

Desenvolupament d'una aplicació web per a donar suport a les tasques d'avaluació dels docents

Memòria de Projecte Final de Màster

Màster Universitari en Aplicacions Multimèdia

Àrea d'especialitat/Itinerari professional

Autor: Llorenç Seguí Capllonch

Consultor: Sergio Schvarstein Liuboschetz

Professora: Laura Porta Simó

11 de juny de 2018

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Desenvolupament d'una aplicació web per a donar suport a les tasques d'avaluació dels docents</i>
Nom de l'autor:	<i>Llorenç Seguí Capllonch</i>
Nom del consultor/a:	<i>Sergio Schvarstein Liuboschetz</i>
Nom del PRA:	<i>Laura Porta Simó</i>
Data de lliurament (mm/aaaa):	<i>06/2018</i>
Titulació o programa:	<i>Màster Universitari en Aplicacions Multimèdia</i>
Àrea del Treball Final:	<i>Treball Final de Màster Professionalitzador</i>
Idioma del treball:	<i>Català</i>
Paraules clau	<i>Client web, API REST, docents</i>
Resum del Treball:	
<p><i>El propòsit d'aquest TFM és dissenyar i desenvolupar una aplicació web (adaptable a les resolucions de diferents dispositius) que permeti als docents enregistrar la informació acadèmica dels alumnes que pertanyin a les assignatures que aquests imparteixen.</i></p> <p><i>Des de un punt de vista funcional, l'aplicació web permetrà als docents poder enregistrar cada una de les activitats i/o treballs de totes aquelles assignatures que imparteix en els diferents centres d'educació secundària en el que estigui contractat. A l'hora, per a cada una de les activitats d'una determinada assignatura el docent podrà enregistrar les qualificacions obtingudes pels alumnes. D'aquesta manera, el càlcul de les qualificacions finals de cada un dels alumnes (a una determinada assignatura) serà dut a terme automàticament per l'aplicació.</i></p> <p><i>Aquesta aplicació també serà una eina de suport cap als docents ja que els hi permetrà organitzar els alumnes i assignatures en els corresponents cursos d'ESO i Batxillerat.</i></p> <p><i>El que es vol aconseguir és que els docents puguin enregistrar tota la informació acadèmica dels seus alumnes i accedir-hi fàcilment a través de qualsevol dispositiu (ordinador, mòbil i tauleta), sense necessitat d'haver de portar tota aquesta informació apuntada a les clàssiques llibretes.</i></p> <p><i>Per tant, el producte a obtenir constarà de una API REST que implementi tota la capa d'accés a les dades i tota la lògica aplicada sobre les mateixes i una aplicació web (client) responsive que s'encarregui de consumir i presentar tota la informació obtinguda per l'API REST.</i></p>	

Abstract

The aim of this project is the design and the development of a responsive-base web application which allows to teachers recording the academic information such as, students' data from courses.

Regarding the functionalities, the proposed web is able to record each activity from teachers courses in each higher school centre. Besides, teachers can record the students' qualification from each course activity. As a consequence, the final marks will be automatically carried out by the application.

Moreover, the proposed application is a support tool for teachers since it allows them to organize their students and subjects. With this, the teachers can record all the academic information of their students and access them easily through any device.

Dedicatòria/Cita

La satisfacció rau en l'esforç, no en l'assoliment. L'esforç total és una victòria completa.

Agraïments

Primerament vull agrair als meus pares Cristofol Seguí i Magdalena Capllonch l'ajut i el recolzament que m'han donat des de que tinc ús de raó. Sou i sereu els meus eterns consellers.

A tots i cada un dels meus amics i amigues, els quals han estat un gran suport en el transcurs del desenvolupament del projecte i són un pilar important de la meua vida.

També agrair a Sergio Schvarstein, per la seva disponibilitat durant tot el projecte, pels seus suggeriments i per confiar amb aquest projecte.

I finalment, a la meua àvia Joana Aina. Potser hagi desaparegut de la meua vida, però no del meu cor.

Abstract

The aim of this project is the design and the development of a responsive-base web application which allows to teachers recording the academic information such as, students' data from courses.

Regarding the functionalities, the proposed web is able to record each activity from teachers courses in each higher school centre. Besides, teachers can record the students' qualification from each course activity. As a consequence, the final marks will be automatically carried out by the application.

Moreover, the proposed application is a support tool for teachers since it allows them to organize their students and subjects. With this, the teachers can record all the academic information of their students and access them easily through any device.

Resum

El propòsit d'aquest **TFM** és dissenyar i desenvolupar una aplicació web (adaptable a les resolucions de diferents dispositius) que permeti als docents enregistrar la informació acadèmica dels alumnes que pertanyin a les assignatures que aquests imparteixen.

Des de un punt de vista funcional, l'aplicació web permetrà als docents poder enregistrar cada una de les activitats i/o treballs de totes aquelles assignatures que imparteix en els diferents centres d'educació secundària en el que estigui contractat. A l'hora, per a cada una de les activitats d'una determinada assignatura el docent podrà enregistrar les qualificacions obtingudes pels alumnes. D'aquesta manera, el càlcul de les qualificacions finals de cada un dels alumnes (a una determinada assignatura) serà dut a terme automàticament per l'aplicació.

Aquesta aplicació també serà una eina de suport cap als docents ja que els hi permetrà organitzar els alumnes i assignatures en els corresponents cursos d'ESO i Batxillerat.

El que es vol aconseguir és que els docents puguin enregistrar tota la informació acadèmica dels seus alumnes i accedir-hi fàcilment a través de qualsevol dispositiu (ordinador, mòbil y tauleta), sense necessitat d'haver de portar tota aquesta informació apuntada a les clàssiques llibretes.

Per tant, el producte a obtenir constarà de una **API REST** [1] que implementi tota la capa d'accés a les dades i tota la lògica aplicada sobre les mateixes i una aplicació web (client) *responsive* [2] que s'encarregui de consumir i presentar tota la informació obtinguda per l'**API REST**.

Paraules clau

Client web, *responsive*, **API REST**, educació secundària, docents, alumnes, qualificacions.

Índex

Capítol 1: Introducció	13
1.Introducció/Prefaci	13
2. Descripció/Definició	14
3. Objectius generals	15
3.1 Objectius principals	15
3.2 Objectius personals	15
4. Metodologia i procés de treball	16
5. Planificació	18
5.1 Planificació de les tasques	18
5.2 Planificació temporal	19
6. Pressupost.....	20
7. Estructura de la resta del document	21
Capítol 2: Anàlisi	22
1. Estat de l'art.....	22
2. Públic objectiu i perfils d'usuari	27
3. Especificació de requisits	28
3.1 Requisits funcionals	28
3.2 Requisits no funcionals	29
4. Requisits tecnològics	29
Capítol 3: Disseny	31
1. Arquitectura general del sistema	31
2. Arquitectura de la informació i diagrames de navegació	33
2.1 Model de dades.....	33
2.2 Arbre de navegació	33
3. Tecnologies, guies i eines emprades	35
5. Tecnologies implicades a cada mòdul de l'arquitectura.....	40
4. Disseny gràfic i interfícies.....	41
4.1 Estils	41
4.2 Usabilitat/UX.....	43
Capítol 4: Implementació	45

1. Implementació de l'API REST	45
2. Implementació de l'aplicació web	48
3. Desplegament de l'aplicació a Heroku.....	50
4. Control de versions i repositori de software.....	50
Capítol 5: Demostració.....	51
1. Funcionament de l'API REST.....	51
1.1 Autenticació a l'API REST	51
1.2 Peticions GET i DELETE	53
1.3 Peticions POST i PUT	54
2. Funcionament de l'aplicació web	59
Capítol 6: Conclusions i línies de futur	65
1. Conclusions	65
2. Línies de futur	66
2.1 Noves funcionalitats	66
2.2 Aprofitament de la capa d'accés a les dades	66
Bibliografia	67
Annexos.....	69
Annex A: Glossari.....	69
Annex B: Elements lliurats.....	69

Figures i taules

Índex de figures

Figura 1: Diagrama de Gantt	19
Figura 2: Pantalla principal de GoConqr (versió web).....	23
Figura 3: Pantalles Attendance	24
Figura 4: Pantalla Edmodo	26
Figura 5: Arquitectura de l'aplicació	32
Figura 6: Model da dades de la base de dades	33
Figura 7: Arbre de navegació de l'aplicació web	34
Figura 8: Exemple d'un JSON.	38
Figura 9: Components <i>Material Design</i> emprats en l'aplicació web	41
Figura 10: Paleta de colors configurada per a l'aplicació web	42
Figura 11: Font Roboto	42
Figura 12: Estructura interfície (Resolució per ordinadors)	43
Figura 13: Estructura interfície (Resolució per <i>smartphone</i>)	44
Figura 14: Estructura interfície (Resolució per tauletes)	44
Figura 15: Model de Django de l'entitat Centre	45
Figura 16: Serializador de Django del model Centre	45
Figura 17: Vista corresponent a les peticions GET (listar) i POST dels Centres	46
Figura 18: Vista corresponent a les peticions GET (obtenir), PUT i DELETE dels Centres	47
Figura 19: Exemple d'un fitxer d'URLs de Django REST Framework	47
Figura 20: Estructura del projecte (ReactJS)	48
Figura 21: Pàgina d'autenticació	59
Figura 22: Pàgina de registre	59
Figura 23: Pàgina de gestió de centres acadèmics	60
Figura 24: Pàgina de gestió d'anys acadèmics i grups de cursos d'un centre	60
Figura 25: Pàgina que llista totes les assignatures impartides.....	61
Figura 26: Pàgina de gestió de les activitats.....	62
Figura 27: Qualificacions d'una activitat obtingudes pels alumnes.	62
Figura 28: Pàgina de gestió dels alumnes.....	63
Figura 29: Qualificacions de cada una de les activitats d'un alumne i la seua mitjana ponderada.	64
Figura 30: Formulari per modificar la informació personal de l'usuari.	64

Índex de taules

Taula 1: Planificació de les tasques	18
Taula 2: Pressupost del projecte	20
Taula 3: Tecnologies implicades a cada mòdul.	40

Capítol 1: Introducció

1.Introducció/Prefaci

El context a on emmarcar el projecte està definit per l'eclosió de dos fenòmens, un social i l'altre tecnològic. Durant els darrers anys, els dispositius tecnològics han evolucionat de forma exponencial. La interacció entre els usuaris i els dispositius s'ha millorat de tal forma que ha repercutit en el nivell i qualitat de vida de les persones. Així doncs, la societat actual disposa d'un enorme mercat d'aplicacions que de cada dia va creixent i que, a l'hora, és aplicable a la majoria dels àmbits.

A l'àmbit de l'educació, són molts els docents de centres d'educació secundària que encara es dediquen a registrar, en llibretes, les qualificacions dels seus alumnes i tota aquella informació necessària per comptabilitzar els seu progrés acadèmics al llarg del curs. La principal problemàtica és el fet de portar a sobre la llibreta i estar constantment apuntant informació.

D'acord amb tot el que s'ha exposat anteriorment, cal remarcar que l'ús de les aplicacions i noves tecnologies per al dia a dia dels docents suposaria una gran avantatge. En primer lloc, per a poder enregistrar tota la informació acadèmica dels alumnes i accedir-hi fàcilment a través d'un únic dispositiu. En segon lloc, per tal de poder automatitzar algunes de les tasques d'avaluació aplicades als alumnes.

És per això que el propòsit és donar resposta a la necessitat de desenvolupar una aplicació web per als docents de centres d'educació secundària que els hi doni suport en les seues tasques d'avaluació. Un dels requisits més important d'aquesta aplicació és que es pugui accedir des de qualsevol dispositiu (ordinador, tauleta o telèfon intel·ligent). D'aquesta manera, l'experiència d'usuari millora [3] quan s'optimitza l'aplicació segons el dispositiu a on es visualitza.

2. Descripció/Definició

A l'àmbit de l'educació i ensenyança, són molts els docents que encara porten a sobre les clàssiques llibretes amb tota la informació relacionada amb les qualificacions dels seus alumnes. A la vegada, la societat actual es troba en un moment a on la tecnologia i els nous dispositius estan agilitzant, optimitzant i perfeccionant algunes de les activitats que les persones realitzen en el seu dia a dia. Particularment, són poques les persones que no portin a sobre ordinadors, tauletes, telèfons intel·ligents o qualsevol altre dispositiu.

Tenint en compte la mencionada revolució tecnològica i la gran quantitat de docents que encara anoten, en llibretes, les qualificacions dels seus alumnes, estaria bé que els docents poguessin accedir i gestionar la informació acadèmica dels alumnes a través dels dispositius que porten a sobre. És per això que la proposta d'aquest **TFM** és desenvolupar una aplicació d'aquestes característiques.

El producte resultant d'aquest projecte és una aplicació web. El fet d'escollir una aplicació de tipus web i no una aplicació nativa (escriptori o mòbil) ve justificat pel fet de poder facilitar a l'usuari el poder accedir-hi des de internet i a través de un gran nombre de dispositius. Per tal de que sigui executable en diferents dispositius, l'aplicació web s'ha dissenyat seguint una filosofia adaptativa o *responsive*. Aquesta filosofia de disseny permet que les interfícies que conformen l'aplicació web s'adaptin a les resolucions dels dispositius a on s'executa.

