b>84</b>
<b>12</b>	<b>Índex de figures</b>	<b>85</b>
<b>13</b>	<b>Agraïments</b>	<b>87</b>

# 1 Introducció, motivacions i objectius

## 1.1 Introducció

L'ACME és una plataforma d'e-learning utilitzada a la Universitat de Girona que permet la correcció automàtica d'un gran nombre d'exercicis de tipologies variades via web.

La principal característica és el seu alt contingut tècnic ja que permet corregir de forma automàtica exercicis matemàtics, informàtics, de base de dades, tipus test, etc. Actualment s'utilitza com a eina de seguiment del treball diari dels alumnes en moltes carreres i facultats de la Universitat de Girona, que els permet consolidar la matèria explicada a classe.

L'ACME a més a més de corregir exercicis de matemàtiques, també pot corregir exercicis amb continguts matemàtics però que són d'altres assignatures com poden ser Física, Química, Estadística, Economia, etc. L'ampliació a aquestes altres assignatures ha estat gràcies a que en els darrers anys s'han anat realitzant diferents Projectes Finals de Carrera en els quals s'han desenvolupat nous correctors per la plataforma, arribant a assolir la xifra de 50 tipologies d'exercicis diferents.

Aquestes 50 tipologies d'exercicis diferents s'han caracteritzat sempre per fer molt fàcil als alumnes tant l'escriptura de les solucions com saber el resultat de la seva correcció. Però un tema que no s'ha pres tanta cura, i que s'ha de millorar, ha estat la sintaxi necessària per escriure aquests exercicis, i per tant no s'ha pensat en facilitar la feina d'escriptura dels exercicis als professors. Cada projectista va anar ampliant, modificant o adaptant la sintaxi al seu gust alhora de fer una nova tipologia d'exercici, fent difícil l'aprenentatge de totes elles i dificultant les tasques d'edició i preparació d'exercicis.

Actualment tots els fitxers que contenen els exercicis de l'ACME estan en format  $\text{\LaTeX}$ , o similar, i com hem dit segueixen uns formats d'escriptura concrets, no estàndards i molts definits per a la seva posterior lectura i interpretació. Aquesta alta especificitat i manca d'estandardització fa difícil l'aprenentatge de l'escriptura per part dels professors que volen fer servir l'ACME. Aquesta dificultat fa que un gran nombre de professors es limitin a escriure exercicis nous a partir d'exercicis antics, desconeixent per complet si amb la tipologia d'exercici que utilitzen estan fent servir tot el potencial, o no, que permet l'eina.

Arribat aquest punt es va veure la necessitat de facilitar i estandarditzar la tasca d'edició i escriptura dels exercicis. És a partir d'aquí que neix aquest Projecte Final de Carrera amb la intenció de desenvolupar una eina que permeti facilitar, millorar i ampliar la metodologia d'escriptura de nous exercicis a l'ACME.

Per tal d'assolir els objectius es definirà i implementarà una nova metodologia d'escriptura d'exercicis ACME. Aquesta nova tipologia d'exercicis serà totalment transversal, possibilitant escriure tots els exercicis amb el mateix format, i cobrirà les mancances del conjunt d'exercicis actuals.

Aquesta nova metodologia d'escriptura d'exercicis serà incorporada a la plataforma sota l'aprovació del responsable de l'ACME. Juntament amb els professors es provarà i testejarà aquesta nova metodologia d'escriptura. La comunicació entre professor i projectista permetrà veure les diferents sensacions dels professors envers la nova metodologia, es corregiran possibles errors que vagin apareixent i també es considerarà i estudiarà qualsevol tipus de petició per part dels professors amb l'objectiu d'introduir millores que siguin útils de cares a que l'eina sigui el més pràctica possible.

## 1.2 Motivacions

Durant tres anys, des dels inicis del tercer curs de la carrera d'Enginyeria Tècnica en Informàtica de Gestió, he estat desenvolupant tasques de becari a l'ACME. Els continguts tècnics de les pàgines web m'interessaven molt i tenia la voluntat d'aprendre'n i va ser per aquests motius pels quals vaig decidir realitzar tant aquest Projecte de Final de Carrera com el de la carrera Tècnica amb el Departament d'Informàtica i Matemàtica Aplicada i concretament amb la plataforma ACME.

## 1.3 Objectius

Els principals objectius d'aquest PFC són la creació d'una nova metodologia per escriure els exercicis de l'ACME i, una vegada definida la metodologia, es farà l'anàlisi, el disseny i la implementació de les modificacions necessàries per tal d'incorporar aquesta metodologia dins la plataforma ACME. El sistema que es desenvoluparà ha de ser compatible amb l'existent i ha de facilitar i millorar el sistema d'escriptura actual de l'ACME.

Per tal de dur a terme aquest Projecte Final de Carrera caldrà:

- Realitzar un estudi del funcionament de l'escriptura dels exercicis actuals a l'ACME.
- Definir i especificar l'estructura per emmagatzemar la nova tipologia d'exercicis.
- Dissenyar i desenvolupar les funcionalitats de la nova tipologia d'exercicis.
- Dissenyar i desenvolupar la nova eina d'escriptura d'exercicis basada en fitxers XML.
- Integrar la nova tipologia d'exercicis dins l'ACME.

## 2 Estudi de viabilitat

L'eina que es pretén desenvolupar serà incorporada a posteriori a l'ACME, per tant, s'ha d'adoptar el mateix tipus de tecnologia que es fa servir a la plataforma, és a dir, tecnologia orientada a web. També s'haurà de disposar del hardware necessari per desenvolupar el Projecte Final de Carrera, servidor configurat, base de dades, llenguatges acceptats, etc. És del tot necessari desenvolupar una avaluació de la planificació, per poder ajustar el pressupost i els recursos humans necessaris en forma d'hores.

Els requisits necessaris per desenvolupar el projecte són:

- Disposició d'un ordinador amb connexió a Internet.
- Disposició dels programes necessaris per a desenvolupar el projecte: editors web, programa d'accés al directori de fitxers del servidor web, etc.
- Servidor web degudament instal·lat i configurat per a realitzar les proves pertinents.

Un cop realitzada l'avaluació, s'ha arribat a la conclusió que el projecte és viable, i que el treball d'una sola persona és necessari per al desenvolupament del projecte amb una dedicació aproximada de 480 hores. És per això doncs que ja es pot començar a desenvolupar el Projecte de Final de Carrera, ja que s'ha comprovat la total viabilitat tecnològica i humana.

### 3 Metodologia

La metodologia que s'ha seguit per al desenvolupament d'aquest projecte ha estat l'anomenada *Model de prototipus*. Aquesta metodologia és útil quan el client coneix els objectius generals pel software però no identifica els requisits detallats d'entrada, processament o sortida.

Aquesta metodologia pertany als models de desenvolupament evolutiu. Amb la recollida inicial de requeriments, es desenvolupa un disseny ràpid que es centri en una representació d'aquells aspectes del programari que seran visibles pel client o usuari final. Aquest disseny condueix a la construcció d'un prototip, el qual és avaluat pel client per una retroalimentació. Gràcies a interacció es refinan els requisits del software que es desenvoluparà. Això permet que el desenvolupador, en aquest cas el projectista, entengui millor el que s'ha de fer i el client pugui veure resultats a curt termini.

En aquesta metodologia s'estableixen clarament quines són les fases que apareixen en la producció del programari. Esquemàticament, el model seqüencial consta de les següents fases:

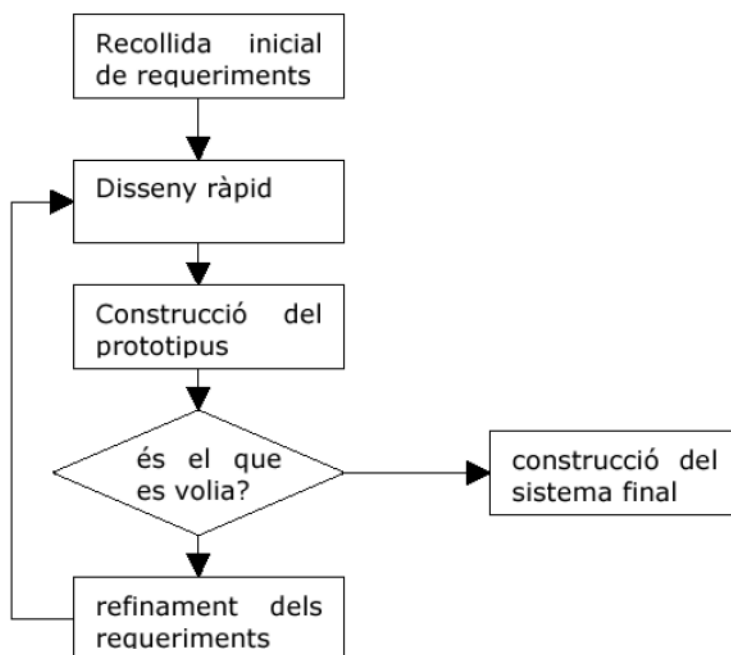


Figura 1: Diferents fases de la metodologia emprada en aquest Projecte Final de Carrera

De fet, la construcció de prototipus no s'ha de veure com un model de construcció de software per si mateix, sinó com una metodologia per ajudar a establir els requeriments de l'aplicació.

## 4 Planificació

Aquest ha estat un projecte de llarga duració. Aquest Projecte Final de Carrera es va iniciar just abans de l'estiu de l'any 2011 i durant els anys 2012 i 2013 fins actualment han durat les tasques de desenvolupament.

Any	2011												2012												2013					
Mes	06	07	08	09	10	11	12	01	02	03	04	05	06	07	08	09	10	11	12	01	02	03	04	05	06					
<b>1. Estructura d'un exercici</b>	■	■	■	■																										
1.1. Estudi del funcionament de la tecnologia XML	■	■	■	■																										
1.2. Desenvolupament del XMLSchema general			■	■	■																									
1.3. Desenvolupament dels XMLSchema específics			■	■	■																									
<b>2. Definició del model</b>					■	■	■	■	■	■	■	■	■	■	■		■	■	■	■										
2.1. Disseny de la jerarquia de classes					■	■	■	■	■	■	■	■	■	■	■		■	■	■	■										
2.2. Implementació en PHP de les classes					■	■	■	■	■	■	■	■	■	■	■		■	■	■	■										
<b>3. Editor d'exercicis</b>																														
3.1. Implementació						■	■	■	■	■	■	■	■																	
3.2. Proves						■	■	■	■	■	■	■	■				■	■	■	■										
<b>4. Implantació a l'ACME</b>																														
4.1. Procés de reconeixement										■	■	■	■																	
4.2. Procés de sorteig										■	■	■	■																	
4.3. Visualització dels enunciats										■	■	■	■																	
4.4. Procés de correcció										■	■	■	■																	
4.5. Proves de funcionament										■	■	■	■				■	■	■	■										
<b>5. Redacció de la memòria</b>																								■	■					

Figura 2: Diagrama de Gantt amb la planificació

## 5 Marc de treball i conceptes previs

### 5.1 Anàlisi de la plataforma ACME

El projecte ACME va sorgir l'any 1998 de la iniciativa d'uns quants professors del Departament d'Informàtica i Matemàtica Aplicada de la Universitat de Girona amb la intenció de poder oferir als alumnes quaderns d'exercicis personalitzats. Des de l'any 1998 fins ara, són molts els becaris i projectistes que han col·laborat en l'ampliació i millora de la plataforma implementant diferents funcionalitats i correctors automàtics.

Amb el llançament d'aquesta plataforma es pretenia aconseguir una major implicació i participació dels alumnes en l'assignatura de Càlcul d'Enginyeria Industrial fent ús de programes de càlcul simbòlic i d'Internet com a via de comunicació, el que va representar tota una revolució educativa l'any 1998.

Una de les característiques més significatives des dels inicis de la plataforma és l'arquitectura modular, la qual cosa permet anar incorporant noves tipologies d'exercicis. El nucli de l'ACME es beneficia de la utilització de la programació orientada a objectes. En el programa s'han creat uns mòduls que permeten l'acoblament d'altres mòduls de manera senzilla, cosa que fa possible que qualsevol projectista pugui fer créixer la plataforma, amb l'únic requeriment que el disseny s'ajusti a un reduït nombre de condicions.

El principal mòdul de l'ACME és el Sistema d'Avaluació Continuada via web. Aquest ajuda l'estudiant a portar l'assignatura al dia i li facilita prendre consciència de quin és el seu nivell. La plataforma es caracteritza pel fet de disposar d'un altre mòdul que actua de repositori únic d'exercicis de diferent tipus: matemàtiques, química, física, programació, bases de dades, economia, etc. Però la part més interessant i potent és que, a més a més, cada un d'aquests tipus d'exercicis consta de diversos enunciats i de paràmetres variables, de manera que, combinant-los, es poden generar molts enunciats diferents d'un mateix exercici. Això vol dir que, per aprendre a resoldre un mateix exercici, els estudiants treballen amb enunciats diferents, la qual cosa els obliga a entendre el procediment per arribar a la solució satisfactòria.

Actualment l'ACME és utilitzat per més de 6000 alumnes en un total de 100 assignatures de diferents facultats de la Universitat de Girona, Universitat Autònoma de Barcelona i diferents instituts de Catalunya. Aquest conté una gran varietat d'exercicis de diferents tipus depenent de l'assignatura que s'imparteixi. El repositori d'exercicis de l'ACME conté prop de 9000 exercicis a disposició dels professors i alumnes que l'utilitzen. Gràcies a la parametrització d'aquests 9000 exercicis, l'ACME és capaç de generar bilions d'enunciats diferents.

### 5.2 Arquitectura Client-Servidor

El navegador és una aplicació capaç d'interpretar les ordres rebudes en forma de codi HTML principalment, i convertir-les en les pàgines que són el resultat d'aquesta ordre.

Quan cliquem sobre un enllaç, en realitat el que passa es que establim una petició d'un



arxiu HTML resident en el servidor (un ordinador que es troba contínuament connectat a la xarxa). Aquest fitxer és enviat i interpretat pel nostre navegador (client).

Així doncs, podem parlar de llenguatges del costat del servidor, que són aquells que són reconeguts, executats i interpretats pel propi servidor i que produeixen una sortida que es redirecciona cap al client, que depenent del format i l'aplicació que ha generat la petició, la interpretarà d'una forma o una altra.

D'altra banda també existeixen els llenguatges del costat del client (entre els quals no només es troba l'HTML, sinó que també el Java i el Javascript, els quals són simplement inclosos en el codi HTML) són aquells que poden ser directament interpretats pel navegador (client).

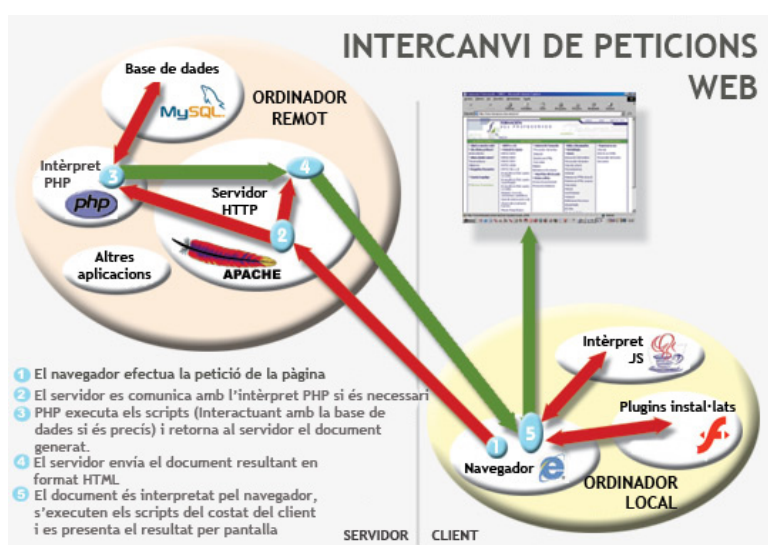


Figura 3: Funcionament de l'arquitectura Client-Servidor d'Internet

Cada un d'aquests tipus tenen les seves avantatges i els seus inconvenients. Així per exemple, un llenguatge del costat del client és totalment independent del servidor, el qual permet que la pàgina pugui ser visible en qualsevol lloc sense la necessitat de contractar un servidor. En canvi, un llenguatge del costat del servidor, que és independent del client, la sortida que generi podrà ser interpretada de la mateixa manera amb qualsevol navegador. Un altre aspecte positiu dels llenguatges del costat del servidor és que els scripts, com hem dit anteriorment són executats i interpretats pel mateix navegador, de manera que aquests romanen ocults pel client. Aquest fet és molt important, ja que d'una forma legítima permet protegir el treball intel·lectual realitzat, i evitar possibles manipulacions per part de l'usuari.

## 5.3 Eines utilitzades

Durant la realització del projecte s'han utilitzat diverses eines informàtiques que es poden agrupar en:

- **Llenguatges de programació i frameworks:** el coneixement i consolidació de llenguatges de programació, en la part d'implementació, ha estat la clau per al correcte desenvolupament del projecte.
- **Entorns de navegació:** s'han utilitzat diferents entorns de navegació pel que fa a la part d'implementació i d'execució. Han ajudat al desenvolupament del projecte i a comprovar que complia totes les seves funcionalitats.

### 5.3.1 Llenguatges de programació del costat del client

#### 5.3.1.1 HTML

HTML, acrònim de *Hypertext Markup Language*, és un llenguatge de marcat dissenyat per estructurar textos i relacionar-los en forma d'hipertext. És usat per descriure l'estructura i el contingut en forma de text, així com per complementar-lo amb objectes com imatges o vídeos. HTML es descriu en forma d'etiquetes prefixades i sufixades pels símbols < i > respectivament. El navegador s'encarregarà de llegir i interpretar les etiquetes. Aquest llenguatge s'ha convertit en un dels formats més populars que existeixen per a la construcció i disseny de pàgines web.

```
<html>
  <head>
    <title>Aquest és el títol de la pàgina</title>
  </head>
  <body>
    <p>Això és un paràgraf</p>
    
    <a href="http://acme.udg.edu">Això és un enllaç</a>
  </body>
</html>
```

Figura 4: Exemple bàsic de HTML

Això és un paràgraf



Això és un enllaç

Figura 5: Visualització d'un exemple bàsic d'HTML

### 5.3.1.2 CSS

CSS, acrònim de Cascading Style Sheets, és un llenguatge de fulls d'estils utilitzat per descriure la semàntica de presentació, aspecte i format, d'un document estructurat en un llenguatge de marques, com l'HTML. La idea que es troba darrere del desenvolupament de CSS és separar l'estructura d'un document de la seva presentació.

```
body { color:red; }  
p { font-size: 20px; font-weight: bold; }  
a { color: green; border: 1px solid black; text-decoration: none; }
```

Figura 6: Exemple bàsic de CSS

**Això és un paràgraf**



**Això és un enllaç**

Figura 7: Visualització d'un exemple bàsic d'HTML amb CSS

### 5.3.1.3 Javascript

Javascript és un llenguatge de programació interpretat. Es defineix com orientat a objectes, basat en prototips, dèbilment tipat i dinàmic amb una sintaxis molt semblant a la del llenguatge Java. És un llenguatge que s'utilitza en la part del client, permet crear pàgines web dinàmiques i millorar la interfície d'usuari entre d'altres coses. Java i Javascript no estan relacionats i tenen semàntiques i propòsits diferents.

El següent codi que tenim a continuació generaria un avís, cada cop que carreguéssim de nou la pàgina. En aquest avís hi hauria el missatge *Hello*:

```
<script type="text/javascript">
  alertMsg();
  function alertMsg() {
    alert("Hello");
  }
</script>
```

Figura 8: Exemple bàsic de Javascript

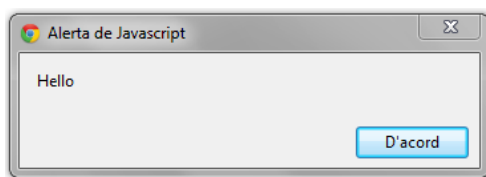


Figura 9: Visualització d'un exemple bàsic de Javascript

## 5.3.2 Llenguatges de programació del costat del servidor

### 5.3.2.1 PHP

PHP, acrònim de Hypertext Pre-processor, és un llenguatge de programació interpretat, dissenyat originalment per la creació de pàgines web dinàmiques. El seu funcionament és molt senzill: s'executa en un servidor web que contingui el PHP instal·lat, el que fa el servidor web és agafar el codi en PHP com l'entrada i l'interpreta, i com a resultat obtenim pàgines web.

Les avantatges d'aquest llenguatge són varies:

- És un llenguatge multi-plataforma.
- Completament orientat al desenvolupament d'aplicacions web dinàmiques amb accés a informació emmagatzemada en una base de dades.

- El codi font escrit en PHP és invisible al navegador i al client, ja que és el servidor qui s'encarrega d'executar el codi i enviar el resultat al navegador.
- Capacitat de connexió amb la majoria dels motors de bases de dades, com MySQL i PostgreSQL.
- Posseeix una àmplia documentació en la seva pàgina oficial.
- Permet aplicar tècniques de programació orientada a objectes.
- Biblioteca nativa de funcions molt àmplia i inclosa.

La plataforma ACME, està basada en aquest llenguatge en la seva versió 5.3.3, així que per fer aquest Projecte Final de Carrera s'ha utilitzat aquest llenguatge per tal de poder integrar tots els mòduls nous a la plataforma.

```
<html>
  <head>
    <title ><?php print "Hello world!"; ?></title >
  </head>
  <body>
    <?php print "Welcome to PHP world "."<br>" ; ?>
    <?php print (date("F d, Y")); ?>
  </body>
</html>
```

Figura 10: Exemple bàsic de PHP

### 5.3.3 Frameworks de programació en Javascript

Un framework és una abstracció de codi comú que introdueix funcionalitats genèriques que poden ser utilitzades per desenvolupar aplicacions de manera ràpida, fàcil, modular i senzilla, estalviant temps i esforç. Llavors, un framework és concret i també incomplet. Concret perquè és, des d'un punt de vista simple, un conjunt de components; incomplet, perquè per si mateixos no poden ser utilitzats, ja que guien a la solució de problemes de programació recurrents.

En la seva majoria, les característiques dels frameworks Javascript són:

- **Compatibilitat.** Agreguen la possibilitat d'escriure codi Javascript totalment compatible amb tots els navegadors i motors Javascript més utilitzats. Això augmenta la portabilitat i eliminen el gran problema d'incompatibilitat entre navegadors i els seus motors intèrprets de Javascript.
- **Comunicació asíncrona.** Fent servir aquest component, és fàcil utilitzar *XMLHttpRequest* per manipular les dades en els elements d'un lloc web, augmentant la interactivitat i l'experiència de l'usuari.

- **DOM.** Maximitzen la capacitat d'afegir, editar, canviar i eliminar elements de manera dinàmica afegint llibreries que faciliten l'ús del DOM.
- **Validació de formularis.** Permeten d'una manera relativament fàcil validar camps dins d'un o varis formularis. Això, des del punt de vista del programador, simplifica i redueix el codi per processar aquests formularis, ja que les dades arriben prèviament validades, reduint els errors de tipus de dades.
- **Efectes visuals.** Utilitzant la manipulació dels elements, es poden crear efectes visuals i animacions. Entre tots els efectes, trobem: aparèixer i desaparèixer, redimensionament, moure, entre d'altres.
- **Emmagatzematge del costat del client.** A més, tenim a la disposició funcions per llegir i escriure galetes, que permet a les aplicacions guardar dades del client, de manera persistent i segura.
- **Maneig JSON.** Incrementa al màxim el maneig de dades, que poden ser utilitzades per presentar informació de manera dinàmica i en temps d'execució.
- **Maneig d'events.** Aquesta característica permet reaccionar d'una manera o altra depenent de les accions de l'usuari.
- **Rebedors de dades.** Permeten utilitzar diferents formats de dades com XML, HTML, Text, JSON, ATOM, entre d'altres.
- **Drag & Drop** (arrossega i deixa anar). És una funcionalitat que dóna la possibilitat d'arrossegar elements dins d'una mateixa pàgina que interactui amb la resta d'elements.

