



**Universitat Oberta
de Catalunya**

Enginyeria Tècnica en Informàtica de Sistemes

MEMÒRIA

PROJECTE FI DE CARRERA

APLICACIONS WEB PER A TREBALL COLLABORATIU

Aplicació web per a correcció automàtica de proves

Alumne: Jordi Rodríguez Díaz
Consultor: Ferran Prados Carrasco
Curs: 2011-2012

Agraïments

M'agradaria iniciar aquesta memòria amb unes paraules d'agraïment a totes les persones que ja sigui directe o indirectament han fet possible l'elaboració d'aquest projecte i la realització d'aquesta carrera.

Concretament m'agradaria agrair als meus pares per el suport que m'han mostrat i ja sigui dit per costejar la carrera ;)

Al meu soci Xavi, gracies per ensenyar-me trucs de maquetació i disseny sense tu m'hagués costat molt més.

I a la meva parella per aguantar-me i donar-me suport quan el codi no funcionava com havia de funcionar i em fotia de mala llet!.

Gracies a tots!!

INDEX

1. Introducció i descripció del projecte.....	4
1.1. Motivacions	4
1.2. Propòsit general.....	4
1.3. Objectius	5
2. Estudi de viabilitat	5
3. Metodologia	6
4. Planificació.....	7
4.1. Diagrama de Gant.....	8
4.2. Errors de planificació.....	8
5. Marc de treball i conceptes previs.....	9
6. Requisits del sistema	10
6.1. Requisits no funcionals.....	11
6.1.1. Distribució.....	11
6.1.2. Rendiment.....	11
6.1.3. Seguretat.....	12
6.1.4. Usabilitat.....	13
6.2. Anàlisi de requisits funcionals.....	14
6.2.1. Actors	14
6.2.2. Tasques/activitats dels actors.....	15
6.2.2.1. Casos d'us inicials.....	15
6.2.2.2. Fitxers de casos d'us	15
7. Estudis i decisions.....	20
7.1. Maquinari.....	21
7.2. Tecnologia.....	21
7.2.1. Llenguatge de programació.....	21
7.2.2. Bases de dades.....	21
7.3. Programari	22
7.4. Framework Codeigniter.....	22
7.4.1. Funcionament intern de Codeigniter	23

7.5.	Llibreries, biblioteques i plugins	23
8.	Disseny del sistema	25
8.1.	Model de Entitat/Relació	25
8.2.	Disseny d'interfícies.....	27
8.2.1.	Interfícies comuns	27
8.2.2.	Interfícies d'administrador.....	28
8.2.3.	Interfícies d'alumne.....	30
8.2.4.	Interfícies de professor.....	33
8.2.5.	Interfícies no prototipades.....	35
8.3.	E-mails d'avís	38
9.	Implementació.....	38
9.1.	Canvis sobre els casos d'us previstos.....	39
9.2.	Estructura de directoris i fitxers.....	40
9.2.1.	Controladors	41
9.2.2.	Models	42
9.2.3.	Vistes.....	42
9.2.4.	Helpers	43
9.2.5.	Javascript	44
9.3.	Problemes sorgits durant la codificació.....	45
9.3.1.	Validar camps amb jQuery Validate	45
9.3.2.	Col·locació d'errors de jQuery Validate	45
9.3.3.	Creació i inici de sessió de login	45
9.3.4.	Que fer al eliminar un usuari, professor o assignatura?	46
9.3.5.	Altres.....	47
9.4.	Pla de proves.....	47
9.5.	Resultats	50
10.	Treball futur	51
11.	Conclusions	52
12.	Bibliografia.....	53
Annex 1.	Instal·lació de l'aplicatiu	55
Annex 2.	Script de creació de bases de dades.....	56

1. Introducció i descripció del projecte

El present treball final de carrera es centra en el desenvolupament d'aplicacions web per a treball col·laboratiu. D'entre les propostes de l'àrea s'ha escollit la realització de una aplicació web per a la correcció automàtica de proves de tipus test.

El projecte es centre en la creació d'una eina des de la qual es puguin crear proves de tipus test per part de docents, i es puguin realitzar per part dels alumnes.

Les proves s'han de corregir de forma automàtica.

En aquesta memòria es comentaran tot el seguit d'accions i decisions que s'han portat a terme durant tota la realització del software web.

1.1. Motivacions

Actualment em trobo treballant com a programador web en la meua pròpia empresa junt a un altre soci que fa les feines de disseny (en el moment d'escriure això en la primera PAC érem tres socis), per tant la motivació principal per agafar aquesta àrea per portar a terme el meu TFC és clara.

Realitzant aquest TFC puc posar en practica els meus coneixements sobre programació web, i si la aplicació surt bé es pot arribar a incorporar en un futur en el catàleg de serveis que ofereix la empresa i comercialitzar-lo a centres d'ensenyament.

En resum aquest TFC em permetrà treballar i centrar-me en una area que em resulta de molt interès per el meu present i futur laboral i conèixer les necessitats en software online de una area que no toca la nostre empresa com es el de l'ensenyament a distancia.

1.2. Propòsit general

El propòsit general d'aquest projecte es obtenir una aplicació web que permeti la correcció automàtica de proves tipus test. El professorat disposarà d'una eina per a la creació de les proves i la gestió d'aquestes. Les proves tindran enregistrades les preguntes, respostes i solucions correctes introduïdes pels professors.

Els alumnes per altre banda enregistraran les solucions que creguin correctes en el moment de realitzar les proves.

Això permetrà que les correccions dels exàmens es realitzin de forma automàtica estalviant temps als docents i permetent als alumnes rebre una nota en un temps mínim.

1.3. Objectius

El projecte es concep com el desenvolupament d'un aplicatiu centrat en el món de la docència virtual. Una eina per a facilitar tant al alumnat com als docents la realització d'exàmens o proves de tipus test de forma ràpida i senzilla.

A partir del propòsit general anterior, es marquen el següent seguit d'objectiu a nivell de l'aplicació:

- Crear un sistema de gestió d'assignatures.
- Crear un sistema de gestió d'usuaris.
- Assignació d'assignatures als usuaris tipus alumne i professor.
- Sistema d'identificació i control de permisos segur.
- Creació, modificació i eliminació d'exàmens per part dels docents.
- Realització dels exàmens per part de l'alumnat.
- Programació d'un sistema de comparació de respostes per calcular la nota.
- Sistema de missatgeria interna entre docents i alumnat.
- Crear un entorn segur, còmode, agradable i intuïtiu.
- Implementar bones practiques i documentar codi.

A nivell personal s'espera assolir els següents objectius:

- Posar en practica tots els coneixements de la carrera d'informàtica i plasmar-los en un software web per a la creació i gestió d'exàmens i petites probes que els alumnes puguin realitzar a distancia.
- Aprofundir i ampliar coneixements principalment de les branques de bases de dades i programació, així com també de sistemes i seguretat al haver de configurar un servidor web per l'allotjament de la web.
- Aprendre a documentar un projecte o software.

2. Estudi de viabilitat

Tenint en compte en nombre d'hores disponible per a la creació del projecte s'ha optat per realitzar una solució lo més simple i adaptada del que podria arribar a ser un sistema de e-learning complet, tot i així s'han pogut complir totes les funcionalitats necessàries.

Pel que fa al maquinari i software necessari, n'hi ha prou amb un portàtil mitjanament modern on muntar un servidor local i desenvolupar el aplicatiu.

Per cobrir les necessitats de software i llibreries no hi ha cap problema ja que tot el que s'ha utilitzat es opensource i es pot descarregar gratuïtament de les corresponents pàgines.

Pel que fa al cost del projecte es pot calcular en cost d'hores de dedicació més cost del programari utilitzat.

Al considerar-se un projecte de final de carrera les hores no les tindrem en compte com a cost, ja que formen part en la activitat acadèmica.

El cost de ja em comentat que es nul ja que es pot descarregar gratuïtament.

El cost de muntar el projecte en producció en un servidor privat per donar disponibilitat al aplicatiu les 24hores del dia durant 365 dies de l'any depèn molt del proveïdor de hosting escollit i de la potencia del maquinari contractat.

3. Metodologia

Actualment s'han posat molt de moda les metodologies de desenvolupament de programari àgils. Com en el nostre cas no tenim un client marcat que ens anés dient el que volia i el que no, hem seguit una metodologia tradicional.

S'han anat realitzant les següents tasques en el següent ordre:

- Anàlisi dels requeriments funcionals i no funcionals que havia de complir la aplicació.
- Disseny de diagrames, i disseny preliminar i esquemàtic de les interfícies d'usuari
- Disseny i creació de les bases de dades necessàries per emmagatzemar la informació
- Codificació de l'aplicatiu
 - Començant per la gestió de les sessions i login.
 - Panell de gestió del administrador per poder crear usuaris.
 - Panell de gestió de professor. Llistats i creació d'exàmens.
 - Panell d'alumne. Realització, i llistats.
 - Enviament de missatges interns.
- Codificació i disseny html i css.
- Testeig i proves de funcionament.

Durant les tasques de codificació de les diferents funcionalitats s'han anat alternant junt amb les de disseny css i html, i testeig.

Com era d'esperar mentre es realitzava tota la feina de codificació per complir els requisits proposat en un primer moment, s'han hagut de modificar, eliminar i afegir funcionalitats no previstes en un primer moment. Això a fet que la realització de tasques no fos seqüencial sinó que fos un bucle en el llistat de tasques cada cop que entrava una nova funcionalitat, refactoritzant el codi cada vegada.

4. Planificació

En un primer moment es va proposar la següent planificació de tasques:

- Elaboració del Pla de treball **PAC1** (21/09/2011 a 02/10/2011)
 - Selecció entre les propostes del projecte a realitzar
 - Realització del pla de treball (objectius, planificació del treball que es realitzarà i descomposició en tasques i fites temporals)

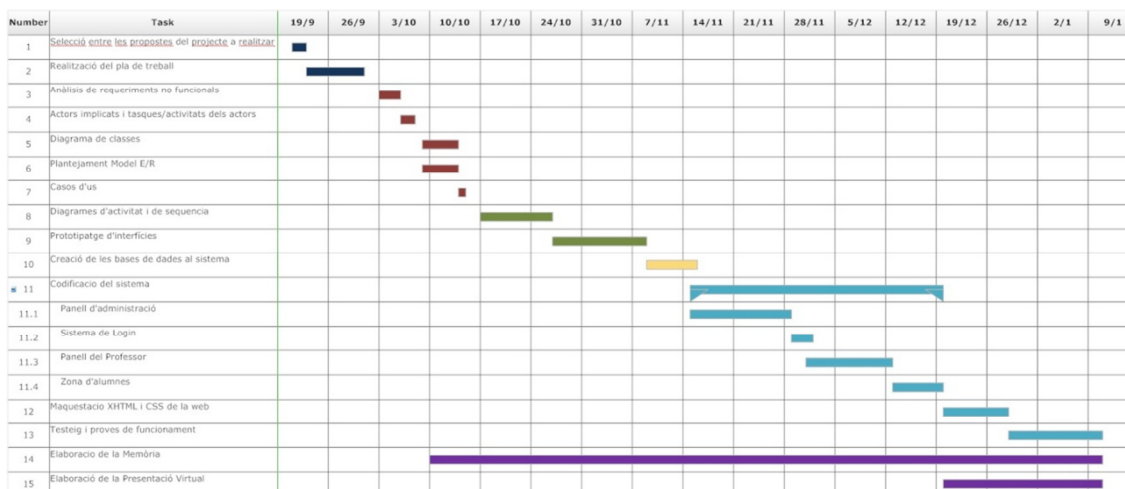
- Especificació i anàlisi **PAC2** (03/10/2011 a 16/10/2011)
 - Anàlisi de requeriments no funcionals (rendiment, distribució, seguretat i usabilitat)
 - Actors implicats i tasques/activitats dels actors
 - Requeriments no funcionals:
 - Diagrama de classes (plantejar les diferents classes que intervenen, les seves relacions i atributs)
 - Plantejar el model E/R i model relacional de les bases de dades que necessitem per guardar les dades.
 - Casos d'us (funcions que tindrà el sistema, i tasques o activitats que desenvoluparan els diferents actors)

- Disseny **PAC3** (17/10/2011 a 08/11/2011)
 - Realitzar els diagrames d'activitat (representació dels fluxos de treball)
 - Realitzar els diagrames de seqüència (interacció entre objectes)
 - Prototipatge de les principals interfícies

- Codificació **PAC4** (09/11/2011 a 10/01/2011)
 - Creació de les bases de dades plantejades anteriorment en el sistema
 - Implementació i/o adaptació de les interfícies de software necessàries:
 - Panell d'administració:
 - Creació i modificació de alumnes
 - Creació i modificació de professors
 - Permisos de alumnes i professors

- Sistema de Login (seguretat, sessions, permisos, rols)
- Panell de professor:
 - Introducció d'alumnes?
 - Creació i enviament dels tests als alumnes
 - Resultats i estadístiques dels tests realitzats (històric)
 - Comunicació de resultats a alumnes
- Alumnes:
 - Visualització i realització dels tests
 - Perfil i resultats.
- Maquetació en XHTML i CSS de la interfície gràfica de la web
- Testeig i proves de funcionament de l'aplicatiu general.
- Memòria **PAC4** (09/11/2011 a 10/01/2011)
 - Elaboració de la Memòria explicativa del treball realitzat.
- Presentació Virtual **PAC4** (09/11/2011 a 10/01/2011)
 - Elaboració de la Presentació Virtual resumint les parts mes rellevants del projecte

4.1. Diagrama de Gant



Imatge 1. Diagrama de Gant

4.2. Errors de planificació

La planificació es va seguir al peu de la lletra fins la part de la codificació. Com es pot veure en el diagrama de Gant, la part de la codificació es va plantejar de forma seqüencial, una funcionalitat darrera una altra. També es va plantejar que la maquetació començaria en acabar la programació.

Com ja he comentat en la metodologia, això no s'ha complert ni es compleix mai en el desenvolupament de software.

Sempre surten coses que s'han de canviar, modificar, i afegir perquè en un principi se t'han passat per alt, o funcionalitats que has de retirar per falta de temps.

Tampoc pots comptar en el moment de planificació amb imprevistos a nivell personal i de feina que et sortiran pel camí.

En general s'ha intentat seguir el ordre de codificació marcat i les dates, aconseguint-ho fins un cert punt.

A mig projecte em va sorgir la idea d'utilitzar AJAX tan en la gestió d'usuaris, com d'assignatures, enviament de formularis, creació dels exàmens, i en tot el que pogués. Això va fer endarrerir més del compte les tasques i haver d'alternar la maquetació amb la programació pel fet de la manipulació del DOM des de Javascript.

Degut a aquest increment de temps de les tasques junt amb la fugida d'un dels socis de l'empresa provocant la pèrdua de temps per tràmits i firmes de documents va fer que algunes de les funcions establertes en les PACS s'hagin suprimit. En propers punts s'expliquen les que s'han suprimit o simplificat.

Per últim un dels errors més grans a destacar de la planificació va ser prendre la decisió d'escriure la memòria i preparar la presentació quan l'aplicatiu estigués pràcticament enllestit. Em vaig trobar que mai estava enllestit, sempre trobava alguna cosa que afegir per millorar-ne el us.

5. Marc de treball i conceptes previs

El treball es planteja com a una aplicació web de e-learning. A continuació es detallen uns quants conceptes que s'han de tenir clars per entendre la memòria.

- **Framework:**

Representa una arquitectura de software que modela les relacions generals de les entitats del domini i proveu una estructura i una metodologia de feina.

