

TREBALL DE FI DE CARRERA  
Aplicacions web per a la correcció  
automàtica de proves



# MEMORIA

## Índice de contenido

|  |    |
|--|----|
| 1. Introducció.....                            | 3  |
| 1.1 Context del projecte.....                  | 3  |
| 1.2 Objectius.....                             | 4  |
| 2. Viabilitat del projecte.....                | 5  |
| 3. Metodologia i planificació.....             | 6  |
| 3.1 Desglossament d'activitats.....            | 6  |
| 3.2 Metodologia.....                           | 9  |
| 3.3 Planificació.....                          | 10 |
| 4. Requisits del projecte.....                 | 13 |
| 4.1 Requeriments funcionals.....               | 13 |
| 4.1.1 Autenticació en el sistema.....          | 13 |
| 4.1.2 Front-end.....                           | 13 |
| 4.1.3 Back-end.....                            | 14 |
| 4.2 Requeriments no funcionals.....            | 15 |
| 4.3 Tecnologies a utilitzar.....               | 17 |
| 4.4 Requeriments de maquinari.....             | 19 |
| 4.4.1 Maquinari entorn de desenvolupament..... | 19 |
| 4.4.2 Maquinari entorn d'exploració.....       | 19 |
| 4.5 Requeriments de software.....              | 20 |
| 4.5.1 Software de desenvolupament.....         | 20 |
| 4.5.2 Software entorn d'exploració.....        | 20 |
| 4.6 Pressupost.....                            | 21 |
| 5. Anàlisi.....                                | 22 |
| 5.1 Casos d'ús.....                            | 22 |
| 5.2 Diagrama de classes.....                   | 27 |
| 5.3 Model E/R.....                             | 29 |
| 5.4 Model relacional de la base de dades.....  | 30 |
| 6. Disseny.....                                | 31 |
| 6.1 Base de dades.....                         | 31 |
| 6.2 Diagrama d'activitat.....                  | 32 |
| 6.3 Interfícies.....                           | 33 |
| 6.3.1 Login.....                               | 34 |
| 6.3.2 Front-end.....                           | 34 |
| 6.3.3 Back-end.....                            | 39 |
| 7. Implementació.....                          | 43 |
| 7.1 Capa model.....                            | 44 |
| 7.2 Capa Controlador.....                      | 48 |
| 7.3 Capa vista.....                            | 52 |
| 7.4 Llibreries externes.....                   | 53 |
| 7.5 Estructura d'arxius del projecte.....      | 55 |
| 7.6 Problemes trobats.....                     | 56 |
| 8. Testeig.....                                | 57 |
| 9. Conclusions.....                            | 58 |
| 10. Manual d'ús.....                           | 58 |
| 11. Bibliografia.....                          | 59 |
| 12. Programari utilitzat.....                  | 60 |

## 1. Introducció

Una de les avantatges d'Internet és la d'acostar tot tipus de coneixements als usuaris, permetent aprendre pràcticament sobre qualsevol temàtica. La formació online ha anat creixent en importància en els últims anys a l'eliminar les barreres geogràfiques i de manca de temps. Aquest factor ha fomentat la necessitat d'eines per poder examinar virtualment als usuaris sobre els coneixements adquirits.

Una de les característiques valorades en aquest tipus d'aplicacions de gestió de proves virtuals és l'autocorrecció, que permet als usuaris obtenir immediatament i sense errors la qualificació de la prova que han realitzat, a l'hora que allibera de feina al personal docent.

Aquesta memòria que ens ocupa detalla el procés de planificació, anàlisi, disseny i implementació d'una aplicació per a la correcció automàtica de proves. Tot el procés que documenta ha estat primerament concebut sobre el paper i ha anat sofrint diferents modificacions i adaptacions a mida que ha anat evolucionant el projecte. Sempre que sigui adequat en cada secció es comentarà el concepte tal com es va pensar inicialment i les variacions produïdes a posteriori.

A diferència d'un projecte real, certes parts del cicle de vida del projecte han estat condicionades, tan temporalment com en contingut, per les fites marcades per les diferents PACs que s'han presentat.

### ***1.1 Context del projecte***

L'aplicació està pensada per incorporar-se en un entorn acadèmic existent, ja sigui un entorn virtual per treball online o un entorn presencial amb exàmens per ordinador. En aquest entorn ja existeixen unes assignatures o matèries de les quals són responsables uns professors o docents. El alumnes es matriculen, o són assignats, a les assignatures.

Aquest entorn acadèmic ja disposarà de la seva pròpia base de dades d'alumnes, professores, assignatures, matrícules, expedients, etc, en la que integrarem el nostre mòdul.

En aquest projecte simularem algunes d'aquestes estructures de dades bàsiques per poder dur a terme la programació. Obviarem la introducció i manteniment d'aquestes dades inicials ja que queda fora de la temàtica del projecte i l'entorn en el que ens integrarem ja disposarà d'aquestes eines.

## 1.2 Objectius

L'objectiu d'aquest projecte és dur a terme l'anàlisi, disseny i implementació d'una aplicació web per a la generació i correcció automàtica de proves. Hi ha dos perfils d'usuaris que farien ús d'una aplicació d'aquestes característiques: alumnes i professors. Els alumnes són el grup d'usuaris que s'examinarien i obtindrien la qualificació de forma automàtica en funció de les respostes donades al sistema. Els professors serien els administradors del sistema, editant tots els seus continguts de les proves de les que s'examinarien els alumnes.

Per tant els objectius del projecte es divideixen en dos apartats ben diferenciats tot i que molt relacionats. Dues interfícies, una pels alumnes o Front-end i una altra pels professors o Back-end.

Aquestes són les característiques que haurà de tenir cada interfície:

- **Back-end [Perfil d'usuari: Professors]**
  - Gestió de les proves acadèmiques vinculades a les assignatures de les que el professor és el responsable.
  - Disseny i manteniment de proves: El professor generarà cadascuna de les proves amb totes les preguntes, respostes i característiques especials de la prova.
  - Llistats de resultats: Visualització dels resultats de les diferents proves dutes a terme pels alumnes

- **Front-end [Perfil d'usuari: Alumnes]**

- Visualització de proves: Llistat de totes les proves assignades a l'alumne, amb els resultats obtinguts en les ja realitzades i la possibilitat d'examinar-se de les proves pendents.
- Realització de proves pendents: L'alumne podrà marcar les respostes de les diferents preguntes i finalitzar la prova per a que el sistema l'evalui i li mostri la puntuació obtinguda de forma automàtica.

El tipus de prova virtual més comú és la de tipus test, on per a cada pregunta es donen diferents possibles respostes entre les quals només una (o sovint més d'una) és correcta. És aquest tipus de prova la que s'implementarà en aquest projecte.

Tal com ja em comentat en l'apartat de Context del Projecte, queda fora de l'abast d'aquesta aplicació la gestió de les dades principals del sistema, com són per exemple les dades dels alumnes, professors, assignatures i matrícules.

## 2. Viabilitat del projecte

Abans de desenvolupar un projecte s'ha d'estudiar primer si serà viable la seva integració, tant a nivell tècnic, econòmic i operatiu.

- **Viabilitat tècnica**: La idea de l'aplicació que ens ocupa és integrar-se en un entorn docent ja funcional, pel que és probable que tècnicament ja es cobreixin els requisits bàsics. Dependrà llavors de la compatibilitat entre el software ja instal·lat i el que es pretén integrar. És aconsellable l'ús de programari multi-plataforma per garantir aquesta compatibilitat.
- **Viabilitat econòmica**: En què millora la implantació del projecte la productivitat del sistema? S'ha d'estudiar si facilitar al personal docent la gestió de les proves acadèmiques i

eliminar la correcció manual d'aquestes últimes comporta un estalvi que compensi la inversió a realitzar.

La utilització de programari gratuït i el baix consum de recursos d'una aplicació d'aquestes característiques, per tant el no necessitar un maquinari específic, redueix el cost de la implantació fent que el projecte sigui més fàcilment viable a nivell econòmic.

- **Viabilitat operativa**: Una vegada implantada, l'aplicació no requereix d'usuaris especialitzats per al seu manteniment, doncs es el propi personal docent el que administra les proves i el sistema el que les puntua automàticament. S'ha de suposar una actitud positiva per part dels usuaris d'aquest perfil, doncs l'eina té com a objectiu el facilitar la seva feina.

### 3. Metodologia i planificació

Una vegada clars els objectius del projecte, s'han d'identificar les diferents tasques a dur a terme de cara a escollir el mètode a seguir i poder planificar més acuradament cadascun dels trams.

#### 3.1 Desglossament d'activitats

En la relació d'activitats i fites a complir afegim la realització i entrega de les diferents PACs de les que consta el projecte, doncs tot i no formar part intrínseca de l'aplicació que volem desenvolupar, condicionen tota la planificació. Aquest és el desglossament d'activitats a dur a terme:

- Creació del pla de treball i planificació temporal del projecte:

Desglossament del projecte en diferents fites i activitats per a poder fer una planificació temporal del projecte més acuradament. Ens serviran després per escollir la metodologia de treball.

Incorporació de diagrames de temps.

- Realització de la documentació de la PAC1:

Redacció de la documentació relacionada para el pla de treball i planificació temporal

- Recull de requeriments funcionals de l'aplicació:

S'han d'analitzar els requeriments de l'aplicació, tan a nivell funcional com no funcional. S'han d'especificar totes les funcionalitats que tindrà l'aplicació web, tan en la interfície Front-End com la Back-End, per tal de fixar tots el objectius a assolir.

Tan importants son els requisits no funcionals com els funcionals. Mentre que la manca d'un d'aquests últims fa que l'aplicació continuï funcionant encara que minvada, el no compliment d'un requeriment no funcional podria arribar a suposar que l'aplicació no funcioni en absolut.

Per tant és tan important fixar les funcionalitats a cobrir per l'aplicació com determinar el requeriments no funcionals d'aquesta, per tal de no trobar-nos amb una sorpresa desagradable.

- Anàlisi dels requeriments tècnics i de programari:

A partir dels requisits funcionals de l'aplicació es detallaran uns requisits de maquinari i programari i es marcarà el pressupost del projecte. Aquests requisits marcaran la viabilitat del projecte, tan a nivell tècnic com econòmic.

- Esquema de les classes principals que formaran l'entramat del sistema:

Especificació dels diferents casos d'ús de l'aplicació per definir els processos que haurà de dur a terme l'aplicació. A partir dels casos d'ús podrem definir les classes que formaran la base del sistema, els seus atributs i les interaccions entre elles.

- Disseny de la base de dades:

Disseny d'un model relacional per a la base de dades per adaptar-se a les classes que hem especificat.

- Realització de la documentació de la PAC2:

Redacció de la documentació dels anàlisis de requeriments, diagrames de casos d'ús, esquemes de classes i el model relacional de la base de dades.

- Definir la lògica de treball de la interfície Back-end:

Definir mitjançant diagrama d'activitats el flux de treball de la interfície Back-end de l'aplicació.

Aquesta interfície permetrà administrar les proves acadèmiques, creant, modificant o eliminant-les.

- Lògica de treball de la interfície Front-end:

Definir mitjançant diagrama d'activitats el flux de treball de la interfície Front-end de l'aplicació.

Aquest interfície és la que permetrà examinar i assignar qualificació de forma automàtica als alumnes.

- Planificació de la usabilitat i seguretat:

Definir el sistema d'assegurar l'autenticació dels usuaris al sistema i evitar un accés no permès.

Dissenyar les interfícies per facilitar l'ús de l'aplicació i fer-la el més pràctica possible.

- Realització de la documentació de la PAC3:

Redacció de la documentació sobre les lògiques de funcionament de l'aplicació, així com el disseny de les interfícies i la seguretat.

- Codificació de la interfície Back-end:

Programació de la interfície que farà possible la gestió de les proves acadèmiques per part del docents.

- Codificació de la interfície Front-end:

Programació de l'interfície farà possible que els alumnes puguin examinar-se i rebre automàticament les qualificacions de les proves assignades.

- Jocs de proves conjuntes i solució d'incidències:

Testeig de les dues interfícies i de tota la casuística possible de cara a trobar possibles errors en la programació. Reparació de qualsevol error detectat.



- Realització de la memòria i presentació virtual del projecte:

Redacció de la memòria final del projecte, creació de la presentació virtual i edició del vídeo de presentació del treball.

### **3.2 Metodologia**

De forma natural al realitzar el desglossament d'activitats sorgeix el mètode de *cascada*, on cadascuna de les fases comença quan acaba l'anterior. En part la realització de la documentació de les PACs serveix com a separador de les diferents fases, pel que aquest mètode encaixa perfectament amb aquestes entregues parcials.

Per aplicar aquest mètode d'entrada hem de tenir clares totes les especificacions, cosa que en projectes encarregats per un client no sempre tindrem ben definits. En el cas de la nostra aplicació tenim clars tots els requisits funcionals pel que podem seguir cadascuna de les fases seqüencialment.

Una de les desavantatges del sistema es que molts dels errors d'anàlisi no es detecten fins a la fase de codificació, moment en el que és més complicat tornar enrere.

Precisament degut a la desavantatge comentada, durant la fase de codificació s'han detectat errors d'anàlisi que han obligat a realitzar modificacions sobre diagrames i esquemes dissenyats anteriorment. Això ens ha portat a una metodologia més pròpia d'un desenvolupament àgil, en el que es redefeixen especificacions, s'implementen, es testegen, es defineixen nous canvis, es codifiquen i testegen, i així cíclicament.

