
This is the **published version** of the bachelor thesis:

Salvador Abad, Àlex; Otazu Porter, Xavier, dir. Portal web per al suport a la gestió de contractació de professorat extern a la UAB. 2021. (958 Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/248462>

under the terms of the  license

Portal web pel suport a la gestió de contractació de professorat extern a la UAB

Alex Salvador Abad

Resum — Davant l'imparable avenç tecnològic que vivim les darreres dècades, són molts els processos de gestió que tradicionalment es realitzaven de manera manual, i actualment es recolzen o han migrat completament a sistemes informàtics que en faciliten la gestió. En el cas particular d'aquest projecte, es proposa desenvolupar una solució *software on demand*, la qual consisteix en un portal web de publicació i gestió de candidatures laborals, que ha de funcionar com a eina de suport per a la gestió de la contractació de professorat extern a la UAB, agilitzant i facilitant així el procés manual actual i adaptant-se a les necessitats del Departament de Ciències de la Computació. És interessant ressaltar que, al tractar-se d'un projecte *software* desenvolupat únicament per una persona, hi ha hagut tasques de diferents àmbits (*disseny, front-end, back-end, testing / QA...*) que han estat coordinades utilitzant metodologies de *software* àgils. En aquest document es detallen les activitats més destacables que s'han realitzat en el cicle de vida del projecte, comentant els canvis ocorreguts, així com els plans de contingència que s'han aplicat per atacar-los. Com a referències més destacables del projecte, es pot destacar la llibreria externa *jquery_week_calendar*, (pública i *open-source*) de la qual se n'ha aprofitat una part d'implementació *front-end*, decisió que va ser presa després d'una recerca i una posterior anàlisi sobre llibreries o *frameworks* actuals relacionats amb horaris o calendaris per tal de recolzar el desenvolupament base del calendari i no "*reinventar la roda*".

Paraules clau— *On demand*: Ajustat a les necessitats (requeriments) del client. - *QA*: Assegurament de la qualitat del software, prevenció i detecció d'errors. - *Stakeholders*: Persones implicades al projecte. - *Waterfall*: Execució de les activitats del projecte en fases seqüencials lineals. - *Open-source*: Codi obert a tothom d'ús gratuït. - *MVP*: Producte mínim viable per a la sortida.

Abstract— Given the unstoppable technological advancement we have experienced in recent decades, many management processes have traditionally been done manually, and are now supported or have been completely migrated to computer systems that facilitate their management. In the particular case of this project, it is proposed to develop a software with *on demand* solution, which consists of a web portal for publishing and managing job applications, which must important function is to be a support tool for managing the hiring of external faculty at the UAB, streamlining and facilitating the current manual process and adapting it to the needs of the Department of Computer Science. It is interesting to note that given the fact that this software project's been developed only by one person, as a result there have been many tasks in different areas (design, front-end, back-end, testing / QA ...) that have been coordinated using agile software methodologies (*Agile*). This document details the most notable activities that have been carried out in the life cycle of the project, commenting on the changes that have taken place, as well as the contingency plans that have been applied to attack them. The most notable references of the project are the external library *jquery_week_calendar*, (public and *open-source*) which has taken advantage of a part of front-end implementation, a decision that was taken after a search and a subsequent analysis of current libraries or frameworks related to schedules or calendars in order to support the basic development of the calendar and not "*reinvent the wheel*".



1 INTRODUCCIÓ - CONTEXT DEL TREBALL

En primera instància, a mode introductor i amb la finalitat de presentar els conceptes més determinants per l'enteniement de les motivacions i objectius del projecte, cal definir quin és el context del problema a resoldre en la fase inicial, prèvia a l'inici del desenvolupament del projecte.

Actualment, el funcionament del procés de contractació de professorat extern al Departament de Ciències de la Computació comporta una feina tediosa, repetitiva i altament dependent de la comunicació bidireccional entre candidats i la persona qui gestiona la contractació anual, la qual ha d'adreçar-se individualment a cada candidat/a i consultar-ne preferències i/o experiència/currículum... alhora que tracta de coordinar l'horari dels professors fixes amb l'objectiu de trobar la combinació ideal de professors, franges i disponibilitat, per així abastir de professorat els horaris de les assignatures del domini del departament.

Arrel d'aquesta situació en què, en primer lloc, la comunicació individualitzada amb tots els candidats/tes suposa un factor bloquejant i difícil de controlar en nombroses

ocasions; en segon lloc, tampoc no es disposa d'una eina de suport tecnològica per aquest procés, més enllà d'un full de càlcul amb informació; i com a tercer factor, un dels responsables de la contractació de professorat anual ha aixecat la mà per tal de buscar solucions exercint de tutor d'aquest TFG; conseqüentment, ha sorgit la motivació de crear un software de suport a la contractació de professorat extern pel Departament de Ciències de la Computació (aplicable a departaments similars).

Després de les reunions inicials entre *Stakeholders*, es varen definir els següents objectius principals, que en el seu acompliment resulten en satisfer la motivació inicial del projecte:

- *Objectiu 1: Entendre i conèixer en profunditat el domini del problema a resoldre així com les casuístiques que cal tenir en compte, a fi de que el disseny i la implementació del software posterior sigui adequada a les necessitats del client.*

Aquest primer objectiu es defineix en base a les necessitats inicials del projecte, especialment a causa dels dubtes i el desconeixement envers la forma i característiques de la solució *software* necessària. A més a més, la poca informació respecte projectes similars en la recerca de l'estat de l'art, evidencia la necessitat d'abordar aquest objectiu separatament, ja que el domini del problema té les seves casuístiques pròpies, i cap *Stakeholder* tenia inicialment un coneixement profund i complet respecte les característiques de la solució final. Aquest objectiu per tant, està estretament lligat a la fase d'anàlisi de requeriments en metodologia *Agile*. [1]

- *Objectiu 2: Disseny i implementació d'un back-end que ha de relacionar la lògica i les dades conegudes sobre domini del problema (horaris, ofertes, usuaris...) per tal d'obtenir les respostes esperades per part del servidor.*

Aquest segon objectiu conté en si mateix dos àmbits de treball diferents, el disseny i la implementació. A la banda de disseny del *back-end* s'ha treballat la lògica de la base de dades, l'arquitectura MVC [2] (model-vista-controlador) i els contractes de dades amb *front-end*; mentre que a la banda de la implementació *back-end*, majoritàriament s'ha treballat en les funcions del model i la lògica dels controladors, així com en d'altres implementacions com l'emmagatzematge de currículums o el *parseig* de dades provinents de *front-end*.

- *Objectiu 3: Disseny i implementació del front-end web que consta d'un horari setmanal editable, scripts que en gestionen la seva informació, i totes les altres pàgines necessàries per navegar entre els casos d'ús existents.*

Com l'anterior objectiu, l'objectiu de *front-end* agrupa disseny i implementació en si mateix, dos passos contigus en iteracions àgils. Destaca a la part de disseny, la creació de les vistes, buscant la generació de valor a través d'obtenir una bona experiència d'usuari (UX). D'altra banda, el conjunt de tasques més significatiu per l'acompliment de la part d'implementació *front-end* ha estat en la creació dels scripts per gestionar la informació dels horaris així com la invocació d'aquests horaris.

- *Objectiu 4: Integració dels diferents mòduls en un portal web, connectat a una base de dades, que ha de relacionar l'implementació back-end i front-end.*

Per aquest objectiu, inicialment s'ha treballat en la gestió de la configuració, integrant servidor, client de la base de dades, i l'IDE de desenvolupament (*PhpStorm* [3], en aquest cas), i posteriorment ha continuat vigent especialment en l'acompliment de contractes de dades entre *back-end*, *front-end* i la base de dades, assegurant així la integritat de les dades i de l'entorn web.

