



## Fòrums UOC

**Alexandre Cabrera Gil**  
Grau d'Informàtica

**Xavier Aracil Díaz**

10-06-2014



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

## FITXA DEL TREBALL FINAL

<b>Títol del treball:</b>	<i>Fòrums UOC</i>
<b>Nom de l'autor:</b>	<i>Alexandre Cabrera Gil</i>
<b>Nom del consultor:</b>	<i>Xavier Aracil Díaz</i>
<b>Data de lliurament (mm/aaaa):</b>	<i>06/2014</i>
<b>Àrea del Treball Final:</b>	<i>Eines del Campus UOC</i>
<b>Titulació:</b>	<i>Grau d'Enginyeria Informàtica</i>
<b>Resum del Treball:</b>	
<p>La Universitat Oberta de Catalunya és un referent en quant a la modernització de la educació i l'ús de les noves tecnologies per a oferir oportunitats a les persones que no disposen del temps o l'espai per a cursar una titulació en una Facultat. Per tant, la qualitat de les seves eines ha d'estar a l'altura del seu nivell transformador.</p> <p>Un estudiant disposa de molts recursos de comunicació, però aquests es basen en models de comunicació obsolets que no són intuïtius i frenen l'aprenentatge. Per això <b>Fòrums UOC</b> pretén donar una aproximació més natural de la interacció de les persones.</p> <p>Per una banda es modernitza el sistema de fòrums agrupant els missatges relacionats en conversacions per tal de facilitar el seguiment dels fils de comunicació i organitzar la informació. També s'ha fet un esforç per netejar la disposició de la informació per facilitar l'accés i relacionar millor els diferents components dels missatges. Una part molt especial de <b>Fòrums UOC</b> recau en que la gran majoria del processament de la informació recau en el client, fet que augmenta la sensació de continuïtat en l'accés a la informació.</p> <p>Finalment, en el món de la immediatesa en que vivim, és molt important disposar d'una eina de comunicació instantània per a que els usuaris puguin interactuar en temps real.</p> <p>Tot es basa en rodejar a l'usuari de la <i>UOC</i> d'eines per a que es senti còmode formant part de la universitat.</p>	

**Abstract:**

In terms of innovative learning methods, the Open University of Catalonia is an example of learning through new technologies. This kind of long-distance studies bring new opportunities to people with no spare time to attend regular lessons at the university.

The university offers plenty of communication resources to students, but most of them are based on non-intuitive and obsolete models that slow down the learning process. This is why **Fòrums UOC** was created: to enhance the interaction between students.

Firstly, **Fòrums UOC** has been updated with a thread association system, which gathers subject related messages in conversation groups. The objective of this system is to ease the understanding of the messages and to organize the information.

Secondly, it aims to clear the user's interface so as to improve the navigation. Thanks to this updated system, users are able to get to the desired information units quick and easily.

But it is the way how it handles the refresh flow what makes **Fòrums UOC** unique. As the heavy tasks are supported by the client, users have a continuous navigation feeling while they are receiving new information.

We are living in a fast-information world, so a bidirectional communication tool is a must to promote student interaction in real time. This novelty has been developed carefully in order to fit with the existing default communication resources.

The main goal **Fòrums UOC** system is to provide quality communication tools, so that *UOC* students can feel they are part of the university community.

**Paraules clau:**

Comunicació, fòrum, conversa, xat, interacció d'estudiants

# Índex

<b>1. Introducció.....</b>	<b>1</b>
1.1. Context i justificació del Treball.....	1
1.2. Objectius del Treball.....	2
1.3 Enfocament i mètode seguit.....	3
1.4 Planificació del treball.....	4
1.4.1 Calendari.....	4
1.4.2 Alteracions de la planificació.....	5
1.5 Breu sumari de productes obtinguts.....	6
1.6 Breu descripció dels altres capítols de la memòria.....	7
<b>2. Obtenció i gestió de requisits.....</b>	<b>9</b>
2.1 Requisits funcionals.....	9
2.2 Requisits no funcionals.....	10
2.3 Diagrama de casos d'ús.....	11
2.4 Casos d'ús.....	12
<b>3. Disseny.....</b>	<b>20</b>
3.1 Model conceptual.....	20
3.2 Diagrama d'arquitectura.....	22
3.2.1 Diagrama d'arquitectura de maquinari.....	22
3.2.2 Diagrama d'arquitectura de programari.....	23
3.3 Diagrama E-R.....	25
3.4 Model de classes.....	26
3.5 Disseny de la interfície d'usuari.....	28
3.5.1 Disseny simple i net.....	29
3.5.2 Adaptació a diferents resolucions.....	31
3.5.3 Navegació intuïtiva.....	32
3.5.4 Comunicació asíncrona.....	33
<b>4. Desenvolupament.....</b>	<b>34</b>
4.1 Avaluació tecnològica.....	34
4.2 Programari utilitzat.....	36
4.2.1 HTML5 + CSS3.....	36
4.2.3 Javascript.....	37
4.2.4 PHP.....	43
4.2.5 MySQL.....	43
4.3 Capes de l'aplicació.....	45
4.3.1 Client.....	46
4.3.2 Servidor.....	47
<b>5. Configuració del servidor i desplegament.....</b>	<b>51</b>

5.1. Connexió al servidor.....	51
5.2. Instal·lació dels serveis necessaris.....	53
5.3. Configuració del servidor.....	55
5.4. Instal·lar l'aplicació.....	56
5.5. Crear la base de dades.....	57
5.6. Executar l'aplicació.....	58
<b>6. Avaluació de costos.....</b>	<b>60</b>
<b>7. Retorn de la inversió.....</b>	<b>62</b>
<b>9. Conclusions.....</b>	<b>66</b>
<b>10. Glossari.....</b>	<b>69</b>
<b>11. Bibliografia.....</b>	<b>71</b>
<b>12. Annexos.....</b>	<b>73</b>
12.1. Annex 1. Manual d'usuari.....	a
12.2. Annex 2. Diagrama de Gantt.....	A

## Llista de figures

Figura 1. Diagrama de Casos d'ús.....	11
Figura 2. Model conceptual.....	20
Figura 3. Diagrama d'arquitectura de maquinari.....	22
Figura 4. Diagrama d'arquitectura de programari.....	23
Figura 5. Diagrama E-R.....	25
Figura 6. Model de Classes.....	26
Figura 7. Esquema d'interfície d'usuari.....	28
Figura 8. Pantalles de l'aplicació.....	29
Figura 9. Detall de la personalització de l'usuari en l'aplicació.....	30
Figura 10. Detall d'un element en l'aplicació.....	30
Figura 11. Detall de la eina de missatgeria instantània.....	31
Figura 12. Detall de menús en la eina de missatgeria instantània.....	31
Figura 13. Mapa de navegació del campus virtual.....	33
Figura 14. Esquema d'autenticació del protocol OAuth 2.0.....	40
Figura 15. Esquema de capes i components de l'aplicació.....	45
Figura 16. Generació de clau privada.....	52
Figura 17. Connexió al servidor utilitzant SSH.....	52

# 1. Introducció

## 1.1. Context i justificació del Treball

**Fòrums UOC** és una aplicació web destinada a millorar la interacció entre els usuaris de la *UOC*. Actualment, tant professors com alumnes, disposen d'un sistema de comunicació basats en la missatgeria o sistema de bústies. Es distingeixen 3 seccions principals:

- **Fòrums:** Tots els usuaris poden participar i serveixen com a eina lliure per a resoldre dubtes i presa de contacte entre alumnes i professorat.
- **Debats:** S'estructuren de forma similar als fòrums, però s'utilitzen per a avaluar la participació dels estudiants en algunes activitats de l'assignatura.
- **Tauler:** l'utilitza el professorat per a transmetre comunicacions oficials i globals de l'assignatura. Els alumnes no hi poden participar, només visualitzar-lo.

Tot i aquesta varietat de seccions, la estructura que ofereixen és la mateixa, cosa que resulta molt poc flexible i poc útil quan s'ha de seguir una conversació en un debat, o s'ha de cercar un missatge concret.

Per tant l'objectiu principal d'aquest projecte, resideix en estructurar la informació que contenen les eines de comunicació del campus virtual aportant una millora estètica i pràctica dels recursos. Els beneficis principals de l'usuari al fer servir aquesta aplicació web seran:

- Tenir les comunicacions de les assignatures en un mateix espai
- Visualitzar de forma ràpida els apartats on hi han hagut canvis des de l'última visita
- Tenir una estructura diferenciada per als diferents tipus de nous missatges (tauler, fòrum, debat)
- Poder accedir a cada assignatura de forma independent per veure els canvis
- Disposar d'una estructura de fòrums i sub-fòrums de fàcil navegació
- Visualitzar els missatges estructurats en conversacions per facilitar el seguiment de les converses
- Disposar d'una eina de cerca
- Accedir a les opcions d'interacció de missatges, de forma senzilla i distribuïda als punts d'acció adequats.
- Disposar d'un servei de missatgeria instantània per a conversacions privades
- Poder seleccionar estudiants de la *UOC* com a amestat
- Veure l'estat de connexió dels usuaris



## 1.2. Objectius del Treball

- **Realitzar una connexió segura entre l'usuari i els servidors de la UOC**  
Una de les característiques més importants de qualsevol aplicació web, te a veure amb la seguretat i la privacitat de les dades que circulen per la xarxa. És fonamental assegurar que les dades de l'usuari no podran ser interceptades. La *UOC* fa servir el protocol *OAuth*<sup>1</sup> que permet fer connexions a tercers de forma segura.
- **Mostrar informació actualitzada de les eines de comunicació en les aules**  
Quan l'usuari entra per veure les novetats en el campus virtual vol veure les novetats que hi pot trobar. Per tant és important mostrar de forma clara quins són els nous missatges, tant en les conversacions com les aules i fòrums que contenen informació nova.
- **Estructurar la informació en apartats de diferent profunditat per facilitar la cerca d'informació**  
Un aspecte clau per millorar la comprensió i llegibilitat de la informació resideix en la estructura del contingut. Per tant, un sistema de nivells (aules/fòrums/conversacions) ajudarà a la navegació i a accedir a la informació rellevant.
- **Presentar els missatges de forma clara i amb un disseny que permeti una lectura fluida i àgil**  
S'ha de definir amb cura la forma en que es mostrarà cada missatge per a que la informació es percebi de forma natural i comprensible. Una bona diferenciació del contingut, emissor, títol, data, fitxers adjunts, fotografia del emissor, elements d'interacció... ajudaran a identificar els missatges, navegar per l'aplicació i comprendre millor la informació.
- **Oferir interacció amb la plataforma i eines de navegació**  
Per a que sigui una eina de comunicació útil, ha de permetre executar les dos vies de la comunicació: rebre i enviar missatges. Això fa que sigui igual d'important poder respondre els missatges, donar-los per llegits i enllaçar amb la informació rellevant.
- **Utilitzar tot el potencial de les tecnologies web per a millorar la experiència d'usuari**  
Des de ja fa uns anys, els navegadors principals ofereixen funcionalitats que poden ajudar a aquest projecte a millorar la seva eficiència i augmentar les seves possibilitats. Per exemple el *WebStorage*<sup>2</sup> per desar dades de la sessió de l'usuari o els *WebSockets*<sup>3</sup> per implementar de forma més eficient el servei de missatgeria instantània.
- **Incorporar una eina de comunicació instantània (xat)**