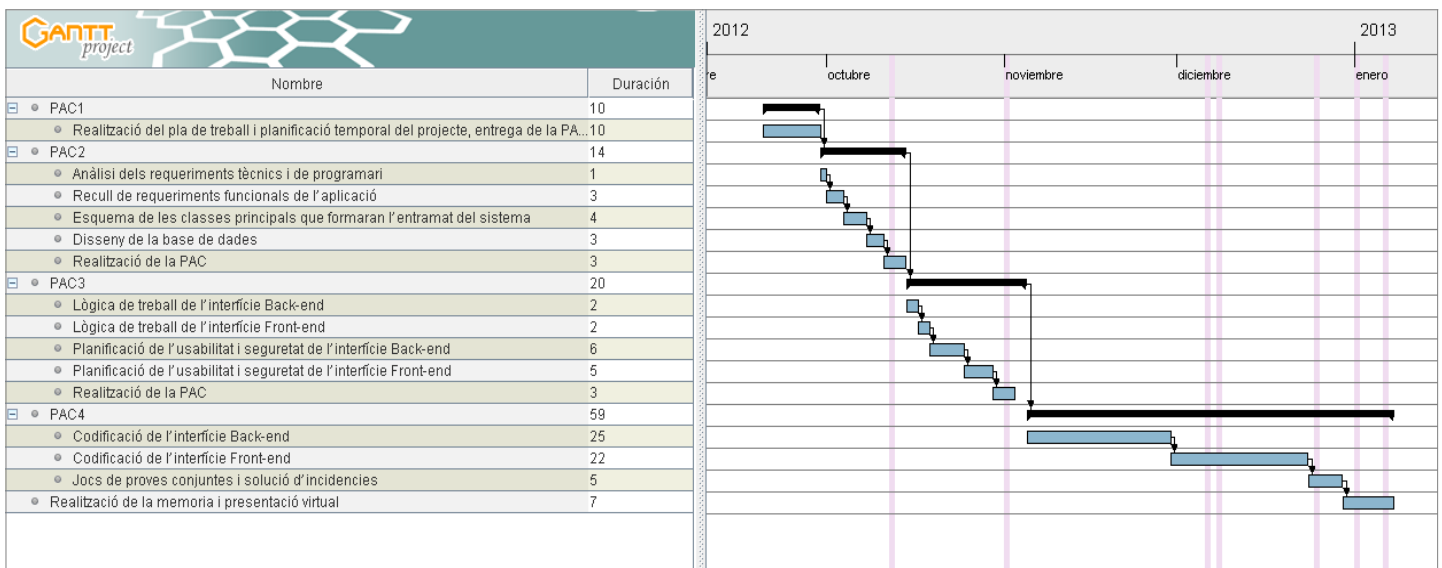
### 3.3 Planificació

Tot i realitzar una previsió temporal de cada una de les fites a assolir, les entregues de les PACs marquen uns temps límit que s’han d’intentar mantenir amb marge suficient per superar qualsevol eventualitat.

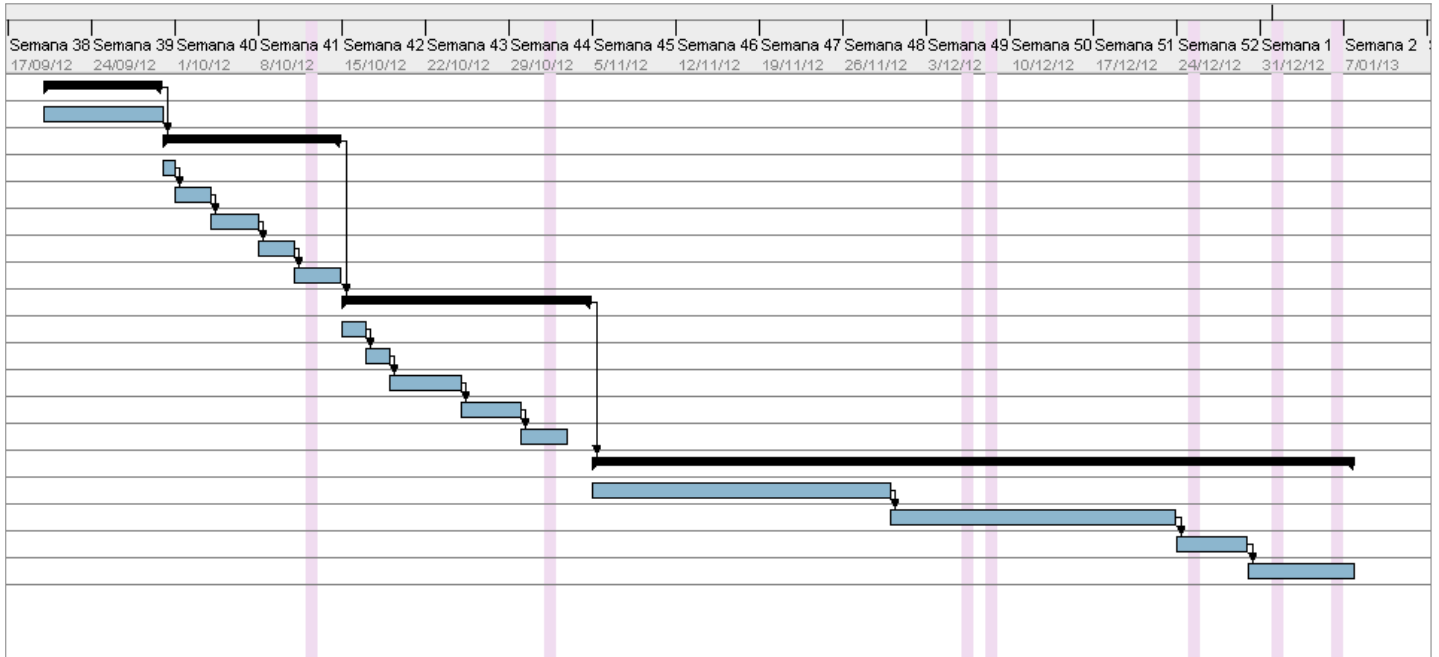
Per a poder fer una planificació realista s’ha de preveure la disponibilitat temporal del recurs humà, en aquest cas un servidor, de cara a calcular el temps diari que es podrà dedicar al projecte. No té sentit inflar la planificació temporal assignant dotzenes d’hores si està clar d’entrada que no serà possible complir-les ni seran necessàries per a dur a terme el projecte.

En el cas que ens ocupa, estimarem una disponibilitat d’una hora diària de dedicació al projecte, per cadascun dels 7 dies de la setmana. Per temes laborals molts dies entre setmana no es podrà complir la hora planificada, que es podrà recuperar sobradament els dies de cap de setmana.

Es preveu que en algunes de les fases incloses dins de la PAC2 i PAC3 s’inverteixi menys temps del planificat. Si es dona el cas aquest temps guanyat s’invertirà avançant feina de les fases de codificació de la PAC4.

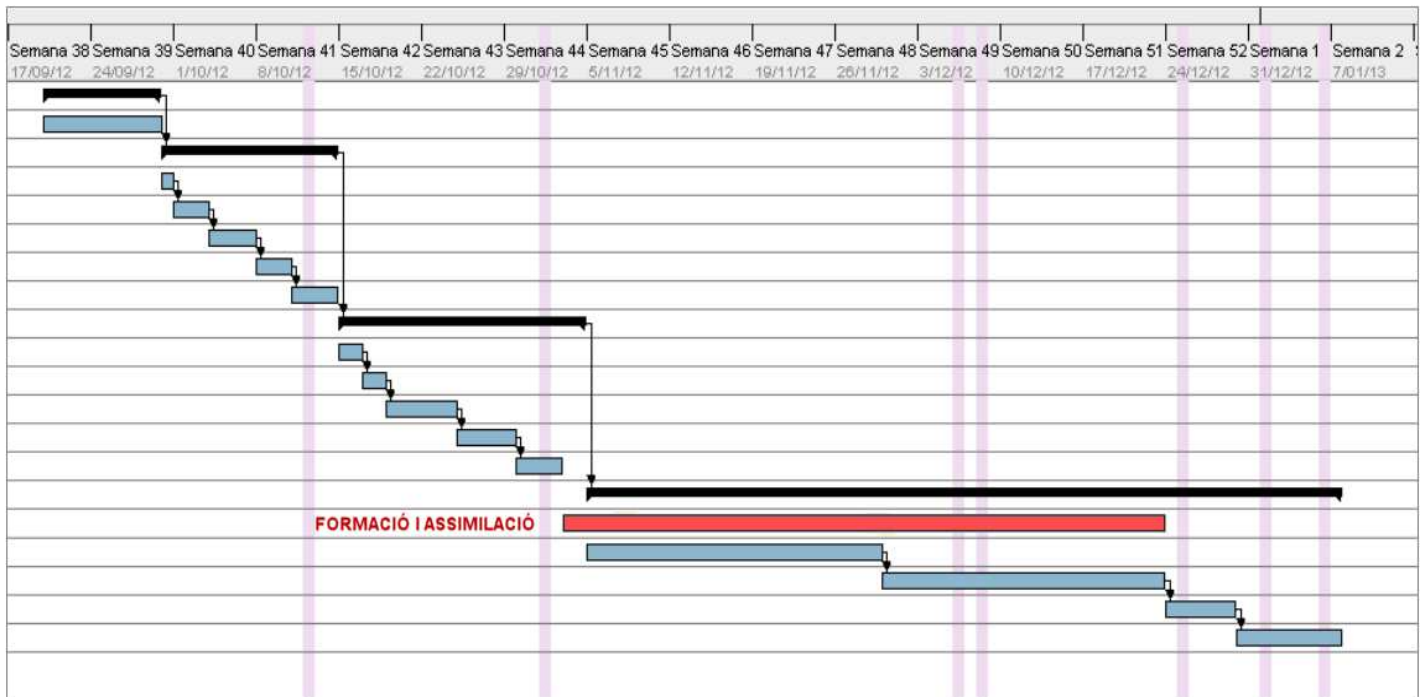


Vista de les activitats a realitzar amb diagrama de temps



*Vista del diagrama de temps ampliat*

La planificació té un error important ja que durant la fase de codificació no es van tenir en compte les hores de formació i d'assimilació de les tecnologies a utilitzar (com PHP, Ajax, JQuery, etc.). Aquest aprenentatge, que en un projecte hauria de ser mínim, ha representat una quantitat d'hores extra important, de manera que la planificació de la fase de codificació ha resultat ser del tot insuficient. S'han hagut d'utilitzar el doble d'hores de les calculades, compartides entre la nova fase de formació i la de codificació, quedant finalment el diagrama de temps d'aquesta manera:



Finalment a la fase de documentació i creació de la memòria del projecte, tot i encaixar bé dins del rang temporal previst, se l'hi ha hagut de destinar el doble d'hores de les inicialment descrites.

## 4. Requisits del projecte

### 4.1 *Requeriments funcionals*

Com ja hem comentat anteriorment, l'aplicació tindrà dos entorns ben diferenciats, cadascun amb unes funcionalitats específiques. A aquests entorns hi accediran els actors corresponents després d'autenticar-se en el sistema.

Per tant definim tres àrees principals de requeriments funcionals: l'autenticació en el sistema, la interfície Front-end i la interfície Back-end. A continuació procedim a detallar els requeriments funcionals de cadascuna d'elles.

#### 4.1.1 Autenticació en el sistema

Es demanarà un usuari i contrasenya que es validaran contra la llista d'usuaris de l'entorn acadèmic. Els usuaris de l'aplicació seran alumnes ja matriculats i professors que hi treballin, per tant no s'implementarà un sistema de registre d'usuaris, queda fora de la funcionalitat de l'aplicació.

Una vegada comprovats els credentials, segons el perfil d'usuari es redirigirà cap a un entorn front-end o cap a un entorn back-end.

#### 4.1.2 Front-end

Aquesta és la interfície a la que accediran els alumnes:

- Es llistaran totes les proves per assignatura a la que l'alumne està matriculat, tant les proves ja finalitzades com les pendents de realitzar.
- Les proves ja finalitzades podran ser consultades i impreses de cara a facilitar l'estudi als alumnes. Si la configuració de la prova ho permet, es veuran les respostes incorrectes senyalades amb la resposta correcta.
- L'alumne podrà realitzar les proves pendents. Una vegada començada una prova és possible que segons la seva configuració hi hagi un temps límit per dur-la a terme, temps que s'anirà esgotant encara que l'alumne surti de l'aplicació.

- Una vegada l'alumne marca com a finalitzada una prova, el sistema la corregirà automàticament i mostrarà la qualificació, que quedarà guardada al sistema.

### 4.1.3 Back-end

Aquesta és la interfície a la que accediran els professors:

- Es llistaran totes les proves per assignatura de la que el professor és el responsable.

Desviació: Inicialment no es va tenir en compte la distinció de les proves per semestres/cursos, de manera que les proves antigues no sortissin llistades d'entrada. Això evitarà que amb el temps la llista sigui massa extensa al mostrar proves no vigents.

- Es podrà crear una prova per a una assignatura. Apart de la configuració general de la prova després es definirà cadascuna de les preguntes de les que constarà la prova i les seves respectives opcions de resposta. La prova portarà una data inici i una data fi d'activació, que serà quan estarà disponible per a que els alumnes la facin.
- Una prova podrà modificar-se sempre i quan la data actual no coincideixi amb el rang de dates d'activació de la prova. Podrà modificar-se les preguntes, afegir-ne de noves o eliminar-ne les existents.

La funcionalitat inicial no permet modificar una prova en curs o passada per exemple per allargar el plaç de vigència. Es permetrà una modificació limitada de les proves vigents o ja passades.

- S'han de poder eliminar totes les proves que no estiguin actives o que només estiguin planificades.
- Per cada prova es podran llistar les qualificacions obtingudes pels alumnes, per tal de veure'n els resultats.