2 PLANTEJAMENT I DEFINICIÓ DEL PROBLEMA

Durant tot el projecte, però amb més èmfasi en les seves fases primerenques, s'ha revisat i fins i tot, en alguna ocasió, s'ha redefinit el problema a solucionar i les seves característiques, ja que al tractar-se d'una solució *on demand*, els requeriments sorgeixen de les necessitats de la part interessada i un dels objectius principals del projecte ha estat vinculat al coneixement del domini del problema.

Primerament, a la fase inicial del projecte, la definició del problema mancava d'alguns detalls, com per exemple el fet que un professor pot tenir *n* assignatures i no només 1, i a més a més tenia l'enfoc estructural consistent en que les "ofertes de docència" es publicarien 1 a 1, manualment, o llegint-se d'un arxiu, seguint el següent diagrama de flux:

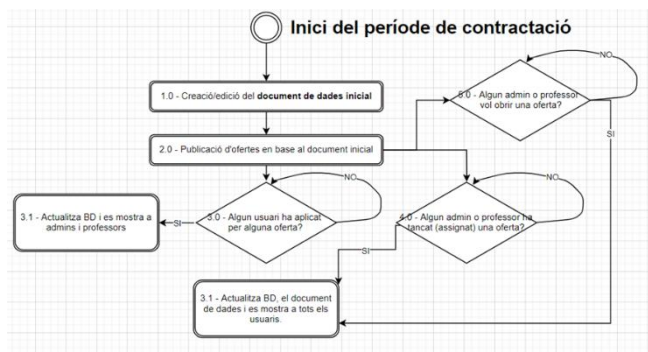


Figura 1. Diagrama definició del problema fase I

En una fase posterior d'anàlisi, es va haver de modificar la concepció prèviament esmentada sobre el flux de la solució del problema a un model amb un enfocament diferent, que elimina les tasques més repetitives al gestor/a que utilitza l'eina. Això es deu al fet que aquesta nova versió substitueix l'assignació manual d'ofertes de tipus *formulari* per la creació d'horaris setmanals de cada assignatura que es publica. En aquesta nova versió l'administrador pot afegir les franges de classe on es necessiten docents i els usuaris/es de la pàgina poden veure i inscriure's a ofertes també a través d'un horari similar al que veuen els estudiants. Veiem a continuació un altre diagrama on s'il·lustra el canvi, eliminant així la creació manual d'ofertes 1 a 1 i la necessitat d'un arxiu auxiliar.

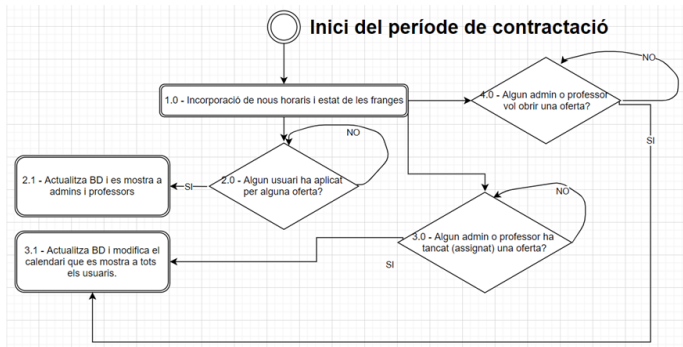


Figura 2. Diagrama definició del problema fase II

- E-mail de contacte: alexsalvadorabad@gmail.com
- Menció realitzada: Enginyeria del Software
- Treball tutoritzat per: Xavier Otazu (Computació)
- Curs 2020/21

En fases posteriors hi ha hagut lleus modificacions sobre el coneixement en el domini del problema, que es documenten al llistat de canvis, ja que no afecten de manera estructural a la manera d'entendre el problema, sinó que han estat causa d'algun requisit poc definit.

3 ESTAT DE L'ART

En aquesta secció es relata el resultat de la recerca sobre quines són les tecnologies i projectes existents a l'actualitat, que guarden similituds amb el projecte desenvolupat, a fi de poder realitzar les comparacions pertinents un cop obtingut el resultat final, i en conseqüència extreure'n conclusions vàlides.

Un cop realitzada la recerca i analitzant-ne els resultats, cal destacar que, sorprenentment, no s'ha aconseguit trobar un portal web que compleixi o s'adeqüi a les necessitats del client d'aquest projecte, ja que tot i haver-hi una immensa quantitat de portals de suport a la contractació com: *LinkedIn*, *Infojobs*, *SocialYou*, *Trovit...* [4], no s'ha trobat cap d'ells que combini les ofertes amb els horaris setmanals, la qual cosa amb força seguretat es deu a la particularitat del cas que es tracta. Tot i això, si que s'han trobat pàgines que compleixen o són molt similars a les característiques de l'horari interactuable desitjat, com en el cas de *Doodle* [5] o el calendari de *Microsoft*, i per tant, s'observa que tant la part de contractació com la de creació d'horaris funcionen amb èxit actualment, però no s'ha trobat un projecte que integri ambdues parts.

4 DEFINICIÓ DE LA METODOLOGIA

Per a un lògic i correcte desenvolupament del projecte, inicialment s'ha definit la metodologia que s'ha utilitzat durant tot el cicle de vida d'aquest.

D'acord amb experiències passades en altres projectes de la carrera, i pel funcionament que he observat també a títol personal en l'àmbit laboral, vaig decidir utilitzar metodologia *Agile*, segmentant el treball en tasques en forma de tiquet, i iterant la realització d'aquestes en *sprints* de 2 setmanes, els quals han de tenir entre 1 i 3 objectius principals de l'*sprint*. La gestió dels tiquets al *board* (pis-sarra) s'ha realitzat utilitzant *Trello*, una eina de suport per a la gestió de projectes en entorn web.



Figura 3. Exemple de tiquets completats a Trello

El plantejament àgil descrit anteriorment ha sigut l'emprat durant tot el projecte, a excepció de les primeres 4 setmanes, on la intenció inicial era treballar aplicant dita metodologia, però la falta d'informació sobre el domini del problema i les característiques de la solució van impedir iterar, ja que no era possible implementar solucions quan el domini encara era confús i indefinit, de manera que es va

realitzar una primera fase d'anàlisi de requeriments, iniciant per tant el projecte com si s'apliqués un model *waterfall*.

D'altra banda, sota l'enfoc del control de versions i *back-ups*, s'ha seguit la metodologia de *git flow* aplicant *CI* (*continuous integration*) [6], ja que al haver-hi un únic desenvolupador exercint de *full-stack*, és una manera còmoda i senzilla de gestionar el control de versions en un repositori, que en el cas d'aquest projecte s'ha generat local i remotament a <https://github.com/1492155/WebProjecteTFG>.

5 ANÀLISI DELS REQUERIMENTS

L'anàlisi de requeriments d'aquest projecte s'ha dividit en dues fases on s'ha treballat de manera diferent.

La primera fase, duta a terme les primeres 3-4 setmanes amb metodologia *waterfall*, consta de la creació inicial de diagrames i documentació de requisits a partir de la informació recollida i analitzada provinent d'entrevistes amb l'*Stakeholder* que posseeix el coneixement del domini, en aquest cas el tutor del TFG, amb la finalitat d'obtenir una base sobre el funcionament esperat, sobre la qual iterar posteriorment.

La segona fase ha estat la part d'anàlisi de requeriments implícita en les iteracions *Agile*, en la que s'ha generat el *Definition of Ready* (*DoR*) de cada tasca a realitzar abans d'implementar-la, minimitzant així la improvisació i clarificant els Criteris d'Acceptació (*AC*) amb que es testegen les històries.

5.1 Captació de requisits

Per a la captació de requisits s'ha emprat com a tècnica principal la realització d'entrevistes periòdiques amb el client, en aquest cas el tutor, en les quals es revisaven els requisits i s'introduïen canvis en cas de ser necessari, ja que el *software* desenvolupat té com a objectiu satisfer les necessitats de qui el demana (*on demand*), però tot i això, també s'han captat altres requisits a través de la recerca i l'anàlisi del domini del problema.