3. Objectius generals

3.1 Objectius principals

El objectius principals del projecte que s'han derivat són:

- Dissenyar i desenvolupar una aplicació que doni suport als docents de centres d'educació secundària en les seves tasques d'avaluació.
- Permetre gestionar, als docents, la creació de centres a on imparteixen classes, les assignatures que imparteixen, els alumnes matriculats, les activitats avaluades a cada assignatura i les qualificacions obtingudes pels alumnes a cada una de les activitats.
- Desenvolupar l'aplicació web seguint un patró arquitectònic de client-servidor.
- Desenvolupar una **API REST** que implementi la capa d'accés a les dades i tota la lògica aplicada sobre les mateixes.
- Desenvolupar un client web, basat en components web, que s'encarregui de consumir i tota la informació obtinguda de la capa d'accés a les dades.
- Oferir una interfície que sigui adaptable a les resolucions dels diferents dispositius a on s'executi l'aplicació.

3.2 Objectius personals

Els objectius personals que satisfà el desenvolupament del projecte són:

- Contribuir en el desenvolupament d'una eina que faciliti les tasques d'avaluació dels docents de centres d'educació secundària.
- Assumir el repte d'aprendre i experimentar amb tecnologies i patrons d'arquitectura recents en el mercat del desenvolupament. En altres paraules, donar un pes important a l'experimentació i aprenentatge.
- Aplicar els coneixements adquirits durant el transcurs del màster per tal de desenvolupar una aplicació.

4. Metodologia i procés de treball

El projecte està enfocat en el desenvolupament d'una aplicació web que dona suport als docents de centres d'educació secundària en la tasca d'avaluació dels alumnes. Per a desenvolupar-lo, s'ha seguit un cicle de vida basat en un model iteratiu e incremental [4]. Aquest model es basa en la filosofia de construir incrementat les funcionalitats del sistema. Totes i cada una de les iteracions s'han dut a terme en les següents fases:

Anàlisis: Aquesta etapa ha consistit en definir els objectius i requeriments del projecte respecte a la orientació del producte, públic objectiu i les característiques a nivell funcional i no funcional de l'aplicació.

Primerament es va dur a terme un estudi del mercat d'aplicacions d'àmbit educatiu per tal de poder justificar la viabilitat de la proposta. Seguidament, es van delimitar les temàtiques i funcionalitats que l'aplicació havia de contemplar a partir de les tasques d'avaluació que duen a terme els docents als seus corresponents alumnes. Finalment, per tal d'assolir un dels objectius personals del projecte, es va dur a terme un estudi de camp, experimentació i formació amb diferents tecnologies, patrons de disseny i patrons arquitectònics punters en el marc actual del disseny i desenvolupament d'aplicacions web.

Planificació: En aquesta etapa s'han definit i classificat les tasques i fites les quals han servit com a referència per analitzar els avanços del projecte. Al·ludint a l'organització del treball a desenvolupar s'han tingut en compte les dates d'entrega establertes pel calendari acadèmic de la **UOC**. En aquest sentit, s'han aplicat tres grans fases per el desenvolupament de l'aplicació. La primera fase ha consistit en dissenyar l'arquitectura del sistema, l'arquitectura de la informació i els models de dades. La segona fase ha consistit en la implementació de les funcionalitats que conformen la capa de negoci (*Back-end*). Finalment, la darrera fase s'ha orientat en el disseny visual (colors, fonts estils, patrons ...) i funcional de l'aplicació web (*Front-end*).

Disseny del sistema: En aquesta etapa s'han portat a terme les tasques relacionades amb identificar els patrons i abstraccions coherents que proporcionen un marc definit i clar per poder interactuar amb el codi font de l'aplicació. En primer lloc, s'han definit els components o mòduls que compondran l'aplicació i com es comunicaran entre ells. En segon lloc, s'ha dissenyat el model de dades que defineix com s'emmagatzemarà la informació tractada per l'aplicació. Finalment, s'ha dissenyat i especificat l'arbre de navegació de l'aplicació web que permet tenir una representació gràfica de l'estructura de navegació d'un lloc web.

Desenvolupament de les funcionalitats de la capa de negoci: En aquesta etapa s'han portat a terme les tasques relacionades amb el desenvolupament de la capa de negoci en forma d'**API REST**. Concretament, s'han desenvolupat totes aquelles funcionalitats que accedeixen a les dades i apliquen lògica sobre les mateixes per tal de que siguin presentades a l'aplicació web.

Disseny i desenvolupament l'aplicació web: En aquesta etapa s'han portat a terme les tasques relacionades amb la definició i disseny de l'aspecte visual de les interfícies que compondran l'aplicació web seguint els patrons de disseny i guies d'estils identificades durant l'etapa d'anàlisi. En segon lloc, s'han portat a terme les tasques relacionades amb el desenvolupament de les funcionalitats que conformaran l'aplicació web. Aquestes funcionalitats s'han programat amb tecnologies web punteres dins el marc del desenvolupament d'aplicacions web per tal d'oferir als usuaris una experiència interactiva amb l'aplicació.

En les dues darreres etapes es van tenir en compte la metodologia àgil per realitzar els avenços i lliurament de cada funcionalitat, manejant flexibilitat en els temps de lliurament respecte a la complexitat de cada funcionalitat desenvolupada.

5. Planificació

El projecte comença el 26 de febrer de 2018 i finalitza l'11 de juny de 2018, dates d'inici i finalització de les activitats acadèmiques conforme a la programació del semestre.

5.1 Planificació de les tasques

La Taula 1 representa les tasques planificades i classificades en 5 fites.

TASCA	DATA D'INICI	DATA DE FI
PAC 1		
Proposta del projecte	26/02/2018	12/03/2018
PAC 2		
Mandat del projecte i planificació	13/03/2018	25/03/2018
PAC 3		
Investigació i proves de les tecnologies	22/03/2018	25/03/2018
Disseny del model de dades relacional	26/03/2018	26/03/2018
Desenvolupament del Create, Read, Update and Delete (CRUD) per l'API (Autenticació i registre d'usuaris, gestió de centres, gestió de cursos i anys acadèmics i creació d'assignatures)	27/03/2018	02/04/2018
Disseny i desenvolupament de les pàgines i components web (Autenticació i registre d'usuaris, gestió de centres, gestió de cursos i anys acadèmics i creació d'assignatures)	03/04/2018	19/04/2018
Desenvolupament dels servies i lògiques per accedir a la capa de dades (Autenticació i registre d'usuaris, gestió de centres, gestió de cursos i anys acadèmics i creació d'assignatures)	03/04/2018	19/04/2018
Configurar servidor	20/04/2018	20/04/2018
Desplegar l'aplicació	21/04/2018	21/04/2018
Documentació	22/04/2018	23/04/2018
PAC 4		
Investigació i proves de les tecnologies	24/04/2018	26/04/2018
Desenvolupament del CRUD per l'API (Gestió d'assignatures, gestió de grups acadèmics, gestió d'alumnes, gestió d'activitats i avaluacions)	27/04/2018	02/05/2018
Disseny i desenvolupament de les pàgines i components web (Gestió d'assignatures, gestió de grups acadèmics, gestió d'alumnes, gestió d'activitats i avaluacions)	03/05/2018	19/05/2018
Desenvolupament dels servies i lògiques per accedir a la capa de dades (Gestió d'assignatures, gestió de grups acadèmics, gestió d'alumnes, gestió d'activitats i avaluacions)	03/05/2018	19/05/2018
Desplegar aplicació	20/05/2018	20/05/2018
Documentació	21/05/2018	21/05/2018
PAC 5		
Correcció d'errades	22/05/2018	31/05/2018
Memòria	01/06/2018	09/06/2018
Presentació	10/06/2018	11/06/2018

Taula 1: Planificació de les tasques

5.2 Planificació temporal

Una vegada establertes les dates d'inici i fi de les tasques, s'ha desenvolupat el següent diagrama de Gantt (Figura 1) per tal de poder mostrar la durada de les tasques i les seues possibles dependències entre elles.

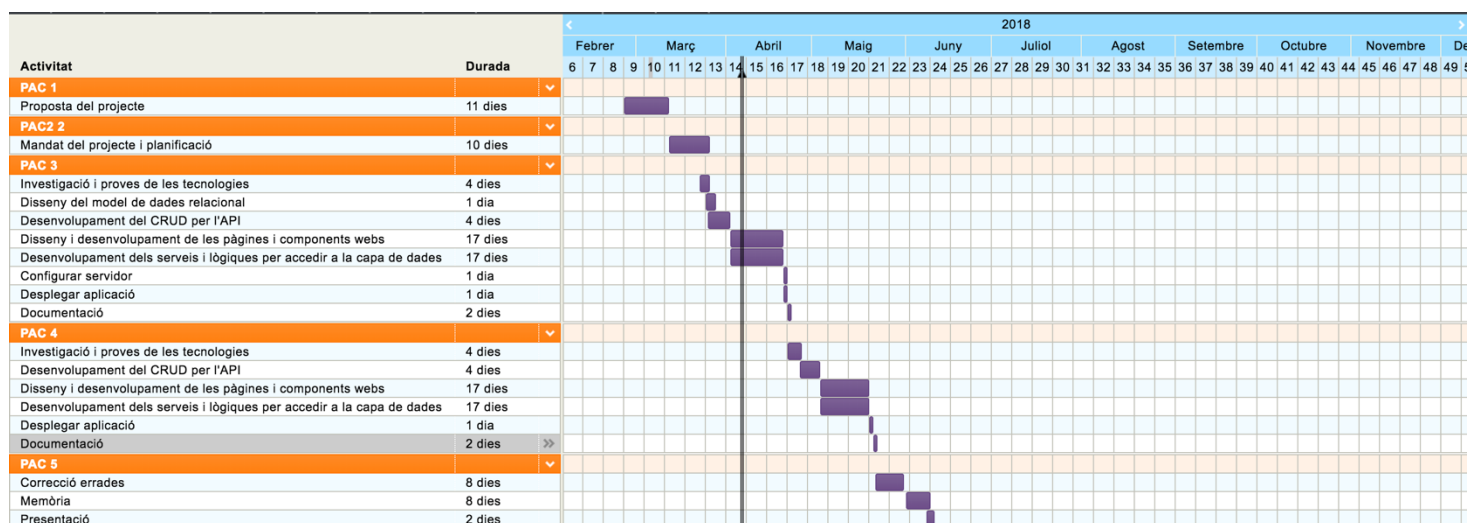


Figura 1: Diagrama de Gantt

6. Pressupost

Els recursos emprats pel desenvolupament del projecte han estat:

- Recursos humans: 1 persona.
- Equip tècnic: ordinador Mac Book Pro Retina.
- Allotjament de l'aplicació: *cloud* gratuït.
- Programari lliure.

Per dur a terme la gestió del projecte i la producció de l'aplicació s'estima una dedicació mitjana de 10 hores setmanals. Tenint en compte que un dels objectius personals d'aquest projecte era la investigació, experimentació i formació en noves tecnologies, la dedicació a les tasques de producció excedeix a 20 hores setmanals aproximadament. Ara bé, les setmanes de durada de la defensa del projecte no es tenen en compte en el cost total del projecte.

A partir de totes aquestes aproximacions, i tenint en compte que el projecte s'inicia el 26 de febrer de 2018 i finalitza l'11 de juny de 2018 (14 setmanes) amb un cost hora de 30€ s'obté el pressupost representat a la Taula 2.

Concepte	Cost	Total
Gestió del projecte	50h * 30€	1500€
Formació en noves tecnologies	40h * 30€	1200€
Disseny	15h * 30€	450€
Programació	50h * 30€	1500€
Desplegar aplicació al servidor	05h * 30€	150€
Allotjament <i>cloud</i>		Gratuït
Ordinador (Amortització de 5 mesos d'una vida útil de 48 mesos amb un valor de 1329€)	138€	138€
Programari lliure (tecnologies emprades)		Gratuït
	SUBTOTAL	4938€
	21 % IVA	1036,98€
	TOTAL	5974,98€

Taula 2: Pressupost del projecte

7. Estructura de la resta del document

El capítol 2 presenta un anàlisi de les solucions existents en el mercat, el públic objectiu de l'aplicació i les característiques i requisits de la mateixa.

El capítol 3 descriu l'arquitectura del sistema, l'arquitectura de la informació tractada per l'aplicació, les tecnologies i patrons utilitzats per a desenvolupar l'aplicació i, també, exposa tot el procés de disseny visual.

El capítol 4 descriu com s'han implementat cada un dels mòduls que conformen l'aplicació desenvolupada.

El capítol 5 presenta les instruccions d'ús de l'aplicació i proporciona els detalls d'accés a la demostració que s'ha preparat.

El Capítol 6 presenta les conclusions del projecte i possibles línies de futur per a l'evolució del sistema.

Capítol 2: Anàlisi

1. Estat de l'art

L'aparició de les "Noves Tecnologies" en les últimes dècades del segle XX ha estat la causa de l'anomenada "Revolució Digital" [5], revolució que, a diferència d'altres anteriors, ha aconseguit que els canvis i les transformacions derivats del que avui es diuen **TIC**, s'hagin produït molt ràpidament en tots els àmbits de la societat. Efectivament, les **TIC** i especialment Internet es desenvolupen i s'incorporen a la vida dels ciutadans a una velocitat vertiginosa. Els efectes que Internet i les seves múltiples aplicacions tenen en la vida dels ciutadans, de les empreses, de les institucions i dels governs s'han manifestat en menys d'una dècada. D'altra banda, s'observen molts canvis en la forma de comunicar-se, d'organitzar-se, fins i tot de treballar o de divertir-se. S'ha configurat una nova societat, la nova Societat de la Informació també anomenada, si es fa una passa més, "Societat del Coneixement", que es caracteritza per la possibilitat d'accedir a volums ingents d'informació i de connectar amb col·lectius o ciutadans fora dels límits de l'espai i del temps. Concretament, un dels factors que han permès que aquests límits es puguin sobrepassar ha estat l'aparició dels telèfons intel·ligents i les tauletes.