---

1 <http://oauth.net/>

2 [http://www.w3schools.com/html/html5\\_webstorage.asp](http://www.w3schools.com/html/html5_webstorage.asp)

3 <http://www.websocket.org/aboutwebsocket.html>

Aquest projecte intenta modernitzar la forma en que els usuaris de la *UOC* es comuniquen. Per tant, només reestructurant el sistema de taulers en que s'organitzen els missatges de l'aula, no s'adequaria als temps en que ens trobem. Avui en dia, les comunicacions tendeixen a ser més immediates, i un servei de missatgeria instantània podria ser una eina molt útil per als membres de la universitat.

- **Realitzar un sistema d'amistats per a comunicar-se de forma senzilla amb els contactes més habituals**

Aquest sistema serà molt útil per a donar utilitat a la eina de comunicació instantània. Permetrà definir usuaris habituals, de forma que facilitarà la tasca de comunicar-se amb els usuaris amb més interacció.

### 1.3. Enfocament i mètode seguit

**Fòrums UOC** pretén constituir una eina de comunicació centralitzada que farà ús de les eines existents de la *UOC* a través de la seva *API* i utilitzarà eines externes per tal d'oferir un servei de comunicació que no existeix actualment a la universitat.

Una forma molt pràctica i usable de elaborar el sistema seria enfocant-lo cap a una una forma de treball anàloga a una aplicació d'escriptori, allunyant-se de la clàssica pàgina web.

Una pàgina web tradicional serveix planes més o menys personalitzades, però estàtiques una vegada han arribat al navegador de l'usuari, fent que cada cop que l'usuari vol interactuar, sigui el servidor el que ha de gestionar la tasca i tornar a enviar informació. Això implica que els temps d'espera trenqui la sensació de conjunt i continuïtat fent que no es pugui percebre com una aplicació.

Per a solucionar aquest problema s'ha optat per implementar un sistema molt intensiu en la part del client fent que els refrescos de la pàgina siguin pràcticament inexistents.

Gràcies a les possibilitats que ens ofereixen les noves propietats del *HTML5* juntament amb el llenguatge de programació *Javascript* i les eines que ens proporciona, es pot modelar una aplicació que s'executi en el propi navegador i que demani les dades sota demanda mitjançant crides *REST* asíncrones.

L'allotjament de l'aplicació es durà a terme en una màquina virtual de *Amazon Web Services* proporcionada per la pròpia *UOC*. Aquest tipus de servidor ofereix una customització molt eficient, per tant es podrà donar servei a tota l'aplicació així com el servei de missatgeria instantània.

## 1.4. Planificació del treball

Per a establir la planificació del projecte, s'ha definit un *diagrama de Gantt*<sup>4</sup> que ens ajudarà a establir, de forma visual, la distribució temporal de les diferents tasques del projecte. En aquest diagrama es pot observar l'inici i final de cada tasca.

### 1.4.1. Calendari

- **Realització de la primera pluja d'idees i planificació**  
Aquesta tasca comprèn el temps necessari per pensar el projecte i realitzar totes les tasques prèvies de planificació i organització. Bàsicament comprèn la càrrega de treball de la primera entrega.
- **Anàlisi de les necessitats del projecte**  
Anàlisi de les propietats i necessitats del projecte.
- **Escriure formalment els requisits i objectius del projecte**  
Definir de forma clara els requisits funcionals del projecte detectats en la fase d'anàlisi i elaborar casos d'ús de l'aplicació web.
- **Disseny del projecte**  
Realitzar el disseny lògic del sistema i definir les seves característiques i restriccions.
- **Estudi i elecció de les tecnologies en que es basarà el projecte**  
Existeixen diferents mètodes per afrontar aquest projecte, i cal una etapa específica per a valorar les diferents opcions abans de començar a implementar el sistema.
- **Desenvolupament del primer prototip per a la segona entrega**  
Establir la arquitectura general de l'aplicació, definir el sistema de comunicació amb la API, implementar el sistema d'autenticació i finalitzar mostrant la informació organitzada jeràrquicament.
- **Memòria entrega final**  
Ficar per escrit totes les etapes que ja estan finalitzades i que es demanaran com a prova de seguiment del projecte.
- **Començar l'elaboració de la memòria**  
Definir la estructura de la memòria global del projecte, incorporar les parts comunes amb la segona prova d'avaluació continuada, seguir redactant apartats, almenys fins al punt de desenvolupament on es troba el projecte.
- **Desenvolupament del prototip final**  
Finalitzar la implementació del sistema de fòrums, incorporar la interacció amb la plataforma, polir el disseny visual i realitzar el sistema de missatgeria instantània.

---

4 [http://es.wikipedia.org/wiki/Diagrama\\_de\\_Gantt](http://es.wikipedia.org/wiki/Diagrama_de_Gantt)

- **Finalitzar la memòria**  
Acabar de redactar els apartats corresponents a les noves implementacions, afegir tota la informació necessària i plasmar la conclusió.
- **Finalitzar tots els recursos extra per a l'entrega**  
Preparar tot el material per a complir les exigències de format. Revisar que tot el material necessari s'afegeix al fitxer d'entrega.
- **Realització de la video-presentació**  
Elaboració d'un resum de la pràctica. Preparar una demostració de les funcionalitats de l'aplicació. Enregistrament de la presentació.
- **Final del projecte**

En el període de temps del projecte hi han 2 conjunts de dies importants festius: 18/21 d'abril que és setmana santa, i 23/25 de maig que és festiu a Lleida. S'han tingut en compte a les tasques que contenen aquests períodes de temps prolongant-los lleugerament. No s'han tingut en compte la resta de festius degut a que tenen una duració curta.

També cal esmentar, que hi ha disponibilitat de treball durant tots els caps de setmana.

En la secció d'annexos es pot trobar una representació gràfica de la planificació realitzada mitjançant un **diagrama de Gantt**.

#### **1.4.2. Alteracions de la planificació**

Com era d'esperar, tota planificació te les seves desviacions i aquesta no n'ha estat una excepció. Aquest projecte te una gran dependència de l'*Open-API* de la *UOC* ja que totes les seves dades es carreguen dinàmicament d'aquest recurs. Per tant, qualsevol problema de disponibilitat extern feia gairebé inoperable l'aplicació, tant per l'apartat de gestió de fòrums com per la eina de missatgeria instantània, que necessita una sessió vàlida de la *UOC* per tal d'autenticar-se i iniciar-se en les sales.

Concretament s'havia distribuït la implementació de l'aplicació en doc blocs temporals: el desenvolupament del primer prototip entre el 28-03-2014 i 14-03-2014, i el desenvolupament del prototip final començant el 01-05-2014 i acabant el 15-05-2014.

En el primer bloc de desenvolupament, el servei va tenir una disponibilitat raonable, on només va deixar d'estar accessible entre les dates 29-03-2014 i 31-03-2014, tot un cap de setmana durant el qual no es va poder avançar en el desenvolupament del prototip.

Malauradament, la disponibilitat del l'*Open-API* durant el segon període va ser molt més deficient, al que se li ha d'afegir l'aparició d'altres imprevistos. Aquest projecte necessita la gestió d'un servidor dedicat per servir els seus fitxers i allotjar el servei de missatgeria instantània que ha d'estar sempre disponible en xarxa. Per aquest motiu la universitat va proporcionar l'accés a una màquina virtual de *Amazon*. Per accedir al servidor era necessari la obtenció de una clau privada per autenticar a l'usuari que intenta iniciar sessió al servidor. L'inici de la configuració del servidor estava planificat per iniciar-se el dia 10-05-2014, però realment no va poder començar fins al 13-03-2014 degut als problemes d'accés a la clau citada. A més a més, aquest període d'implantació del prototip final va ser contemporània amb l'actualització de l'entorn de proves de l'*Open-API*, el funcionament d'aquesta implantació no va ser tant tranquil com s'esperava i ocasionava un gran nombre de fallades en l'accés del servei. Concretament, el servei va estar inaccessible des del 14-05-2014 i no es va recuperar fins al 20-05-2014, un gran nombre de dies que va penalitzar molt l'organització de la càrrega de treball.

Tots aquests problemes fan que de 31 dies de treball planificats per dedicar a la implementació i implantació de la solució, 11 fossin molt poc productius degut a incidències i imprevistos bloquejants. Aproximadament un 33% del temps planificat és una penyora molt important tenint un temps tant reduït, i el producte final se'n ha ressentit, tenint que eliminar algunes funcionalitats previstes que haurien donat molt de valor al prototip presentat.

## **1.5. Breu sumari de productes obtinguts**

Al final de la realització del projecte podem trobar un prototip funcional que amplia les possibilitats de comunicació disponibles a la *UOC* que permeten una comunicació més directa i eficient entre els seus usuaris.

Per una banda, s'ha realitzat una reestructuració de la forma en que s'estructuren els recursos de comunicació de la *UOC*. Hem passat d'una estructura monopolitzada per missatges individuals organitzats de forma temporal, a una estructura governada per la relació entre missatges organitzats per la temporalitat del grup i no pas d'una entitat única.