5.2 Elaboració de diagrames

A la primera fase d'anàlisi de requeriments, amb l'objectiu d'il·lustrar i esquematitzar la informació coneguda i acotar el funcionament de la solució a implementar, es varen crear diferents diagrames representatius del sistema. A continuació es comenten els aspectes més destacables d'aquests

- **Diagrames de flux:** S'han dissenyat diferents versions que representen el flux bàsic (*happy path*) de l'aplicatiu en els casos d'ús relacionats amb la gestió d'ofertes, a fi d'observar si el funcionament que s'il·lustra aconsegueix millorar la problemàtica inicial, analitzant si el flux és lògic i útil per tots els usuaris i Stakeholders.

- **Diagrama de classes:** S'ha dissenyat un diagrama de classes que ha estat la base de disseny per a la posterior creació de la base de dades (Annex A3). Cal destacar que l'objectiu principal d'aquest diagrama ha estat el de relacionar els conceptes d'horari, oferta, franja i assignatura; per tal d'aplicar la lògica de les seves relacions i atributs a la base de dades.
- **Diagrama de casos d'ús:** El diagrama de casos d'ús pretén il·lustrar quines són les accions i escenaris que poden succeir dins l'aplicació web desenvolupada, per cadascun dels diferents perfils d'usuari que s'han contemplat (admin, professor i usuari). Tot i que el diagrama de casos d'ús general de l'aplicació ha patit canvis desde la seva versió inicial (Annex A2), ha estat una referència molt útil durant el temps de vida del projecte, per tal de guanyar un punt de vista general i prendre millors decisions de disseny del *software* posteriorment.

5.3 Document SRS (Software Requirements Specification)

S'ha documentat una bona part de la informació sobre els requeriments del projecte en un document tipus SRS. Aquest document fa un recull de tota la informació: requisits, prerequisits, casos d'ús, propòsit i informació d'interès per al desenvolupament del nostre sistema. En aquest document es troba recopilada tota la informació necessària per a poder tenir una idea general del projecte de software a desenvolupar. El propòsit d'aquest document és poder definir perfectament el comportament de l'aplicació a realitzar i del sistema a desenvolupar. Més concretament, aquest informe recull els requisits de l'aplicatiu a desenvolupar i les funcionalitats necessàries per a poder dur a terme el disseny del sistema.

A la següent figura en veiem un exemple:

Id. Requisit	REQ-F-2
Títol	Formulari de registre comprovació correu electrònic
Descripció	El sistema ha de mostrar un missatge d'error informatiu quan l'usuari ha intentat registrar-se amb un correu electrònic amb format invàlid.
Cas d'ús relacionat	Registre d'usuari
Prioritat	Màxima (3)

Figura 4. Exemple de requisit funcional del SRS

Com podem observar, al document SRS es mostren les relacions entre els requisits funcionals de l'aplicació i els casos d'ús que podem trobar al diagrama. També es mostra la prioritat que s'ha establert per cada requisit respecte la

sortida del projecte, on *màxima* (figura 4) equival a essencial per a la sortida i per tant aquest requisit queda dins l'MVP.

6 GESTIÓ DE LA CONFIGURACIÓ

En paral·lel amb la primera fase d'anàlisi de requeriments, s'ha realitzat una tasca de recerca, així com la seva posterior anàlisi, amb l'objectiu de triar quines eines cal utilitzar per desenvolupar la solució *software*.

Les eines triades després d'analitzar les diferents possibilitats per crear el portal web són:

- En la part de gestió de la configuració, per estandarització i coneixement previ de l'eina, s'utilitza gitBash [7], un emulador de consola Git de Linux en Windows, el qual permet gestionar el repositori de treball i les seves diferents versions.
- Com a IDE, s'utilitza PHPStorm, ja que està orientat a programar en PHP, el llenguatge principal que s'utilitza en la part del servidor d'aquest projecte, alhora que permet utilitzar SQL amb la llibreria PDO [8] i *mysqli* [9].
- Com a client per a la integració de la base de dades, el servidor, i l'entorn de programació, s'utilitza XAMPP [10], un paquet que inclou el servidor *open-source* Apache, el client de base de dades PHPMyAdmin [11], així com d'altres eines útils per desenvolupadors web.

7 GESTIÓ DEL DESENVOLUPAMENT

En aquesta part del document s'esmenten les fites més destacables en els diferents àmbits implicats en el desenvolupament d'aquest projecte, a fi d'aprofundir en algunes de les tasques més importants que s'han dut a terme en aquest treball de fi de grau.

7.1 Disseny del software

Pel que fa a l'enfocament de disseny del software, es diferencien 4 grans blocs abordats: el disseny del mapa de pantalles, el disseny de la base de dades, el disseny individual de cada vista i la utilització de patrons coneguts de disseny del *software* per treballar en la qualitat del codi (QA). A continuació s'esmenten els comentaris relatius a aquest conjunt de tasques de disseny.

- **Disseny del mapa de pantalles:** Cronològicament, es tracta de la primera de les tasques de disseny que s'ha portat a la pràctica. La lògica d'abordar aquesta tasca en primera instància respecte les demés tasques de disseny, rau en l'obtenció d'un primer model lògic, que permet generar una imatge fidel sobre quin és el comportament esperat, alhora que es relaciona amb el diagrama de casos d'ús, assegurant per tant la coherència en el disseny i un primer enfoc en el

funcionament previst de l'aplicació. Cal destacar que aquesta tasca ha sigut de gran utilitat a l'hora d'implementar a posteriori el patró d'arquitectura MVC, ja que el mapa de pantalles és fàcilment transformable a "index" i "recursos", la qual cosa permet establir còmodament la base d'aquest patró de programació.

- **Disseny individualitzat de les vistes:**
En el disseny individual de cada vista de la pàgina, les tasques més costoses han estat: el disseny de les pàgines de vista, modificació i creació d'horaris, i el disseny del menú principal i els llistats d'informació.
- **Disseny de la base de dades:**
Donat que s'ha aplicat metodologia Agile en la realització d'aquest projecte, s'han prioritzat canvis en diverses ocasions, i per tant el disseny de la base de dades s'ha vist modificat en algunes iteracions, en les quals les implicacions d'algun canvi aplicat tenen afectació en la lògica de la base de dades. Actualment la vista de dissenyador de la base de dades correspon a la següent figura:

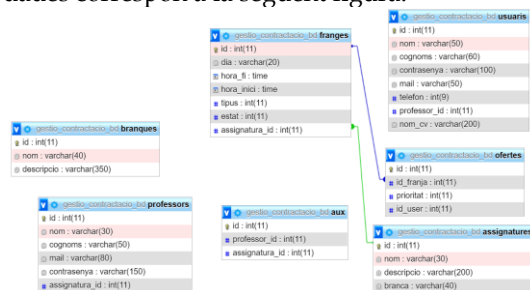


Figura 5. Base de dades vista de dissenyador fase final.

Destaquen les relacions entre ofertes, franges i assignatures, així com la taula auxiliar "aux" que permet la relació N-N entre professors i assignatures (modificació aplicada al pla de contingència del canvi A13 de l'apèndix).

- **Utilització de patrons de disseny del software:**
Tot i que no ha estat contemplat com a tasques unitàries a Trello [12], s'ha procurat que en el desenvolupament de la solució s'apliquessin els patrons de disseny oportuns. En la solució final destaquen els 2 patrons següents:
 - **MVC (Model-Vista-Controlador):** Per la experiència en projectes anteriors amb aquest patró, i el seu ús en tecnologies actuals s'ha utilitzat aquest patró d'arquitectura des del primer moment.
 - **State:** S'ha aplicat el patró de disseny de software State al disseny del header de la pàgina, ja que en aquesta pàgina s'hi encapsulen

4 possibles estats, que corresponen als diferents perfils d'usuari possibles i depenen d'una variable de sessió que es controla des de back-end.

