



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escola Tècnica Superior d'Enginyeria Informàtica

Ampliació i migració del lloc web per a la conversió del
llenguatge de marques a l'estàndard QTI/IMS

Treball Fi de Grau

Grau en Enginyeria Informàtica

AUTOR/A: Brines I Garcia, Joaquim

Tutor/a: Martí Campoy, Antonio

CURS ACADÈMIC: 2022/2023

Resum

El convertidor de llenguatge de marques a QTI és una aplicació Web que permet crear bateries de preguntes per a Poliformat de manera quasi automàtica i molt productiva. Aquest projecte té dos objectius. El primer, migrar l'aplicació Web de l'actual tecnologia, ASPX, a PHP, per utilitzar els servidors de la Universitat Politècnica de València. El segon, al mateix temps que es produeix la migració, millorar la seva estructura interna i codificació, per a afovir el seu manteniment, correcció i inclusió de noves funcionalitats.

Paraules clau: aplicació web, marques, qti, avaluació, automatització

Resumen

El convertidor de lenguaje de marcas a QTI es una aplicación Web que permite crear baterías de preguntas para Poliformat de manera casi automática y muy productiva. Este proyecto tiene dos objetivos. El primero, migrar la aplicación Web de la tecnología actual, ASPX, a PHP, para utilizar los servidores de la Universitat Politècnica de València. El segundo, al mismo tiempo que se produce la migración, mejorar su estructura interna y codificación, para favorecer su mantenimiento, corrección e inclusión de nuevas funcionalidades.

Palabras clave: aplicación web, marcas, qti, evaluación, automatización

Abstract

The markup language to QTI converter is a Web application that allows you to create question sets for Poliformat in an almost automatic and very productive way. This project has two objectives. First, to migrate the Web application from the current technology, ASPX, to PHP, to use the Universitat Politècnica de València servers. The second is to improve its internal structure and coding at the same time as the migration takes place, in order to facilitate its maintenance, correction and inclusion of new functionalities.

Key words: web application, markup, qti, evaluation, automation

Índex

Índex	3
Índex de figures	5

1 Introducció	1
1.1 Motivació	1
1.2 Objectius	2
1.2.1 Objectius generals	2
1.2.2 Objectius específics	3
1.3 Llenguatge de marques.	4
1.3.1 Definició	4
1.3.2 Exemples de llenguatges de marques	4
1.3.3 Llenguatge de marques propi	5
1.4 Estàndard QTI	5
2 Estat de l'art	7
2.1 Treball previ	7
2.2 Llenguatge ASP.NET	7
2.3 Llenguatge PHP	8
2.4 Diferències entre llenguatges	9
2.4.1 Avantatges i Desavantatges de PHP i ASP.NET	11
3 Requisits	13
3.1 Perspectiva del producte	13
3.1.1 Propòsit	13
3.1.2 Referències	14
3.2 Requisits funcionals	14
3.2.1 Usuari no registrat	14
3.2.2 Usuari registrat.	15
3.3 Requisits no funcionals	15
4 Planificació del treball	17
4.1 Metodologia	17
4.1.1 Sprints	18
5 Anàlisi	20
5.1 Vista de l'usuari	20
5.2 Vista del desenvolupador	21
5.3 Exemples de preguntes	23
5.3.1 Pregunta numèrica	23
5.3.2 Pregunta multiple	24
5.3.3 Pregunta de completar els buits	25
6 Disseny	26
6.1 Vista d'usuari	26

6.2	Vista del desenvolupador	26
6.3	Arquitectura de tres capes	27
6.3.1	Capa de presentació	27
6.3.2	Capa d'aplicació	28
6.3.3	Capa d'accés a les dades	28
6.4	Model client - servidor	29
6.4.1	Client	29
6.4.2	Servidor	30
7	Implementació	31
7.1	DOMDocument	31
7.1.1	Estructura	32
7.1.2	Suport de múltiples idiomes	32
7.1.3	Gestió de fitxers al servidor	32
7.2	Eines i tecnologies utilitzades	34
7.2.1	Diagrams.net	34
7.2.2	Visual Studio Code	34
7.2.3	Servidor HTTP Apache	34
7.2.4	CSSGradient	34
7.2.5	Navegadors	34
8	Proves	35
9	Conclusions	36
9.1	Conclusions	36
9.2	Projectes futurs	36
9.2.1	Gestió d'usuaris	36
9.2.2	Gestió de fitxers	36
9.2.3	Previsualitzador de preguntes	37
9.2.4	Afegir tipus de preguntes i opcions	37
9.3	Valoració personal.	37
	Bibliografia	40
<hr/>		
	Apèndixs	
A	Llenguatge de marques propi	41
B	Esbossos de la pàgina web	43
C	Guía d'ús	48
D	Glosari	50

Índex de figures

1.1	Text en llenguatge de marques.	2
1.2	Fragment del fitxer xml.	2
1.3	Pes dels fitxers.	2
1.4	Gràfic comparatiu de pes dels fitxers - Disseny propi.	3
1.5	Comparació Markdown (esquerra) i HTML (dreta) - Imatge procedent de https://markdown.es	5
2.1	Antiguetat de cada llenguatge - Disseny propi	9
2.2	Rendiment de cada llenguatge - Disseny propi	10
2.3	Seguretat de cada llenguatge - Disseny propi	10
2.4	Suport de la comunitat de cada llenguatge - Disseny propi	11
3.1	Diagrama d'aplicació - Disseny propi	16
4.1	Planificació dels sprints - Disseny propi	18
5.1	Pàgina d'inici original.	20
5.2	Editor original sense número de línia.	21
5.3	Exemple de codi en l'editor Notepad++	21
5.4	Pregunta numèrica en Poliformat	23
5.5	Pregunta de selecció múltiple en Poliformat	24
5.6	Pregunta de completar els buits en Poliformat	25
6.1	Pàgina d'inici anterior.	27
6.2	Pàgina d'inici de la UPV.	27
6.3	Pàgina d'inici actual.	28
6.4	Arquitectura de tres capes - Disseny propi	29
6.5	Model client-servidor - Disseny propi	30
7.1	Vulnerabilitat en la gestió de fitxers en PHP - Disseny propi	33
B.1	Esbós 1	43
B.2	Esbós 2	44
B.3	Esbós 3	44
B.4	Esbós 4	45
B.5	Esbós 5	45
B.6	Esbós 6	46
B.7	Esbós 7	46
B.8	Disseny final.	47
B.9	Disseny final.	47
C.1	Pagina d'inici. L'usuari tria si vol carregar un fitxer o escriure o pegar directament en l'editor.	48

C.2	Editor. L'usuari pot escriure o pegar les qüestions, donar nom al fitxer resultat i el nom de la bateria de preguntes.	49
C.3	Càrrega d'arxiu. L'usuari pot carregar el fitxer directament, donar nom al fitxer resultat i el nom de la bateria de preguntes.	49
C.4	Descàrrega d'arxiu. La conversió s'ha realitzat i l'usuari pot descarregar l'arxiu.	49

CAPÍTOL 1

Introducció

A continuació s'exposen quins motius han donat lloc a dur a terme aquest projecte, quins són els objectius establits i com s'ha estructurat la memòria.

1.1 Motivació

Aquest treball s'ha elaborat continuant els projectes creats pels alumnes Héctor Herraiz Muñoz i continuat per Jorge Brusel Borrero, tots dos tutoritzats pel docent Antonio Martí Campoy, on es desenvolupa una aplicació web que soluciona un problema que té el professorat que treballa amb exàmens en la plataforma 'Poliformat': generar bateries de preguntes d'una manera ràpida i senzilla sense haver d'invertir temps en gestions del format del document i poder dedicar aquest temps a la pròpia creació de les preguntes.

L'aplicació creada facilita aquesta tasca perquè permet convertir text escrit emprant un llenguatge simple de marques propi en un fitxer .xml compatible amb el sistema de poliformat. La causa per la qual es proposa i crea aquest treball és el resultat d'analitzar els punts que presenten una millora respecte als projectes anteriors. Els principals són: el llenguatge de programació utilitzat, ASP.NET, que no s'integra amb l'ecosistema d'aplicacions web que posseeix la Universitat Politècnica de València (UPV) i el propi codi que presenta gran marge de millora des d'un punt de vista de llegibilitat, sostenibilitat i futures ampliacions.

En les figures 1.1 i 1.2 es mostra la diferència entre una qüestió escrita amb el format xml i la mateixa pregunta fent servir el llenguatge de marques. El primer ocupa 5 línies mentre el segon 671 línies.

Es pot observar com el llenguatge de marques és prou més lleuger, 163 bytes que ocupen un fitxer escrit amb llenguatge de marques que contrasten amb els 25 kilobytes que ocupa un fitxer xml. En la figura 1.3 es mostra en una terminal el pes dels arxius i en la figura 1.4 s'ha realitzat un gràfic per comparar visualment la diferència.

Però el més important és que el llenguatge de marques dissenyat pel tutor en els anterior projectes, és amigable per al professor. Aquest pot llegir els fitxers, fer modificacions de manera senzilla i amb molt poc de risc de fer malbé l'estructura de les preguntes. I encara més, permet de manera senzilla, automatitzar la creació de preguntes fent servir eines com Excel o programació en qualsevol llenguatge.

El que resta de memòria s'utilitzarà el terme "l'aplicació" referint a l'aplicació web desenvolupada.

```
<QF><M1,5> En pedra, paper o tisores:<br>
La pedra guanya a: {tisores}
Per a no perdre contra la pedra, utilitzes:
{pedra | paper| pedra i paper | pedra o paper}
</Q>
```

Figura 1.1: Text en llenguatge de marques.

```
<material>
  <mattext charset="ascii-us" texttype="text/plain" xml:space="default" />
</material>
</rubric>
<presentation label="F1B">
  <flow class="Block">
    <material>
      <mattext charset="ascii-us" texttype="text/plain" xml:space="default"><![CDATA[ En pedra, paper o tisores:<br><br>La pedra guanya a: ]]></mattext>
    </material>
    <material>
      <mattext charset="ascii-us" texttype="text/plain" xml:space="default"><![CDATA[<br>Per a no perdre contra la pedra, utilitzes: <br>]]></mattext>
    </material>
    <response_str ident="F1B00" rcardinality="Ordered" rtiming="No">
      <render_fib charset="ascii-us" columns="5" encoding="UTF_8" fibtype="String" prompt="Box" rows="1" />
    </response_str>
    <material>
      <mattext charset="ascii-us" texttype="text/plain" xml:space="default"><![CDATA[<br>]]></mattext>
    </material>
    <response_str ident="F1B01" rcardinality="Ordered" rtiming="No">
      <render_fib charset="ascii-us" columns="5" encoding="UTF_8" fibtype="String" prompt="Box" rows="1" />
    </response_str>
    <material>
      <mattext charset="ascii-us" texttype="text/plain" xml:space="default" />
    </material>
  </flow>
</presentation>
<resprocessing>
  <outcomes>
    <decvar defaultval="0" maxvalue="1.5" minvalue="0" varname="SCORE" vartype="Integer" />
  </outcomes>
  <rescondition continue="Yes">
    <conditionvar>
      <or>
        <varequal case="No" respident="F1B00"><![CDATA[<![CDATA[tisores]]>]]></varequal>
      </or>
    </conditionvar>
    <setvar action="Add" varname="SCORE">0</setvar>
  </rescondition>
  <rescondition continue="Yes">
    <conditionvar>
      <or>
        <varequal case="No" respident="F1B01"><![CDATA[<![CDATA[pedra| paper| pedra i paper | pedra o paper]]>]]></varequal>
      </or>
    </conditionvar>
  </rescondition>
</resprocessing>
```

Figura 1.2: Fragment del fitxer xml.

```
ubuntu@ubuntu-K53SJ:/home/tfg$ ll -h
total 40K
drwxr-xr-x 2 root root 4,0K abr 12 17:24 ./
drwxr-xr-x 4 root root 4,0K abr 12 17:20 ../
-rw-rw-rw- 1 root root 163 abr 12 17:49 llenguatgeDeMarques.txt
-rw-rw-rw- 1 root root 25K abr 12 17:49 qti.xml
```

Figura 1.3: Pes dels fitxers.

1.2 Objectius

1.2.1. Objectius generals

El principal objectiu d'aquest treball és la migració de l'aplicació des del llenguatge ASP.NET al llenguatge PHP, aprofitant aquesta migració per a millorar el codi, en els aspectes de llegibilitat, manteniment i facilitat d'ampliacions de funcionalitats.

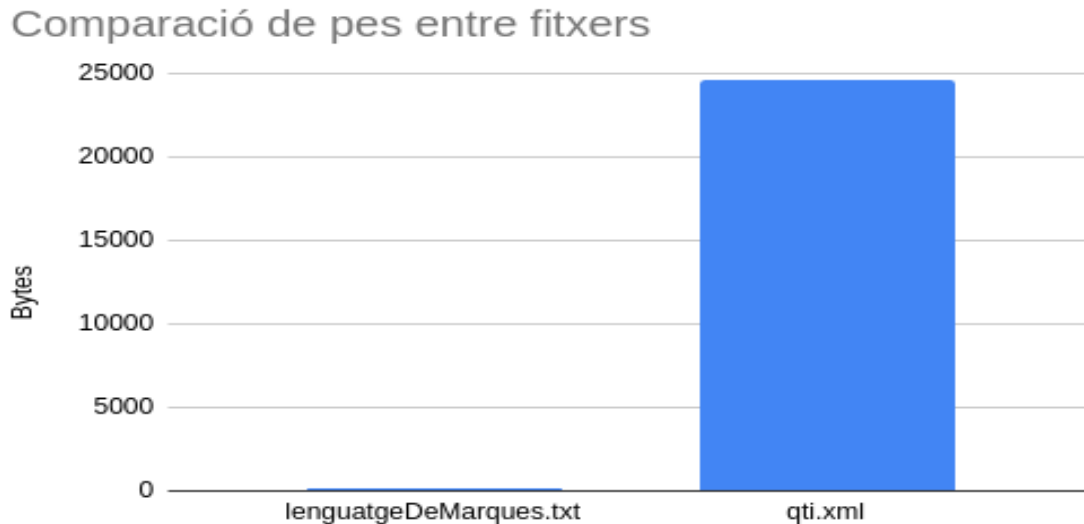


Figura 1.4: Gràfic comparatiu de pes dels fitxers - Disseny propi.