S'ha afegit l'opció de poder veure des del llistat qualsevol prova realitzada per un alumne.

- Es podrà consultar les qualificacions de cadascun dels alumnes, de totes les proves de les assignatures de les que és responsable el professor.

S'ha afegit l'opció de poder veure des del llistat qualsevol prova realitzada per un alumne.

- Nova funcionalitat que permetre crear una prova a partir d'una altra prova existent, duplicant la configuració, les preguntes i respostes.

## 4.2 *Requeriments no funcionals*

- Rendiment:

Es tracta d'una aplicació que no genera una càrrega important de sol·licituds ni al servidor ni a la base de dades, ja que tot el tràfic surt de la interacció de l'usuari amb la interfície. Només en certes moments puntuals pot haver una càrrega d'usuaris connectats, però al no generar un tràfic de peticions molt dens poden ser gestionats per un sistema sense un requeriments de hardware especials.

- Seguretat:

El principal esforç de cara a la seguretat de l'aplicació radica en garantir la identitat de l'usuari i que accedeixi únicament a la interfície adequada al seu perfil.

Els usuaris s'han d'autenticar al sistema mitjançant un usuari i una contrasenya, que està emmagatzemada en la base de dades de forma encriptada. Una vegada comprovat en el servidor que els credencials són correctes s'inicialitzen unes variables de sessió que emmagatzemen l'usuari i el perfil de cara a permetre que accedeixi a una interfície o a un altre.

A cada petició al servidor es comprova que les variables de sessió són correctes per donar la resposta demanada o per anul·lar un accés no permès.

- Mantenibilitat:

L'aplicació s'ha de poder modificar fàcilment per adaptar-se a un canvi o per corregir un comportament no adequat.

S'ha optat per un sistema Model Vista Controlador (MVC) per codificar l'aplicació. Aquest sistema, del que es parlarà més detalladament en l'apartat 7. *Implementació*, permet mantenir separades la interfície de l'usuari, la lògica de control i la gestió de dades. D'aquesta manera es poden mantenir i escalar separatament els cadascun dels diferents elements sense afectar als altres.

És senzill tenir una còpia de l'aplicació corrent en un servidor web virtual i una còpia de la base de dades com a entorn de programació. Una vegada modificada l'aplicació en aquest entorn de prova i una vegada testejat, es pot actualitzar a l'entorn real ràpidament sense interrompre el funcionament del sistema.

- Usabilitat:

La principal funció de l'aplicació és la realització de proves, pel que ha de ser àgil d'utilitzar, sense complicacions que no aportin res realment útil. En tots els llistats cada registre ha de disposar d'icones per cadascuna de les operacions disponibles de manera que sigui ràpid i senzill accedir a la funcionalitat desitjada.

- Accessibilitat:

Hi ha una sèrie de mesures estàndards per facilitar l'accessibilitat d'una pàgina web, com etiquetar totes les imatges i contingut multimèdia que pugui no ser visualitzat correctament per l'usuari, utilitzar tamany més grans de lletres i més contrast de colors de fons, facilitar la navegació per a persones amb mobilitat reduïda, etc.

En el cas de l'aplicació web es prendran les següents mesures:

- Totes les imatges e icones estaran etiquetades



- Es disposara d'un enllaç a una versió d'alt contrast de la pàgina, que ocuparà a demés tot l'espai disponible en pantalla.
- Tota la navegació i operacions funcionen a través del ratolí. Tot i que en teoria s'hauria de facilitar la navegació per teclat per a persones amb mobilitat reduïda, l'aparició de diferents softwares de mouse virtuals molt accessibles com “Headmouse” fan que sigui més còmoda una navegació d'aquest tipus,

### **4.3 Tecnologies a utilitzar**

El llenguatge base escollit per a implementar el projecte és PHP. És un llenguatge de la part del servidor però que es pot integrar dins de les pàgines HTML per a que siguin dinàmiques. El codi PHP és interpretat pel servidor que retorna la pàgina web resultant.

Algunes de les característiques del llenguatge PHP per les quals ha estat escollit pel projecte son:

- És lliure i gratuït, per tant es pot utilitzar i distribuir lliurement i sense cost algun.
- Al ser un llenguatge interpretat pel servidor, no ha de ser compilat, pel que no requereix cap software especial per programar. Qualsevol editor de text serveix per programar amb PHP.
- Es pot realitzar una programació orientada a objectes, de manera que permet escriure codi estructurat que faciliti la creació de sistemes basat per exemple en el MVC.
- Pot connectar-se amb molts motors diferents de bases de dades, sobretot amb sistemes lliures com MySQL i PostgreSQL.

L'estructura de les pàgines web estarà construïda en HTML amb un ús bastant extens de CSS, de manera que permeti modificacions posteriors de la Vista amb el mínim esforç.

Tenint en compte els requisits funcionals es fa patent que serà necessari un ús molt interactiu de l'aplicació, interacció que amb un llenguatge de servidor es fa molt farragosa. La solució per millorar la usabilitat es la tecnologia Ajax. Ajax no és un llenguatge en si, sinó l'ús conjunt d'una serie de llenguatges, com son HTML, Javascript, PHP, etc.

La incorporació d'Ajax permet un ús més àgil de les aplicacions web, amb avantatges com la de poder consultar i actualitzar informació que es troba en el servidor sense haver de refrescar completament la pàgina. També ens permet actuar en temps real sobre elements de la pàgina web, canviant les seves propietats al vol.

Una vegada començada la programació s'ha incorporat l'ús de la llibreria jQuery de Javascript. Aquesta llibreria permet l'accés i modificació d'elements de la pàgina web de forma simplificada, estalviant molta codificació Javascript. També evita haver d'invertir temps en atendre a la compatibilitat entre navegadors, doncs és la pròpia llibreria la que ho gestiona de forma transparent al programador.

Com a motor de bases de dades l'elecció inicial ha estat MySQL. El motiu principal és que en entorns de programació lliure es pot distribuir el servidor de MySQL sense cost. A més a més una de les seves característiques és ser multi-plataforma, pel que es pot integrar en pràcticament qualsevol entorn sense problemes.

Un altre motiu per utilitzar MySQL és que és el gestor de bases de dades integrat en l'eina XAMPP. Es tracta d'una aplicació que incorpora un servidor web Apache, el gestor de bases de dades MySQL i l'interpret de codi de PHP. XAMPP és una eina de programació que permet testear aplicacions web localment sense necessitat d'accedir a Internet, i s'utilitzarà durant la codificació del projecte.

El projecte es va plantejar com un repte acadèmic, pel que la llicència GPL de MySQL permetia usar-lo sense cap cost. Però si es planteja aquest projecte com una eina comercial llavors el gestor de bases de dades no és gratuït i s'han d'adquirir llicències. Això faria augmentar el cost del projecte considerablement, tenint en compte que la majoria de programari utilitzat és gratuït.

Una alternativa podria ser el gestor de bases de dades PostgreSQL, que es publica sota llicència BSD, més permissiva que la GPL en quan a projectes comercials. Disposa de pràcticament les mateixes característiques que MySQL i el llenguatge PHP hi pot treballar de la mateixa manera.

## **4.4 Requeriments de maquinari**

### **4.4.1 Maquinari entorn de desenvolupament**

L'eina XAMPP ens permet executar un servidor web i de bases de dades localment, pel que només és necessari un equip de programació. La programació és a nivell de text i es testeja en navegadors web, pel que l'equip de desenvolupament pot tenir un perfil d'usuari bàsic, no necessita de característiques especials.

### **4.4.2 Maquinari entorn d'exploració**

L'aplicació requereix d'un servidor web i un gestor de bases de dades que poden estar en un mateix equip físic. Aquest equip servidor ha d'estar connectat a la xarxa de l'entorn on s'integra, i tenir sortida a Internet només si l'aplicació ha de ser accessible des de l'exterior. Es pot instal·lar fàcilment en un servidor ja existent.

Les característiques de l'equip servidor dependran del tamany de l'entorn acadèmic on s'instal·li. Gran part de la interacció dels usuaris amb l'aplicació es gestiona en l'equip client, pel que la càrrega al servidor és baixa, limitant-se pràcticament als accessos a la base de dades i a la interpretació de les pàgines PHP. Només per entorns amb una gran concurrència d'usuaris es podria necessitar d'un nou servidor.

## 4.5 Requeriments de software

De cara a reduir el màxim possible el cost del projecte es tractarà d'utilitzar sempre que sigui possible software gratuït. En tots els casos les característiques d'aquest tipus de programari cobriran sobradament les necessitats del projecte.

### 4.5.1 Software de desenvolupament

Per desenvolupar aquest projecte es necessitaran les següents eines:

- Editor de text per a la programació amb HTML, Javascript i PHP.
- Servidor web
- Gestor de bases de dades, junt amb una interfície gràfica per a facilitar l'administració
- Editor d'imatges per a les icones i logotips que necessiti l'aplicació
- Navegadors web en els que testejar l'aplicació. A ser possible s'haurien d'utilitzar els tres més estesos.
- Processador de text per a generar la documentació del projecte
- Diversos generador de diagrames i esquemes per la planificació i anàlisi del projecte.

A mida que el projecte ha anat avançant s'han escollit uns o altres programes per cobrir les necessitats. En la secció 10. *Programari utilitzat* es pot trobar una relació de tot el software utilitzat finalment durant tot el cicle de vida del projecte.

### 4.5.2 Software entorn d'execució

A l'hora d'integrar l'aplicació en l'entorn final, es requereix tenir instal·lat el següent programari:

- Equip Servidor:
  - Servidor web que rebí les peticions i serveixi les pàgines web
  - Gestor de bases de dades que emmagatzemi les dades i atengui les peticions del servidor web

- Equips clients:
  - Navegador web actualitzat

## **4.6 Pressupost**

L'aplicació té com a objectiu integrar-se en un entorn acadèmic ja funcional, pel que la despesa de maquinari pot ser descartada a menys que es tracti d'un sistema ja obsolet que no pugui fer front a cap aplicació extra.

Tant en el desenvolupament com en la integració final del projecte es tractarà d'utilitzar sempre que sigui possible software gratuït pel que podem arribar a aconseguir reduir a zero la despesa de programari.

Per tant ens trobem que el cost del projecte serà pràcticament només el cost de la programació. Segons la planificació temporal es requereixen unes 90 hores per dur a terme les funcionalitats previstes. El preu hora de programació és molt relatiu i podria variar moltíssim depenent de molts factors, però establint uns 40 euros ens donaria un cost de programació de 3.600 euros.

És possible que s'haguessin d'afegir algunes hores més en cas d'haver d'instal·lar, configurar i posar en marxa l'aplicació en l'entorn final, depenent de la capacitat del client per fer-ho pel seu compte.

Tal com s'ha comentat en l'apartat 4.3, en el cas del gestor de bases de dades MySQL la llicència GPL limita el tipus d'ús que es pot fer de l'aplicació, pel que és possible que al cost del projecte s'hagués d'afegir les llicències de MySQL.

En l'error de planificació comentat en l'apartat 3.3, les hores destinades finalment a la codificació de l'aplicació han estat el doble de les previstes. Aquestes hores extra (unes 60) han estat de formació i assimilació de les tecnologies a utilitzar, per tant és un cost que s'ha de comptabilitzar si es tracta d'una eina d'aplicació interna, però no és un cost que es pugui imputar a un client.

## 5. Anàlisi

Una vegada definits els requisits funcionals de l'aplicació podem proposar uns esquemes que serveixin de base per al posterior disseny.

### 5.1 Casos d'ús

Mitjançant un diagrama de casos d'ús podem definir el model de comportament que ha de tenir l'aplicació per cobrir els requisits descrits. En aquest diagrama es defineixen els actors (marcats amb un ninot) i els casos d'ús (indicats amb una el·lipse).