Es pot entendre com una aplicació genèrica incompleta per on començar a treballar.

Una base per centrar-te en la lògica de la teva aplicació i extraure't de les funcions de baix nivell que ja proveu el framework.

- **MVC**

Model Vista Controlador es un patró d'arquitectura de software que separa les dades de l'aplicació, la interfície d'usuari i la lògica de negoci en tres components separats.

- **Model:** Representa la estructura de dades, solen ser classes que contenen les instruccions per recuperar dades de bases de dades, eliminar-les o modificar-les. Totes les consultes SQL estaran dintre del Model.
- **Vista:** Es la informació que es presenta al usuari, es a dir el disseny i maquetació en HTML. Tindrem aquí totes les planes en HTML que es presentaran al usuari.
- **Controlador:** Es el intermediari entre el model i les vistes. Rep la petició web, crida al model apropiat en cas de ser necessari, realitza les operacions necessàries, i retorna les dades dinàmiques a la vista perquè siguin mostrades al usuari. Tot el codi PHP.

- **Codeigniter**

Es un framework PHP que simplifica als programadors la feina de començar un projecte des de zero. Ja inclou llibreries i una estructura bàsica sobre la que començar a treballar. En apartats següents s'entrarà en més detall sobre aquest framework.

- **JQuery**

Es una biblioteca de javascript, que permet simplificar la manera d'interactuar amb documents HTML, manipular el DOM, gestionar esdeveniments, interactuar amb la tecnologia AJAX, crear animacions, etc... La gran majoria de projectes web l'utilitzen per la seva facilitat d'us i el gran numero de llibreries creades sobre ella.

- **DOM**

O *Document Object Model* es essencialment una interfície de programació d'aplicacions que proporciona un conjunt estàndard d'objectes per representar documents HTML i XML. A través del DOM els programes poden accedir i modificar el contingut, estructura i estils dels documents HTML.

6. Requisits del sistema

A continuació es descriuen els requisits, tan funcionals com no funcionals que ha d'acomplir el sistema.

6.1. Requisits no funcionals

Els requisits no funcionals, son aquells que no descriuen ni les funcions, ni les accions que portarà a terme la aplicació, sinó que fan relació a les característiques d'aquest com pot ser el rendiment, la distribució, la seguretat i la usabilitat.

6.1.1. Distribució

La aplicació es distribuirà en format de open source o de codi lliure.

El centre o docent interessat es podrà descarregar la aplicació en format comprimit i empaquetat en un zip.

Per al seu funcionament serà necessari un servidor amb suport:

- PHP versió 5.3 o superior.
- Mysql versió 5.X
- Configurat el mail per l'ús de la funció mail() de php.

Junt amb els arxius pel seu funcionament s'adjuntarà un document amb les instruccions per la seva instal·lació i configuració.

Per falta de temps s'ha hagut de prescindir del script d'instal·lació automatitzat, en substitució s'adjunta un sql per a la creació de les bases de dades, i un txt amb els arxius de configuració que s'han de modificar.

Per altre banda per als centres que no vulguin haver de contractar un servidor i muntar el sistema ells mateixos, es facilitarà una opció amb una cuota mensual on l'aplicatiu s'allotjarà en els nostres servidors i es subministrarà una adreça on ja estarà instal·lat i configurat.

6.1.2. Rendiment

El rendiment de les aplicacions online es molt important de cara a la experiència d'usuari. Com php es un llenguatge interpretat pel servidor, s'haurà de tenir present la quantitat de usuaris concurrents que pot arribar a tenir la aplicació abans de contractar un servidor o un altre.

L'aplicació en si no presenta a priori operacions de gran complexitat, per lo tant no tindria que suposar un gran estres per un servidor normal.

Pel que fa al equip dels usuaris del sistema, no ha d'haver cap problema de rendiment ja que les operacions computacionals del javascript no son de gran complexitat.

En el cas que el centre opti per la opció de descarregar i administrar ell mateix el sistema en els seus propis servidors, seria ell l'encarregat de configurar i escollir el servidor per un rendiment òptim prevén el nombre d'alumnes que faran servir l'aplicació.

En el nostre cas hauríem de realitzar tests d'estrès de la aplicació per arribar a calcular quantes aplicacions podem arribar a tenir per diferents centres en un mateix servidor o distribuir la carrega entre varis per oferir una bona experiència d'usuari.

Una altre opció possible seria utilitzar les solucions cloud computing tan de moda actualment, que permeten augmentar dinàmicament la capacitat de processament i memòria del servidor segons la carrega d'aquest en cada moment, o aixecar mes maquines de la mateixa potencia i distribuir la carrega entre aquestes i tot això automàticament.

Com a exemple de solució cloud computing tenim Amazon AWS.

6.1.3. Seguretat

El tema de la seguretat de les dades en aquest tipus de plataformes es una cosa que s'ha de tenir molt en compte.

Una mala seguretat por comportar que un alumne aconseguixi accés al servidor, això vol dir que tindria llibertat per fer el que vulgues. Canviar les preguntes, les notes, els usuaris, contrasenyes, etc..

En aquest treball ens centrarem a la seguretat en la programació de la aplicació, donarem per suposat que el servidor esta actualitzat, els firewalls activats i configurats, i en definitiva sense cap forat.

Centrant-nos en la programació segura de la aplicació s'han tingut presents els següents forats que aquesta por tenir:

- **Sessions segures** i rols ben especificats per evitar que un alumne pugui accedir com a professor o lo que seria pitjor com a administrador.
Per a complir això s'ha creat un controlador per a cada tipus d'usuari, on tindrà tots els mètodes de cada pagina que se l'hi ha de mostrar. Cada controlador verifica en el seu constructor la sessió activa i si te accés o no als mètodes.
També s'ha creat un helper pel control de sessions que agrega una signatura a les dades de sessió i la codifica.
- Filtrar totes les dades amb les quals consultem a la base de dades per evitar **injeccions sql**, que permetrien accés a la base de dades i fins i tot a la shell del sistema.
Totes les dades rebudes es filtren amb la funció `FILTER_VAR` de PHP5.

- Filtrar dades introduïdes per evitar un atac **XSS** o **Cross site scripting** que permet la execució de scripts externs amb els que robar dades del usuari o cookies. Totes les dades rebudes es filtren amb la funció `FILTER_VAR` de PHP5.
- Evitar accions per peticions GET sense comprovar el usuari loguejat. **CSRF** o **Cross site request forgery**
- Utilitzar el protocol SSL/TLS per l'intercanviar dades entre usuari i servidor per evitar que les seves dades d'accés siguin capturades per un sniffer. No s'ha implementat aquest tipus de seguretat en el servidor. Queda lliure a la elecció del centre que desitgi implementar el nostre software.

A la hora de realitzar el treball s'han tingut en compte tots i cadascun dels punts i s'explicaran junt amb el codi en apartats posteriors.

6.1.4. Usabilitat

Al realitzar una aplicació dirigida a l'usuari s'ha de tenir molt present la usabilitat d'aquesta.

Una bona usabilitat farà que l'usuari entengui el funcionament de l'aplicació a primera vista sense haver de llegir llargs manuals, ni sent cap expert.

En aquest apartat entra a tenir gran importància el disseny, casi mes que la programació. La col·locació dels els botons, els menús, colors, etc.. son coses a tenir molt presents per que la experiència d'usuari sigui lo mes bona possible.

S'ha de seguir unes pautes establertes a l'hora de col·locar tots els components i seguir-les en tots els apartats de l'aplicació, utilitzar colors agradables a la vista, diferenciar enllaços de botons, afegir llegendes o tooltips d'ajuda en els apartats més complicats, utilitzar icones descriptives, etc..

En aplicacions web es sol deixar de banda la usabilitat en les parts de backend o administració ja que no queden de cara al públic.

En el nostre cas el administrador es un usuari més de la aplicació així que s'ha de sentir còmode i l'entorn haurà de ser el mes intuïtiu possible.

Com a programadors tenim molt clar el funcionament del que hem desenvolupat, i ens pensem que a tothom li seria igual de fàcil d'entendre, però en la majoria de ocasions ens adonarem que no es així.

Pel que fa a la usabilitat de la nostre aplicació s'ha creat de manera que el seu us sigui el mes còmode i fàcil possible.

S'ha utilitzat el mateix estil en tots els apartats, totes les llistes segueixen un mateix patró, per les accions s'han col·locat icones. Com les icones poden ser confoses en segons quin cas s'han posat tooltips a cadascuna, amb lo qual quan situem el cursor per sobre ens explica la acció que realitza.

S'ha realitzat gran part de la interacció amb AJAX, per fer més còmode l'ús i evitar recarregar la pàgina cada vegada.

En cas de retornar error s'han creat unes capes o modals per informar al usuari.

En general crec que la usabilitat de la aplicació es força correcte.

6.2. Anàlisi de requisits funcionals

Els requisits funcionals recullen el que ha de fer l'aplicació per als diferents tipus actors que te. Acceptar, verificar, emmagatzemar dades, etc..

A continuació es presentaran els diferents actors de l'aplicació, els diagrames de casos d'us d'aquests i per últim es detallaran i documentaran un per un els diferents casos d'us en el fitxer de casos d'us.

6.2.1. Actors

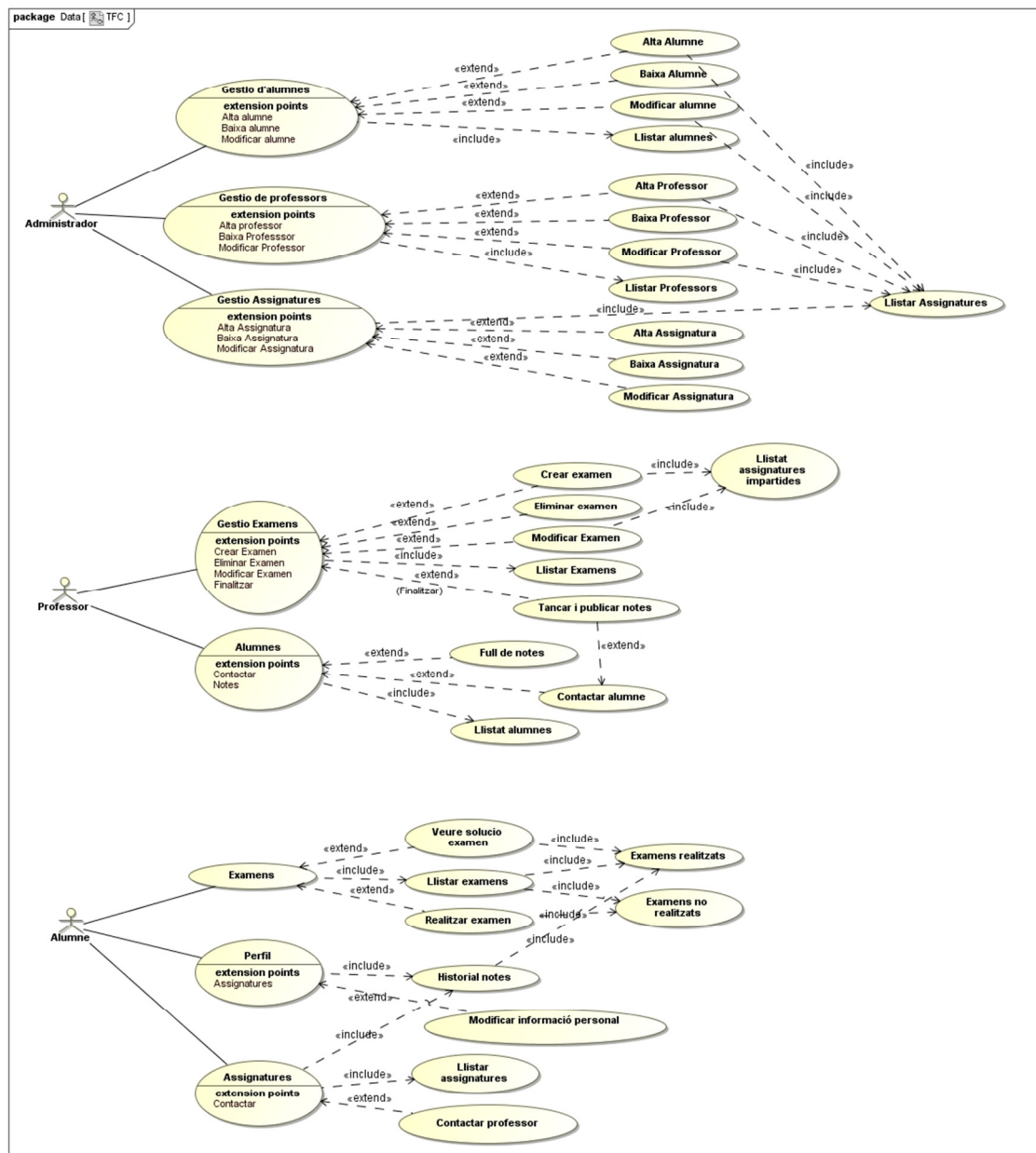
L'aplicació presenta 3 perfils d'usuari amb unes funcions ben diferenciades entre ells que son els següents:

- Per una part tenim els usuaris de tipus **administrador**, que seran els encarregats de donar d'alta i modificar els professors, els alumnes, les assignatures, els cursos, i relacionar-los entre ells.
- Per altre tenim els **professors**, encarregats de generar exàmens, publicar notes, comunicar als alumnes els resultats, etc..
- I per últim el perfil **d'alumne**, encarregat de resoldre els exàmens proposats pels seus professors.

Cada perfil te la seva pròpia zona i controlador, diferent dels demes perfils. A priori només havien de compartir la zona de login però s'han creat dos controladors més que comparteixen els perfils com son el de missatgeria i el de usuari per les funcions d'edició de contrasenya.

6.2.2. Tasques/activitats dels actors

6.2.2.1. Casos d'us inicials



Imatge 2. Casos d'ús inicials

6.2.2.2. Fitxers de casos d'us

A continuació documentem textualment el diferents casos d'us dels 3 actors implicats:

Hi ha una precondició genèrica per a tots els casos d'us que es que el actor ha d'estar autenticat com a tal. No la inclouré en el document per no repetir.

Aquest es el fitxer de casos d'us que s'havia plantejat en un principi, en el apartat de implementació es comentaran les modificacions que a sofert.

- **Administrador**

Llistar alumnes

Resum funcionalitat: Llistar els alumnes per pantalla

Casos d'ús relacionats: Alta alumne, baixa alumne, modificar alumne

Precondició: estar logat

Postcondició: aparèixer en pantalla el llistat d'alumnes.

Al entrar a la gestió d'alumnes apareixerà el llistat de tots els alumnes paginats si es precis.

Alta alumne

Resum funcionalitat: Afegir un alumne a la base de dades.

Casos d'ús relacionats: Baixa alumne, modificar alumne, Llistat assignatures

Precondició: el alumne no existeix a la base de dades.

Postcondició: el alumne esta a la base de dades.

L'administrador introdueix les dades de l'alumne, NIF, nom, i selecciona de les assignatures disponible, les que el alumne esta matriculat.

Baixa alumne

Resum funcionalitat: Eliminar de la base de dades el alumne.

Casos d'ús relacionats: Llistat d'alumnes

Precondició: el alumne existeix i apareix al llistat

Postcondició: el alumne no existeix a la base de dades.

A partir de la llista l'administrador polsarà el icona 'donar de baixa' de la fila de l'alumne i confirmarà la seva eliminació.

Modificar alumne

Resum funcionalitat: Modificar les dades basics del alumne.