1.2.2. Objectius específics

A continuació es descriuen els objectius específics del projecte

- **Modernitzar la pàgina web.**

Modernitzar aspectes visuals de la pàgina web canviant l'estructura del text, canviant la paleta de colors, afegint l'opció de seleccionar l'idioma entre múltiples idiomes o mostrar el número de línia del text introduït en el quadre de text previ a la conversió.

- **Migració del codi font escrit en ASP.NET al llenguatge PHP.**

Migrar el codi font escrit en el llenguatge de programació ASP.NET al llenguatge PHP, per poder integrar-lo millor amb els servidors de la UPV.

- **Afegir una capa de seguretat a la pàgina web mitjançant protocol SSL.**

Afegir una capa de seguretat mitjançant l'ús del protocol SSL, sigles que en anglès signifiquen 'Service Secure Layer' o servei de capa de seguretat. Aquest protocol està disponible gràcies a PLESK, que és el sistema utilitzat per la UPV.

- **Millores en el codi.**

Refactorització del codi, perquè adquireixi una estructura més llegible i conseqüentment facilite el manteniment de l'aplicació per a futurs desenvolupadors.

Llançament i gestió d'excepcions en cas de fallades amb el text proporcionat i bases per a ampliacions futures.

- **Gestió d'errors.**

Afegir una gestió d'errors mitjançant fitxers .log on s'emmagatzemen els errors produïts amb els fitxers proporcionats per l'usuari. Aquests fitxers els pot veure exclusivament un administrador del sistema per a identificar la causa dels errors.

- **Gestió d'usuaris.**

Incorporar una gestió d'usuaris amb les operacions bàsiques de creació, edició i esborrat de comptes.

- **Gestió d'arxius dels usuaris.**

Gestió de fitxers al servidor per part de l'usuari, emmagatzematge, edició i esborrat d'arxius.

1.3 Llenguatge de marques.

En aquest apartat es definirà el concepte de 'llenguatge de marques' així com exemples reals i l'ús del llenguatge al projecte.

1.3.1. Definició

Un llenguatge de marques és un tipus de codificació del llenguatge mitjançant l'ús d'etiquetes o marques que compleixen una funcionalitat concreta i definida pels desenvolupadors del llenguatge. Mitjançant l'ús d'aquestes etiquetes es pot convertir un text pla a un text amb una estructura preparada per a ser llegida per un programa que aplique les regles d'aquestes etiquetes i el dote del format corresponent.

1.3.2. Exemples de llenguatges de marques

L'exemple més clar d'un llenguatge de marques és el cas del llenguatge 'HTML' utilitzat per a dissenyar els elements que componen una pàgina web. Tant els títols, les imatges, els botons, el text i la resta d'elements estan escrits emprant les marques apropiades per a aconseguir l'estructura desitjada.

Un altre exemple de llenguatge de marques és el llenguatge 'Markdown' el qual, mitjançant l'ús d'etiquetes, pot convertir un text al llenguatge 'HTML' descrit abans, simplificant la tasca de la creació de pàgines web.

La imatge 1.5 mostra la conversió d'un text amb llenguatge 'Markdown' al llenguatge 'HTML'.

Com es pot observar, tots dos són llenguatges de marques, però les etiquetes o marques que utilitzen estan definides per cadascun com també la seua funcionalitat.

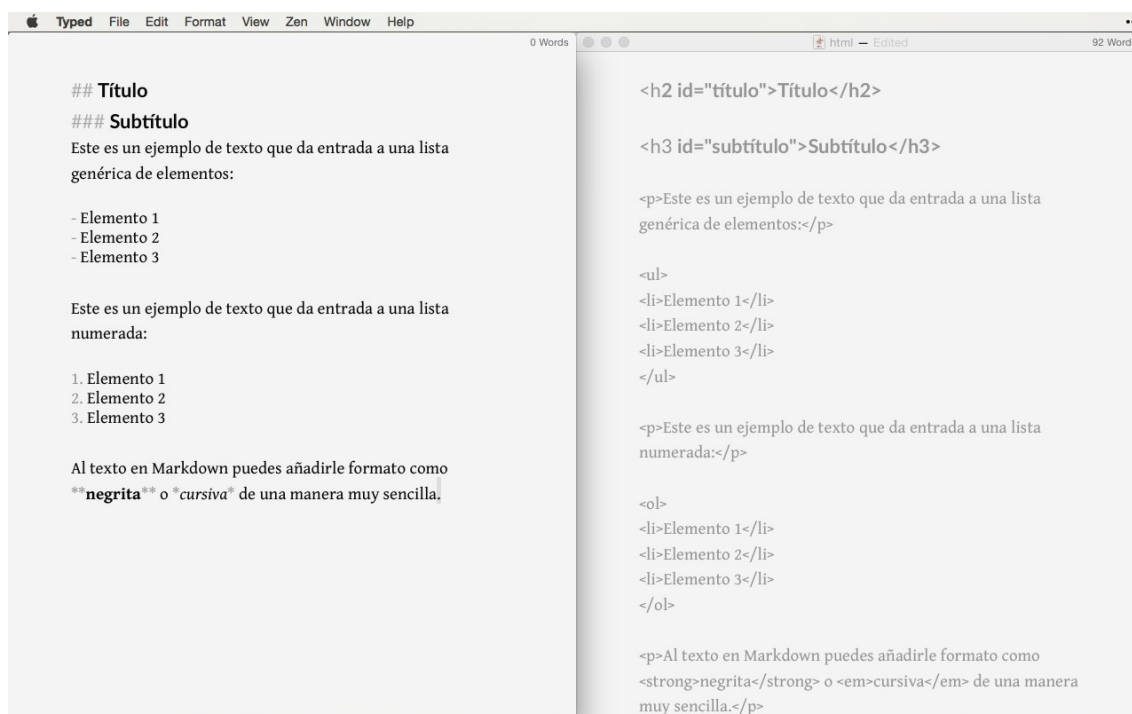


Figura 1.5: Comparació Markdown (esquerra) i HTML (dreta)
- Imatge procedent de <https://markdown.es>

1.3.3. Llenguatge de marques propi

Respecte a l'aplicació desenvolupada en aquest treball, s'usa el llenguatge de marques establert on les marques estan escrites entre els símbols '`<`' i '`>`', igual que en el llenguatge 'HTML' descrit anteriorment, que disposen d'una funcionalitat concreta definida pels projectes precedents a aquest.

Amb aquestes marques es defineixen les preguntes, respostes i altres atributs com a puntuació, les opcions que mostra la pregunta així com la, o les, respostes correctes, el tipus de pregunta, la qual pot ser numèrica, de text o de completar els blocs buits.

Per exemple, una nova pregunta numèrica es definiria amb la marca `<QN>` a l'inici de l'enunciat seguit de la marca amb la puntuació de la pregunta i la pregunta amb la resposta, o possibles respostes, entre els símbols de claus `{}` i finalitzant amb la marca de tancament de pregunta `</Q>`.

Un dels avantatges del nostre llenguatge és que permet la creació automàtica de bateries de preguntes. Per a crear aquestes bateries es pot fer servir Excel i el seu 'Autocompletar'. També Word o qualsevol llenguatge de programació per a generar el fitxer de text amb les preguntes.

A l'annex A es detalla el llenguatge de marques que es va crear.

1.4 Estàndard QTI

L'estàndard QTI, acrònim anglès de "Question and Test Interoperability", desenvolupat per l'organització IMS Global, és una especificació d'arxius XML que se-

gueixen un format determinat per l'estàndard per crear i intercanviar arxius de preguntes i proves a través de múltiples programes i sistemes dispars. Proporciona un nivell comú per a representar l'estructura dels diferents tipus de preguntes i de respostes, mitjançant el qual s'habilita la interoperabilitat transparent entre sistemes. El propòsit d'aquesta especificació és proporcionar un marc consistent per representar l'estructura de preguntes, respostes i tota classe de materials d'examen, facilitant la interoperabilitat entre diversos sistemes.

El principal motiu de la utilització d'un nivell comú com QTI és que assegura que es poden compartir els documents estandarditzats a través d'una varietat de paràmetres amb un alt grau de consistència i precisió, proporcionant un procés més modern i eficaç a l'hora de la creació de fitxers. Cal afegir que la implementació d'un nivell com QTI ajuda a mitigar la potencial falla dins de la pregunta i simplificar el procés de creació, oferint una representació clara i inequívoca d'informació i promovent l'enteresa global de l'aplicació.

CAPÍTOL 2

Estat de l'art

En aquest apartat s'especifica el treball previ existent sobre el qual s'ha desenvolupat l'aplicació.

2.1 Treball previ

El treball pren com a inici el programa realitzat per Jorge Brusel Borrero, escrit en el llenguatge ASP.NET, llenguatge de programació orientat a programes en el núvol, i té l'objectiu de crear una aplicació web que permeti als docents generar un document al sofisticat estàndard QTI/IMS a partir d'un document escrit en un senzill llenguatge de marques.

L'estàndard QTI/IMS és àmpliament utilitzat per plataformes de docència com és el cas de la UPV.

2.2 Llenguatge ASP.NET

.NET és un entorn de desenvolupament de programari desenvolupat per Microsoft que permet als desenvolupadors crear i desplegar diverses aplicacions, incloent-hi aplicacions web, aplicacions d'escriptori, aplicacions mòbils i aplicacions de jocs. Va ser introduït per primera vegada l'any 2002 i des de llavors s'ha convertit en un dels marcs més populars entre els desenvolupadors.

El marc ASP.NET proporciona una sèrie de llenguatges de programació per desenvolupar aplicacions. També inclou un Common Language Runtime (CLR), que és responsable d'executar el codi escrit en aquests llenguatges de programació. El CLR proporciona diverses característiques, com la gestió de memòria, seguretat i gestió d'excepcions, per assegurar que el codi s'executa de manera eficient i segura.

Un dels avantatges clau d'utilitzar el marc ASP.NET és que proporciona una vasta biblioteca de components preconstruïts, anomenats biblioteques de classe, que es poden utilitzar per realitzar tasques de programació comunes. Aquestes biblioteques de classes inclouen components per gestionar connexions de bases de dades, crear interfícies d'usuari i gestionar comunicacions de xarxa, entre d'altres.

Un altre avantatge d'utilitzar el marc ASP.NET és que proporciona un ric conjunt d'eines per desenvolupar, provar i desplegar aplicacions. Visual Studio Integrated Development Environment (IDE) és una eina potent que suporta diversos llenguatges de programació i proporciona funcions com l'edició de codi, la depuració i l'anàlisi de rendiment.

El marc ASP.NET també proporciona un entorn d'execució, anomenat ASP.NET Core runtime, que permet als desenvolupadors construir i executar aplicacions en múltiples plataformes, incloent-hi Windows, macOS i Linux. Això permet als desenvolupadors escriure codi una vegada i desplegar-lo en múltiples plataformes, que poden reduir significativament el temps de desenvolupament i el cost.

A més, el marc ASP.NET proporciona diverses funcions de seguretat, com ara seguretat d'accés al codi i seguretat basada en rols, que es poden utilitzar per assegurar que les aplicacions són segures i protegeixen contra diversos tipus d'atacs, com ara la creació de scripts entre llocs i la injecció SQL.

En general, el marc ASP.NET proporciona una plataforma potent i flexible per desenvolupar una àmplia gamma d'aplicacions. La seua extensa biblioteca de components preconstruïts, un ric conjunt d'eines de desenvolupament, i entorn d'execució multiplataforma el fan una elecció ideal per al treball acadèmic en el desenvolupament de programari.

2.3 Llenguatge PHP

PHP és un llenguatge de script del costat del servidor utilitzat per al desenvolupament web. Va ser introduït per primera vegada el 1995 i des de llavors s'ha convertit en un dels llenguatges de programació més utilitzats per al desenvolupament web.

Un dels avantatges clau de PHP és que és un llenguatge de programari lliure, el que significa que pot ser servit i modificat lliurement pels desenvolupadors. Això ha donat lloc a una gran comunitat activa de desenvolupadors que contribueixen constantment al desenvolupament de PHP.

PHP s'utilitza principalment per a la construcció de pàgines web dinàmiques i aplicacions web. Proporciona diverses funcions, com ara connectivitat de base de dades, gestió de formularis i gestió de sessions, que permeten als desenvolupadors crear aplicacions web interactives i sensibles.

Un altre avantatge de PHP és que pot ser fàcilment integrat amb diverses bases de dades, com MySQL, Oracle, i PostgreSQL. Això fa que siga una opció ideal per construir aplicacions web guiades per bases de dades.

A més, PHP proporciona una àmplia gamma de funcions i biblioteques integrades que faciliten als desenvolupadors realitzar tasques de programació comunes, com ara la manipulació de cadenes, gestió de fitxers i processament d'imatges.

PHP també proporciona una gamma de marcs, com Laravel, CodeIgniter, i Symfony, que proporcionen funcions addicionals i funcionalitats per al desenvolupament web. Aquestes eines poden reduir significativament el temps de

desenvolupament i ajudar els desenvolupadors a crear aplicacions web escalables i sostenibles.