7.2 Back-end

A la part del servidor i el tractament de les dades rebudes pel client, el back-end, els 5 conjunts de tasques més destacables durant el desenvolupament han estat:

- **Definició de la lògica de sessions:**
Per al correcte funcionament de l'aplicació, el servidor necessita emmagatzemar certes informacions en variables que anomenem sessions. El control de l'ús d'aquestes variables ha estat fonamental, ja que s'han definit les quines són les sessions que s'utilitzen així com el significat dels valors que prenen, i s'han provat a la part de testing amb Data-Driven Testing [13], provant, no exclusivament, els punts conflictius dels valors de les sessions i comprovant-ne el funcionament.
- **Neteja i emmagatzematge de dades dels horaris:**
Per a guardar la informació d'un horari nou que conté vacants per als usuaris, cal parsejar (netejar, extreure'n contingut) una sèrie de dades que s'envien desde front-end i que es reben a back-end a través de certes cookies conegudes. Un cop realitzat aquest procés, es realitzen les crides a les funcions del model que actualitzen els valors de les sessions i insereixen les dades en el format adient a les taula oferta de la base de dades.
- **Gestió d'usuaris:**
D'acord al diagrama de casos d'ús del projecte, tenim 3 perfils d'usuari diferents:

- **Usuari:** Si no està loguejat només pot: Registrar-se, Loguejar, Veure el llistat de branques, Veure el llistat d'assignatures d'una branca, Veure horari d'una assignatura, Veure el "qui som"; si l'usuari ha loguejat a més a més pot realitzar el cas d'ús: Inscriure's a una assignatura i Fer log-out.
- **Professor:** Hereta els casos d'ús de l'usuari loguejat i també té el cas d'ús: Modificar assignatura, de l'assignatura que té assignada a la base de dades (s'assigna quan l'admin registra un professor).
- **Admin:** Hereta els casos d'ús de l'usuari loguejat i també té els casos d'ús: Afegir assignatura, Eliminar Assignatura, Gestionar candidats, Afegir/Eliminar professor i Afegir/Eliminar Branca.

A la lògica de back-end s'ha controlat mitjançant el patró de disseny del software State el valor d'una variable de la sessió, la qual realitza un toggle (activació/desactivació) de funcionalitats en funció

del seu valor, que representa el tipus d'usuari.

- **Funcions del model:**

Una part important del treball a la banda del servidor ha estat en el desenvolupament de les funcions del model, que es comuniquen amb la base de dades utilitzant la llibreria oberta *PDO* de PHP, que ens permet l'execució de *queries* (consultes) SQL. A la carpeta *model* es troben els 5 arxius on hi ha totes les funcions que es relacionen amb la base de dades. En cada arxiu hi trobem funcions que difereixen entre elles per la funcionalitat en que estan implicades (*users*, *offers*, *subjects*...).

Una de les parts destacables d'entre les diverses funcions desenvolupades, és la funció de *checkLogin()*, que no només comprova si les dades d'accés són correctes o no, sinó que també aplica a la lògica de sessions la informació sobre el tipus d'usuari que està loguejant.

Com a funcionalitat extra, a part de les consultes d'inserció, modificació o eliminació de la base de dades, a l'arxiu *users.php* de la carpeta del model, a la funció *saveCV()*, es realitza el guardat dels arxius amb extensió *.pdf* que envia l'usuari al sol licitar una vacant, realitzant les comprovacions necessàries respecte el tipus d'arxiu, amb la finalitat d'evitar un problema greu de seguretat deixant una entrada d'arxius totalment lliure.

- **Gestió dels candidats:**

Per tal de satisfer el cas d'ús de l'administrador "Veure candidatures", s'ha implementat una pàgina inicialment no prevista al mapa de pantalles en la qual l'usuari (admin) selecciona una assignatura d'entre les existents, i el servidor respon amb la informació dels candidats que hi han aplicat, les seves preferències i un enllaç al seu currículum. Pel desenvolupament d'aquesta part, s'han creat *queries* que troben la informació sobre l'assignatura desitjada, alhora que s'ha realitzat una tasca important de *parseig* de la informació obtinguda des de la base de dades.

7.2 Front-end

En l'apartat de treball a la part de front-end els conjunts de tasques més importants que s'han abordat durant el desenvolupament del projecte han estat els següents:

- **Invocació del calendari setmanal correctament formatat i inicialització de variables:**

Sense cap dubte, aquesta és la tasca que més gruix ha suposat en hores a la banda de front-end, per això es comenten a continuació certs aspectes que han sigut decidits envers la programació d'aquesta part, i que tenen una importància i prioritat alta respecte els requisits de l'aplicació. Per a la implementació dels scripts d'invocació, s'ha

aprofitat part d'un codi open-source de github que genera un calendari mensual i permet controlar els events de drag "arrossegat" i drop "deixar", utilitzant funcions de la llibreria pública de *jquery*, versió 1.8.11 [14].

En primer lloc, per a la invocació del calendari buit, s'executa la primera etapa de l'script que he anomenat "schedule.js". En aquesta etapa, es declaren els elements HTML principals identificats amb els seus respectius ID. En aquesta mateixa part d'inicialització, es realitza també una neteja de local storage i cookies, ja que ambdós elements són utilitzats posteriorment a l'hora de guardar informació sobre l'horari que crea l'usuari.

Seguidament, un cop realitzada l'inicialització general de l'estructura HTML desde l'script, es renderitzen les files de l'horari amb les hores representants de cada franja, definint a l'inici de l'script, dins la configuració, quina hora d'inici i fi volem. Dins cada franja, s'han definit franges de l'ordre d'un quart de grandària respecte l'original, representant els quarts d'hora i afegint també aquesta proporció franja-subfranges a la configuració de l'script.

Un cop renderitzat (o invocat) l'horari, es defineixen les variables de tipus de classe (teoria, problemes, pràctiques), assignades al botó corresponent, així com es defineix un tipus de classe "indefinit" i invàlid, per quan l'usuari no selecciona cap tipus de classe. En aquest punt, es va prendre la decisió de no posar un tipus de classe per defecte, ja que no estalviaria clics a l'usuari, sinó que pot generar confusió, i s'ha preferit mostrar una alerta informativa si no s'ha seleccionat el tipus.

Un cop l'usuari ha seleccionat el tipus de franja que desitja crear, quan clica sobre l'horari s'inicia la crida a *EventNew(...)*, funció que rep per paràmetre el tipus d'event i la informació de l'horari actual, i interpretant el recorregut del mouse quan l'usuari deixa de seleccionar setjeja un nou Event, cridant la funció *EventRender(...)*. En aquest procés de creació d'events, que es pot traduir a creació de franges, es guarda la informació a l'emmagatzematge local del navegador client, alhora que s'enregistren les dades utilitzant algunes cookies que es creen en el procés.

Quan el calendari s'està mostrant, però no per crear, sinó per veure (usuari), es desactiven (*toggle*) els events de creació, no utilitzant en aquest cas cookies per escriure, sinó que realitzant un procés de lectura respecte el servidor, utilitzant l'accés a elements del DOM ocults per l'usuari, on el servidor ha volcat la informació sobre l'horari d'aquesta assignatura concreta.

- **Establiment dels contractes de dades amb *back-end*:**

Una tasca de recerca i estrictament relacionada amb la integració del software de front-end (Javascript amb jQuery i AJAX) amb el software de back-end (HTML, CSS, PHP i MySQL), ha estat l'establiment dels contractes de dades entre ambdós parts, tot i que formalment s'ha contemplat com a tasca de front-end ja que la necessitat de realitzar aquesta tasca va aparèixer a l'hora de voler passar informació de l'usuari i el navegador al servidor, i no en el moment contrari.

Pels formularis ha estat senzill establir el contracte, ja que es realitzaven crides de tipus *POST*, en que els identificadors i el tipus de dada són prous per establir el contracte, però per a la part d'emmagatzematge dels horaris ha estat vital establir un contracte de dades, ja que les dades que arriben al servidor són dinàmiques i variants, i no pas estàtiques i constants com en el cas dels formularis.

- **Creació de l'script de guardar horari:**

Aquest conjunt de tasques ha estat abordat posteriorment respecte a l'anterior descrit en aquest apartat, ja que sense un contracte entre *front-end* i *back-end*, la solució probablement aportaria futurs impediments en el moment de dur-se a terme.