Casos d'ús relacionats: Llistat d'alumnes, llistat assignatures.

Precondició: el alumne existeix i apareix al llistat

Postcondició: les dades del alumne s'han modificat a la base de dades.

A partir de la llista l'administrador polsarà el icona 'modificar' de la fila de l'alumne i entrarà en una plana on podrà editar les dades bàsiques del alumne així com també les assignatures en les que esta matriculat.

Llistar professors

Resum funcionalitat: Llistat de professors del sistema (mateixa funcionalitat llistat alumnes)

Alta professor

Resum funcionalitat: Donar d'alta en el sistema un professor (igual que alumnes, també es seleccionarà les assignatures que imparteix)

Baixa professor

Resum funcionalitat: Igual que baixa d'alumne a partir del llistat de professors

Modificació professor

Resum funcionalitat: Igual que modificació d'alumne.

Llistar assignatures, alta assignatura, baixa assignatura, modificació assignatura.

Aquests casos d'us són pràcticament iguals als de professors i alumnes per lo tant no els detallarem.

- **Professor**

Crear examen

Resum funcionalitat: Permet al professor crear un examen de tipus test

Casos d'ús relacionats: Llistar assignatures impartides.

Precondició: Estar autenticat com a professor

Postcondició: El examen amb les preguntes i les respostes correctes estan incorporades a la base de dades.

El professor crearà un examen seleccionant l'assignatura del examen i assignant-li un títol i una petita descripció o text. Seguidament passarà a introduir preguntes i les possibles respostes marcant la resposta correcta en cada cas.

Eliminar examen

Resum funcionalitat: Eliminar un examen de la llista.

Casos d'ús relacionats: Llistar exàmens.

Precondició: Existeix el examen

Postcondició: El examen no existeix a la base de dades.

A partir de la llista d'exàmens el professor eliminarà un examen clicant a la icona eliminar i confirmant la seva eliminació.

Modificar examen

Resum funcionalitat: Permet modificar un examen, les seves preguntes i respostes.

Casos d'ús relacionats: Llistar exàmens

Precondició: El examen existeix i no ha estat començat per cap alumne.

Postcondició: Les dades del examen han canviat a la base de dades.

A partir del llistat el professor pot clicar al icona de modificar examen, i a partir d'aquí modificar les dades bàsiques del examen, afegir preguntes, eliminar preguntes, afegir respostes a les preguntes, eliminar respostes, i canviar la resposta correcte a una pregunta.

Llistar examen

Resum funcionalitat: Llistat d'exàmens.

Casos d'ús relacionats: tancar i publicar notes.

Precondició: cap

Postcondició: En pantalla apareix llistat d'exàmens.

En pantalla apareix un llistat d'exàmens amb les opcions sobre cadascun d'aquests com eliminar, modificar. També apareix el numero d'alumnes que han resolt l'examen. Si hi ha temps s'inclourà una icona per accedir a les estadístiques de resultats del examen.

Tancar i publicar notes

Resum funcionalitat: Dona per finalitzat el examen i comunica les notes als alumnes.

Casos d'ús relacionats: Llistat d'exàmens, contactar alumne.

Precondició: El examen existeix i no ha estat tancat.

Postcondició: El examen esta marcat com a finalitzat i els mails han estat enviats.

Quan un examen passi de la data límit de realització, el professor tancarà el examen fent que cap alumne més el pugui respondre, el sistema calcularà els resultats de cada alumne els guardarà en el sistema i enviarà un mail a cadascun dels alumnes que han realitzat el examen comunicant les notes.

Full de notes

Resum funcionalitat: Accedir al resum de notes d'un alumne

Casos d'ús relacionats: Llistat d'alumnes

Precondició: Existeix el alumne i es alumne de una assignatura del professor.

Postcondició: Apareix per pantalla resum de notes

Es presentara per pantalla un resum de les notes d'un alumne seleccionat, de les assignatures que imparteix el professor.

Contactar alumne

Resum funcionalitat: El professor envia un missatge al alumne

Casos d'ús relacionats: Llistat d'alumnes, notes

Precondició: Existeix el alumne i es alumne de una assignatura del professor.

Postcondició: Mail enviat al alumne.

A partir del llistat d'alumnes, el professor pot escollir un per enviar-li un mail o missatge. Se li mostrarà un requadre per a escriure el text del mail.

També pot comunicar-li automàticament les notes d'un examen, o el resum de notes.

Llistat alumnes

Resum funcionalitat: Es mostra un llistat dels alumnes del professor

Casos d'ús relacionats:

Precondició: Cap

Postcondició: Apareix en pantalla un llistat d'alumnes

El professor pot accedir a la llista d'alumnes de les assignatures que imparteix. A partir de la llista pot comunicar-se amb ells, o accedir al resum de les seves notes.

- **Alumne**

Llistar exàmens

Resum funcionalitat: Llistat d'exàmens separats per assignatures, tan realitzats com per realitzar.

Casos d'ús relacionats: Realitzar examen, veure solució.

Precondició: Estar autenticat com alumne

Postcondició: Llistat per pantalla d'exàmens.

Només autenticar-se el alumne accedirà a un llistat d'exàmens. Els exàmens vindran separats per assignatures. Es llistaran els realitzats i els no realitzats.

Els realitzats apareixerà la nota en cas que s'hagi publicat i un icona per veure en detall.

Els no realitzats mostraran la data màxima per la seva realització, i un icona per realitzar-lo.

Realitzar examen

Resum funcionalitat: Permet al alumne resoldre un examen.

Casos d'ús relacionats: Llistar exàmens.

Precondició: El examen existeix, no ha passat la data límit i no ha estat respost per l'alumne anteriorment.

Postcondició: Les respostes del alumne han estat guardades a la base de dades.

El alumne accedeix al examen, permeten llegir les preguntes i seleccionar les respostes que cregui correctes. Un cop finalitzat l'enviarà i no podrà tornar enrere a corregir cap resposta

Veure solució examen

Resum funcionalitat: Veure la solució del examen i on a fallat al contestar

Casos d'ús relacionats: Llistar exàmens.

Precondició: El examen existeix, a estat respost, ha finalitzat, i el professor la donat per tancat.

Postcondició: Mostra examen per pantalla i solucions.

Es mostra al alumne el examen realitzat en el seu dia, amb les seves respostes marcades en verd en cas de ser correctes, i en cas de ser incorrectes es marcarà la seva resposta en vermell i la correcte en negre.

Historial de notes

Resum funcionalitat: Mostra el expedient de notes

Casos d'ús relacionats:

Precondició: Alumne autenticat

Postcondició: Mostra notes per pantalla.

Es mostrarà un llistat de les assignatures en les que esta matriculat i per cada assignatura els exàmens realitzats amb les notes de cadascun.

Modificar informació personal

Resum funcionalitat: Permet al alumne afegir informació al seu perfil.

Casos d'ús relacionats:

Precondició: Alumne autenticat

Postcondició: Dades del alumne modificades a la base de dades

El administrador donarà d'alta el alumne amb unes dades bàsiques: NIF i nom. L'alumne podrà ampliar la informació posant una foto de perfil, un text, aficions i lo que sens acudeixi.

Contactar professor

Resum funcionalitat: Permet al alumne contactar amb el professor d'una assignatura

Casos d'ús relacionats:

Precondició: Alumne autenticat, alumne matriculat en assignatura del professor.

Postcondició: Mail o missatge enviat al professor

El alumne podrà contactar a traves de l'aplicatiu amb el professor d'una assignatura en concret a partir d'un llistat d'assignatures en les que esta matriculat. Se li mostrarà un camp de text per escriure el missatge.

7. Estudis i decisions

En aquest apartat es descriu el maquinari i tecnologia i s'enumeren les llibreries i programari utilitzat en la realització del projecte, tot justificant les decisions.

7.1. Maquinari

Per a la realització i desenvolupament del projecte s'ha utilitzat com a maquinari el meu portàtil particular, un DELL XPS15z amb les següents especificacions de hardware: Intel Core i7-2620M a 2.70GHZ i 8GB de RAM

7.2. Tecnologia

En quant a tecnologia utilitzada podem separar la del llenguatge de programació i la de bases de dades.

7.2.1. Llenguatge de programació

Dels tres possibles llenguatges que es proposaven per a la realització de la aplicació web, s'ha optat per realitzar-la en PHP.

Dels 3 llenguatges proposats PHP era el únic que havia utilitzat prèviament, els altres 2 tant el JSP com el .NET no els he tocat mai, i tampoc em criden gaire. Son dos llenguatges que a no ser que els necessiti aprendre per obligació no em posaria a estudiar.

La elecció es clarament justificada ja que si hagués hagut d'aprendre un nou llenguatge que mai he tocat a la part que realitzar tot el projecte, no crec que hagués aconseguit ni la meitat de funcionalitats.

7.2.2. Bases de dades

Pel que fa al sistema gestor de bases de dades s'ha optat per mysql per el èxit demostrat que te el sistema, i el gran us estès d'aquest en aplicacions semblants. També es podia haver optat per utilitzar Postgree ja que l'haviem donat durant el curs.

La elecció ha estat un com podia haver estat l'altre.

Una altre possibilitat hagués estat intentar utilitzar un sistema NOSQL tan en boca de tots últimament com Cassandra o MongoDB però no em semblava adient per aquest tipus de projecte. No aportava ninguna millora i era perdre el model relacional.

Pel que fa al sistema d'emmagatzematge de dades en mysql s'ha optat per INNODB enfront de MYISAM ja que no volia prescindir de les transaccions en segons quines operacions de esborrat i emmagatzematge i perdre la integritat de les dades.

7.3. Programari

Pel que fa al programari utilitzat per a la realització del projecte ha estat el següent:

- **XAMPP** - <http://sourceforge.net/projects/xampp/>
Servidor web lliure que incorpora apache, intèrpret de php, mysql y phpmyadmin
- **Eclipse + plugins php, js, css** - <http://www.eclipse.org/>
Entorn de desenvolupament de gratuït i multiplataforma utilitzat per a la codificació de php, js i css
- **Notepad++** - <http://notepad-plus-plus.org/>
Editor de text lliure amb suport per múltiples llenguatges utilitzat per a la edició en remot dels arxius un cop penjats al servidor.
- **Filezilla** - <http://filezilla-project.org/>
Client FTP lliure i gratuït per pujar l'aplicació al servidor
- **Photoshop**
Editor gràfic utilitzat per el disseny de les interfícies, botons, etc...
- **MySQL Workbench** - <http://www.mysql.com/products/workbench/>
Eina visual per el disseny de bases de dades.
- **Navicat Lite** - <http://www.navicat.com/>
Versió gratuïta de l'eina d'administració de bases de dades per a la modificació de valors i testeig
- **Microsoft Word i Powerpoint**
Per a la creació de la memòria i la presentació virtual

7.4. Framework Codeingiter

Codeingiter es un framework PHP que utilitza un patró MVC.

S'ha escollit aquest framework ja que es un projecte mitja i està més que provada la seva eficàcia i velocitat en aquest tipus de projectes. Si hagués estat un projecte mes

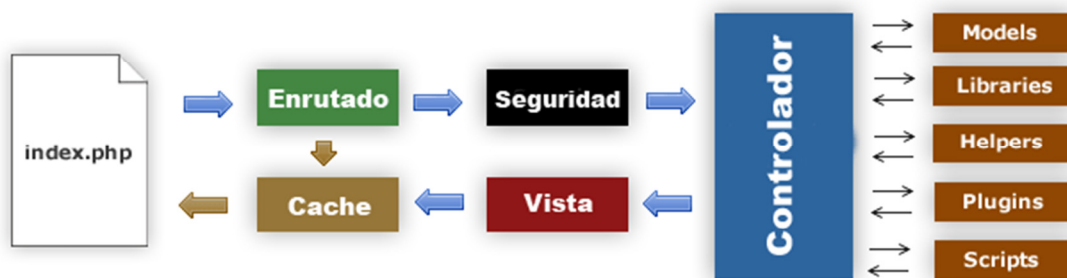
gran igual haguéssim escollit Symfony o Zend que son frameworks molt mes robustos.

Codeigniter inclou tot un seguit de llibreries i funcions que ajuden al desenvolador pel tractament de sessions, bases de dades, urls amigables, etc...

Seria molt extens explicar cadascuna d'aquestes llibreries.

En el apartat d'implementació s'intentarà fer un resum de les utilitzades en el projecte.

7.4.1. Funcionament intern de Codeigniter



Imatge 3. Diagrama funcionament Codeigniter

En el diagrama anterior es pot observar el funcionament intern del framework.

El index.php inicialitza el nucli de codeigniter, el encaminador examina la petició HTTP i determina que s'ha de fer. En cas que existeixi cache de lo demanat, es retorna el HTML sense passar pel sistema. Si no hi ha cache el mòdul de seguretat controla i filtra totes les dades enviades abans de fer-los arribar al controlador.

El controlador carregarà els models, llibreries, plugins etc. necessaris per realitzar les operacions.

El controlador enviarà les dades a la vista oportuna, es renderitzarà el HTML a retornar, i si la cache esta activada es guardarà per una futura petició, y finalment s'enviarà el HTML al usuari que ha iniciat la petició HTTP.

7.5. Llibreries, biblioteques i plugins

- **960 Grid Css** - <http://960.gs/>

Es una biblioteca CSS per facilitat la maquetació web. Et permet crear un container de 960px d'amplada d'aquí el seu nom, i a partir d'uns estils anar maquetant el contingut en columnes de X amplada.

- **jQuery 1.7.1** - <http://jquery.com/>

Es la biblioteca Javascript més estesa per a facilitat la manipulació del DOM, controlar peticions en AJAX, gestionar esdeveniments...

S'utilitza en el projecte principalment en les peticions AJAX i es necessària per a les següents llibreries/plugins.

- **jQuery Ui 1.8.16** - <http://jqueryui.com/>

Es una llibreria que s'afegeix a jQuery que permet la utilització de una gran quantitat de widgets i animacions extres. En el projecte només s'ha carregat el widget per seleccionar dates. Era necessari aquest plugin per poder fer servir el següent.

- **jQuery Timepicker Addon** - <http://trentrichardson.com/examples/timepicker/>

Es un plugin que estén la funcionalitat del DatePicker de JQuery UI permeten afegir selecció d'hora, minuts i segons. S'ha escollit aquest entre els pocs que es van trobar que permetessin seleccionar data i hora per ser el més complet i agradable a la vista.

Es utilitzat en la creació d'exàmens per part dels professors per seleccionar la data d'inici i fi d'aquest de manera més còmode i usable.

- **jQuery Colorbox** - <http://jacklmoore.com/colorbox/>

Plugin de jQuery per mostrar lightboxes, que són les típiques capes que es mostren per sobre el contingut de tota la web posant la resta més fosc. Es dels més utilitzats en aquesta funcionalitat.

S'utilitza per mostrar el retorn de errors en peticions AJAX en comptes de mostrar el típic alert de Javascript, i mostrar el formulari d'enviament de missatges entre professors i alumnes.

- **jQuery Form Plugin** - <http://jquery.malsup.com/form/>

Plugin de jQuery que fa que els formularis de una pagina web s'enviïn per AJAX de manera asíncrona. S'utilitza en la gran majoria de formularis per fer una navegació més ràpida i clara.

- **jQuery Validate** - <http://bassistance.de/jquery-plugins/jquery-plugin-validation/>

Plugin de jQuery que et permet establir unes regles a cada camp d'un formulari i evita que sigui enviat si no es compleixen les regles establertes amb les dades omplertes. En cas de no complir-se les regles et mostra quins camps i perquè estan malament amb un error.