En general, PHP és un llenguatge potent i flexible per al desenvolupament web. La seua naturalesa de codi obert, connectivitat de bases de dades, funcions incorporades i marcs el converteixen en una opció ideal per al treball acadèmic en el desenvolupament web.

2.4 Diferències entre llenguatges

PHP i ASP.NET són dos llenguatges de programació populars utilitzats per al desenvolupament web. Tot i que comparteixen algunes similituds, també tenen diferències significatives en termes de les seues característiques, rendiment i suport comunitari. A continuació es mostra una comparació entre tots dos llenguatges:

En la figura 2.1 es veu com PHP va començar l'any 1994 mentre que ASP.NET a 2002. Respecte al rendiment, els dos són ràpids com es veu a la figura 2.2 però ASP.NET és lleugerament més ràpid pel fet que és un llenguatge compilat. Continuant amb la seguretat, figura 2.3, és difícil establir quin és més segur, degut a que PHP es de codi obert i ASP.NET de codi propietari de Microsoft. Per últim, el suport de la comunitat, figura 2.4, la comunitat de PHP és la més activa, ja que ha sigut un llenguatge de programari lliure des dels orígens.

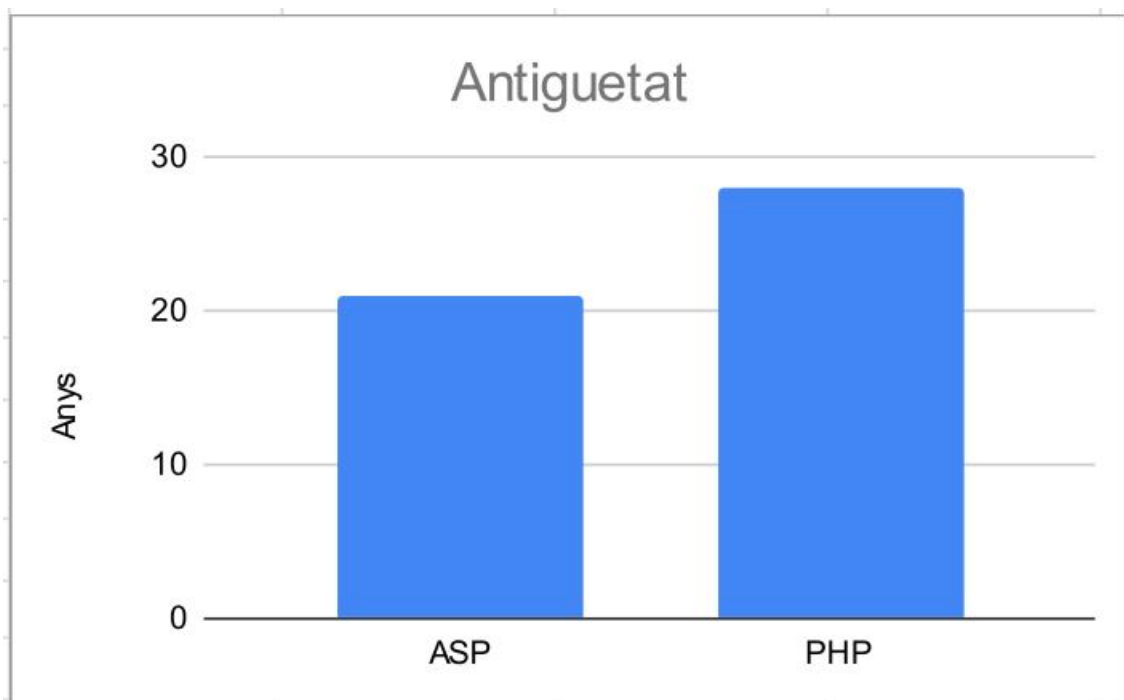


Figura 2.1: Antiguetat de cada llenguatge - Disseny propi

- Tipus de comportament:
PHP és un llenguatge de 'scripting', el qual vol dir que s'interpreta en temps d'execució.

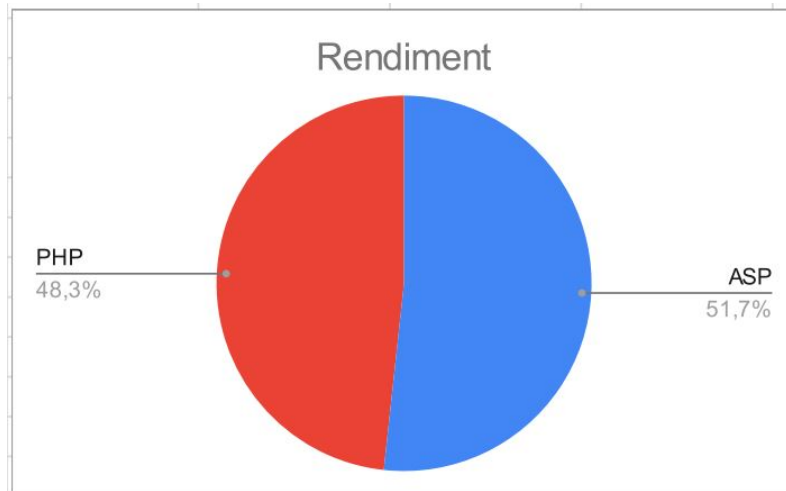


Figura 2.2: Rendiment de cada llenguatge - Disseny propi

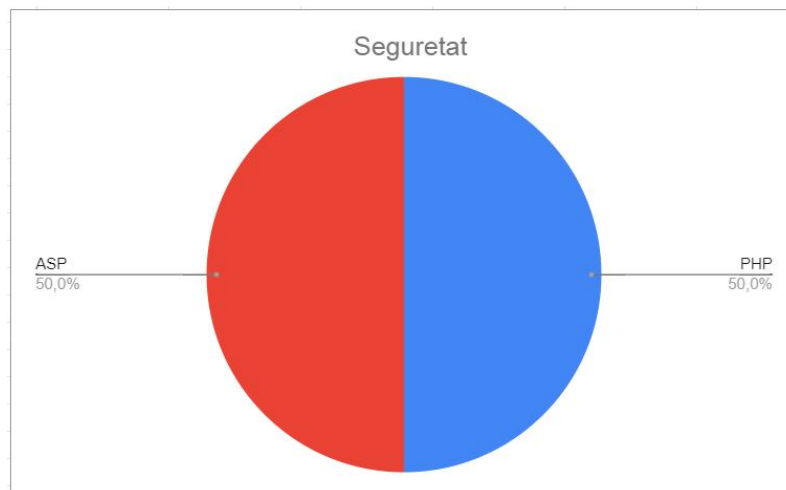


Figura 2.3: Seguretat de cada llenguatge - Disseny propi

Per altra banda, ASP.NET és un llenguatge compilat, el qual vol dir que el codi es compila en un fitxer executable abans que es pugui executar.

- Sintaxi de la llengua:

La sintaxi de PHP és relativament fàcil d'aprendre i entendre, fent-lo una elecció ideal per a principiants.

ASP.NET, d'altra banda, té una sintaxi més complexa, però proporciona una millor comprovació de tipus i gestió d'errors.

- Rendiment:

PHP és conegut per la seua velocitat d'execució i d'inici. Gràcies al fet que és un llenguatge que s'interpreta en temps d'execució, els 'scripts' de PHP poden executar-se ràpidament, fent-lo una elecció popular per a desenvolupadors de web. Tanmateix, PHP pot requerir més recursos del servidor

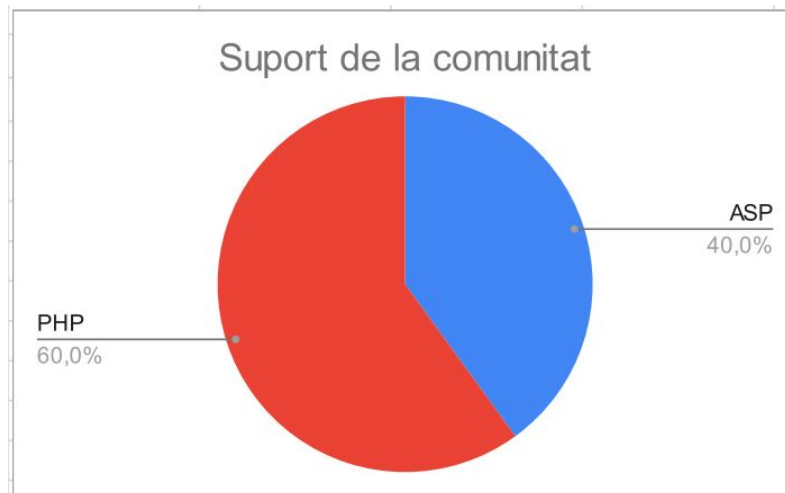


Figura 2.4: Suport de la comunitat de cada llenguatge - Disseny propi

quan treballa amb dades i aplicacions grans per a processar.

ASP.NET és conegut pel seu rendiment. Com que el codi es compila en un fitxer executable, es perd velocitat a l'hora d'iniciar, però una vegada compilat s'executa més ràpidament que PHP.

- Eines de desenvolupament:

Tant PHP com ASP.NET tenen una àmplia gamma d'eines de desenvolupament i IDE disponibles. Tots dos poden ser desenvolupats amb potents IDE que tenen característiques avançades per desenvolupar i depurar. Destaquen 'Visual studio code' i 'Netbeans'.

- Suport comunitari:

PHP té una gran comunitat activa de desenvolupadors que contribueixen al seu desenvolupament i proporcionen suport a través de fòrums i comunitats en línia.

ASP.NET també té una gran comunitat de desenvolupadors, però se centra més en el desenvolupament a escala empresarial.

2.4.1. Avantatges i Desavantatges de PHP i ASP.NET

Avantatges de PHP:

- Ràpid desplegament al ser un llenguatge compilat en temps d'execució
- És fàcil d'aprendre i entendre.
- Una àmplia gamma d'eines i marcs de desenvolupament.
- Una comunitat gran i activa de desenvolupadors.

- Un bon suport a les tecnologies de codi obert.
- Està suportat per la majoria de servidors, inclòs IIS¹.

Desavantatges de PHP:

- Llenguatge interpretat, que pot afectar el rendiment.
- Suport limitat per al desenvolupament a escala empresarial.

Avantatges d'ASP.NET:

- Rendiment ràpid i escalable.
- Eines avançades de desenvolupament i IDE.
- Una gran comunitat de desenvolupadors amb un enfocament en el desenvolupament a escala empresarial.

Desavantatges d'ASP.NET:

- Sintaxi complexa amb gran corba d'aprenentatge.
- Suport limitat per a tecnologies de codi obert.
- El programari propietari i les eines poden ser cares.
- Soles suportat per IIS.

En general, tant PHP com ASP.NET són llenguatges potents i flexibles per al desenvolupament web, i l'elecció entre els dos depèn de les necessitats específiques del projecte. PHP és una bona elecció per a projectes més reduïts, mentre que ASP.NET és més adequada per a projectes més grans a escala empresarial.

¹ Acrònim d'Internet Information Services', és un servidor web versàtil executat en un sistema Windows.

CAPÍTOL 3

Requisits

3.1 Perspectiva del producte

El treball que es realitzarà consisteix a migrar una pàgina web d'un llenguatge de programació a un altre prou diferent. La pàgina web ha de complir els requisits d'usabilitat i ser intuïtiva perquè qualsevol usuari amb coneixements bàsics de l'aplicació siga capaç de fer-ne un ús correcte d'aquesta.

1. Pàgina web amb interfície actualitzada.
2. Pàgina web amb connexió segura.
3. Pàgina web amb suport per a múltiples idiomes.
4. Convertir un text, amb el format de llenguatge de marques, introduït en l'editor a un fitxer .xml
5. Convertir un arxiu de text pla, amb el format de llenguatge de marques, a un fitxer .xml
6. Gestió d'errors mitjançant fitxers .log
7. Registre d'usuari.
8. Inici de sessió d'usuari.
9. Recuperació d'un compte.
10. Borrat de compte.

3.1.1. Propòsit

La finalitat d'aquestes especificacions és definir amb precisió les funcionalitats que posseeix l'aplicació així com establir els límits i les funcionalitats que no es poden realitzar.

3.1.2. Referències

IEEE Recommended Practice for Software Requirements Specification. ANSI/IEEE 830/1998.

3.2 Requisits funcionals

Es desenvoluparà el diagrama d'usabilitat 3.1 en dues parts: si l'usuari no està registrat i si ho està.

3.2.1. Usuari no registrat

Canviar d'idioma.

Elegant l'idioma desitjat traduir la pàgina actual i les següents pàgines.

Convertir un text amb l'editor en línia.

Escriure en quadre de text les preguntes fent-ne ús del llenguatge de marques proporcionat, afegir el títol de la bateria de preguntes i dotar de nom a l'arxiu generat.

Convertir un text carregant un arxiu.

Adjuntar un fitxer escrit completament amb text pla i caràcters Unicode, afegir el títol de la bateria de preguntes i dotar de nom a l'arxiu generat.

Mostrar errors en una conversió.

En cas d'error al convertir el text, la pàgina web mostrarà un missatge d'error informant l'usuari de la línia on s'ha produït la fallada.

Conversió completada.

La pàgina web informarà que el procés ha finalitzat satisfactòriament i l'arxiu està disponible per a descarregar.

Registre d'usuari.

Registrar un nou usuari amb un nom d'usuari disponible, una contrasenya i un correu electrònic per a recuperar les credencials d'accés.

3.2.2. Usuari registrat.

Les mateixes funcions i altres addicionals tenen els usuaris registrats respecte als no registrats. Les addicionals són:

Inici de sessió.