La creació de l'script de "guardar horari" ha estat la tasca que ha generat com a resultat l'script "saver.js", on s'implementa una crida asíncrona en AJAX, recollint les dades de les cookies s'han generat a l'script "schedule.js", les quals resulten d'interpretar els botons de tipologia, l'assignatura introduïda, hora d'inici i fi... alhora de crear un event, i finalment envia les dades al servidor en el format del contracte.

Destaca la creació d'aquest script ja que és un *milestone* necessari pel funcionament del cas d'ús "Guardar horari", fonamental per l'*MVP* del projecte.

7.2 QA i Testing

A continuació es detallen els aspectes més determinants per part del seguiment de QA durant el desenvolupament del projecte.

- **TDD - Test-Driven Development:** Tot i que en la fase més inicial del projecte, la intenció va ésser la d' utilitzar *TDD* [15] com a metodologia principal de treball, generant tests sempre abans de desenvolupar, pel cost de temps inicial que suposava plantejar molts tests per poc codi, tenint en compte que inicialment es va realitzar una primera fase d'anàlisi de requeriments, es va decidir

mudar la metodologia de *testing* a *Data-Driven Testing* i *ET* (Exploratory testing).

- **DDT - Data Driven Testing**

Per a totes les implementacions de l'entorn web, s'han dissenyat casos de test com el que s'il·lustra a la figura següent, especificant els passos a seguir i les dades. En tractar-se de *DDT*, sempre s'han comprovat valors límit i possibles valors d'error.

Identificador	TC-12
Descripció	Registre usuari invàlid
Pre-condició	Pàgina de registre d'usuari
Cas d'ús afectat	UC-1
Passos a seguir	Emplenar tots els camps amb valor 'a'
Resultat esperat	Registre no completat, missatge d'error informatiu: "El mail ha de contenir '@"

Figura 6. Test case del registre d'usuari

- **Proves de regressió i smoke testing:**

En cada ocasió en que s'ha treballat en la programació d'alguna *feature* nova o modificació s'ha realitzat una prova prèvia i posterior de regressió o *smoke-testing* [16], consistent en la revisió end to end (E2E) de les funcionalitats bàsiques de la pàgina. Aquest tipus de prova, tot i que pugui aparentar tenir menys context que d'altres, és especialment útil per detectar problemes d'integració que han estat generats involuntàriament i podrien passar inadvertits, augmentant amb el temps el cost de solucionar-los.

- **Exploratory testing (ET):**

Al finalitzar qualsevol tasca de programació que impliqués un canvi en alguna funcionalitat, s'han realitzat proves E2E d'Exploratory Testing [17] basades en buscar la situació límit del programa i, per tant, els punts més problemàtics. Cal destacar que s'han detectat bastants defectes i bugs gràcies a aplicar aquesta tècnica, tots solucionats actualment excepte un d'ells, un bug relacionat amb la creació de l'horari que es produeix al donar-se cert solapament entre franges, però que afecta únicament de manera visual a l'administrador.

8 GESTIÓ DEL CANVI

Com s'ha detallat anteriorment, la metodologia que s'ha aplicat en aquest projecte té com a punt favorable que és bona en l'adaptació al canvi, comparant-la amb el model clàssic de *waterfall*, on el cost d'un canvi creix exponencialment amb el temps per a qualsevol projecte. El fet d'haver treballat seguint aquest model ha permès la incorporació de canvis amb implicacions múltiples (Annex A1. Llistat de

canvis), arribant tot i així al MVP del projecte, la qual cosa és un factor molt positiu tenint en compte que el desenvolupament realitzat és d'un *software on demand*, i per tant es pot dir que l'adaptació a les necessitats del client ha estat molt bona.

9 ANÀLISI I DISCUSSIÓ DE RESULTATS

En aquesta secció es repassen i discuteixen els resultats obtinguts en aquest projecte amb la finalitat d'observar fins a quin punt s'ha pogut arribar i poder així justificar si és suficient o no per tal de considerar el projecte assolit amb èxit.

9.1 Definició de l'MVP

Durant el projecte, per tal de classificar el punt en que es troba el desenvolupament en qualsevol moment, s'ha definit una llista de casos d'ús/funcionalitats que, en el cas d'ésser totes completes, conformen l'MVP del projecte, és a dir, la versió de sortida amb els requisits mínims per tal que el producte sigui viable, i per tant satisfaci les necessitats del client. El criteri per definir quins casos d'ús són essencials i quins no, és fruit de l'estudi profund del domini del problema, i també de les indicacions del tutor, ja que ha exercit com a client d'aquest *software on demand*.

Llista casos d'ús del MVP (Minimum Viable Product):

Nom del cas d'ús	Actor	Fet?
Veure branques de coneixement	Tots	✓
Veure assignatures d'una branca	Tots	✓
Registre d'usuari	Tots	✓
Log in d'usuari	Tots	✓
Log out	Tots	✓
Veure l'horari d'una assignatura	Tots	✓
Crea l'horari d'una assignatura	Admin	✓
Elimina l'horari d'una assignatura	Admin	✓
Cerca les candidatures d'una assignatura	Admin	✓
Veure CV d'un candidat	Admin	✓
Sol·licita candidatura a una assignatura	User	✓

Figura 7. Casos d'ús MVP

Llista casos d'ús posteriors al MVP:

Nom del cas d'ús	Actor	Fet?
Afegeix compte de professor	Admin	✓
Assigna assignatures a un professor	Admin	✓
Desvincula un professor d'una assignatura	Admin	✓
Log in de professor	Professor	✓
Modifica l'horari d'una assignatura (Les meves assignatures)	Professor i admin	X

Com es pot observar a les taules anteriors, l'assoliment de l'MVP és una fita de gran importància per aquest projecte. No obstant, ha quedat pendent finalitzar el desenvolupament del cas d'ús de modificació d'un horari, funcionalitat pensada per l'ús dels propis professors que a través de l'accés a "les meves assignatures" (modificat arrel del canvi A12 de l'apèndix), havien de poder gestionar la seva assignatura.

9.2 Revisió de resultats

En aquest apartat es realitza un breu repàs sobre les funcionalitats més importants aconseguides i que formen part de l'MVP.

Gestió d'usuaris:

Figura 8 Part del formulari de registre d'usuari

Figura 9 Formulari d'inici de sessió

Aquests són els formularis de registre i log in d'usuaris (no professors). Depenent del tipus d'usuari es mostraran unes opcions o d'altres al header, d'acord amb els casos d'ús que pertocquen a aquest tipus d'usuari.

Figura 10. Header unlogged user



Figura 11. Header logged user

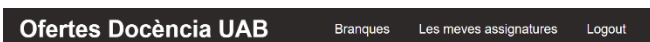


Figura 12. Header professor fixe

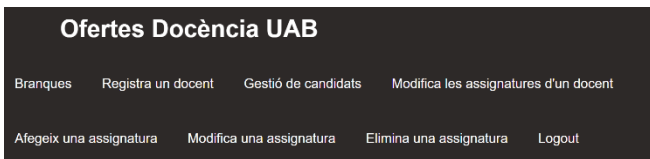


Figura 13. Header administrador

A continuació veiem com és el procés de creació d'un horari per part de l'administrador, així com el posterior procés de vista i sol·licitud d'oferta per part d'un usuari loguejat:

Creació i vista d'horaris:



Figura 14. Creació horari Sistemes Multimedia (admin)

Com podem veure, al moment de la creació d'un horari nou, hem d'utilitzar els botons de tipologia per assignar el tipus de franja que dibuixem, i disposem també d'un botó d'esborrar per si ens calgués tornar a començar. Un cop preparat per publicar-se l'horari, s'assigna a una de les branques existents i se li afegeix el nom i la descripció.