Amb aquest canvi s'intenta fer evolucionar els recursos de comunicació per tal de dotar-los d'una estructura moderna, seguint el rumb que estan seguint els referents de comunicació del sector. Aquest fet, no sol ajuda a mantenir la sensació de conversa de les aportacions dels usuaris, també aprofita els punts forts i comuns dels recursos que està acostumat a fer servir l'usuari per a que la corba d'aprenentatge de la eina sigui molt més suau. Per tant, quan un usuari accedeixi a l'aplicació, trobarà una selecció de les seves assignatures i dels recursos de comunicació associats, agrupats per assignatura. Des d'aquest menú podrà navegar per tots els missatges de l'assignatura i

podrà veure tota la informació que contenen amb una organització més intuïtiva i esperada dels missatges ja que estaran agrupats per converses.

Ahora l'aplicació disposa d'una eina de comunicació instantània (xat) que permet la comunicació entre usuaris de en temps real i de forma directa. Aquesta eina mostra tots els usuaris que estan connectats actualment a l'aplicació i que estan cursant les mateixes assignatures que l'estudiant. Per a afavorir la interacció entre usuaris més afins, també s'ha afegit la possibilitat de marcar usuaris com a amics els quals apareixeran en una llista destacada per a un fàcil accés.

L'eina de comunicació instantània intenta solucionar una carència en quant a usabilitat de la *UOC*. Moltes assignatures demanen interacció freqüent i intensiva entre usuaris, els quals han de buscar vies de comunicació externa per tal de complir amb la tasca adjudicada. A més a més, moltes de les tasques habituals dels usuaris s'han de solucionar amb eines que no són adequades degut a la immediatesa de les mateixes.

## **1.6. Breu descripció dels altres capítols de la memòria**

La documentació del projecte està separada en set apartats. En ells s'abordaran tots els aspectes més importants del desenvolupament, des dels inicis de la planificació fins a la posada en marxa del prototip.

Es començarà explicant la fase d'obtenció i gestió de requisits on es cerca quines característiques ha de complir el nou sistema, per a fer-ho es defineixen uns requisits -tant funcionals, com no funcionals- que seran utilitzats per generar els casos d'ús. Per acabar es mostraran de forma detallada els casos d'ús seleccionats, que ens serviran per a prioritzar quines funcionalitats són més importants per a l'aplicació.

Una vegada es tenen clars els requisits del sistema, el següent pas consisteix en definir el disseny del sistema. Es parla de totes les etapes del disseny que definiran la arquitectura abans de començar amb la implementació del servei. Aquest apartat inclou varis diagrames de l'aplicació com el model conceptual, diagrames d'arquitectura (Maquinari i programari), diagrama E-R de la base de dades, model de classes i disseny de la interfície on també s'explica quins criteris i passos s'han seguit per a definir la interfície d'usuari.

Després de establir la base estructural de l'aplicació esdevé el torn del desenvolupament de l'aplicació. Comença realitzant una avaluació tecnològica de les diferents possibilitats i metodologies que es podrien aplicar per desenvolupar el prototip. Més endavant es fa una explicació detallada de quines són les tecnologies utilitzades finalment per a complir la tasca que s'han definit en els apartats anteriors. Finalment es realitza una explicació de com s'ha estructurat el desenvolupament, quines entitats i estratègies s'han adoptat per a gestionar tant la banda del servidor,

com la banda del client, i com s'ha integrat l'ús de l'*Open-API* com a proveïdor de dades de l'aplicació.

Seguidament, es fa un anàlisi dels requeriments d'instal·lació del servidor des de la configuració inicial fins al manteniment necessaris per a executar i proporcionar tots els serveis de l'aplicació. Alhora dona les pautes a seguir per a realitzar la connexió al servidor, transferència de fitxers. A més a més també inclou la estructura de la base de dades i una breu explicació de com iniciar el servei de base de dades de forma satisfactòria.

Un cop finalitzada l'explicació del desenvolupament, es realitza una avaluació de costos relativa tant al desenvolupament, com a la implantació com a possibles millores de l'aplicació i s'avaluarà quin possible rendiment pot produir per tal de recuperar la inversió inicial.

A més a més, es detallen un conjunt d'idees realitzables en el futur que podrien canviar i millorar l'oferta del prototip, i fent més atractiva l'aplicació final.

Finalment, tota la gestió del projecte ve acompanyada d'un apartat de conclusions o s'aborda un resum de l'experiència obtinguda realitzant tot el projecte, com ha sigut treballar amb les tecnologies escollides, i quina és la avaluació dels resultats obtinguts.

Al final d'aquest document es pot trobar un recull de seccions d'interès per al lector: glossari de termes, bibliografia, i annexos.

## 2. Obtenció i gestió de requisits

### 2.1. Requisits funcionals

- **El sistema ha de permetre accedir a les dades del campus virtual d'un usuari en concret.**

L'aplicació es una eina que permet veure els fòrums als que te accés un estudiant, per tant és necessari que l'usuari es pugui identificar.

- **Un usuari ha de poder accedir a tots els recursos de comunicació de totes les aules.**

Per a que aquesta eina sigui útil, ha de mostrar tota la informació imprescindible. Part d'aquesta informació recau en la representació de totes les aules i els seus recursos de comunicació (taules, fòrums, debats).

- **Un usuari ha de poder accedir als missatges dels recursos de comunicació.**

Cada recurs de comunicació contindrà missatges i/o converses i l'usuari ha de ser capaç de visualitzar-ho en la seva totalitat.

- **El sistema ha de mostrar el nombre de missatges que te cada recurs i el nombre de missatges que no han sigut llegits.**

Per a una ràpida identificació de les novetats, cada recurs de comunicació ha de mostrar la informació rellevant del seu contingut. Cal destacar la importància que tenen els missatges no llegits ja que és informació nova.

- **El sistema ha de permetre accedir als sub-apartats dels recursos de comunicació.**

Els fòrums poden estar dividits en sub-apartats els quals constitueixen un altre recurs de comunicació en si mateixos i poden contenir converses.

- **El sistema ha d'organitzar els missatges en converses.**

No serveix només mostrar els missatges. El sistema ha de ser capaç d'identificar una relació entre els missatges i mostrar els mateixos de forma que faciliti la lectura i el seguiment de les aportacions.

- **L'usuari ha de poder identificar les converses que contenen missatges nous.**

Igual que els fòrums, els missatges han de ser mostrats de forma cronològica i senyalar clarament els que contenen informació que encara no ha sigut visualitzada.

- **El sistema ha de mostrar la informació rellevant de cada un dels missatges: tema, emissor, data de publicació i cos del missatge.**

La base de l'aplicació resideix en poder llegir de forma còmoda i completa els missatges que es troben en els recursos de comunicació. Per a fer-ho, es bàsic que es mostri la informació més important dels missatges.

- **L'usuari ha de poder respondre a una conversació.**

Una altra part molt important del sistema és el fet de poder interactuar amb els altres estudiants de la UOC. La forma de fer-ho mitjançant els fòrums és participant a les converses. I per a fer-ho és necessari poder respondre a missatges previs.



- **L'usuari ha de poder respondre a una conversació utilitzant parts d'altres missatges en forma de cita.**

Una altra forma d'interactuar resideix en utilitzar parts de les aportacions anteriors per tal de poder respondre de forma més personal i orientada.

- **El sistema ha d'oferir una eina de missatgeria instantània**

Aquesta aplicació pretén simplificar i facilitar les comunicacions entre usuaris, i una bona eina per a fer-ho és donar la possibilitat de iniciar converses ràpides amb altres usuaris.

- **L'usuari ha de poder seleccionar altres usuaris com a amics per tal de poder iniciar converses instantànies**

Una part molt important de les converses instantànies és la facilitat que ofereixen per tal d'establir comunicacions. Per tant, ha de ser igual de fàcil poder cercar l'usuari amb el se es vol iniciar la conversa.

## 2.2. Requisits no funcionals

- **L'usuari ha de poder iniciar la sessió de forma segura.**

La *UOC* és un cas clar de sistema al núvol amb gran sensibilitat de dades. Per tant, qualsevol aplicació que n'utilitzi les seves dades també ha de ser conseqüent amb el nivell de seguretat de que gaudeixen els estudiants de la *UOC*.

- **L'usuari ha de poder identificar clarament cada aula i els seus apartats.**

No es suficient mostrar tota la informació, també s'ha de mostrar de forma que les aules quedin ben diferenciades, on es vegin clarament els seus apartats i que es pugui accedir de forma fàcil i intuïtiva als seus apartats.

- **El sistema ha de mostrar amb claredat amb quins elements es pot interactuar.**

Es molt comú veure pàgines on es gestiona malament la interacció de l'usuari. Els enllaços i botons han de ser molt clars i diferenciats de la informació plana per tal de que l'usuari no es senti mai perdut.

- **El sistema ha de ser coherent amb el idioma.**

El sistema utilitzarà diferents fonts d'informació, utilitats i gestió d'errors. Per tant, és important mantenir una continuïtat en el idioma.

- **El sistema ha de permetre navegar de forma fluida i sense punts morts.**

Un sistema de fòrums te el risc de propiciar la navegació endavant i d'oferir poques possibilitats de navegació endarrere o transversal. És fàcil caure en el parany de facilitar l'accés a un missatge en concret, però descuidar el retorn a la seva aula, a la conversació següent o simplement canviar d'aula.

- **El sistema ha de permetre una comunicació instantània fiable**

Un usuari ha de tenir la certesa que, si el sistema li indica que pot iniciar una conversa instantània amb un altre usuari, ho podrà fer amb la seguretat de que el missatge arribarà al receptor.