S'utilitza per validar tots els formularis de la banda del client abans d'enviar-se i ser validat per el servidor.

- **jQuery Tipsy** - <http://onehackoranother.com/projects/jquery/tipsy/>

Aquest plugin jQuery no el tenia previst implementar en un principi però el vaig veure un cop acabat el projecte i em va agradar. Serveix per posar títols d'ajuda a un objecte HTML i que es mostrin al passar el cursor per sobre d'aquest.

Em va sembla útil implementar-lo en tots els icones per aclarir la acció que realitzen cadascun.

Fins aquí el resum totes les llibreries i plugins que s'han utilitzat en aquest projecte. En l'apartat d'implementació mostraré com s'han utilitzat.

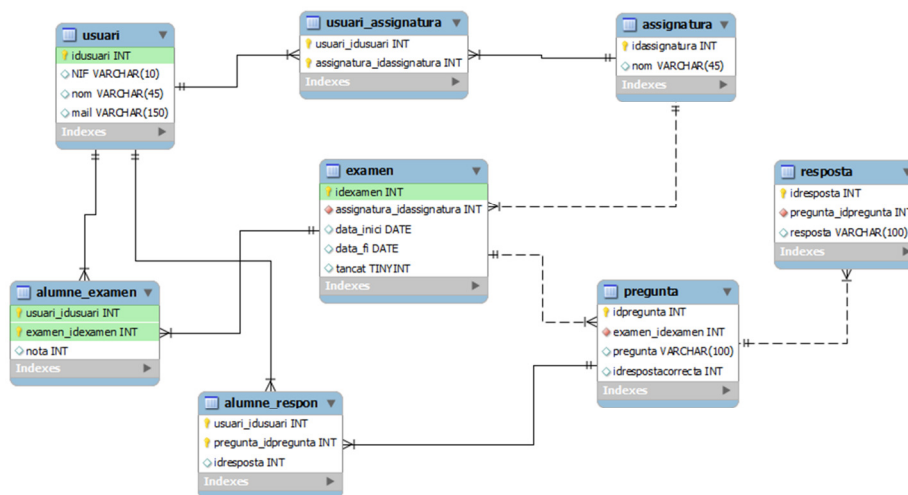
8. Disseny del sistema

En aquest apartat es presenten el disseny de les interfícies d'usuari, els diagrames de les bases de dades, la distribució de fitxers, i les bases de dades físiques tot comparant la primera proposta amb la final.

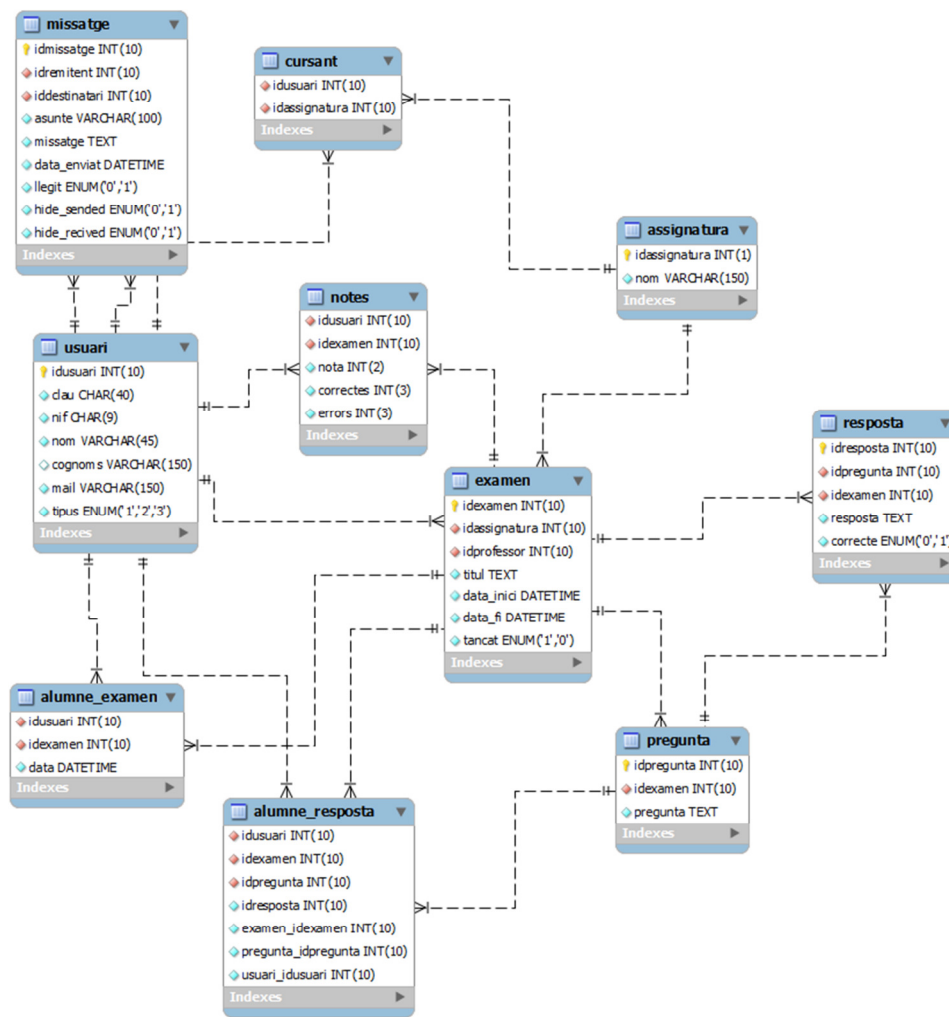
8.1. Model de Entitat/Relació

Es mostra els esquema Entitat/Relació del anàlisi de les bases de dades en el moment de plantejar el projecte i es compara amb l'esquema de la base de dades final.

S'analitzen les modificacions que a patit i es justifiquen aquestes.



Imatge 4. Diagrama E/R - Inici del projecte



Imatge 5. Diagrama E/R - Fi del projecte

La taula que en un principi tenia el nom d'usuari_assignatura ha passat a dir-se cursant. S'ha creat una nova taula notes relacionada amb examen i alumne on en guardaran les notes del exàmens, el numero de respostes errònies i correctes.

Les taules alumne_resposta, resposta, passen a estar relacionades amb examen per unes consultes més còmodes.

La taula examen guarda en id de professor que l'ha creat, ja que al final una mateixa assignatura pot tenir varis professors que la imparteixin i creïn exàmens sobre ella. D'aquesta manera podem saber les dades del professor que ha creat l'examen i enviar-li un missatge. També serveix per comprovar que només pot editar, corregir i veure les notes d'un examen el professor que l'ha creat.

El camp per marcar la resposta correcte passa de la taula pregunta a la taula resposta. D'aquesta manera podem comparar el array de respostes correctes amb el de respostes d'alumne.

S'ha afegit tots els camps que en un principi no es van posar.

Per últim s'ha creat una taula missatges on guarda els missatges interns enviats entre usuaris.

8.2. Disseny d'interfícies

Comparem el prototipatge de les interfícies en un primer moment amb el disseny final de l'aplicatiu.

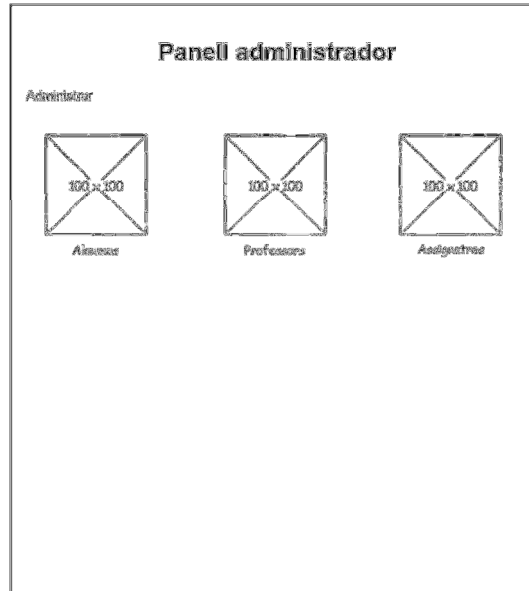
8.2.1. Interfícies comuns

Interfície 1. Login (Esquema)

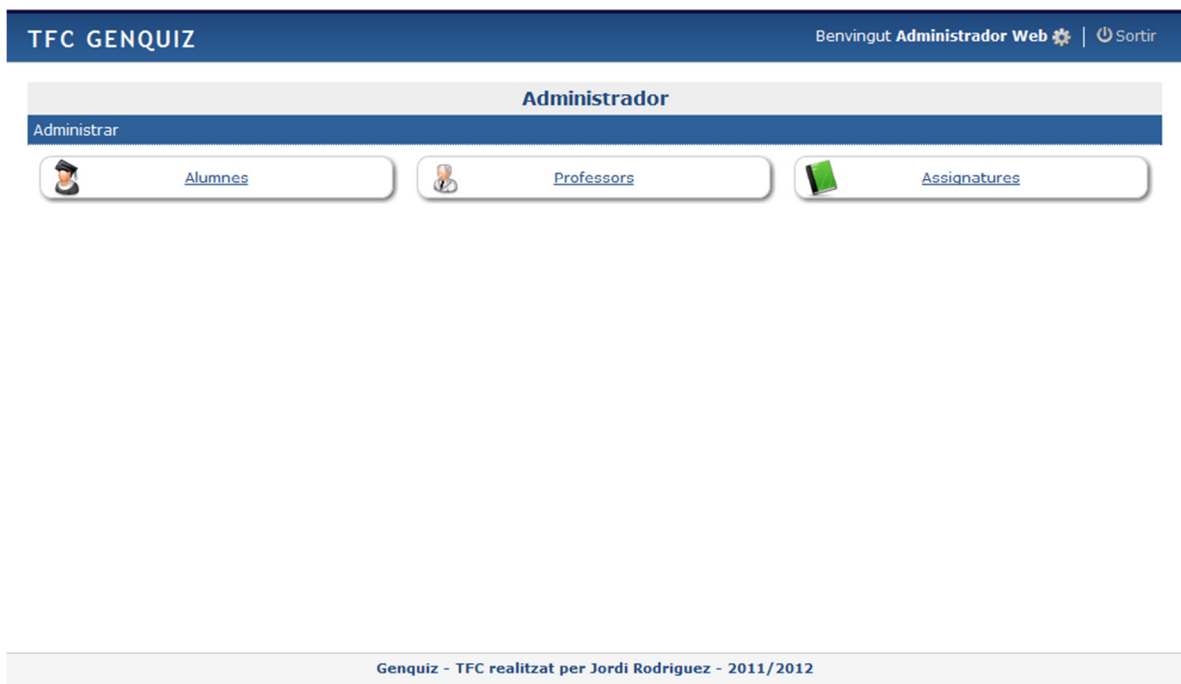
Interfície 2. Login (Final)

Es pot dir que la interfície de login ha seguit la mateixa pauta que l'esquema

8.2.2. Interfícies d'administrador

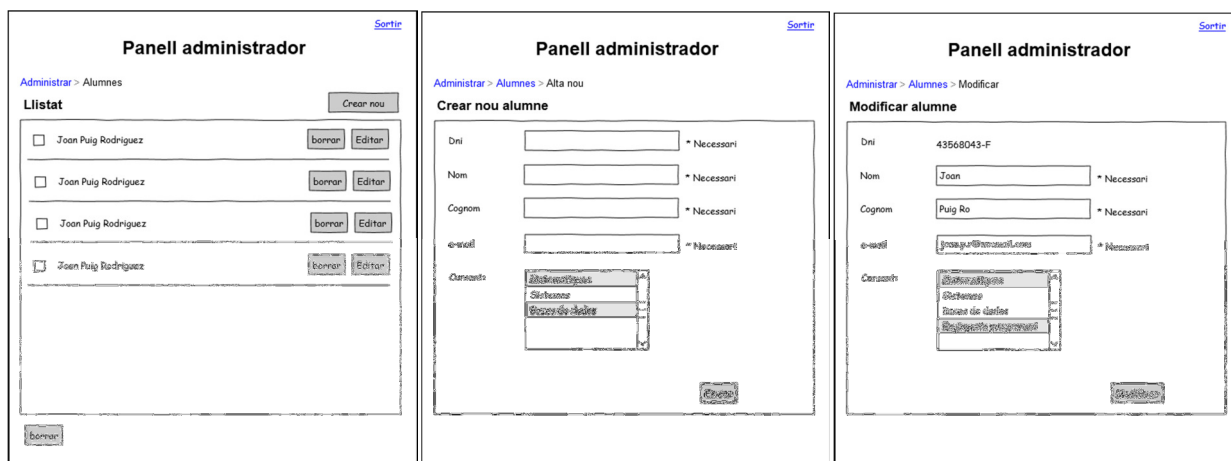


Interfície 3. Dashboard administrador (esquema)

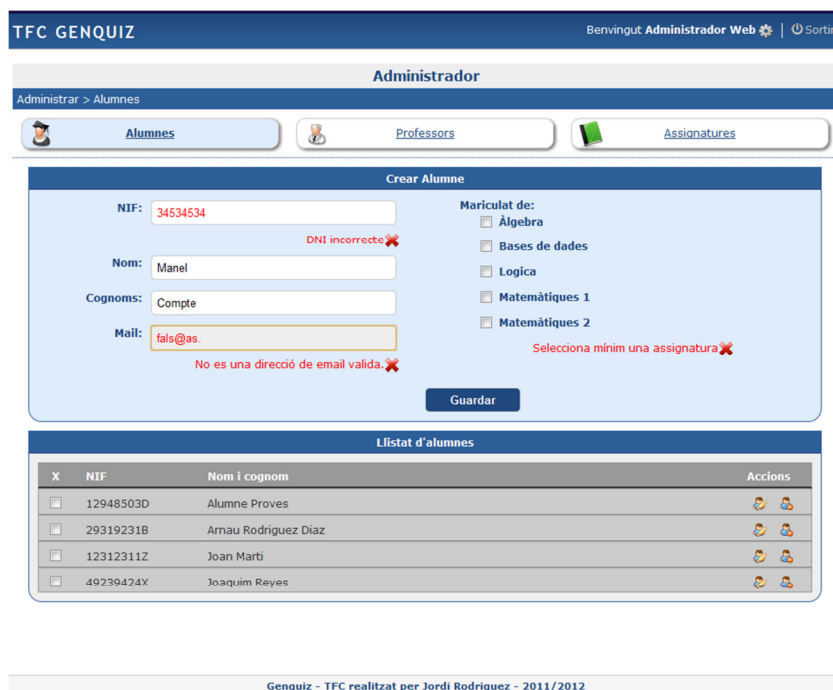


Interfície 4. Dashboard administrador (final)

S'han afegit icones, botons en relleu per accedir a les funcions principals, i totes les pàgines tenen el mateix header i footer.