Iniciar sessió amb l'usuari i contrasenya corresponents.

Gestió de credencials.

Recuperació de les credencials donat un correu vàlid que estiga associat a un compte de l'aplicació.

Gestió del compte.

Canvi de credencials d'accés, usuari, contrasenya i correu de recuperació. Esborrar el compte i totes les dades associades a ell.

Gestió de fitxers.

Operacions bàsiques de creació, obtenció, edició i esborrat dels arxius de l'usuari.

3.3 Requisits no funcionals

Usabilitat

L'aplicació ha de ser intuïtiva i fàcil d'utilitzar.

Manteniment

El codi ha de ser el més llegible possible perquè un desenvolupador afegisca noves funcionalitats ràpidament.

Adaptabilitat

La pàgina web ha de mostrar-se igual independentment del navegador web utilitzat.

Seguretat

La pàgina ha de comptar amb una connexió segura.

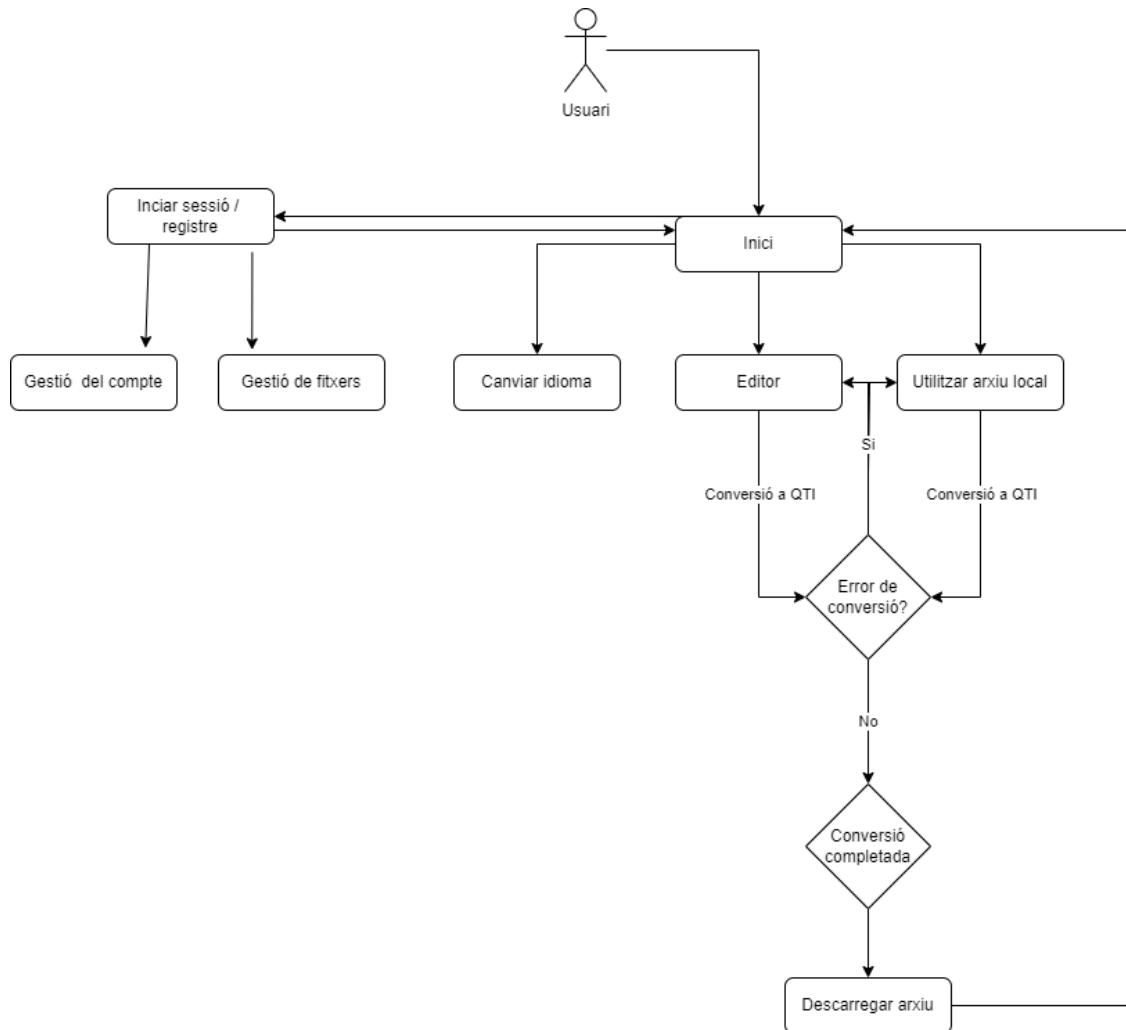


Figura 3.1: Diagrama d'aplicació - Disseny propi

CAPÍTOL 4

Planificació del treball

4.1 Metodologia

La metodologia seguida ha estat Scrum, distribuint els diferents problemes a resoldre en tasques concretes a realitzar en sprints de 3 setmanes. El procediment a seguir serà el següent:

1. Ús de l'aplicació existent.
2. Revisió i estudi del codi font.
3. Estudiar les diferents opcions disponibles.
4. Identificar quines operacions es poden realitzar localment i quines al servidor.
5. Programar en llenguatge JavaScript les funcions executades a la màquina del client.
6. Migrar el codi a PHP i processar les funcions exclusives del servidor.
7. Convertir un fitxer de llenguatge de marques fent ús de l'aplicació desenvolupada.
8. Emprant el mateix fitxer de marques, comparar el fitxer convertit a format QTI obtingut amb un fitxer QTI generat per l'anterior sistema.
9. Reestructurar i optimitzar el codi.
10. Comprovar que els canvis no afecten negativament al programa.

4.1.1. Sprints

La figura 4.1 mostra gràficament els sprints que es realitzaran.

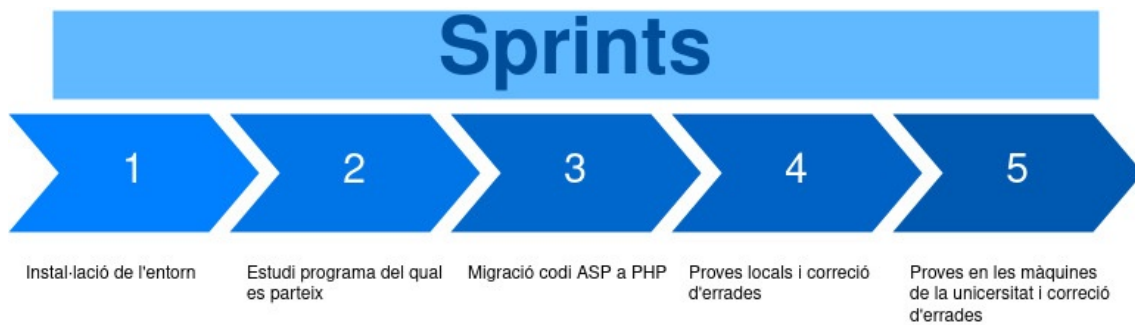


Figura 4.1: Planificació dels sprints - Disseny propi

Primer sprint

El primer sprint es dedicarà exclusivament a instal·lar en el disc dur de treball el sistema operatiu Ubuntu 20.04.5 amb les eines que s'utilitzaran, les quals són:

- El navegador Mozilla Firefox on es realitzaran les proves.
- L'editor de codi 'Visual Studio Code' amb les extensions de PHP corresponents per a la detecció d'errors i depuració.
- Servidor Apache per a desenvolupar localment sense necessitar un domini ni connexió a la xarxa.
- Per últim, configurar l'entorn amb els paràmetres i llibreries necessàries per al desenvolupament de l'aplició.

Segon sprint

El segon sprint serà el contacte amb el programa original del qual es parteix. S'estudiarà com funciona la pàgina web, associar les funcions proporcionades amb el codi font, formació bàsica en el llenguatge asp.net i equivalències amb el llenguatge PHP.

Tercer sprint

En el tercer sprint es procedirà a la migració del codi ASP.NET a PHP, simultàniament a la migració es tindrà en compte possibles seccions que puguin ser millorades mitjançant tècniques de reutilització de codi, separació de tasques amb l'ús de funcions, quan siga possible separar en fitxers el codi i millorar la llegibilitat del codi.

Quart sprint

El quart sprint se centrarà a convertir bateries de preguntes en l'antic sistema i el nou, executat localment al servidor Apache, per a comparar els fitxers resultants. En cas que existisquen diferències notables entre els dos fitxers donarà lloc a la fase de proves on s'utilitzaran eines de depuració com punts de parada en el codi o execució pas a pas per a solucionar l'error.

Cinqué sprint

L'últim sprint es dedicarà a instal·lar l'aplicació en l'entorn de la universitat per a realitzar més proves i comprovar la funcionalitat. En cas d'errors s'iniciarà la fase de proves per a resoldre el problema.

CAPÍTOL 5

Anàlisi

5.1 Vista de l'usuari

En la fase d'anàlisi del projecte, s'han observat diversos punts que poden millorar:

Les pàgines per les quals es navega durant l'ús de l'aplicació tenen una bona estructura, els elements es veuen correctament distribuïts sense saturar al lector d'informació i mostrant clarament les opcions disponibles.

Per a complir l'objectiu de modernitzar la pàgina web es marquen tres punts principals:

1. Color: Les pàgines web soles contenen els colors blanc i negre. A pesar de ser una aplicació sòbria, es troba a faltar més varietat en la paleta de colors.



Figura 5.1: Pàgina d'inici original.

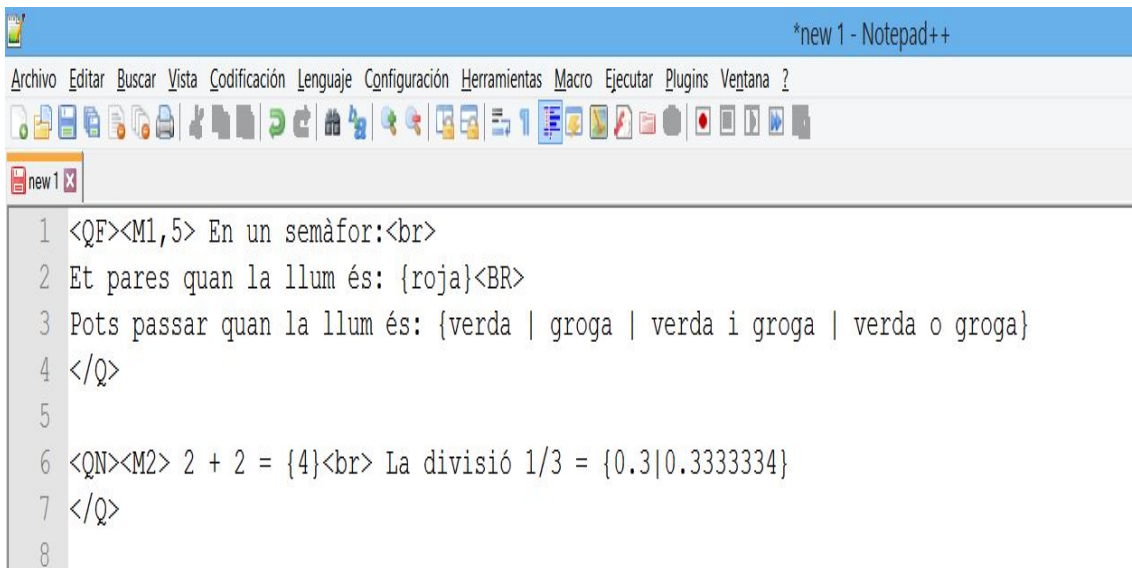
2. Idioma: Encara que la funcionalitat és prou senzilla, tenint en compte que hi ha poques pàgines per navegar i el text està molt optimitzat per a ajudar a l'usuari, una funcionalitat que permet internacionalitzar l'aplicació és oferint diversos idiomes entre els quals elegir.

3. Editor: En l'editor en línia es pot escriure o apegar el text que es desitja convertir, però es pot millorar mostrant el número de línia com es mostra en altres editors de text com 'Notepad++'. La figura 5.2 mostra l'editor sense número de línia i en la figura 5.3 es veu com el número de línia ofereix una ajuda visual útil.

```
<QF><M1,5> En un semàfor:<br>
Et pares quan la llum és: [roja]<BR>
Pots passar quan la llum és: [verda | groga | verda i groga |
verda o groga]
</Q>

<QN><M2> 2 + 2 = [4]<br> La divisió 1/3 = [0.3|0.3333334]
</Q>
```

Figura 5.2: Editor original sense número de línia.



```
1 <QF><M1,5> En un semàfor:<br>
2 Et pares quan la llum és: {roja}<BR>
3 Pots passar quan la llum és: {verda | groga | verda i groga |
4 verda o groga]
5 </Q>
6 <QN><M2> 2 + 2 = {4}<br> La divisió 1/3 = {0.3|0.3333334}
7 </Q>
8
```

Figura 5.3: Exemple de codi en l'editor Notepad++

5.2 Vista del desenvolupador

Com a conseqüència de les característiques del llenguatge ASP.NET, les pàgines web s'implementen amb dos fitxers: els d'extensió .aspx i .aspx.cs.

En un fitxer .aspx es troba la vista de l'usuari, és a dir, el codi HTML de la pàgina web, i en un fitxer .aspx.cs es troben les funcions a executar pel servidor.

Des del punt de vista del desenvolupador, quatre són els arxius principals:

1. Pàgina d'inici.

Arxiu: Default.html

La pàgina d'inici o pàgina principal, és la primera pàgina que veu l'usuari a partir de la qual es pot navegar a altres pàgines de l'aplicació.