## 2.3. Diagrama de casos d'ús

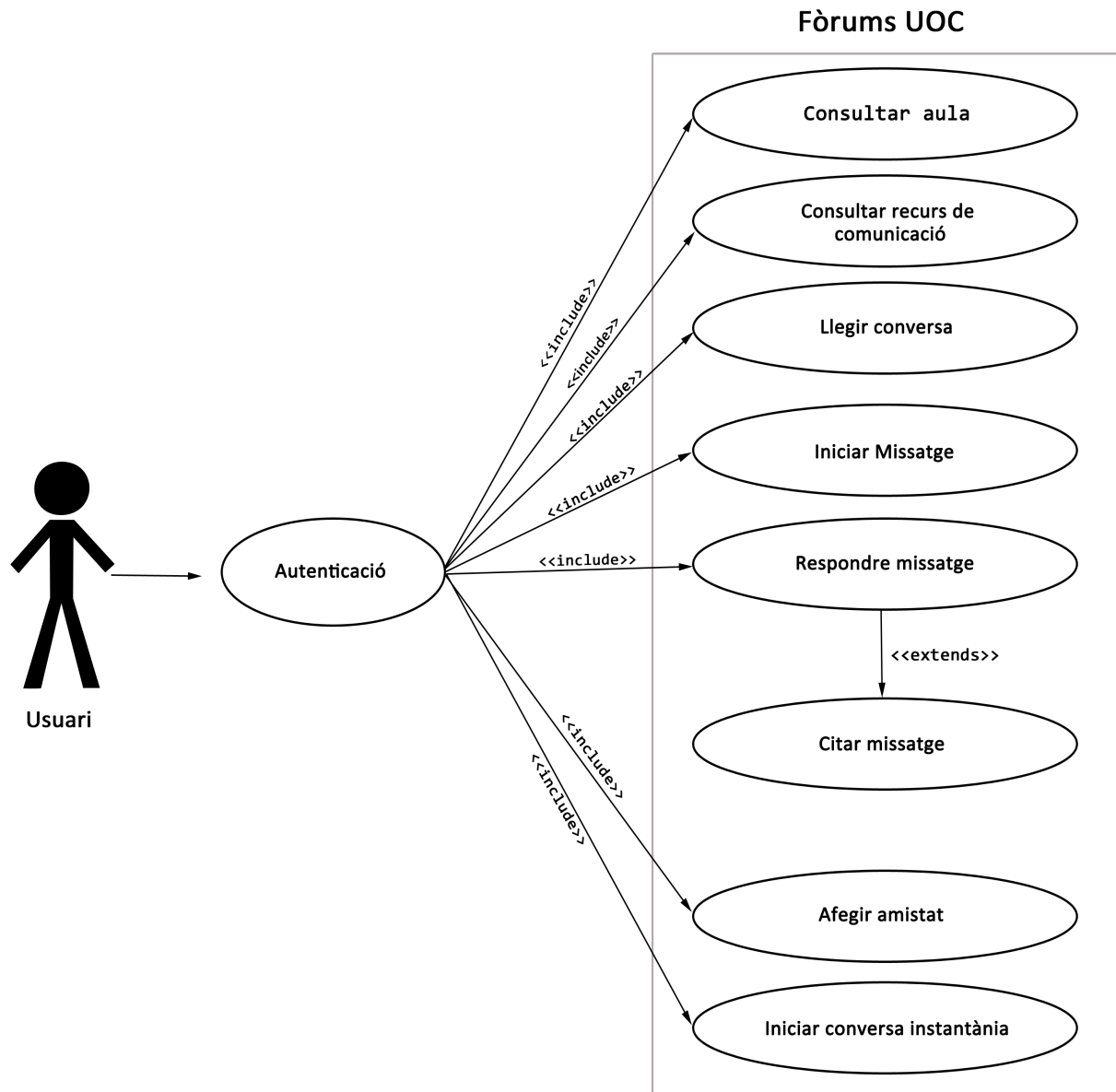


Figura 1. Diagrama de Casos d'ús

## 2.4. Casos d'ús

Cas d'ús	1. Entrar al sistema
<b>Actor principal</b>	Usuari
<b>Àmbit</b>	Sistema informàtic de lectura dels Fòrums UOC
<b>Nivell d'objectiu</b>	usuari
<b>Stakeholders i interessos</b>	
estudiant	Vol visualitzar les seves dades personals
<b>Precondició</b>	El usuari te un compte de la <i>UOC</i>
<b>Garanties mínimes</b>	-
<b>Garanties en cas d'èxit</b>	L'usuari inicia la sessió als Fòrums UOC
<b>Escenari principal d'èxit</b>	
1	L'usuari accedeix a l'aplicació
2	L'usuari accedeix a l'inici de sessió
3	El sistema el redirigeix al servei d'autenticació de la <i>UOC</i>
4	L'usuari introdueix les seves credencials
5	L'usuari escolleix quins permisos donarà a l'aplicació Fòrums UOC
6	El sistema d'autenticació de la <i>UOC</i> redirigeix a l'usuari als Fòrums UOC amb la seva sessió iniciada
<b>Extensions</b>	
4.1	Les credencials no són correctes i torna al pas 3
5.1	Els permisos no són suficients per a iniciar la sessió als Fòrums UOC i es mostra un missatge d'error

Cas d'ús	2. Visualitzar les aules disponibles
<b>Actor principal</b>	Usuari
<b>Àmbit</b>	Sistema informàtic dels Fòrums UOC
<b>Nivell d'objectiu</b>	Usuari
<b>Stakeholders i interessos</b>	
Usuari	Vol veure la informació disponible a l'aplicació

<b>Precondició</b>	L'usuari ha iniciat la sessió al sistema informàtic (Cas d'ús 1)
<b>Garanties mínimes</b>	-
<b>Garanties en cas d'èxit</b>	L'usuari visualitza les aules que te disponibles
<b>Escenari principal d'èxit</b>	
1	L'usuari accedeix al sistema informàtic dels Fòrums UOC
2	El sistema mostra totes les aules que l'usuari te disponibles i els recursos de comunicació que cada assignatura te associats
<b>Extensions</b>	
2.1	L'usuari no te aules disponibles i el sistema mostra un avís explicant la situació

<b>Cas d'ús</b>	<b>3. Revisar si hi han missatges nous</b>
<b>Actor principal</b>	Usuari
<b>Àmbit</b>	Sistema informàtic dels Fòrums UOC
<b>Nivell d'objectiu</b>	Usuari
<b>Stakeholders i interessos</b>	
Usuari	Vol identificar els recursos de comunicació que contenen missatges nous
<b>Precondició</b>	L'usuari ha iniciat una sessió vàlida als Fòrums UOC (Cas d'ús 1)
<b>Garanties mínimes</b>	-
<b>Garanties en cas d'èxit</b>	L'usuari coneix si existeixen recursos de comunicació amb missatges nous
<b>Escenari principal d'èxit</b>	
1	L'usuari accedeix al sistema informàtic dels Fòrums UOC
2	El sistema mostra les aules que l'usuari te disponibles i els recursos que tenen associades
3	El sistema mostra el recompte total de missatges que conté cada recurs de comunicació i també mostra, de forma destacada, el nombre de missatges nous
<b>Extensions</b>	
3.1	L'usuari vol conèixer quins són els missatges nous d'un recurs concret

3.2	El sistema mostra un detall de les conversacions que conte el recurs escollit i en mostra de forma destacada els que tenen actualitzacions
-----	--

Cas d'ús	4. Llegir una conversació
<b>Actor principal</b>	Usuari
<b>Àmbit</b>	Sistema informàtic dels Fòrums UOC
<b>Nivell d'objectiu</b>	Usuari
<b>Stakeholders i interessos</b>	
Usuari lector	Vol llegir un missatge contingut als recursos de comunicació de la <i>UOC</i>
Usuaris redactors	Volen que els seus missatges arribin als seus destinataris
<b>Precondició</b>	L'usuari ha iniciat una sessió vàlida als Fòrums UOC (Cas d'ús 1)
<b>Garanties mínimes</b>	-
<b>Garanties en cas d'èxit</b>	L'usuari visualitzarà el missatge que vol llegir
<b>Escenari principal d'èxit</b>	
1	L'usuari accedeix al sistema informàtic dels Fòrums UOC
2	El sistema mostra un llistat d'aules i recursos de comunicació
3	L'usuari escolleix un recurs de comunicació per a veure les seves converses
4	El sistema mostra les converses que conté un recurs de comunicació ordenades per data d'actualització descendent
5	L'usuari escolleix una conversa per a llegir els missatges que conté
6	El sistema mostra els missatges que conté una conversa ordenats de forma cronològica
<b>Extensions</b>	
5.1	La conversa que busca no es troba en la pàgina actual i l'usuari navega a una altra pàgina per cercar-la. Torna al pas 4
6.1	L'usuari no troba el missatge desitjat a la pàgina actual i navega a la pàgina següent per cercar-lo. Torna al pas 6.

Cas d'ús	5. Enviar un missatge nou
<b>Actor principal</b>	Usuari
<b>Àmbit</b>	Sistema informàtic dels Fòrums UOC
<b>Nivell d'objectiu</b>	Usuari
<b>Stakeholders i interessos</b>	
Usuari redactor	Vol enviar un missatge que pugui ser visualitzat per els altres usuaris que tinguin accés al recurs de comunicació
Lector	Vol visualitzar els missatges enviats per els altres usuaris
<b>Precondició</b>	L'usuari ha iniciat una sessió vàlida als Fòrums UOC (Cas d'ús 1)
<b>Garanties mínimes</b>	-
<b>Garanties en cas d'èxit</b>	L'usuari enviarà un missatge que serà desat al sistema de persistència de la <i>UOC</i>
<b>Escenari principal d'èxit</b>	
1	L'usuari accedeix al sistema informàtic dels Fòrums UOC
2	El sistema mostra un llistat d'aules i recursos de comunicació
3	L'usuari escolleix un recurs de comunicació per a veure les seves converses
4	El sistema mostra les converses que conté un recurs de comunicació ordenades per data d'actualització descendent
5	L'usuari vol enviar un missatge nou, i per tant, iniciar una nova conversa. Per a fer-ho es dirigeix a la secció de redacció on podrà introduir un tema i un contingut pel seu missatge
6	El sistema envia la petició a la <i>UOC</i> i es publica al recurs de comunicació seleccionat
<b>Extensions</b>	
5.1	L'usuari no te permís per a publicar missatges en el recurs de comunicació seleccionat (És estudiant i el recurs és un tauler de l'aula)