Existeixen varis frameworks de Javascript: *MooTools*, *ExtJs*, *Prototype*, *Scriptaculous*, *Dojo Toolkit*, *YUI*, entre d'altres. Però el que s'ha escollit per a fer servir en aquest Projecte Final de Carrera ha estat el **JQuery**. Els motius han estat per la seva extensa documentació, per la seva simplicitat, per permetre desenvolupar un codi net i elegant, i també perquè ja es va fer servir en el Projecte de la Tècnica i ja es tenia experiència en aquest llenguatge de programació.

Tot seguit, adjuntarem una taula que ens permetrà comparar les característiques de cada framework, juntament amb les seves avantatges i desavantatges.

	Prototype	jQuery	YUI	ExtJS	MooTools
Latest Version	1.6.1	1.4.1	3.00	3.1	1.2.4
License	MIT	MIT & GPL	BSD	Commercial & GPL v3	MIT
<b>Browser Compatibility</b>					
IE Support	6.0+	6.0+	6.0+	6.0+	6.0+
Firefox Support	1.5+	2.0+	3.0+	1.5+	2.0+
Safari Support	2.0.4+	3.0+	4.0+	3.0+	2.0+
Opera Support	9.25+	9.0+	10.0+	9.0+	9.0+
Chrome Support	1.0+	1.0+	Not verified	Not verified	Not verified
<b>Core Features</b>					
Ajax Support	Yes	Yes	Yes	Yes	Yes
DOM Manipulation	Yes	Yes	Yes	Yes	Yes
DOM Traversal	Yes	Yes	Yes	Yes	Yes
Event Handling	Yes	Yes	Yes	Yes	Yes
JSON	Yes	Yes	Yes	Yes	Yes
Selectors	Yes	Yes	Yes	Yes	Yes
<b>UX/UI Enhancements</b>					
Accordion	No	jQuery UI	No	Yes	MooTools More
Animation	scriptaculous	Yes	Yes	Yes	Yes
Auto Completion	scriptaculous	No	Yes	Yes	No
Browser History	scriptaculous	No	Yes	Yes	No
Calendar	No	jQuery UI	Yes	Yes	No
Charting	No	No	Yes	Yes	No
Drag and Drop	scriptaculous	jQuery UI	Yes	Yes	MooTools More
Grids	No	No	Yes	Yes	MooTools More
Progress Bar	No	jQuery UI	Yes	Yes	No
Resizing	No	jQuery UI	Yes	Yes	No
Rich Text Editor	No	No	Yes	Yes	No
Slider	scriptaculous	jQuery UI	Yes	Yes	MooTools More
Tabs	No	jQuery UI	Yes	Yes	No
Themes	No	jQuery UI	Yes	Yes	MooTools More
Tree Views	No	No	Yes	Yes	No

Figura 11: Taula de comparacions de frameworks Javascript 2

### 5.3.3.1 JQuery

JQuery és un framework que emfatitza la interacció entre Javascript i HTML. Amb facilitat podria ser considerat la llibreria més utilitzada en l'actualitat. Aquesta biblioteca creada inicialment per John Resig, permet simplificar la manera d'interactuar amb els documents HTML, gestionar esdeveniments, desenvolupar animacions i afegir interacció amb la tecnologia AJAX en pàgines web. JQuery ofereix una sèrie de funcionalitats basades en Javascript que d'una altra manera requerien molt més codi, és a dir, amb les funcions d'aquesta biblioteca s'aconsegueixen bons resultats en menys temps i espai:

- Selecció d'elements DOM.
- Interactivitat i modificacions de l'arbre DOM, incloent suport per CSS 1-3 i un plugin bàsic de XPath.
- Events.
- Manipulació de la fulla d'estils CSS.
- Efectes i animacions.
- Animacions personalitzades.
- AJAX.
- Suporta extensions (jQuery UI).
- Utilitats com obtenir informació del navegador, operar amb objectes i vectors.
- Compatible amb els navegadors Mozilla Firefox 2+, Internet Explorer 6+, Safari 3+, Opera 1.6+ i Google Chrome 8+.

JQuery consisteix en un sol fitxer Javascript que conté les funcionalitats comuns del DOM, events, efectes i AJAX. La característica principal de la biblioteca és que permet canviar el contingut d'una pàgina web sense necessitat de recarregar-la, mitjançant la manipulació de l'arbre DOM i peticions AJAX. Per això utilitza els selectors `$()` o `jQuery()`.

La forma d'interactuar amb la pàgina és mitjançant el selector `$()`, un alias de `jQuery()`, que rep com a paràmetre una expressió CSS o el nom d'una etiqueta HTML i retorna tots els elements que concordin amb l'expressió donada. Un cop obtinguts, se'ls pot aplicar qualsevol de les funcions que facilita la biblioteca. La podem trobar completa a l'apartat Documentation de l'adreça [www.jquery.com](http://www.jquery.com).



```

$("#taula"); //Retornarà l'element amb id="taula"
$(".actiu"); //Retornarà una matriu d'elements amb class="actiu"

//S'elimina l'estil (amb removeClass()) i se li aplica un de nou (amb
  addClass()) a tots els nodes amb class="actiu"
$(".actiu").removeClass("actiu").addClass("inactiu");

//Anima tots els components amb class="actiu"
$(".actiu").slideToggle("slow");

```

Figura 12: Exemple bàsic de JQuery

### 5.3.3.2 JQueryUI

JQuery UI és una biblioteca de components pel framework JQuery al qual s'afegeix un conjunt de plugins, widgets i efectes visuals per la creació d'aplicacions web. Cada component o mòdul es desenvolupa d'acord amb la filosofia de JQuery (find something, manipulate it: troba alguna cosa, manipula-ho).

La biblioteca es divideix en quatre mòduls:

- **Nucli:** Conté les funcions bàsiques per la resta de mòduls.
- **Interaccions:** Afegeix comportament complexos als elements:
  - Draggable:** Fa que l'element pugui ser arrossegat.
  - Droppable:** Fa que l'element respongui a elements arrossegables.
  - Resizable:** Permet redimensionar l'element.
  - Selectable:** Permet seleccionar entre una llista d'elements.
  - Sortable:** Ordena una llista d'elements.
- **Widgets:** És un conjunt complet de controls UI. Cada control té un conjunt d'opcions configurables i se'ls pot aplicar estils CSS.
  - Accordion:** Menú amb efecte acordió.
  - Dialog:** Finestres emergents amb contingut.
  - Slider:** Element per elegir entre un rang de valors.
  - Tabs:** Pestanyes.
  - Datepicker:** Calendari gràfic.
  - Progressbar:** Barra de progrés.
- **Efectes:** Una API per afegir transicions animades i facilitades per interaccions.

JQuery UI s'utilitza igual que qualsevol altre extensió per JQuery. Únicament s'ha d'afegir els fitxers .js a la pàgina; primer ha d'aparèixer la biblioteca JQuery i després la resta.

```
<script type="text/javascript" src="jquery-1.9.min.js"/>
<script type="text/javascript" src="jquery-ui-1.10.3.custom.min.js"/>
```

Figura 13: Com incloure el JQuery a la nostra pàgina web

A la pàgina oficial de JQuery UI apareix una extensa documentació sobre com usar i configurar els mòduls i components d'aquesta llibreria.

Com JQuery, també és compatible amb els navegadors Mozilla Firefox 2+, Internet Explorer 6+, Safari 3+, Opera 1.6+ i Google Chrome 8+.

### 5.3.4 Altres llenguatges utilitzats

#### 5.3.4.1 $\text{\LaTeX}$

La idea principal de  $\text{\LaTeX}$  és ajudar qui escriu un document a centrar-se en el contingut més que en la forma. La qualitat tipogràfica dels documents realitzats amb  $\text{\LaTeX}$  és comparable a la d'una editorial científica de primera línia. La manera d'elaborar un document en aquest llenguatge es resumeix en dues passes: primerament la de crear un fitxer pla amb les ordres i comandes adequades, que contingui el text que volem imprimir. La segona consisteix en processar aquest fitxer. El processador interpreta les ordres escrites en ell i compila el document, deixant-lo preparat perquè pugui ser enviat a la sortida corresponent, ja sigui una pantalla o la impressora. Fent-ho d'aquesta manera aconseguim un document amb una qualitat de presentació realment alta. Mitjançant aquest llenguatge, entre d'altres, s'han pogut desenvolupar enunciats a l'ACME en forma d'imatge i redactar la memòria d'aquest Projecte Final de Carrera.



Figura 14: Logotip del llenguatge LaTeX

```
Calcula el quadre d'amortitzaci\''{o} i especifica :
  \begin{itemize}
    \item Import que representa la primera quota.
    \item Quota de l'any despr\''{e}s de l'adquisici\''{o}
    \item Import que representa la darrera quota.
  \end{itemize}
Introdueix el resultat separat per comes i per ordre tal i com diu l'enunciat
.
```

Figura 15: Exemple bàsic de codi LaTeX

### 5.3.4.2 Mathematica

Mathematica és un llenguatge utilitzat en àrees científiques, d'enginyeria i matemàtiques. Originalment va ser concebut per Stephen Wolfram, qui continua sent el líder del grup de matemàtics i programadors que desenvolupen el producte. Està basat en la reescriptura de termes i suporta l'ús de programació funcional i de procediments. El llenguatge Mathematica és molt extens i permet fer una gran quantitat d'accions com visualitzar dades en 2D i 3D, manipular i solucionar dades organitzades en matrius, solucionar tot tipus de sistemes d'equacions, utilitzar funcions d'una àmplia llibreria de funcions elementals i especials per matemàtiques, entre d'altres.

```
(* Mètode accelerat *)
If [ETipus==1,
  sumaAnys=0;
  For [i=1, i<=EVidaUtil, i++, sumaAnys=sumaAnys+i];
  part=(EValor-EResidual)/sumaAnys;
  amortitzacio=part*EVidaUtil;
  pendent=EValor-amortitzacio;
  SolInici=amortitzacio;
  SolFinal=part;
  aaim=SolInici;
  mult=EVidaUtil-1;
  For [i=2, i<=EAnyIntermig, i++,
    amortitzacio=part*mult;
    pendent=pendent-amortitzacio;
    aaim=aaime+amortitzacio;
    If [i==EAnyIntermig, If [EResultat==1, SolMig=pendent];
  If [EResultat==2, SolMig=amortitzacio]; If [EResultat==3, SolMig=aaime]];
  mult=mult-1
];
];
```

Figura 16: Exemple bàsic de codi Mathematica

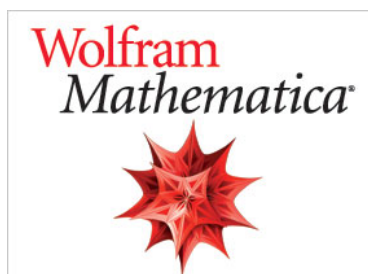


Figura 17: Logotip del llenguatge Mathematica

### 5.3.5 Entorn de desenvolupament

El meu Projecte Final de Carrera és una aplicació web, i per tant, en l'entorn de desenvolupament diversos programes. Bàsicament el cicle de treball s'ha basat en:

- Desenvolupar el codi font amb un editor de text.
- Pujar els fitxers a un servidor utilitzant un software específic.
- Testejar la feina feta amb diferents navegadors web.
- Generar la documentació pertinent.

#### 5.3.5.1 Navegadors web

Un navegador web és una aplicació software que permet a l'usuari recuperar i visualitzar un document d'hipertext (HTML). Aquests són necessaris per visualitzar els exercicis implementats i comprovar que compleixen totes les funcionalitats que esperàvem. Avui en dia existeixen diversos navegadors, però els més usats són Internet Explorer, Mozilla Firefox, Google Chrome, Safari i Opera. Cadascun té la seva manera d'interpretar el codi que conté una pàgina web i alhora de desenvolupar els exercicis s'ha procurat que en els diversos navegadors tot funcionés correctament.



Figura 18: Diferents navegadors que s'han utilitzat en aquest Projecte de Final de Carrera

#### 5.3.5.2 SSH Secure Shell

SSH és un protocol segur de comunicació a través de la xarxa. SSH Secure Shell és un conjunt d'eines que van ser dissenyades des d'un inici per oferir una seguretat màxima i permetre l'accés remot als servidors de manera segura utilitzant el protocol SSH. Per realitzar les transferències de fitxers al servidor, s'ha utilitzat el programa WinSCP, que és un client SFTP gràfic que implementa aquest protocol, que permetia connectar-se al servidor de l'ACME. També ha estat útil el PuTTY, client de terminal TCP, que ha permès executar codi Mathematica, a dins de l'ACME.



Figura 19: Logotips del WinSCP i del PuTTY

### 5.3.5.3 Notepad++

Notepad++ és un editor de codi font lliure que suporta diversos llenguatges de programació i funcions sota l'entorn de Windows. Ha estat el programa mitjançant el qual hem desenvolupat tots els fitxers de codi del projecte.



Figura 20: Logotip del Notepad ++

## 6 Metodologia d'escriptura d'exercicis tradicional

L'ACME té 50 tipologies d'exercicis diferents. Tot i això, no hi ha 50 formes diferents d'escriure els exercicis, moltes tipologies comparteixen la forma d'escriptura però es distingeixen en la forma que l'alumne introdueix la solució (com és el cas dels exercicis de matemàtiques i de química inorgànica), o en la forma que es corregeix l'exercici (com succeeix amb els exercicis de base de dades).

Com s'ha dit anteriorment, una de les principals característiques de l'ACME és que permet escriure exercicis base on cadascun d'aquests té moltes variants que generaran diferents enunciats pels alumnes. És a dir, tots els alumnes fan els mateixos exercicis, però com que hi ha moltes variants de cadascun d'ells, cada alumne disposa d'un conjunt d'enunciats diferents als que pugui tenir qualsevol altre alumne. Aquestes diferències entre exercicis venen donades pels diferents paràmetres que s'utilitzen a l'hora de definir-lo i pels textos que expliquen els objectius de l'exercici. La majoria de les definicions dels exercicis d'ACME es realitzen amb  $\LaTeX$ , ja que permet renderitzar i escriure fórmules matemàtiques de forma senzilla.

Els diferents redactats i paràmetres, que fan que un mateix exercici tingui moltes variants, són definits i entrats al sistema pels professors, de manera que cada exercici té la seva definició associada. És llavors quan la funció de generació d'enunciats fa servir la definició de l'exercici per a poder crear totes les seves variants. Aquesta funció de generació s'encarrega d'escollir a l'atzar un dels redactats i un conjunt de paràmetres per finalment generar un enunciat.

Per tal de fer l'anàlisi del format escriptura actual d'exercicis de l'ACME ens centrarem un únic tipus d'exercici que sigui representatiu. Els exercicis tipus 12, són una evolució dels exercicis matemàtics inicials de l'ACME (actual tipus 1), i aquesta tipologia d'exercici es pot considerar la base a partir de la qual la plataforma ha anat creixent.

### 6.1 Anàlisi de l'escriptura d'un exercici

El primer que fa falta per crear un conjunt d'enunciats és la definició de l'exercici. Un exercici, com ja s'ha comentat anteriorment, pot tenir diferents enunciats i diferents paràmetres dels quals la funció de generació s'encarregarà d'escollir-ne un de cada a l'atzar. Un exercici com a mínim ha de tenir un paràmetre amb un únic valor d'aquest paràmetre, és a dir, que pot haver-hi exercicis en els quals no hi hagi cap tipus de variació.

Les definicions de cada exercici s'implementen en fitxers separats. En un fitxer únicament hi haurà la definició d'un tipus d'exercici en particular. El fitxer que conté la definició de l'exercici ha de ser en format  $\LaTeX$  i que sigui compilable pel  $\LaTeX$ . El fitxer ha d'estar estructurat d'una manera molt concreta i complir estrictament unes regles sintàctiques i semàntiques, que s'expliquen a continuació.

### 6.1.1 Capçalera del fitxer

Al principi de cada fitxer hi ha d'haver una petita capçalera que identifiqui de quin tipus d'exercici es tracta. La capçalera ha de coincidir amb el número de tema i el número d'exercici amb el que s'especifica en el nom del fitxer.

```
12
\begin{center}
\large {\bf Problema —ACME}
\end{center}
```

Figura 21: Exemple de capçalera d'un exercici

El número que hi ha a la primera línia de codi correspon al número del tipus d'exercici que serà el que s'implementi.

### 6.1.2 Enunciat dels exercicis

Un exercici pot tenir diverses redaccions dels enunciats, independentment dels paràmetres d'aquest. Els enunciats es definiran de la següent forma:

```
* % <E>
Doneu la derivada de la funci\ 'o $f(x)=$
% <1>
{\bf P1}
.
% </E>
\par
```

Figura 22: Exemple de definició d'un paràmetre en l'enunciat

L'enunciat no és més que una breu explicació del procediment que s'ha de seguir per a fer l'exercici de manera correcta.

L'identificador que indica el començament d'un nou enunciat és % < E >. Cal respectar les majúscules i els espais en blanc entre el % i < E >, ja que sinó la funció de generació no identificarà bé l'estructura. A partir d'aquesta marca comença el redactat de l'exercici. El final d'enunciat el marcaran els caràcters % < /E >.

Es poden posar tants enunciats com es cregui oportú, un darrere l'altre, sempre respectant la sintàctica explicada. En l'exemple d'enunciat anterior es poden apreciar els caràcters % < 1 > {\bf P1}. Això fa que es mostri el valor escollit a l'atzar del paràmetre que s'indiqui en el redactat de l'enunciat.

A l'hora de mostrar el valor d'un paràmetre dins del mateix enunciat s'ha d'indicar seguint una sintaxi concreta. Primerament s'ha de posar l'identificador `% <N >`, on es substituirà N pel número de paràmetre desitjat. És molt important fer un canvi de línia després d'aquest identificador i escriure, ja en la línia següent, els caràcters `{\bf P1}` seguits d'un altre salt de línia. Aquests salts de línia no es tindran en compte a l'hora de generar la imatge de l'enunciat però és obligat definir-ho d'aquesta manera per a mantenir l'estètica de l'enunciat.

La funció de generació agafarà l'expressió del paràmetre N que s'hagi escollit a l'atzar i la inclourà allà on s'hi hagin introduït aquests caràcters que s'han explicat. Un paràmetre pot aparèixer a l'enunciat tantes vegades com es desitgi i sempre s'utilitzarà la mateixa expressió generada en cada repetició.

Si es desitja mostrar més d'un paràmetre en l'enunciat, s'haurà de seguir el mateix procediment anterior però canviant N pel número de paràmetre escollit. Obligatòriament tots els paràmetres s'han de mostrar en algun dels redactats definits. Cap paràmetre pot ser definit i no mostrat en cap dels enunciats.

A continuació s'adjunta un exemple d'un exercici amb 3 paràmetres a dins de l'enunciat:

```
* % <E>
Calcula el quadre d'amortitzaci\''{o} utilitzant el m\''{e}tode
% <1>
{\bf P1}
i especifica :
  \begin{itemize}
    \item Import que representa la primera quota.
    \item
      % <2>
      {\bf P2}
      % <3>
      {\bf P3}
      any despr\''{e}s de l'adquisici\''{o}
    \item Import que representa la darrera quota.
  \end{itemize}
Introdueix el resultat separat per comes i per ordre tal i com diu l'enunciat
. Si el primer valor que ens demanen \''{e}s de $350 \text{\EUR}$, el segon \''{e}
s de $795.25 \text{\EUR}$ i el darrer \''{e}s de $733.33 \text{\EUR}$, haurem d'escriure
: \begin{verbatim}350,795.25,733.33\end{verbatim}
% </E> \par
```

Figura 23: Exemple d'un exercici amb 3 paràmetres declarats

El contingut del redactat de l'enunciat passarà per un compilador  $\text{\LaTeX}$ , per la qual cosa s'ha de fer servir aquest llenguatge a l'hora d'escriure el text. Per escriure enunciats més complexos i incloure-hi taules, llistats i demés, s'han de fer servir la sintaxi específica del  $\text{\LaTeX}$  perquè el compilador pugui reconèixer-ho.



### 6.1.3 Definició dels paràmetres de l'exercici

Tot seguit, després de la definició dels redactats, ha d'anar a continuació la definició dels paràmetres de l'exercici. Un paràmetre pot ser qualsevol expressió numèrica: funcions, integrals, punts de coordenades, números i, fins i tot, una cadena de caràcters. Com ja s'ha dit, com a mínim n'hi ha d'haver un.

Els caràcters que indiquen el començament i el final de la definició de paràmetres han de ser `% <P >` i `% </P >` respectivament. Entre aquests dos grups de caràcters s'afegirà la definició de cada paràmetre. La definició de cada paràmetre haurà d'anar precedida dels caràcters `% <N >` i un salt de línia, on es substituirà N pel número de paràmetre que s'estigui definint. Per a informar de la finalització de la seva definició, s'haurà d'indicar mitjançant `% </N >`. Entre aquests dos identificadors s'hauran d'escriure les expressions que podrà adoptar cada paràmetre. De cares al reconeixement sintàctic és important que cada expressió estigui escrita en la mateixa línia del fitxer.

Les expressions dels paràmetres constaran de dues parts ben diferenciades:

- El valor **visual**, que serà utilitzat a l'hora de mostrar el paràmetre en el redactat de l'enunciat i haurà d'estar escrit en format L<sup>A</sup>T<sub>E</sub>X. Aquest valor serà el que veuran els alumnes en els enunciats.
- El valor **calculat**, que serà utilitzat en el codi de correcció per calcular la solució correcta de l'exercici i avaluar la correctesa de la solució enviada per l'alumne. Aquest valor haurà de ser escrit en llenguatge Mathematica, que és el llenguatge que utilitza l'ACME a l'hora de corregir els diferents exercicis que envien els alumnes.

Cada valor s'haurà d'escriure amb un \$ al principi i un altre al acabar, de manera que es pugui identificar sintàcticament on es troba l'inici i el final del valor del paràmetre. Entre els dos \$ pot haver-hi qualsevol tipus de caràcter excepte el mateix \$, que indica el final d'expressió. Els valors visual i calculat han d'estar separats pels caràcters `\#`.

```
\underline{Par\ 'ametres:}\par
\vspace{0.5cm}
% <P>
{\bf P1} \par % <1>
  $1984115$\#1984115$. \par
  $1984238$\#1984238$. \par
  $1984342$\#1984342$. \par
% </1>
{\bf P2} \par % <2>
  $6059494$\#6059494$. \par
  $6059537$\#6059537$. \par
  $6059311$\#6059311$. \par
% </2>
% </P>
```

Figura 24: Exemple de declaració de paràmetres

En el quadre anterior es pot veure un exemple d'un exercici que té dos paràmetres amb diverses expressions cadascun.

També hi ha la possibilitat d'afegir **variables** dins dels paràmetres. Aquesta opció fa que encara hi hagi més diversitat en la generació d'enunciats únics, augmenta considerablement la aleatorietat i disminueix dràsticament el percentatge de possibilitats de que dos enunciats siguin iguals i es generin amb els mateixos paràmetres.