Figura 15. Vista del llistat d'assignatures amb una única assignatura (usuari)



Figura 16. Vista d'una assignatura (usuari)

En fer clic sobre "M'interessa alguna franja", l'usuari loguejat que no és professor pot inscriure's a una assignatura a través del següent formulari:

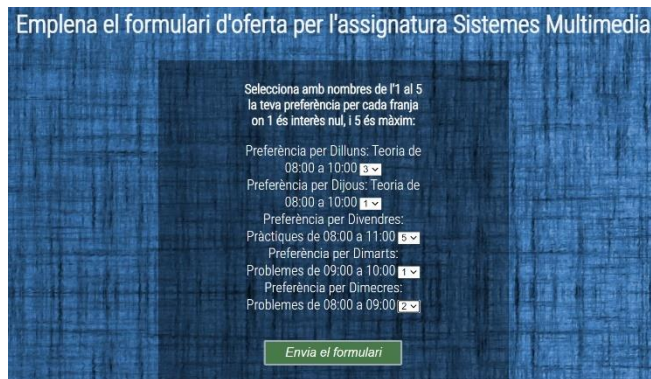


Figura 17. Formulari de candidatura a una assignatura (usuari)

Com es pot observar, la sol·licitud obliga a l'usuari a seleccionar una preferència per totes les franges i no només per una. Això és una estratègia adoptada en base a l'experiència en el procés de contractació explicada pel client, ja que són molt poques les vegades en que una posició es queda fixe i immòbil durant tot el procés de contractació, i per això resulta molt útil tenir informació sobre altres preferències. De totes maneres, totes les preferències es setegen a 1 (interès nul), de mode que s'eviten confusions i l'usuari només modificarà aquelles preferències on hi ha un mínim d'interès.

També s'ha desenvolupat el sistema per gestionar comptes de professors, als quals l'administrador pot afegir o eliminar assignatures, que queden reflectides dins de cadascuna d'aquests comptes al desplegable "les meves assignatures" del header de professor (figura 18).

Finalment també hi ha desenvolupada la funcionalitat de gestió de candidatures, la qual funciona adequadament i permet recuperar la informació sobre els usuaris que han aplicat a una assignatura (nom, mail, preferències i CV).



Figura 18. Formulari de cerca de candidatures

10 PROBLEMES TROBATS I FUTURES LÍNIES DE MILLORA

Abans de concloure i tancar el projecte, es comenten en aquesta breu secció quins han estat els impediments/problemes més significatius durant el desenvolupament d'aquest projecte així com les possibles futures línies de millora en treballs similars.

El principal problema que ha alentit l'avenç del projecte en algun moment ha estat el desconeixement inicial sobre la

forma de la solució, fet que va motivar canvis (Veure A12, Apèndix), postposant lleument l'inici de la implementació. L'altre problema destacable ha estat la definició poc detallada d'algun requisit, la qual cosa ha generat canvis també (A11, A13), ralentitzant en part el projecte pel fet d'haver d'esborrar i/o refer codi.

Com a possibles línies de millora, per al següent projecte realitzaria tallers periòdics (*refinements*) on es treballi la definició dels requisits de la solució, reunint tots els Stakeholders implicats, a fi d'evitar problemes similars als obtinguts, que impliquen canvis i per tant trenquen amb el flux de treball desitjat. També com a línia de millora, realitzaria la definició de l'MVP en una fase primerenca del projecte, ja que tenir la referència del producte mínim viable ha estat important per l'organització del desenvolupament.

11 CONCLUSIÓ FINAL

Finalment, a tall de conclusió, es relacionen els objectius inicials amb el treball desenvolupat i l'estat de l'art actual, per així poder jutjar críticament quin ha estat el nivell d'assoliment del projecte.

El primer objectiu, relacionat amb el coneixement del domini del problema, es considera completament assolit, ja que s'han tingut en compte totes les casuístiques del procés real actual, alhora que s'han realitzat els canvis necessaris per tal que la solució *software* proposada s'hi adapti correctament.

Els següents objectius, tant per la part *back-end*, com per la part *front-end* i la integració d'aquestes, es consideren assolits també, ja que el portal web es troba operatiu i la lògica d'aquest permet realitzar correctament tots els casos d'ús contemplats a l'MVP (amb algun cas d'ús més).

En darrer lloc, cal destacar que el projecte es considera finalitzat amb èxit, ja que s'han implementat i testejat totes les funcionalitats del producte mínim viable, tot i haver introduït diferents canvis durant el cicle de vida del projecte i, en especial, tenint en compte que a la recerca de l'estat de l'art s'ha trobat que és un projecte força únic i atípic pels seus requeriments i la relació que guarden els conceptes d'horari, oferta i assignatura.

AGRAIMENTS

Per finalitzar, vull agrair al tutor del meu projecte, Xavier Otazu, la seva ajuda i implicació durant tot el seguiment d'aquest treball. També m'agradaria agrair als meus amics i familiars l'ànim, el suport i la confiança que han depositat en mi al llarg d'aquests darrers 4 anys de carrera. Finalment, en especial vull donar les gràcies a la meua àvia Dolors per donar-me sempre la motivació necessària i acompanyar-me durant aquesta etapa universitària. Gràcies a tots i totes!

BIBLIOGRAFIA

- [1] «What Is Agile Methodology in Project Management?» Wrike, <https://www.wrike.com/project-management-guide/faq/what-is-agile-methodology-in-project-management/>.
- [2] Davis, Ian. «What Are The Benefits of MVC?» Internet Alchemy, <http://blog.iandavis.com/2018/12/what-are-the-benefits-of-mvc/>.
- [3] «PhpStorm: The Lightning-Smart IDE for PHP Programming by JetBrains». JetBrains, <https://www.jetbrains.com/phpstorm/>.
- [4] «Las 20 Páginas Web con más ofertas de Empleo en España». EmpleoMarketing, 20 de enero de 2018, <https://empleo-marketing.com/mas-ofertas-de-empleo-espana/>.
- [5] Doodle. <https://doodle.com/es/>.
- [6] Atlassian. «What Is Continuous Integration». Atlassian, <https://www.atlassian.com/continuous-delivery/continuous-integration>.
- [7] Git for Windows. <https://gitforwindows.org/>.
- [8] PHP: PDO - Manual. <https://www.php.net/manual/en/book.pdo.php>.
- [9] PHP: MySQLi - Manual. <https://www.php.net/manual/es/book.mysqli.php>.
- [10] XAMPP Installers and Downloads for Apache Friends. <https://www.apachefriends.org/es/index.html>.
- [11] contributors, phpMyAdmin. «PhpMyAdmin». PhpMyAdmin, <https://www.phpmyadmin.net/>.
- [12] Trello. <https://trello.com/b/G6r0PAZ8/tfg-web-per-al-suport-a-la-contractacio-de-professorat>.
- [13] Data-driven testing - Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Data-driven_testing.
- [14] Muetton, Julien. themouette/jquery-week-calendar. 2010. 2021. GitHub, <https://github.com/themouette/jquery-week-calendar>.
- [15] Desarrollo guiado por pruebas - Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Desarrollo_guiado_por_pruebas.
- [16] «Guidelines for Smoke Testing». MSDN Library for Visual Studio 2005, [https://msdn.microsoft.com/en-us/library/ms182613\(VS.80\).aspx](https://msdn.microsoft.com/en-us/library/ms182613(VS.80).aspx).
- [17] What is Exploratory Testing? Techniques with Examples. <https://www.guru99.com/exploratory-testing.html>.
- [18] State. <https://refactoring.guru/design-patterns/state>.

APÈNDIX

A1. LLISTAT DE CANVIS

En aquesta secció es documenten els canvis que han estat més rellevants durant el desenvolupament del projecte.

A11. Canvi de “la meva assignatura” a “les meves assignatures”

Motiu del canvi: Falta de detall en l'especificació del requisit del cas d'ús “Modificar una assignatura/Modificar la meva assignatura” i en conseqüència desenvolupament d'una funcionalitat errònia que va ésser detectada en una de les reunions de seguiment amb l'Stakeholder (tutor)

Implicacions del canvi: Canvis a la base de dades: “Creació d'una taula auxiliar per permetre la relació N-N entre professors i assignatures”. Modificació d'algunes funcions de l'arxiu “users.php” i prioritització del grup de tasques que engloben els casos d'ús “Assigna una assignatura a un professor” i “Desvincula un professor d'una assignatura”.