D'entre tots els àmbits que envolten la societat actual, l'efecte de les "Noves Tecnologies" sobre l'educació i ensenyament encara suposa, en comparació a altres àmbits, un impacte baix en alguns aspectes. Fent un anàlisi del mercat actual d'aplicacions d'àmbit educatiu, és presenten les següents:

GoConqr

Aplicació que ofereix als docents la possibilitat de impartir les seues classes d'una forma més amena. L'aplicació consisteix en que el docent enregistra les seues idees, formes d'impartir les classes, imatges, traçar mapes mentals, etc. D'aquesta manera, tota aquesta informació pot ésser compartida amb altres docents que estiguin enregistrats a l'aplicació.

L'aplicació permet als usuaris gestionar la següent informació:

- Assignatures: Apartat que permet la creació d'assignatures i gestió dels recursos i temes associats a cada una d'aquestes.
- Cursos: Apartat que permet la creació de cursos i associar-hi mòduls a impartir, temes i qualsevol recurs útil.
- Grups: Apartat útil per a construir grups de discussió sobre temes concret. Aquests grups estan formats per usuaris enregistrats a l'aplicació.
- Descobrir grups: Apartat que permet als usuaris descobrir i unir-se a grups públics.
- Contingut suggerit: Apartat que permet accedir a contingut actualitzat i relacionat amb els temes i assignatures impartides.

L'aplicació es troba disponible per Android, iOS i versió web. Hi ha una versió gratuïta i una versió pagament, la qual ofereix un nombre major de funcionalitats i no disposa de publicitat.



Figura 2: Pantalla principal de GoConqr (versió web).

Attendance

Aplicació que permet als docents actualitzar els informes d'assistència dels seus alumnes. Amb l'aplicació es pot emmagatzemar dades e imatges dels alumnes per tal de que els docents puguin passar llista d'una forma més àgil.

L'aplicació permet als usuaris gestionar la següent informació:

- Cursos: Apartat per crear els diferents cursos gestionats pel docent.
- Períodes acadèmics (trimestres): Apartat per a crear trimestres i assignar-li el seu corresponent període de temps.
- Esdeveniments: Apartat que permet associar períodes acadèmics amb cursos.
- Estudiants: Apartat associat a un curs que permet donar d'alta els estudiants amb la seua corresponent informació.
- Grups d'estudiants: Apartat que permet formar grups d'estudiants d'un curs determinat.
- Calendari: Apartat que permet afegir les hores exactes en que s'imparteixen les classes.

L'aplicació és gratuïta i es troba únicament desenvolupada per al Sistema Operatiu Android.

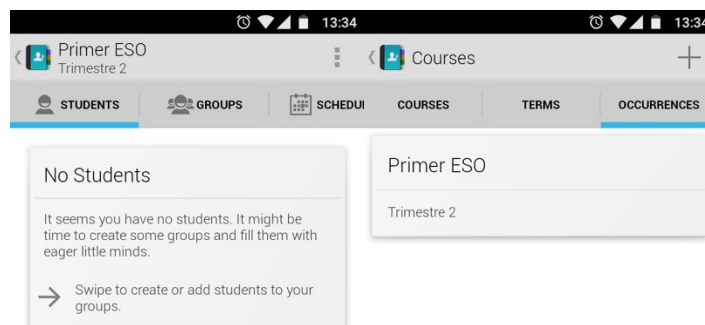


Figura 3: Pantalles Attendance

Edmodo

Aplicació que permet establir la comunicació entre els alumnes i professors en un entorn tancat i privat a mode de *microblogging* [6]. Concretament és una aplicació que permet crear grups als quals se'ls pot assignar tasques, programar el seu enviament i gestionar el progrés del grup.

L'aplicació permet als usuaris gestionar la següent informació:

- Creació de classes: apartat que permet crear classes, assignant-hi alumnes i les tasques corresponents.
- Gestió de classes: apartat que permet gestionar i monitoritzar les diferents classes creades anteriorment.
- Unir-se a classes: apartat que permet als alumnes poder-se unir a les diferents classes creades pels professors i, d'aquesta manera, disposar d'un canal de comunicació entre els dos perfils.
- Creació de grups: apartat que permet crear grups de treball a partir d'un conjunt d'alumnes assignats a una classe.
- Gestió de grups: apartat que permet gestionar els diferents grups de treball creats anteriorment.
- Unir-se a grups: apartat que permet als alumnes poder-se unir a les diferents grups de treball creats pels professors i, d'aquesta manera, disposar d'un canal de comunicació (dins un grup de treball) entre els dos perfils.

Aquesta aplicació únicament es comercialitza com a una aplicació web, és a dir, no es troba disponible per Android i iOS.

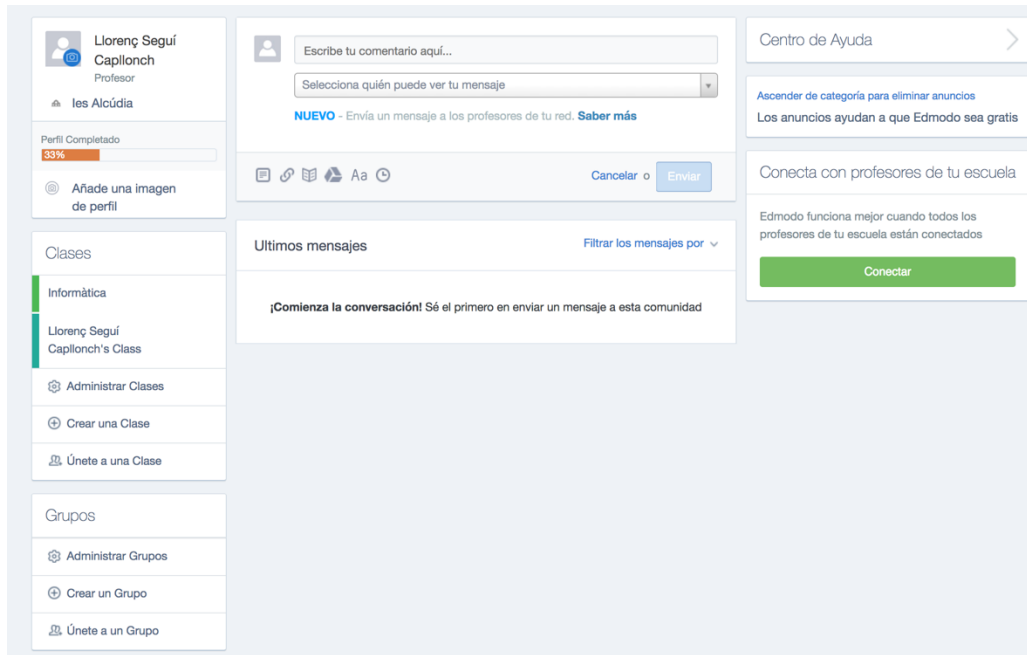


Figura 4: Pantalla Edmodo

Com es pot observar, les aplicacions presentades tenen com a finalitat ajudar als docents a impartir les classes d'una forma més amena, compartir informació entre ells i els seus alumnes, fins hi tot algunes d'elles dur el control de les assistències etc. Ara bé, durant tota aquesta etapa d'estudi del mercat d'aplicacions d'àmbit educatiu són poques les aplicacions que puguin substituir a les clàssiques llibretes de qualificacions que els docents porten a sobre. És per això que sorgeix la necessitat de desenvolupar una aplicació que permeti als docents enregistrar tota la informació acadèmica dels seus alumnes. Concretament, l'aplicació ha de permetre enregistrar les diferents activitats i proves que els alumnes han de realitzar a cada una de les assignatures. A la vegada, per a cada una de les activitats s'ha de poder enregistrar les qualificacions obtingudes pels alumnes i fer els càlculs corresponents per a obtenir les notes finals. Per tal de que l'aplicació es pugui emprar des de qualsevol dispositiu i, a la vegada, sigui independent del sistema operatiu des d'on s'executa, l'aplicació serà de tipus web i adaptable a diferents resolucions per tal de que l'experiència d'usuari sigui la més confortable possible pel públic a qui va dirigida l'aplicació.

2. Públic objectiu i perfils d'usuari

El públic objectiu a qui va dirigida l'aplicació són docents de Centres d'Educació Secundària que tenen la necessitat de disposar d'una aplicació que els hi permeti registrar, des de qualsevol dispositiu, les qualificacions dels seus alumnes i aplicar-hi les tasques d'avaluació sobre aquestes.

3. Especificació de requisits

A l'hora d'especificar els requisits, aquests seran classificats en requisits funcionals i requisits no funcionals. Els requisits funcionals són aquells que descriuen el comportament del sistema. Els requisits no funcionals són aquells que es refereixen a qüestions com la tecnologia concreta a utilitzar, on es durà a terme la instal·lació i operació del sistema, etc.

3.1 Requisits funcionals

Els requisits funcionals de l'aplicació són els següents:

- L'aplicació disposarà d'un formulari de registre per als usuaris.
- L'aplicació permetrà als usuaris autenticar-se amb la seua corresponent compte enregistrada.
- L'aplicació permetrà enregistrar, modificar i eliminar centres d'educació secundària.
- L'aplicació permetrà classificar la informació dels alumnes i activitats en anys acadèmics.
- L'aplicació permetrà crear, modificar i eliminar grups d'alumnes associats a cursos d'ESO i Batxillerat.
- L'aplicació permetrà crear, modificar i eliminar assignatures associades a un curs escolar.
- L'aplicació permetrà associar períodes d'avaluació a les assignatures.
- L'aplicació permetrà matricular alumnes a les assignatures.
- L'aplicació permetrà modificar els alumnes ja matriculats.
- L'aplicació permetrà crear activitats de tipus avaluable i de tipus no avaluable.
- L'aplicació permetrà associar les activitats a períodes d'avaluació.
- Les activitats de tipus avaluable tindran una ponderació i una qualificació associada.
- L'aplicació permetrà modificar les activitats ja enregistrades.
- L'aplicació permetrà qualificar les activitats entregades pels alumnes.
- L'aplicació permetrà calcular la mitjana ponderada dels alumnes en una determinada assignatura.

3.2 Requisits no funcionals

Els requisits no funcionals de l'aplicació són:

- L'aplicació s'executarà des d'un navegador web.
- L'aplicació estarà desenvolupada emprant tecnologies i eines de desenvolupament web.
- Les interfícies que conformaran l'aplicació web tindran un disseny adaptable a qualsevol dispositiu i resolució.
- L'aplicació web separarà la part visual de la part de dades i tractament lògic de les mateixes.
- L'aplicació web es desplegarà en un servidor per tal de que es pugui accedir des de internet.

4. Requisits tecnològics

Un dels objectius de la proposta és el de investigar i experimentar amb noves tecnologies que són punteres dins del mercat de desenvolupament d'aplicacions i serveis web. Per tal de poder assumir aquest objectiu, s'enumeraran i justificaran breument les següents necessitats:

- Tecnologies de desenvolupament de clients web basades en components web: Al tractar-se d'una aplicació web, el que es vol aconseguir es que aquesta es desenvolupi emprant components web, els quals són petits contenidors que encapsulen el codi font necessari per als mateixos.
- Eines, guies i patrons d'estils per al disseny web: Per tal de que l'aplicació web sigui usable per als usuaris, es fera ús d'estils i patrons de disseny més punteres del mercat.
- Tecnologies de desenvolupament de capes d'accés a dades: Per tal de poder separar la capa de presentació de la capa d'accés i tractament de la informació, es necessitarà emprar tecnologies que permetin desenvolupar aquest tipus de software.
- Tecnologies de suport a l'emmagatzemament de les dades: Es requereix un sistema d'emmagatzematge permanent de la informació així com tecnologies per poder accedir-hi.
- Protocol de comunicació segura e intercanvi de dades: Sent una aplicació client-servidor es necessiten disposar de tecnologies estandarditzades per comunicar de forma segura i intercanviar informació entre ambdues parts.

- *Plataformes de desenvolupament i eines de suport:* Es requereixen totes aquelles eines necessàries (IDEs, Editors de text, Control de versions ...) per facilitar els desenvolupaments.

Capítol 3: Disseny

1. Arquitectura general del sistema

L'arquitectura d'un sistema ens diu quins patrons i abstraccions coherents s'han emprat per tal d'interactuar amb el codi font del programari.

L'aplicació desenvolupada en aquest **TFM** ha estat plantejada, principalment, com una aplicació client-servidor. Per tal de poder desenvolupar l'aplicació seguint aquest plantejament, s'ha emprat la tècnica de programació per capes [7]. Aquesta tècnica té com a objectiu primordial separar la lògica de negoci de la lògica de disseny. Per complir aquest objectiu, aquest patró proposa l'ús de tres capes:

- Capa de presentació: és la capa que veu l'usuari, presenta la informació a aquest, li comunica la informació i captura la informació en un mateix procés. També és coneguda com a interfície gràfica.
- Capa de negoci: és la capa on resideixen els programes que s'executen, es reben les peticions de l'usuari i s'envien les respostes després d'aquest procés. S'anomena capa de negoci perquè és on s'han d'establir les regles a complir-se. Aquesta capa es comunica entre la capa de presentació i la capa de dades.
- Capa de dades: és on resideixen i es gestionen les dades. Sol estar constituït per un o més gestors de base de dades i mòduls que permeten realitzar la gestió d'accés i consulta a la informació pertanyent a la base de dades.

Per a aquesta aplicació, la solució de tres capes ha estat implementada sobre una arquitectura de dos nivells (Figura 5). Particularment l'aplicació (Client web) comprendria presentació i les accions executades per l'usuari final, i el servidor comprendria lògica de negoci i dades (**API REST**).