Els actor són els rols dels usuaris que intervenen en l'aplicació, mentre que els casos d'ús són les diferents funcionalitats i processos del sistema. En el nostre projecte identifiquem dos actors, que són per un cantó l'alumne i per altre el professor. Ambdós parteixen d'una base, que és la de ser usuaris del sistema, pel que hereten d'un actor comú.



|                 |   |
|-----------------|---|
| <b>Procés</b>   | Es poden veure la llista de proves de les assignatures a les que està matriculat, tant les ja finalitzades com les pendents de realitzar. L'alumne disposa d'unes icones per poder consultar una prova ja finalitzada o entrar a fer una prova pendent. |
| <b>Resultat</b> | Llista de proves a les que té accés l'alumne  |

| <b>[CU03] Realitzar prova pendent</b> |   |
|---------------------------------------|---|
| <b>Descripció</b>                     | L'alumne pot fer una prova que tingui pendent   |
| <b>Actors</b>                         | Alumne  |
| <b>Precondicions</b>                  | L'alumne ha d'estar autènticat en el sistema i la prova no ha exhaurit el temps de realització permès   |
| <b>Procés</b>                         | Es mostren les preguntes de la prova amb les seves possibles respostes. L'alumne va marcant les respostes que considera correctes i finalitza la prova. |
| <b>Resultat</b>                       | Es realitza la prova, si es finalitza es dirigirà a ser corregida   |

| <b>[CU04] Obtindre qualificació</b> |  |
|-------------------------------------|--|
| <b>Descripció</b>                   | La prova és corregida i puntuada   |
| <b>Actors</b>                       | Alumne   |
| <b>Precondicions</b>                | L'alumne ha d'estar autènticat en el sistema i ha de finalitzar la prova   |
| <b>Procés</b>                       | Una vegada finalitzada la prova el sistema comprovarà les respostes correctes i en funció del barem de puntuació mostrarà i guardarà la qualificació de l'alumne |
| <b>Resultat</b>                     | La prova es marca com a realitzada i l'alumne obté la seva puntuació   |

| <b>[CU05] Consultar prova ja finalitzada</b> |   |
|--|---|
| <b>Descripció</b>                            | Es pot consultar una prova ja realitzada anteriorment de cara a facilitar l'estudi  |
| <b>Actors</b>                                | Alumne  |
| <b>Precondicions</b>                         | L'alumne ha d'estar autènticat en el sistema i la prova ha d'estar ja finalitzada i puntuada.   |
| <b>Procés</b>                                | Es mostren les preguntes de la prova amb les respostes que va marcar l'alumne. Si la configuració de la prova ho permet es mostrarà la resposta correcta de cadascuna de les preguntes. |
| <b>Resultat</b>                              | Es mostra el contingut d'una prova realitzada i, si el sistema ho permet, les solucions   |

| <b>[CU06] Llistar proves per assignatura</b> |  |
|--|--|
| <b>Descripció</b>                            | Es llisten les proves disponibles per cadascuna de les assignatures  |
| <b>Actors</b>                                | Professor  |
| <b>Precondicions</b>                         | El professor ha d'estar autènticat en el sistema   |
| <b>Procés</b>                                | Es mostren les proves vinculades a les assignatures de les quals es responsable el professor. Des d'aquest llistat es porta la gestió de proves, doncs el professor podrà crear-ne de noves, |



|                 |  |
|-----------------|--|
|                 | modificar les existents o eliminar-les.                                |
| <b>Resultat</b> | Es mostren les proves vinculades al professor per tal de gestionar-les |

| <b>[CU07] Llistar qualificacions per prova</b> |   |
|--|---|
| <b>Descripció</b>                              | Es llisten les qualificacions obtingudes pels alumnes en cadascuna de les proves  |
| <b>Actors</b>                                  | Professor   |
| <b>Precondicions</b>                           | El professor ha d'estar autenticat en el sistema                                  |
| <b>Procés</b>                                  | Es mostren les qualificacions obtingudes pels alumnes en cadascuna de les proves. |
| <b>Resultat</b>                                | Consulta de les qualificacions per prova  |

| <b>[CU08] Llistar qualificacions per alumnes</b> |   |
|--|---|
| <b>Descripció</b>                                | Es llisten les qualificacions de l'alumne   |
| <b>Actors</b>                                    | Professor   |
| <b>Precondicions</b>                             | El professor ha d'estar autenticat en el sistema  |
| <b>Procés</b>                                    | Es mostren les qualificacions dels alumnes matriculats a assignatures de les quals es responsable el professor. |
| <b>Resultat</b>                                  | Es mostren l'expedient dels alumnes en les assignatures depenents del professor                                 |

| [CU09] Crear proves |   |
|---------------------|---|
| Descripció          | Creació d'una prova nova  |
| Actors              | Professor   |
| Precondicions       | El professor ha d'estar autenticat en el sistema  |
| Procès              | S'haurà d'escollir l'assignatura (d'entre les que es responsable el professor) sobre la que es farà la prova. Es configuren els diferents paràmetres de la prova, entre ells l'interval de dates en les que la prova estarà activa.<br>A continuació s'introdueix cadascuna de les preguntes amb les seves corresponents opcions de resposta, indicant quina o quines són les correctes.<br>Es guarda la prova, que estarà disponible per fer-se dins l'interval de dates configurat. |
| Resultat            | S'ha creat una nova prova per a una assignatura   |

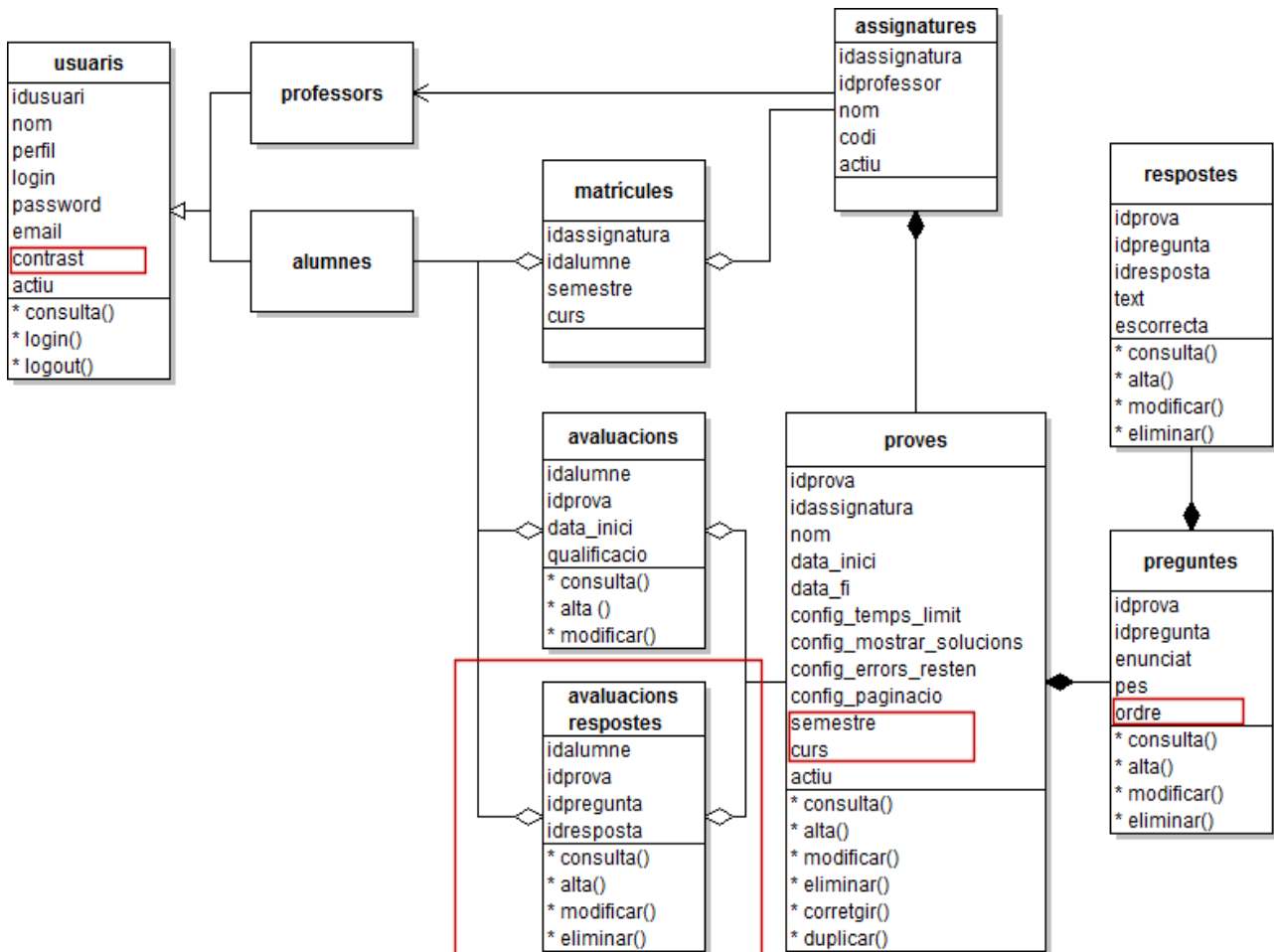
| [CU10] Modificar proves |  |
|-------------------------|--|
| Descripció              | Modificació d'una prova existent   |
| Actors                  | Professor  |
| Precondicions           | El professor ha d'estar autenticat en el sistema i la prova no ha d'estar en curs  |
| Procès                  | Es pot modificar la configuració general de la prova. També es pot accedir a les preguntes i respostes per tal d'afegir-ne noves, modificar-les o eliminar-ne algunes. |
| Resultat                | S'ha modificat una prova existent.   |

| [CU11] Eliminar proves |  |
|------------------------|--|
| Descripció             | Eliminació d'una prova existent  |
| Actors                 | Professor  |
| Precondicions          | El professor ha d'estar autenticat en el sistema i la prova no ha d'haver-se activat encara.                               |
| Procès                 | Després d'un missatge de confirmació, tant la prova com les preguntes i respostes associades seran esborrades del sistema. |
| Resultat               | S'ha esborrat una prova existent.  |

| [CU12] Imprimir |   |
|-----------------|---|
| Descripció      | Impressió d'informació mostrada per pantalla  |
| Actors          | Alumne i Professor  |
| Precondicions   | L'alumne o el professor han d'estar autenticats en el sistema   |
| Procès          | Els llistats i les proves mostrades en pantalla es podran marcar per imprimir o per guardar en PDF. S'obrirà una pantalla nova amb la informació però sense tot l'entorn web que l'envoltava generada en un arxiu PDF que es podrà guardar en l'equip local de l'usuari o imprimir. |
| Resultat        | Impressió de la informació en un document PDF   |

## 5.2 Diagrama de classes

Veiem el diagrama de classes proposat:



Hi ha una serie de classes (usuari, matrícules i assignatures) que no disposen de mètodes per la modificació de les dades, doncs la seva gestió cau fora de l'abast de l'aplicació tal com s'ha parlat a l'apartat de suposicions inicials.

Les classes *alumnes* i *professors* hereten de la classe *usuari*, doncs comparteixen els mateixos atributs i mètodes. Estan definides per uns atributs de registre i per un mètode principal, que és el que comprova els credencials de l'usuari a la base de dades quan entra al sistema.

La classe *assignatures* està relacionada amb la de *professors* de manera que un professor és l'encarregat d'una o més assignatures.

La classe *matrícules* relaciona la classe dels *alumnes* i la de *assignatures*, indicant els alumnes que estan matriculats a una assignatura concreta durant un semestre d'un curs.

Una de les principals classes és la de *proves*, que es la que defineix els atributs i mètodes de les proves que dissenyaran els professors pels alumnes. Les classes *preguntes* i *respostes* estan relacionades amb la de *proves* de forma jeràrquica, de manera que l'existència de les *respostes* depèn de l'existència de les *preguntes* que a la vegada existeixen en una *prova* concreta.

Les proves tenen un interval de dates que defineixen la seva activitat, a més d'una serie d'atributs de configuració que defineixen el comportament de les proves tant en la seva execució com en la correcció. A part dels mètodes per a la seva gestió la classe de *proves* disposa d'un mètode per a la seva correcció, en la que intervendran les classes relacionades.

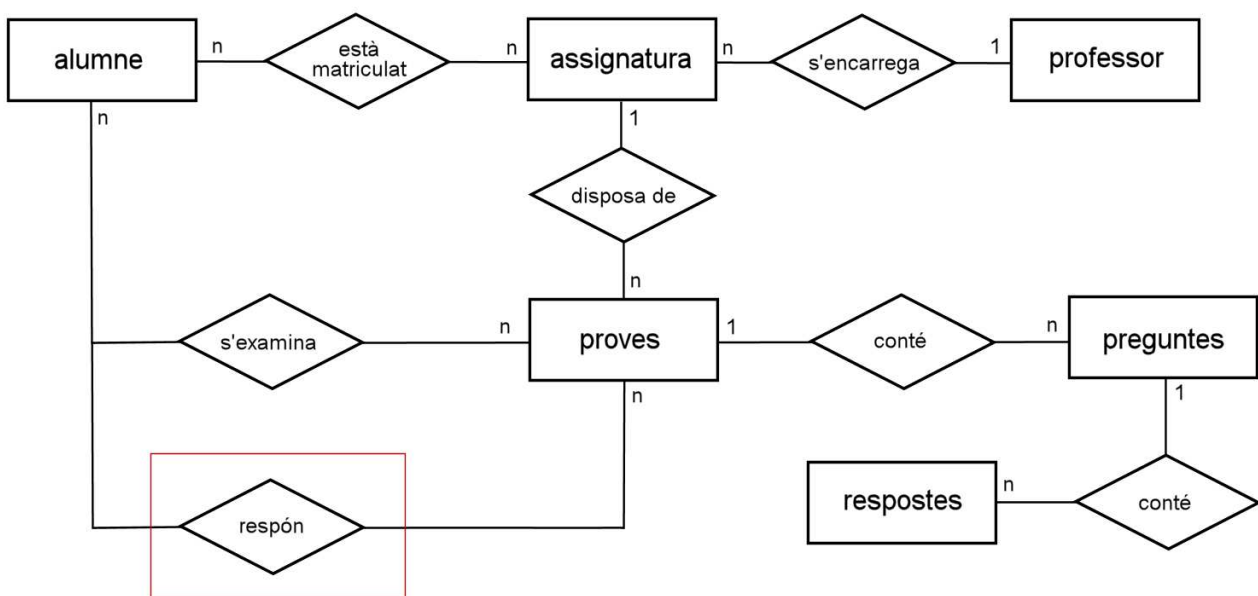
La classe de *avaluacions* relaciona la classe d'*alumnes* amb la de *proves* definint quan un alumne ha començat a realitzar una prova. Quan finalitza aquesta, un mètode assigna qualificació a l'alumne deixant com a finalitzada aquest avaluació.

Marcats amb rectangles vermells estan les diferències de l'esquema de classes previst inicialment i el que queda una vegada aplicades certes modificacions que s'han fet durant la fase de codificació.

La principal és la creació d'una classe nova de *avaluacions\_respostes*, on es defineixen les respostes que marquen els alumnes al realitzar les proves. Es un error d'omissió el que en l'anàlisi inicial no existís ja aquesta classe.

### 5.3 Model E/R

Mitjançant un model entitat/relació identifiquem les principals entitats de l'aplicació i com es relacionen entre elles. És el primer pas de cara a dissenyar el model relacional de la base de dades.