El sistema de variables es tracta d'un sorteig addicional dins del sorteig del valor del paràmetre. S'inclou el nom de la variable enmig de la definició de l'expressió del paràmetre i a continuació es declararen els possibles valors que pot agafar la variable, separats per comes. Aquesta definició ha d'estar estrictament estructurada sintàcticament: ha d'estar entre \$, precedit d'una coma (,) i els possibles valors de la variable han d'estar degudament delimitats per claudàtors ( $\{$  i  $\}$ ) i el nom de la variable ha d'estar en majúscules.

```

\vspac{1cm}
\underline{Par\ 'ametres:}\par \vspac{0.5cm}
% <P>

{\bf P1} \par % <3>
$<li>El valor de les mercaderies és: X €</li>$, $X
  =\{13804,12371,12518,13452,13163\}\$ \# $ X $, $X
  =\{13804,12371,12518,13452,13163\}$ .

$<li>Mercaderies: X €</li>$, $X=\{13804,12371,12518,13452,13163\}\$ \# $ X $, $
  X=\{13804,12371,12518,13452,13163\}$ .
$<li>Valor de les mercaderies en stock: X €</li>$, $X
  =\{13804,12371,12518,13452,13163\}\$ \# $ X $, $X
  =\{13804,12371,12518,13452,13163\}$ .

$ \$ \# $ 0 $.
% </1>

% </P>

```

Figura 25: Exemple de declaració d'un paràmetre amb variables

En el quadre anterior es pot veure un exemple de dues expressions d'un paràmetre amb variables. Les variables s'han de declarar tant en la part visual com en la part calculada. El valor escollit, òbviament, serà el mateix als dos costats de l'expressió.

A més a més, es pot forçar el sorteig dels paràmetres dels enunciats, és a dir que, la selecció d'una expressió d'un paràmetre pot condicionar la selecció de l'expressió del següent paràmetre. Per a fer això s'ha d'indicar, al costat de la definició de cada expressió, quins són els valors vàlids del següent paràmetre del llistat. Per exemple, introduint aquests caràcters  $\{Combinada : 1, 2, 3\}$  en la definició d'una expressió d'un paràmetre, si a l'hora de fer el sorteig és escollida aquesta expressió, quan sigui moment d'escollir quin valor pertoca al següent paràmetre, només tindrà en compte pel sorteig aleatori els tres primers valors.

En el quadre següent es pot apreciar que el paràmetre 1 només pot combinar amb determinades expressions del paràmetre 2 segons l'expressió que hagi estat escollida en el primer paràmetre:

```

\vspace{1cm}
\underline{Par\ 'ametres:}\par \vspace{0.5cm}
% <P>
{\bf P1} \par % <1>
$PERFUMS$\#\# $. \{Combina:1\} \par
$FRAGÀNCIES$\#\# $. \{Combina:1\} \par
$IL .LUMINÀRIA$\#\# $. \{Combina:2\} \par
$LLETRATS$\#\# $. \{Combina:2\} \par
$PICAPLETS$\#\# $. \{Combina:3\} \par
$ADVOCATS$\#\# $. \{Combina:3\} \par
% </1>

{\bf P2} \par % <2>
$perfums.$\#\# $.
$llums.$\#\# $.
$jurídics.$\#\# $.
% </2>
\par
% </P>

```

Figura 26: Exemple de declaració d'un paràmetre amb combinacions

#### 6.1.4 Codi Mathematica de resolució

Tots els exercicis han de contenir un codi de resolució per comprovar la correctesa de les solucions enviades pels alumnes. Aquest codi vindrà encapçalat pel títol `\underline{CodiMathematica : }`, i estarà comprès entre els tags `\begin{verbatim}` i `\end{verbatim}`. En el codi hi han d'aparèixer tots els paràmetres de l'exercici, per la qual cosa no està permès ometre el valor de cap paràmetre. En el codi de correcció, d'alguna manera s'ha de disposar de la solució que ha enviat l'alumne en cada pas per, un cop calculada la solució correcta amb els paràmetres que s'han escollit prèviament, poder comparar-la. Per a poder disposar d'aquesta informació, s'inclourà la paraula clau SO al principi del codi de correcció. Aquesta paraula, a l'hora de corregir un exercici, serà substituïda per la solució enviada per l'alumne.

En el quadre següent s'hi poden veure les paraules clau d'un codi de correcció. La capçalera a l'inici, la cadena SO que es substituirà amb la solució enviada per l'alumne, P1 i P2 que es substituiran pels valors dels paràmetres corresponents, i finalment les paraules claus d'avaluació, com són Correcte, Incorrecte i Incomprensible. És important també que el codi Mathematica contingui les paraules clau per avaluar el resultat de la correcció. No és obligatòria la seva presència. A continuació es pot veure un exemple de codi de correcció:

```

\underline{Codi Mathematica:}
\begin{verbatim}

Clear [ sol , p1 , p2 , f , g ];
sol=SO;
p1=P1;
p2={P2};
f [ x_ ]=p2 [[ 1 ]];
g [ x_ ]=p2 [[ 2 ]];
s=Solve [ { y==f [ x ] , y==g [ x ] } , { x , y } ];
s= x /. s;
x1=Min [ s ];
x2=Max [ s ];
Integrate [ p1*(f [ x ]-g [ x ])^2 , { x , x1 , x2 } ];
v=N [%];
If [ Abs [ (N [ sol ]-v)/v ] <= 5*10^(-4) , Print [ " Correcte " ] ,
Print [ " Incorrecte " ] , Print [ " Incomprensible " ] ]
\ end{verbatim}

```

Figura 27: Exemple de codi de resolució

## 6.2 Exercici d'exemple

L'exemple que s'adjunta a continuació correspon a un dels nous exercicis que es van implementar en el meu Projecte Final de Carrera de l'Enginyeria Tècnica en Informàtica de Gestió. Es tracta d'un exercici d'Economia i Administració d'Empreses.

L'exercici demana que l'alumne calculi les amortitzacions corresponents a cada exercici econòmic d'un capital establert, utilitzant un dels mètodes disponibles, lineal o quotes fixes.

```

12
\begin{center}
\large {\bf Problema 3.02---ACME}
\end{center}

\underline{Enunciats:}\par \vspace{0.5cm}

* % <E>
Una empresa petrolera est\ '{a} considerant l'opci\ '{o} de comprar els drets
d'exploraci\ '{o}
d'un dels dos jaciments del mar del nord (A i B). Si compra els drets de A,
hi ha una probabilitat de
$
% <1>
{\bf P1}
$
que trobi petroli a la primera, amb un benefici de
$
% <2>
{\bf P2}
$

```

euros. Si falla, pot decidir intentar-ho una altra vegada, en aquest cas la probabilitat d'èxit és de

\$

% <3>

{\bf P3}

\$

amb un benefici de

\$

% <4>

{\bf P4}

\$

euros.

\begin{enumerate}

\item Quin benefici obtenim si comprem el jaciment A?

\item i si comprem el jaciment B?

\end{enumerate}

\textbf{NOTA: D'ona les respostes separades per comes i arrodonida a tres decimals. Si voleu evitar problemes amb els decimals no arrodoniu els calculs intermitjos}

EXEMPLE: Si comprant el jaciment A ens surt 10 i comprant el B ens surten 5 llavors posar: \textbf{10,5}

% </E>

\par

\vspace{1cm}

\underline{Paràmetres:}\par \vspace{0.5cm}

% <P>

{\bf P1} \par % <1>

\$0.35\#\\$0.35\$. \par

\$0.4\#\\$0.4\$. \par

% </1>

{\bf P2} \par % <2>

\$70000\#\\$70000\$. \par

\$72000\#\\$72000\$. \par

% </2>

{\bf P3} \par % <3>

\$0.35\#\\$0.35\$. \par

\$0.4\#\\$0.4\$. \par

% </3>

{\bf P4} \par % <4>

\$66000\#\\$66000\$. \par

\$67000\#\\$67000\$. \par

% </4>

{\bf P5} \par % <5>

\$26000\#\\$-26000\$. \par

\$27400\#\\$-27400\$. \par

% </5>

{\bf P6} \par % <6>

\$12000\#\\$12000\$. \par

\$12800\#\\$12800\$. \par

% </6>

```

{\bf P7} \par % <7>
$0.35$\#$0.35$. \par
$0.4$\#$0.4$. \par
% </7>
{\bf P8} \par % <8>
$50000$\#$50000$. \par
$52000$\#$52000$. \par

% </8>
{\bf P9} \par % <9>
$0.35$\#$0.35$. \par
$0.4$\#$0.4$. \par
% </9>
{\bf P10} \par % <10>
$38000$\#$38000$. \par
$40000$\#$40000$. \par
% </10>
{\bf P11} \par % <11>
$26000$\#$$-26000$. \par
$27400$\#$$-27400$. \par
% </11>
{\bf P12} \par % <12>
$8000$\#$8000$. \par
$10000$\#$10000$. \par
$12000$\#$12000$. \par
% </12>
%</P>

\vspace{1cm}

\underline{Codi Matem\`atica:}
\begin{verbatim}
Clear [ sol , p1 , p2 , p3 , p4 , p5 , p6 , p7 , p8 , p9 , p10 , p11 , p12 , k1 , k2 , s1 , s2 , s3 ];

sol={SO};
SetOptions [ $Output , PageWidth->Infinity ];
correcte=True;
missatgeError="";
If [ VectorQ [ sol , NumberQ ] == False , correcte=False ; missatgeError=missatgeError <> " <
br> La soluci3n introduida ha de ser una llista de n3meros " ];
If [ Length [ sol ] != 2 , correcte=False ; missatgeError=missatgeError <> " <br> La soluci3
3n introduida ha de ser una llista de 2 n3meros " ];

If [ Not [ correcte ] , Print [ " Missatge: Hi ha errors: " <> missatgeError <> "
ESintactic " ]; Quit [ ] ];
p1=P1;
p2=P2;
p3=P3;
p4=P4;
p5=P5;
p6=P6;
p7=P7;
p8=P8;
p9=P9;
p10=P10;
p11=P11;

```

```

p12=P12;

k1=p3*p4+(1-p3)*p5;
k2=p9*p10+(1-p9)*p11;

s1= (1-p1)*k1+p1*p2;
s2= p7*p8+(1-p7)*k2;

pBona[x-,y-]:= Abs[x-y]<0.001;

dacord=True;
textError="";
If[Not[pBona[sol[[1]],s1]],dacord=False;textError=textError<<"<br>En 1'
  apartat 1"];
If[Not[pBona[sol[[2]],s2]],dacord=False;textError=textError<<"<br>En 1'
  apartat 2"];

If[dacord,Print["Correcte"],Print["Missatge:Hi ha error(s): <br>"]<<
  textError<<" FMissatge"]];
\ end{verbatim}

```

En aquest exemple s'han esborrat molts possibles valors en els paràmetres i s'ha reduït l'enunciat per tal de no fer massa extens aquest exemple. A partir d'aquest enunciat, ACME pot generar una gran quantitat d'exercicis entre ells el que és visualitza a continuació en el que s'han substituït els paràmetres per algun dels valors possibles que poden agafar.

### 6.3 Consideracions a la metodologia tradicional

A continuació, es resumeixen els punts positius que té la manera tradicional d'escriure exercicis per l'ACME. També es farà un llistat de totes les restriccions amb les que es troben els professors a l'hora de crear nous exercicis.

**Aspectes positius** i que cal mantenir a l'hora de dissenyar la nova metodologia d'escriptura d'exercici nous per l'ACME:

- Molta **aleatorietat** a l'hora de generar exercicis diferents pels alumnes. La metodologia proporciona els recursos necessaris perquè les possibilitats de que es sorteigi un exercici amb els mateixos paràmetres siguin molt baixes.
- El fet que pugui haver-hi diferents redactats d'enunciats i que permeti generar els enunciats en  $\LaTeX$  com a **imatges**.
- La metodologia de sorteig de **paràmetres** és molt completa. És la principal eina per fer que els exercicis siguin únics en els diferents dossiers d'exercicis. Caldria mantenir l'estructura de **variables** i de **combinacions**.

**Restriccions** de l'actual metodologia d'escriptura d'exercicis de l'ACME, que, amb la nova eina, caldria desbloquejar i permetre noves funcionalitats:

- La complexitat d'haver de crear un fitxer amb una estructura sintàctica tant restrictiva i en un format a mig camí del  $\LaTeX$  en combinació amb tags propis i que poden anar canviant segons la tipologia d'exercici.
- La impossibilitat de generar paràmetres calculats a partir dels valors sortejats d'altres paràmetres. La llista de valors que permet l'escriptura actual és molt potent, però encara queden exercicis on seria més interessant calcular valors o de forma aleatòria o a partir d'un conjunt de restriccions, o fer l'enginyeria inversa a partir d'una solució calcular els paràmetres que s'han de mostrar als alumnes.
- Només es poden crear redactats d'enunciats en el format que el tipus d'exercici permet, que en la seva majoria és  $\LaTeX$ . Interessaria tenir la possibilitat de redactar en format text simple o HTML, o de poder-los combinar a gust o necessitat del professor.
- No es poden crear exercicis d'una forma raonable, on les seves solucions siguin de diferents tipologies, però amb els mateixos paràmetres, i en el mateix exercici. Un bon exemple es dona a economia, un exercici on l'alumne ha de realitzar un balanç de situació i un comptes de pèrdues i guanys; o bé a base de dades on es demana fer el model entitat-relació i el posterior disseny relacional d'una base de dades.
- L'usuari se l'ha d'abstreure del complex llenguatge de tags necessari per escriure un exercici. Alhora que el nou llenguatge de tags ha de ser el més estàndard possible.
- El nou format d'escriptura hauria d'incorporar metadades com el títol de l'exercici, l'explicació, l'autor, classificació,... informació que ara mateix està deslligada de l'exercici.



## 7 Nova metodologia d'escriptura d'exercicis

Un cop analitzada la metodologia d'escriptura d'exercicis tradicional de l'ACME, es procedirà a definir la nova metodologia, els requisits i funcionalitats específiques que tindrà l'aplicació per tal de resoldre els punts febles i mantenir els punts forts de la implementació tradicional.

Per tant, després de veure les principals mancances de l'aplicació tradicional, els requisits que haurà de complir la metodologia que es desenvolupa en aquest Projecte de Final de Carrera són les següents:

- La nova metodologia ha de conviure amb l'existent, per tant els professors que utilitzin ACME podran fer servir exercicis en el format antic o en el nou indistintament.
- Abordarà les mancances del conjunt d'exercicis actuals.
- Haurà de permetre combinar diferents apartats a dins un mateix exercici de forma natural.
- Els enunciats hauran de poder ser escrits en diferents formats indistintament.
- Cada apartat podrà associar-se a un nucli corrector en particular i diferent.
- La possibilitat de poder generar paràmetres calculats a partir dels valors dels altres paràmetres.
- Es permetrà diferents tipus de correcció i visualització del conjunt global de preguntes.

Tots aquests requisits pretenen assolir els objectius que s'han plantejat en aquest Projecte de Final de Carrera.

Per a complir amb el primer dels requisits que s'han proposat, que la nova metodologia convisqui amb l'existent, és farà que aquesta nova metodologia d'escriptura d'exercicis correspongui a una nova tipologia d'exercici de l'ACME. Actualment a l'ACME existeixen moltes tipologies diferents d'exercicis com són els exercicis matemàtics, de programació, químics... Cadascun d'aquests exercicis porta associat un número de tipus d'exercici. Llavors, doncs es procedirà a crear un nou tipus d'exercici de l'ACME i se li donarà el número 0, amb aquest petit canvi es podrà enllaçar tota la part antiga d'exercicis amb la part nova de forma raonable.

S'hauran de fer modificacions a nivell de codi a la funció d'addició i identificació de nous exercicis, a la funció de generació o sorteigs d'exercicis i a la funció de correcció de l'ACME perquè tracti de forma específica aquesta nova tipologia que es crea en aquest Projecte de Final de Carrera.

D'aquesta manera s'aconsegueix fer que la nova metodologia d'escriptura pugui funcionar integrada completament dins la plataforma ACME, mantenint actiu el funcionament de la metodologia tradicional d'escriptura, i tots els mòduls interrelacionats.

Els diferents passos que s'hauran de seguir per a la creació de la nova metodologia d'escriptura d'exercicis seran els següents:

1. **Definició de l'estructura de l'exercici:** es definiran els diferents atributs d'un exercici i s'emmagatzemaran en un fitxer de manera estructurada, reconegut per l'ACME i que sigui fàcil d'extreure tota la informació.
2. **Implementació d'un editor que permeti crear un exercici:** a diferència de la metodologia tradicional, que s'ha de construir el fitxer amb la definició de l'exercici, en aquest PFC es crearà un formulari web de manera que la creació de nous exercicis no impliqui la construcció d'un fitxer amb tantes restriccions sintàctiques, sinó que es pretén proporcionar totes les facilitats possibles perquè a partir del formulari, la creació d'un nou exercici sigui fàcil i intuïtiu de dur-ho a terme.
3. **Adaptar la plataforma ACME perquè reconegui correctament aquesta nova tipologia creada:** un cop obtingut el fitxer definició del nou exercici, s'haurà d'adaptar la plataforma ACME perquè reconegui aquests fitxers amb la nova estructura i els nous exercicis puguin ser afegits a la base de dades de l'ACME.
4. **Adaptar el procés de sorteig d'exercicis de l'ACME:** a part d'adaptar l'ACME perquè reconegui aquests nous fitxers, també s'haurà d'adaptar-lo perquè faci el sorteig d'exercicis de forma correcta i separada dels exercicis tradicionals.
5. **Adaptar el procés de visualització d'exercicis de l'ACME:** un cop sortejats exercicis de tipologia nova, caldrà també adaptar la visualització d'exercicis de l'ACME, de manera que els enunciats puguin ser mostrats correctament.
6. **Adaptar el procés de correcció d'exercicis de l'ACME:** i finalment, també caldrà adaptar el sistema de correcció d'exercicis de l'ACME perquè amb la nova tipologia d'exercicis s'afegeixen noves possibilitats de correcció que amb l'anterior versió no es contemplaven.

Seguidament, s'adjunta una taula comparativa entre la metodologia d'escriptura tradicional i la nova metodologia d'escriptura d'exercicis implementada en aquest Projecte Final de Carrera:

<b>Esriptura tradicional</b>	<b>Nova escriptura</b>
Estructura del fitxer definició pròpia per cada tipologia d'exercici.	Estructura comuna per a totes les tipologies d'exercici.
Editor Windows que permet editar el fitxer definició a mà. Cal tenir coneixement tècnic.	Editor web intuïtiu i fàcil d'usar. No cal tenir coneixement tècnic per crear un nou exercici.
Una sola pregunta per un exercici.	Vàries preguntes de diferents tipologies en un sol exercici.
Visualització vertical i correcció total.	S'incorpora la visualització horitzontal i la correcció incremental (per apartats separats).
Enunciats únicament en format $\text{\LaTeX}$ .	Diferents formats d'escriptura d'enunciats.
Sorteig de paràmetres d'entre llistat de valors.	Incorporació de paràmetres calculats.

## 7.1 Estructura d'un exercici

Un dels aspectes de la metodologia d'escriptura d'exercicis tradicional que es vol mantenir és el fet que cada exercici tingui un fitxer individual que contingui la seva pròpia definició i que llavors a partir d'aquest fitxer, l'ACME pugui reconèixer-lo i tractar-lo. Ara bé, aquesta estructura haurà de ser comuna per a tots els exercicis, de manera que qualsevol tipus d'exercici faci servir la mateixa estructura per al seu fitxer definició. Hi haurà aspectes de la metodologia tradicional que es mantindran, però també n'hi haurà molts d'altres que es modificaran i s'afegiran de nous.

El nou fitxer definició d'un exercici haurà d'estar preparat per tenir camps on es puguin emmagatzemar l'**autor** de l'exercici, un **títol** que l'identifiqui, una breu **descripció** explicant el seu propòsit i l'**idioma** amb el que està escrit. També haurà de contenir dos atributs que defineixin el tipus de **visualització** amb el que serà mostrat l'exercici i el tipus de **correcció** que s'aplicarà a l'hora d'avaluar les solucions enviades per l'alumne.

Llavors, un exercici estarà compost per dos llistats globals: un de **preguntes** i un altre de **paràmetres**. Cada pregunta individual haurà de tenir un llistat d'**enunciats**, **fitxers** i **solucions**.

Cada enunciat haurà de tenir els seus propis atributs, un camp **text** on hi haurà el redactat de l'enunciat i un altre camp **tipus** que es pugui indicar amb quin llenguatge ha estat escrit. Pel que fa als fitxers també dos camps: **URL** amb l'adreça on està emmagatzemat i **tipus** que contindrà la tipologia de fitxer que es tracta. I referent a una solució, haurà de tenir el **tipus** de solució que es tracta, el tipus de **corrector** que es farà servir a l'hora d'avaluar i finalment el fitxer definició haurà d'estar habilitat per a poder afegir els atributs específics de cada tipus de solució.

Pel que fa als paràmetres, el fitxer definició haurà de permetre afegir el **número** de paràmetre i el **tipus** del qual es tracta. Llavors, igual que en les solucions, s'haurà de poder afegir nous atributs específics de cada tipologia de paràmetre.

Llavors per cada tipus de solució i paràmetre que s'implementi s'haurà de definir el seu propi fitxer definició especificant els atributs específics de l'element. D'aquesta manera es farà que l'aplicació sigui extensible, permetent afegir a posteriori tants tipus de solucions com paràmetres com es vulgui.

### 7.1.1 Tecnologia utilitzada

Després de definir els requeriments que ha de tenir l'estructura del nou fitxer definició, es va veure que el llenguatge que es fa servir pels fitxers definició tradicionals no era del tot adequat i es va decidir canviar el llenguatge utilitzat. S'ha passat de fer servir el llenguatge  $\text{\LaTeX}$  a fer servir un llenguatge específic per estructurar dades com és el llenguatge **XML**.

XML és un llenguatge de marques reconegut com el llenguatge per excel·lència per encapsular informació. És un format universal per documents i dades estructurades a Internet.

Aquest estàndard permet l'intercanvi d'informació estructurada entre diferents plataformes. És acceptat per la majoria dels llenguatges de programació. S'ha escollit aquest llenguatge per diversos motius:

- Fàcilment processable pel llenguatge de programació del costat del servidor que utilitza l'ACME, el PHP. A més, aquest llenguatge té els seus propis mètodes de verificació i tractament de fitxers XML, per la qual cosa ha estat un punt clau en l'elecció d'aquest llenguatge per a la definició dels exercicis.
- La informació s'emmagatzema en text pla i separa de manera eficient el contingut de la seva presentació o format.
- La seva estructura i el seu anàlisi sintàctic són fàcils d'entendre.
- És extensible. Després de fer el disseny i posar-lo en funcionament és senzill afegir noves etiquetes, realitzar modificacions, etc. De manera que un cop implementada l'aplicació se li podran afegir noves funcionalitats sense cap tipus d'inconvenient.
- És el llenguatge per estructurar informació per excel·lència. És reconegut universalment per aplicacions de diferents plataformes i facilita la compartició i intercanvi de la informació.

Primer de tot, s'ha de definir com ha de ser l'estructura dels documents XML que es volen construir. Per a fer això s'utilitzarà XML Schema, un llenguatge d'esquema utilitzat per descriure l'estructura i les restriccions dels continguts dels documents XML de forma molt precisa, més enllà de les normes sintàctiques imposades pel propi llenguatge XML. La programació en Schema XML es basa en namespaces o espais de nom. Cada namespace conté elements i atributs que estan estretament relacionats amb el namespace. D'aquesta manera, a l'hora de definir un element o un atribut d'un namespace, sempre es crearà una connexió entre els diferents camps d'aquest. Després d'escriure un Schema XML es pot confirmar la correcta realització mitjançant la validació d'esquemes XML.