Data: Setmana 11 del cicle de vida del projecte.

A12. Canvi en el model de la solució, es passa d'ofertes fixes i limitades a creació d'horaris d'assignatura

Motiu del canvi: El plantejament de la solució que s'estava analitzant a la primera fase d'anàlisi de requeriments amb metodologia “waterfall”, generava molts dubtes ja que el plantejament de generació manual d'ofertes que es proposava, aparentment no agilitzava massa el procés manual actual.

Implicacions del canvi: Durant les entrevistes amb l'Stakeholder es van realitzar petites sessions de debat i *brainstorming* respecte el problema que motiva aquest canvi. Es va acabar amb la idea d'eliminar l'oferta manual i el document de dades inicial per implementar un model de generació d'ofertes a través d'un horari setmanal i es va començar a treballar en l'anàlisi i disseny d'aquest nou model.

Data: Setmana 4-5 del cicle de vida del projecte.

A13. Canvi d'ubicació del camp CV (currículum) de la sol·licitud de candidatura al registre d'usuari

Motiu del canvi: Falta de detall en l'especificació de requisits del registre d'usuari i en conseqüència: plantejament erroni de la solució del cas d'ús “sol·licitar oferta” en la fase de desenvolupament final, ja que en el formulari de cada sol·licitud d'oferta realitzada es demanava a l'usuari inserir un arxiu *.pdf* currículum.

Implicacions del canvi:

Aquest canvi, com d'altres esmentats anteriorment, té més d'un àmbit d'impacte. En primera instància, cal un petit redisseny a la base de dades, eliminant la taula auxiliar *currículums*, on s'emmagatzemaven els noms dels documents emmagatzemats. Seguidament,

```

gestio_contractacio_bd currículums
id : int(11)
usuari_id : int(11)
nom_cv : varchar(200)
    
```

Figura 19. Taula eliminada al canvi 3

es modifica la taula usuaris, afegint el camp *nom_cv*. Co-lateralment, a la banda de codificació, cal modificar totes les funcions del model relacionades amb les taules esmentades, que concretament es troben dins els arxius *offers.php* i *users.php*, refactoritzant també els controladors si és necessari. Paral·lelament, cal modificar tant el formulari de registre d'usuari, com el de sol·licitud d'oferta, ahora que es validen les funcionalitats tant del *happy-path* com d'altres casuístiques que puguin haver estat afectades indirectament. Els casos d'ús afectats pel canvi són: *Registre d'usuari*, *Sol·licitud d'oferta* i *gestionar candidats*.

Data: Setmana 16 del cicle de vida del projecte.

A2. DIAGRAMA DE CASOS D'ÚS GENERAL

A21. Versió 0

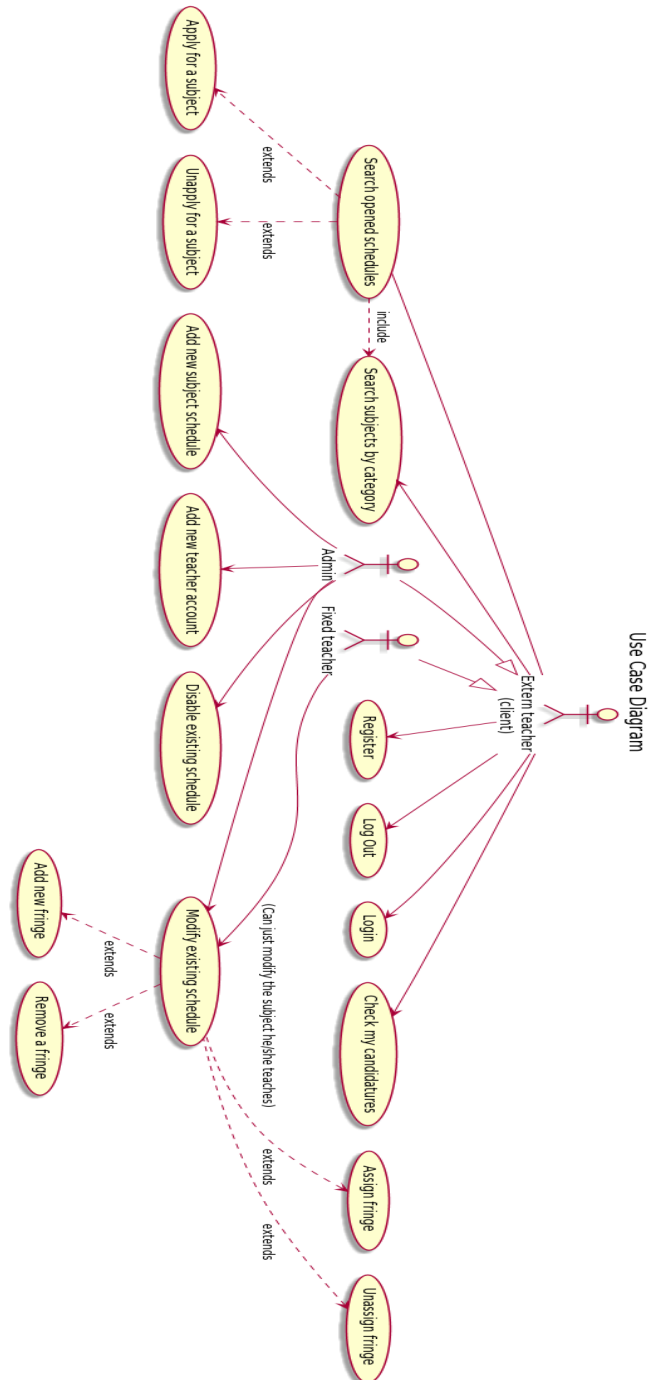


Figura 20. Diagrama de casos d'ús setmana 10 - Versió 0

A22. Versió 1 (MVP)

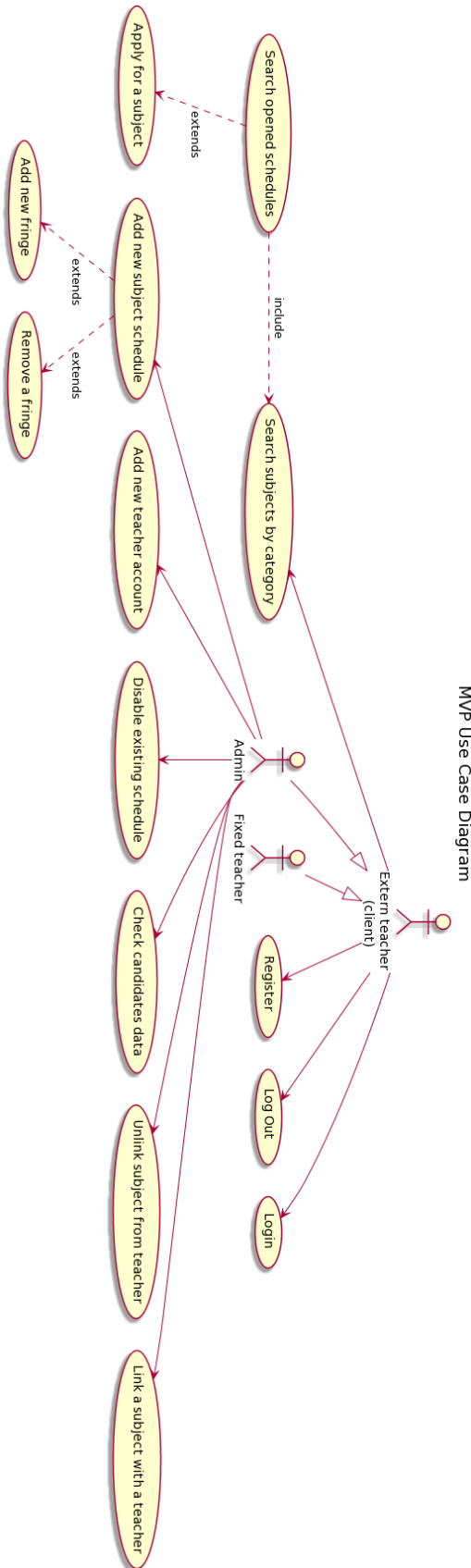


Figura 21. Diagrama de casos d'ús versió MVP - Versió 1

A23. Versió 2 (Producció final)