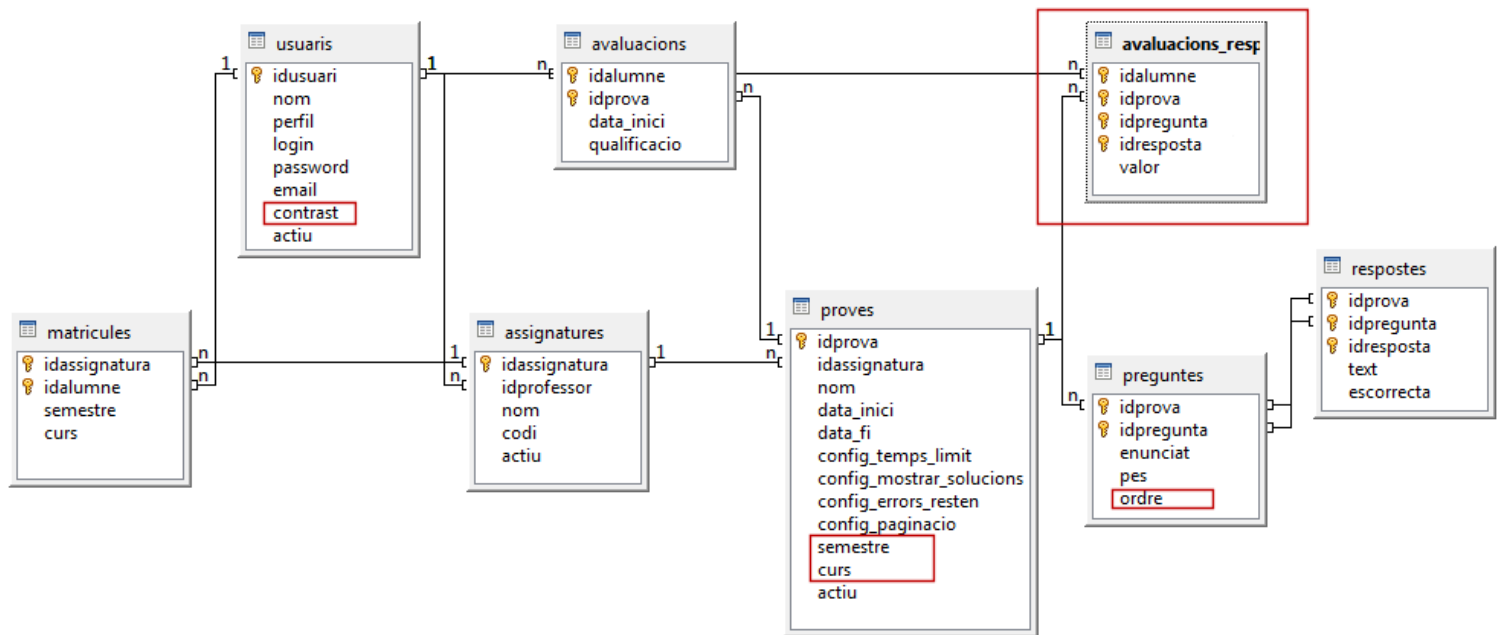


Cadascuna de les entitats disposa d'uns atributs que guardarem en la seva respectiva taula de la base de dades. Les relacions **1->n** es guarden mitjançant atributs en les taules relacionades. En canvi les relacions **n->n** necessiten una taula pròpia en la que emmagatzemar-se.

L'error comés en el diagrama de classes respecte a l'omissió de la classe d'*avaluacions\_respostes* ha anat passant de fase en fase, pel que en el model entitat-relació també hi faltava (marcada pel rectangle vermell).

### 5.4 Model relacional de la base de dades

A partir del model entitat-relació anterior i del diagrama de classes previst, es pot crear el model relacional de la base de dades, que no és més que un disseny previ de l'estructura que tindrà la base de dades de l'aplicació.



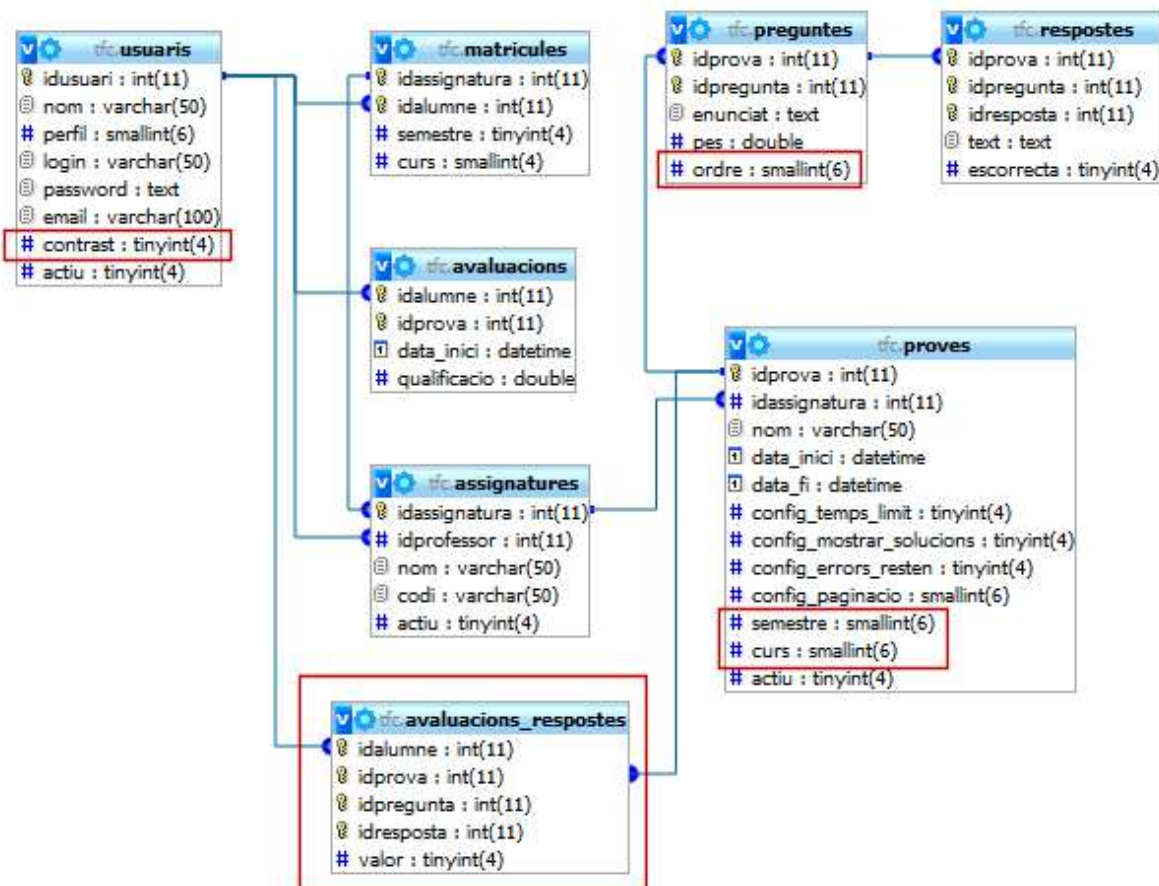
Seguint arrossegant l'error d'anàlisi, faltava afegir la taula *avaluacions\_respostes* per guardar la relació “respon” del model E/R. Marcats amb un rectangle vermell es veuen els camps afegits posteriorment.

## 6. Disseny

En aquest fase treballarem a partir de totes les decisions preses en la fase d'anàlisi per tal de dissenyar l'esquelet de l'aplicació.

### 6.1 Base de dades

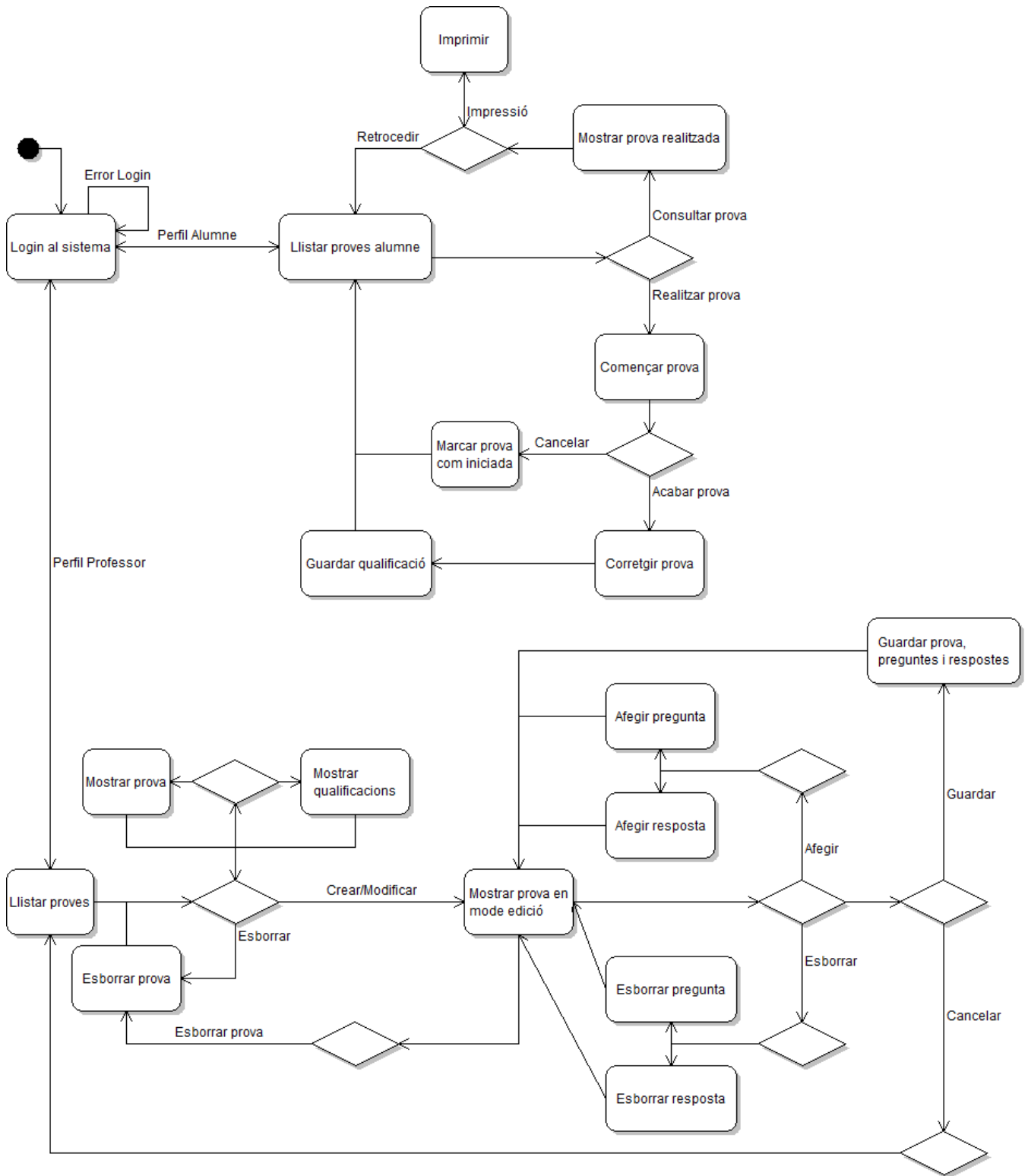
Començarem per la creació de la base de dades doncs ja disposem d'un model molt avançat del que hem d'implementar. Utilitzarem la interfície gràfica phpMyAdmin per a crear la base de dades sobre el gestor MySQL. Aquesta és la representació gràfica del dissenyador del phpMyAdmin:



La taula *avaluacions\_respostes* no contemplada inicialment està marcada amb un rectangle vermell, igual que els camps que es van afegir durant la codificació de l'aplicació

### 6.2 Diagrama d'activitat

El diagrama d'activitat és un diagrama de flux dels processos de l'aplicació. Podem veure en el nostre esquema com una vegada superada l'autenticació al sistema el flux se separa cap a una de les dues interfícies on va treballant fins a que es desconnecta tornant a la pantalla de login.





En el fil de procés que ens porta a la interfície d'alumnes, tot el procés es gestiona des de el llistat de proves disponibles per l'alumne. Des d'ell podem consultar les proves ja realitzades (amb l'opció d'imprimir-les) com començar-ne les pendents, que finalitzen quan es confirma la prova (i es rep la qualificació) o quan se surt.

El fil de procés que ens porta a la interfície de professors disposa de més opcions. També parteix del llistat de les proves disponibles a partir de les quals es poden consultar, imprimir, etc, o entrar en mode edició. En aquest subprocés podem editar i guardar les dades, afegir i esborrar preguntes i afegir i esborrar respostes.

El diagrama mostra de forma simplificada els processos, doncs seria poc aclaridor reflectir totes les possibilitats en l'esquema. En el manual d'ús de l'aplicació es detallarà amb més precisió tot el cicle.

### **6.3 Interfícies**

El format dels esquelets de les interfícies de l'aplicació s'han dissenyat directament en HTML, sense implementar cap funcionalitat de cara a trobar el disseny adient sense invertir temps en programació. Aquest esquelet després rep el format a partir de fulles d'estil CSS que es troben definides en arxius separats que són llegits al carregar-se la pàgina web al navegador.

Si les pàgines HTML son definides amb una gran dependència de les fulles d'estil, aconseguim poder canviar completament l'aparença de la interfície simplement carregant una fulla d'estil o una altra. Apart de facilitar el redisseny de les pàgines, és especialment útil a l'hora de mostrar per exemple un entorn d'alt contrast per a persones que ho necessitin.

En la nostra aplicació tenim dues fulles d'estil, una principal i una altra d'alt contrast, guardades en els arxius “*style.css*” i “*style\_contraste.css*” respectivament.