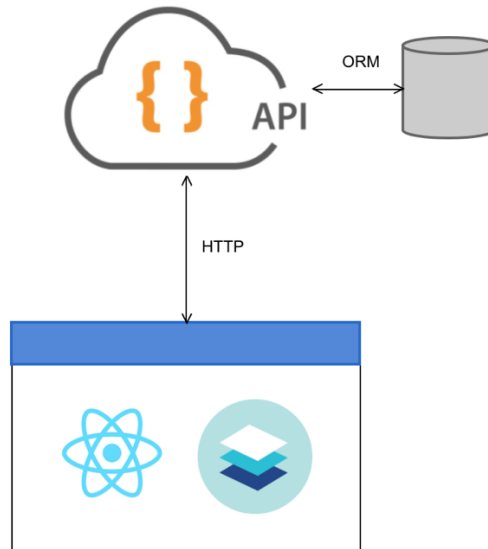


Figura 5: Arquitectura de l'aplicació

Les dues capes es comuniquen a través de peticions **HTTP**. Concretament, el client du a terme, a partir de les accions dels usuaris, les peticions de la informació a l'**API REST**. Aquesta, processa cada petició (obtenint la informació d'una base de dades relacional) i retorna la informació sol·licitada cap al client en format **JSON**. Una vegada el client ha rebut la informació, la processa per tal de plasmar-la a la corresponent interfície.

2. Arquitectura de la informació i diagrames de navegació

2.1 Model de dades

La informació emprada per l'aplicació s'emmagatzema a una base de dades relacional, la qual és atacada per l'API REST. Per tal d'entendre quines són les dades que s'emmagatzemen, la Figura 6 mostra un model de dades relacional representat en forma de diagrama UML [8]:

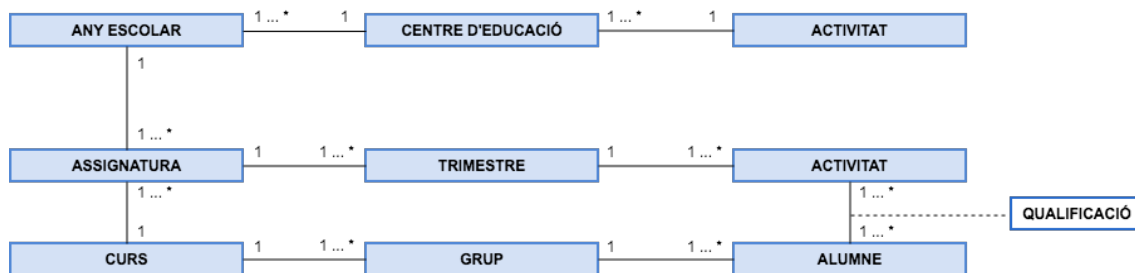


Figura 6: Model de dades de la base de dades

Com es pot observar, la majoria de relacions entre les taules són relacions un a un, exceptuant la relació entre les Activitats i els alumnes.

2.2 Arbre de navegació

Per tal de poder tenir un coneixement de com navegar per l'aplicació web es durà a terme un arbre de navegació del lloc [9]. Els mapes de navegació proporcionen una representació esquemàtica de l'estructura de l'hipertext, indicant els principals conceptes inclosos en l'espai de la informació i les interrelacions que hi ha entre ells. Un mapa és, per exemple, una representació completa (o resumida) del lloc web per orientar el lector / usuari durant el recorregut o per facilitar-li un accés directe al lloc que li interessi. Reflectirà l'estructura del lloc web per mitjà d'enllaços als nodes principals, i aquests també poden desenvolupar-se per mostrar els sub-nodes.

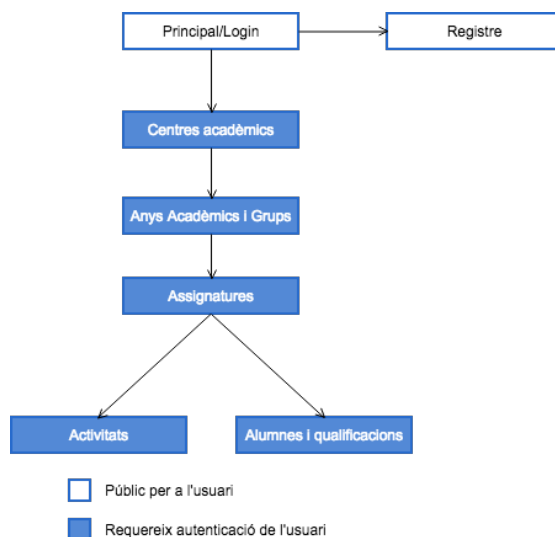


Figura 7: Arbre de navegació de l'aplicació web

Existeixen diferents tipus d'arbres de navegació. El representat a la Figura 7 es correspon a un tipus de navegació composta ja que combina els sistemes de navegació lineal (es rep la informació en un ordre adequat) i jeràrquic (es presenten diferents opcions que permeten visualitzar pàgines més específiques).

3. Tecnologies, guies i eines emprades

HTML5

HTML5 [10] és un llenguatge markup (de fet, les sigles d'**HTML** signifiquen Hyper Text Markup Language) usat per estructurar i presentar el contingut per a la web. Concretament, **HTML5** és la cinquena revisió d'**HTML**.

Javascript

JavaScript [11] és un llenguatge de programació interpretat, dialecte de l'estàndard ECMAScript [12]. Es defineix com orientat a objectes, basat en prototips, imperatiu, dèbilment tipat i dinàmic. S'utilitza principalment en la seva forma del costat del client, implementat com a part d'un navegador web permetent millores en la interfície d'usuari i pàgines web dinàmiques, encara que existeix una forma de JavaScript del costat del servidor.

CSS3

CSS [13] és un llenguatge usat per definir la presentació d'un document estructurat escrit en **HTML** o **XML** (i per extensió en **XHTML**). El **W3C** (World Wide Web Consortium) [14] és l'encarregat de formular l'especificació dels fulls d'estil que serviran d'estàndard per als agents d'usuari o navegadors.

Material Design

Material Design [15] es un estàndard de disseny instaurat per Google que permet facilitat l'experiència d'usuari i tenir un disseny unificat per a diferents resolucions. Com el seu nom indica, es basa en objectes materials. Aquests objectes s'agrupen a través de diferents capes per aconseguir una aparença de relleu amb ombres i colors plans. Un altre dels pilars fonamentals és l'ús d'animacions i transicions lògiques, amb l'objectiu de donar la sensació que els objectes es guien per les lleis de la física.

Node JS i NPM

Node.JS [16] és un entorn en temps d'execució multi-plataforma, de codi obert, per a la capa del servidor (però no limitant-se a això) basat en el llenguatge de programació ECMAScript, asíncron, amb I/O de dades en una arquitectura orientada a esdeveniments i basat en el motor V8 de Google. Va ser creat amb l'enfocament de ser útil en la creació de programes de xarxa altament escalables, com per exemple, servidors web.

Node JS s'ha instal·lat per tal de poder disposar del gestor de paquets **NPM** [17] que permeti gestionar totes les dependències utilitzades per a desenvolupar el client web.

React JS

React [18] (també anomenada React.js o ReactJS) és una llibreria Javascript de codi obert per crear interfícies d'usuari (emprant **HTML** + Javascript + **CSS**) amb l'objectiu d'animar al desenvolupament d'aplicacions en una sola pàgina. És mantingut per *Facebook*, *Instagram* i una comunitat de desenvolupadors independents.

React intenta ajudar als desenvolupadors a construir aplicacions que fan servir dades que canvien tot el temps. El seu objectiu és ser senzill, declaratiu i fàcil de combinar. React només fa servir la interfície d'usuari en una aplicació; està construïda únicament per utilitzar el patró de disseny model-vista-controlador (**MVC**).

Material-UI-Next

Material-UI-Next [19] és una llibreria que disposa de components web de React dissenyats seguint l'estàndard Material Design de Google.

Webpack

Webpack [20] és un sistema de bundling per preparar el desenvolupament d'aplicacions web per a producció. En certa mesura és oïlla considerar un *Browserify* avançat ja que te moltes opcions de Configuració. Permet d'alguna manera automatitzar els processos principals que són transpilar i pre-processar codi de .scss a .css, de ECMAScript7 a ECMAScript 5 / 6, etc.

Babel

Babel [21] és una eina que ens permet transformar el nostre codi JavaScript d'última generació (o amb funcionalitats extremes) a JavaScript que qualsevol navegador o versió de Node.JS entengui.

Python

Python [22] és un llenguatge de programació interpretat on la seua filosofia posa l'accent en una sintaxi que afavoreixi un codi llegible.

Es tracta d'un llenguatge de programació multiparadigma, ja que suporta orientació a objectes, programació imperativa i, en menor mesura, programació funcional. És un llenguatge interpretat, usa tipus dinàmics i és multiplataforma.

Django REST Framework

Django **REST** Framework [23] és un marc de treball que permet desenvolupar **APIs REST** sobre Django, emprant Python com a llenguatge de programació. Concretament, aquest framework disposa d'un potent **ORM** [24] que facilita molt l'accés i tractament de la informació emmagatzemada a una base de dades.

SQLite

SQLite [25] és un sistema de gestió de bases de dades relacional compatible amb **ACID**. A diferència dels sistemes de gestió de bases de dades client-servidor, el motor de SQLite no és un procés independent amb el qual el programa principal es comunica. En lloc d'això, la biblioteca SQLite s'enllaça amb el programa passant a ser part integral del mateix.

El programa utilitza la funcionalitat de SQLite a través de trucades simples a subrutines i funcions. Això redueix la latència en l'accés a la base de dades, pel fet que les crides a funcions són més eficients que la comunicació entre processos. El conjunt de la base de dades (definicions, taules, índexs, i les pròpies dades), són guardats com un sol fitxer estàndard en la màquina host. Aquest disseny simple s'aconsegueix bloquejant tot el fitxer de base de dades al principi de cada transacció.

JSON

JavaScript Object Notation [26] és un format lleuger d'intercanvi de dades. Es constitueix per dues estructures:

- Una col·lecció de parells del tipus "nom": "valor". En diversos llenguatges és un objecte, un registre, una estructura, un diccionari, una taula *hash*, una llista amb claus o un matriu associada.
- Una llista ordenada de valors. En la majoria de llenguatges, es realitzaria amb una matriu, una llista o una seqüència.

La Figura 8 mostra un exemple d'un **JSON**:

```
{ "menu": {  
  "id": "file",  
  "value": "File",  
  "popup": {  
    "menuitem": [  
      { "value": "New", "onclick": "CreateNewDoc()" },  
      { "value": "Open", "onclick": "OpenDoc()" },  
      { "value": "Close", "onclick": "CloseDoc()" }  
    ]  
  }  
}
```

Figura 8: Exemple d'un **JSON**.

JWT

JSON Web Token [27] és un estàndard obert (**RFC-7519**) basat en **JSON** per crear un *token* que serveixi per enviar dades entre aplicacions o serveis i garantir que siguin vàlids i segurs.

El cas més comú d'ús dels **JWT** és per gestionar l'autenticació en aplicacions mòbils o web. Per això quan l'usuari es vol autenticar mana les seves dades d'inici de sessió al servidor, aquest genera el **JWT** i l'hi envia a l'aplicació client, després en cada petició el client envia aquest testimoni que el servidor fa servir per verificar que l'usuari aquest correctament autenticat i saber qui és.

Heroku

Heroku [28] és una plataforma com a servei de computació en el Núvol que suporta diferents llenguatges de programació. Particularment es emprada per a desplegar aplicacions al Núvol i fer que aquestes puguin ser accessibles des de Internet.

Git

Git [29] és un programari de control de versions, pensant en l'eficiència i la fiabilitat del manteniment de versions d'aplicacions quan aquestes tenen un gran nombre d'arxius de codi font. El seu propòsit és portar registre dels canvis en arxius d'ordinador i coordinar el treball que diverses persones realitzen sobre arxius compartits.

GitHub

GitHub [30] és una forja (plataforma de desenvolupament col·laboratiu) per allotjar projectes utilitzant el sistema de control de versions Git. S'utilitza principalment per a la creació de codi font de programes d'ordinador.

Visual Studio Code

Visual Studio Code [31] és un editor de codi font desenvolupat per Microsoft per a Windows, Linux i MacOS. Inclou suport per a la depuració, control integrat de Git, ressaltat de sintaxi, finalització intel·ligent codi, fragments i refactorització de codi. També és personalitzable, de manera que els usuaris poden canviar el tema de l'editor, les dreceres de teclat i les preferències. És gratuït i de codi obert, encara que la descàrrega oficial està sota programari propietari.

5. Tecnologies implicades a cada mòdul de l'arquitectura

Tenint en compte l'arquitectura de l'aplicació a la primera secció d'aquest capítol, les tecnologies implicades a cada una de les parts queden plasmades a la Taula 3.

Tecnologia, guia i eina	Mòdul (<i>Front-End</i> o <i>Back-End</i>)
HTML5	Client web
JavaScript	Client web
CSS3	Client web
Material Design	Client web
Node JS i NPM	Client web
React JS (Material-UI-Next)	Client web
Webpack	Client web
Babel	Client web
Python	API REST
Django REST Framework	API REST
SQLite	API REST
JSON	Client web i API REST
JWT	Client web i API REST
Heroku	Client web i API REST
Git	Client web i API REST
GitHub	Client web i API REST
Visual Studio Code	Client web i API REST

Taula 3: Tecnologies implicades a cada mòdul.

4. Disseny gràfic i interfícies

4.1 Estils

Els estils que conformen l'aplicació web segueixen la normativa *Material Design* [32] de Google. Aquesta normativa reb el seu nom per estar basada en objectes materials a on la profunditat del disseny, les superfícies, les vores, les ombres i els colors tenen un protagonisme important. *Material Design* és un disseny amb una tipografia clara, caselles ben ordenades imatges i colors cridaners. Tots aquests aspectes ajuden a l'usuari a no perdre el focus i el sentit de l'ordre i el marcat.