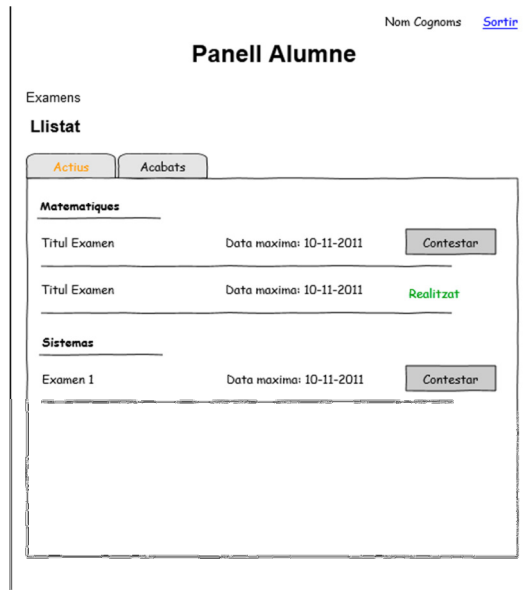
Interfície 5. Llistat d'alumnes, afegir i editar (esquemes)



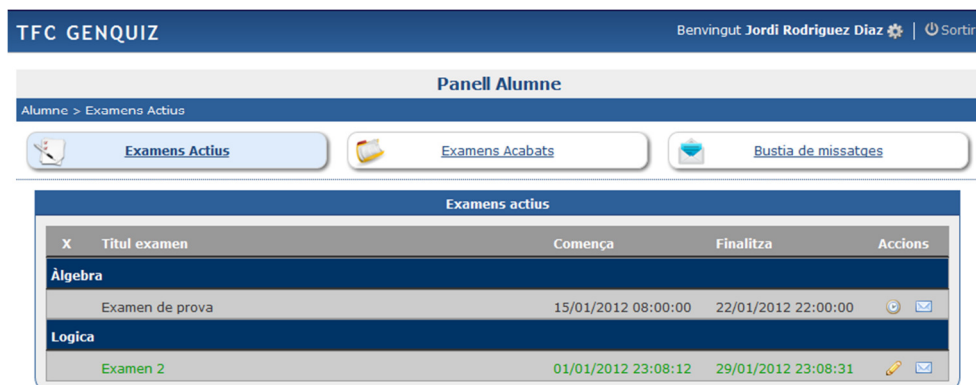
Interfície 6. Administració Alumnes (final)

En aquesta interfície si que hi ha hagut un canvi substancial. En una sola pàgina s'ha ajuntat gracies a AJAX i jQuery totes les funcions. Es mostra el formulari per introduir alumnes nous, a sota el llistat d'alumnes, i en cas de clicar sobre editar es carreguen les dades al formulari per editar-lo. Cada cop que es guarda un alumne nou, s'edita o s'esborra, s'actualitza automàticament el llistat. El validador mostra errors si els camps son incorrectes. La interfície de administrador de professors es idèntica a aquesta.

8.2.3. Interfícies d'alumne



Interfície 7. Llistat exams actius (esquema)



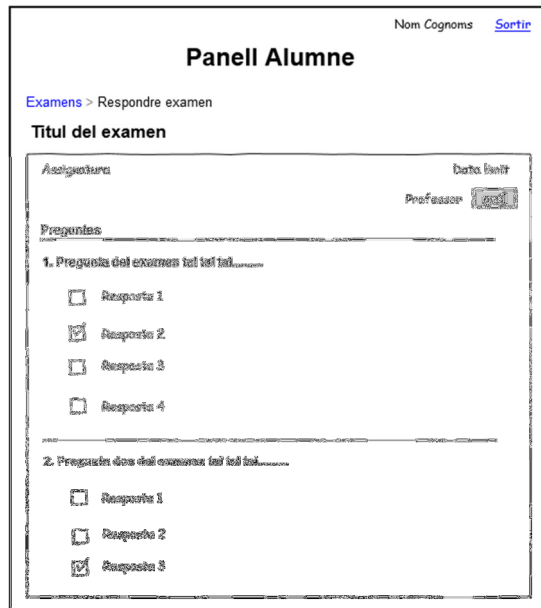
Interfície 8. Exàmens actius (final)

La idea principal es la mateixa, però es conserva el menú de navegació en totes les planes de l'alumne. S'ha afegit la data d'inici del examen.

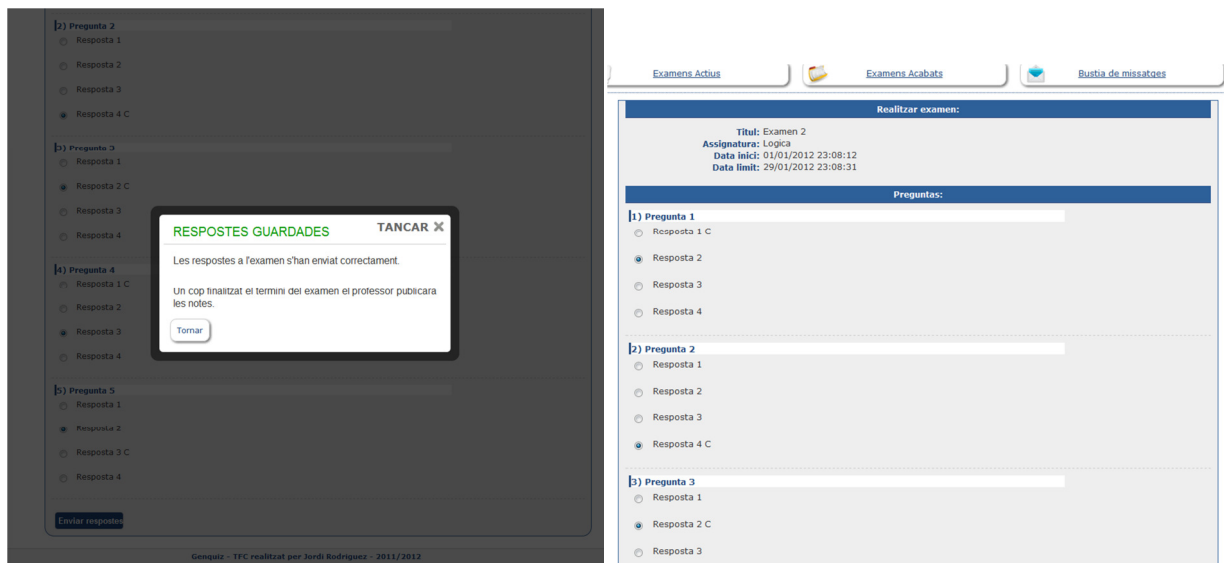
S'ha afegit icones per a realitzar les accions. En aquesta pàgina n'hi ha quatre de diferents.

- 🕒 El rellotge significa que encara no esta obert (data inici>actual)
- 📌 Al clicar al llapis entres en la realització del examen.
- 🟢 Si ja l'has realitzat però encara no ha finalitzat apareix un stick verd
- 📧 I el sobre obre una modal per enviar missatges al professor que ha creat el examen.

El llistat d'exàmens finalitzats es igual amb els icones d'acció de veure els resultats o esperar la correcció.

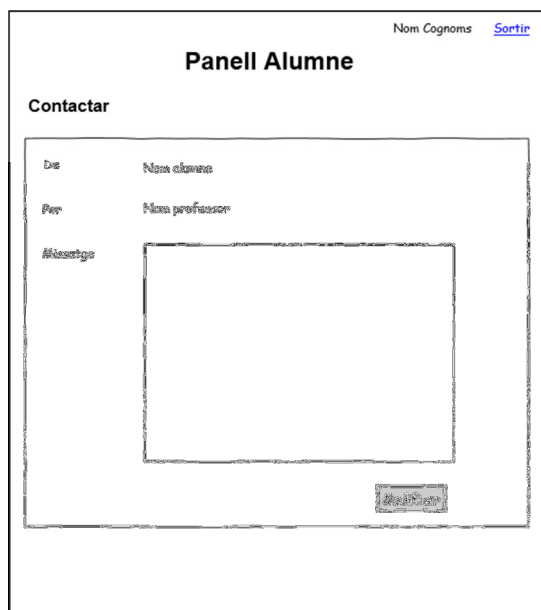


Interfície 9. Realitzar examen (esquema)



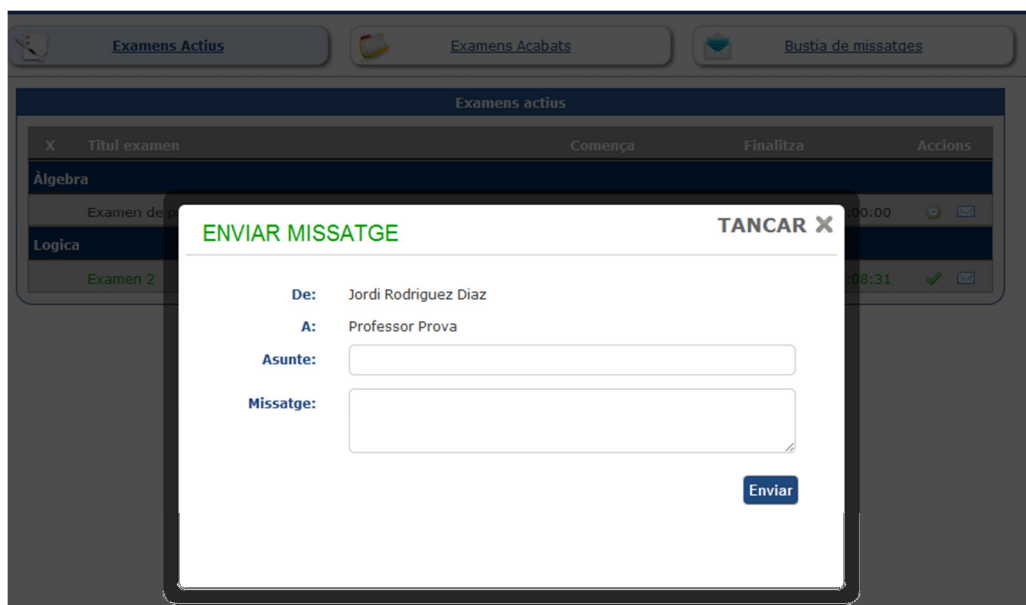
Interfície 10. Realitzar examen + modal respostes enviades (final)

En aquest cas el plantejament inicial i el resultat final també son molt similars. El enviament de respostes es realitza per AJAX i es mostra una modal informativa al alumne si tot a funcionat correctament.



The wireframe shows a page titled "Panell Alumne" with a "Contactar" section. It includes a form with fields for "De:" (Name of student), "Per:" (Name of professor), and "Missatge:" (Message). A "Enviar" button is located at the bottom right of the form area. The page also has a "Nom Cognoms" label and a "Sortir" link in the top right corner.

Interfície 11. Enviar missatge (esquema)



The screenshot shows a modal window titled "ENVIAR MISSATGE" with a "TANCAR X" button. The form is pre-filled with "De: Jordi Rodriguez Diaz" and "A: Professor Prova". There are input fields for "Asunte:" and "Missatge:". An "Enviar" button is at the bottom right. The background shows a table of exams with columns for "Titul examen", "Comença", "Finalitza", and "Accions".

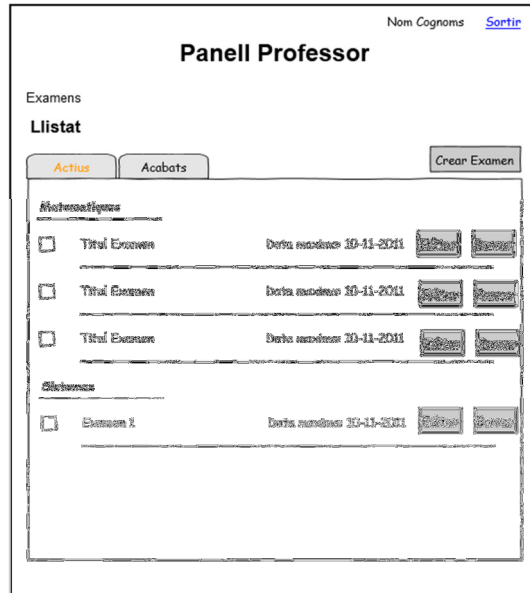
Interfície 12. Enviar missatge (final)

Aquesta es va plantejar en un principi com una pàgina on aparegués un formulari per enviar el missatge. En la versió final gracies a la utilització de colorbox, s'ha confeccionat el enviament de missatges en una modal sense perdre el punt de pagina on estar.

Es una manera molt mes rapida, simple i còmode d'enviar el missatge al usuari i tornar tancant la modal al punt on estaves.

Passa a ser una interfície comú tant per alumnes com professors.

8.2.4. Interfícies de professor

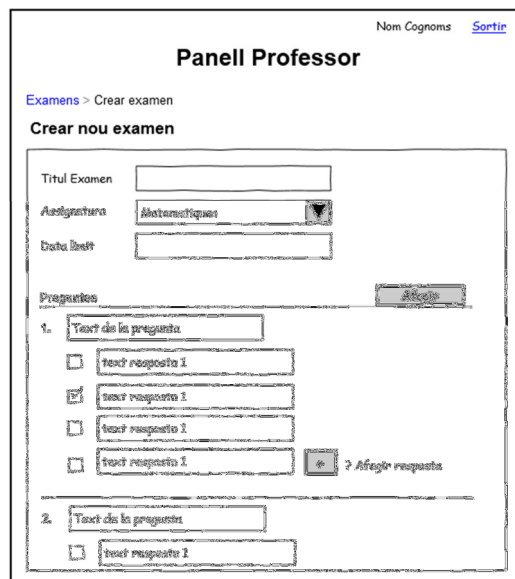


Interfície 13. Llistat exàmens actius (esquema)

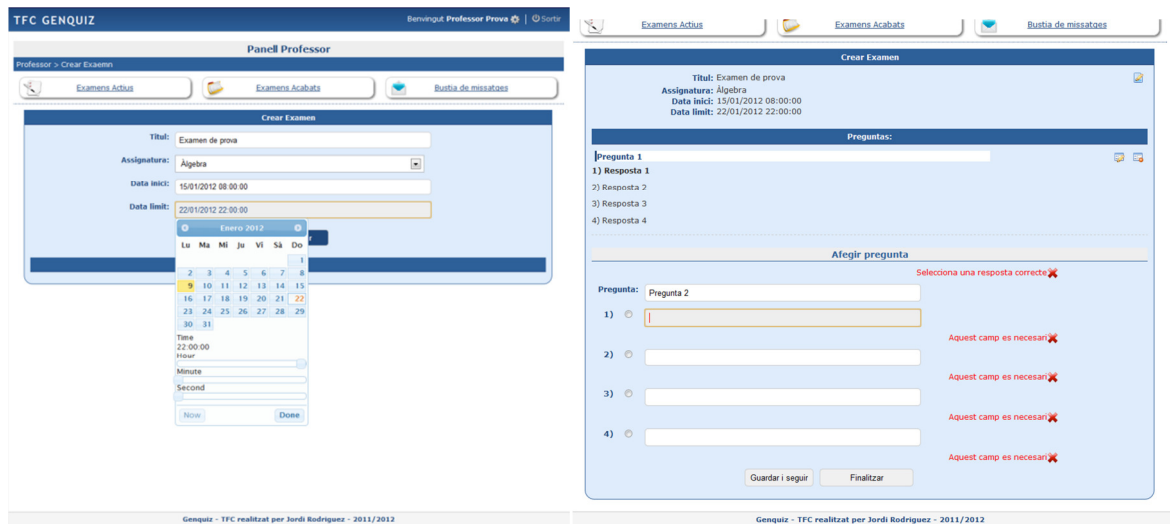


Interfície 14. Llistat exàmens actius (final)

Tampoc a canviat molt de l'esquema original. S'ha afegit la data d'inici, s'han canviat els botons per icones, i s'ha eliminat la opció de editar i eliminar l'examen en cas d'haver començat ja que no seria lògic que un alumne hagués respost un examen i després sigues modificat.



Interfície 15. Crear examen (esquema)



Interfície 16. Crear examen + afegir preguntes (final)

S'ha afegit un camp de data d'inici i s'ha col·locat un calendari per escollir les dates i les hores. S'ha suprimit el afegir respostes a cada pregunta. Al omplir el formulari de pregunta s'han col·locat 2 botons un per guardar la pregunta i seguir afegint, i un altre per finalitzar la creació de preguntes.