2. Editor en línia.

Arxiu: Editor.aspx.cs

Al fitxer de l'editor es troba el quadre de text on s'introdueix el text a convertir així com el nom de l'arxiu que es generarà i el títol de la bateria que es guarda en el camp corresponent del fitxer xml. A escala de codi, executa multitud de funcions: obté les dades introduïdes per l'usuari, crea l'arxiu xml i redirecciona a la pàgina de descàrrega.

3. Càrrega d'arxiu en llenguatge de marques.

Arxiu: Upload.aspx.cs

La pàgina de càrrega d'arxiu és idèntica a la pàgina de l'editor llevat dels títols i en vegada de tindre un quadre de text, es té un botó per a carregar l'arxiu amb les preguntes en el llenguatge de marques al sistema. A escala de codi, també executa multitud de funcions, seguint la mateixa estratègia que a l'editor: crea l'arxiu xml i redirecciona a la pàgina de descàrrega.

4. Pàgina de descàrrega d'arxiu.

Arxiu: documentotraducido.aspx.cs

Un usuari que ha sigut redirigit a aquesta pàgina ja té disponible el fitxer al servidor per a la seua descàrrega. Per a descarregar-lo és suficient amb prémer el botó de descàrrega.

S'ha observat com hi ha algunes funcions de l'editor i la càrrega d'arxius que comparteixen codi. Al modificar un també s'ha de modificar l'altre per a mantindre els canvis.

També, que en la funció 'crearXml', que s'encarrega de convertir el text en llenguatge de marques en un fitxer xml al servidor, existeixen dos blocs de codi que s'encarreguen de guardar una sèrie d'atributs amb el seu valor per defecte, independentment del tipus de preguntes generades. La funcionalitat és correcta, però el codi es pot optimitzar.

5.3 Exemples de preguntes

En aquesta secció es mostren tres exemples d'arxius escrits amb el llenguatge de marques i la posterior vista en Poliformat. La figura 5.4 mostra un exemple de pregunta numèrica, la figura 5.5 mostra un exemple de pregunta múltiple i la figura 5.6 mostra un exemple de pregunta de completar els buits.

5.3.1. Pregunta numèrica

Text en llenguatge de marques

<QF><M1,5> Completa la serie: uno, {dos}, tres, {cuatro}. </Q>

<QF> En un semáforo:
 Te paras cuando la luz es: {roja}
 Puedes pasar cuando la luz es: {verde | amarilla | verde y amarilla | verde o amarilla} </Q>

<QF><M1,5>Gos en castellà és: {perro} </Q>

<QF>26 en base 10 se representa en hexadecimal por: 0x{1A | 0x1A} </Q>

EXÁMENES



Exámenes **Añadir** Baterías de preguntas Registro de eventos Informe de actividad Papelera

Preguntas: numerica

5 Preguntas existentes - 7 puntos totales

[Añadir parte](#) | [Vista previa](#) | [Imprimir](#) | [Configuración](#) | [Publicar](#)

Parte Default - 5 preguntas

[Copiar a la batería](#) | [Editar](#)

Añadir pregunta

Pregunta Respuesta numérica - puntos

[Borrar](#) | [Editar](#)

Las tres primeras potencias de dos son: ____, ____, y ____.
Respuesta correcta:1, 2, 4

Pregunta Respuesta numérica - punto

[Borrar](#) | [Editar](#)

Mitja quarta de bajoqueta són ____ Kg
Respuesta correcta:0.125

Figura 5.4: Pregunta numèrica en Poliformat

5.3.2. Pregunta multiple

Text en llenguatge de marques

<QM>La fórmula C₄H₁₀ corresponde al butano. <op><rc>Verdadero <op>Falso </Q>

<QM><M2>Un microcontrolador es un computador <op><rc>Sí <op>No </Q>

<QM><M2><MN0,35>Son filósofos: <op><rc> David Hume. <op>Homer Simpsons. <op><rc> Erasmo de Rotterdam. </Q>

Preguntas: multiple

3 Preguntas existentes - 5 puntos totales

[Añadir parte](#) | [Vista previa](#) | [Imprimir](#) | [Configuración](#) | [Publicar](#)

Parte Default - 3 preguntas

[Copiar a la batería](#) | [Editar](#)

Añadir pregunta

Pregunta Una correcta, seleccionar una. - punto

[Borrar](#) | [Edita](#)

La fórmula C₄H₁₀ corresponde al butano.

- A. Verdadero
- B. Falso

Respuesta correcta:A

Pregunta Una correcta, seleccionar una. - puntos

[Borrar](#) | [Edita](#)

Un microcontrolador es un computador

- A. Sí
- B. No

Respuesta correcta:A

Figura 5.5: Pregunta de selecció múltiple en Poliformat

5.3.3. Pregunta de completar los huecos

Texto en lenguaje de marcas

<QF><M1,5> Completa la serie: uno, {dos}, tres, {cuatro}. </Q>

<QF> En un semáforo:
 Te paras cuando la luz es: {roja}
 Puedes pasar cuando la luz es: {verde | amarilla | verde y amarilla | verde o amarilla} </Q>

<QF><M1,5>Gos en castellà és: {perro} </Q>

<QF>26 en base 10 se representa en hexadecimal por: 0x{1A | 0x1A} </Q>

EXAMENES 🔗 🗑️

Exámenes **Añadir** Baterías de preguntas Registro de eventos Informe de actividad Papelera

Preguntas: huecos - 2

4 Preguntas existentes - 5,5 puntos totales

[Añadir parte](#) | [Vista previa](#) | [Imprimir](#) | [Configuración](#) | [Publicar](#)

Parte Default - 4 preguntas [Copiar a la batería](#) | [Editar](#)

Añadir pregunta

Pregunta Completar los espacios en blanco - puntos [Borrar](#) | [Edita](#)

Completa la serie: uno, ____, tres, ____.
Respuesta correcta: dos, cuatro

Pregunta Completar los espacios en blanco - punto [Borrar](#) | [Edita](#)

En un semáforo:
Te paras cuando la luz es: ____

Figura 5.6: Pregunta de completar los huecos en Poliformat

CAPÍTOL 6

Disseny

6.1 Vista d'usuari

Per a assolir l'objectiu de modernitzar la pàgina web, es van idear diferents propostes però cap s'adequava al propòsit de l'aplicació, a l'annex B es troben alguns esbossos creats a la fase de disseny. Per aquesta raó, en vegada de construir la pàgina web amb altre disseny, s'ha establert la base anterior a la que introduir canvis mínims que afecten a l'experiència de l'usuari.

Originàriament, a la pàgina principal predominaven els colors blanc, negre i tonalitats de gris, figura 6.1. Una de les idees per a canviar l'aparença era col·locar una imatge al fons, però van sorgir problemes amb la llegibilitat del text i es va descartar aquesta aproximació.

Una altra idea era utilitzar un color sòlid com a fons. Tot i que la icona de QTI encaixava amb gran varietat de colors, el text es tornava cada vegada més il·legible. Es van fer proves canviant el color del text, però a causa de les modificacions l'aplicació no semblava destinada als docents i també es va descartar la idea.

Finalment, establert que el color de la lletra havia de ser negre, inspirat per la pàgina de la UPV 6.2, es va decidir un degradat roig-blanc, on el fons roig cobria una quarta part de la pàgina permetent introduir altre color al registre, però sense modificar la facilitat de lectura.

També s'ha afegit la funcionalitat de canviar a un fons de color roig amb lletres blanques quan se situa el cursor per sobre. Aquests canvis es veuen en el disseny final 6.3.

6.2 Vista del desenvolupador

En el llenguatge ASP.NET és necessari separar els fitxers amb codi HTML dels fitxers amb les funcions del sistema. En PHP no és necessari i es pot encapsular al mateix fitxer .php el codi HTML i el codi PHP.



Figura 6.1: Pàgina d'inici anterior.

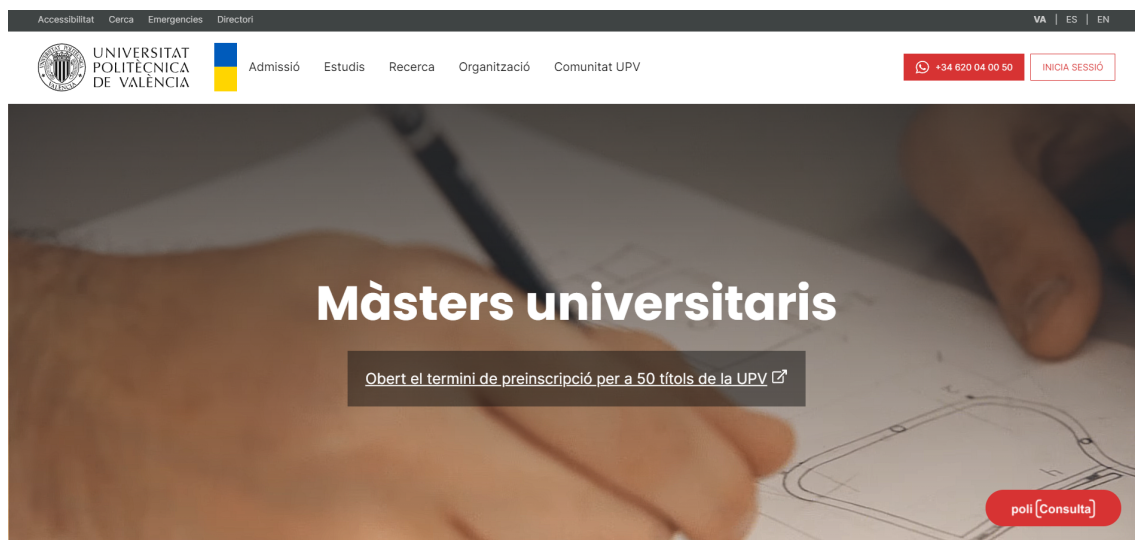


Figura 6.2: Pàgina d'inici de la UPV.

6.3 Arquitectura de tres capas

En aquest projecte s'ha optat per una arquitectura en tres capes que separa la nostra aplicació en tres capes diferenciades: capa de presentació, capa de negoci i capa d'accés a dades.

La figura 6.4 representa gràficament les relacions entre les tres capes:

6.3.1. Capa de presentació

El nivell de presentació és la interfície d'usuari i la capa de comunicació de l'aplicació, on l'usuari final interactua amb l'aplicació. El seu objectiu principal és mostrar informació a l'usuari i recopilar informació de l'usuari. Aquest nivell

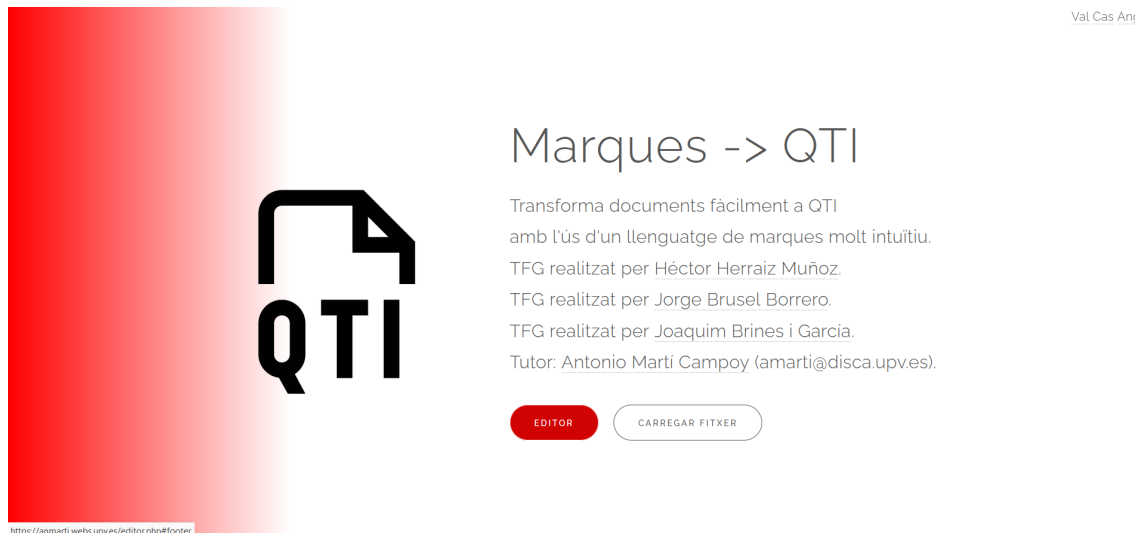


Figura 6.3: Pàgina d'inici actual.

superior es pot executar en un navegador web, com una aplicació d'escriptori o una interfície gràfica d'usuari (GUI), per exemple. Els nivells de presentació web normalment es desenvolupen utilitzant HTML, CSS i JavaScript. Les aplicacions d'escriptori es poden escriure en diversos llenguatges, depenent de la plataforma.

6.3.2. Capa d'aplicació

Aquesta és la capa on es realitzen les funcions de l'aplicació, la conversió de marques a xml en format QTI i la posterior descàrrega.

El nivell d'aplicació, també conegut com el nivell lògic o el nivell mitjà, és el nucli de l'aplicació. En aquest nivell, es processa la informació recopilada en el nivell de presentació, a vegades contra una altra informació en el nivell de dades, usant la lògica empresarial, un conjunt específic de regles de negoci. El nivell d'aplicació també pot afegir, suprimir o modificar dades en el nivell de dades. El nivell d'aplicació normalment es desenvolupa usant Python, Java, Perl, PHP o Ruby, i es comunica amb el nivell de dades utilitzant cridades d'API.

S'ha elegit PHP com a llenguatge de programació per a aquesta capa.