L'aplicació web desenvolupada és consistent a l'ús d'elements d'**UI**, paletes de colors, icones i topografies propis de la norma *Material Design*.

Respecte als elements d'**UI**, s'han emprat botons flotants, botons elevats, *chips*, diàlegs, menús, camps de text i notificacions. La Figura 9 mostra cada un d'aquests elements.



Figura 9: Components *Material Design* emprats en l'aplicació web

Respecte als colors de l'aplicació web, aquests estan configurats seguint les paletes de colors establertes per la norma de *Material Design*. Concretament, s'ha optat per una paleta de blaus i grisos i blanc (aquest es el color primari per a *Material Design*) per tal de dotar-hi un aspecte cridaner, com molt bé es vol aconseguir amb la norma *Material Design*. La Figura 10 mostra la paleta de colors configurada per aquesta aplicació.



Colors aplicació web	
500	#3F51B5
50	#E8EAF6
100	#C5CAE9
200	#9FA8DA
300	#7986CB
400	#5C6BC0
500	#3F51B5
600	#3949AB
700	#303F9F
800	#283593
900	#1A237E

Figura 10: Paleta de colors configurada per a l'aplicació web

Respecte a la tipografia, s'ha utilitzat la família de fonts tipogràfiques *Roboto* [33] de Google. La Figura 11 mostra la font *Roboto*.

ABCDEFGHIJKLMN
OPQRSTUVWXYZÀÅ
abcdefghijklmnopqr
stuvwxyzàåéîõøü&1
234567890(\$£€.,!?)

Figura 11: Font Roboto

4.2 Usabilitat/UX

Les interfícies de l'aplicació web, aquestes tenen la mateixa estructura. Estan formades per una barra principal de l'aplicació i el contingut de la pàgina en la que l'usuari està navegant. Un pes important es que les interfícies s'han dissenyat per tal de que es puguin adaptar a diferents resolucions. Em altres paraules, les interfícies estan construïdes segun la tècnica de disseny *responsive*. La Figura 12, 13 i 14 mostren, respectivament, l'estructura de les interfícies per a resolucions d'ordinador, *smartphone* i tauleta.



Figura 12: Estructura interfície (Resolució per ordinadors)

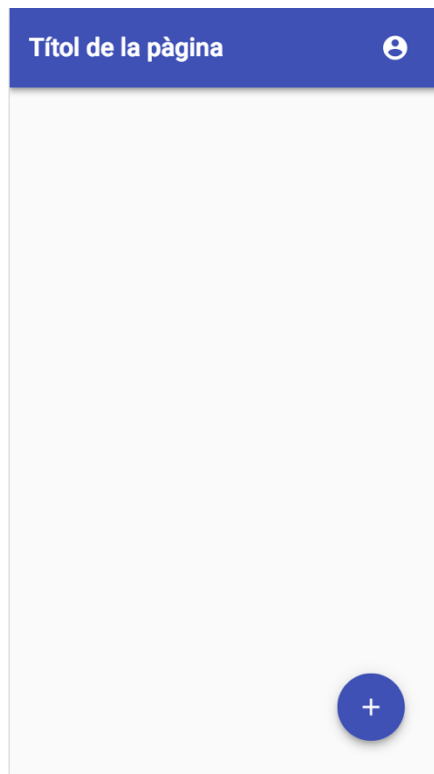


Figura 13: Estructura interfície (Resolució per *smartphone*)

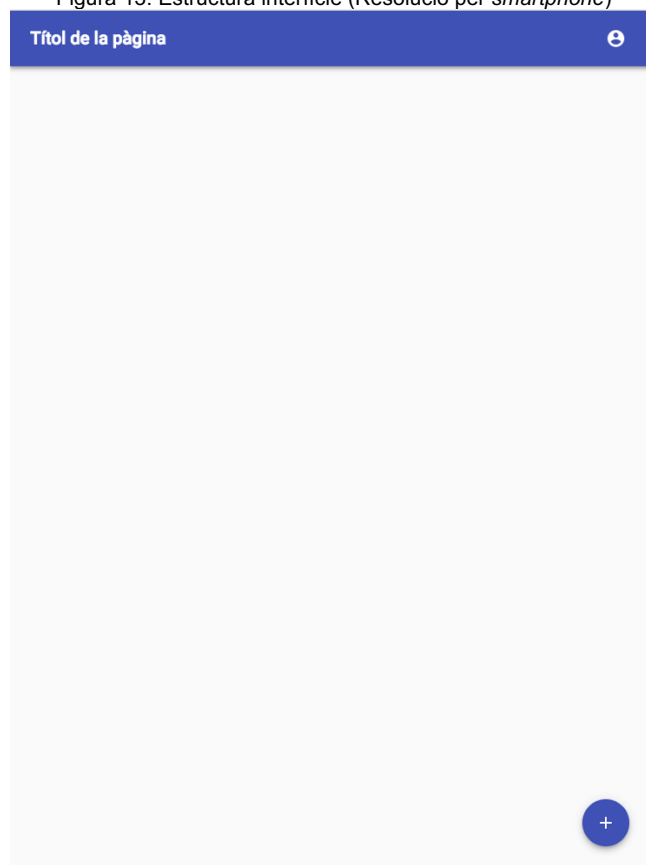


Figura 14: Estructura interfície (Resolució per tauletes)

Capítol 4: Implementació

1. Implementació de l'API REST

Partint del model de dades explicat a la secció 2.1 del capítol 3, s'ha utilitzat el marc de treball Django **REST** Framework per tal de desenvolupar l'**API REST** que fera us de les dades emmagatzemades a la Base de dades.

A partir del model de dades, s'han desenvolupat els models de Django (fitxer *model.py*). Un model de Django és una classe que ve a representar un objecte emmagatzemat a la Base de dades. En altres paraules, un model de Django ens permet gestionar manipular registres de Base de dades en forma d'objectes. La Figura 15 es mostra un exemple de model de Django.

```
class Centre(models.Model):
    nom = models.CharField(max_length=100, default='')
    ubicacio = models.CharField(max_length=100, default='')
    professor = models.ForeignKey(Professor, related_name='alta', on_delete=models.CASCADE)

    def __str__(self):
        return self.nom
```

Figura 15: Model de Django de l'entitat Centre

En segon lloc, una vegada implementats tots els models s'han implementat els serialitzadors de Django (fitxer *serializer.py*). Un serialitzador de Django es aquella classe que representa tota aquella informació dels models que es vol serialitzar en format **JSON** (pot ser un conjunt dels atributs o el total d'atributs d'un model). La Figura 16 mostra un exemple de serialitzador de Django.

```
class CentreSerializer(serializers.ModelSerializer):
    class Meta:
        model = Centre
        fields = '__all__'
```

Figura 16: Serialitzador de Django del model Centre

En tercer lloc, una vegada implementats tots els serialitzadors, s'implementen les vistes (fitxer *views.py*) les quals representen el procés de consulta de models, tractament de la informació obtinguda i serialització de la mateixa (en format **JSON**). Concretament, s'implementen tantes vistes com tipus de peticions (GET, POST, PUT, DELETE) a realitzar a cada una d'elles.

Per llistar e inserta recursos (peticions GET i POST) les vistes es desenvolupen com es mostra la Figura 17.

```
class CentreViewList(APIView):  
    def get(self, request, format=None):  
        try:  
            token = request.META['HTTP_AUTHORIZATION']  
            jwt.decode(token, 'secret', algorithms=['HS256'])  
        except Exception:  
            return Response({'error': 'Token is invalid or does not exist'}, status=401)  
  
        objectes = Centre.objects.all()  
        serializer = CentreSerializer(objectes, many=True)  
        return Response(serializer.data)  
  
    def post(self, request, format=None):  
        try:  
            token = request.META['HTTP_AUTHORIZATION']  
            jwt.decode(token, 'secret', algorithms=['HS256'])  
        except Exception:  
            return Response({'error': 'Token is invalid or does not exist'}, status=401)  
  
        serializer = CentreSerializer(data=request.data)  
        if serializer.is_valid():  
            serializer.save()  
            return Response(serializer.data, status=status.HTTP_201_CREATED)  
        return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)
```

Figura 17: Vista corresponent a les peticions GET (llistar) i POST dels Centres

Per obtenir un recurs, modificar un recurs i eliminar un recurs (peticions GET, PUT i DELETE) les vistes es desenvolupem com es mostra a la Figura 18:

```
class CentreViewDetail(APIView):
    def get_object(self, pk):
        try:
            return Centre.objects.get(pk=pk)
        except Centre.DoesNotExist:
            raise Http404

    def get(self, request, pk, format=None):
        try:
            token = request.META['HTTP_AUTHORIZATION']
            jwt.decode(token, 'secret', algorithms=['HS256'])
        except Exception:
            return Response({'error': 'Token is invalid or does not exist'}, status=401)

        objecte = self.get_object(pk)
        serializer = CentreSerializer(objecte)
        return Response(serializer.data)

    def put(self, request, pk, format=None):
        try:
            token = request.META['HTTP_AUTHORIZATION']
            jwt.decode(token, 'secret', algorithms=['HS256'])
        except Exception:
            return Response({'error': 'Token is invalid or does not exist'}, status=401)

        objecte = self.get_object(pk)
        serializer = CentreSerializer(objecte, data=request.data)
        if serializer.is_valid():
            serializer.save()
            return Response(serializer.data)
        return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)

    def delete(self, request, pk, format=None):
        try:
            token = request.META['HTTP_AUTHORIZATION']
            jwt.decode(token, 'secret', algorithms=['HS256'])
        except Exception:
            return Response({'error': 'Token is invalid or does not exist'}, status=401)

        objecte = self.get_object(pk)
        objecte.delete()
        return Response(status=status.HTTP_204_NO_CONTENT)
```

Figura 18: Vista corresponent a les peticions GET (obtenir), PUT i DELETE dels Centres

Finalment, per tal de que les vistes puguin ser trucades a través de peticions **HTTP**, cal configurar les **URLs**. Aquestes **URLs** es configuren al fitxer *urls.py*. La figura 19 mostra l'exemple del fitxer.

```
urlpatterns = [
    url(r'^$', include(router.urls)),
    url(r'^login/$', views.LoginView.as_view()),
    url(r'^api-auth/', include('rest_framework.urls', namespace='rest_framework')),
    url(r'^professors/$', views.ProfessorViewList.as_view()),
    url(r'^professors/(?P<pk>[0-9]+)/$', views.ProfessorViewDetail.as_view()),
    url(r'^assignatures/$', views.AssignaturaViewList.as_view()),
    url(r'^assignatures/(?P<pk>[0-9]+)/$', views.AssignaturaViewDetail.as_view()),
    url(r'^cursos/$', views.CursViewList.as_view()),
    url(r'^cursos/(?P<pk>[0-9]+)/$', views.CursViewDetail.as_view()),
    url(r'^centres/$', views.CentreViewList.as_view()),
    url(r'^centres/(?P<pk>[0-9]+)/$', views.CentreViewDetail.as_view()),
]
```

Figura 19: Exemple d'un fitxer d'URLs de Django REST Framework

2. Implementació de l'aplicació web

El desenvolupament d'aquesta aplicació s'ha plantejat com una aplicació de pàgina única amb l'objectiu de donar una experiència més fluida als usuaris com una aplicació d'escriptori. Per poder assolir aquest plantejament, l'aplicació s'ha desenvolupat aplicant components web. Els components web són elements que es caracteritzen per tenir el seu propi codi **HTML**, JavaScript i **CSS**. La avantatge que tenen els components web és que poden ser reutilitzables i cridats des de altres components.

Per tal de poder desenvolupar l'aplicació web seguint aquestes característiques, s'ha emprat la llibreria ReactJS. A partir d'aquesta llibreria, s'han creat tots els components web necessaris per a construir l'aplicació. La Figura 20 mostra tots els components webs creats.

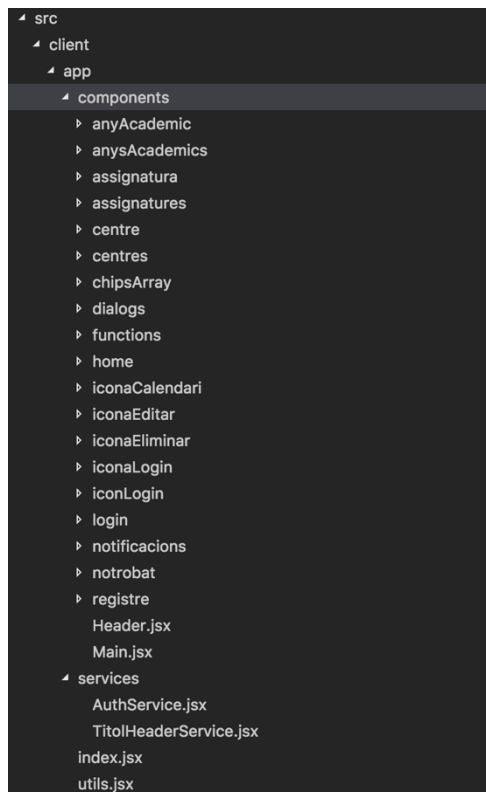


Figura 20: Estructura del projecte (ReactJS)

Els components web són arxius d'extensió `jsx`. JSX és un pseudollenguatge que permet modelar elements **HTML** des de Javascript. Aquí es presenta un exemple de sintaxis `jsx`:


```
render () {  
  return (  
    <div>  
      <Header/>  
      <Main/>  
    </div>  
  );  
}
```

Cada component de ReactJS està compost del codi JavaScript que utilitza per fer les accions i les propietats **CSS** per tal de donar-li estils als elements. Els estils **CSS** es representen de la següent manera:

```
const styles = theme => ({  
  root: theme.mixins.gutters({  
    paddingTop: 16,  
    paddingBottom: 16,  
    marginTop: theme.spacing.unit * 3,  
  }),  
  textField: {  
    marginLeft: theme.spacing.unit,  
    marginRight: theme.spacing.unit,  
    width: '100%',  
  },  
  button: {  
    margin: theme.spacing.unit,  
  },  
});
```

Per tal de que el navegador pugui interpretar els arxius d'extensió `jsx`, s'ha emprat Webpack i *Babel* per tal de poder traspilar el codi JSX a JavaScript (ECMAScript 5) per tal de que pugui ser interpretat pels navegadors.