En tot moment es poden editar les dades del examen i les preguntes i respostes. Això fa que es canviïn les dades de tipus texts per camps de formulari. Al guardar la edició oculta els inputs i mostra de nou el text modificat.

8.2.5. Interfícies no prototipades

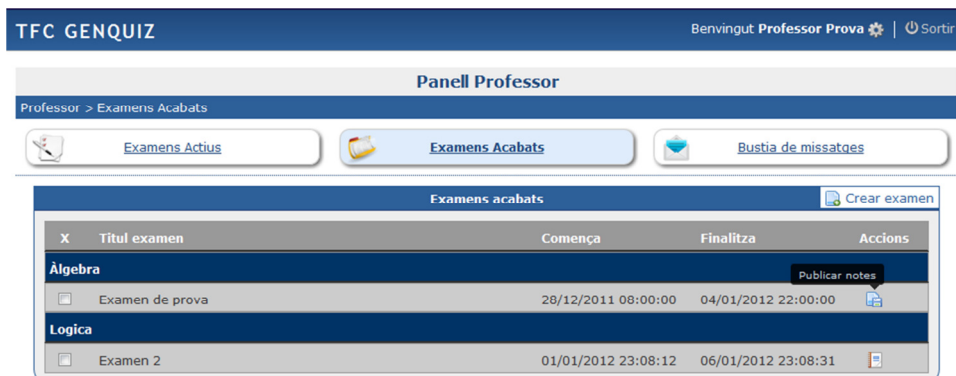
En aquest apartat es mostren algunes de les interfícies finals no esquematitzades en un principi.



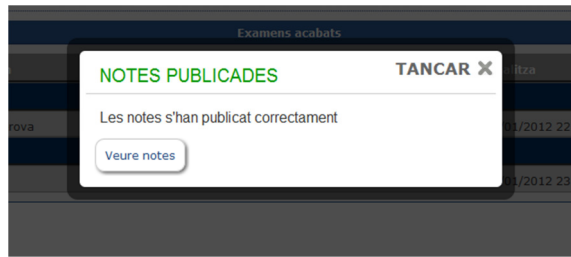
Interfície 17. Adminsitrar assignatures



Interfície 18. Exàmens acabats alumne



Interfície 19. Exàmens acabats professor



Interfície 20. Publicar notes modal

TFC GENQUIZ Benvingut Professor Prova ✨ | Sortir

Panel Professor

Professor > Examen Acabats > Resultats Examen 2

[Examen Actius](#) | [Examen Acabats](#) | [Bustia de missatges](#)

Resultats del examen

Títol: Examen 2
 Assignatura: Lògica
 Data inici: 01/01/2012 23:08:12
 Data límit: 06/01/2012 23:08:31

Llistat de notes de alumnes

NIF	Nom i cognom	Data realització	Correctes	Incorrectes	Nota	Accions
12312311Z	Joan Martí	10/01/2012 00:20:29	5	0	10	
29319231B	Arnau Rodriguez Diaz	10/01/2012 00:21:39	1	4	2	
47870438D	Jordi Rodriguez Diaz	09/01/2012 23:18:58	2	3	4	

Genquiz - TFC realitzat per Jordi Rodriguez - 2011/2012

Interfície 21. Resultat notes examen

TFC GENQUIZ Benvingut Professor Prova ✨ | Sortir

Panel Professor

Professor > Examen Acabats > Resultats Examen 2

[Examen Actius](#) | [Examen Acabats](#) | [Bustia de missatges](#)

Examen: Examen 2, Realitzat per: Jordi Rodriguez Diaz

Títol: Examen 2 Alumne: Jordi Rodriguez Diaz
 Assignatura: Lògica NIF: 47870438D
 Data inici: 01/01/2012 23:08:12 Respostes correctes: 2
 Data límit: 06/01/2012 23:08:31 Respostes incorrectes: 3
Nota final examen: 4

Respostes correctes / Respostes alumne:

1) Pregunta 1

Resposta 1 C

Resposta 2

Resposta 3

Resposta 4

2) Pregunta 2

Resposta 1

Resposta 2

Resposta 3

Resposta 4 C

3) Pregunta 3

Resposta 1

Resposta 2 C

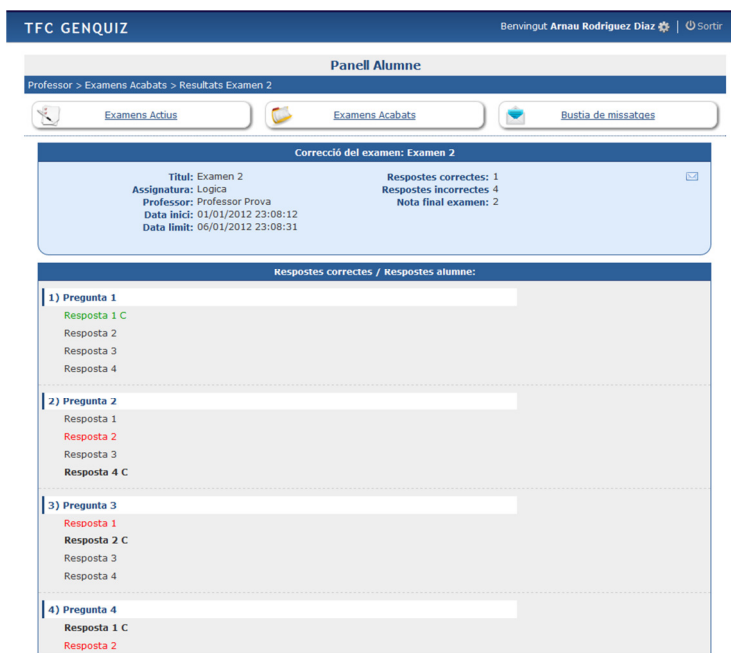
Resposta 3

Resposta 4

4) Pregunta 4

Resposta 1 C

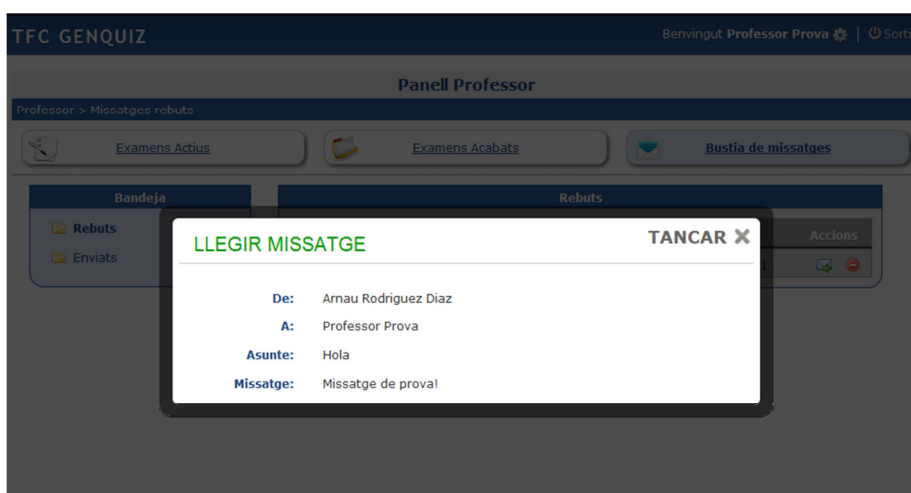
Interfície 22. Respostes correctes d'un alumne des de el panell de professor



Interfície 23. Resum de respostes correctes desde el panell d'alumne



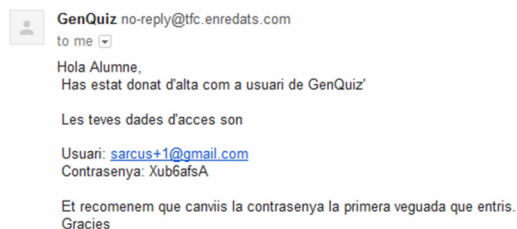
Interfície 24. Busties de missatges interns. Alumne enviats / Professor rebuts



Interfície 25. Modal llegir message

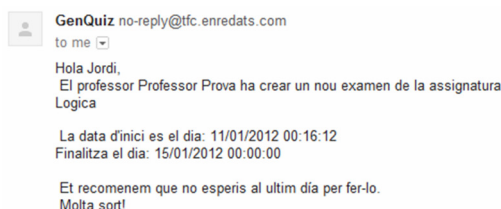
8.3. E-mails d'avís

Es mostren a continuació els dos tipus de e-mails d'avís que s'envien als usuaris.



Imatge 6. Mail d'avís de creació nou usuari amb contrasenya provisional

Quan el administrador crea un usuari per seguretat se li genera una contrasenya que només rebrà el usuari per e-mail. Se li recomana canviar-la per una més fàcil de recordar.



Imatge 7. Mail de creació d'examen

Quan un professor clica en finalitzar durant la creació d'un examen, s'envia un mail a tots els usuaris matriculats de la assignatura avisant del examen

9. Implementació

La implementació de l'aplicació s'ha realitzat seguint el fitxer de casos d'us inclòs en apartats anteriors, realitzant els canvis que s'han cregut convenients per portar el projecte a bon port.

Primer s'ha codificat la plana principal de login, tot seguit s'ha creat un usuari administrador a la base de dades i s'ha codificat el panell de administració d'assignatures, alumnes i professors en aquest ordre.

A partir d'aquí s'ha seguit amb la implementació del panell per a professors, codificant i dissenyant les planes per a llistar i crear exàmens.

Com era necessari que algú realitzes els exàmens s'ha codificat i dissenyat el apartat dels alumnes amb els llistats i la realització d'exàmens.

Amb això ja s'ha pogut seguir codificant les parts restants de usuaris i professors.

Per últim s'ha codificat el sistema de missatgeria que no estava del tot previst en un principi com a tal si no com un simple formulari que envies per mail.

Ja amb tota la programació pràcticament acabada, s'ha canviat el text de les accions per icones, s'ha maquetat i arreglat tot de forma vistosa, i s'ha afegit llibreries js per

millorar la usabilitat. Entre les que destaquen els tooltips dels icones, i els datepickers per seleccionar dates.

A continuació es detallaran les modificacions que ha sofert des del plantejament inicial, concretament dels casos d'us plantejats, les proves i tests realitzats, els models de dades i organització de fitxers, i alguns algorismes a destacar.

9.1. Canvis sobre els casos d'us previstos

- **Administrador**

La part d'administrador no hi ha hagut canvis a destacar segons la planificació dels casos d'us que es van detallar de bon principi.

La única cosa a comentar es que ara es realitza tot des de la mateixa plana per lo tant la edició d'alumnes, professors i assignatures es carreguen les dades en el formulari de creació.

També s'han introduït validadors. En el cas del DNI ha de ser correcte i no constar ja en la base de dades, i el email també se'n valida el format i que no existeixi ja.

Al crear un usuari es genera una contrasenya aleatòria i s'envia al mail del usuari.

- **Alumnes**

Al entrar en el panell ja no va directament al llistat de exàmens sinó que va a un dashboard on li apareixen els tres botons dels apartats principals. S'ha fet així pensant en una futura ampliació utilitzant aquesta plana per informar de novetats o el que sens acudeixi als alumnes.

Llistat d'exàmens: El llistat d'exàmens passa a dividir-se en llistat d'exàmens actius i llistat d'exàmens acabats.

Els exàmens actius son els que o encara no han començat o estan oberts a la realització. En les interfícies incloses anteriorment es pot apreciar.

Els acabats son els que ja ha acabat el seu termini de realització. Si el professor els ha donat per corregits apareixerà la nota.

Historial de notes: S'ha prescindit d'aquesta plana ja que el llistat d'exàmens acabats mostra totes les notes de tots els exàmens realitzats separats per assignatures.

Modificar informació personal: En un primer moment es volia que els usuari tinguessin un perfil amb foto i altres dades personals apart de les basiques. Al final s'ha limitat a només poder canviar la contrasenya del usuari.

Contactar: Es contacta a partir del llistat d'exàmens amb el professor que l'ha creat. En el apartat 'Comuns' es detallen les funcionalitats afegides al sistema de comunicació

- **Professors**

Crear examen: En aquest cas la funcionalitat es la mateixa però la manera en que es crea el examen varia una mica. A introduir les dades bàsiques del examen aquestes es guarden en la base de dades, i tot seguit es van creant i guardant una per una les preguntes amb les seves respostes.

Al finalitzar el examen s'envia un mail als alumnes matriculats de l'assignatura del examen aviant.

Llistat d'exàmens: S'ha dividir en llistat d'exàmens actius i acabats. Els actius es poden editar i esborrar si encara no ha passat la data d'inici. Els acabats es poden publicar les notes cosa que executa el script de comparació de respostes i guarda les notes.

Tancar i publicar notes: S'ha prescindit d'enviar el mail als usuaris ja que els apareixeran els exàmens amb la nota en exàmens acabats.

Full de notes i llistat d'alumnes: S'ha hagut de prescindir d'aquests dos mòduls per falta de temps així com també de les estadístiques.

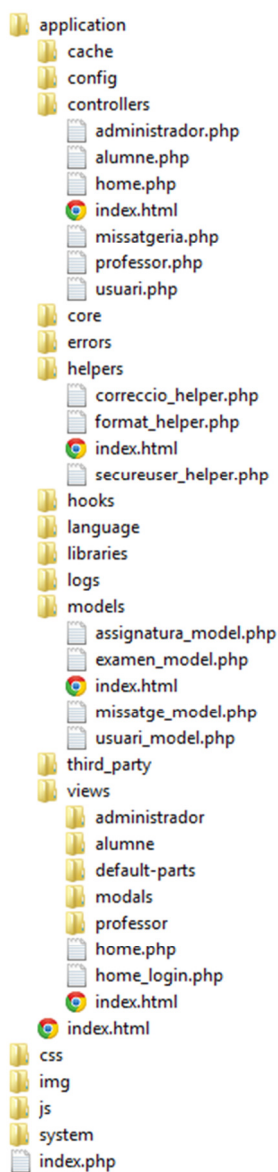
- **Comuns**

S'ha creat un nou cas d'us comú per als tres usuaris, que només s'havia plantejat en els alumnes com es la pagina de perfil. Aquesta permet modificar la contrasenya d'accés i en el cas dels alumnes i professors mostra també un llistat de les assignatures de les que esta matriculat o es docent.

S'ha creat una nova funcionalitat relacionada amb el enviament de missatges. De bon principi havia quedat en el aire quin seria el mètode de contacte entre professors i alumnes. El que s'ha fet es crear una bústia de missatge en el aplicatiu on es mostren en carpetes separades els missatges enviats i rebuts.

Per veure els missatges només s'ha de clicar, el missatge es marca com a llegit i es mostra un colorbox amb el contingut del missatge. Igualment des de la bandeja es pot respondre als missatges rebuts.

9.2. Estructura de directoris i fitxers



Codeigniter mostra una estructura de directoris i fitxers semblan a aquesta.

El MVC de codeigniter separa els controladors, les vistes i els models.

El primer que s'ha realitzat ha estat editar els arxius de la carpeta config, config.php i databases.php

En config.php s'especifiquen variables de configuració com pot ser la URL base. En el nostre cas em definit una local i una remota.