En la fase de recull de requeriments funcionals ja vam identificar les tres interfícies de les que disposaria l'aplicació. Tot i això al formar part del mateix entorn, ha d'existir una coherència visual,

un tema comú. Per tant les interfícies compartiran la mateixa fulla d'estils, tant l'estàndard com la d'alt contrast.

### 6.3.1 Login

La primera de totes és la pantalla de login, que ens permetrà l'autenticació al sistema:

**ENTORN VIRTUAL**

Usuari:

Contrassenya:

### 6.3.2 Front-end

Una vegada el sistema determini si l'usuari és un alumne o un professor, permetrà l'accés a una interfície o a una altra. En el cas dels alumnes l'entorn principal seguirà aquest disseny:

Benvingut Alex Hernandez.

[desconnectar](#)

A continuació disposes de la llista d'assignatures amb les seves proves corresponents.

[versió alt contrast](#)

| Assignatura                     | Prova                         | Estat       | Inici            | Fi               | Qualificació | Opcions   |
|---------------------------------|-------------------------------|-------------|------------------|------------------|--------------|---|
| Fonaments de programació en PHP | Tema 2 - Variables            | Realitzada  | 20/12/2012 10:00 | 31/12/2012 10:00 | C            |    |
|                                 | Tema 3 - Estructures          | En Curs     | 01/01/2013 10:00 | 20/01/2013 12:00 |              |    |
| Disseny gràfic per ordinador    | Tema 1 - Software             | Realitzada  | 23/12/2012 00:00 | 25/12/2012 00:00 | C            |    |
|                                 | Tema 2 - Eines bàsiques       | Pendent     | 05/01/2013 15:00 | 08/01/2013 15:00 |              |    |
| Algoritmes i programació        | Tema 1 - Concepte d'algorisme | Realitzada  | 15/12/2012 10:00 | 20/12/2012 10:00 | A            |    |
|                                 | Tema 2 - Tipus de programa    | Planificada | 08/01/2013 11:30 | 10/01/2013 11:30 |              |    |

**I la seva versió en alt contrast:**





Benvingut **Alex Hernandez**.[desconnectar](#)

A continuació disposes de la llista d'assignatures amb les seves proves corresponents.

[versió normal](#)

| Assignatura                     | Prova                         | Estat       | Inici            | Fi               | Qualificació | Opcions |
|---------------------------------|-------------------------------|-------------|------------------|------------------|--------------|---------|
| Fonaments de programació en PHP | Tema 2 - Variables            | Realitzada  | 20/12/2012 10:00 | 31/12/2012 10:00 | C            |         |
|                                 | Tema 3 - Estructures          | En Curs     | 01/01/2013 10:00 | 20/01/2013 12:00 |              |         |
| Disseny gràfic per ordinador    | Tema 1 - Software             | Realitzada  | 23/12/2012 00:00 | 25/12/2012 00:00 | C            |         |
|                                 | Tema 2 - Eines bàsiques       | Pendent     | 05/01/2013 15:00 | 08/01/2013 15:00 |              |         |
| Algoritmes i programació        | Tema 1 - Concepte d'algoritme | Realitzada  | 15/12/2012 10:00 | 20/12/2012 10:00 | A            |         |
|                                 | Tema 2 - Tipus de programa    | Planificada | 08/01/2013 11:30 | 10/01/2013 11:30 |              |         |

Es mostraran la llista de proves actives de les assignatures en les que l'alumne està matriculat, indicant quines estan finalitzades (i amb quina qualificació), quines estan en curs i quines estan planificades pel futur.

Per cadascuna de les proves el sistema bloquejarà o desbloquejarà les diferents opcions, com son la de realitzar la prova, consultar-la o imprimir-la, depenent de l'estat en el que es trobi.

En el cas que l'alumne vulgui realitzar la prova l'entorn seguirà el següent disseny:

### Fonaments de programació en PHP

#### Tema 3 - Estructures

1. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec eu dui mauris. Fusce lacinia velit molestie tellus rhoncus volutpat.

- a) Text resposta opció 1
- b) Text resposta opció 2
- c) Text resposta opció 3
- d) Text resposta opció 4

2. Cras felis massa, porttitor id iaculis ut, dapibus id lacus. Morbi pulvinar bibendum risus tempor volutpat. Integer nec metus ac lacus rutrum volutpat eu ac ante. Proin ac felis odio, vel tristique dolor. Pellentesque in leo leo. Nam eu nisi felis. Aliquam erat volutpat.

- a) Text resposta opció 1
- b) Text resposta opció 2
- c) Text resposta opció 3
- d) Text resposta opció 4

I la seva versió en alt contrast:

**Fonaments de programació en PHP**

**Tema 3 - Estructures**

1. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec eu dui mauris. Fusce lacinia velit molestie tellus rhoncus volutpat.

- a) Text resposta opció 1
- b) Text resposta opció 2
- c) Text resposta opció 3
- d) Text resposta opció 4

2. Cras felis massa, porttitor id iaculis ut, dapibus id lacus. Morbi pulvinar bibendum risus tempor volutpat. Integer nec metus ac lacus rutrum volutpat eu ac ante. Proin ac felis odio, vel tristique dolor. Pellentesque in leo leo. Nam eu nisi felis. Aliquam erat volutpat.

- a) Text resposta opció 1
- b) Text resposta opció 2
- c) Text resposta opció 3
- d) Text resposta opció 4

**6.3.3 Back-end**

Els usuaris amb perfil de professor accediran a un entorn diferent, amb el disseny següent:

Benvingut Pere Ubach. [desconnectar](#)

[versió alt contrast](#)

**Proves per assignatura**
**Qualificacions per alumne**

A continuació disposes de la llista d'assignatures al teu càrrec amb les seves proves corresponents.

Veure proves de semestres anteriors

| Prova                                  | Estat       | Inici            | Fi               | Realitzades | Opcions |
|--|-------------|------------------|------------------|-------------|---------|
| <b>Fonaments de programació en PHP</b> |             |                  |                  |             |         |
| Tema 2 - Variables                     | Finalitzada | 20/12/2012 10:00 | 31/12/2012 10:00 | 1/1         |         |
| Tema 3 - Estructures                   | En Curs     | 01/01/2013 10:00 | 20/01/2013 12:00 | 0/1         |         |
| <b>Disseny gràfic per ordinador</b>    |             |                  |                  |             |         |
| Tema 1 - Software                      | Finalitzada | 23/12/2012 00:00 | 25/12/2012 00:00 | 1/1         |         |
| Tema 2 - Eines bàsiques                | En Curs     | 05/01/2013 15:00 | 08/01/2013 15:00 | 0/1         |         |
| <b>Algoritmes i programació</b>        |             |                  |                  |             |         |
| Tema 1 - Concepte d'algoritme          | Finalitzada | 15/12/2012 10:00 | 20/12/2012 10:00 | 1/1         |         |
| Tema 2 - Tipus de programa             | Planificada | 08/01/2013 11:30 | 10/01/2013 11:30 | 0/1         |         |

## I la seva versió d'alt contrast:

Benvingut **Pere Ubach**.[desconnectar](#)[versió normal](#)[Proves per assignatura](#)[Qualificacions per alumne](#)

A continuació disposes de la llista d'assignatures al teu càrrec amb les seves proves corresponents.

 Veure proves de semestres anteriors

| Prova                                  | Estat       | Inici            | Fi               | Realitzades | Opcions |
|--|-------------|------------------|------------------|-------------|---------|
| <b>Fonaments de programació en PHP</b> |             |                  |                  |             |         |
| Tema 2 - Variables                     | Finalitzada | 20/12/2012 10:00 | 31/12/2012 10:00 | 1/1         |         |
| Tema 3 - Estructures                   | En Curs     | 01/01/2013 10:00 | 20/01/2013 12:00 | 0/1         |         |
| <b>Disseny gràfic per ordinador</b>    |             |                  |                  |             |         |
| Tema 1 - Software                      | Finalitzada | 23/12/2012 00:00 | 25/12/2012 00:00 | 1/1         |         |
| Tema 2 - Eines bàsiques                | En Curs     | 05/01/2013 15:00 | 08/01/2013 15:00 | 0/1         |         |
| <b>Algoritmes i programació</b>        |             |                  |                  |             |         |
| Tema 1 - Concepte d'algoritme          | Finalitzada | 15/12/2012 10:00 | 20/12/2012 10:00 | 1/1         |         |
| Tema 2 - Tipus de programa             | Planificada | 08/01/2013 11:30 | 10/01/2013 11:30 | 0/1         |         |

Es mostraran les diferents proves agrupades per les assignatures de les quals el professor és el responsable, indicant el seu estat, l'interval de dates en les que es podrà realitzar, la proporció d'alumnes que ja l'han realitzat i un conjunt d'icones d'operacions disponibles.

En funció de l'estat de la prova i la data actual s'activaran certes operacions i es desactivaran d'altres per cadascuna de les proves. Les possibles operacions son crear una prova nova (aquesta opció és a nivell d'assignatura), modificar-la, consultar-la, imprimir-la, esborrar-la, veure les qualificacions dels alumnes que ja l'han realitzat o crear-ne un duplicat (aquesta última opció es va afegir posteriorment al disseny).

La pantalla d'edició de proves, que es farà servir tant en l'alta com en la modificació, constarà de dues parts. En la part superior es podrà configurar les característiques de la prova i en la part inferior es podran anar afegint preguntes i respostes per tal de completar la prova.








Nom de la prova:

|                         |   |  |
|-------------------------|---|--|
| Data inici:             | <input type="text" value="08/01/2013 11:30"/> | (Data inici de la prova)   |
| Data fi:                | <input type="text" value="10/01/2013 11:30"/> | (Data fi de la prova)  |
| Temps límit:            | <input type="text" value="0"/>                | (Temps límit per fer la prova en minuts. Indicar 0 si no es vol limitar) |
| Paginació de preguntes: | <input type="text" value="0"/>                | (Nº de preguntes que es mostraran per pàgina, 0 sense paginació)         |
| Semestre:               | <input type="text" value="2"/>                | (Semestre en el que serà vàlida la prova)                                |
| Curs:                   | <input type="text" value="2012"/>             | (Curs en el que serà vàlida la prova)                                    |
| Els errors resten:      | <input type="checkbox"/>                      | (Indica si resta equivocar-se en les preguntes.)                         |
| Mostrar Solucions:      | <input checked="" type="checkbox"/>           | (Permetre veure les solucions d'una prova ja realitzada)                 |
| Prova activa:           | <input checked="" type="checkbox"/>           | (Si la prova no està activa no es mostrarà als alumnes)                  |

[Editar preguntes](#)


[Ordenar preguntes](#)

Pregunta nº 1 Puntuació    

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec eu dui mauris. Fusce lacinia velit molestie tellus rhoncus volutpat.

- a.  Resposta correcta  
- b.  Resposta correcta  
- c.  Resposta correcta  
- d.  Resposta correcta  


I la seva versió en alt contrast:




**Nom de la prova:**


Data inici:  (Data inici de la prova)  
 Data fi:  (Data fi de la prova)  
 Temps límit:  (Temps límit per fer la prova en minuts. Indicar 0 si no es vol limitar)  
 Paginació de preguntes:  (Nº de preguntes que es mostraran per pàgina, 0 sense paginació)  
 Semestre:  (Semestre en el que serà vàlida la prova)  
 Curs:  (Curs en el que serà vàlida la prova)  
 Els errors resten:  (Indica si resta equivocar-se en les preguntes.)  
 Mostrar Solucions:  (Permetre veure les solucions d'una prova ja realitzada)  
 Prova activa:  (Si la prova no està activa no es mostrarà als alumnes)


Editar preguntes
Ordenar preguntes


Pregunta nº 1 Puntuació  

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec eu dui mauris. Fusce lacinia velit molestie tellus rhoncus volutpat.

a.  Resposta correcta  

b.  Resposta correcta  

c.  Resposta correcta  

d.  Resposta correcta  

Els llistats de resultats i de proves d'alumnes segueixen el mateix format que el llistat de proves inicial, per tant no cal incloure'ls en aquesta secció.

## 7. Implementació

Arribant a aquesta fase tenim ja definit:

- Els requeriments funcionals i no funcionals de l'aplicació.
- Les tecnologies que aplicarem durant la implementació del projecte
- Les classes que identifiquen les diferents entitats amb els seus atributs i mètodes bàsics
- El disseny tan sobre paper com físic de la base de dades
- El diagrama d'activitats amb el flux d'execució de l'aplicació
- El disseny de les diferents interfícies en HTML i CSS

Abans de començar amb la codificació s'ha de definir l'arquitectura del sistema. Tal com es comenta en l'apartat 4.2 *Requeriments no funcionals*, s'ha triat una arquitectura tipus Model Vista Controlador (MVC) per codificar l'aplicació.

El sistema MVC separa l'aplicació en tres capes: model, vista i controlador.

- El model s'encarrega de la comunicació amb la base de dades i la gestió i lògica de les dades processades.
- La vista la formen les pàgines web que veu l'usuari de l'aplicació.
- El controlador gestiona els events i peticions generats per l'usuari des de la seva interacció amb la vista.

Aquesta arquitectura permet un manteniment i escalabilitat senzills de l'aplicació. A l'estar separades per capes podem actuar sobre una d'elles sense que afecti a les demès.

La codificació duta a terme no ha seguit un patró MVC pur, doncs sobretot durant les primeres fases de codificació el desconeixement del llenguatge i les seves possibilitats han fet que tot i seguir la filosofia de la separació en tres capes hi hagi hagut ocasions en les que no s'hagi respectat.