Cas d'ús	6. Respondre un missatge
<b>Actor principal</b>	Usuari
<b>Àmbit</b>	Sistema informàtic dels Fòrums UOC
<b>Nivell d'objectiu</b>	Usuari
<b>Stakeholders i interessos</b>	
Usuari redactor	Vol enviar un missatge que pugui ser visualitzat per els altres usuaris que tinguin accés al recurs de comunicació
Lector	Vol visualitzar els missatges enviats per els altres usuaris
<b>Precondició</b>	L'usuari ha iniciat una sessió vàlida als Fòrums UOC i està visualitzant una conversa (Cas d'ús 4)
<b>Garanties mínimes</b>	-
<b>Garanties en cas d'èxit</b>	L'usuari enviarà un missatge relacionat amb una conversa i serà desat al sistema de persistència de la <i>UOC</i>
<b>Escenari principal d'èxit</b>	
1	L'usuari accedeix a una conversa en concret
2	El sistema mostra tots els missatges de la conversa
3	L'usuari accedeix a la secció de resposta del missatge introduint el missatge desitjat
4	L'usuari envia el missatge
5	El sistema envia la petició a la <i>UOC</i> i es publica al recurs de comunicació que conté la conversa
<b>Extensions</b>	
-	-

Cas d'ús	7. Respondre un missatge citant una aportació anterior
<b>Actor principal</b>	Usuari
<b>Àmbit</b>	Sistema informàtic dels Fòrums UOC
<b>Nivell d'objectiu</b>	Usuari
<b>Stakeholders i interessos</b>	
Usuari redactor	Vol enviar un missatge que pugui ser visualitzat per els altres usuaris que tinguin accés al recurs de

Lector	comunicació Vol visualitzar els missatges enviats per els altres usuaris
<b>Precondició</b>	L'usuari ha iniciat una sessió vàlida als Fòrums UOC i està visualitzant una conversa (Cas d'ús 4)
<b>Garanties mínimes</b>	-
<b>Garanties en cas d'èxit</b>	L'usuari enviarà un missatge relacionat amb una conversa i serà desat al sistema de persistència de la <i>UOC</i>
<b>Escenari principal d'èxit</b>	
1	L'usuari accedeix a una conversa en concret
2	El sistema mostra tots els missatges de la conversa
3	L'usuari selecciona el missatge que vol citar
4	El sistema dirigeix a l'usuari a la secció de resposta on ha introduït el text citat utilitzant un format especial
5	L'usuari envia el missatge
6	El sistema envia la petició a la <i>UOC</i> i es publica al recurs de comunicació que conté la conversa
<b>Extensions</b>	
1.1	El recurs de comunicació no té cap conversa i no pot citar cap missatge

Cas d'ús	8. Iniciar una conversació de missatgeria instantània
<b>Actor principal</b>	Usuari
<b>Àmbit</b>	Sistema informàtic dels Fòrums UOC
<b>Nivell d'objectiu</b>	Usuari
<b>Stakeholders i interessos</b>	
Usuari emissor	Vol enviar un missatge instantani a un altre usuari
Usuari receptor de la conversa	Vol rebre el missatge que li estan escrivint en temps real
<b>Precondició</b>	L'usuari ha iniciat una sessió vàlida als Fòrums UOC (Cas d'ús 1)
<b>Garanties mínimes</b>	-
<b>Garanties en cas d'èxit</b>	L'usuari receptor rebrà un missatge en temps real
<b>Escenari principal d'èxit</b>	



1	L'usuari accedeix a la eina de missatgeria instantània
2	L'usuari busca a un altre usuari amics per iniciar la conversa
3	El sistema li mostra la llista d'amics connectats
4	L'usuari selecciona un usuari de la llista
5	El sistema obre una nova conversa instantània
6	L'usuari escriu un nou missatge
7	El sistema envia el missatge a l'usuari receptor i obra una nova conversa en la seva eina de comunicació instantània
<b>Extensions</b>	
3.1	No hi ha cap amic connectat
3.2	L'usuari ha de seleccionar amics abans de començar converses instantànies o cercar-los en els fòrums

<b>Cas d'ús</b>	<b>9. Respondre un missatge citant una aportació anterior</b>
<b>Actor principal</b>	Usuari
<b>Àmbit</b>	Sistema informàtic dels Fòrums UOC
<b>Nivell d'objectiu</b>	Usuari
<b>Stakeholders i interessos</b>	
Usuari	Vol afegir un amic per poder realitzar converses instantànies
<b>Precondició</b>	L'usuari ha iniciat una sessió vàlida als Fòrums UOC (Cas d'ús 1)
<b>Garanties mínimes</b>	-
<b>Garanties en cas d'èxit</b>	L'usuari afegeix un amic a la llista
<b>Escenari principal d'èxit</b>	
1	L'usuari accedeix a un recurs de comunicació
2	El sistema mostra les converses existents en el recurs
3	L'usuari selecciona un usuari per a ser afegit a la llista d'amistats
4	El sistema guarda a l'usuari com a amic
<b>Extensions</b>	
3.1	L'usuari cercat no es iniciador de cap conversa

3.2	L'usuari selecciona una conversa
3.3	El sistema mostra els missatges de la conversa seleccionada
3.4	L'usuari selecciona un usuari per a ser afegit a la llista d'amistats
3.5	El sistema guarda a l'usuari com a amic

# 3. Disseny

## 3.1. Model conceptual

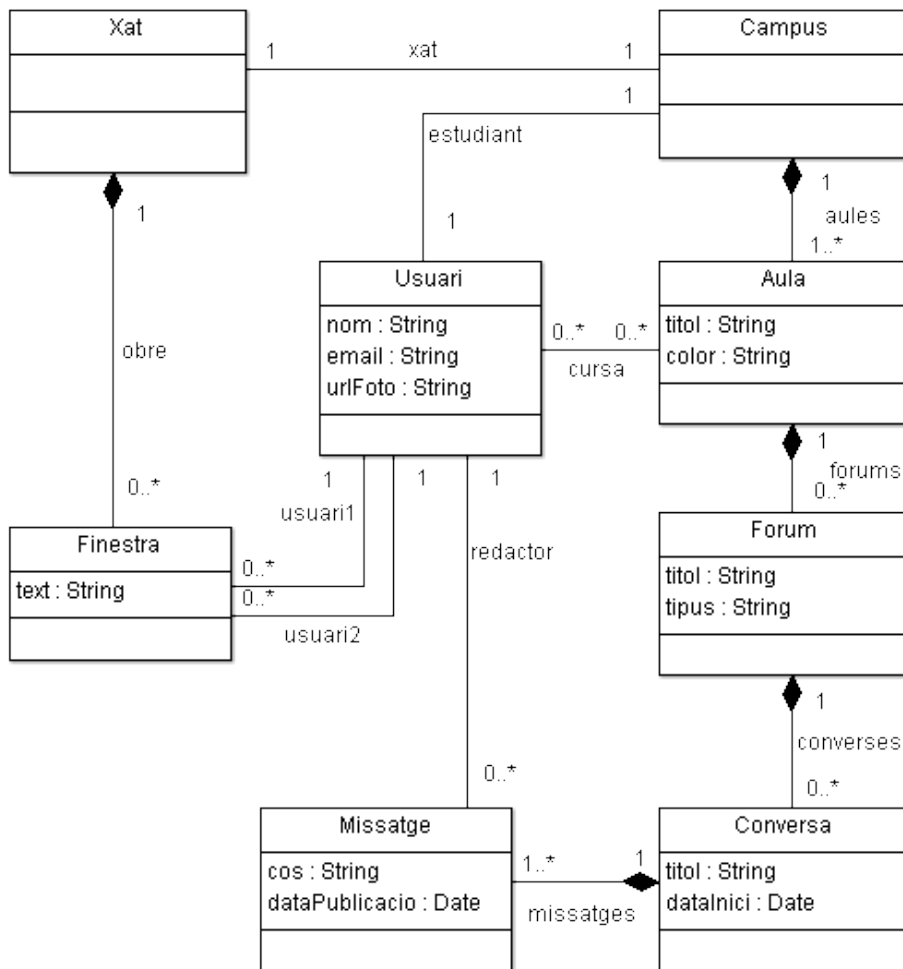


Figura 2. Model conceptual

En el model conceptual podem veure com tota l'estructura està dissenyada per donar suport comunicatiu a l'usuari, i per això n'és una entitat central. Alhora podem veure com hi ha 2 blocs comunicatius diferenciats a ambdues bandes d'usuari.

També s'ha de destacar el campus com a nexa d'unió entre l'usuari i les eines de comunicació. Per tant el campus serà la plataforma que permetrà a l'usuari accedir als recursos de comunicació dels que pot disposar.

Per un costat tenim el recurs de comunicació de les aules i els seus fòrums de comunicacions. Cada estudiant té accés a un nombre d'aules relacionades amb els estudis que està cursant en un moment donat. Igualment, per la naturalesa dels estudis en xarxa, cada aula té assignats uns recursos de comunicació que com a mínim són un taulell i un fòrum. Aquests recursos de comunicació estan formats per converses. Les converses són grups de missatges relacionats entre ells que esdevenen dintre d'un fòrum i ajuden a mantenir el seguiment d'un fil de comunicació iniciat per un missatge concret. Al seu temps, els missatges són aportacions dels usuaris que es caracteritzen per tenir un text que forma el cos del missatge i que estan enviats per un usuari i data concreta.

En el campus podem trobar la zona d'aules on hi ha la missatgeria persistent, però també es pot accedir a un sistema de missatgeria instantània. Aquest sistema permet iniciar converses instantànies amb qualsevol usuari de la *UOC* que disposi d'una sessió activa als **Fòrums UOC**. Per tant, per a fer-ho necessitarà estar en el mateix instant veient l'aplicació. Si es compleix el cas, l'usuari pot demanar a la eina de xat que obri una finestra de comunicació entre els 2 usuaris.