3. Desplegament de l'aplicació a Heroku

Una vegada desenvolupats els dos mòduls, aquests s'han desplegat a la plataforma *cloud* Heroku. En primer lloc, s'ha efectuat el registre d'una compte gratuïta a Heroku. El servei gratuït permet disposar fins a cinc aplicacions. En aquest cas s'han donat d'alta dues aplicacions (client web i **API REST**). Una vegada creades les dues aplicacions, el procés de desplegament consisteix en afegir el codi font de cada mòdul l'aplicació a un repositori de Git de Heroku. En altres paraules, s'ha d'executar una instrucció *push* de Git a un repositori de Heroku. Ja que a aquest projecte s'ha emprat Git com a programari de control de versions no, aquest procés no ha suposat cap dificultat. Una vegada desplegats cada un dels mòduls de l'aplicació, Heroku ha generat les següents rutes:

- Client web: <https://tfm-client-app-1.herokuapp.com>
- **API REST**: <https://sleepy-hollows-69752.herokuapp.com/>

4. Control de versions i repositori de software

Per tal de poder gestionar àgilment els canvis aplicats sobre el codi font, s'ha emprat Git com a software de control de versions. D'aquesta manera, es pot tenir un registre de canvis aplicats sobre el codi font de l'aplicació i anar guardant versions del producte en totes les fases de desenvolupament. El codi font i les versions del mateix s'han emmagatzemat a Github. Concretament, s'han creat dos repositoris (un per cada mòdul de l'aplicació). Aquests són:

- Client web: https://github.com/llorenssequi/tfm_client_app
- **API REST**: https://github.com/llorenssequi/tfm_api

Aquests repositoris són públics, la qual cosa implica que els usuaris són lliures de clonar-se el codi font de l'aplicació.

Capítol 5: Demostració

1. Funcionament de l'API REST

Per tal de comprovar el funcionament de l'aplicació s'ha d'accedir a la següent ruta:

<https://sleepy-hollows-69752.herokuapp.com/>

Totes les peticions cap a l'API REST són peticions **HTTP** de tipus GET, POST, PUT i DELETE. Cada una de les peticions (exceptuant la petició GET d'autenticació o *login*, la qual només du la capçalera corresponent al tipus de contingut) han d'incorporar les següents capçaleres:

```
Authorization: jwt_resposta_peticio_login  
Content-Type: application-json
```

La primera capçalera és per estar autenticat a cada una de les cridades per tal de que l'accés a la informació sigui segur. La segona capçalera representa el tipus de contingut de les peticions i respostes **HTTP**. En aquest cas el tipus és **JSON**.

Respecte a la **URL** de l'API REST, el present document mostra la ruta del servidor a on es troba allotjada l'API REST (sleepy-hollows-69752). Ara bé, en el cas de que es volgués provar en local, cal substituir sleepy-hollows-69752 per 127.0.0.1:8000 (consultar el manual d'instal·lació per a més informació).

Per tal de poder fer proves de l'API REST es pot fer a través de la instrucció **CURL** dels terminals de **UNIX** o emprant un software per a fer peticions **HTTP**. Un dels més famosos del mercat és POSTMan [1].

1.1 Autenticació a l'API REST

Per tal de poder realitzar les peticions cap a l'API REST cal estar autenticat. Per a poder-se autenticar i obtenir el **JWT** per afegir com a capçalera de les peticions restants s'ha de seguir el següent procés:

Registrar un compte (Professor). Aquest procés consisteix en llançar la següent petició post (únicament amb la capçalera de Content-Type):

<http://sleepy-hollows-69752.herokuapp.com/professors/>

```
{
  "email": "prova@prova.com",
  "password": "prova1234",
  "nom": "Prova",
  "congnom_1": "Prova_cng_1",
  "congnom_2": "Prova_cng_2",
  "data_naixement": "1993-09-21"
}
```

Una vegada s'ha enregistrat el Professor, ja es pot llançar la següent petició POST per tal de poder obtenir un JWT que permetrà realitzar les peticions restants a l'API REST:

<http://sleepy-hollows-69752.herokuapp.com/login/>

```
{
  "email": "prova@prova.com",
  "password": "prova1234"
}
```

Amb aquesta petició s'obté la següent resposta:

```
{
  "token":
  "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwiZXhwIjoxNTIzOTc2NTczLmVudCJlbWFpbCI6ImxzzZWdlaUBob3R1bGJlZHMuY29tIn0.DiD7RvNP1mgd6I1EnibgK3ZuM8yc66ufUzvtNUDQXJo",
  "professor": {
    "nom": "Prova",
    "congnom_1": " Prova_cng_1",
    "congnom_2": " Prova_cng_2",
    "email": "prova@prova.com"
  }
}
```

Com es pot observar, s'obté un **JWT** el qual serà passat com a capçalera Authorization a les peticions restants de l'**API REST**.

1.2 Peticions GET i DELETE

Per tal de poder llistar, recuperar i eliminar tota la informació relacionada amb els Professors, Assignatures, Cursos i Centres acadèmics llançar, respectivament, les següents peticions (amb les corresponents capçaleres d'autenticació i tipus de contingut):

GET (Llistar i recuperar)

Llistar: <http://sleepy-hollows-69752.herokuapp.com/professors/>

Recuperar: <http://sleepy-hollows-69752.herokuapp.com/professors/<id>/>

Llistar: <http://sleepy-hollows-69752.herokuapp.com/assignatures/>

Recuperar: <http://sleepy-hollows-69752.herokuapp.com/assignatures/<id>/>

Llistar: <http://sleepy-hollows-69752.herokuapp.com/cursos/>

Recuperar: <http://sleepy-hollows-69752.herokuapp.com/cursos/<id>/>

Llistar: <http://sleepy-hollows-69752.herokuapp.com/centres/>

Recuperar: <http://sleepy-hollows-69752.herokuapp.com/centres/<id>/>

Llistar: <http://sleepy-hollows-69752.herokuapp.com/anysacademics/>

Recuperar: <http://sleepy-hollows-69752.herokuapp.com/anysacademics/<id>/>

Llistar: <http://sleepy-hollows-69752.herokuapp.com/alumnes/>

Recuperar: <http://sleepy-hollows-69752.herokuapp.com/alumnes/<id>/>

Llistar: <http://sleepy-hollows-69752.herokuapp.com/trimestres/>

Recuperar: <http://sleepy-hollows-69752.herokuapp.com/trimestres/<id>/>

Llistar: <http://sleepy-hollows-69752.herokuapp.com/activitats/>

Recuperar: <http://sleepy-hollows-69752.herokuapp.com/activitats/<id>/>

Llistar: <http://sleepy-hollows-69752.herokuapp.com/grups/>

Recuperar: <http://sleepy-hollows-69752.herokuapp.com/grups/<id>/>

Llistar: <http://sleepy-hollows-69752.herokuapp.com/qualificacions/>

Recuperar: <http://sleepy-hollows-69752.herokuapp.com/qualificacions/<id>/>

A on <id> correspon a l'identificador del recurs en concret que es vol recuperar.

DELETE

<http://sleepy-hollows-69752.herokuapp.com/professors/<id>/>

<http://sleepy-hollows-69752.herokuapp.com/assignatures/<id>/>

<http://sleepy-hollows-69752.herokuapp.com/cursos/<id>/>

<http://sleepy-hollows-69752.herokuapp.com/centres/<id>/>

<http://sleepy-hollows-69752.herokuapp.com/anysacademics/<id>/>

<http://sleepy-hollows-69752.herokuapp.com/alumnes/<id>/>

<http://sleepy-hollows-69752.herokuapp.com/trimestres/<id>/>

<http://sleepy-hollows-69752.herokuapp.com/activitats/<id>/>

<http://sleepy-hollows-69752.herokuapp.com/grups/<id>/>

<http://sleepy-hollows-69752.herokuapp.com/qualificacions/<id>/>

A on <id> correspon a l'identificador del recurs en concret que es vol eliminar.

1.3 Peticions POST i PUT

Per tal de poder crear i modificar total la informació relacionada amb els Professors, Assignatures, Cursos i Centres acadèmics llançar, respectivament, les següents peticions (amb les corresponents capçaleres d'autenticació i tipus de contingut):

POST

Per a cada una de les peticions de tipus POST, s'indica la ruta i el cos de la petició, el qual està representat en format **JSON**.

<http://sleepy-hollows-69752.herokuapp.com/professors/>

```
{  
  "email": "string",  
  "password": "string",  
  "nom": "string",  
  "congnom_1": "string",  
  "congnom_2": "string",  
  "data_naixement": "aaaa-mm-dd" //Exemple: 1993-09-21  
}
```

*Nota: Aquesta petició, com ja s'ha explicat anteriorment, no requereix de capçalera d'autenticació perquè és la que s'encarrega de registrar les comptes a emprar per generar **JWT**.

<http://sleepy-hollows-69752.herokuapp.com/assignatures/>

```
{  
  "nom": "String",  
  "curs": Nombre,  
  "anyAcademic": Nombre  
}
```

<http://sleepy-hollows-69752.herokuapp.com/cursos/>

```
{  
  "nom": "String",  
  "nivell": "ESO/Batxillerat", //Un dels dos valors  
}
```

<http://sleepy-hollows-69752.herokuapp.com/centres/>

```
{  
  "nom": "String",  
  "ubicacio": "String", //Un dels dos valors  
}
```

<http://sleepy-hollows-69752.herokuapp.com/anysacademics/>

```
{  
  "anyInici": Nombre,  
  "anyFi": Nombre,  
  "centre": Nombre  
}
```

PUT

Per a cada una de les peticions de tipus PUT, s'indica la ruta i el cos de la petició, el qual està representat en format **JSON**.

<http://sleepy-hollows-69752.herokuapp.com/professors/>

```
{  
  "email": "string",  
  "password": "string",  
  "nom": "string",  
  "congnom_1": "string",  
  "congnom_2": "string",  
  "data_naixement": "aaaa-mm-dd" //Exemple: 1993-09-21  
}
```

<http://sleepy-hollows-69752.herokuapp.com/assignatures/>

```
{  
  "nom": "String",  
  "curs": Nombre,  
  "anyAcademic": Nombre  
}
```

<http://sleepy-hollows-69752.herokuapp.com/cursos/>


```
{  
  "nom": "String",  
  "nivell": "ESO/Batxillerat", //Un dels dos valors  
}
```

<http://sleepy-hollows-69752.herokuapp.com/centres/>

```
{  
  "nom": "String",  
  "ubicacio": "String", //Un dels dos valors  
}
```

<http://sleepy-hollows-69752.herokuapp.com/anysacademics/>

```
{  
  "anyInici": Nombre,  
  "anyFi": Nombre,  
  "centre": Nombre  
}
```

<http://sleepy-hollows-69752.herokuapp.com/alumnes/>

```
{  
  "nom": "String",  
  "congnom_1": "String",  
  "congnom_2": "String",  
  "grup": Nombre,  
  "centre": Nombre  
}
```

<http://sleepy-hollows-69752.herokuapp.com/trimestres>

```
{  
  "nom": "String",  
  "assignatura": Nombre  
}
```

<http://sleepy-hollows-69752.herokuapp.com/activitats/>

```
{  
  "nom": "String",  
  "ponderacio": Nombre,  
  "avaluable": Boolea,  
  "trimestre": Nombre  
}
```

<http://sleepy-hollows-69752.herokuapp.com/grups/>

```
{  
  "nom": "String",  
  "curs": Nombre,  
  "centre": Nombre  
}
```

<http://sleepy-hollows-69752.herokuapp.com/qualificacions/>

```
{  
  "qualificacio": "Nombre", //Exemple: "7.0"  
  "alumne": Nombre,  
  "activitat": Nombre  
}
```

2. Funcionament de l'aplicació web

Per tal de comprovar el funcionament de l'aplicació s'ha d'accedir a la següent ruta:

<https://tfm-client-app-1.herokuapp.com>

Compte de prova:

Email: lsc025@uoc.edu

Contrasenya: 123

Accedint a la ruta proporcionada, es mostrarà una pantalla de *login* (Figura 21) que permetrà als usuaris autenticar-se. En el cas de no disposar d'un compte, es pot accedir al formulari de registre (Figura 22).