## 7.1 Capa model

En aquesta capa definim totes les classes PHP que s'encarregaran de la interacció amb la base de dades, ja sigui en consulta com en actualització. Cadascuna de les classes s'ha definit en un arxiu separat amb el mateix nom de la classe per tal de tenir-ho tot ben separat. Aquestes són les classes utilitzades:

- **cUsuario**: Aquesta classe gestiona tot el referent als usuaris de l'aplicació, ja siguin tan alumnes com professors.
  - Atributs: Array *info*, que conté tots els camps de la taula *Usuarios*
  - Mètodes:
    - **datosUsuario(\$idusuario)**: Consulta la taula *Usuarios* i retorna tots els camps de l'usuari especificat dins l'array atribut *info*.
    - **login(\$idusuario, \$password)**: Comprova els credencials de l'usuari i en cas d'èxit omple l'atribut *info* amb les dades de l'usuari que ha fet el login.
    - **logout()**: Buida l'atribut *info* de manera que l'usuari deixa d'estar connectat
    - **getListaPruebas(\$semestre, \$curso)**: Retorna la un objecte amb la llista de proves lligades a l'alumne per al semestre/curs especificat.
    - **getListaPruebasProfesor(\$semestre, \$curso)**: Retorna la un objecte amb la llista de proves lligades al professor per al semestre/curs especificat. Si no s'especifica semestre/curs llavors no filtra per aquests camps.
    - **cambiarContraste()**: Canvia la preferència de l'usuari respecte a veure l'aplicació en format normal o en format d'alt contrast.
  
- **cExamen**: Aquesta classe gestiona tot el referent a les proves
  - Atributs: Array *info*, que conté tots els camps de la taula *Proves*.
  - Mètodes:
    - **datosPrueba(\$idprueba)**: Consulta la taula *Proves* i retorna tots els camps de la prova especificada dins l'array atribut *info*.

- `updatePrueba()`: Actualitza el registre carregat actualment en l'array *info* en la taula de *Proves*.
- `nextId()`: Retorna el següent codi identificador lliure de la taula *Proves*.
- `insertaPrueba()`: Insereix un nou registre a la taula *Proves* amb les dades de l'array *info*.
- `deletePrueba()`: Elimina de la taula *Proves* el registre carregat actualment en l'array *info*.
- `duplicarPrueba()`: Crea una nova prova a partir de la prova carregada actualment en l'array *info*. Duplica també totes les preguntes i respostes per associar-les a la nova prova.
- `getListaPreguntas($pagina_actual)`: Retorna un objecte amb la llista de dades de les preguntes associades a la prova carregada actualment en l'array *info*. El paràmetre serveix per delimitar la consulta a l'hora de paginar la vista resultant.
- `getEstadoPruebasAlumno($registro, &$estado, &$opciones)`: A partir d'un registre de la taula *Proves* especificat calcula i retorna l'estat d'aquesta prova i un array amb les opcions que estaran actives o inactives. S'utilitza des de la interfície d'alumnes.
- `getEstadoPruebasProfesor($registro, &$estado, &$opciones)`: A partir d'un registre de la taula *Proves* especificat calcula i retorna l'estat d'aquesta prova i un array amb les opcions que estaran actives o inactives. S'utilitza des de la interfície de professors.
- `getResultadosPrueba()`: Retorna un objecte amb la llista d'alumnes que tenen disponible la prova especificada en l'array *info* i les qualificacions obtingudes.
- `getResultadosAlumnos()`: Retorna un objecte amb la llista de proves de les que és responsable un professor i les qualificacions obtingudes pels alumnes.
- `getEstadoResultados($registro, &$estado, &$opciones)`: A partir d'un registre del llistat de resultats especificat calcula i retorna l'estat de l'avaluació d'aquesta prova i un array amb les opcions que estaran actives o inactives.
- `empezarPrueba($idalumno)`: Crea (a través de la crida d'una altra classe) un nou registre a la taula *Avaluacions* per indicar que s'ha iniciat la prova carregada actualment a l'array *info*.
- `getHoraInicioPrueba($idprueba)`: Consulta (a través de la crida d'una altra classe) la

data i hora a la que es va començar la prova carregada actualment a l'array *info*.

- **esFinPrueba(\$pagina\_actual)**: Retorna un booleà especificant si estem a l'última pàgina segons la paginació de la prova carregada actualment a l'array *info*.
  - **corregirPrueba()**: Funció que corregeix la prova carregada actualment a l'array *info* i guarda la qualificació (a través de la crida d'una altra classe) a la taula *Avaluacions*.
- **cPregunta**: Aquesta classe gestiona tot el referent a les preguntes de les proves
    - Atributs: Array *info*, que conté tots els camps de la taula *Preguntes*.
    - Mètodes:
      - **datosPregunta(\$idpregunta)**: Consulta la taula *Preguntes* i retorna tots els camps de la pregunta especificada dins l'array atribut *info*.
      - **nextId()**: Retorna el següent codi identificador lliure de la taula *Preguntes*.
      - **insertaPregunta()**: Insereix un nou registre a la taula *Preguntes* amb les dades de l'array *info*.
      - **updatePregunta()**: Actualitza el registre carregat actualment en l'array *info* en la taula de *Preguntes*.
      - **deletePregunta()**: Elimina de la taula *Preguntes* el registre carregat actualment en l'array *info*.
      - **deleteRespuestas()**: Elimina de la taula *Respuestas* les respostes relacionades amb la pregunta carregada actualment en l'array *info*.
      - **getListaRespuestas()**: Retorna un objecte amb la llista de respostes relacionades amb la pregunta carregada actualment en l'array *info*.
- **cRespuesta**: Aquesta classe gestiona tot el referent a les respostes de les proves
    - Atributs: Array *info*, que conté tots els camps de la taula *Respuestas*.
    - Mètodes:
      - **datosRespuesta(\$idrespuesta)**: Consulta la taula *Respuestas* i retorna tots els camps de la resposta especificada dins l'array atribut *info*.
      - **insertaRespuesta()**: Insereix un nou registre a la taula *Respuestas* amb les dades de l'array *info*.

- **deleteResposta():** Elimina de la taula *Respostes* el registre carregat actualment en l'array *info*.
- **cAvaluaciones:** Aquesta classe gestiona tot el referent a l'avaluació de proves i alumnes
  - Atributs:
    - Array *info*, que conté tots els camps de la taula *Avaluacions*.
    - Array *inforesp*, que conté tots els camps de la taula *Avaluacions\_respostes*
  - Mètodes:
    - **datosAvaluacion(\$idalumno, \$idprueba):** Consulta la taula *Avaluacions* i retorna tots els camps de l'avaluació especificada per l'alumne i la prova dins l'array atribut *info*.
    - **datosRespuestaAvaluacion(\$idalumno, \$idprueba, \$idpregunta, \$idrespuesta):** Consulta la taula *Avaluacions\_respostes* i retorna tots els camps de la resposta d'avaluació especificada per l'alumne, prova, pregunta i resposta, dins l'array atribut *inforesp*.
    - **updateAvaluacion():** Actualitza el registre carregat actualment en l'array *info* en la taula de *Avaluacions*.
    - **insertaAvaluacion():** Insereix un nou registre a la taula *Avaluacions* amb les dades de l'array *info*.
    - **updateRespuestaAvaluacion():** Actualitza el registre carregat actualment en l'array *inforesp* en la taula de *Avaluacions\_respostes*.
    - **insertaRespuestaAvaluacion():** Insereix un nou registre a la taula *Avaluacions\_respostes* amb les dades de l'array *inforesp*.
    - **deleteRespuestaAvaluacion():** Elimina de la taula *Avaluacions\_respostes* el registre carregat actualment en l'array *inforesp*.
- **cFecha:** Aquesta classe no accedeix a la base de dades, s'encarrega de la gestió de dates
  - Atributs: *fechahora*, camp data/hora
  - Mètodes:
    - **cFecha(...):** Depenent dels paràmetres carrega en l'atribut *fechahora* la data/hora del sistema o l'especificada en el paràmetre.

- **SumaTiempo(...)**: Afegeix un interval de temps a la data especificada en l'atribut *fecha*, guardant el resultat en l'atribut mateix.
- **esMayorQue(\$fecha)**: Retorna un booleà que especifica si la data/hora de l'atribut *fecha* és major que el de la data/hora especificada al paràmetre.
- **esMenorQue(\$fecha)**: Retorna un booleà que especifica si la data/hora de l'atribut *fecha* és menor que el de la data/hora especificada al paràmetre.
- **getFechaHora()**: Retorna el valor de l'atribut *fecha*
- **getFechaHoraDMY()**: Retorna el valor de l'atribut *fecha* en format “d/M/Y H:i”
- **getFechaHoraMySQL()**: Retorna el valor de l'atribut *fecha* en format “Y-M-d H:i” que utilitza MySQL.

## 7.2 Capa Controlador

Aquesta capa controla tant els events com les peticions a la capa de model. En l'aplicació el codi de la capa controlador està separat en una serie d'arxius PHP i Javascript que es carreguen allà on son necessaris.

### Arxius PHP

- *usuarios\_control.php*

Es fa servir des del llistat inicial tan en la interfície d'alumnes com en la de professors

- “contraste”: Si rep aquesta petició es comunica amb la capa model per a canviar la configuració de contrast en les opcions de l'usuari

- *examed\_control.php*

S'utilitza en la vista encarregada de mostrar les preguntes de la prova i si s'escau examinar-se. En funció de la petició que rep s'encarrega d'inicialitzar certes variables o en alguns casos de realitzar modificacions a la base de dades:

- “realizar”: Si rep aquesta petició farà una crida a la capa model per en cas de ser necessari generar un nou registre a la taula *Avaluacions* per indicar que l'usuari acaba de començar una prova.



- “guardar”: Si rep aquesta petició farà les crides oportunes per guardar les respostes introduïdes per l'usuari en la taula *Avaluacions\_respostes*. Després si detecta que s'ha acabat la prova farà la crida per a que s'auto-corregeixi.

- *examenmod\_control.php*:

S'utilitza en la vista d'alta/modificació de proves des de la interfície de professors. En funció de la petició que rep s'encarrega d'inicialitzar certes variables o en alguns casos de realitzar modificacions a la base de dades:

- “insertar\_preg”: Si rep aquesta petició farà una crida per inserir un registre nou en la taula *Preguntes*, és a dir, crearà una pregunta nova.
- “guardar\_prueba”: Si rep aquesta petició farà una crida per actualitzar les dades de la prova que tenim carregada.
- “crear”: Si rep aquesta petició farà una crida per inserir un nou registre en la taula *Proves*, és a dir, crearà una prova nova.

- *examenmod\_bd.php*:

S'utilitza en la vista d'alta/modificació de proves des de la interfície de professors, però per a crides fetes des d'events Javascript. En funció de la petició que rep s'encarrega de realitzar modificacions a la base de dades:

- “saveorden”: Si rep aquesta petició farà una crida per actualitzar l'ordre en el que es volen mostrar les preguntes d'una prova.
- “savepreg”: Si rep aquesta petició farà les crides oportunes per guardar les dades d'una pregunta i les seves respostes associades.
- “delpreg”: Si rep aquesta petició farà les crides oportunes per esborrar la pregunta indicada i les respostes associades.
- “delresp”: Si rep aquesta petició farà una crida per esborrar la resposta indicada.
- “delprueba”: Si rep aquesta petició farà una crida per esborrar la prova indicada.
- “copiarprueba”: Si rep aquesta petició farà una crida per duplicar la prova indicada.

## Arxius Javascript (jQuery)

- alumnos.js: Gestiona els events de la vista principal de la interfície d'alumnes
  - `$(a.logout).live('click', function() {...})`: Gestió de l'event click realitzat en l'enllaç de desconnexió de la pàgina. Executa el formulari de logout de la pàgina.
- profesores.js: Gestiona els events de la vista principal de la interfície de professors
  - `$(a.logout).live('click', function() {...})`: Gestió de l'event click realitzat en l'enllaç de desconnexió de la pàgina. Executa el formulari de logout de la pàgina.
  - `$(a.delprueba).live('click', function() {...})`: Gestió de l'event click realitzat en l'icona d'esborrar prova. Envia la petició “*delprueba*” al controlador “*examenmod\_bd.php*” per a esborrar la prova indicada.
  - `$(a.copiarprueba).live('click', function() {...})`: Gestió de l'event click realitzat en l'icona de duplicar prova. Envia la petició “*copiarprueba*” al controlador “*examenmod\_bd.php*” per a duplicar la prova indicada.
  - `$("#tablink1").live("click", function() {...})`: Gestió de l'event click realitzat en la pestanya “Proves per assignatura”. S'encarrega de cridar la funció “*f\_changeTab*” per fer el canvi de secció.
  - `$("#tablink2").live("click", function() {...})`: Gestió de l'event click realitzat en la pestanya “Qualificacions per alumne”. S'encarrega de cridar la funció “*f\_changeTab*” per fer el canvi de secció.
  - `f_changeTab(cual)`: S'encarrega de fer el canvi de secció, amagant l'anterior i mostrant la nova.
- exameded.js: Gestiona els events de la consulta i realització de proves
  - `finTiempo()`: Funció que s'executa quan s'exhaureix el temps de realització d'una prova en curs, que executa el formulari de finalització de l'examen.
- examenmod.js: Gestiona els events de la part d'alta i modificació de proves
  - `$(a.delprueba).live('click', function() {...})`: Gestió de l'event click realitzat en l'icona d'esborrar prova. Envia la petició “*delprueba*” al controlador “*examenmod\_bd.php*” per a esborrar la prova indicada.
  - `$(a.delpreg).live('click', function() {...})`: Gestió de l'event click realitzat en l'icona

d'esborrar pregunta. Envia la petició “*delpreg*” al controlador “*examenmod\_bd.php*” per a esborrar la pregunta indicada.