6.3.3. Capa d'accés a les dades

El nivell de dades, a vegades denominat nivell de base de dades o nivell d'accés a dades, és on s'emmagatzema i gestiona la informació processada per l'aplicació. Pot ser un sistema de gestió de bases de dades relacional com PostgreSQL, MySQL, MariaDB, Oracle, DB2, Informix o Microsoft SQL Server, o un servidor de bases de dades NoSQL com Cassandra, CouchDB o MongoDB. En una aplicació de tres capes, tota la comunicació travessa la capa d'aplicació, la capa d'accés a les dades no pot ser accedida directament des de la capa de presentació.

Per a aquest projecte s'ha elegit 'phpmyadmin' per ser una eina de gestió de base de dades versàtil i amb gran suport de la comunitat. La gestió d'usuaris i dels seus fitxers resideix en aquesta capa. Es conta amb dues taules de base

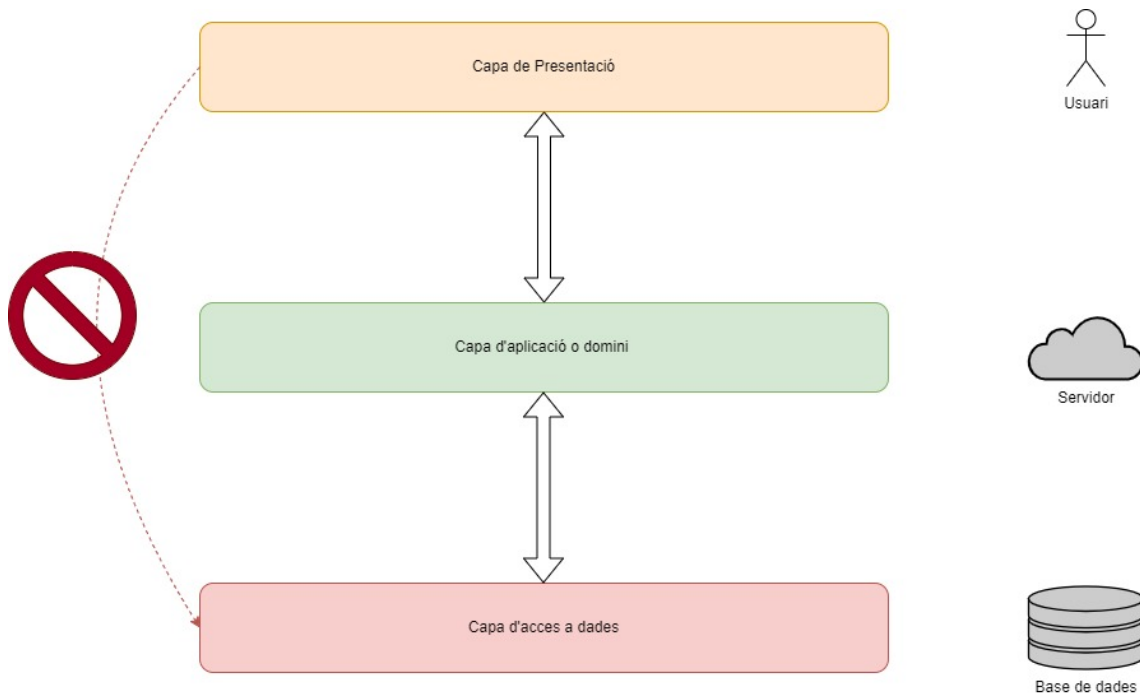


Figura 6.4: Arquitectura de tres capes - Disseny propi

de dades: una amb els comptes dels usuaris i altra amb els fitxers guardats. La taula d'usuaris conté quatre atributs que són: identificador únic, nom d'usuari, contrasenya i correu de recuperació. La taula de fitxers conté tres atributs que són: identificador únic, identificador de l'usuari propietari i nom del fitxer.

6.4 Model client - servidor

El model client-servidor és similar a l'arquitectura anterior de tres capes, però ara soles està representat el client i el servidor sense definir estrictament l'àmbit de cadascun. És una estructura d'aplicació distribuïda que divideix les tasques o càrregues de treball entre proveïdors i sol·licitants de serveis, denominats servidors i clients, respectivament.

6.4.1. Client

És la part del model que inicia i desencadena les accions. Mitjançant l'ús de peticions el client es comunica amb el servidor mitjançant una pàgina web. En el nostre cas, el client representa un usuari de l'aplicació que accedeix a la pàgina web i converteix un text a llenguatge de marques.

Des del punt de vista del client:

- El client executarà exclusivament funcions a la capa de presentació.
- El client envia una sol·licitud de servei a un servidor.
- El client rep una resposta del servidor i la processa.

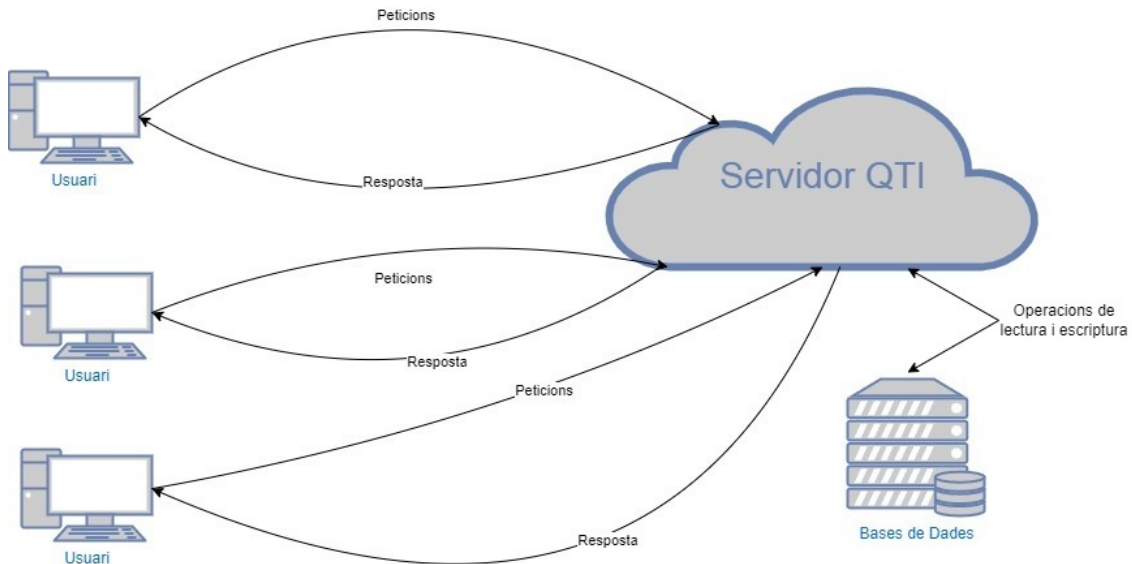


Figura 6.5: Model client-servidor - Disseny propi

- El client pot interactuar amb diversos servidors simultàniament
- El client depèn del servidor per a obtenir els serveis sol·licitats, però té el control sobre l'inici de la sol·licitud.

6.4.2. Servidor

És la part del model que rep les peticions de la part del client, processa les dades i executa les funcions programades.

Des del punt de vista del servidor:

- El servidor executa funcions de la capa d'aplicació i d'accés a les dades.
- El servidor espera les peticions dels clients i proporciona una resposta o servei.
- El servidor pot atendre diversos clients simultàniament.
- El servidor proporciona el servei sol·licitat, però no té control sobre l'inici de les peticions.
- El servidor pot proporcionar serveis amb estat en funció dels requisits o sense.

CAPÍTOL 7

Implementació

A l'hora de la implementació, per a assegurar el correcte funcionament del codi primerament s'ha determinat escriure el codi PHP mantenint, amb el major grau possible, l'estructura utilitzada en l'original. Durant aquesta fase s'han mantingut els noms de les variables i l'esquema utilitzat.

Per a la conversió del text escrit en llenguatge de marques a un fitxer .xml amb la distribució dels elements seguint l'esquema d'etiquetes i els sagnats corresponents, una primera idea va ser guardar en una variable global el text final i que en cada avanç del codi es guardés en una nova variable el nou text convertit, permetent concatenar-la a la variable global fins que tot el codi s'ha executat, s'ha processat tot el text i l'únic que resta és guardar en un fitxer xml la variable global.

Aquesta idea afegia complexitat addicional al codi per estar manipulant cadenes de text, les quals també comptaven amb cadenes de text internes i al procés d'unió de variables s'havia de tindre un control del sagnat de totes les línies per a obtenir un fitxer amb l'organització dels elements adequada.

La segona idea i la implementada finalment a l'aplicació, és l'ús de l'eina 'DOMDocument'.

7.1 DOMDocument

El DOMDocument en PHP és una classe que proporciona una interfície de programació per crear i manipular documents amb format XML i HTML. Forma part de l'extensió PHP DOM, que és una implementació de l'estàndard Document Object Model (DOM) per a documents XML i HTML.

La classe DOMDocument s'utilitza per crear i manipular objectes de document, que representen un document XML o HTML com un arbre de nodes. Cada node de l'arbre està representat per un objecte d'un tipus específic, com DOMElement, DOMText, DOMComment, etc. Aquests nodes es poden accedir i modificar utilitzant diversos mètodes proporcionats per la classe DOMDocument, com ara createElement(), createTextNode(), appendChild(), etc.

La classe DOMDocument s'utilitza comunament en aplicacions PHP que necessiten generar o manipular documents XML o HTML. És especialment útil per

a tasques com analitzar i validar documents XML, generar pàgines HTML dinàmiques i extraure dades de pàgines web.

Com a conclusió, la classe DOMDocument en PHP proporciona una manera potent i flexible de crear, manipular i travessar documents XML i HTML, convertint-lo en una eina valuosa per a desenvolupadors que treballen amb aplicacions web que requereixen la gestió i processament de dades estructurades.

7.1.1. Estructura

El codi s'estructura en diferents fitxers, cadascun amb funcionalitats relacionades i programades amb llenguatges diferents:

- El bloc de la pàgina web escrita amb llenguatge html.
- Els estils que donen format al codi html en format css.
- Les funcions que s'executen en la màquina del client en llenguatge javascript.
- I les funcions que han d'executar-se al servidor en llenguatge php.

7.1.2. Suport de múltiples idiomes

Per a afegir la funcionalitat de poder canviar d'idioma la pàgina web, primerament s'ha creat un directori anomenat 'lang' on es troben tres fitxers php amb les traduccions corresponents a cada idioma, actualment hi ha suport per al Valencià, Castellà i Anglès, però si en el futur fora necessari afegir altres idiomes com el Francès, Italià, etc. sols caldria crear un nou fitxer amb la traducció al nou idioma, afegir el fitxer al selector d'idioma i totes les pàgines es traduirien.

Tots els textos utilitzats en els fitxers .php s'han guardat en variables en vegegada del mateix text, i són aquestes variables les que es defineixen en els fitxers d'idioma anteriorment descrits. També és possible canviar l'idioma en totes les pàgines per les quals es navega a l'hora d'utilitzar l'aplicació.

Aquesta metodologia garanteix reduir temps de desenvolupament futur i mantindre la qualitat del codi.

7.1.3. Gestió de fitxers al servidor

Perquè l'usuari de l'aplicació siga capaç de descarregar l'arxiu que acaba de convertir, és necessari que es guarde primerament en el servidor, perquè és qui realitza el procés de conversió. Una opció és guardar els arxius amb el nom que l'usuari ha establert i seguidament al prémer el botó de descàrrega, es llig el fitxer del servidor per a guardar-lo en la màquina del client.

En cas que d'existir un arxiu amb el mateix nom, se sobreescriu l'arxiu. Amb aquesta estratègia s'evita múltiples arxius de nom igual que s'acumulen al servidor ocupant espai i recursos.

Però té una gran vulnerabilitat de seguretat i privacitat, pot donar-se el cas que dos usuaris realitzen la conversió de fitxers amb nom idèntic, un utilitza l'aplicació, però es manté en la pàgina de descàrrega sense guardar, a continuació altre usuari realitza la conversió i el seu codi sobreescriu el del primer usuari. En aquest cas el primer usuari no soles hauria perdut el seu fitxer sinó, el més greu, podria descarregar l'arxiu d'altre usuari. Aquest tipus d'errors són totalment inassumibles en una aplicació tan sensible.

La figura 7.1 mostra gràficament el problema descrit.

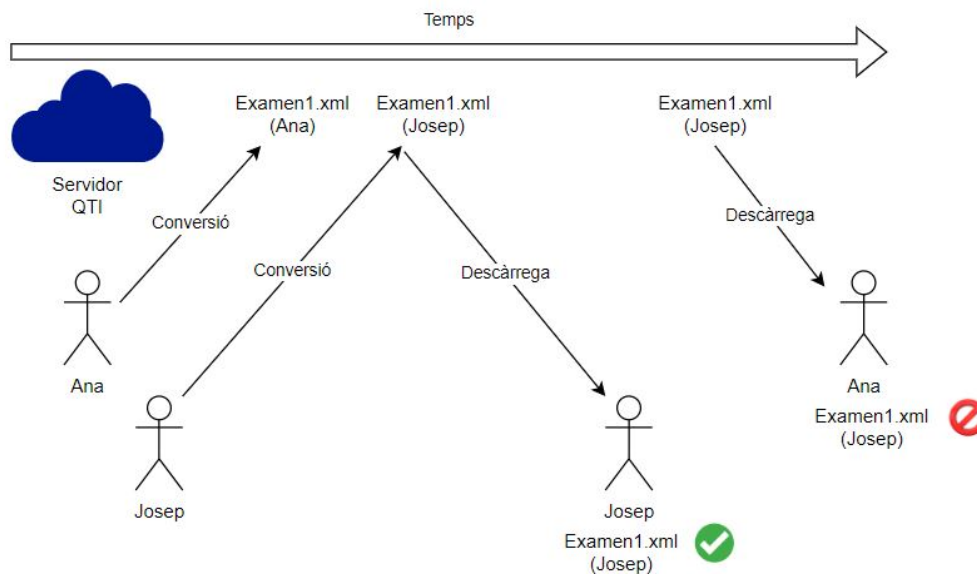


Figura 7.1: Vulnerabilitat en la gestió de fitxers en PHP - Disseny propi

S'hi han pres dues mesures per a solucionar la vulnerabilitat.