També s'ha afegit el següent per definir els tipus d'usuari, la longitud de salt del hash de seguretat d'un helper creat pel control de sessions i URL's pels diferents tipus d'arxius.

```

/* tipus usuari - url panell */
$config['tipus_usuari'] =
array('1'=>'administrador', '2'=>'professor',
'3'=>'alumne');

/* Salto seguridad session */
$config['length_salt'] = 50;

/* ATAJOS URL */
define('URL', $config['base_url']);
define('IMG', URL . 'img/');
define('JS', URL . 'js/');
define('CSS', URL . 'css/');
define('FONTS', URL . 'fonts/');

```

En el arxius database.php només s'han de configurar els paràmetres del servidor de bases de dades.

El autoload.php permet especificar quines llibreries i helpers es carregaran automàticament. En el nostre cas:

```

$autoload['libraries'] = array('session', 'database');
$autoload['helper'] = array('url', 'secureuser', 'format');

```

9.2.1. Controladors

Els controladors són els encarregats de executar tota la lògica de la aplicació. En Codeigniter venen declarats en forma de classe. Codeigniter sap quin controlador i mètode ha d'executar segons els paràmetres de la URL. En primer paràmetre representa el controlador en si, el segon el mètode del controlador a executar, i la resta són passats com a variables al mètode.

Un exemple:

Quan entrem a la URL: <http://tfc.enredats.com/administrador/alumnes/>

Estem cridant el mètode `alumnes` de la classe `administrador`. Cada mètode carrega si es necessari un model per consultar les bases de dades i envia tota la informació necessària a una vista en forma de array.

En el cas de no tenir paràmetres la URL, es carrega el mètode `index` de la classe per defecte (en el nostre cas [home.php](#)).

En un inici es va crear un controlador per a cada actor: [administrador.php](#), [usuari.php](#) i [professor.php](#).

Cada un incorpora la comprovació de sessió i tipus d'usuari en el constructor, cosa que fa que totes les pàgines de cada actor siguin segures.

Ja casi al final del projecte quan vaig plantejar el sistema de missatgeria, em vaig donar compte que la funcionalitat era idèntica tan per alumnes com professor, per tant es va crear un nou controlador [missatgeria.php](#)

El controlador [usuari.php](#) també es va crear per cobrir una funció comú en els tres actors que intervenen, ja que tots poden canviar la contrasenya de perfil.

9.2.2. Models

Els models són els encarregats de fer totes les consultes a la base de dades. Codeigniter igual que els controladors també els especifica com a classes amb mètodes. Cada mètode s'encarrega d'una consulta a la base de dades i retorna el resultat al controlador que l'ha cridat.

En el nostre cas s'ha creat un model per a cada gran objecte. assignatures, exàmens, missatges i usuaris. Cada model te les consultes necessàries de cada objecte.

Es podria haver pensat en crear model preguntes, respostes, però normalment van lligades a un examen i no s'ha vist necessari fins i tot contraproductiu.

9.2.3. Vistes

Les vistes són com plantilles en HTML on amb el mínim de codi php es printen les dades que es reben del controlador en el lloc pertinent i és lo que veu el usuari.

Si s'ha de canviar la col·locació de botons, marges, etc.. Només s'ha d'editar la vista que pertoca.

No entraré a detallar les vistes que s'han creat. S'han separat en carpetes per tipus d'usuari, els noms són bastant entenedors i no deixa de ser codi HTML fàcil de comprendre.

9.2.4. Helpers

Els helpers son arxius php que només contenen funcions, un cop carregats pots cridar les funcions des de qualsevol punt, fins i tot des de la vista. Si els inclou en el autoload no t'has de preocupar de res.

En el nostre cas s'han creat tres helpers.

[format_helper.php](#) només conté dos funcions per convertir el format de les dates d'americà a europeu.

[secureuser_helper.php](#) per controlar les sessions d'usuari. Esta format per una funció que crea la sessió, una que comprova si hi ha sessió i es valida, i una que la destrueix (per desbloquejar).

```
function create_login_session($iduser, $nomcomplet, $tipus)
{
    $CI =& get_instance();
    $length_salt = $CI->config->item('length_salt');
    $random = cadena_aleatoria(60);
    $salt = substr($random, 0, $length_salt);
    $ip_user = $_SERVER['REMOTE_ADDR'];
    $useragent = $_SERVER['HTTP_USER_AGENT'];
    $hash_user = sha1($salt . $ip_user . $useragent . $iduser . $tipus);
    $signature = $salt . $hash_user;

    $sessiondata = array('isLogged' => true, 'idusuari' => $iduser, 'nom' =>
    $nomcomplet, 'tipus' => $tipus, 'signature' => $signature);
    $CI->session->set_userdata($sessiondata);
}
```

Codi 1. Crea una variable de sessió signatura amb un hash de les dades d'usuari tot encriptat per poder comprovar que no s'ha produït suplantació de la sessió.

```
function is_logged()
{
    $CI =& get_instance();
    $length_salt = $CI->config->item('length_salt');
    if($CI->session->userdata('idusuari') && $CI->session->userdata('isLogged') ==
    TRUE){
        $iprecreate = $_SERVER['REMOTE_ADDR'];
        $useragentrecreate = $_SERVER['HTTP_USER_AGENT'];
        $signaturerecreate = $CI->session->userdata('signature');

        $saltrecreate = substr($signaturerecreate, 0, $length_salt);

        $originalhash = substr($signaturerecreate, $length_salt, 40);
        $hashrecreate = sha1($saltrecreate . $iprecreate . $useragentrecreate . $CI-
        >session->userdata('idusuari') . $CI->session->userdata('tipus'));
        if (!$hashrecreate == $originalhash) {
            $sessiondata = array('isLogged' => '', 'idusuari' => '', 'nom' => '',
            'tipus' => '', 'signature' => '');
            $CI->session->unset_userdata($sessiondata);
            return false;
        }else{
            return true;
        }
    }else{
        return false;
    }
}
```

Codi 2. Comproba si la signatura es correcte i si esta loguejat.

correccio helper.php conte dues funcions d'ajuda a la correcció dels examens. La primera adapta un array de respostes de manera que les claus siguin el id de pregunta i el valor el id de resposta.

```
$arr_return = array();  
foreach($arr_respostes as $r){  
    $arr_return[$r['idpregunta']] = $r['idresposta'];  
}
```

Codi 3. Adaptar array respores

La segona funció el rep 2 arrays, el primer amb les respostes correctes d'un examen i el segon amb les respostes que ha marcat un usuari. (els 2 adaptats per la funció anterior). Retorna un array amb la nota, el numero de correctes i el numero de errors.

```
function comparar_respostes($arr_correctes, $arr_alumne){  
    if($arr_alumne==null) return array('correctes'=>'0', 'errors'=>'0', 'nota'=>'0');  
    $correctes = 0;  
    $errors = 0;  
    $npreguntes = 0;  
    foreach($arr_correctes as $idpregunta => $idresposta){  
        if(isset($arr_alumne[$idpregunta]) && $arr_alumne[$idpregunta]==$idresposta){  
            $correctes++;  
        }else{  
            $errors++;  
        }  
        $npreguntes++;  
    }  
    $nota = $correctes*10/$npreguntes;  
    $nota = round($nota, 2);  
    return array('correctes'=>$correctes, 'errors'=>$errors, 'nota'=>$nota);  
}
```

Codi 4. Comparar respostes i retornar nota

9.2.5. Javascript

Al utilitzar AJAX per gran part del projecte, s'ha hagut d'escriure bastant codi Javascript. S'han creat quatre arxius de codi js tres pels diferents actors que es carreguen depenent de quin actor estigui logat, i un global que engloba funcionalitats comuns dels tres actors.

Les funcions del Javascript s'han separat en objectes i variables de funció. Similar a classes i mètodes.

El global te els objectes missatge i usuari que son comuns. També te una funció per mostrar un loading ('barra de carrega') que es mostra mentre s'executa qualsevol petició ajax amb el servidor.

Totes les peticions retornen una cadena en format json que indica si hi ha hagut error o no.

No entrarem a detallar cada Javascript perquè tots tenen una estructura similar i es poden entendre mirant el codi.

9.3. Problemes sorgits durant la codificació

Passem a detallar alguns dels mal de caps mes rellevants a la hora de codificar la aplicació:

9.3.1. Validar camps amb jQuery Validate

La validació de camps de text no presenta ninguna complicació, la complicació es va presentar al haver de crear regles noves.

La validació del NIF correcte em va costar una bastant que funciones correctament.

La validació dels radio butons de creació i edició de preguntes també em va portar uns quants mal de caps ja que s'havia de comprovar que cada pregunta tingues una resposta correcte marcada i en cas contrari treure un error.

Els selectors de jQuery es una cosa que no tenia gens per la ma i em va suposar hores de prova i error fins trobar la forma correcte.

Aquí es pot veure com es va aconseguir que comptes del formulari tots els inputs de tipus radio que estiguessin marcats.

```
return $("#frmCrearPregunta > input:radio:checked").length >= 1;
```

9.3.2. Col·locació d'errors de jQuery Validate

La col·locació dels errors en el lloc adequat mitjançant CSS també em va portar hores de feina. Als errors que es creen venen amb una classe error, el problema es quan en alguns llocs vols que surti sota del input, altres com a les preguntes ha de sortir un sol error per tots els radio buton, etc..

La majoria de vegades em trencava la col·locació dels camps de text i se'm movia tot de lloc. El CSS es una de les coses en general que m'ha costat més.

Es podria dir que la maquetació no es el meu fort.

9.3.3. Creació i inici de sessió de login

Per a la creació del salt del la signatura de sessió utilitzava principalment una funció que em generava una cadena aleatòria de longitud 50.

Mes endavant em va donar per substituir aquella funció que no m'agradava gaire per la següent, i aprofitar la mateixa per generar les contrasenyes d'usuaris.

```
function cadena_aleatoria($long = 50) {
    $str = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz1234567890";
    $cad = "";
    $str_len = strlen($str);
    for($i=0;$i<$long;$i++) {
        $cad .= substr($str,rand(0,$str_len-1),1);
    }
    return $cad;
}
```

Codi 5. Funció cadena aleatoria de longitud variable

El problema va estar en que en `substr($str,rand(0,$str_len-1),1);` em vaig deixar el -1.

Semblarà una tonteria però el que passava es que a vegades generava la cadena de longitud 50 i a vegades no, provocant que el usuari al atzar es pogués logar i altres no sense treure cap tipus d'error.

Vaig perdre pràcticament un mati intentant trobar el perquè de tot plegat. Printant totes les variables de sessió, els hash, salt, etc... Va ser bastant desesperant al detectar la tonteria d'error.

9.3.4. Que fer al eliminar un usuari, professor o assignatura?

Aquest tema em va portar hores de reflexió sense trobar una bona solució a tot plegat. Si el administrador elimina una assignatura que passa? Si la assignatura deixava d'existir no podien haver exàmens d'aquella assignatura en el sistema perquè no tindria sentit.

Al final després de donar moltes voltes vaig decidir que al eliminar una assignatura, professor o alumne, s'eliminés tot lo relacionat amb aquests.

Vaig pensar que el administrador ja sabia el que es feia i si decidia eliminar igual es que el alumne s'ha donat de baixa, o s'ha acabat el semestre i la assignatura ja no s'imparteix, etc..

Adjunto la consulta d'eliminació d'assignatura per mostrar tot el que s'ha d'eliminar de la base de dades. S'utilitzen transaccions per evitar errors que deixin les taules sense integritat.

```
function deleteAssignatura($idassignatura){
    $this->db->trans_start();
    $this->db->query('DELETE FROM cursant WHERE idassignatura="'.$idassignatura.'" ');
    $this->db->query('DELETE FROM alumne_examen WHERE idexamen IN (SELECT idexamen FROM examen WHERE idassignatura="'.$idassignatura.'" ');
    $this->db->query('DELETE FROM alumne_resposta WHERE idexamen IN (SELECT idexamen FROM examen WHERE idassignatura="'.$idassignatura.'" ');
    $this->db->query('DELETE FROM notes WHERE idexamen IN (SELECT idexamen FROM examen WHERE idassignatura="'.$idassignatura.'" ');
    $this->db->query('DELETE FROM pregunta WHERE idexamen IN (SELECT idexamen FROM examen WHERE idassignatura="'.$idassignatura.'" ');
    $this->db->query('DELETE FROM resposta WHERE idexamen IN (SELECT idexamen FROM examen WHERE idassignatura="'.$idassignatura.'" ');
    $this->db->query('DELETE FROM examen WHERE idassignatura="'.$idassignatura.'" ');
    $this->db->query('DELETE FROM assignatura WHERE idassignatura="'.$idassignatura.'" ');
    $this->db->trans_complete();
    return $this->db->trans_status();
}
```

Codi 6. Mètode eliminar Assignatura

9.3.5. Altres

A part de les descrites també em va costar bastanta feina treure la manera més òptima per fer la correcció dels exàmens. Al final el resultat es de lo més senzill. Es fa un select de les respostes correctes de un examen concret, es fa un select de les respostes d'un alumne en un examen, es formatgen els arrays i es comparen en un bucle les respostes oficials amb les del alumne.

El Javascript en general va ser un mal de cap constant, si alguna línia contenia errors era un show per trobar on estava i corregir-lo. Deixaven de funcionar totes les funcions i tan el Chrome com el Firefox atribuïen el error on no tocava.

9.4. Pla de proves

A continuació s'especifiquen algunes de les proves que s'han realitzat un cop finalitzada tota la codificació.