- `$('a.delresp').live('click', function() { ... } )`: Gestió de l'event click realitzat en l'icona d'esborrar resposta. Envia la petició “*delresp*” al controlador “*examenmod\_bd.php*” per a esborrar la resposta indicada.
- `$('a.addresp').live('click', function() { ... } )`: Gestió de l'event click realitzat en l'icona d'afegir resposta. Introdueix en la pàgina HTML actual el codi necessari per fer aparèixer una resposta en blanc nova.
- `$('a.savepreg').live('click', function() { ... } )`: Gestió de l'event click realitzat en l'icona de guardar pregunta. Envia la petició “*savepreg*” al controlador “*examenmod\_bd.php*” per a guardar les dades i respostes de la pregunta indicada.
- `$('#guardar_orden').live('click', function() { ... } )`: Gestió de l'event click realitzat en el botó de guardar ordre. Envia la petició “*saveorden*” al controlador “*examenmod\_bd.php*” per a guardar l'ordre de les preguntes.
- `$("a#tablink1").live("click", function() { ... } )`: Gestió de l'event click realitzat en la pestanya “Editar preguntes”. S'encarrega de fer el canvi de secció, amagant l'anterior i mostrant la nova.
- `$("a#tablink2").live("click", function() { ... } )`: Gestió de l'event click realitzat en la pestanya “Ordenar preguntes”. S'encarrega de fer el canvi de secció, amagant l'anterior i mostrant la nova.

## 7.3 Capa vista

En aquesta capa tenim totes les pàgines HTML amb codi PHP integrat per omplir les dades obtingudes de la capa model.

Totes les pàgines comencen fent una càrrega inicial com la següent:

```
<?php
    require_once("inicializa.php");
    require_once("includes/usuarios_control.php");
?>
```

En el primer arxiu, comú a totes les pàgines tenim:

- Càrrega de les variables de sessió
- Definició d'una funció encarregada de carregar les classes necessàries
- Càrrega de les funcions de connexió amb la base de dades en l'arxiu *conecta.php*
- Càrrega de les funcions generals definides en l'arxiu *general.php*. Aquest arxiu conté algunes variables globals definides i les següents funcions:
  - `f_autocargador($clase)`: Funció que carrega una classe especificada.
  - `f_Qualificacion($nota)`: Funció que retorna la nota (A, B, C, D) en funció de la puntuació obtinguda a la prova
  - `f_limpiarString($cadena)`: Funció que neteja una cadena de text de cometes i altres expressions que puguin donar problemes al guardar-los en la base de dades
- Crides a les funcions de consulta de dades de les classes `cUsuario` i `cExamen` (amb les dades d'usuaris i proves respectivament).

El segon arxiu, que varia depenent de la pàgina web, carrega el que seria la capa de control amb les funcions JQuery encarregades de gestionar els events de la pàgina.

Més endavant, mentre definim les capçaleres de la pàgina web, es carrega l'arxiu *header.php* que és comú a totes les pàgines. Entre altres definicions conté les següents:

```
<link href="style<?php echo($g_usuario->info["contraste"]==1?"_contraste:"); ?>.css" rel="stylesheet"
type="text/css" media="all" />
<link href="js/jquery.countdown.css" rel="stylesheet" type="text/css" media="all" />
<link href="js/ui/jquery.ui.tabs.css" rel="stylesheet" type="text/css" media="all" />
<script type="text/javascript" src="js/jquery.js"></script>
<script type="text/javascript" src="js/jquery.maskedinput.js"></script>
<script type="text/javascript" src="js/jquery.countdown.js"></script>
<script type="text/javascript" src="js/ui/jquery.ui.core.js"></script>
<script type="text/javascript" src="js/ui/jquery.ui.widget.js"></script>
<script type="text/javascript" src="js/ui/jquery.ui.mouse.js"></script>
<script type="text/javascript" src="js/ui/jquery.ui.sortable.js"></script>
<script type="text/javascript" src="js/ui/jquery.ui.tabs.js"></script>
```

En la primer línia incorpora un arxiu de fulles d'estil o un altre en funció de la configuració de contrast de l'usuari. A continuació incorpora altres fulles d'estil i tots els arxius Javascript que necessita l'aplicació. Tots son arxius de la llibreria jQuery amb les definicions de diferents funcionalitats que s'utilitzen en les pàgines web.

Tot l'aspecte visual de les pàgines HTML depèn molt d'estils definits en CSS. Això ens permet realitzar canvis en la visualització de tota l'aplicació simultàniament només modificant els arxius CSS.

## 7.4 Llibreries externes

Durant la implementació de l'aplicació es van incorporar dos jocs de llibreries per tal de cobrir certes necessitats.

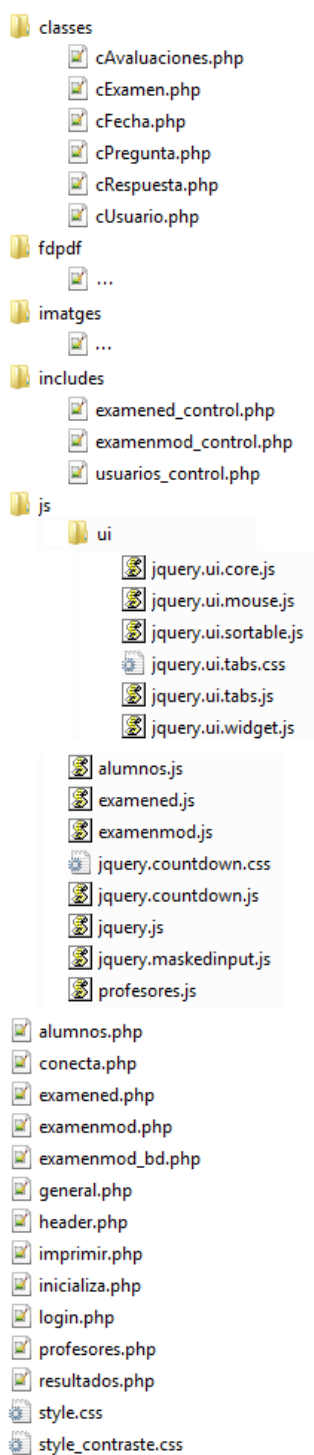
- jQuery: És una llibreria que permet l'accés i modificació d'elements de la pàgina web de forma simplificada, estalviant molta codificació Javascript. També evita haver d'invertir temps en atendre a la compatibilitat entre navegadors, doncs és la pròpia llibreria la que ho

gestiona de forma transparent al programador.

A més a més la llibreria aporta classes de controls ja programats, que han servit en diferents situacions:

- Contador de temps: Una prova pot tenir configurat un temps màxim per fer-se. jQuery Countdown aporta un objecte de compte enrere al que se li pot programar un temps i que llença un event quan el temps s'exhaureix.
- Màscara en camps de text: En la pantalla d'edició de proves de la interfície de professors, els camps de “*Data Inici*” i “*Data Fi*” han estat configurats amb una màscara d'entrada mitjançant la llibreria jQuery Maskedinput.
- Arrossegat per ordenar: En la pantalla d'edició de proves de la interfície de professors, les respostes es poden ordenar arrossegant-les amb el ratolí canviant-les de posició, el mateix que amb les preguntes dins de la pestanya “*Ordenació de preguntes*”. De la gestió d'aquestes funcionalitats s'encarrega la llibreria jQuery Sortable.
- Fdpdf: És una llibreria gratuïta per a la generació d'arxius PDF. Té moltes limitacions però compleix la funcionalitat requerida en aquesta aplicació pel que la seva senzillesa es converteix en una virtut. Així evitem haver d'instal·lar extensions al servidor web per tal de generar els arxius PDF.

## 7.5 Estructura d'arxius del projecte



## 7.6 Problemes trobats

La majoria de problemes que han sorgit durant la codificació del projecte han estat relacionats amb el desconeixement dels llenguatges per a dur a terme certes funcionalitats. A base d'anar llegint documentació de PHP, jQuery, Ajax, etc, i de moltes hores de prova/error s'han anat resolent tots els problemes.

Un tipus d'obstacle molt comú en molts llenguatges i que en la programació en PHP ha sigut igual de problemàtic és el tractament de dates i hores. MySQL fa servir un format “Y-M-d H:i:s” que després per tractar amb algunes funcions pròpies de PHP es fa complicat. Finalment es va crear una classe cFecha pròpia per tractar, encara que de manera poc ortodoxa, amb les dates en diferents formats segons les necessitats.

Un altre tipus de problema no tan lligat a la codificació en si ha sigut el de descobrir errors en el plantejament inicial que ha fet que s'hagi de repassar tot el procés d'anàlisi i disseny per tal de poder-los implementar. Alguns d'aquests errors han estat:

- Omissió d'una entitat que emmagatzemi les respostes que un alumne fa en una prova determinada. No només és necessària pel desenvolupament de la prova, sinó de cara a consultar les respostes en qualsevol moment després d'acabada la prova.

Es va afegir una nova classe al diagrama de classes inicial, una nova relació al model E/R de la base de dades i finalment una taula nova, *Avaluacions\_respostes*, al gestor MySQL.

- No es va tenir en compte l'opció de que la llista de proves es filtrés per semestre/curs, per evitar que amb el temps els professors treballessin sempre sobre una llista extensa de proves la majoria de les quals serien antigues.

Es va afegir uns nous atributs al diagrama de classes de la classe Prova i uns nous camps a la base de dades. A la vista principal de la interfície de professors es va posar l'opció de poder consultar proves antigues però que per defecte només sortissin les del semestre/curs actuals.



- Des del principi es va pensar en permetre que les respostes a les preguntes és poguessin canviar d'ordre lliurement, però no es va preveure fer el mateix amb les preguntes.

Es va afegir un nou atribut al diagrama de classes de la classe Preguntes i un nou camp a la base de dades on guardar l'ordre en el que es volen mostrar les preguntes.

## 8. Testeig

A mida que s'han anat assolint diferents fases de desenvolupament s'han anat provant per tal d'arreglar errors i polir processos. De totes maneres un mateix no és el millor tester, així que es va sol·licitar ajuda a usuaris aliens al projecte per tal de testejar l'aplicació i la seva usabilitat. Els resultats han sigut prou satisfactoris doncs només s'han hagut d'arreglar petites incidències i retocar aclarir algun text o descripció.

L'aplicació s'ha testejat en els següents navegadors:

- Mozilla Firefox: De la v15.0.1 a la v17.0.1
- Google Chrome: v23.0.1271
- Microsoft Internet Explorer: v9.0.8112

Versions més antigues dels navegadors haurien de mostrar igualment les pàgines amb mínimes diferències, a excepció de l'Internet Explorer que és el que pot presentar més problemes amb versions més antigues de la 8.

## 9. Conclusions

Aquest projecte ha estat un manera molt interessant d'introduir-se en l'anàlisi, disseny i implementació d'aplicacions web, que amb les tecnologies actuals poden assolir funcionalitats i una usabilitat molt properes a les d'eines de treball local.

Ha sigut també una manera de posar en pràctica molts coneixements teòrics que s'han anat donant durant tota la carrera, alguns dels quals costaven de veure-li la utilitat. A l'aplicar-los en el projecte de construcció d'una aplicació real molts dels conceptes i teories han cobrat més sentit.

## 10. *Manual d'ús*

Junt amb la documentació del projecte podem trobar el manual d'ús de l'aplicació, que detalla el funcionament del programa.

## 11. Bibliografia

- Google: Buscador utilitzat per la cerca de documentació i exemples de codi

<http://www.google.es>

- Wikipedia: Enciclopèdia online en castellà

<http://es.wikipedia.org>

- PHP: Web oficial del llenguatge PHP

<http://www.php.net/>

- API jQuery: Documentació de la API de la llibreria jQuery

<http://api.jquery.com/>

- API jQuery UI: Documentació de la APE de les llibreries d'interacció de jQuery

<http://api.jqueryui.com/>

- W3Schools: Documentació sobre multitud de llenguatges de programació web

<http://www.w3schools.com/>

- Tutorial jQuery: Pàgina amb multitud de tutorials interessants de la llibreria jQuery

<http://www.tutorialjquery.com/>

- XAMPP: Web de l'aplicació XAMPP (Apache, MySQL, PHP i Perl)

<http://www.apachefriends.org/en/xampp.html>

- FPDF: Web de la llibreria FPDF de creació de PDFs des de PHP

<http://www.fpdf.org/>

- MySQL: Web oficial del gestor de bases de dades MySQL

<http://www.mysql.com/>

## 12. Programari utilitzat

Aquesta és la llista de software que s'ha utilitzat en el desenvolupament del projecte:

- Editor de text per programació PHP: Notepad++, de la v.5.8.6 a la v6.1.3
- Servidor web local: Apache 2.4.2 (XAMPP 1.8.0)
- Servidor local de bases de dades: MySQL 5.5.25a (XAMPP 1.8.0)
- Interfície gràfica: phpMyAdmin 3.5.2 (XAMPP 1.8.0)
- Edició d'imatges: GIMP 2.8.2
- Navegadors de testeig:
  - Mozilla Firefox: De la v15.0.1 a la v17.0.1
  - Google Chrome: v23.0.1271
  - Microsoft Internet Explorer: v.9.0.8112

Per a la realització de la documentació:

- Processador de text: LibreOffice Writer, de la v3.5.4 a la v.3.6.4
- Edició d'imatges: GIMP 2.8.2
- Aplicació Diagrames Gantt: GanttProject 2.5.5
- Diagrames de classes i casos d'ús: Violet UML Editor 0.21.1
- Model relacional de la base de dades: LibreOffice Base 3.5.4
- Presentació: LibreOffice Impress v.3.6.4