1. Afegir al nom de l'arxiu creat al servidor l'hora de la creació del fitxer convertida en segons, la qual és un número de deu xifres.
2. Afegir a continuació de l'hora de creació un valor aleatori que pot ser d'una fins a deu xifres.

Amb aquestes mesures, prenent l'exemple anterior com a referència, ja no es generarà l'arxiu examen1.xml sinó examen1_01234567891.xml amb una llargària de nom variable dependent del segon valor aleatori. Es considera en aquest punt més segur el sistema de gestió d'arxius, que a pesar de ser altament millorable mitjançant tècniques d'enciptació, per exemple, per al propòsit d'aquest projecte és suficient.

Gestió d'arxius antics al servidor

Un problema que es té al servidor, és que els arxius generats no s'esborren, per tant, amb el pas del temps el seu nombre creix i utilitzen recursos innecessàriament. Una primera aproximació va ser d'esborrar l'arxiu quan l'usuari el

guardara, però com que l'arxiu s'havia esborrat del servidor, l'usuari, en cas que desitjara descarregar-lo de nou no podria i hauria de repetir tot el procés de conversió. S'ha sacrificat la gestió d'arxius per a millorar l'experiència d'ús de l'usuari.

Ha sigut implementat un fitxer php que, al ser executat, esborra tots els documents xml que tenen una diferència entre l'hora de creació i l'actual de més d'una hora. Es decideix que una hora és suficient per a poder descarregar l'arxiu convertit.

Aquest fitxer ha de ser executat manualment o mitjançant l'ús de 'Cron', on es podria establir que s'executara el codi cada hora. L'ús o no del codi proporcionat ha de ser decidit pels administradors del sistema.

7.2 Eines i tecnologies utilitzades

Per al desenvolupament de l'aplicació s'han emprat diverses eines. Aquestes són:

7.2.1. Diagrams.net

La pàgina web diagrams.net ha resultat realment útil per a crear els diagrames de flux mostrats al llarg de la memòria.

7.2.2. Visual Studio Code

Editor de codi per al desenvolupament de l'aplicació amb les extensions 'PHP Intelephense' i 'PHP Debug' per a detectar errades durant la programació i durant l'execució del codi.

7.2.3. Servidor HTTP Apache

Servidor lliure i de codi obert apte per a sistemes UNIX i Windows. S'hi ha instal·lat en la màquina local per al desenvolupament de l'aplicació.

7.2.4. CSSGradient

La pàgina web <https://cssgradient.io/> facilita la creació d'un degradat de colors directament a llenguatge css. S'ha utilitzat en la creació del fons de l'aplicació.

7.2.5. Navegadors

S'han emprat els navegadors Mozilla Firefox i Opera durant tot el temps del projecte.

CAPÍTOL 8

Proves

Al llarg del desenvolupament de l'aplicació s'han realitzat diferents proves i han sigut corregits els errors trobats.

En la fase de proves s'ha interactuat amb l'aplicació des del punt de vista de l'usuari. Amb aquesta finalitat s'han col·locat els fitxers php en el servidor, fent ús de l'url:

<https://anmarti.webs.upv.es/index.php>

Les proves realitzades han sigut:

1. Comprovar que la versió de PHP és compatible amb la utilitzada per al desenvolupament.
2. Comprovar la navegació entre pàgines.
3. Mostrar números de línia en l'editor.
4. Al canviar d'idioma es canvien tots els textos i es manté l'idioma escollit al canviar de pàgines.
5. Convertir un text des de l'editor.
6. Convertir un text carregant un fitxer al sistema.

Han sigut utilitzats els mateixos fitxers de la fase d'anàlisi 5.4, 5.5 i 5.6 amb un resultat satisfactori.

CAPÍTOL 9

Conclusions

9.1 Conclusions

Les conclusions que s'extrauen d'aquest projecte són les següents: Tots els objectius s'han assolit adequadament excepte el de tindre una gestió d'usuaris.

Des del punt de vista d'un usuari de l'aplicació, s'ha modernitzat l'aspecte visual de la pàgina afegint color al fons i als botons, també es mostra el número de línia en l'editor. Mitjançant botons es pot seleccionar fàcilment l'idioma de la pàgina canviant també les següents pàgines de l'aplicació.

Des del punt de vista d'un desenvolupador, s'ha reorganitzat el codi i s'ha millorat la seua llegibilitat, manteniment i possibilita ampliacions futures.

Aquest projecte defineix una base sobre la qual es podrà ampliar i millorar l'aplicació.

9.2 Projectes futurs

A continuació es mostren alguns dels projectes que poden sorgir utilitzant aquest treball com a base.

9.2.1. Gestió d'usuaris

Implementar una gestió d'usuaris on es guarde la informació en una base de dades.

S'han de comparar els avantatges i desavantatges que comporten la utilització d'una base de dades relacional i una base no relacional. A més a més del xifratge de credencials d'accés mitjançant tècniques d'encriptació com l'ús de la funció criptogràfica 'hash'.

9.2.2. Gestió de fitxers

En cas de no tindre usuaris, estudiar diferents estratègies per a evitar emmagatzemar arxius al servidor que no s'empraran, com el esborrat automàtic al des-

carregar l'arxiu, esborrat periòdic de tots els arxius amb una antiguitat definida o esborrat manual per part dels administradors del sistema.

Aquest punt és interessant quan es té un sistema de gestió d'usuaris estable i funcional. Oferir als usuaris la capacitat de gestionar els arxius al servidor associats al seu compte. Operacions de lectura, escriptura, edició i esborrat sobre aquests arxius.

9.2.3. Previsualitzador de preguntes

Afegir a la pàgina web una secció en la qual a l'introduir el text en llenguatge de marques o en format xml mostre com es veuria a Poliformat.

9.2.4. Afegir tipus de preguntes i opcions

Afegir més marques amb noves funcionalitats o millorar les existents. Addicionalment, opcions com l'exclusió mútua o la distinció de majúscules i minúscules prenen valors predefinitos, la finalitat del futur treball és oferir a l'usuari canviar els valors que desitge i la resta mantindre el valor per defecte.

9.3 Valoració personal.

Personalment, ha sigut educatiu en molts aspectes treballar en aquest projecte.

Fa un any vaig començar les pràctiques d'empresa en una companyia de telefonia on m'encarregava de tasques de manteniment i desenvolupament de la seua aplicació web. Aquesta aplicació utilitza el llenguatge PHP i m'ha servit per a tindre una base de programació en aquest llenguatge. Per tant, tinc experiència amb projectes que estaven a l'inici del desenvolupament i formava part del procés de creació com també projectes establits on m'encarregava de solucionar errors al codi o afegir noves funcionalitats, però no de migrar d'un llenguatge de programació a un altre. Respecte als llenguatges de programació utilitzats, en PHP ja tenia una formació a l'hora de crear el programa però no pel que fa a ASP, ha suposat un repte en conjunt i principalment la fase d'anàlisi del codi al mateix temps que es buscaven noves estratègies per a millorar-lo.

No s'han realitzat totes les millores que voldria, però estic satisfet amb el treball aconseguit, no tinc dubtes que altre estudiant podrà continuar aquest projecte i millorar el producte final.

Des del punt de vista acadèmic, el coneixement proporcionat per nombroses assignatures m'han ajudat al llarg del procés de desenvolupament i sense el qual no hauria pogut dur a terme aquest treball. Al marge de les assignatures que proporcionaven una base per al desenvolupament de programari, dos van destacar:

- Gestió de Projectes: A l'inici i durant el desenvolupament, s'observava el temps dedicat a cada tasca i el posterior ajust de temps respecte a la planificació inicial.

- Interfícies Persona Computador: A l'hora de modernitzar la pàgina web han sigut molt útils els coneixements apresos. L'aplicació original ja comptava amb una bona estructura i, per tant, els canvis han sigut mínims però significatius.

Agraïments

El camí per a completar aquest projecte ha sigut difícil i complicat, però m'ha ajudat a créixer tant en l'àmbit professional com personal. No tinc dubtes que els coneixements adquirits seran útils en altres etapes.

Done les gràcies al meu tutor Antonio Martí per oferir-me participar en un treball com aquest i sense el qual no hauria pogut desenvolupar l'aplicació ni redactar aquesta memòria. També a tots els amics que directament o indirectament m'han recolzat tot aquest procés, especialment a Òscar, Johnny, Alfonso, Raúl, Carles, Jaume, Miguel i Xavi. I per últim, i no menys important, a la meua família, per ajudar-me i estar sempre al meu costat quan els necessitava encara que no ho sabera.

A tots, moltes gràcies.

Bibliografía

- [1] Convertidor de lenguaje de marcas a formato QTI/IMS Herraiz Muñoz, H. (2015) <https://riunet.upv.es/handle/10251/55690>.
- [2] Sitio web para la conversión de lenguaje de marcas a formato QTI/IMS Brusel Borrero, J. (2018). <https://riunet.upv.es/handle/10251/106892>.
- [3] Documentació PHP <https://www.php.net>.
- [4] Clase DomDocument de PHP <https://www.php.net/manual/es/class.domdocument.php>.
- [5] What is ASP.NET <https://dotnet.microsoft.com/es-es/learn/aspnet/what-is-aspnet>.
- [6] Frontend style guides. <https://www.nngroup.com/articles/front-end-style-guides/>.
- [7] IMS QTI Ims question & test interoperability specification. *IMS Global Learning Consortium*, 2005.
- [8] Servidor apache <https://httpd.apache.org>.
- [9] Shneiderman, B. y Plaisant, C. *Designing the User Interface. Pearson 5th ed.*, 2010.
- [10] Thomas M. Connolly, Carolyn E. Begg, Addison-Wesley *Database Systems: A Practical Approach to Design, Implementation and Management, 5/E*, 2010.
- [11] B. Ward, *How Linux works: what every superuser should know*, No Starch Press, 2015.

APÈNDIX A

Llenguatge de marques propi

El convertidor.

<https://qti.webs.upv.es/brines/index.php>

Preguntas de rellenar huecos.

`<QF><M1,5>` En un semáforo:`
`

Te paras cuando la luz es: {roja}`
`

Puedes pasar cuando la luz es: {verde | amarilla | verde y amarilla | verde o amarilla}

`</Q>`

<code><QF></code>	Inicio de pregunta.
<code><M1,5></code>	Puntuación para respuesta correcta
<code>
</code> <code>
</code>	Salto de línea. No es necesario
{ roja }	Respuesta correcta. Siempre entre llaves.
{verde amarilla}	Respuesta correcta múltiples opciones.
<code></Q></code>	Fin de pregunta.
Observaciones	No se distingue entre mayúsculas y minúsculas.
	No se puede activar "Mutuamente exclusivas"
	Se ignoran espacios en blanco.

Preguntas de respuesta numérica.

`<QN><M2>` $2 + 2 = \{4\}$ `
` La división $1/3 = \{0.3|0.3333334\}$ `</Q>`

<code><QN></code>	Inicio de pregunta.
<code><M2></code>	Puntuación para respuesta correcta
<code>
</code>	Salto de línea.
{ 4 }	Respuesta correcta. Siempre entre llaves.
{0.3 0.3333334}	Rango para la respuesta correcta
<code></Q></code>	Fin de pregunta.
Formato de número	Cualquiera aceptado por PoliformaT: entero; real con separador de punto o coma; indicador de signo, coma flotante, números complejos.
	Cualquier símbolo que indique porcentaje, moneda, unidad de medida o base de representación debe ir fuera de las {}

Preguntas de respuesta múltiple.

<QM><M1><MN0,5>Un microcontrolador es un computador.

<op><rc>Sí.

<op>No. </Q>

<QM>	Inicio de pregunta.
<M2>	Puntuación para respuesta correcta
<MN0,5>	Puntuación que se resta para respuesta errónea
<op>	Respuesta a elegir. ¡En minúsculas!
<rc>	Respuesta correcta. ¡En minúsculas! Al menos una.
</Q>	Fin de pregunta.
Observaciones	Las respuestas siempre se aleatorizan. Se sugiere la fórmula “Ninguna otra respuesta es correcta”
	Se puede definir tantas respuestas correctas como se desee, pero al menos tiene que haber una.

Dar formato

 	Salto de línea. Equivalente a la tecla INTRO
texto	Texto en negrita
<I>texto</I>	Texto en cursiva
<U>texto</U>	Texto subrayado
_{texto}	Subíndice
^{texto}	Superíndice
	Imagen (cuidado con el nombre)
nombre	Enlaces web
Ω	Símbolos HTML.
{	Es importante terminar el código con ; https://dev.w3.org/html5/html-author/charref
Observaciones	Para insertar { o } como texto de la pregunta y no como respuesta, hay que usar los código HTML: { y } (incluyendo el ;)
	Es posible que algunos códigos HTML dependan del operativo o navegador utilizado.
	El alumno puede ver el nombre de la imagen.
	El enlace a una imagen en el sitio Poliformat de una asignatura puede cambiar cada año.