Un exemple de l'estructura d'un document esquema buit seria el següent:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
</xsd:schema>
```

Figura 28: Exemple d'estructura d'un document esquema XML buit

Un exemple de definició d'un XML Schema senzill seria:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="Llibre">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="Títol" type="xsd:string"/>
        <xsd:element name="Autors" type="xsd:string" maxOccurs="10"/>
        <xsd:element name="Editorial" type="xsd:string"/>
      </xsd:sequence>
      <xsd:attribute name="Preu" type="xsd:double"/>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Figura 29: Exemple de XML Schema senzill

Es pot veure com en ambdós casos s'inicien les declaracions indicant la versió de XML que s'utilitzarà i la codificació que es fa servir. Aquests dos camps són necessaris per interpretar l'esquema. A la següent línia de codi es declara l'element *xsd:schema*, arrel del document de definició d'esquema. A més, es declara l'espai de noms en el que s'emmagatzemaran els noms dels elements del document. Perquè un document XML estigui correctament format, a partir de l'esquema anterior, haurà de contenir un element arrel anomenat *Llibre*, el qual tindrà tres fills i un atribut. Els fills seran *Títol*, *Editorial* i *Autors* que podrà aparèixer fins a un màxim de 10 vegades. El fet de que estiguin agrupats en una seqüència indica que els elements han d'aparèixer en ordre, és a dir, primer el títol, llavors els autors i per últim l'editorial. Els tres elements són de tipus *string*. L'atribut del llibre anomenat *Preu* és de tipus *double*.

Un cop explicat el llenguatge que s'ha utilitzat per a definir l'estructura dels exercicis i els motius pels quals s'ha escollit aquest llenguatge en concret, a continuació s'explicarà punt per punt els detalls del fitxer XML Schema que s'ha definit per als exercicis ACME.

### 7.1.2 Fitxer definició d'un exercici

En aquest apartat s'explica detalladament el contingut del fitxer XML Schema que conté la definició de l'estructura d'un exercici ACME.

Es comença el fitxer declarant la versió d'XML que es fa servir, així com l'element arrel del fitxer i una petita descripció del contingut del fitxer:

```

<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      This XSD file provide the definition of ACME exercise. Each exercise
      has firstly author, title that identifies it, a brief description
      of its purpose and the language in which is written. Then we define
      the different exercise parts. Each part could have one or more
      descriptors and its corresponding instructions and strategy to be
      evaluated the solution, which has been sent.
    </xs:documentation>
  </xs:annotation>

```

Figura 30: Capçalera fitxer definició d'un exercici

A continuació, es declara l'element base. S'anomenarà *Exercise* i es declara com un tipus complex que conté una seqüència de varis elements:

```

<xs:element name="exercise">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="author" type="xs:token" minOccurs="1"/>
      <xs:element name="title" type="xs:token" minOccurs="1"/>
      <xs:element name="description" type="xs:normalizedString" minOccurs="1"/>
      <xs:element name="language" type="xs:token" minOccurs="1"/>
      <xs:element name="display" default="vertical" minOccurs="1">
        <xs:simpleType>
          <xs:restriction base="xs:token" >
            <xs:enumeration value="vertical" />
            <xs:enumeration value="horitzontal" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="correction" default="total" minOccurs="1">
        <xs:simpleType>
          <xs:restriction base="xs:token" >
            <xs:enumeration value="incremental" />
            <xs:enumeration value="total" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="statement" type="xs:statement" minOccurs="0"/>
      <xs:element name="questions" type="questionList" minOccurs="1"/>
      <xs:element name="parameters" type="parameterList"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

Figura 31: Element Exercise

Com es pot apreciar, un exercici haurà de tenir un **autor**, un **títol**, una **descripció**, un **idioma**, un mètode de **visualització** i un mètode de **correcció**, com a elements simples. El mètode de visualització haurà de ser vertical o horitzontal, en canvi, el mètode de correcció haurà de coincidir amb incremental o total.

També, un exercici haurà de contenir, de manera opcional, un element tipus Statement, que es defineix més endavant, que correspondrà a l'**enunciat general**, un llistat de **preguntes** i un altre de **paràmetres**.

A continuació es defineix el tipus *questionList*, que conté un grup de *question* i que són declarades seguidament:

```
<xs:complexType name="questionList">
  <xs:group ref="question"/>
</xs:complexType>

<xs:group name="question">
  <xs:sequence>
    <xs:element name="statements" type="statementList" minOccurs="1"/>
    <xs:element name="files" type="associatedFiles" minOccurs="0"/>
    <xs:element name="solutions" type="solutionList" minOccurs="0"/>
  </xs:sequence>
</xs:group>
```

Figura 32: Element questionList

Una pregunta contindrà tres llistats: un de **enunciats**, un de **fitxers** associats, i un altre de **solucions**, que són definits a continuació.

```
<xs:complexType name="statementList">
  <xs:group ref="statement"/>
</xs:complexType>

<xs:group name="statement">
  <xs:sequence>
    <xs:element name="text" type="xs:string"/>
    <xs:element name="type" default="text">
      <xs:simpleType>
        <xs:restriction base="xs:token">
          <xs:enumeration value="latex"/>
          <xs:enumeration value="text"/>
          <xs:enumeration value="html"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:group>
```

Figura 33: Element statementList

Un **enunciat** haurà d'estar compost per un element tipus String anomenat **text** i un altre element anomenat **tipus** que podrà ser un dels següents valors: latex, text o html. Pel que fa als **fitxers**, hauran d'estar formats per un element anomenat **URL** i d'un altre element que podrà ser un dels valors del llistat: data, stats, image, vídeo, sound, graph o other.

```
<xs:complexType name="associatedFiles">
  <xs:group ref="file" />
</xs:complexType>

<xs:group name="file">
  <xs:sequence>
    <xs:element name="url" type="xs:anyURI"/>
    <xs:element name="type" default="data" >
      <xs:simpleType>
        <xs:restriction base="xs:token" >
          <xs:enumeration value="data" />
          <xs:enumeration value="stats" />
          <xs:enumeration value="image" />
          <xs:enumeration value="video" />
          <xs:enumeration value="sound" />
          <xs:enumeration value="graph" />
          <xs:enumeration value="other" />
        </xs:restriction >
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:group>
```

Figura 34: Element associatedFiles

Finalment, l'últim dels llistats que contindrà un element tipus Question, seran les **solucions**:

```
<xs:complexType name="solutionList">
  <xs:group ref="solution"/>
</xs:complexType>

<xs:group name="solution">
  <xs:sequence>
    <xs:element name="type" type="xs:unsignedInt"/>
    <xs:element name="correctorType" type="xs:unsignedInt"/>
    <xs:any />
  </xs:sequence>
</xs:group>
```

Figura 35: Element solutionList

Una solució estarà formada per un element anomenat type, de tipus enter, que contindrà el número de tipus de solució; un altre element de tipus enter que indicarà quin corrector



es farà servir per a corregir la pregunta; i finalment s'afegeixen els caràcters `<xs:any/>`. Aquesta marca indica que una solució pot contenir qualsevol altre contingut a continuació. Això fa que es puguin afegir els camps específics de cada tipus de solució sense cap tipus d'inconvenient en la validació de l'XML.

Cada tipus de solució tindrà el seu propi fitxer definició i a l'hora d'afegir el nou tipus de solució s'haurà de definir prèviament el fitxer XSD que contingui els camps específics per aquell tipus de solució.

Pel que fa als paràmetres:

```
<xs:complexType name="parameterList">
  <xs:group ref="parameter"/>
</xs:complexType>

<xs:group name="parameter">
  <xs:sequence>
    <xs:element name="number" type="xs:unsignedInt"/>
    <xs:element name="type" type="xs:string"/>
    <xs:any />
  </xs:sequence>
</xs:group>
```

Figura 36: Element parameterList

Un element de tipus paràmetre haurà de contenir un element de tipus enter corresponent al número de paràmetre, seguit d'un element tipus string que contindrà el tipus. Finalment, es torna a fer ús del camp `xs:any`, de cares a afegir nou contingut corresponent als dos tipus de paràmetres que es pretenen implementar.

Igual que a les solucions, cada tipus de paràmetre tindrà el seu propi fitxer definició, de manera que l'aplicació no estarà restringida a un sol tipus de paràmetre, sinó que se n'hi podran afegir a posteriori. Per afegir nous tipus de paràmetre s'haurà de definir prèviament el seu fitxer XSD que contingui els camps específics per aquell tipus de paràmetre.

Finalment, acaba la definició del fitxer amb la següent línia:

```
</xs:schema>
```

Figura 37: Acabament del fitxer definició d'un exercici

### 7.1.3 Altres fitxers definició

De cares a que la nova metodologia d'escriptura d'exercicis no sigui tancada i sigui extensible per a poder afegir nous tipus de solucions i paràmetres en un futur, s'han definit de manera separada els fitxers definició de cada tipus de solució i paràmetre implementats en aquest Projecte de Final de Carrera.

Aquests fitxers hauran de ser semblants al que s'ha explicat en el punt anterior i s'hi hauran de definir els atributs específics de cada tipus de solució i paràmetre.

Un exemple de fitxer definició d'un tipus de solució o paràmetre podria ser aquesta:

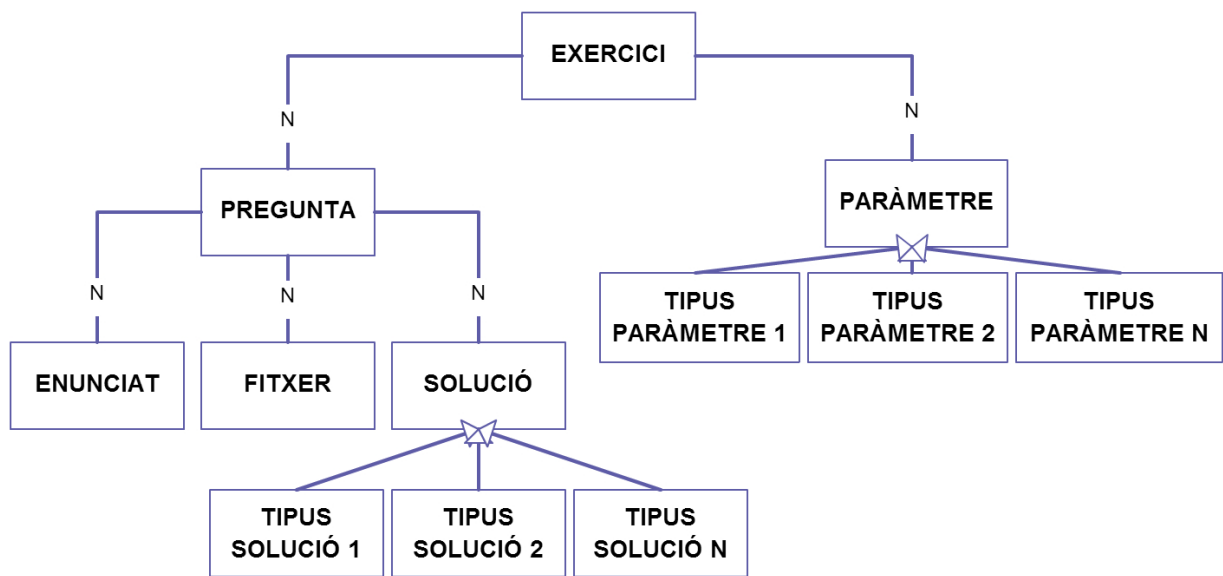
```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="content">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="text" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Figura 38: Exemple de fitxer definició d'un tipus de solució o paràmetre

En aquest cas, com a atributs d'aquest element únicament hi haurà un camp anomenat **text**, el contingut del qual haurà de ser de tipus cadena de text. Aquests fitxers hauran de ser guardats juntament amb el fitxer definició de l'exercici i s'agafaran automàticament pel verificador per cada tipologia de solució a l'hora de comprovar la correcta estructura del fitxer XML.

### 7.1.4 Estructura d'arbre del fitxer definició

Com a resum general de l'estructura del fitxer definició s'adjunta un diagrama on permet veure de manera clara com estan relacionats els diferents elements d'un exercici:



[online diagramming & design] [creately.com](https://creately.com)

Figura 39: Estructura d'arbre del fitxer definició

L'element arrel, l'exercici, disposa d'un conjunt de preguntes i paràmetres. Cada una de les preguntes disposa del seu corresponent llistat d'enunciats, fitxers i solucions. Com es pot apreciar, de les entitats solució i paràmetre en pegen entitats específiques que seran les diferents tipologies d'aquestes. És aquí on entra en joc el camp **xsd:any** que s'ha afegit al fitxer definició de l'exercici per a poder adherir els atributs específics de la tipologia sense que s'hagi de canviar l'estructura principal del fitxer definició de l'exercici. Això també està molt lligat amb el polimorfisme dels objectes PHP que es crearan més endavant.

## 7.2 Definició del model

Un cop especificada quina estructura ha de tenir un exercici, en aquest apartat es definirà en detall el model juntament amb tots els elements que el compondran. A partir del contingut del fitxer XML es crearà l'orientació a objectes corresponent a partir del següent diagrama de classes:

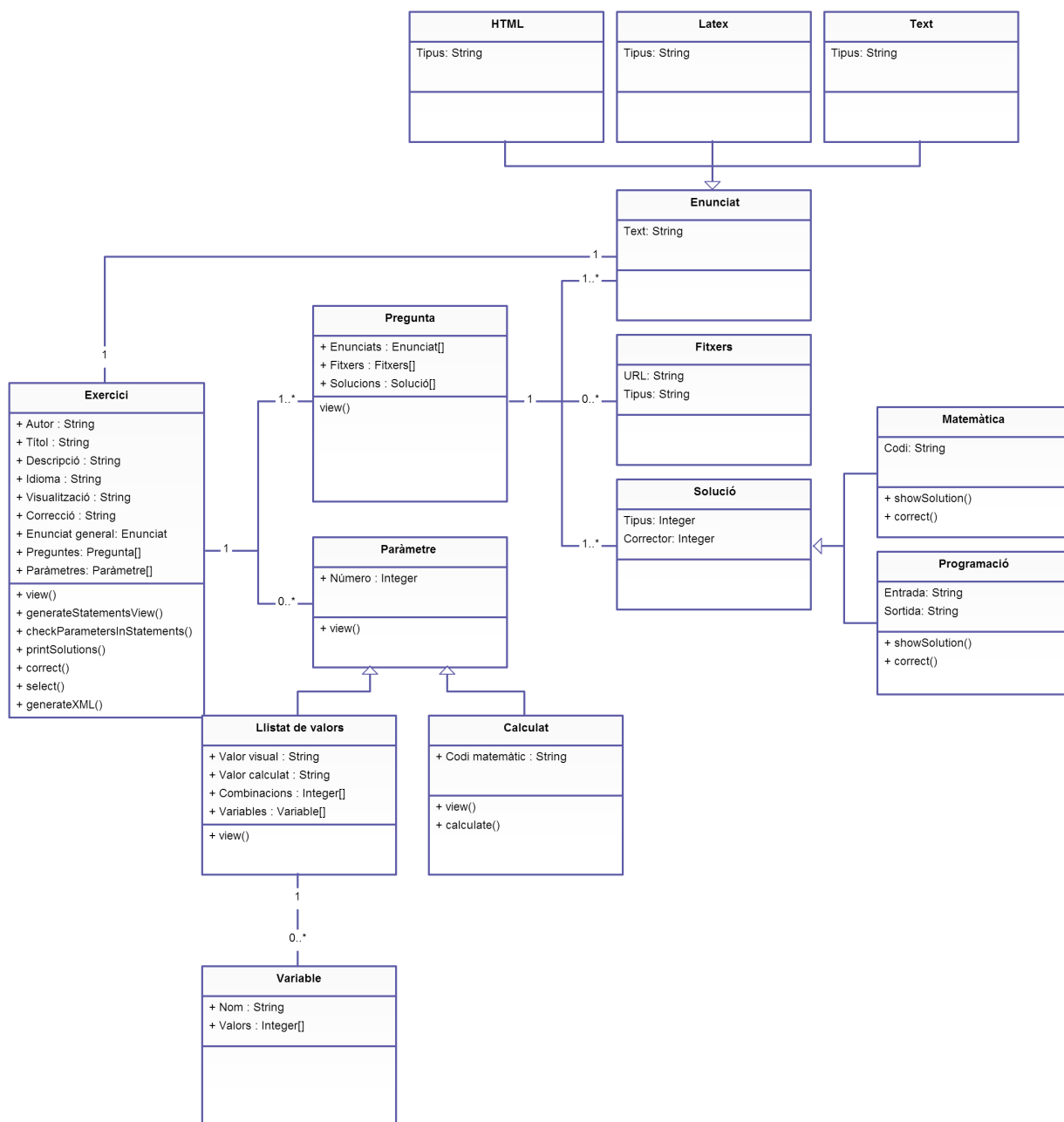


Figura 40: Diagrama de classes del model

L'objecte Exercici conté l'**autor** de l'exercici, un **títol** que l'identifica, una breu **descripció** i l'**idioma** amb el que està escrit. Els exercicis es poden mostrar i corregir de diferents

maneres, llavors tenen dos atributs que defineixen el tipus de **visualització** amb el que és mostra l'exercici i el tipus de **correcció** que s'aplica a l'hora d'avaluar les solucions enviades per l'alumne. Pel que fa a la **visualització**:

- **Vertical** mostrarà una pregunta sota l'altra.
- **Horitzontal** mostrarà una sola pregunta, sent necessari clicar el botó Següent per accedir a la propera pregunta.

Pel que fa a la **correcció**:

- **Incremental** aturarà el procés de correcció si una pregunta no és correcta.
- **Total** corregirà sempre totes les preguntes independentment del resultat de la seva correcció.

El resultat global de la correcció sempre serà el més negatiu de tots els resultats parcials. En cas de seleccionar visualització horitzontal i correcció incremental, l'alumne no podrà passar a la següent pregunta fins tenir resolta la que està fent.

De manera opcional, es podrà afegir un **enunciat general**. Un enunciat general correspondrà a un redactat que apareixerà sempre al capdamunt de l'exercici. Pot ser destinat a aportar informació genèrica que sigui necessària per a l'alumne per resoldre totes les preguntes.

Després, un exercici estarà compost per un llistat de preguntes i un altre de paràmetres. A continuació s'explica en detall cadascun dels elements que forma part d'un exercici.

### 7.2.1 Preguntes

Un exercici conté un llistat de preguntes on podrà haver-n'hi una o més d'una. El conjunt d'aquestes preguntes seran les activitats que haurà de solucionar l'alumne. Una pregunta està composta per un llistat d'enunciats, un de fitxers i un altre de **solucions**.

#### 7.2.1.1 Enunciats

Cada pregunta podrà tenir un o més enunciats. Els textos per plantejar la pregunta poden ser diferents per augmentar la variabilitat, encara que tots els textos han de tenir el mateix significat, per exemple: *Troba les següents integrals indefinides* i *Calcula una primitiva de les següents funcions*.

En enunciat podrà ser de tres tipologies diferents, totes elles estretament relacionades amb la classe pare Enunciat, que tindrà un atribut comú que serà la un camp de text on s'hi

podrà posar el text de l'enunciat. Dins del text de l'enunciat s'hi podran mostrar el valor dels paràmetres. Per a fer-ho, s'escriuran com a #P1, #P2... on N és el número del paràmetre. Els valors que poden prendre els paràmetres es defineixen al mateix paràmetre. Un paràmetre #PN es pot utilitzar les vegades que es vulgui en tot l'exercici, ja sigui en un mateix enunciat o en preguntes diferents. Ara bé, tots els paràmetres han de ser mostrats als enunciats com a mínim una vegada.

Les classes específiques tindran accés als atributs de la classe superior i disposaran d'un atribut específic per cada una que indicarà el tipus d'enunciat del qual es tracta. Els tipus d'enunciats que s'han implementat en aquest Projecte de Final de Carrera en són tres. S'ofereix la possibilitat de que els enunciats puguin ser escrits en diferents llenguatges, en HTML, L<sup>A</sup>T<sub>E</sub>X o simple Text. Les característiques de tots ells són explicades a continuació.

## HTML

L'enunciat s'escriu en codi HTML i s'interpretarà com a tal. No es respectaran els salts de línia. Per exemple, la redacció següent:

```
<p>
  Trobeu el volum d'un cos les seccions del qual per plans perpendiculars a l
  'eix de les <i>x</i> són <b>quadrats</b> de base continguda entre les
  corbes <b>x=y^2 i y=x^2</b> del pla <i>(x,y)</i>.
</p>
```

Figura 41: Exemple codi HTML

Es mostrarà de la següent manera:

Trobeu el volum d'un cos les seccions del qual per plans perpendiculars a l'eix de les  $x$  són **quadrats** de base continguda entre les corbes  $x = y^2$  i  $y = x^2$  del pla  $(x,y)$ .

## L<sup>A</sup>T<sub>E</sub>X

L'enunciat s'escriu en codi L<sup>A</sup>T<sub>E</sub>X. La longitud màxima que pot tenir l'enunciat és l'equivalent a un foli DIN-A4. A continuació es mostra un exemple d'enunciat amb tres matrius:

```
Calcula el quadre d'amortitzaci\''{o} i especifica :
  \begin{itemize}
    \item Import que representa la primera quota.
    \item Any despr\''{e}s de l'adquisici\''{o}
    \item Import que representa la darrera quota.
  \end{itemize}
Introdueix el resultat separat per comes i per ordre tal i com diu l'enunciat
.
```

Figura 42: Exemple codi L<sup>A</sup>T<sub>E</sub>X

Es mostrarà de la següent manera:

Calcula el quadre d'amortització i especifica:

- Import que representa la primera quota.
- Any després de l'adquisició
- Import que representa la darrera quota.

Introdueix el resultat separat per comes i per ordre tal i com diu l'enunciat.

Per a més informació sobre la sintaxis i semàntica del  $\text{\LaTeX}$ , es pot consultar a la wikipedia del llenguatge  $\text{\LaTeX}$  (<http://en.wikibooks.org/wiki/LaTeX>).

### Text

L'enunciat s'escriurà en text pla. En aquest cas es respectaran els salts de línia.

#### 7.2.1.2 Fitxers

Aquest apartat està destinat a afegir tots els fitxers que puguin anar adjunts a l'enunciat. Els atributs de cada fitxer són: la **URL** on estarà situat el fitxer i el **tipus** de fitxer que es tractarà. Es podrà adjuntar qualsevol tipus de fitxer: document, imatge, gràfic, vídeo, música, etc.

#### 7.2.1.3 Solucions

Cada pregunta pot tenir diverses solucions. Una solució defineix com l'alumne entrarà la resposta a la pregunta i com es corregirà aquesta. Amb la solució s'especifica el tipus de pregunta que es tracta. L'eina ha estat dissenyada perquè es puguin anar incorporant diferents tipus de solucions amb el pas del temps, de manera que en un futur, professors i projectistes puguin anar alimentant aquesta nova metodologia d'escriptura d'exercicis.

A l'ACME actualment existeixen diversos tipus de solució i depenent del tipus de solució, aquesta té uns atributs o altres que s'han de definir. És per això que s'ha dissenyat l'estructura de classes de manera que l'objecte Solució contingui atributs comuns per tots els tipus de solució i uns atributs específics que depenguin de cada tipus de solució. Els atributs comuns seran el **tipus de solució** i el **tipus de corrector**. Això va molt lligat amb l'estructura del fitxer definició de l'exercici on s'han inclòs camps oberts per a poder afegir aquests atributs específics.

En aquest Projecte de Final de Carrera s'ha implementat totalment el tipus de solucions més bàsic de l'ACME, la **solució expressió matemàtica**, i de manera parcial el tipus de solució **algorisme/programa informàtic**.

## SOLUCIÓ EXPRESSIÓ MATEMÀTICA

Pel que fa al primer tipus de solució implementada, l'alumne entrarà les seves solucions escrivint una expressió en una caixa de text. La pregunta serà corregida utilitzant el codi del corrector seleccionat, que serà l'únic atribut específic de la solució, anomenat **text**. En el codi de correcció, la solució que haurà entrat l'alumne es trobarà en una variable anomenada *SO*, que serà substituïda en el moment de la correcció. Dins el codi de correcció s'haurà d'escriure el resultat de la correcció amb paraules clau. Aquestes són *Correcte*, *Incompreensible* i *Incorrecte*.

Al ser un tipus de solució, s'ha de definir l'estructura que seguirà al fitxer XML per a poder identificar i verificar els atributs correctament. El fitxer definició d'aquesta tipologia de solució serà la següent:

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="content">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="text" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Figura 43: Fitxer definició de la tipologia solució expressió matemàtica

Com es pot comprovar, l'estructura defineix que l'element, de nom **content**, únicament conté un atribut anomenat **text**, que serà on s'hi posarà el codi de correcció, tal i com ja s'ha explicat.

## SOLUCIÓ ALGORISME/PROGRAMA INFORMÀTIC

Pel que fa al segon tipus de solució implementada de manera parcial en aquest Projecte de Final de Carrera, l'alumne enviarà un o varis fitxers amb el codi font del programa, que serà corregit amb el llenguatge de programació seleccionat als atributs comuns de la solució.

Per tal de corregir l'algorisme enviat per l'alumne s'han de definir un conjunt de proves d'execució. Per cada prova d'execució s'han de definir les entrades que volem aplicar a l'algorisme i escriure la sortida que s'espera rebre. L'entrada i sortida de la prova d'execució són els atributs específics d'aquest tipus de solució i seran anomenats *input* i *output*. Per afegir més proves d'execució, s'afegiran més solucions d'aquesta tipologia amb diferents valors pels atributs d'entrada i de sortida.



Exemple de prova d'execució:

### Entrada

```
0
10
1
10
```

Figura 44: Entrada de prova d'execució

### Sortida

```
ENTRAR DIES:
ENTRAR HORES:
ENTRAR MINUTS:
ENTRAR SEGONS:
TOTAL SEGONS: 36070
```

Figura 45: Sortida de prova d'execució

Com es pot comprovar en el quadre anterior, a la sortida només s'ha d'escriure el que s'espera que l'algorisme escrigui per pantalla. S'ha de tenir molta cura amb les majúscules, minúscules i els espais, perquè el corrector és molt estricte a l'hora de comparar les sortides.

Al ser un tipus de solució, s'ha de definir l'estructura que seguirà al fitxer XML per a poder identificar i verificar els atributs correctament. El fitxer definició d'aquesta tipologia de solució serà la següent:

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="content">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="input" type="xs:string"/>
        <xs:element name="output" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Figura 46: Fitxer definició de la tipologia algorisme/programa informàtic

Com es pot comprovar, l'estructura defineix que l'element, de nom **content**, contindrà dos atributs, un anomenat **input** i l'altre **output**, que serà on s'hi posaran les entrades i sortides pels jocs de proves de l'algorisme.

### 7.2.2 Paràmetres

Els paràmetres són l'eina principal per generar exercicis diferents pels alumnes, de manera que un alumne tingui un exercici amb les mateixes preguntes que el seu company però una mica diferents pel que fa a certs valors de l'enunciat.

De cada paràmetre s'ha de tenir un valor visible i un valor de correcció:

- El valor **visible** és el valor que substituirà l'etiqueta #PN en els enunciats i el que veurà l'alumne. Per escriure'l s'ha de tenir en compte el format d'escriptura a partir del tipus d'enunciat seleccionat.
- El valor **matemàtic** és el valor intern que s'utilitzarà per corregir l'exercici. Per aquest valor s'ha de tenir en compte el format a partir del tipus seleccionat en la solució de les preguntes.

Com a exemple, imaginem que un paràmetre sigui el terme d'un polinomi. Normalment se sol obviar el símbol \* de multiplicar i s'escriu  $xy$  en comptes de  $x * y$ . En aquest cas el valor visible seria  $xy$  i el valor matemàtic  $x * y$ , perquè el programa de correcció necessita el símbol de multiplicar.

L'ordre en què es defineixen els paràmetres és molt important perquè dins dels enunciats els paràmetres s'escriuen com #P1, #P2..., sent N el número de paràmetre.

L'estructura de les classes ha estat dissenyada de manera que es puguin anar afegint tipologies de paràmetres en un futur. En aquest Projecte de Final de Carrera s'han implementat dos tipus de paràmetres i el valor d'un paràmetre podrà ser escollit d'entre un llistat de valors o bé podrà ser un paràmetre calculat. A continuació, s'expliquen els detalls de cadascun.

#### 7.2.2.1 Llistat de valors

En aquest tipus de paràmetre, el valor d'aquest serà escollit a l'atzar d'entre un llistat de valors. Aquest tipus de paràmetre tindrà com a atributs el seu valor **visual**, el valor **calculat**, un conjunt de **variables** i **combinacions**. A continuació s'expliquen les característiques d'aquests dos últims atributs.

### VARIABLES

Un valor pot ser una expressió creada a partir de variables. En el valor visual i/o matemàtic del paràmetre s'haurà d'escriure el nom de la variable i llavors, s'hauran de definir els valors que podrà prendre aquesta variable. Una variable tindrà l'atribut **nom** que serà el seu identificador i podrà ser compost per paraules amb lletres minúscules o majúscules, i amb números. I també tindrà un altre atribut anomenat **valors** que contindrà tota la sèries de valors que podrà agafar la variable.

## COMBINACIONS

Hi ha la possibilitat de limitar els valors que poden ser seleccionats del següent paràmetre quan es triï un determinat valor del llistat de la valors del paràmetre que s'està definint. Introduint al camp **combinacions** la correlació de números separats per comes s'indicarà quins valors del següent paràmetre podran ser utilitzats a l'hora de sortejar. Si es deixa el camp en blanc, s'utilitzaran tots els valors del següent paràmetre indistintament.

### 7.2.2.2 Paràmetre calculat

L'altre tipus de paràmetre que s'ha implementat en aquest Projecte de Final de Carrera, el valor del paràmetre serà calculat en el moment d'assignar-li l'exercici a l'alumne, seguint el codi de generació que s'introdueixi al camp de text. Aquest serà l'únic atribut que tindrà aquest tipus de paràmetre.

Amb el codi introduït al camp de text es generarà el valor visible i el valor matemàtic del paràmetre. En el codi es poden utilitzar paràmetres anteriors per calcular aquests. Es podran incloure en el codi com a P1, P2, etc. I aquests seran substituïts pels valors corresponents abans de dur a terme el càlcul.

## 7.3 Editor d'exercicis

L'editor d'exercicis del Projecte ACME és una eina creada per facilitar la creació d'exercicis i dirigida als professors que s'encarreguen d'elaborar exercicis per l'ACME. Actualment ja existeix un editor que permet modificar els fitxers definició de nous exercicis per l'ACME. Primerament s'expliquen les característiques de l'editor que disposa l'ACME actualment, s'enumeraran els aspectes positius i inconvenients que té, i a continuació es proposaran els requeriments que haurà de tenir de l'editor que s'implementa en aquest Projecte de Final de Carrera.

### 7.3.1 Editor antic

L'editor que actualment disposa l'ACME per a la creació de nous exercicis es tracta d'una aplicació executable en Windows que ofereix totes i cadascuna de les funcionalitats de la metodologia d'escriptura d'exercicis tradicional, explicades en el punt 6 d'aquesta mateixa memòria. Amb aquesta aplicació es pot crear i modificar el codi dels fitxers definició dels exercicis de manera més còmode.

La pantalla principal de l'editor de textos és la finestra que s'utilitza per crear els diferents tipus d'exercicis. A partir d'aquesta pantalla, es pot accedir a totes les funcionalitats que proporciona l'editor:

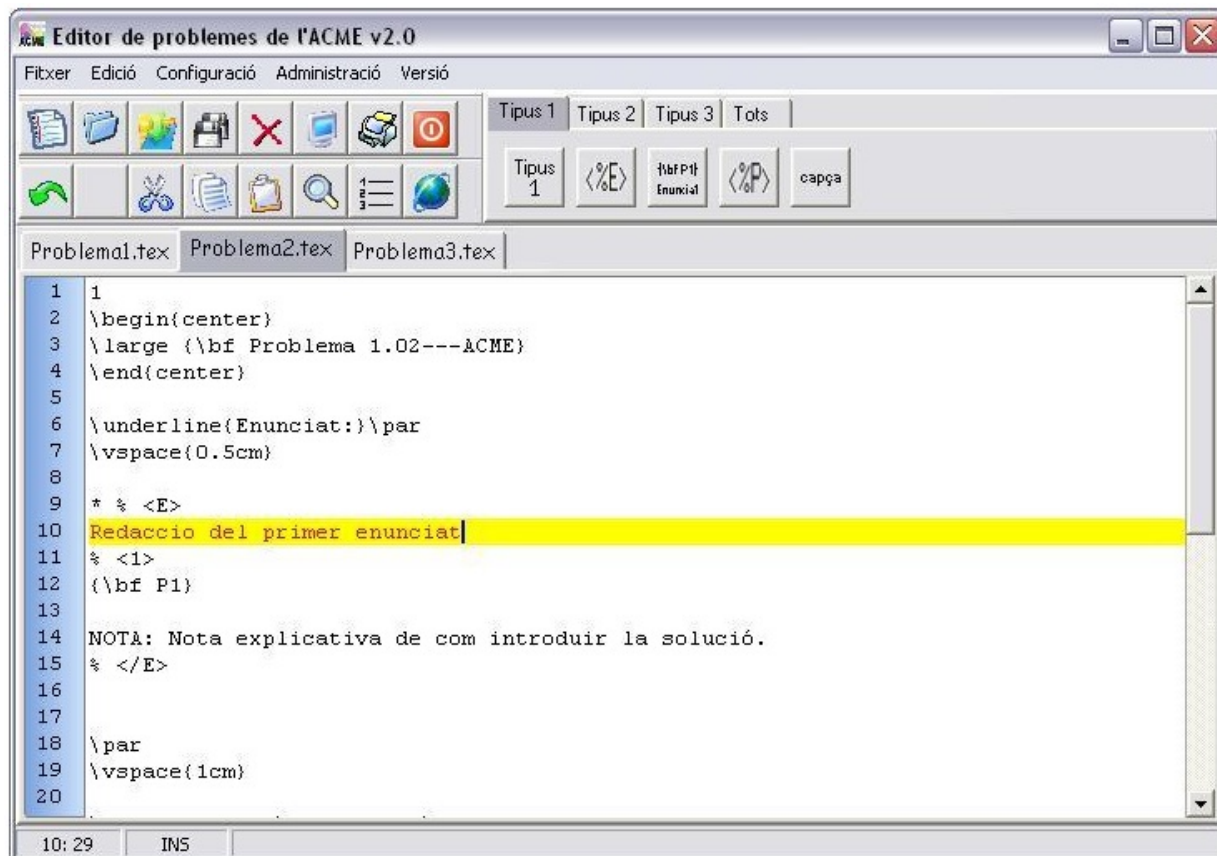


Figura 47: Captura de la pantalla principal de l'editor antic

El menú principal està situat a la part superior de la pantalla principal. El menú mostra un accés ràpid a totes les accions que disposa l'editor de problemes. Totes les funcionalitats estan agrupades en diferents grups lògics per tal de facilitar l'accés.

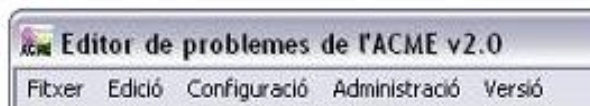


Figura 48: Captura de la barra principal de l'editor antic

La barra d'eines agrupa un conjunt de funcionalitats del menú principal del programa amb l'objectiu de disposar d'un accés ràpid a les funcionalitats més importants de l'aplicació i també les funcionalitats que s'utilitzen més cops. La barra d'eines es divideix en dos grups, la barra superior agrupa les funcionalitats dedicades a fitxer i la barra inferior agrupa les accions dedicades a l'edició del text dels problemes.

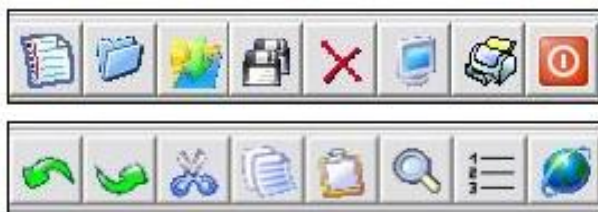


Figura 49: Captura de la barra d'eines de l'editor antic

La barra de problemes agrupa el conjunt de components en els diversos tipus de problemes que s'han configurat:

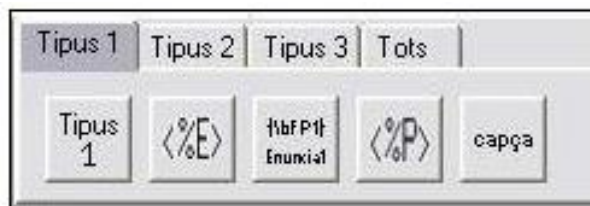
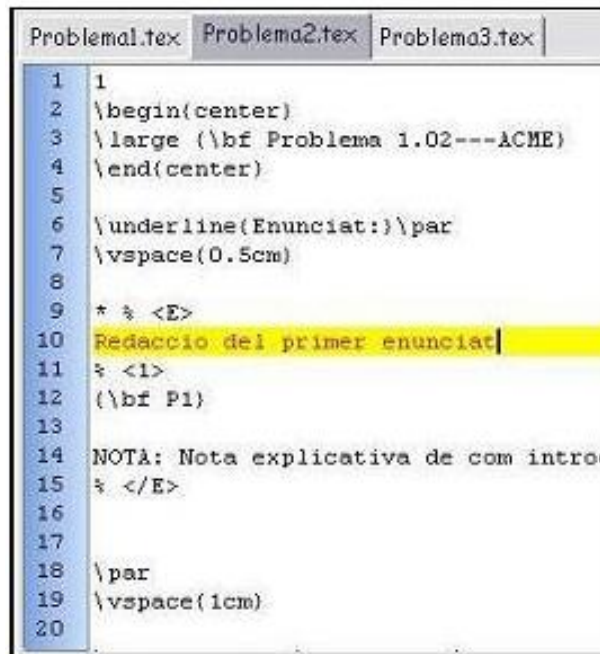


Figura 50: Captura de la barra de problemes de l'editor antic

La barra està formada per un conjunt de pestanyes on cada una representa un tipus de problema. Selecciónant una pestanya, automàticament es mostren les components que té configurades aquest tipus de problema.

La zona de treball és la part de l'editor de problemes on s'escriu el text dels problemes. Totes les funcionalitats de l'aplicació van dirigides al text de la zona de treball:



```
1 1
2 \begin{center}
3 \large (\bf Problema 1.02---ACME)
4 \end{center}
5
6 \underline{Enunciat:}\par
7 \vspace{0.5cm}
8
9 * % <E>
10 Redaccio del primer enunciat|
11 % <1>
12 (\bf P1)
13
14 NOTA: Nota explicativa de com intro
15 % </E>
16
17
18 \par
19 \vspace{1cm}
20
```

Figura 51: Captura de la zona de treball de l'editor antic

La zona de treball està formada per un conjunt de pestanyes on cada una representa un problema o fitxer. Aquesta part incorpora un marge esquerre on apareixen els números de línia i ressalta la línia activa.

Un cop escrit el fitxer definició del problema, s'haurà de verificar la correcta escriptura de l'exercici. Per a fer això, existeix la pantalla de verificació, que s'utilitza per enviar l'exercici que s'està editant a l'ACME per realitzar les accions de verificació, visualització i testeig:



Usuari	prodem1
Clau	*****
Acció	Verificació
Fitxer	C:\ACMEeditor\Docs\10p13n1.tex
URL de connexió	https://acme.udg.es/editor/verificador.php
Ajuda	No

Aceptar Cancelar Configuració

Figura 52: Captura de la pantalla de connexió de l'editor antic

En aquesta pantalla cal omplir les dades de l'enviament ja que cal entrar dins l'ACME per poder realitzar aquestes accions. Això indica que cal estar donat d'alta a l'ACME i disposar d'un nom d'usuari i contrasenya. Si aquestes dades no són correctes, les accions no podran ser realitzades. Les accions disponibles són verificació, visualització i testeig. Cadascuna d'aquestes portaran a una pantalla diferent que durà a terme l'acció corresponent.

Si l'ACME verifica correctament l'exercici que s'ha escrit apareixerà una pantalla informant que l'exercici s'ha verificat de manera correcta. Altrament ens mostrarà el llistat d'errors que ha detectat al fitxer definició de l'exercici, que s'hauran de corregir fins a obtenir la pantalla següent:



Figura 53: Captura de la pantalla de verificació de l'editor antic

Un cop verificat correctament el problema, també es pot mostrar-ne un enunciat d'exemple. Per a fer això l'editor ens proporciona un formulari per escollir quin valor es vol que agafin els paràmetres:

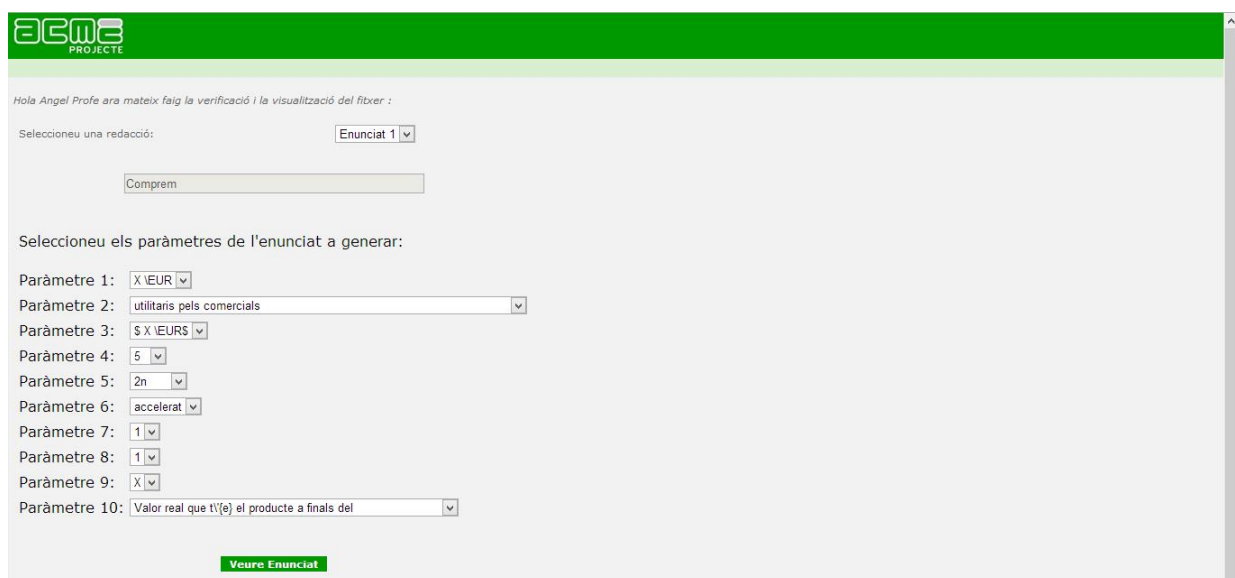


Figura 54: Captura de la pantalla d'elecció de paràmetres de l'editor antic

Un cop escollits els valors desitjats per cada paràmetre, es prem el botó Veure enunciat i ens durà a la següent pantalla, generant la imatge corresponent de l'enunciat amb els valors assignats als paràmetres tal i com s'ha indicat anteriorment:

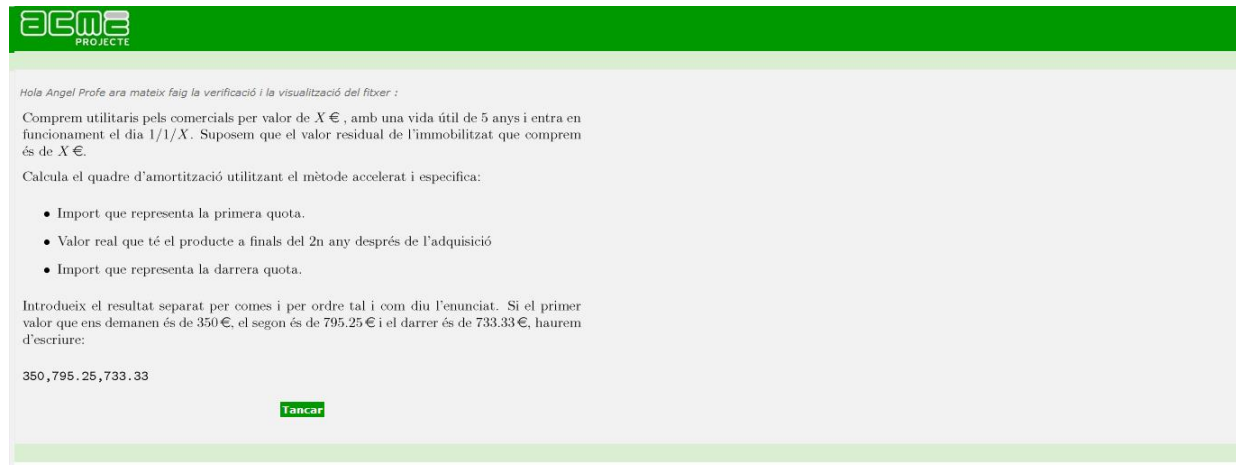


Figura 55: Captura de la pantalla de visualització d'un enunciat de l'editor antic

D'aquesta manera es pot veure un enunciat d'exemple de l'exercici que s'acaba de redactar.

I finalment una altra de les coses que permet fer l'editor de l'ACME és testejar el codi corrector de l'exercici. Per a fer això, permet assignar als paràmetres uns valors en concret i provar d'introduir una solució al codi de correcció per veure els resultats que dona. Ensenyarà el mateix formulari que a la visualització però amb la característica afegida d'un camp extra en el que s'haurà d'entrar la solució a l'exercici:

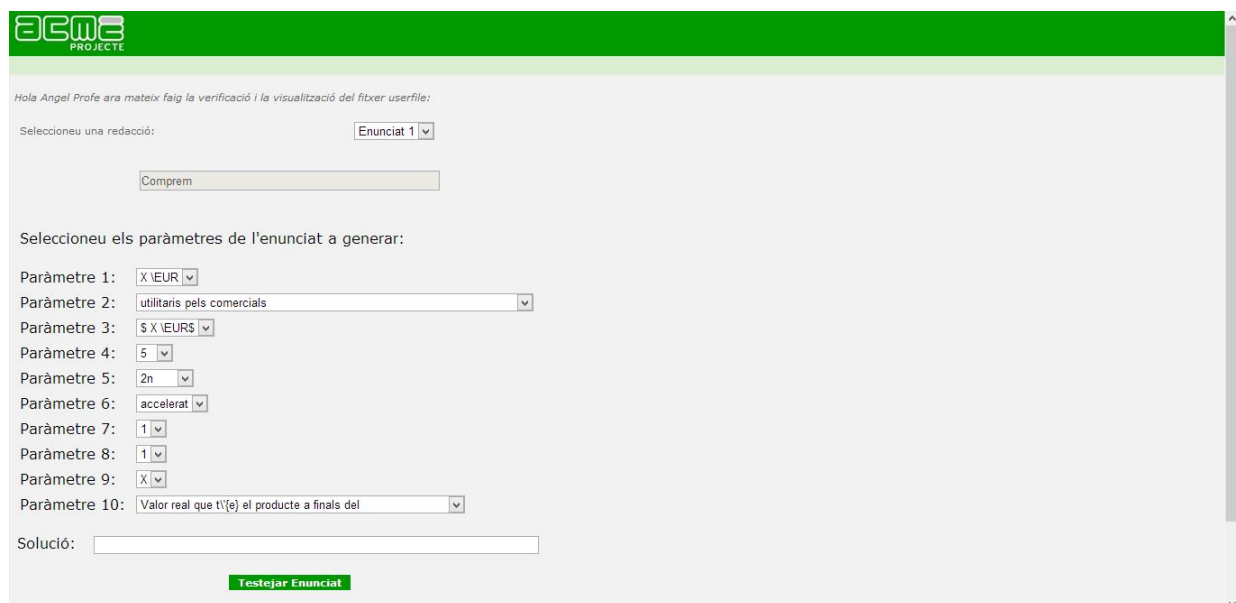


Figura 56: Captura de la pantalla de testeig de l'editor antic



Un cop escollit el valor dels paràmetres i entrada la solució, es pot fer clic al botó Testejar enunciat i executarà el codi de correcció amb els valors introduïts a cada paràmetre i la solució entrada. A la següent pantalla, mostrarà el codi Mathematica de correcció i al final de tot els resultats de corregir l'exercici:

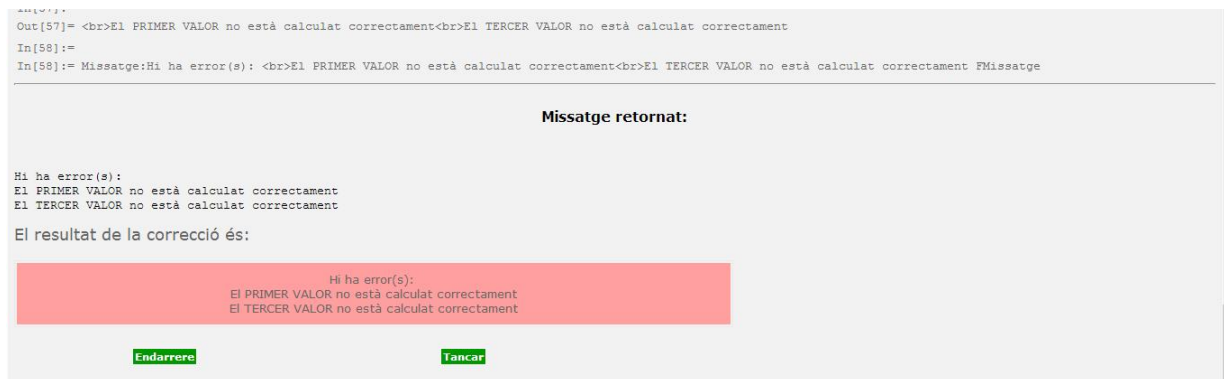


Figura 57: Captura de la pantalla de resultat de l'editor antic

Com a resum de l'anàlisi, s'enumeren els aspectes positius que caldrien mantenir a l'hora de definir el nou editor, així com també els punts febles que s'han detectat en aquest editor antic:

### Aspectes positius:

- Permet **verificar** si un exercici està correctament escrit. El nou editor també ha d'incorporar aquesta funcionalitat, permetent comprovar de manera automàtica si un exercici ha estat ben creat.
- Permet **visualitzar** un exercici d'exemple sortejat. És positiu pel professor que crea el nou exercici poder veure com quedaran visualitzats els enunciats que prèviament ha redactat.
- Permet **testejar** el codi de correcció de l'exercici, permetent escollir els valors per cada paràmetre i entrar una solució de forma manual.

### Aspectes negatius:

- Només pot ser executat en Windows. Usuaris en Linux i altres sistemes operatius no poden fer ús de l'editor.
- És un editor purament de fitxer de codi. Té poques funcionalitats més que un editor de text normal i corrent.
- Es modifica l'exercici en l'aplicació i per efectuar la verificació i posterior visualització es fa en una pantalla externa.

### 7.3.2 Nou editor d'exercicis

Un cop analitzat l'editor actual de l'ACME i vistos els seus aspectes positius i els seus punts febles, és hora d'establir els requeriments que ha de tenir el nou editor d'exercicis:

- El nou editor ha de permetre afegir i treure de manera dinàmica enunciats, paràmetres i demés elements de l'exercici.
- S'ha de poder executar en qualsevol sistema operatiu.
- Ha de ser un editor molt fàcil i intuïtiu de fer servir.
- Ha de tenir l'opció de descarregar-se el fitxer definició de l'exercici creat actualment per poder seguir modificant-lo a posteriori.
- S'ha de poder verificar i visualitzar l'exercici sense haver de canviar de pantalla.

L'editor està creat en forma de formulari web per tal de facilitar la tasca d'afegir, modificar i eliminar les diferents parts que componen un exercici de manera intuïtiva i sense haver de preocupar-se per l'estructura interna del format amb el que es guarda el fitxer definició de l'exercici, a diferència de la metodologia tradicional d'escriptura d'exercicis.

Com s'ha pogut veure, l'estructura d'un exercici pot constar d'una o més preguntes i d'un o més paràmetres. Cada pregunta pot tenir varis enunciats i vàries solucions. Els paràmetres es defineixen a part i corresponen a tot l'exercici, de manera que es pot utilitzar un mateix paràmetre en una o més preguntes. Opcionalment, un exercici pot tenir un enunciat general que sempre serà visible. Sota aquestes premisses bàsiques s'haurà de dissenyar el nou editor d'exercicis.

#### 7.3.2.1 Pantalla inicial

A l'iniciar l'aplicació, l'editor ofereix dues possibilitats: començar un exercici completament nou o carregar un exercici ja existent, pujant el fitxer XML que contingui la seva definició:



Figura 58: Captura de la pantalla inicial del nou editor

Si s'escull Crear un exercici a partir d'un ja existent apareixerà un camp d'entrada mitjançant el qual es podrà pujar un arxiu XML amb la definició de l'exercici i posteriorment carregarà totes les seves dades per poder seguir editant-lo:

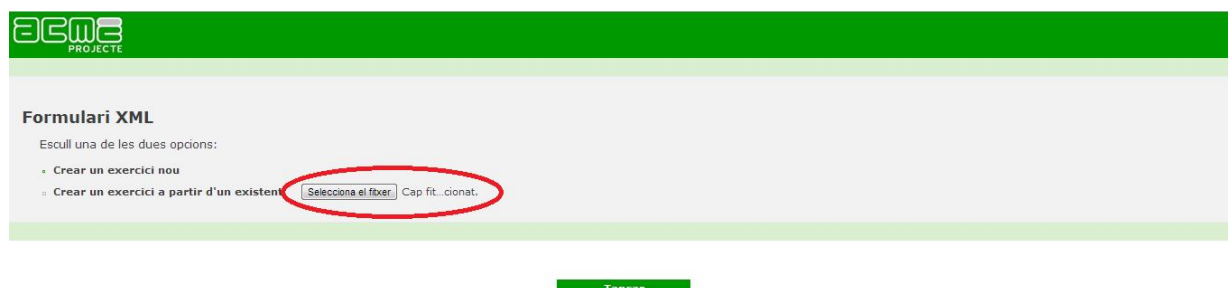


Figura 59: Captura de la pantalla inicial del nou editor seleccionant fitxer existent

Altrament, si s'escull l'opció Crear un exercici nou, redirigirà l'usuari cap a la pantalla principal de l'editor on es podrà començar a crear el nou exercici:

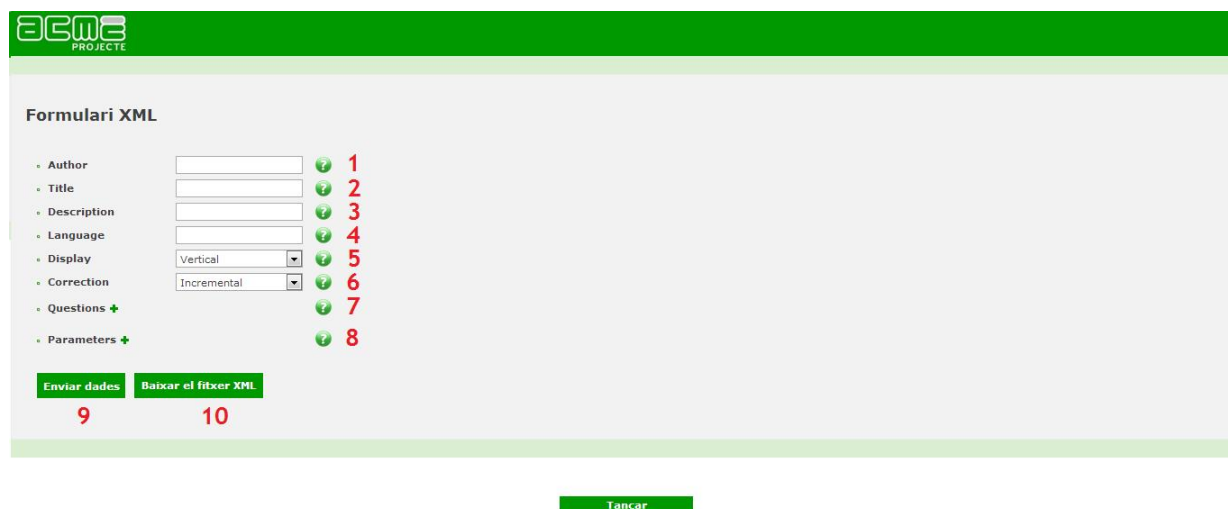


Figura 60: Captura de la pantalla principal del nou editor

1. Camp on s'introduirà l'autor de l'exercici
2. Camp on s'introduirà el títol de l'exercici
3. Camp on s'introduirà la descripció de l'exercici
4. Camp on s'introduirà l'idioma amb el que està escrit l'exercici
5. Camp on es seleccionarà la visualització de l'exercici

6. Camp on es seleccionarà el tipus de correcció de l'exercici
7. Camp on s'afegiran totes les preguntes que contindrà l'exercici
8. Camp on s'afegiran tots els paràmetres que tindrà l'exercici
9. Botó per a verificar i visualitzar l'exercici.
10. Botó per a descarregar-se el fitxer XML amb la definició de l'exercici.

### 7.3.2.2 Visualització i correcció

Els exercicis com que es poden mostrar i corregir de diferents maneres, s'ha habilitat un camp per escollir el tipus de visualització i correcció de l'exercici. Les opcions que hi ha per la visualització són:

- Vertical que mostrarà una pregunta sota l'altra.
- Horitzontal que mostrarà una sola pregunta, sent necessari clicar el botó Següent per accedir a la propera pregunta.

Pel que fa a la correcció:

- Incremental que aturarà el procés de correcció si una pregunta no és correcta.
- Total que corregirà sempre totes les preguntes independentment del resultat de la seva correcció.


### 7.3.2.3 Preguntes

Es poden afegir preguntes a l'exercici amb el botó **+** que hi ha al costat del títol Questions. Quan fem clic a aquest botó apareixeran nous camps per omplir corresponents a la nova pregunta:



Figura 61: Captura de la pantalla de preguntes del nou editor

Una pregunta està composta per un enunciat general (opcional), i un llistat d'enunciats, fitxers i solucions.

Per eliminar una pregunta o un paràmetre, es pot fer clicant els botons  que hi ha al costat de cada una. Només es podrà eliminar l'última pregunta d'entre totes les que hi ha entrades.

### 7.3.2.4 Enunciats

Cada pregunta podrà tenir un o més enunciats. Cada enunciat tindrà dos atributs: un camp de text on s'hi podrà posar el text de l'enunciat i un altre camp d'entrada on permetrà seleccionar el tipus d'enunciat.

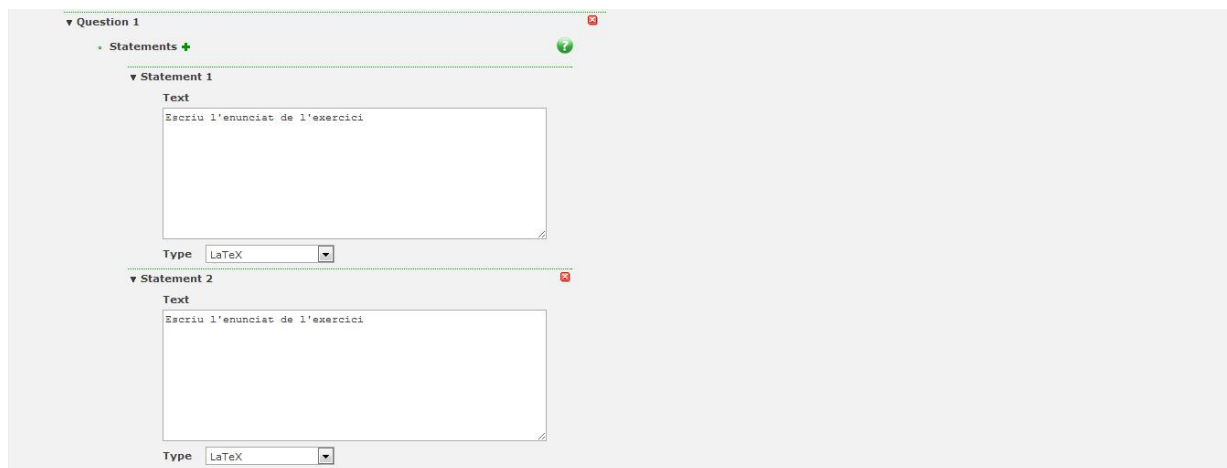


Figura 62: Captura de la pantalla d'enunciats del nou editor

Dins del text de l'enunciat s'hi podran mostrar el valor dels paràmetres. Per a fer-ho, s'escriuran com a #P1, #P2... on N és el número del paràmetre. Els valors que poden prendre els paràmetres es defineixen al seu propi apartat del formulari. Un paràmetre #PN es pot utilitzar les vegades que es vulgui en tot l'exercici, ja sigui en un mateix enunciat o en preguntes diferents. Ara bé, l'editor obligarà mostrar com a mínim una vegada cada paràmetre en algun dels enunciats declarats.

### 7.3.2.5 Enunciat general


De manera opcional, es podrà afegir un enunciat general. Es podrà fer fent clic al botó  del costat de General Statement. A continuació es desplegaran els mateixos camps que un enunciat normal i corrent. Un camp de text on s'hi podrà posar el text de l'enunciat i un altre camp d'entrada on permetrà seleccionar el tipus d'enunciat:  $\LaTeX$ , HTML i Text.



Figura 63: Captura de la pantalla d'enunciat general del nou editor