## 3.2. Diagrama d'arquitectura

### 3.2.1. Diagrama d'arquitectura de maquinari

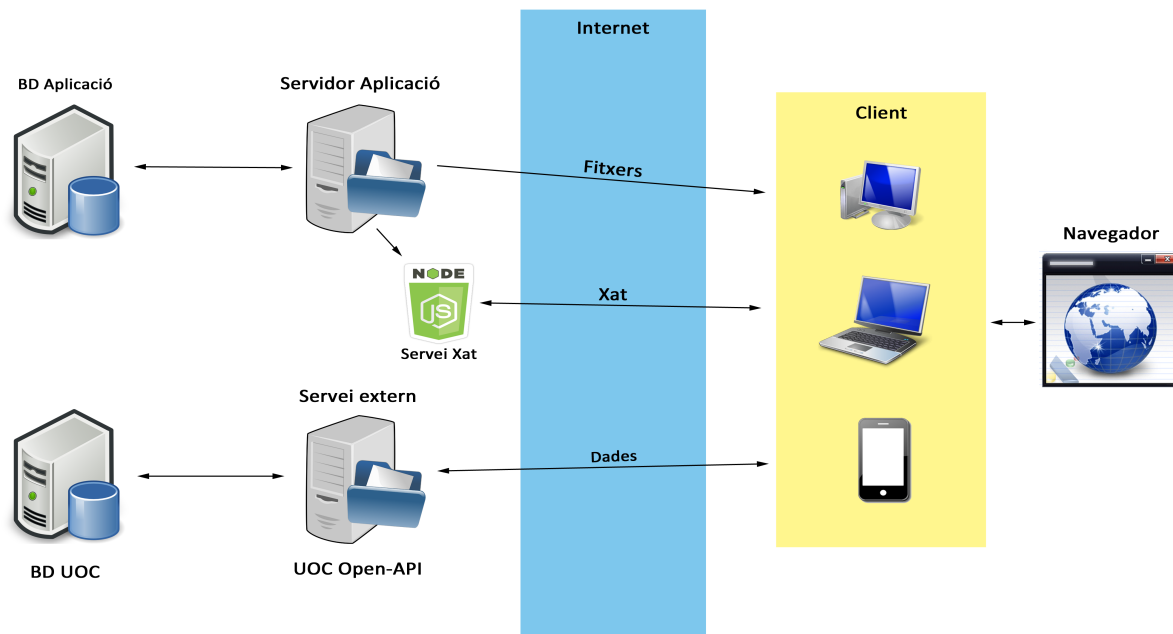


Figura 3. Diagrama d'arquitectura de maquinari

**Fòrums UOC** està dissenyada per a fer un ús casi exclusiu de la part de client (Front-end) per a la execució. Això vol dir que la gran majoria de procediments s'executaran en la màquina del client i que el servidor només s'ocuparà de servir els fitxers necessaris i les dades que demani l'aplicació en cada moment.

Com podem veure al diagrama d'arquitectura, l'aplicació es serveix des d'un servidor central. En aquest servidor es troben tots els fitxers que utilitzarà el client web per a poder mostrar l'aplicació, maquetar-la i fer que es comporti com ha de comportar-se.

Aquest servidor també té la tasca de comunicar-se amb la base de dades que té informació persistent de les dades que es necessiten per servir la eina de xat. Ahirora el xat també correrà en aquest servidor, això significa que el servidor també disposarà d'un dimoni que sempre estarà actiu amb el servei de xat activat per a ser servit sota demanda del client. Aquest canal serà bidireccional ja que el client enviarà les seves dades de connexió juntament amb els possibles missatges que vulgui enviar a un altre usuari.

Tota aquesta informació del servidor central la reben els clients. El client de la nostra aplicació és un navegador web corrent en qualsevol sistema informàtic que disposi d'una interfície gràfica per visualitzar les dades i poder interactuar amb ella. Per tant,

el client rebrà els fitxers del servidor i serà interpretada per tal de poder mostrar l'aplicació i que s'executi com cal.

Una de les primeres tasques que ha de fer el client és demanar autorització als servidors de informació oberta de la *UOC* per a demanar informació. Per tant també haurà de ser capaç d'accedir als servidors de la *UOC* per tal de accedir a les dades de l'usuari que vol interactuar amb l'aplicació de **Fòrums UOC**.

### 3.2.2. Diagrama d'arquitectura de programari

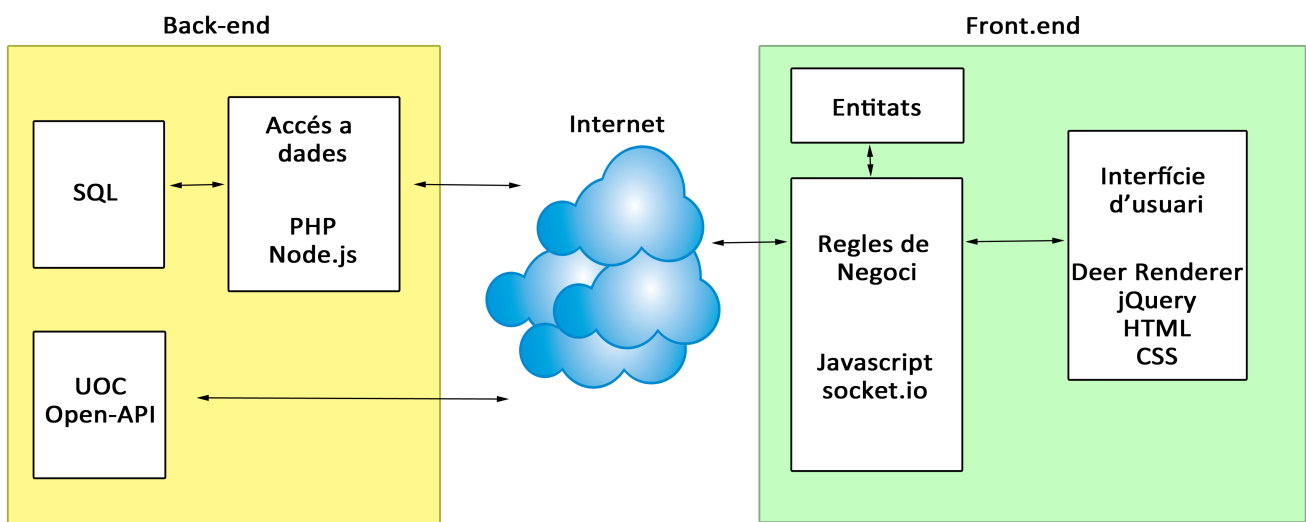


Figura 4. Diagrama d'arquitectura de programari

Basem la arquitectura de l'aplicació en una aplicació web que treballa en el client, és a dir, la lògica de negoci i de presentació es troba en el *Front-end*. Com no pot ser d'altra forma, s'utilitzarà el *HTML*, *CSS*, i *Javascript* per organitzar les diferents capes de les que s'haurà d'ocupar el client.

La capa de presentació està gestionada purament per el client, utilitzant el *HTML* i funcions de transformació dels seus mòduls mitjançant *Javascript* i alguns mòduls més avançats.

De la lògica de negoci se n'ocuparà íntegrament el client mitjançant la comunicació asíncrona amb el serveis de persistència i les pròpies regles establertes que esdevindran depenent de la interacció amb el usuari.

En canvi, el sistema de persistència està dividit en 3 apartats principals:

- Primer tenim un senzill sistema de persistència que es basa en un simple mecanisme de clau-valor ja existent en el mateix client que ens servirà per desar informació de la sessió de l'usuari.
- Per altra banda tenim el sistema de persistència del nostre servidor, que ens ajudarà a millorar les prestacions que ofereix la *UOC* i ens permetrà servir amb més eficàcia la missatgeria instantània.
- Finalment tenim el sistema de persistència extern que ens ofereix el servei *Open-API* de la *UOC*. En aquest servei podrem autenticar a un usuari i recuperar la informació del campus virtual d'aquest usuari i així nodrir els elements interactius de **Fòrums UOC**.

Pel que fa al servidor, s'ocuparà principalment de servir les peticions de fitxers que li facin els clients. No obstant també s'haurà d'encarregar de gestionar la persistència de dades de les que és encarregat de manejar (totes les dades que la funcionalitat de la *UOC* no ofereix).

A demés també ha de mantenir el sistema de comunicació instantània sempre disponible. Això ho farà gràcies a un sistema de execució oberta que estarà sempre escoltant els ports de comunicació per si s'inicia alguna comunicació des del client. El client, és a dir el navegador, es comunicarà amb el servidor a través de *WebSockets*, un tipus de comunicació *TCP* especial per als navegadors web.

### 3.3. Diagrama E-R

Tot i que **Fòrums UOC** es nodreix de les dades que li proporciona la *UOC*, el disseny de l'eina de missatgeria instantània necessita ampliar les necessitats de persistència per mantenir un registre dels usuaris que inicien sessió i fer-ho d'una forma segura tant per l'aplicació com per els mateixos usuaris. Alhora, es necessita mantenir un registre dels usuaris que han indicat que volen mantenir una relació d'amistat amb altres usuaris.

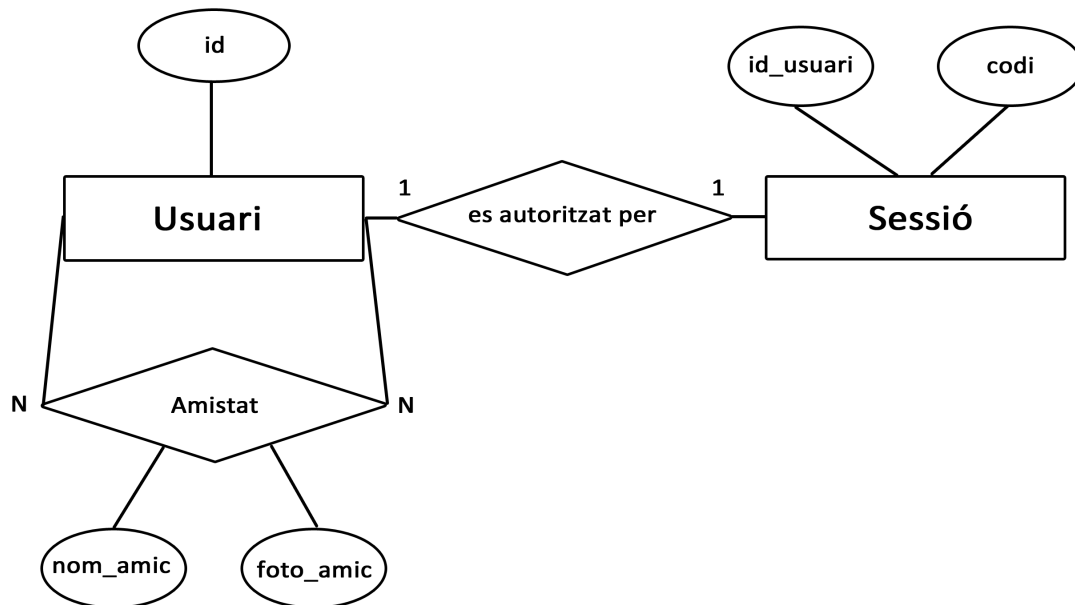


Figura 5. Diagrama E-R

Com es pot apreciar al diagrama, disposarem d'una entitat que hem anomenat sessió, aquesta entitat serà l'encarregada de registrar les sessions dels usuaris. Aquestes sessions s'utilitzaran únicament per comprovar la legitimitat d'accés a la eina de missatgeria instantània, per tant no necessitarem més informació que l'identificador de l'usuari i el codi intern de comprovació. També es pot veure que la relació amb l'usuari és unitària, és a dir, un usuari només pot tenir una sessió vàlida en cada moment. Per tant, un usuari només pot tenir un codi, i un codi només pot estar associat a un usuari.