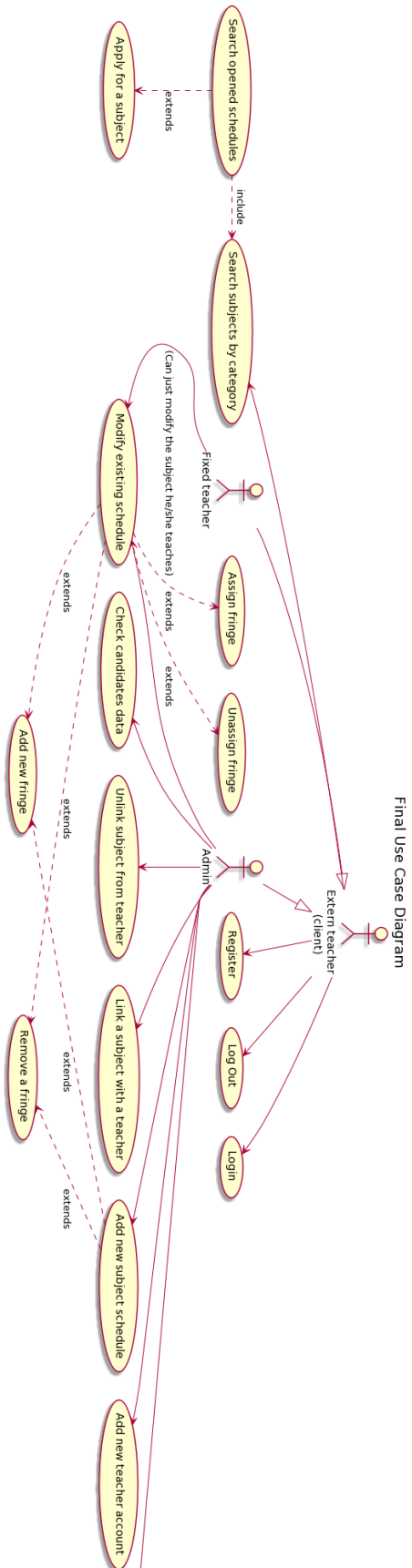


Figura 22. Diagrama de casos d'ús versió final - Versió 2

A3. DIAGRAMA DE CLASSES

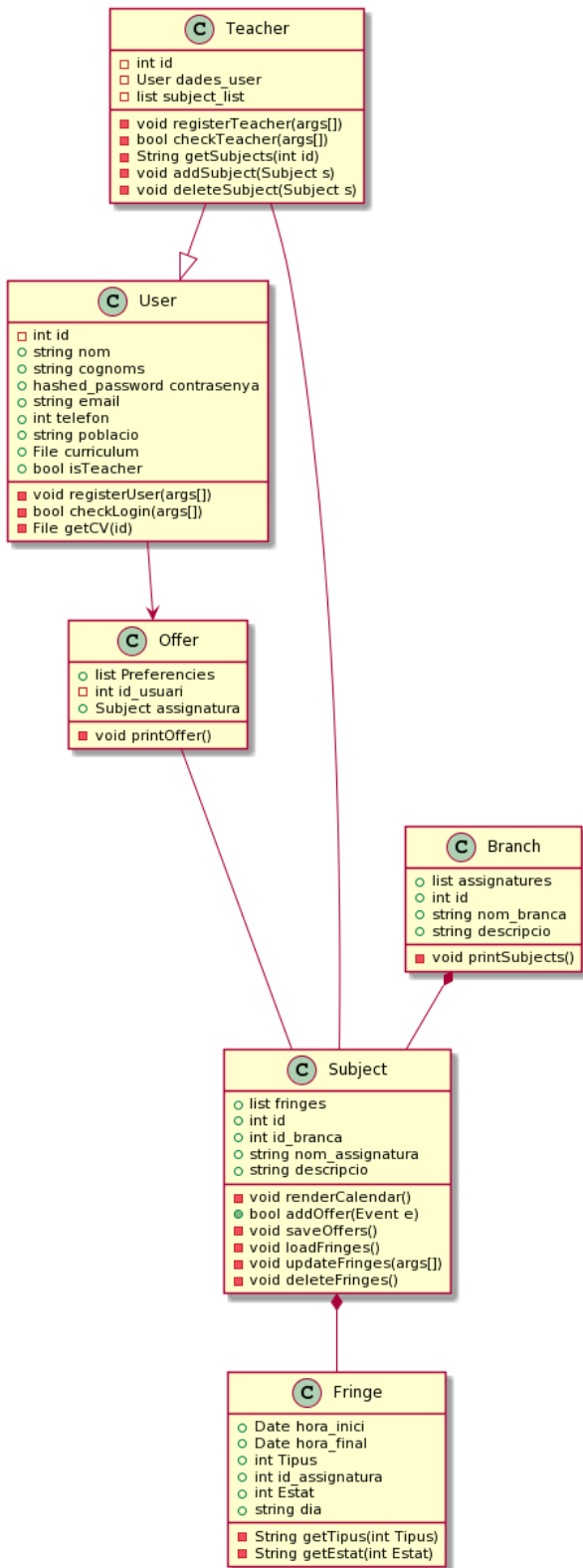


Figura 23. Diagrama de classes dels conceptes oferta - assignatura - horari

A4. DIAGRAMES DE GANTT – GESTIÓ DE LA PLANIFICACIÓ

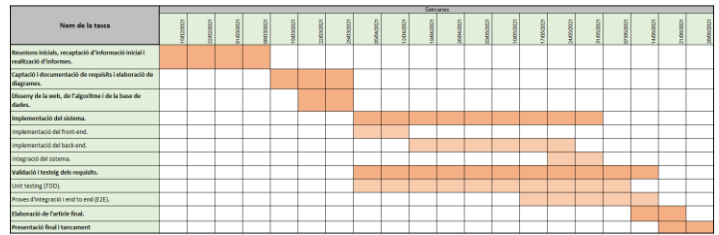


Figura 24. Diagrama de Gantt Fase inicial

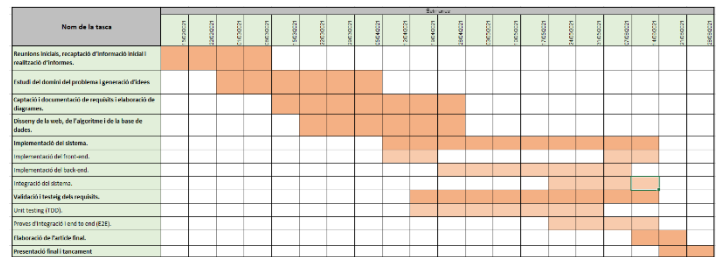


Figura 25. Diagrama de Gantt a mig desenvolupament (setmana 16)

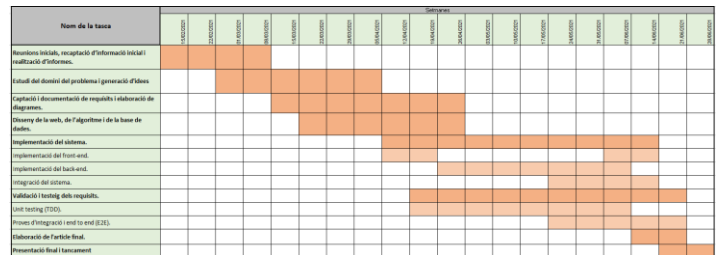


Figura 26. Diagrama de Gantt Fase final (últimes setmanes)