### 7.3.2.6 Fitxers

Aquest apartat està destinat a afegir tots els fitxers que puguin anar adjunts a l'enunciat. Es podrà fer fent clic al botó **+** del costat de Files. Els atributs de cada fitxer són: la URL on estarà situat el fitxer i el tipus de fitxer que es tractarà. Es podrà adjuntar qualsevol tipus de fitxer: document, imatge, gràfic, vídeo, música, etc.

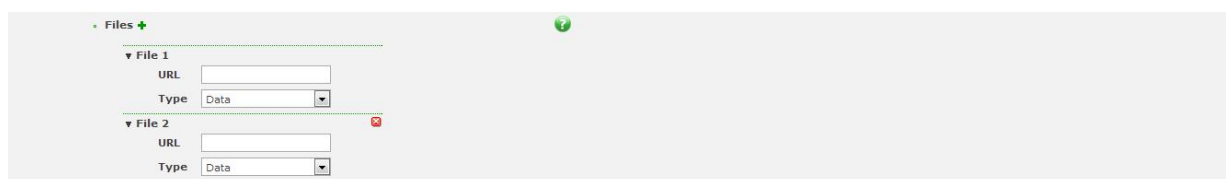


Figura 64: Captura de la pantalla de fitxers del nou editor

### 7.3.2.7 Solucions

Cada pregunta pot tenir varies solucions. A l'ACME actualment existeixen diversos tipus de solució i depenent del tipus de solució, aquesta té uns atributs o altres que s'han de definir. És per això que s'han estructurat les solucions de manera que continguin atributs comuns per tots els tipus de solució i uns atributs específics que depenguin del tipus de solució. Els atributs comuns seran el tipus de solució i el tipus de corrector. Per emmagatzemar els tipus de solució i correctors existents, s'han hagut de crear unes taules SQL a la base de dades de l'ACME:

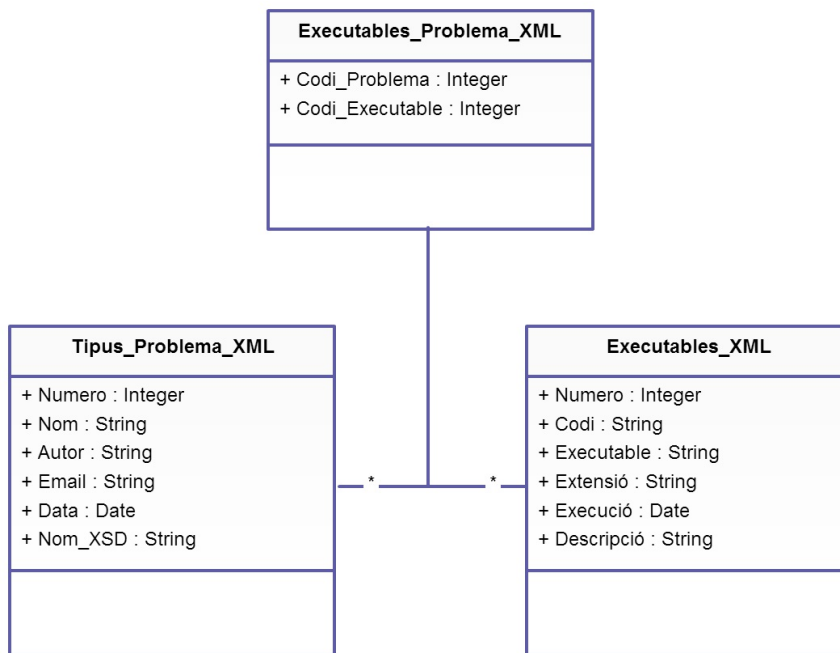


Figura 65: Diagrama de classes de les noves taules SQL

La taula Tipus\_Problema\_XML emmagatzemarà totes les tipologies de solucions que estaran disponibles en la nova metodologia d'escriptura d'exercicis. Com a atributs tindrà el número que l'identificarà, un nom que correspondrà a una descripció ràpida del tipus de solució, el nom de l'autor que l'hagi creat, el seu correu electrònic, la data en què ha estat creada i el més important dels camps, un camp anomenat nom\_xsd que contindrà el nom del fitxer definició de la solució.

Com s'ha explicat en punts anteriors d'aquesta memòria, per cada tipologia s'ha hagut de dissenyar un XML Schema amb els elements o camps específics que componen l'estructura de cada tipus de solució. Associant el tipus de solució amb el nom del seu fitxer definició corresponent és quan es podrà fer la verificació de la correcta construcció del fitxer XML sense que s'hagi de canviar l'estructura principal del fitxer definició de l'exercici.

Com ja s'ha dit anteriorment, en aquest Projecte de Final de Carrera s'ha implementat totalment el tipus de solucions més bàsic de l'ACME, la solució expressió matemàtica, i de manera parcial el tipus de solució algorisme/programa informàtic.

Pel que fa a la taula d'Executables\_XML, la seva funció principal serà emmagatzemar els diferents correctors mitjançant els quals l'ACME podrà desenvolupar les tasques de correcció d'exercicis. Els atributs seran un numero que l'identifiqui, un codi i una descripció mitjançant els quals puguin ser identificats ràpidament en l'editor i executable, extensió i execució, atributs destinats a emmagatzemar les rutes dels executables i els diferents paràmetres que s'han d'afegir per a poder executar correctament.

En aquestes dues tipologies de solucions que s'han implementat en aquest Projecte de Final de Carrera s'hi ha afegit un total de 4 correctors diferents, dos per a la solució matemàtica

(Mathematica i Maxima) i els altres dos per a la solució de programació (C++ i Java).

I finalment, referent a la tercera taula creada, la d'Executables\_Problema\_XML, ha estat creada producte d'aquesta relació molts a molts i serveix per associar quins executables van amb cada tipologia de solució.

## SOLUCIÓ EXPRESSIÓ MATEMÀTICA

Per a crear una solució d'aquesta tipologia, s'haurà de seleccionar el tipus de solució 1- Expressió matemàtica i el corrector que es desitgi utilitzar. Un cop seleccionat aquest tipus de solució, es carregaran automàticament els camps específics d'aquest tipus de solució. Per aquesta tipologia, l'únic camp que apareixerà al seleccionar aquest tipus de solució serà una caixa on s'hi podrà afegir el codi de correcció que utilitzarà l'ACME a l'hora d'avaluar la solució enviada per l'alumne:



Figura 66: Captura de la pantalla de solució expressió matemàtica del nou editor

Les regles i sintaxi de cada corrector poden ser diferents, però en tots els codis de correcció hi haurà d'haver el següent:

- La solució que entri l'alumne es trobarà a una variable anomenada SO, que serà substituïda en el moment de la correcció.
- S'ha d'escriure el resultat de la correcció amb paraules clau. Aquestes són Correcte, Incomprensible i Incorrecte.

## SOLUCIÓ ALGORISME/PROGRAMA INFORMÀTIC

Pel que fa al segon tipus de solució implementada de manera parcial en aquest Projecte de Final de Carrera, l'alumne enviarà un o varis fitxers amb el codi font del programa, que serà corregit amb el llenguatge de programació seleccionat als atributs comuns de la solució. Per la qual cosa, a part dels camps comuns de les solucions, al seleccionar l'opció 2 – Algorisme/Programa informàtic, canviarà el contingut del camp de selecció del corrector i s'hi carregaran els compiladors disponibles. A més també es carregaran els atributs específics d'aquest tipus de solució. En aquest cas n'hi haurà dos: un anomenat input, que com ja s'ha explicat servirà per posar-hi l'entrada de l'algorisme; i un altre anomenat output on s'hi posarà la sortida esperada després de l'execució de l'algorisme amb l'entrada especificada anteriorment.



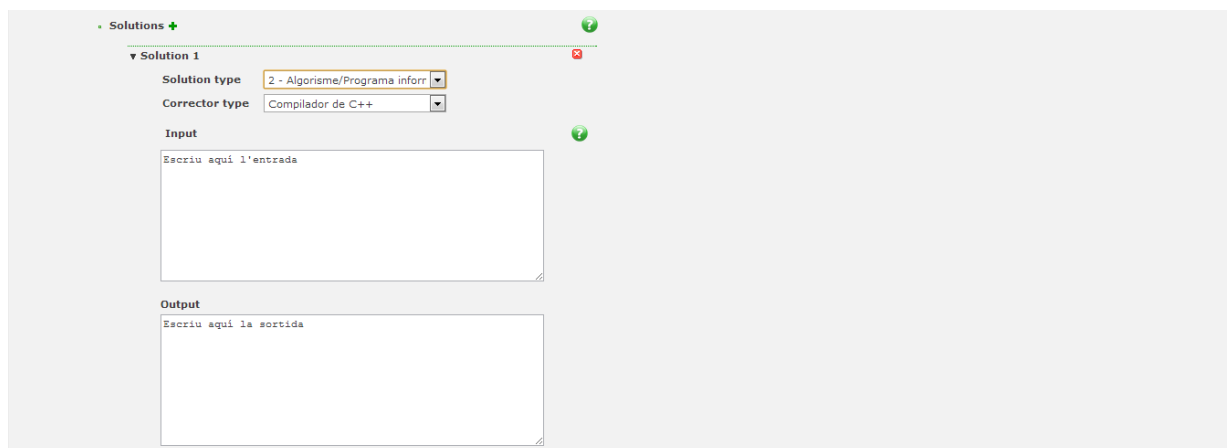


Figura 67: Captura de la pantalla de solució algorisme/programa informàtic del nou editor

Per afegir més d'una prova d'execució, s'hauran d'afegir més solucions d'aquesta tipologia al formulari, mitjançant el botó **+** del costat de Solutions. D'aquesta manera es podran afegir tants tests d'execució com es cregui convenient per avaluar l'algorisme enviat per l'alumne.

### 7.3.2.8 Paràmetres

Es poden afegir paràmetres a l'exercici amb el botó **+** que hi ha al costat del títol Parameters. Per eliminar un paràmetre, es pot fer clicant els botons **✖** que hi ha al costat de cada un. Només es podrà eliminar l'últim paràmetre d'entre tots els que hi ha entrats.

L'ordre en què es defineixen els paràmetres és molt important perquè dins dels enunciats els paràmetres s'escriuen com #P1, #P2..., sent N el número de paràmetre.

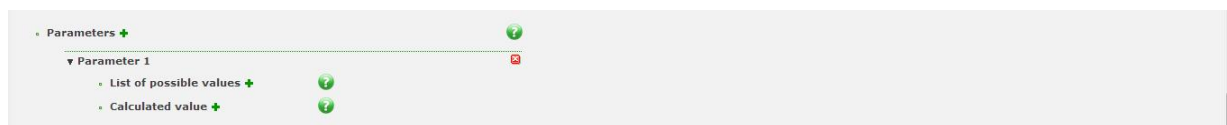


Figura 68: Captura de la pantalla de paràmetres del nou editor

Al afegir un nou paràmetre, demana quin és el tipus de paràmetre que es vol afegir. Les opcions disponibles són llistat de valors o paràmetre calculat. A continuació, s'expliquen els detalls de cadascun.

## LLISTAT DE VALORS

Si s'escull la primera opció List of possible values, el valor del paràmetre serà escollit a l'atzar d'entre un llistat de valors. Al clicar aquesta opció, apareixeran els següents camps per omplir:

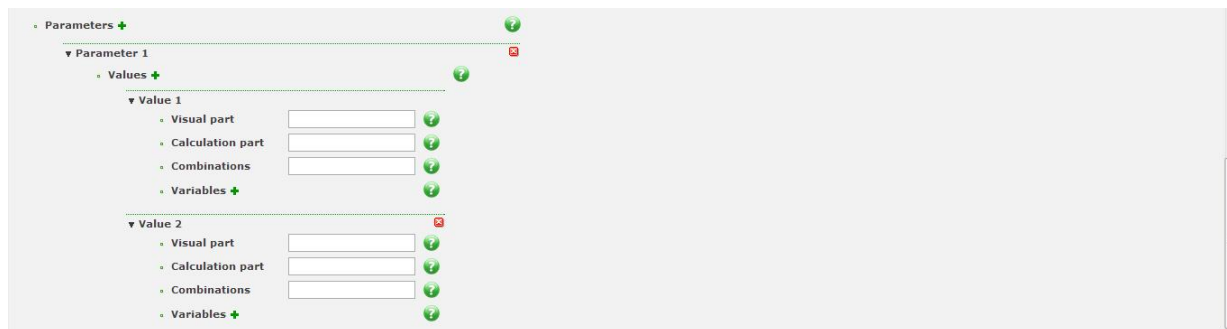


Figura 69: Captura de la pantalla de llistat de valors del nou editor

S'han d'afegir tots els valors entre els que s'escollirà el paràmetre. Per afegir nous valors es pot fer clicant el símbol **+** on diu Values de manera que aniran apareixent un sota l'altre. Per eliminar-ne, es podrà fer fent clic al símbol **✖**. Només es podrà eliminar l'últim valor d'entre tots els que hi ha entrats.

Per cada valor s'hauran de definir el valor visible i el valor matemàtic als dos camps de text corresponents. També es podrà fer ús dels atributs de variables i combinacions.

## Variables

Com ja s'ha explicat, un valor pot ser una expressió creada a partir de variables. En el valor visual i/o matemàtic s'ha d'escriure el nom de la variable amb el caràcter **#** al davant. Llavors, a l'apartat de variables s'hauran de definir els valors que podrà prendre aquesta variable. Per a fer això s'haurà de clicar el símbol del costat de Variables:

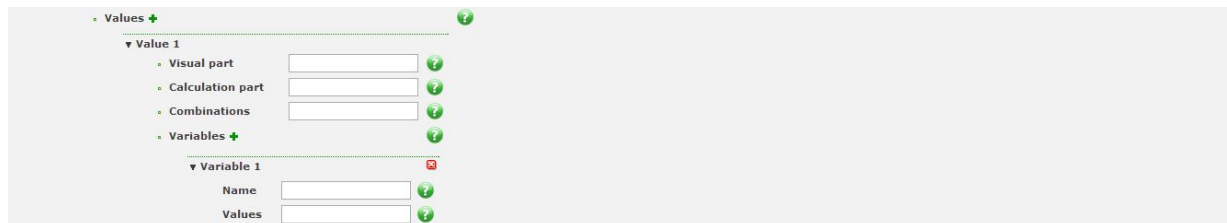


Figura 70: Captura de la pantalla de variables del nou editor

A l'atribut Name s'hi haurà de posar el nom de la variable sense el **#** al davant. En canvi, al camp de text corresponent Values s'hi afegiran el llistat de valors que podrà prendre la variable, separats de comes.

Imaginem que tenim una expressió  $Ax-By$  on  $A$  és una variable que pot prendre els valors 2, 3 i 4, i  $B$  és una altra variable que pot prendre els valors  $\frac{1}{2}$ ,  $\frac{3}{4}$  i 7. S'hauria d'omplir el formulari de la següent manera:

Figura 71: Exemple de variables del nou editor

Els noms de les variables poden ser paraules amb lletres minúscules o majúscules, i amb números. Com es pot apreciar, pel valor visible ha estat necessari un espai en blanc entre la variable  $\#A$  i la incògnita  $x$ . Altrament, l'interpret hagués interpretat que el nom de la variable era  $\#Ax$ .

## Combinacions

Hi ha la possibilitat de limitar els valors que poden ser seleccionats del següent paràmetre quan es triï un determinat valor de la llista de la valors del paràmetre que s'està definint. Introduint al camp Combinacions la correlació de números separats per comes s'indicarà quins valors del següent paràmetre podran ser utilitzats a l'hora de sortejar. Si es deixa el camp en blanc, s'utilitzaran tots els valors del següent paràmetre indistintament.

Figura 72: Exemple de combinacions del nou editor

En l'exemple anterior, el paràmetre 1 únicament podrà ser combinat amb el primer i el segon valor del segon paràmetre. D'aquesta manera, el tercer valor del segon paràmetre mai podrà ser escollit ja que l'apartat de combinacions restringeix el seu ús.

### PARÀMETRE CALCULAT

En canvi, si s'escull la segona opció Calculated values, el paràmetre serà calculat en el moment d'assignar-li l'exercici a l'alumne, seguint el codi de generació que s'introdueixi al camp de text:

Amb el codi introduït al camp de text s'ha de generar el valor visible i el valor matemàtic del paràmetre. Aquests dos valors cal que estiguin separats pels caràcters # perquè l'AC-ME ho pugui capturar.

Per exemple, el codi per tenir un paràmetre amb valor de correcció  $x/2$  que s'hagués de substituir en un enunciat tipus L<sup>A</sup>T<sub>E</sub>Xo Math, hauria de generar #\frac{x}{2} # X/2 #, de manera que el primer valor seria el visible i el segon el matemàtic.

En el codi es poden utilitzar paràmetres anteriors per calcular aquests. S'han d'incloure en el codi com a P1, P2, etc. I aquests seran substituïts pels valors corresponents abans de dur a terme el càlcul.

El codi Mathematica en aquest cas, sempre haurà de ser d'aquest estil:

```
...
Print["#" <> ToString[valor_visual] <> "#" <> ToString[valor_calculat] <> "#"];
```

Figura 73: Patró de codi Mathematica en paràmetre calculat

#### 7.3.2.9 Enviar dades

Amb el botó Enviar dades es pot comprovar com es veurà l'exercici. Per la visualització prèvia de l'exercici es mostraran els enunciats de les diferents preguntes una sota l'altre.



Figura 74: Exemple de visualització en el nou editor

### 7.3.2.10 Missatges d'error

Quan es vol visualitzar l'exercici es comprova si el format de l'exercici és correcte i conté les parts necessàries. Si l'exercici és incorrecte aleshores es mostren els error a la part dreta de la pantalla.

Exemples d'errors són: no definir un paràmetre que hi ha en algun enunciat, no afegir una solució a les preguntes o bé tenir un paràmetre definit que no s'utilitzi.

### 7.3.2.11 Baixar el fitxer XML

Amb el botó Baixar el fitxer XML es podrà descarregar la definició de l'exercici creat a partir del formulari. Abans cal haver clicat el botó Enviar dades i verificar que tot estigui correctament introduït, sense que l'editor hagi donat cap error.

El fitxer que es descarrega serà d'extensió XML i contindrà la definició de l'exercici creat a partir del formulari. Serà llavors quan amb aquest fitxer definició es podrà afegir a l'ACME, punt que s'explica en el següent apartat d'aquesta memòria.

## 7.4 Implantació del nou tipus d'exercici a l'ACME

Un cop definida l'estructura que haurà de tenir el fitxer definició d'un nou exercici i després de crear aquest fitxer definició mitjançant el nou editor que s'ha explicat en l'apartat anterior, és moment de realitzar tota la sèrie de canvis en el codi base de l'ACME per fer que la plataforma pugui reconèixer aquesta nova tipologia d'exercici i processar aquest nou tipus de fitxer, en llenguatge XML, de forma correcta. Els canvis que s'hauran de fer a la plataforma s'enumeren a continuació:

- **Procés de reconeixement del nou tipus de fitxer definició XML.** Quan s'afegeix un nou exercici a l'ACME, la mateixa plataforma comprova que el fitxer estigui correctament estructurat.
- **Procés de sorteig d'exercicis.** L'ACME té la seva pròpia metodologia per sortejar exercicis. Al haver canviat el tipus d'exercici, s'haurà d'adaptar l'ACME perquè dependent del tipus d'exercici faci servir un algorisme o altre.
- **Visualització dels enunciats dels exercicis.** S'haurà de modificar la manera en què l'ACME visualitza els enunciats dels exercicis ja que amb la nova tipologia s'han introduït canvis en la visualització.
- **Procés de correcció dels exercicis.** Cada tipus de solució diferent tindrà la seva pròpia manera de corregir els exercicis. És per això que s'haurà d'adaptar la plataforma perquè reconegui si es tracta d'un exercici dels nous per aplicar un algorisme de correcció o altre.

Generalment, tots els canvis que s'hauran de fer al codi base de la plataforma ACME estaran orientats a detectar en cada pas, si l'exercici que s'està tractant correspon a un exercici dels tradicionals, o bé correspon a un exercici dels nous. En cas que es tracti d'un exercici dels antics, s'hauran de dur a terme els processos corresponents que ja estan implementats com fins actualment. En canvi, si es tracta d'un dels exercicis nous, s'hauran d'aplicar els seus propis processos que s'han definit en aquest Projecte de Final de Carrera.

### 7.4.1 Procés de reconeixement del nou tipus de fitxer XML

Després de crear un exercici mitjançant l'editor que s'ha explicat en apartats anteriors i de descarregar-se el fitxer XML que contingui la definició d'aquest, l'ACME haurà de poder reconèixer aquest fitxer i ser capaç de construir l'estructura d'objectes a partir del contingut d'aquest fitxer XML.

L'ACME, com ja s'ha explicat, identifica les característiques de cada exercici mitjançant un fitxer definició propi de cada un amb una estructura comuna i és a partir d'aquest que és capaç de sortejar, visualitzar i corregir l'exercici. Cada fitxer definició d'exercici que es puja a l'ACME es verifica si és correcte o no. És per això que per aquesta nova tipologia d'exercicis s'ha hagut d'implementar el mètode que verifiqui que l'exercici compleix totes les especificacions explicades en el punt 7.1.

Si al pujar el fitxer es detecta algun error dins del fitxer, la plataforma informarà de quin error es tracta i passi el que passi sempre es verifica la totalitat del fitxer. Es podria dir que el constructor de la classe Exercici és com un compilador/verificador del llenguatge d'escriptura de problemes de l'ACME.

En canvi, si la verificació acaba amb èxit es dona d'alta l'exercici a la base de dades i el fitxer de definició de l'exercici queda guardat al directori que correspongui dins del sistema ACME, però encara no es generen els enunciats. És a dir, pels alumnes aquests problemes encara no són visibles, simplement l'exercici ha quedat verificat i emmagatzemat dins la plataforma ACME, a punt per ser sortejat.

A continuació es pot veure un exemple on es pot veure el resultat de la verificació:

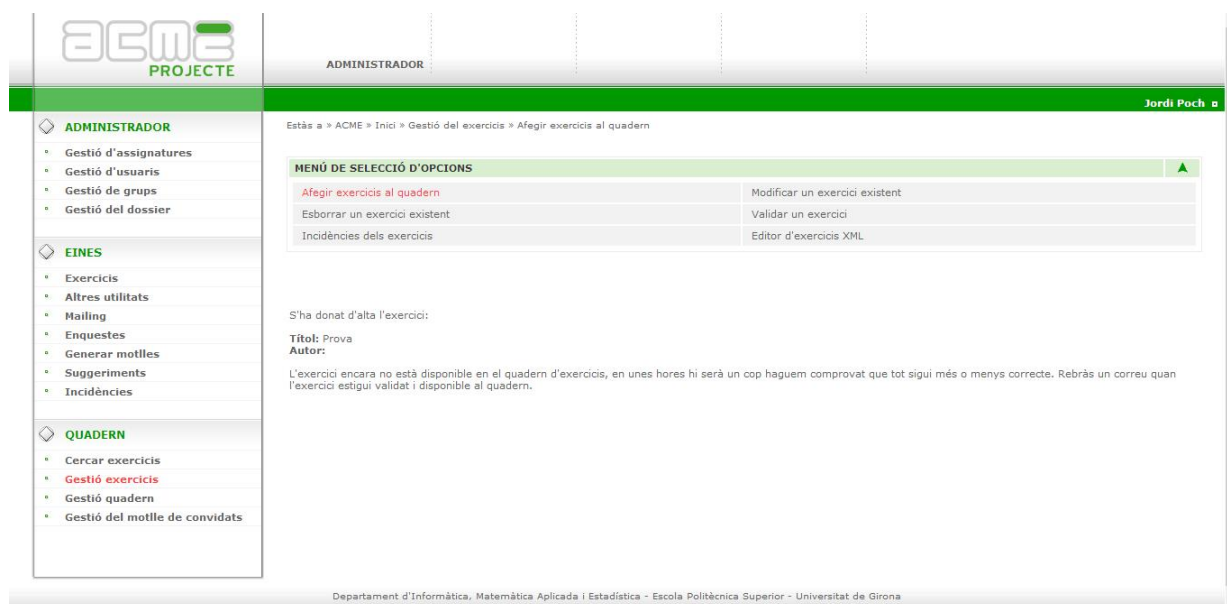


Figura 75: Exemple de verificació d'un exercici

Un cop s'hagi verificat els fitxers correctament ja es pot procedir a fer el sorteig dels enunciats. Per a la nova tipologia d'exercicis també s'ha hagut de crear un nou procediment que realitzi el procés amb els ampliacions i millores que s'han introduït amb la nova estructura dels nous exercicis. Aquest procés és explicat en el següent apartat d'aquesta memòria.

#### 7.4.2 Procés de sorteig d'exercicis

Un cop pujat i verificat el nou tipus d'exercici a l'ACME, ja es pot dur a terme un sorteig d'exercicis pels alumnes. El sorteig està segmentat en tres parts ben diferents. Primerament es realitzarà el sorteig de paràmetres, on per cada paràmetre que tingui l'exercici es seleccionarà un valor per cada un. Un cop sortejats els paràmetres es procedirà al sorteig dels enunciats, on per cada pregunta s'haurà d'escollir un enunciat dels disponibles a l'atzar. Finalment, un cop sortejats els paràmetres i els enunciats, s'haurà de recórrer el conjunt

dels enunciats escollits i substituir tots els paràmetres que apareguin en els redactats pels seus corresponents valors que hauran estat escollits.

Per explicar i entendre bé el funcionament del sorteig de paràmetres, a continuació s'adjunta un diagrama d'activitat que permetrà veure bé tots els passos que es segueixen a dur a terme el procés d'adjudicació de paràmetres:

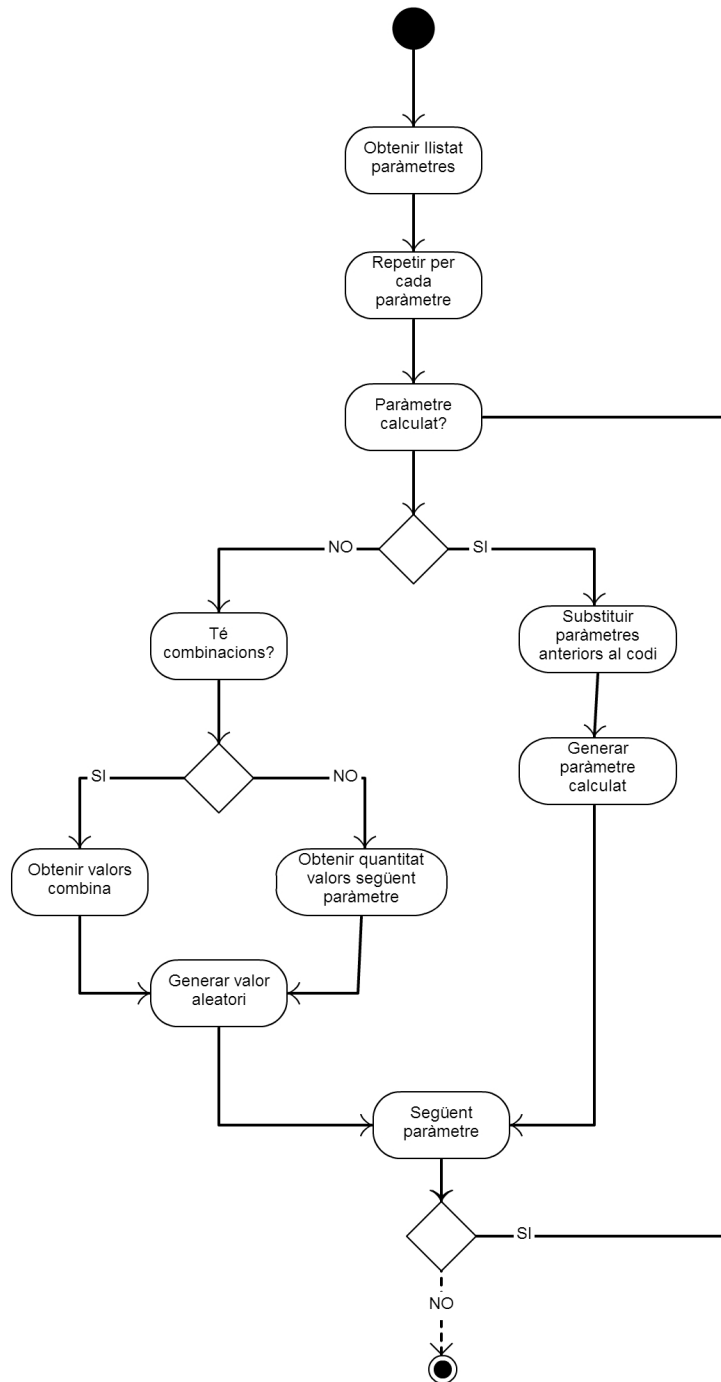


Figura 76: Diagrama d'activitat del sorteig de paràmetres



Per sortejar els paràmetres es seguirà el següent procediment:

- Primer de tot s'haurà d'obtenir el llistat complet de paràmetres de l'exercici. Un cop obtingut aquest llistat s'haurà de recórrer cada paràmetre del llistat i efectuar el sorteig.
- Si es tracta d'un paràmetre de tipologia llistat de valors, s'haurà de comprovar si el paràmetre anterior conté algun tipus de combinacions. Com ja s'ha explicat, les combinacions restringeixen el sorteig dels possibles valors de paràmetres, de manera que si el valor escollit de l'anterior paràmetre contenia alguna combinació, només hauran d'entrar en el procés de sorteig els valors del paràmetres que toquin. Si resulta que el valor escollit no tenia cap tipus de combinació, entraran al sorteig tots els possibles valors del paràmetre.
- Un cop determinat quins són els valors que entraran al sorteig, es generarà un número aleatori i s'escollirà un dels valors.
- En canvi, si es tracta d'un paràmetre calculat, abans d'executar el codi del paràmetre que generarà el seu valor, com que s'hi poden incloure valors de paràmetres anteriors a l'actual, s'haurà de substituir en el mateix codi aquests paràmetres anteriors que han pogut ser inclosos pels seus valors corresponents sortejats.
- Un cop substituïts els anteriors paràmetres en el codi adjunt al paràmetre calculat, s'executarà aquest codi i es generarà un valor visual i un calculat per aquest paràmetre.

Un cop sortejats els valors dels paràmetres ja es pot dur a terme el sorteig dels enunciats dels exercicis. Per sortejar els enunciats es seguirà el següent procediment:

- Primer de tot s'haurà d'obtenir el llistat complet de preguntes de l'exercici. Un cop obtingut aquest llistat s'haurà de recórrer cada pregunta del llistat i efectuar el sorteig.
- Per cada pregunta s'haurà d'obtenir el llistat complet d'enunciats dels quals hi ha disponibles. Un cop obtingut aquest llistat, se n'escollirà un de manera completament aleatòria.

A continuació s'adjunta un diagrama d'activitat per a entendre bé el funcionament del sorteig i veure cadascun dels passos que es segueixen per a efectuar la selecció:

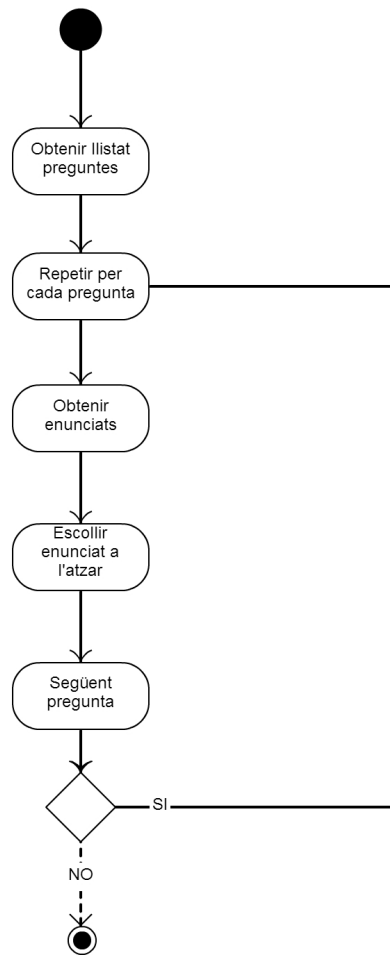


Figura 77: Diagrama d'activitat del sorteig d'enuncisats

Un cop s'han sortejat tant els paràmetres com els enuncisats, l'últim pas del procés de sorteig d'un exercici de la nova tipologia es tracta de substituir els paràmetres que apareguin en els redactats dels enuncisats escollits pels valors corresponents de cada paràmetre. Per a substituir els paràmetres en els redactats dels enuncisats s'haurà de seguir el següent procediment:

- Primerament s'hauran d'obtenir el conjunt de paràmetres i enuncisats que s'han sortejat prèviament.
- Un cop obtinguts aquests dos llistats, s'haurà de recórrer cada enunciat i efectuar les substitucions de tots els paràmetres.
- Per efectuar les substitucions dels paràmetres en un enunciat, s'haurà de recórrer el llistat de paràmetres però s'haurà de fer començant per l'últim dels paràmetres i acabant pel primer de tots. El procediment ha de ser aquest ja que si es fes començant pel primer, substituint els caràcters P1 pel valor sortejat d'aquest paràmetre, també substituiria tots els paràmetres P10, P11, P12, etc. Al haver-hi els caràcters P1 dins dels identificadors de cadascun d'aquests paràmetres també els substituiria. Això també passaria amb P2 i P20, P21, P22, etc. És per aquest motiu que primer es

comença pels últims i quan llavors toca substituir els primers, al estar tots aquests paràmetres que poden portar conflicte ja substituïts, ja no es té aquest problema.

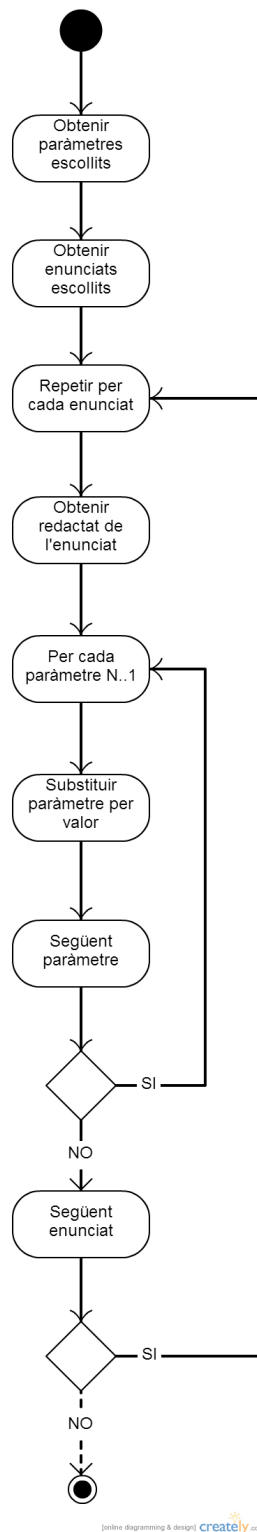


Figura 78: Diagrama d'activitat del sorteig d'enunciat

Un cop realitzats correctament aquests tres passos, ja es tenen exercicis sortejats i disponibles perquè els alumnes puguin accedir-hi. El següent pas és crear els mètodes necessaris en la jerarquia de classes del nou model perquè els nous tipus d'exercicis es mostrin correctament, tema que es tracta en el següent punt d'aquesta memòria.

### 7.4.3 Visualització dels enunciats dels exercicis

Una altra de les coses que s'ha hagut de modificar per implantar la nova metodologia d'escriptura d'exercicis ACME i conseqüentment el nou model d'exercici que s'ha dissenyat en aquest Projecte de Final de Carrera és la visualització dels enunciats dels exercicis un cop ja han estat sortejats.

Pel que fa a les modificacions de codi de l'ACME, en aquest cas, els únics canvis que s'han hagut de fer han estat orientats a detectar si l'exercici que s'està tractant correspon a un exercicis dels tradicionals, o bé correspon a un exercici dels nous. En el cas que es tracti d'un exercici dels antics, es procedirà a la visualització tradicional dels enunciats com està ja implementada a l'ACME. En canvi, si es tracta d'un dels exercicis nous, s'haurà de fer que l'ACME executi els nous mètodes de visualització que s'han implementat.

En la definició del nou model d'exercici s'ha implementat el mètode corresponent encarregat de la visualització d'un exercici. S'ha fet ús del potencial que proporciona la orientació a objectes, evitant que únicament la classe pare s'encarregui de la visualització global de l'exercici, sinó que aquesta tasca vagi sent delegada cap als objectes fills. D'aquesta manera no es centralitza tot en una única classe i a l'hora d'afegir nous elements no s'haurà de modificar el codi base de l'estructura del model, sinó que implementant el mètode específic de la nova classe n'hi haurà prou perquè l'exercici es pugui mostrar correctament.

Per explicar i entendre bé com funciona el procés de visualització d'un exercici i com es deriven les tasques entre els diferents objectes, s'adjunta un diagrama de seqüència del procés de visualització d'un exercici:

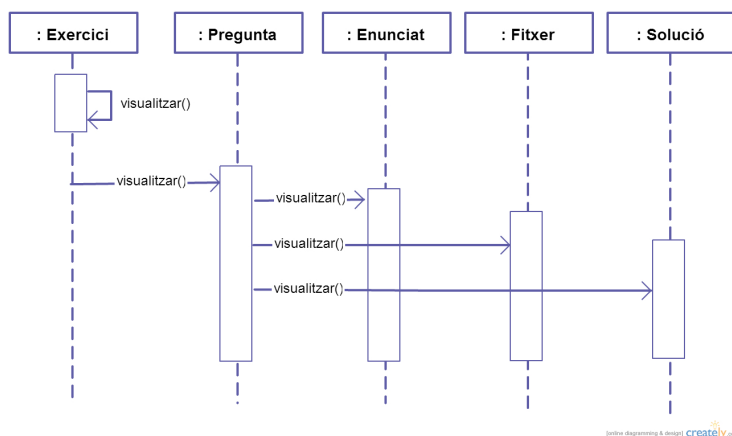


Figura 79: Diagrama de seqüència del procés de visualització d'un exercici

Com es pot veure, l'objecte Exercici fa la seva tasca de visualització i després crida el mètode de visualitzar cadascuna de les preguntes que conté. L'objecte pregunta fa el mateix a la vegada i crida els mètodes de visualització de l'enunciat, possibles fitxers i del tipus de solució que té associada.

Com ja s'ha explicat, un exercici pot ser mostrat de dues maneres. Si s'ha seleccionat que l'exercici sigui mostrat de manera vertical, es mostraran una pregunta sota l'altra. En canvi, si s'ha seleccionat que es mostri de manera horitzontal, mostrarà una sola pregunta de manera que no es podrà veure la següent pregunta fins tenir resolta la que la precedeix. A continuació s'adjunta una captura de pantalla d'un exercici d'exemple mostrat de manera vertical i mostrat de manera horitzontal:

## VISUALITZACIÓ VERTICAL

### Enunciat

La variable aleatòria  $X$  mesura l'índex d'audiència (en milions) de cert programa diari de TV. Se sap que  $X$  segueix una distribució Normal, que un 46% dels dies el programa té un índex inferior a 1.2 milions, i que el 16% dels dies té un índex superior a 1.9 milions.

**Nota:** Dóna les solucions a tots els apartats arrodonint com a mínim en el segon decimal.

### Pregunta 1

Determineu la esperança de la distribució Normal.

Solució:

### Pregunta 2

Determineu la desviació típica de la distribució Normal.

Solució:

### Pregunta 3

Quina és la probabilitat que en un dia determinat l'índex d'audiència sigui superior a  $\left[\frac{2x}{7}\right]$  milions d'espectadors?

**Nota:**  $[x]$  denota la part entera de  $x$ .

Solució:

**Corregir**

Figura 80: Captura de pantalla d'un exercici amb visualització vertical

## VISUALITZACIÓ HORITZONTAL

EXERCICI: Recta que passa per dos punts de l'activitat Activitat ◀ ▶

**ATENCIÓ:** en aquest exercici hauràs d'enviar a corregir pregunta a pregunta, no podràs veure la següent pregunta fins tenir resolta la que la precedeix.

◀ 1 2 3 ▶

### Pregunta 1

Troba el vector director de la recta que passa pels punts  $(-6, 8)$  i  $(-1, 8)$ . Si el vector és  $(2, 1)$  has d'escriure  $\{2,1\}$  (fes servir claus en lloc de parèntesi).

Solució:

**Corregir**

Figura 81: Captura de pantalla d'un exercici amb visualització horitzontal

#### 7.4.4 Procés de correcció dels exercicis

Amb el nou model d'exercici que s'ha definit en aquest Projecte Final de Carrera també s'ha hagut d'implementar un nou procés de correcció per a la nova tipologia d'exercicis. Igual que en les altres modificacions que s'han fet en el codi base de l'ACME, s'ha hagut de fer que, a l'hora de corregir un exercici, detecti si es tracta d'un exercici corresponent a la nova metodologia implementada, i si és el cas, que executi el nou procés de correcció en comptes del procediment tradicional.

En el disseny del nou model d'exercici s'hi defineixen dues maneres de corregir un exercici. Si és seleccionat el mètode incremental, el procés de correcció s'aturarà si una pregunta no és correcta. En canvi, si es selecciona el mètode total corregirà sempre totes les preguntes independentment del resultat de la seva correcció.

La manera com es mostra un exercici, en vertical o horitzontal, té especial importància a l'hora de avaluar-ne les solucions enviades per l'alumne. En un exercici mostrat de manera vertical s'hauran d'avaluar la totalitat de les preguntes, i en canvi, un exercici mostrat de manera horitzontal, únicament s'haurà d'avaluar una única pregunta. A part d'aquest incís, el procediment de correcció serà el mateix pels dos modes de visualització de l'exercici. A continuació s'explica de manera detallada els passos que es segueixen per avaluar una pregunta. Primer s'adjunta el diagrama d'activitat corresponent i després se n'expliquen els detalls d'implementació:

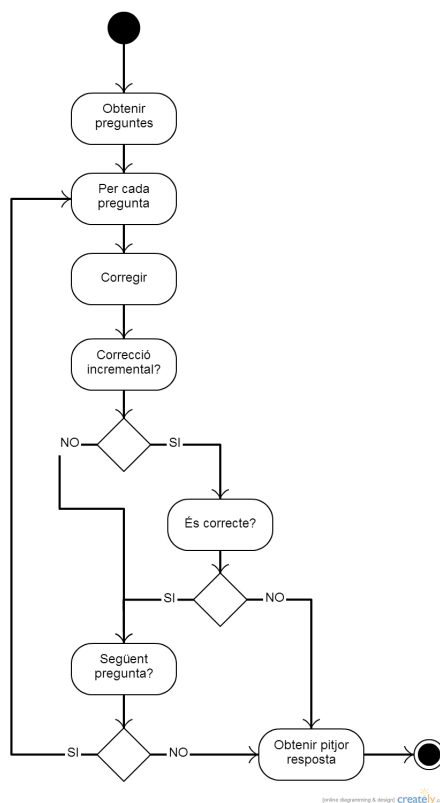


Figura 82: Diagrama d'activitat del procés de correcció d'un exercici

Per corregir una pregunta es seguirà el següent procediment:

- Primer de tot, s'haurà d'obtenir el conjunt de preguntes que s'han de corregir i s'haurà de recórrer aquest llistat per anar corregint cada pregunta de manera individual.
- Un cop es sàpiga el resultat de l'avaluació d'aquella pregunta, es comprovarà si es tracta d'un exercici amb correcció incremental.
- Si es tracta d'un exercici amb correcció total, es passarà a corregir la següent pregunta independentment del resultat d'avaluar aquesta pregunta.
- En canvi, si es tracta d'un exercici amb correcció incremental, s'haurà de comprovar si el resultat obtingut és el correcte. Si el resultat d'avaluar la pregunta és correcte, es passarà a avaluar la següent pregunta. En canvi, si la pregunta s'ha avaluat i el resultat ha estat negatiu, el procés de correcció s'aturarà i retornarà un resultat d'avaluació incorrecte.

El resultat global de la correcció sempre serà el més negatiu de tots els resultats parcials. En cas de seleccionar visualització horitzontal i correcció incremental, l'alumne no podrà passar a la següent pregunta fins tenir resolta la que està fent.

## 8 Implantació i resultats

El codi font definitiu de la nova metodologia d'escriptura d'exercicis que s'ha dissenyat i implementat en aquest Projecte de Final de Carrera, ja han estat pujats al servidor <http://acme.udg.edu>, la qual cosa vol dir que ja ha estat implantada definitivament. Durant el desenvolupament de tots els exercicis i per les primeres proves, hem fet servir un servidor alternatiu, també propietat de l'ACME: <http://acme2.udg.edu>.

Aquesta nova metodologia d'escriptura d'exercicis ja està més que consolidada dins de l'ACME i ja ha estat feta servir per nombrosos professors per a crear nous tipus d'exercicis. En general, el grau de satisfacció ha estat molt i més quan es compara aquesta nova metodologia amb la metodologia tradicional d'escriptura d'exercicis per l'ACME. Ja farà quasi mig any que es fa servir i ja hi ha 136 exercicis creats amb aquesta nova metodologia.



## 9 Conclusions

L'objectiu d'aquest Projecte Final de Carrera Nova metodologia d'escriptura d'exercicis per l'ACME era desenvolupar una metodologia d'escriptura d'exercicis diferent, innovadora i a l'abast de qualsevol professor per a poder crear nous tipus d'exercicis amb facilitat i sense haver de necessitar coneixements sobre programació.

El fet és que aquesta nova metodologia d'escriptura d'exercicis ja està implantada a l'ACME, ja farà quasi mig any que es fa servir i ja hi ha 136 exercicis creats amb aquesta nova metodologia. És per això que considero que s'han complert tots els objectius proposats de manera satisfactòria.

Aquest ha estat un projecte que m'ha permès consolidar tots els coneixements sobre e-learning, diferents llenguatges de programació i l'ús de tecnologies web, que ja havia adquirit amb la realització del Projecte Final de Carrera de l'Enginyeria Tècnica en Informàtica de Gestió. Llenguatges de programació com el HTML, el CSS, el Javascript, el JQuery, el PHP, i d'altres com el Mathematica i el LaTeX són tots els que he pogut consolidar amb èxit, durant el transcurs del projecte. Experimentar l'ambient i la metodologia que es segueix quan es forma part d'un projecte en equip com és l'ACME, també m'ha servit per seguir formant-me com a persona.

Cursar la carrera d'Enginyeria Superior en Informàtica m'ha servit per adquirir coneixements bàsics de programació, però pocs d'aquests coneixements m'han estat útils a l'hora de desenvolupar aquest P/TFC. Sinó que, en la gran majoria, els coneixements necessaris i imprescindibles per a desenvolupar aquest projecte han estat adquirits pel meu compte mitjançant tutorials en línia i auto aprenentatge.

Per tant, considero que aquest projecte m'ha servit moltíssim per aprendre nous coneixements i noves tecnologies, abans desconegudes per mi. L'experiència ha estat molt positiva i em considero privilegiat d'haver creat aquesta nova metodologia d'escriptura d'exercicis i que pugui servir a tants professors perquè puguin crear nous exercicis per l'ACME. Trobo molt satisfactori que aquest projecte no només hagi estat útil per mi, sinó que també ho sigui per molts usuaris de l'ACME i professors.

## 10 Ampliacions i millores

Aquest Projecte Final de Carrera ha establert els fonaments d'un model d'escriptura pels exercicis d'ACME, a partir d'ara queda molta feina per fer. Com a ampliacions i millores, es podrien afegir més funcionalitats i potenciar l'ús del nou editor que s'ha implementat en aquest Projecte Final de Carrera. Algunes de les noves funcionalitats que es podrien afegir són:

- Ampliar el nombre de solucions possibles: respostes test, química inorgànica, base de dades...
- Afegir botons fixes a les parts superior i inferior de la pantalla d'addició de preguntes i paràmetres per no haver d'anar amunt i avall fent scroll.
- Reordenar preguntes i paràmetres de manera fàcil i còmode fent possible l'arrossegament (drag and drop) d'aquests elements.
- Incorporar l'opció de testejar el codi de correcció de les preguntes en el mateix editor. També donar la possibilitat de seleccionar els valors dels paràmetres per fer les proves pertinents.
- Implementar un sistema de detecció d'errors a nivell de l'editor que faciliti al màxim l'escriptura, testeig i visualització d'exercicis.
- Afegir l'opció de desar l'exercici automàticament per poder seguir-hi treballant en una altra estona.
- Ampliar les metadades del fitxer XML amb totes que hi ha guardades amb cada exercici del repositori.
- Afegir un botó que permeti incorporar l'exercici que s'està creant de manera automàtica al quadern d'exercicis de l'ACME sense haver de descarregar-se el fitxer i afegir-lo manualment.
- Afegir botons per l'escriptura ràpida dels enunciats (codificació de matrius amb latex, funcions predefinides, símbols, ...) o l'escriptura de codis de correcció (comprovar si la solució envia és un vector, un número, una matriu,...). S'hauria d'aplicar la idea dels botons que feien la mateixa funcionalitat en l'anterior editor de problemes ACME.
- Incorporar un manual d'ajuda en el mateix editor.

## 11 Bibliografia

Aquesta és la bibliografia més utilitzada en aquest Projecte Final de Carrera, entre moltes altres pàgines web, articles i llibres que s'han consultat:

- PHP Hipertext Preprocessor, manual: <http://www.php.net>
- JQuery Javascript Library: <http://www.jquery.com>
- Enciclopèdia lliure: <http://www.wikipedia.org>
- Informació sobre HTML, CSS i web en general: <http://www.w3schools.com>
- Mathematica: <http://www.wolfram.com/mathematica/>
- Stackoverflow: <http://www.stackoverflow.com/>

## 12 Índex de figures

### Índex de figures

1	Diferents fases de la metodologia emprada en aquest Projecte Final de Carrera	7
2	Diagrama de Gantt amb la planificació	8
3	Funcionament de l'arquitectura Client-Servidor d'Internet	10
4	Exemple bàsic de HTML	11
5	Visualització d'un exemple bàsic d'HTML	12
6	Exemple bàsic de CSS	12
7	Visualització d'un exemple bàsic d'HTML amb CSS	12
8	Exemple bàsic de Javascript	13
9	Visualització d'un exemple bàsic de Javascript	13
10	Exemple bàsic de PHP	14
11	Taula de comparacions de frameworks Javascript 2	16
12	Exemple bàsic de JQuery	18
13	Com incloure el JQuery a la nostra pàgina web	19
14	Logotip del llenguatge LaTeX	19
15	Exemple bàsic de codi LaTeX	19
16	Exemple bàsic de codi Mathematica	20
17	Logotip del llenguatge Mathematica	20
18	Diferents navegadors que s'han utilitzat en aquest Projecte de Final de Carrera	21
19	Logotips del WinSCP i del PuttY	21
20	Logotip del Notepad ++	22
21	Exemple de capçalera d'un exercici	24
22	Exemple de definició d'un paràmetre en l'enunciat	24
23	Exemple d'un exercici amb 3 paràmetres declarats	25
24	Exemple de declaració de paràmetres	26
25	Exemple de declaració d'un paràmetre amb variables	27
26	Exemple de declaració d'un paràmetre amb combinacions	28
27	Exemple de codi de resolució	29
28	Exemple d'estructura d'un document esquema XML buit	37
29	Exemple de XML Schema senzill	38
30	Capçalera fitxer definició d'un exercici	39
31	Element Exercise	39
32	Element questionList	40
33	Element statementList	40
34	Element associatedFiles	41
35	Element solutionList	41
36	Element parameterList	42
37	Acabament del fitxer definició d'un exercici	42
38	Exemple de fitxer definició d'un tipus de solució o paràmetre	43
39	Estructura d'arbre del fitxer definició	44
40	Diagrama de classes del model	45
41	Exemple codi HTML	47
42	Exemple codi L <sup>A</sup> T <sub>E</sub> X	47

43	Fitxer definició de la tipologia solució expressió matemàtica . . . . .	49
44	Entrada de prova d'execució . . . . .	50
45	Sortida de prova d'execució . . . . .	50
46	Fitxer definició de la tipologia algorisme/programa informàtic . . . . .	50
47	Captura de la pantalla principal de l'editor antic . . . . .	53
48	Captura de la barra principal de l'editor antic . . . . .	54
49	Captura de la barra d'eines de l'editor antic . . . . .	54
50	Captura de la barra de problemes de l'editor antic . . . . .	54
51	Captura de la zona de treball de l'editor antic . . . . .	55
52	Captura de la pantalla de connexió de l'editor antic . . . . .	55
53	Captura de la pantalla de verificació de l'editor antic . . . . .	56
54	Captura de la pantalla d'elecció de paràmetres de l'editor antic . . . . .	56
55	Captura de la pantalla de visualització d'un enunciat de l'editor antic . . .	57
56	Captura de la pantalla de testeig de l'editor antic . . . . .	57
57	Captura de la pantalla de resultat de l'editor antic . . . . .	58
58	Captura de la pantalla inicial del nou editor . . . . .	59
59	Captura de la pantalla inicial del nou editor seleccionant fitxer existent . .	60
60	Captura de la pantalla principal del nou editor . . . . .	60
61	Captura de la pantalla de preguntes del nou editor . . . . .	61
62	Captura de la pantalla d'enunciats del nou editor . . . . .	62
63	Captura de la pantalla d'enunciat general del nou editor . . . . .	63
64	Captura de la pantalla de fitxers del nou editor . . . . .	63
65	Diagrama de classes de les noves taules SQL . . . . .	64
66	Captura de la pantalla de solució expressió matemàtica del nou editor . . .	65
67	Captura de la pantalla de solució algorisme/programa informàtic del nou editor . . . . .	66
68	Captura de la pantalla de paràmetres del nou editor . . . . .	66
69	Captura de la pantalla de llistat de valors del nou editor . . . . .	67
70	Captura de la pantalla de variables del nou editor . . . . .	67
71	Exemple de variables del nou editor . . . . .	68
72	Exemple de combinacions del nou editor . . . . .	68
73	Patró de codi Mathematica en paràmetre calculat . . . . .	69
74	Exemple de visualització en el nou editor . . . . .	70
75	Exemple de verificació d'un exercici . . . . .	72
76	Diagrama d'activitat del sorteig de paràmetres . . . . .	73
77	Diagrama d'activitat del sorteig d'enunciats . . . . .	75
78	Diagrama d'activitat del sorteig d'enunciats . . . . .	76
79	Diagrama de seqüència del procés de visualització d'un exercici . . . . .	77
80	Captura de pantalla d'un exercici amb visualització vertical . . . . .	78
81	Captura de pantalla d'un exercici amb visualització horitzontal . . . . .	78
82	Diagrama d'activitat del procés de correcció d'un exercici . . . . .	79

## 13 Agraïments

Vull transmetre els meus sincers agraïments al meu director, en Ferran Prados, pels excel·lents consells, recomanacions i suport continu, que m'han ajudat a realitzar tota la feina feta.

També a en Ferran, en Josep Soler i en Jordi Poch per haver-me donat l'oportunitat de formar part en un gran projecte com és l'ACME i dipositar la seva confiança en la meva persona.

I finalment, agrair a tothom que m'envolta el suport i ànims que m'han transmès durant el transcurs d'aquest Projecte Final de Carrera.