APÈNDIX B

Esbossos de la pàgina web

Per a la tasca de modernitzar la pàgina web s'han realitzat diversos test i provat diferents combinacions de colors. Seguit es mostren alguns esbossos creats durant el procés. En la figura **B.1** als botons se li ha afegit un contorn roig però no és un canvi significatiu. A la figura **B.2** els botons destaquen, però sembla buit el fons. S'ha continuat amb la **B.3** però el color saturava la pàgina. El mateix ocorre amb la **B.4**. A la figura **B.5** el text és difícil de llegir. A la figura **B.6** es torna a un fons de color però amb un gradient vertical, es decideix que és millor en horitzontal pel fet que la imatge de QTI se situa a l'esquerra. L'esbós **B.7** és el que més s'adequa al tipus d'aplicació. Com sembla que el color és un poc intrusiu al text, en el disseny final s'ha acurtat el gradient, figures **B.8** i **B.9**.



Marques -> QTI

Transforma documents fàcilment a QTI
amb l'ús d'un llenguatge de marques molt intuïtiu.
TFG realitzat per [Héctor Herraiz Muñoz](#).
TFG realitzat per [Jorge Brusel Borrero](#).
TFG realitzat per [Joaquim Brines i García](#).
Tutor: [Antonio Martí Campoy \(amarti@disca.upv.es\)](#).

EDITOR

CARREGAR FITXER

Figura B.1: Esbós 1



Marques -> QTI

Transforma documents fàcilment a QTI amb l'ús d'un llenguatge de marques molt intuïtiu.

TFG realitzat per [Héctor Herraiz Muñoz](#).

TFG realitzat per [Jorge Brusel Borrero](#).

TFG realitzat per [Joaquim Brines i García](#).

Tutor: [Antonio Martí Campoy \(amarti@disca.upv.es\)](mailto:amarti@disca.upv.es).

[EDITOR](#)[CARREGAR FITXER](#)

Figura B.2: Esbós 2

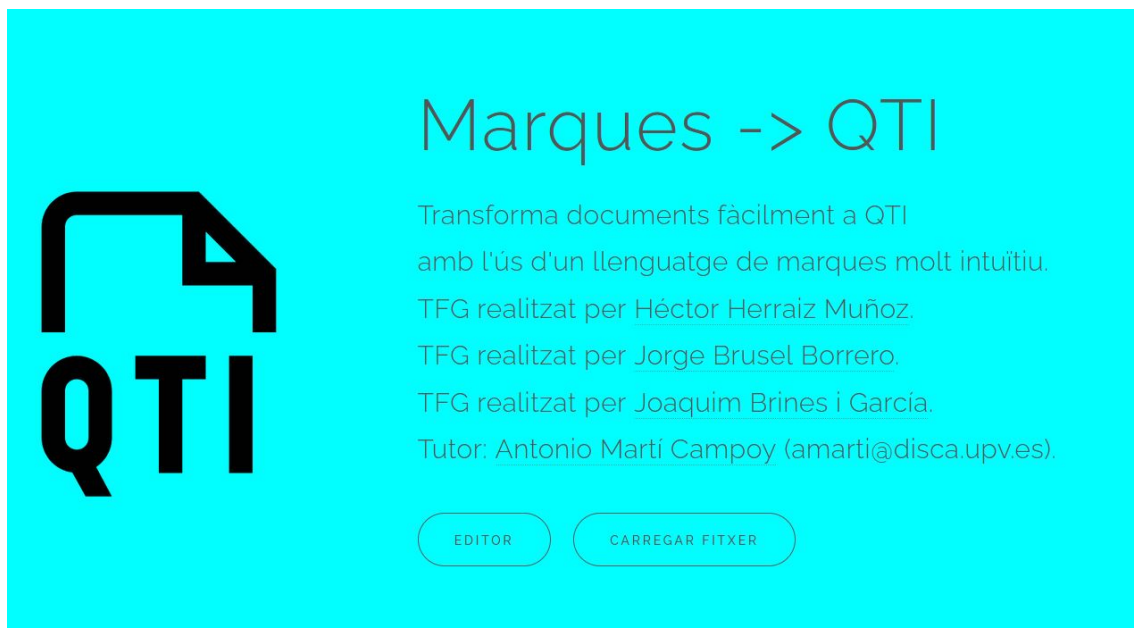


Figura B.3: Esbós 3



Figura B.4: Esbós 4



Figura B.5: Esbós 5



Figura B.6: Esbós 6



Figura B.7: Esbós 7

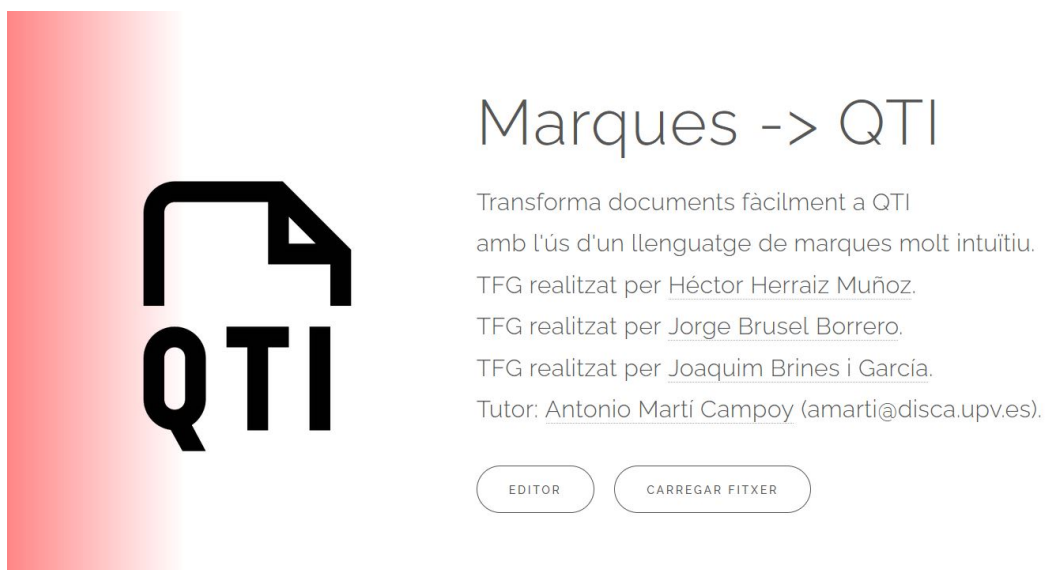


Figura B.8: Disseny final.



Figura B.9: Disseny final.

APÈNDIX C

Guía d'ús

En aquest apèndix es mostrarà gràficament el procés d'utilització de l'aplicació.



Figura C.1: Pagina d'inici.

L'usuari tria si vol carregar un fitxer o escriure o pegar directament en l'editor.

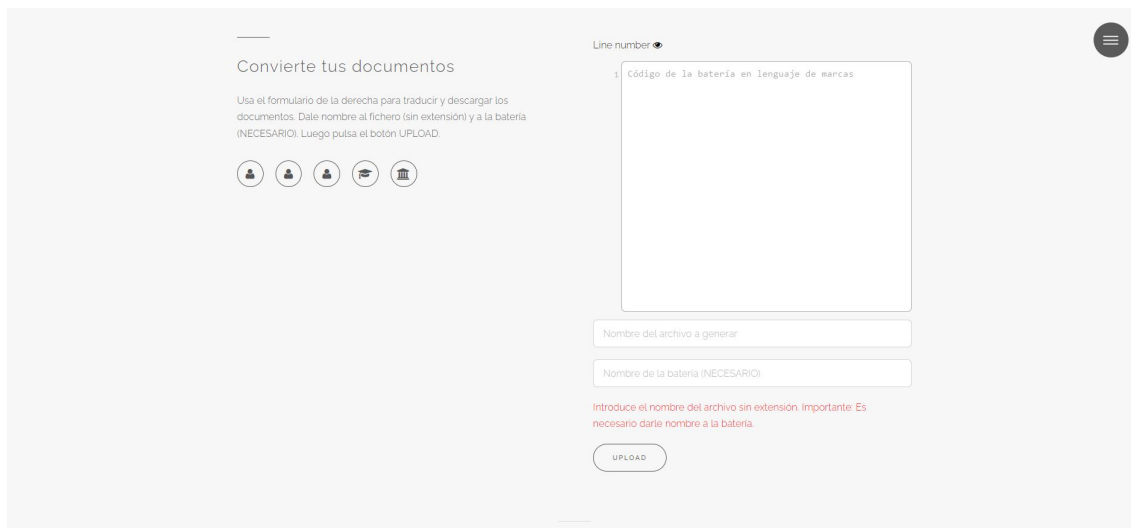


Figura C.2: Editor.

L'usuari pot escriure o pegar les qüestions, donar nom al fitxer resultat i el nom de la bateria de preguntes.

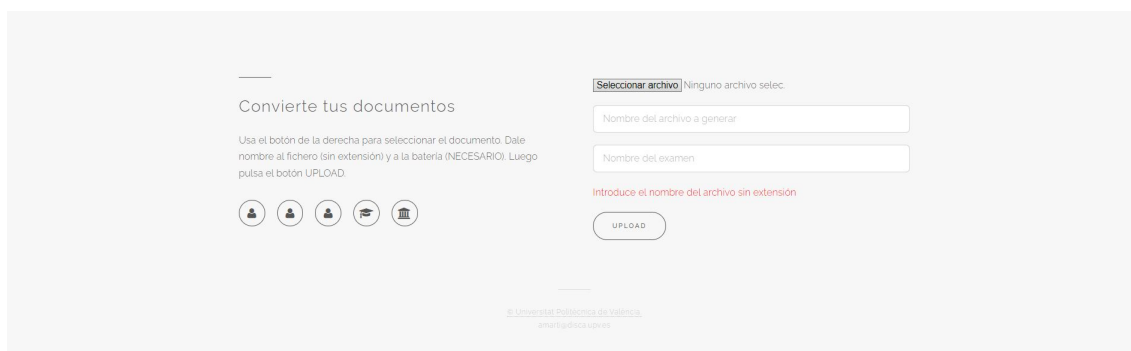


Figura C.3: Càrrega d'arxiu.

L'usuari pot carregar el fitxer directament, donar nom al fitxer resultat i el nom de la bateria de preguntes.



Figura C.4: Descàrrega d'arxiu.

La conversió s'ha realitzat i l'usuari pot descarregar l'arxiu.

APÈNDIX D

Glosari

- Aplicació web: programa que s'executa en un servidor perquè un usuari, mitjançant l'ús de navegadors web, el pugui utilitzar.
- Client: persona que fa ús d'una aplicació web.
- Servidor: Ordinador on s'executen els programes i s'emmagatzemen les dades per a ser utilitzats pels clients.
- Etiqueta: Conjunt de caràcters encapsulats entre els símbols '<' i '>' que equivalen a operacions més complexes en el llenguatge destí.
- Llenguatge de marques: llenguatge de programació específic a l'aplicació format per etiquetes, que té l'objectiu de simplificar el codi.
- XML: és un tipus de document utilitzat per a representar la informació.
- QTI/IMS: És un llenguatge de marques creat com a estàndard per a la creació d'exàmens en els diferents sistemes en l'àmbit educatiu.
- PHP: Acrònim recursiu de 'PHP: Hypertext Preprocessor', és un llenguatge de programació de codi obert, molt utilitzat en aplicacions web.
- ASP.NET: llenguatge de programació desenvolupat per l'empresa Microsoft, de codi tancat, molt utilitzat en aplicacions web.
- IIS: Acrònim d'Internet Information Services', és un servidor web versàtil executat en un sistema Windows.
- Programari lliure: programari amb llibertat d'ús, distribució i modificació. El codi sol ser accessible a qualsevol desenvolupador.
- Programari propietari: programari amb restriccions d'algun tipus: d'ús, distribució o modificació. El codi no sol ser accessible.

ANNEX

OBJECTIUS DE DESENVOLUPAMENT SOSTENIBLE

Grau de relació del treball amb els Objectius de Desenvolupament Sostenible (ODS).

Objectius de Desenvolupament Sostenible	Alt	Mig	Baix	No procedeix
ODS 1. Fi de la pobresa.				X
ODS 2. Fam zero.				X
ODS 3. Salut i benestar.			X	
ODS 4. Educació de qualitat.	X			
ODS 5. Igualtat de gènere.				X
ODS 6. Aigua neta i sanejament.				X
ODS 7. Energia assequible i no contaminant.				X
ODS 8. Treball decent i creixement econòmic.				X
ODS 9. Indústria, innovació i infraestructures.				X
ODS 10. Reducció de les desigualtats.				X
ODS 11. Ciutats i comunitats sostenibles.				X
ODS 12. Producció i consum responsables.				X
ODS 13. Acció pel clima.				X
ODS 14. Vida submarina.				X
ODS 15. Vida d'ecosistemes terrestres.				X
ODS 16. Pau, justícia i institucions sòlides.				X
ODS 17. Aliances per a aconseguir objectius.				X

Com a resultat del projecte, s'ha desenvolupat una eina que simplifica la tasca de crear bateries de preguntes per a la plataforma Poliformat.

El treball realitzat es relaciona directament amb l'objectiu de desenvolupament sostenible 4, Educació de qualitat, concretament al punt 4.3. El docent té disponible aquesta eina per a reduir el temps de donar format al fitxer i dedicar aquest temps a la creació de preguntes de més qualitat que permeten una millor avaluació de l'alumnat.

També es relaciona en menor mesura, amb l'objectiu 3, Salut i benestar. Eines com la desenvolupada en aquest projecte ajuden a l'usuari accelerar el temps dedicat a tasques específiques, concretament, gràcies a la reducció de temps a l'hora de crear bateries de preguntes, el docent redueix el temps d'exposició a les pantalles, finalitza abans la tasca permetent-li descansar i conservar la seua salut i benestar.