També podem veure la naturalesa de la relació d'amistat. Aquesta relació uneix un usuari amb un altre usuari per a que el segon quedi relacionat amb el primer. La relació té una multiplicitat múltiple, pel que un usuari pot tenir molts amics, i un usuari pot ser amic de molts usuaris. S'ha de notar que un usuari i un amic només poden existir un cop.

Finalment, cal destacar que tot i existir la entitat Usuari en el diagrama, és una entitat que no existirà a la pràctica en el nostre sistema, ja que la gestió dels usuaris ve controlada per les dades externes proporcionades per la *UOC*.



### 3.4. Model de classes

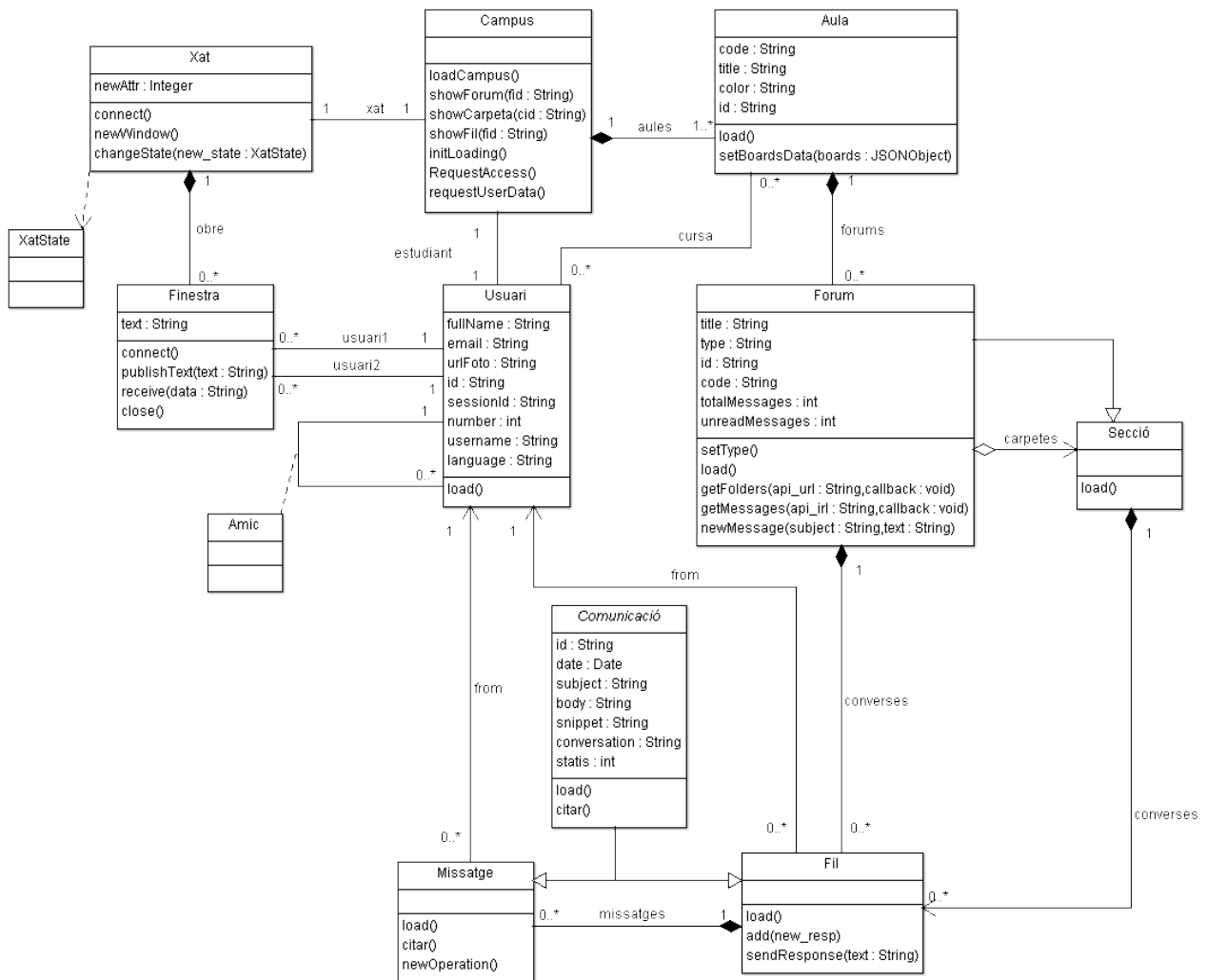


Figura 6. Model de Classes

La classe central del sistema és **Campus**. És la encarregada d'inicialitzar el sistema i preparar l'aplicació preparada per a que l'usuari pugui iniciar la sessió. Alhora s'encarrega de gestionar els canvis en la interfície d'usuari i de controlar l'estat de les diferents eines de les que disposarà l'usuari: **Xat** i sistema de fòrums.

L'inici de sessió també és gestionat per la classe arrel **Campus**, que s'encarregarà de respondre a l'inici de sessió i de carregar les dades de l'usuari una vegada comprovi que la sessió s'ha inicialitzat de forma correcta. En el mateix moment es comença la càrrega de les aules.

El **Campus** té associada una col·lecció d'**Aula**. Aquesta, per la seva banda serà la encarregada de gestionar els seus propis recursos de comunicació i inicialitzar-los. També tenen associada una instància de l'usuari per tal de poder accedir a les seves dades amb més facilitat.

Una vegada seleccionem un Fòrum, aquest serà l'encarregat de carregar les seves dades. Un Fòrum conté Fils i també pot contenir Seccions. Les Secció és una classe específica de Fòrum que no pot contenir fòrums. És a dir, els Fòrums poden tenir fills, però les Seccions no. Ambdós són una composició de Fil. Un Fòrum serà l'encarregat de iniciar un nou Fil, el que significa, que enviarà missatges nous.

Els Fils són converses, per tant, composicions de Missatge. Alhora Fil i Missatge hereten de Comunicació *Comunicació* és una classe abstracta que agrupa la gran majoria de atributs dels diferents tipus de entitats de resposta d'un recurs de comunicació. Aquestes classes es diferencien en que Fil es un missatge iniciador de conversa, i que per tant, te una col·lecció de Missatge associats.

Fil te la responsabilitat de afegir nous Missatges a la seva col·lecció quan s'està inicialitzant el Fòrum i també de gestionar les respostes que es produeixen dintre del mateix Fil. Comunicació defineix uns mètodes comuns per als seus fills que poden usar independentment del tipus d'instància que siguin. Ambdues classes derivades tenen associada una instància de la classe Usuari que representarà el emissor de la comunicació.

El Campus també te associat una instància de Xat, la eina de missatgeria instantània que s'inicialitzarà i es connectarà amb el servidor tant bon punt l'usuari s'autentifiqui. El Xat gestionarà la seva màquina d'estats que regularà la visibilitat de l'usuari envers altres usuaris del sistema. També serà l'encarregat d'iniciar noves finestres de comunicació quan algun Usuari vulgui iniciar una conversa. En aquest cas, el Xat instanciarà una nova Finestra i aquesta es connectarà amb el servidor per poder enviar i rebre comunicacions a través dels mètodes adients.

Un Usuari pot tenir una sèrie d'Amics que és una associació entre Usuari i Usuari que ens ajudarà a establir noves converses instantànies de forma més senzilla.

### 3.5. Disseny de la interfície d'usuari

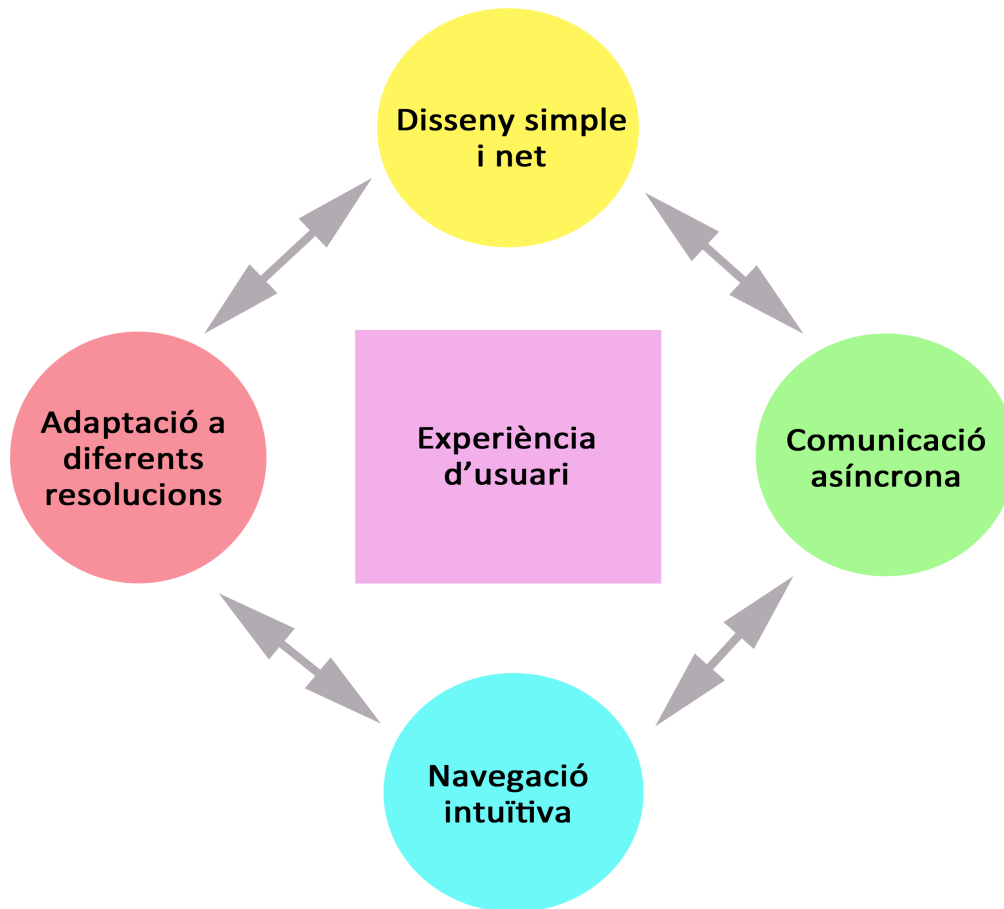


Figura 7. Esquema d'interfície d'usuari

El disseny de la interfície d'usuari s'ha volgut basar en millorar la experiència de l'usuari al visitar l'aplicació. Això implica tenir en consideració alguns punts claus per a que un visitant sigui capaç d'accedir a l'aplicació, navegar i executar la tasca que desitgi de forma fàcil, ràpida i efectiva.