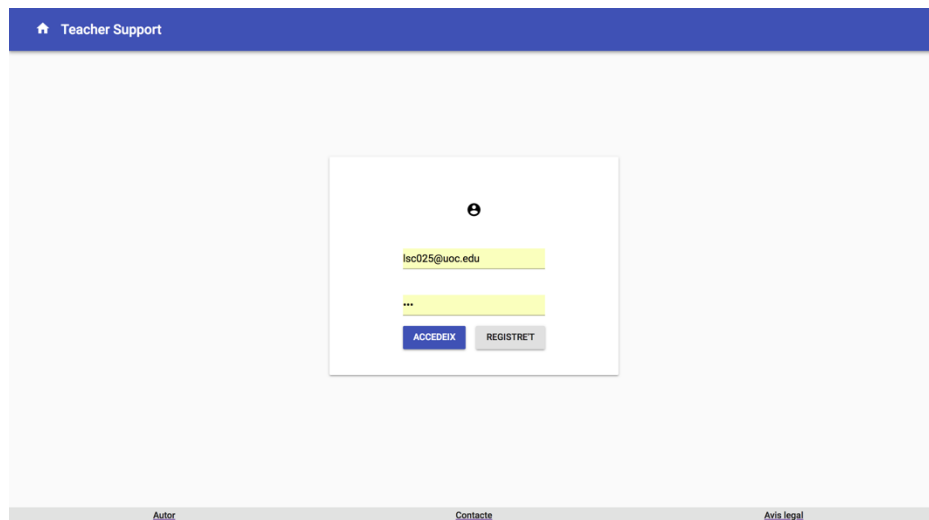


Figura 21: Pàgina d'autenticació

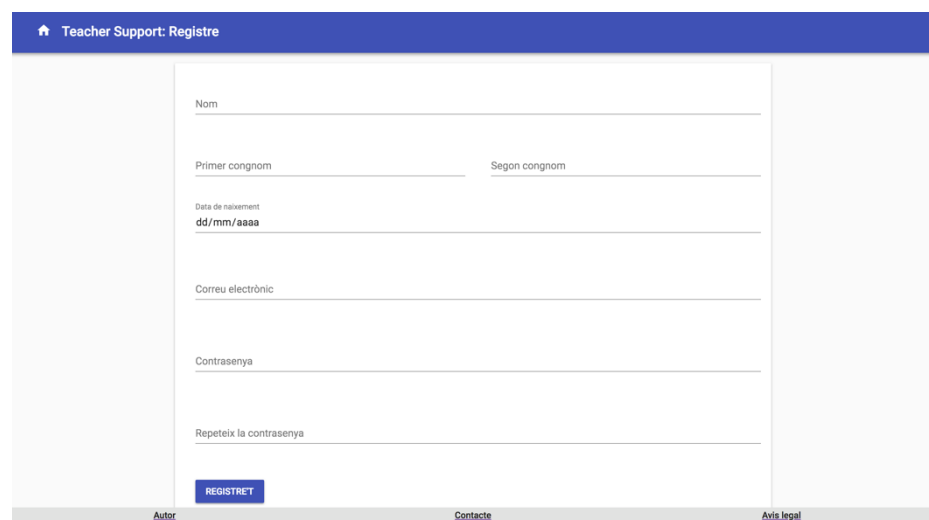


Figura 22: Pàgina de registre

Una vegada autenticats, s'accedeix a la pàgina de gestió dels centres acadèmics en els que està impartit classe un professor (Figura 23).

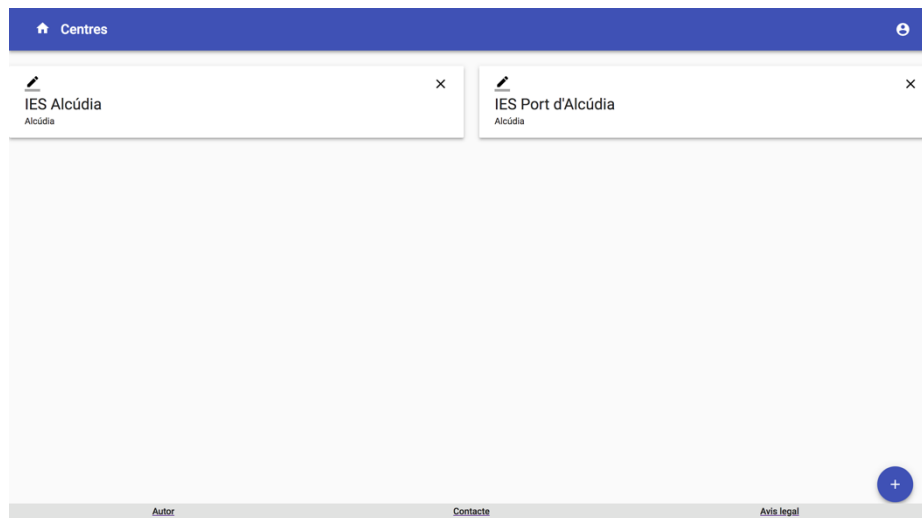


Figura 23: Pàgina de gestió de centres acadèmics

Quan es selecciona un centre acadèmic, es poden gestionar els anys acadèmics d'aquell centre i els grups que cada curs del centre té disponibles (cal recordar que un curs escolar pot estar compost per diferents grups, per exemple, Primer ESO A, Primer ESO B ...). La Figura 24 mostra la pàgina de gestió d'anys acadèmics i grups d'un centre seleccionat.

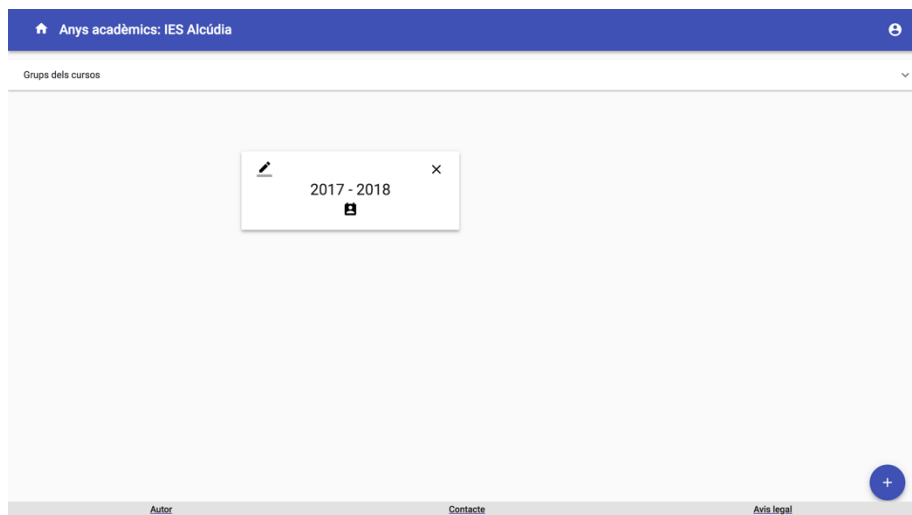


Figura 24: Pàgina de gestió d'anys acadèmics i grups de cursos d'un centre

Respecte a la creació de grups, si un curs no disposa de grups associats es crearà un grup per defecte per tal de poder-hi matricular alumnes, ja que es pot donar el cas de que, en alguns centres, els cursos acadèmics no estiguin dividits en diferents grups escolars.

Quan es selecciona un any acadèmic es llisten totes les assignatures impartides per un professor (en un centre i any acadèmic determinat). La figura 25 mostra un clar exemple d'aquesta pàgina.

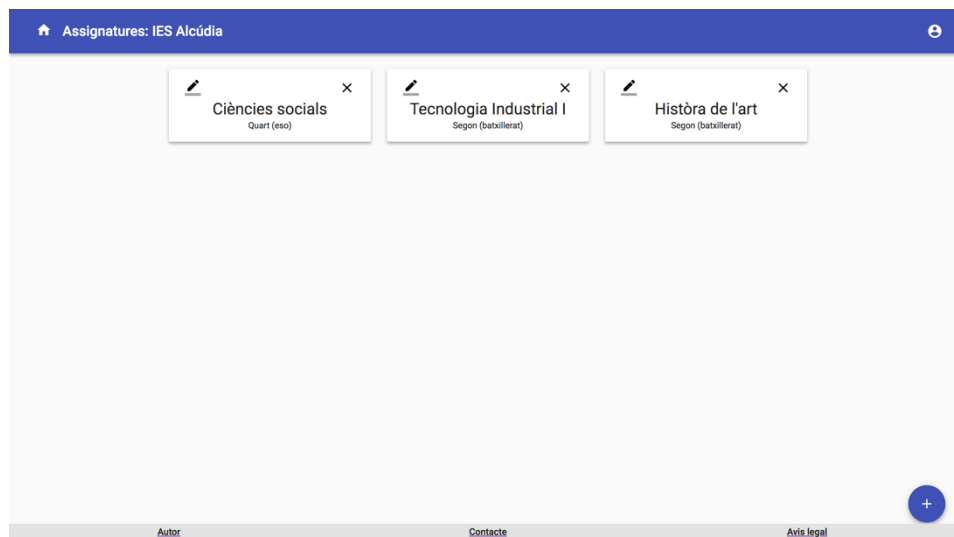


Figura 25: Pàgina que llista totes les assignatures impartides.

A l'hora de crear una assignatura, a aquesta se li poden associar semestres o períodes acadèmics. D'aquesta manera es poden classificar les activitats dins períodes d'avaluació. Ara be, en el cas de que no s'associïn semestres, es crearà un per defecte.

Quan es selecciona una assignatura, s'accedeix a la pàgina que permet gestionar les activitats i els alumnes matriculats a l'assignatura. Com es pot observar a la Figura 26, hi ha dues pestanyes. La primera pestanya s'encarrega del registre i modificació de les activitats. Les activitats van associades a un trimestre o període d'avaluació. El trimestre es selecciona a través del selector situat a la part superior-esquerra de la pantalla.

A l'hora de crear activitats, aquestes es poden configurar com a avaluables o no avaluables. Les activitats avaluables tenen una ponderació dins l'assignatura i es tenen en compte en el càlcul de la mitjana ponderada. Les activitats no avaluables no tenen ponderació associada i no es tenen en compte en el càlcul de la mitjana ponderada.

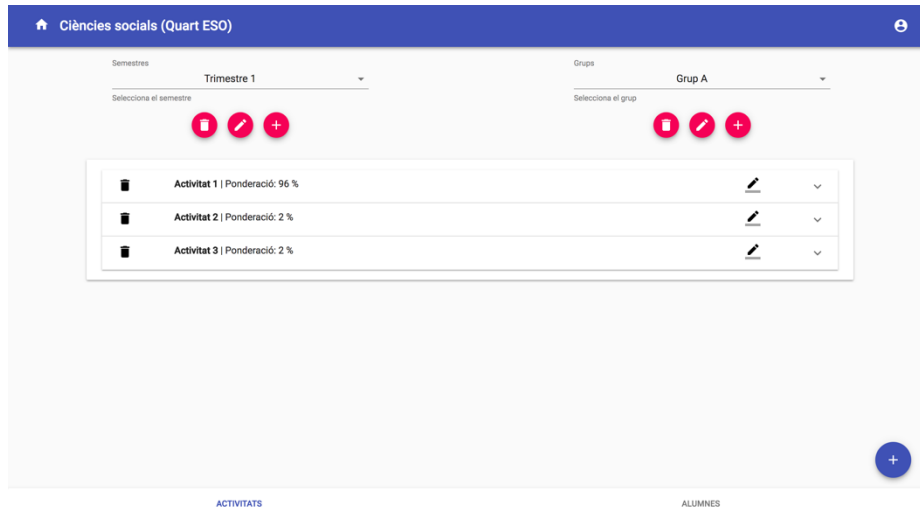


Figura 26: Pàgina de gestió de les activitats.

Cada una de les activitats es poden modificar, eliminar o qualificar. Per tal de poder-la qualificar basta clicar sobre l'activitat i es llisten (en forma de desplegable) els alumnes matriculats a l'assignatura per tal de poder-los qualificar. La Figura 27 mostra com es qualifiquen els alumnes.

Alumne	Qualificacio
Albert Pou Quirós	7 A
Francesc Sastre Cabot	2 S
Josep Lluís Miró Coll	6 A
Joan Font Rosillo	5 A
Lorenzo Seguí Capllonch	4 S
fnsdfn sdnvos dvdsov	1,7 S

Figura 27: Qualificacions d'una activitat obtingudes pels alumnes.

Per a cada un dels alumnes, hi ha associat un camp de text que permet inserta o modificar la qualificació obtinguda per l'alumne a l'activitat.

La segona pestanya s'encarrega del registre i modificació dels alumnes. Els alumnes van associats a un grup. Els grups es seleccionen a través del selector situat a la part superior-dreta de la pantalla. La Figura 28 mostra el llistat d'alumnes quan es selecciona la pestanya d'alumnes.