Comunes tres actors

Login		
Validació de correu	OK	
Validació de usuari + email base de dades	OK	
Denegació d'accés usuari incorrecte	OK	
Sortir (eliminar sessió)	OK	
Redirecció a url de panell segons el tipus d'usuari	OK	
Si la sessió a caducat redirigir a login	OK	
Evitar injecció de codi	OK	

Gestió de perfil

Validació de camps no vuits	OK	
Validació coincidència de claus	OK	
Administradors no llisten assignatures	OK	
Llistat d'assignatures alumnes	OK	
Llistat d'assignatures professors	OK	

Comuns alumnes i professors

Missatges		
Enviament de missatge	OK	
Lectura de missatge i marcat com a llegit un cop obert el missatge	OK	
Eliminació del missatge rebut	OK	No s'elimina de la base de

		dades només s'oculta de la bandeja ja que només es guarda un registre per missatge.
Validacions camps necessaris	OK	
Llistats de missatges rebuts i enviats	OK	
Enviament d'email d'avís	NO	No implementat pot ser molt molest
Avis en el menú de nous missatges	NO	No implementat per falta de temps

Administrador

Creació d'usuaris (Alumnes i Professors)

Validació de camps no vuits	OK	
Validació format e-mail	OK	
Validació format DNI	OK	Valida la lletra segons els números
No permetre repetició de e-mail ni NIF	OK	
Validació selecció mínima d'una assignatura	OK	
Guardat de dades correctament	OK	
Enviament d'email amb contrasenya generada aleatòriament	OK	

Creació d'assignatures

Validar nom no vuit	OK	
Guardat de dades	OK	

Modificació i esborrat d'alumnes i professors i assignatures

Carregar les dades en el formulari per AJAX	OK	
No permetre repetició de e-mail ni NIF en cas de usuaris.	NO	S'ha oblidat incloure en la edició aquesta comprovació
Validació selecció mínima d'una assignatura (usuaris)	OK	
Validacions de formats de camps	OK	
Recarregar el llistat després d'editar o eliminar	OK	

Professor

Creació d'examen, edició i esborrat

Validació dades bàsiques	OK	
Enregistrat dades bàsiques	OK	
Selector de dates i temps i validació de data	OK	
Validació de data inici no inferior a la actual	OK	
Validació de data final no inferior a inici	OK	
Afegir pregunta mostrar formulari	OK	

Editar dades bàsiques substituint text per formulari i posterior enregistrament	OK	
Validar camps de text de pregunta i respostes	OK	
Validació mínim una correcte seleccionada	OK	
Eliminació de pregunta de la base de dades i del DOM HTML	OK	
Pulsació en finalització i enviament d'e-mails als usuaris de l'assignatura	OK	Ok nomes en servidor remot en local dona error al no tenir servidor de correu
Comprovació al editar que es el professor creador sinó denegar	OK	
Comprovació al editar que no ha estat començat sinó denegar	OK	
No permetre eliminació d'exàmens començats	OK	
Enregistrat de preguntes respostes i correcte	OK	
No desquadrar al introduir respostes i preguntes molt llargues	OK	

Llistat exàmens actius i acabats

Comprovació al editar que no ha estat començat sinó denegar	OK	
No permetre eliminació d'exàmens començats	OK	
Llistat correcte i icones segons dates	OK	
Publicar notes no permetre sinó ha passat la data de fi	OK	
Publicar notes, enregistrar notes de tots els alumnes de la assignatura, hagin realitzat o no el examen	OK	
Eliminar examen	OK	
Veure llistat notes del examen i no permetre en cas que no s'hagin publicat les notes ni donat per tancat	OK	Es mostra un llistat dels alumnes amb les notes que han tret
Veure en detall examen realitzat per alumne i no permetre en cas que no s'hagin publicat les notes ni donat per tancat	OK	Es mostra les preguntes que ha respost marcant les correctes i incorrectes
Denegar access a totes les funcions en un examen que no has creat	OK	

Alumnes

Llistat exàmens actius i acabats

Llistats d'exàmens amb icones pertitnets i opcions habilitades sogons pertoqui	OK	
--	----	--

No permetre accedir als exàmens no iniciats encara (data_inici<actual)	OK	
Mostrar les notes en els exàmens acabats	OK	
No permetre accedir a exàmens ja caducats	OK	
No permetre accedir a revisar examen si no s'ha publicat notes i ha estat tancat	OK	
Marcar exàmens com a realitzat	OK	

Realització d'examen

Obligar al usuari a respondre totes les preguntes marcant una resposta	OK	Aquest punt pot ser discutible, es podria canviar treien simplement el validador.
Enregistrar totes les respostes	OK	
Denegar accés a la realització d'exàmens que son d'assignatures de les que no ha estat matriculat	OK	
Avisar al alumne de la correcte recepció de les respostes o el corresponent error (feedback)	OK	
Redirigir al llistat un cop enviades les respostes i mostrat el missatge.	OK	

Revisió d'examen realitzat

Sinó s'han publicat les notes denegar accés	OK	
Sinó li pertany per assignatures matriculat denegar accés	OK	
Mostrar respostes del alumne correctes en verd i incorrectes en vermell i correcte de professor en negreta	OK	

9.5. Resultats

Com s'ha pogut anar veient al llarg de la memòria, s'ha complert tots els objectius proposats de bon començament.

Podem observar totes les pantalles de funcionament de l'aplicació en el apartat Disseny, on s'expliquen les funcionalitats de cada pàgina.

També hem detallat les mínimes funcions que s'han eliminat i les que s'han afegit.

Es pot assegurar per lo tant que el programa compleix amb lo proposat i es 100% funcional.

Es pot comprovar el funcionament accedint per: <http://tfc.enredats.com>

E-mail administrador: admin@tfc.enedats.com

Contrasenya: 123123

10. Treball futur

S'enumeren algunes de les moltes millores i funcionalitats que es podrien afegir en futures ampliacions del aplicatiu:

- Permetre veure el historial de notes d'un alumne
- Estadístiques de resultats d'un examen en concret amb gràfics de barres o circulars
- Avisos en el menú de missatges nous sense llegir
- Possibilitat d'afegir un numero variable de respostes als exàmens
- Possibilitat de marcar varies respostes com a correctes per part dels professors i els alumnes
- Possibilitat d'adjuntar imatges o documents a les preguntes dels exàmens.
- Implementar un sistema de logs per emmagatzemar els errors produïts i així poder-los corregir.
- Mostrar informació d'interès en el dashboard dels diferents actors implicats en comptes de que estigui blanc. Es podrien mostrar si hi ha missatges nous, si hi ha exàmens nous obert en període de realització que encara no s'han completat, etc..
- Poder posar en contacte alumnes i professors amb l'administrador de l'aplicatiu.
- Refactoritzar el codi php eliminant codi duplicat en els diferents controladors.
- Separar els menús de les vistes en arxius separats per no haver de modificar totes les vistes en cas d'afegir un nou boto.
- Millorar el disseny i la maquetació ja que m'he centrat més en la programació i no li he pogut dedicar el suficientment temps.
- Possibilitat de confeccionar un perfil d'usuari mes complet adjuntant foto de perfil, aficions, etc. Com s'havia plantejat de bon començament.

- Guardar un historial de les dades eliminades al esborrar alumnes, professors i assignatures per una possible consulta futura.
- Sistema d'instal·lació automàtic

Així podríem seguir enumerant coses que es podrien millorar i afegir en futures versions que si el temps no hagués estat tan limitat possiblement s'haguessin tingut en compte i intentat portar a terme.

11. Conclusions

Ja per acabar, si repassem tot el desenvolupament del projecte i del TFC en si, es pot dir que s'ha complert amb tots els objectius proposats de bon començament. Tan a nivell de l'aplicatiu com a nivell personal.

Aquesta aplicació s'ha de prendre com a punt de partida de allò que podria arribar a ser si se li dediquessin moltes més hores i no solament un semestre.

No s'ha de comparar amb altres plataformes de e-learning més grans i conegudes com pot ser Moodle, ja que es possible que ens deprimim.

S'ha de pensar que les grans plataformes tenen més d'una persona darrera treballant, són experts en la matèria dels sistemes de docència online i no porten 3 o 4 mesos desenvolupament el seu software sinó que porten anys.

A nivell personal em sento molt orgullós del que he aconseguit realitzant aquest TFC, no tan sols parir de 0 una aplicació web 100% funcional i que compleix els objectius marcats sinó també per el que he arribat a aprendre en matèria de:

Bases de dades: disseny, relacions, joins múltiples, transaccions, etc..

PHP i Codeigniter: em sento que coneix-ho molt millor aquest framework que presenta infinitat de possibilitat i extensions.

Disseny i maquetació: Es una àrea que no toco i que mai he portat bé i no esperava treure'n tan bon resultat. *'Igual podré començar a prescindir del dissenyador ;)'*

Es podria dir que la part més costosa de tot el projecte a part de la codificació ha estat el Anàlisi de requisits, casos d'us, diagrames..

Tinc la costum de començar els projectes per la taulada, es a dir per la codificació deixant una mica de banda l'anàlisi previ. En aquest sentit m'ha ajudat a entendre millor aquesta part del desenvolupament de software, i a tenir-la present a partir d'ara i aplicar-la sempre en projectes futurs simplificant així tasques futures.

En conclusió la valoració final es positiva, i tot i que al començament el vaig agafar una mica amb desgana, poc a poc al veure lo que anava sortint i naixent alguna cosa i que funcionava, m'enganxava més i més fins el punt de tirar-se'm el temps a sobre.

En resum m'hagués agradat disposar de més temps, però em sento orgullós de lo aconseguit.

12. Bibliografia

A continuació s'enumeren alguns dels recursos i fonts d'informació consultats per a la realització d'aquest tfc

Wikipedia

Pagina d'enciclopedia lliure

<http://es.wikipedia.org/wiki/Wikipedia:Portada>

XAMPP

Servidor LAMP per windows

<http://sourceforge.net/projects/xampp/>

PHP

Documentació oficial de PHP

<http://php.net/>

Mysql

Documentació oficial de Mysql

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

jQuery

Pagina oficial amb molta documentació d'aquesta biblioteca js

<http://jquery.com/>

jQuery Ui 1.8.16

Documentació de jQueryUi

<http://jqueryui.com/>

Scripts jQuery

Query Timepicker Addon - <http://trentrichardson.com/examples/timepicker/>

jQuery Colorbox - <http://jacklmoore.com/colorbox/>

jQuery Form Plugin - <http://jquery.malsup.com/form/>

jQuery Validate - <http://bassistance.de/jquery-plugins/jquery-plugin-validation/>

jQuery Topsy - <http://onehackoranother.com/projects/jquery/topsy/>

Pàgines d'ajuda a programadors

Stackoverflow - <http://stackoverflow.com/>

ForsdelWeb - <http://www.forsdelweb.com/f13/funcion-validar-dni-completo-391990/>

Material de la carrera

Annex 1. Instal·lació de l'aplicatiu

Per instal·lar l'aplicació en un servidor propi cal que el servidor tingui suport per a PHP i MYSQL.

El PHP ha de tenir activats els mòduls rewrite per les url friendly.

Ha de tenir un servidor instal·lat per fer us de l'enviament de e-mails.

Per a instal·lar s'ha de seguir les següents pautes per ordre:

1. Descomprimir el ZIP en una carpeta
2. Editar el arxiu /application/config/config.php i posar la url base segons si es vol treballar en producció en test o en local

```
case 'production':
    $config['base_url'] = 'http://tfc.enredats.com/';
break;
case 'testing':
    $config['base_url'] = 'http://tfc.enredats.com/';
break;
case 'development':
    $config['base_url'] = 'http://tfc.localhost/';
break;
```

3. Editar el index.php i definir l'entorn de desenvolupament segons el punt anterior

```
define('ENVIRONMENT', 'production');
```

4. Crear la base de dades i carregar executar el arxiu bases_de_dades.sql que esta dins el ZIP o en el Annex, per crear les bases de dades basiques.

5. Editar el arxiu /application/config/database.php i col·locar les dades de configuració de la base de dades creada.

```
$db['default']['hostname'] = 'localhost';
$db['default']['username'] = 'tfc';
$db['default']['password'] = 'yWfTJchsjnMfvKR ';
$db['default']['database'] = 'tfc';
```

6. Entrar a la url base on apareixerà la pantalla de login.
7. Introduir les dades per defecte del usuari administrador
E-mail: admin@tfc.enredats.com
Contrasenya: 123123

Ja es pot començar a crear usuaris, assignatures!

Annex 2. Script de creació de bases de dades

```

/*
BASES DE DADES DEL TFC
JORDI RODRIGUEZ DIAZ
APLICACIONS WEB PER A TREBALL COLABORATIU
GENQUIZ
*/

SET FOREIGN_KEY_CHECKS=0;

-----
-- Table structure for `alumne_examen`
-----
DROP TABLE IF EXISTS `alumne_examen`;
CREATE TABLE `alumne_examen` (
  `idusuari` int(10) NOT NULL,
  `idexamen` int(10) NOT NULL,
  `data` datetime NOT NULL,
  PRIMARY KEY (`idusuari`,`idexamen`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-----
-- Records of alumne_examen
-----

-----
-- Table structure for `alumne_resposta`
-----
DROP TABLE IF EXISTS `alumne_resposta`;
CREATE TABLE `alumne_resposta` (
  `idusuari` int(10) NOT NULL,
  `idexamen` int(10) NOT NULL,
  `idpregunta` int(10) NOT NULL,
  `idresposta` int(10) NOT NULL,
  PRIMARY KEY (`idusuari`,`idexamen`,`idpregunta`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-----
-- Records of alumne_resposta
-----

-----
-- Table structure for `assignatura`
-----
DROP TABLE IF EXISTS `assignatura`;
CREATE TABLE `assignatura` (
  `idassignatura` int(1) NOT NULL AUTO_INCREMENT,
  `nom` varchar(150) NOT NULL,
  PRIMARY KEY (`idassignatura`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-----
-- Records of assignatura
-----

-----
-- Table structure for `cursant`
-----
DROP TABLE IF EXISTS `cursant`;
CREATE TABLE `cursant` (
  `idusuari` int(10) NOT NULL,
  `idassignatura` int(10) NOT NULL,
  PRIMARY KEY (`idusuari`,`idassignatura`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-----
-- Records of cursant
-----

-----
-- Table structure for `examen`
-----
DROP TABLE IF EXISTS `examen`;
CREATE TABLE `examen` (
  `idexamen` int(10) NOT NULL AUTO_INCREMENT,
  `idassignatura` int(10) NOT NULL,
  `idprofessor` int(10) NOT NULL,
  `titul` text NOT NULL,
  `data_inici` datetime NOT NULL,
  `data_fi` datetime NOT NULL,

```

```

`tancat` enum('1','0') NOT NULL DEFAULT '0',
PRIMARY KEY (`idexamen`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-----
-- Records of examen
-----

-----
-- Table structure for `missatge`
-----
DROP TABLE IF EXISTS `missatge`;
CREATE TABLE `missatge` (
  `idmissatge` int(10) NOT NULL AUTO_INCREMENT,
  `idremitent` int(10) NOT NULL,
  `iddestinatari` int(10) NOT NULL,
  `asunte` varchar(100) NOT NULL,
  `missatge` text NOT NULL,
  `data_enviat` datetime NOT NULL,
  `llegit` enum('0','1') NOT NULL DEFAULT '0',
  `hide_sended` enum('0','1') NOT NULL DEFAULT '0',
  `hide_recived` enum('0','1') NOT NULL DEFAULT '0',
  PRIMARY KEY (`idmissatge`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-----
-- Records of missatge
-----

-----
-- Table structure for `notes`
-----
DROP TABLE IF EXISTS `notes`;
CREATE TABLE `notes` (
  `idusuari` int(10) NOT NULL,
  `idexamen` int(10) NOT NULL,
  `nota` int(2) NOT NULL,
  `correctes` int(3) NOT NULL,
  `errors` int(3) NOT NULL,
  PRIMARY KEY (`idusuari`,`idexamen`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-----
-- Records of notes
-----

-----
-- Table structure for `pregunta`
-----
DROP TABLE IF EXISTS `pregunta`;
CREATE TABLE `pregunta` (
  `idpregunta` int(10) NOT NULL AUTO_INCREMENT,
  `idexamen` int(10) NOT NULL,
  `pregunta` text NOT NULL,
  PRIMARY KEY (`idpregunta`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-----
-- Records of pregunta
-----

-----
-- Table structure for `resposta`
-----
DROP TABLE IF EXISTS `resposta`;
CREATE TABLE `resposta` (
  `idresposta` int(10) NOT NULL AUTO_INCREMENT,
  `idpregunta` int(10) NOT NULL,
  `idexamen` int(10) NOT NULL,
  `resposta` text NOT NULL,
  `correcte` enum('0','1') NOT NULL DEFAULT '0',
  PRIMARY KEY (`idresposta`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-----
-- Records of resposta
-----

-----
-- Table structure for `usuari`
-----
DROP TABLE IF EXISTS `usuari`;
CREATE TABLE `usuari` (
  `idusuari` int(10) NOT NULL AUTO_INCREMENT,
  `clau` char(40) NOT NULL,

```

```
`nif` char(9) NOT NULL,  
`nom` varchar(45) NOT NULL,  
`cognoms` varchar(150) DEFAULT NULL,  
`mail` varchar(150) NOT NULL,  
`tipus` enum('1','2','3') NOT NULL,  
PRIMARY KEY (`idusuari`)  
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;  
  
-----  
-- Records of usuari  
-----  
INSERT INTO `usuari` VALUES ('1', '601f1889667efaebb33b8c12572835da3f027f78', '12345678X',  
'Administrador', 'Web', 'admin@tfc.enredats.com', '1');
```