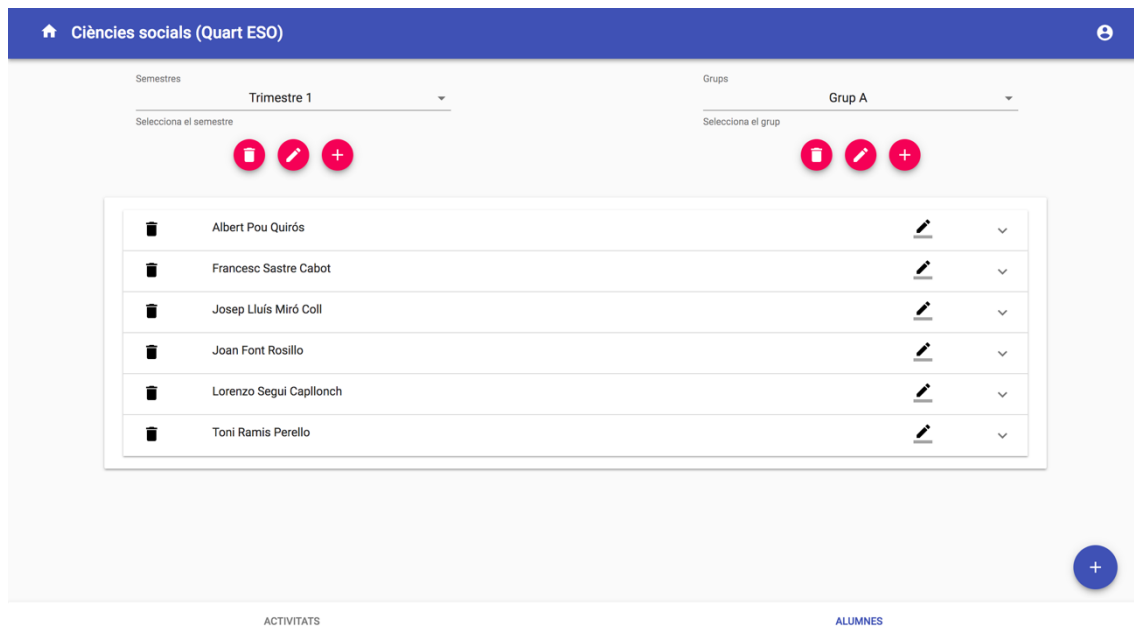
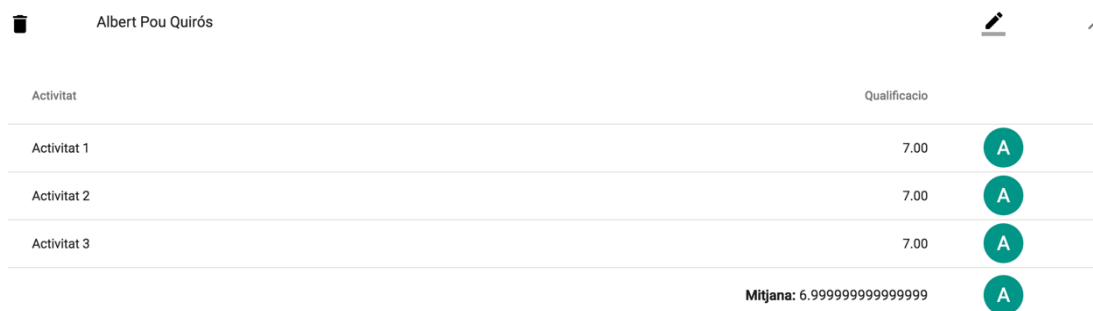


Figura 28: Pàgina de gestió dels alumnes.

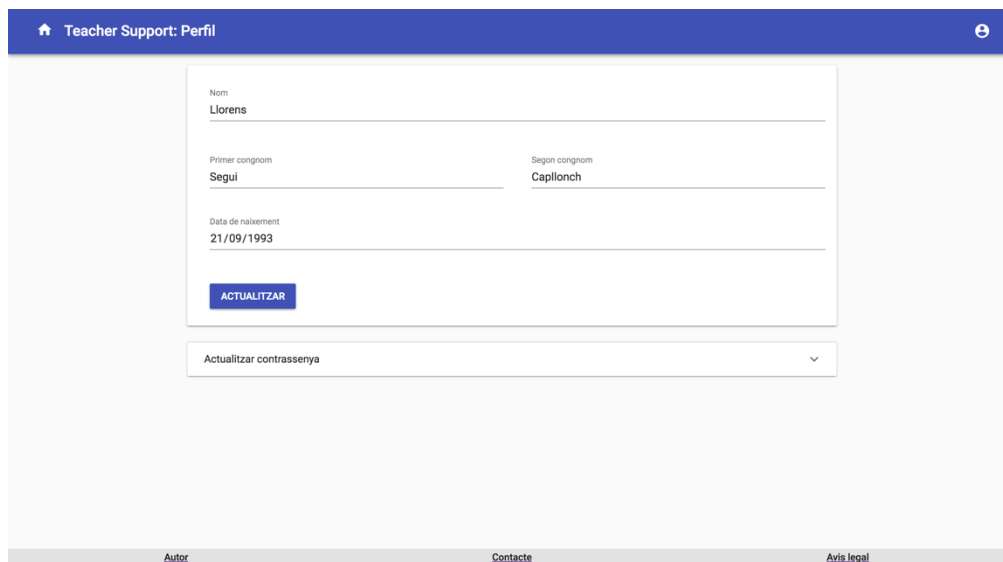
Cada un dels alumnes es pot modificar (icona llapis), eliminar (icona paperera) o consultar les seves qualificacions i mitjana. Per a dur a terme aquesta darrera acció basta clicar sobre l'alumne i es llistaran (en forma de desplegable) les qualificacions de les activitats i la seua corresponent mitjana ponderada (d'aquelles activitats avaluable). La Figura 29 mostra les qualificacions de les activitats realitzades per un alumne i la seua corresponent mitjana ponderada.



Activitat	Qualificació
Activitat 1	7.00 A
Activitat 2	7.00 A
Activitat 3	7.00 A
Mitjana: 6.999999999999999 A	

Figura 29: Qualificacions de cada una de les activitats d'un alumne i la seua mitjana ponderada.

Per una altre banda, els usuaris poden modificar la seu informació personal (associada al compte creat). Per això cal dirigit-se al menú situat a la part superior-dreta i seleccionar l'opció "El meu perfil". Una vegada seleccionada l'opció es presentarà un formulari que permet modificar la informació del compte (Figura 30).



Teacher Support: Perfil

Nom
Llorens

Primer cognom
Seguí

Segon cognom
Capllonch

Data de naixement
21/09/1993

ACTUALITZAR

Actualitzar contrassenya

Autor Contacte Avis legal

Figura 30: Formulari per modificar la informació personal de l'usuari.

Finalment, al footer (part inferior) de la web, s'han afegit enllaços que proporcionen informació d'autor, informació de contacte e informació legal.

Capítol 6: Conclusions i línies de futur

1. Conclusions

Els objectius plantejats a l'inici del projecte s'han assolit en general, amb un grau de satisfacció força elevat del procés i els resultats. El projecte ha permès experimentar amb noves tecnologies i metodologies de desenvolupament en les que es tenia poca o cap experiència i ha resultat un procés molt enriquidor a nivell personal.

El desenvolupament d'aplicacions web *Front-end* com a **SPA**, completament auto-contingudes, desenvolupades com a aplicacions complertes, i que només es comuniquin amb el servidor (**API REST**) per recuperar les dades és un trencament important respecte a la metodologia tradicional de desenvolupament d'aplicacions web basades en pàgines. Tot i que no és la solució més adequada per a totes les situacions, els avantatges que aporta aquesta arquitectura en les situacions en les que si és adequada són molt importants.

El fet de desenvolupar una aplicació web (*Front-end*) *responsive* emprant la llibreria ReactJS i integrar-la amb tecnologies i eines com ECMAScript6, Webpack i *Babel* ha permès entrar amb contacte amb aplicacions basades amb components web, els quals representen estàndards molt punters a l'àmbit del desenvolupament web. A sobre, desenvolupar el disseny seguin els patrons i guies d'estils establerts per Material Design han donat lloc a un producte usable i que es pot accedir des de qualsevol dispositiu.

A la vegada, desenvolupar una **API REST** (*Back-end*) emprant *Python* ha permès explorar unes tècniques, principis i metodologies completament noves ja que l'experiència prèvia havia estat Java, Node JS i .NET.

Respecta a la planificació i metodologia de treball seguida al llarg del projecte, durant la primera entrega es van haver de re-planificar algunes tasques per tal d'evitar que el procés d'aprenentatge i experimentació amb noves tecnologies frenessin el desenvolupament de l'aplicació. D'aquesta manera, es van poder evitar futurs riscos i problemes durant l'etapa de desenvolupament.

2. Línies de futur

2.1 Noves funcionalitats

Per tal de poder potenciar les característiques de l'aplicació, es podrien implementar les següents funcionalitats:

- Possibilitat de enregistrar l'adreça de correu electrònic dels alumnes matriculats per tal de que puguin ser notificats cada vegada que es publiquen les qualificacions finals d'una assignatura o cada vegada que es crea una nova activitat.
- Possibilitat d'enregistrar les adreces de correu dels pares dels alumnes per tal de que aquests puguin estar al corrent del procés acadèmic dels seus respectius fills.
- Possibilitat de poder exportar i traslladar els alumnes matriculats a una assignatura a assignatures d'altres anys acadèmics. D'aquesta manera, el procés de matriculació dut a terme pels docents seria més àgil.
- Integrar l'aplicació amb **APIs** públiques com Telegram o Facebook Messenger per tal de que els alumnes rebin notificacions a través d'aquestes aplicacions de missatgeria.

2.2 Aprofitament de la capa d'accés a les dades

Com ja s'ha explicat en els capítols anteriors, l'arquitectura d'aquesta aplicació segueix el patró client-servidor. A través d'aquest patró, s'ha pogut separar la capa de tractament i accés a dades de la capa visual. Aquest tipus de separació té l'avantatge de que es pot accedir a la informació des de qualsevol tipus de client o *software*. És per això que, en aquest cas, es podria reaprofitar l'**API REST** per tal de poder desenvolupar un altre tipus d'aplicació amb les mateixes característiques. Exemples d'altres tipus d'aplicació podrien ser les aplicacions mòbils o les aplicacions d'escriptori.

Bibliografia

- [1] Viquipèdia, "Concepte REST", https://en.wikipedia.org/wiki/Representational_state_transfer
- [2] Viquipèdia, "Disseny *Responsive*", https://en.wikipedia.org/wiki/Responsive_web_design
- [3] Viquipèdia, "Experiència d'usuari", https://en.wikipedia.org/wiki/User_experience
- [4] Viquipèdia, "Metodologia iterativa e incremental",
https://en.wikipedia.org/wiki/Iterative_and_incremental_development
- [5] Viquipèdia, "Revolució Digital", https://es.wikipedia.org/wiki/Revoluci%C3%B3n_Digital
- [6] Viquipèdia, "Microblogging", <https://es.wikipedia.org/wiki/Microblogging>
- [7] Viquipèdia, "Arquitectura per capes", https://en.wikipedia.org/wiki/Multitier_architecture
- [8] Portal Visual paradigma, "UML",
<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/>
- [9] Fòrum Inboundcycle, "Arbre de navegació",
<https://www.inboundcycle.com/blog-de-inbound-marketing/bid/195257/el-rbol-web-qu-es-y-para-qu-sirve>
- [10] Mozilla Developers, "HTML", <https://developer.mozilla.org/es/docs/Web/HTML>
- [11] Viquipèdia, "JavaScript", <https://en.wikipedia.org/wiki/JavaScript>
- [12] Viquipèdia, "ECMAScript", <https://en.wikipedia.org/wiki/ECMAScript>
- [13] Viquipèdia, "CSS" https://en.wikipedia.org/wiki/Cascading_Style_Sheets
- [14] World Wide Web Consortium, <https://www.w3.org/>
- [15] Material Design, "Patrons de disseny", <https://material.io/design/>
- [16] Node JS, "Documentació", <https://nodejs.org/en/about/>
- [17] Viquipèdia, "NPM" [https://en.wikipedia.org/wiki/Npm_\(software\)](https://en.wikipedia.org/wiki/Npm_(software))
- [18] ReactJS, "Documentació", <https://reactjs.org/>
- [19] Material UI Next, "Documentació", <https://material-ui.com/>
- [20] Webpack, "Documentació", <https://webpack.js.org/>
- [21] Babel, "Documentació", <https://babeljs.io/learn-es2015/>
- [22] Python, "Documentació", <https://www.python.org/>
- [23] Django REST Framework, "Documentació" <http://www.django-rest-framework.org/>
- [24] Viquipèdia, "ORM", https://en.wikipedia.org/wiki/Object-relational_mapping
- [25] SQLite, "Documentació", <https://www.sqlite.org/about.html>
- [26] Viquipèdia, "JSON", <https://en.wikipedia.org/wiki/JSON>
- [27] Viquipèdia, "JWT", https://en.wikipedia.org/wiki/JSON_Web_Token
- [28] Viquipèdia, "Heroku", <https://en.wikipedia.org/wiki/Heroku>
- [29] GIT, "Sobre GIT", <https://git-scm.com/about>
- [30] Viquipèdia, "GitHub", <https://en.wikipedia.org/wiki/GitHub>
- [31] Visual Studio Code, "Documentació" <https://code.visualstudio.com/docs>
- [32] Material Design, "Guia d'estils", <https://material.io/guidelines/>
- [33] Google Fonts, "Tipografia Roboto", <https://fonts.google.com/specimen/Roboto>

Annexos

Annex A: Glossari

TFM: Treball Final de Màster
API: Application Programming Interface
REST: Representational State Transfer
UOC: Universitat Oberta de Catalunya
TIC: Tecnologies de la Informació i Comunicació
HTTP: Hypertext Transfer Protocol
JSON: JavaScript Object Notation
UML: Unified Modeling Language
HTML: HyperText Markup Language
XML: eXtensible Markup Language
CSS: Cascading StyleSheets
XHTML: eXtensible HyperText Markup Language
W3C: World Wide Web Consortium
NPM: Node Package Manager
MVC: Model View Controller
ORM: Object Relational Mapping
ACID: Atomicity, Consistency, Isolation and Durability
JWT: JSON Web Token
UI: User Interface
URL: Uniform Resource Locator
SPA: Single Page Application
RFC: Request For Comments

Annex B: Elements lliurats

Arxiu Isc025_Manual_instalar_API.pdf: Aquest arxiu es correspon al manual de instal·lació de l'API REST en un entorn local.

Arxiu Isc025_Manual_instalar_Client_web.pdf: Aquest arxiu és correspon al manual de instal·lació del client web en un entorn local.

Arxiu Isc025_Presentacio_Publica.pptx: Presentació del projecte amb un conjunt de transparències.

Arxiu Isc025_Presentacio_Academica.mp4: Presentació (en vídeo) del projecte dirigida cap al tribunal d'avaluació.