La usabilitat d'una aplicació web, i en general de qualsevol interfície gràfica es valora segons alguns indicadors clau:

- **Facilitat d'aprenentatge.** Valora la dificultat que troben els usuaris d'executar una tasca la primera vegada que es troben amb una situació. Una bona estratègia a adoptar proclama que si una funcionalitat necessita anar acompanyada d'un text d'ajuda, es que no és prou usable.
- **Eficiència.** Valora el temps que tarda un usuari en dur a terme una tasca de forma satisfactòria.
- Capacitat de ser recordat. Ens indica si un usuari seria capaç de recordar el procediment per dur a terme una tasca si ja l'ha dut a terme prèviament. És el pas següent a la facilitat d'aprenentatge.

- **Eficàcia.** Ve definida per la quantitat d'errors que comet o pot cometre un usuari al realitzar una tasca. Si aquest varem no es compleix, significarà que hem de guiar de forma més efectiva el procés de l'usuari evitant que pugui cometre errors i no trencar així el seu camí pel procés.
- **Satisfacció.** Aquest indicador és més subjectiu i pot servir de resum dels anteriors. Indica com de agradable ha sigut el procés per a l'usuari. Sol estar molt lligat amb la senzillesa del procés i la percepció de conèixer en tot moment tant on s'està com quin serà el pròxim pas.

Per a aconseguir aquests objectius s'ha dissenyat la interfície d'usuari pensant en quatre factors, a través dels quals s'ha construït la capa de presentació de la aplicació.

### 3.5.1. Disseny simple i net

El disseny gràfic d'una aplicació es un punt molt important dintre de la percepció d'usabilitat. Un disseny adequat en cada context, estudiant el públic objectiu i les intencions dels mateixos es torna bàsic. Aquest fet no implica que hi hagi un disseny global que s'adeqüi a totes les aplicacions, cada cas tindrà les seves particularitats i necessitarà de solucions diferents.

A **Fòrums UOC** s'han identificat les necessitats principals dels usuaris com a cercar informació i comunicació. Aquestes necessitats generen unes característiques de l'aplicació molt concretes en les que podem destacar: gran quantitat de contingut textual i grups d'informació agrupats.

Tenint en compte aquests factors, s'ha optat per proveir una estructura minimalista, sense càrrega en la estètica buscant donar força a la informació. D'aquesta forma es pot maximitzar la visibilitat del contingut i millorar la forma en que es presenta la informació, facilitant així la lectura i l'accés als mateixos continguts.

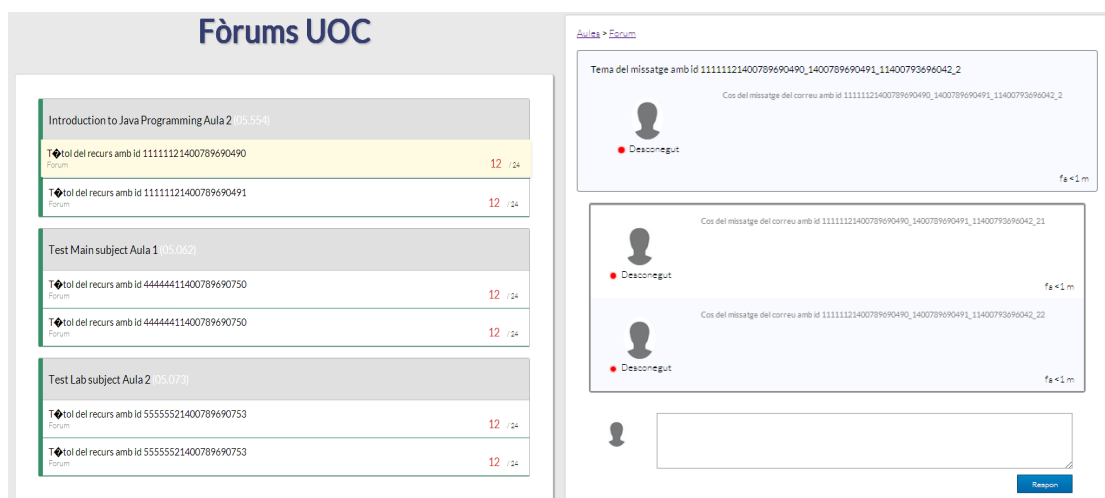


Figura 8. Pantalles de l'aplicació

Alhora, al ser una aplicació que recull informació de la UOC, s'ha mantingut la línia de disseny que adopta la universitat fent prevaldre els colors blanc, blau marí i algunes tonalitats de gris per a organitzar els diferents apartats. Amb aquest fet, l'usuari es pot sentir més familiar amb un medi que transmet sensació de comunitat amb les eines que acostuma a usar com a estudiant de la UOC. De igual forma s'ha utilitzat els identificadors de color de cada assignatura per diferenciar les diferents assignatures que conformen el campus de l'estudiant, fet que ajuda a seleccionar la informació a primer cop d'ull i a mantenir la coherència amb els procediments de la universitat.

S'ha aprofitat la disponibilitat de dades de l'usuari per a personalitzar l'aspecte de les diferents pantalles amb la intenció de familiaritzar a l'usuari amb l'estètica de l'aplicació.

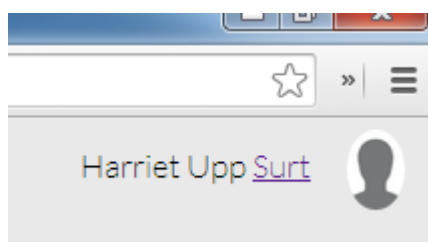


Figura 9. Detall de la personalització de l'usuari en l'aplicació

Per a donar vida a la navegació, s'ha intentat realitzar canvis bruscs de visualització, buscant sempre un pas gradual que no des-ubiqui a l'usuari. Alhora s'ha aprofitat la informació personal dels diferents usuaris que participen a l'aula, així com del seu ús de l'aplicació per tal de afegir riquesa visual a la presentació i generar dinamisme en el flux de navegació.

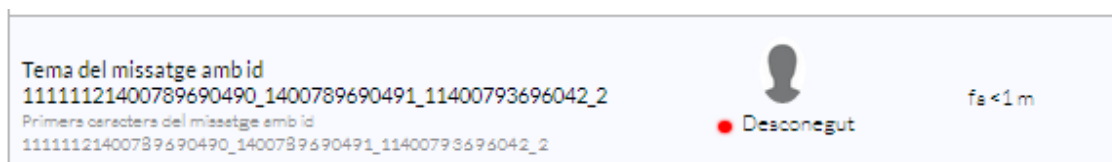


Figura 10. Detall d'un element en l'aplicació

Finalment, l'apartat que oferia més llibertat a l'hora de organitzar la seva distribució era la eina de missatgeria instantània. Aquesta eina comporta una interacció més directa amb un gran component social, per tant s'ha pres com a referència com estan enfocant aquest tipus de comunicació les principals xarxes socials. En la majoria dels casos es decanten per oferir una secció a la part inferior de la pantalla on estan lligades les converses, i situant una llista de usuaris connectats a la dreta de la pantalla. En el cas que ens ocupa, s'ha procedit de forma similar, situant una barra inferior on es situaran les converses que hi hagin disponibles. El disseny de les finestres és clar i senzill, amb la informació suficient per a que l'usuari pugui prendre accions sense error. Fent que l'usuari realitzi tasques de la forma que ja coneix, redueix molt la corba d'aprenentatge de la eina, redueix la frustració de l'usuari i n'augmenta la satisfacció.

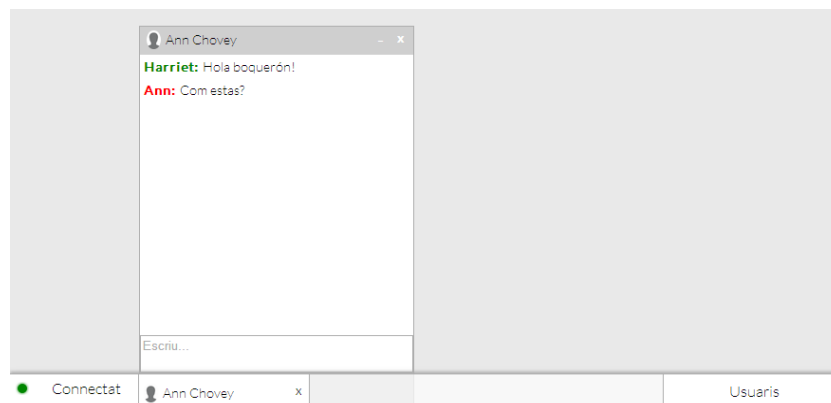


Figura 11. Detall de la eina de missatgeria instantània

Junt amb la barra de converses de xat, també hi ha a disposició de l'usuari un petit menú de connexió i la llista d'usuaris connectats. El menú de connexió permet saber en tot moment en quin estat es troba, ajudant de forma visual a entendre el significat per a conèixer la situació amb un sol cop d'ull. La llista d'usuaris, per la seva banda, mostra un llistat d'usuaris connectats. Aquesta llista està organitzada per assignatures, identificades per el seu color, en aquest espai trobarem també els usuaris marcats com a amistsats. Com en tota l'aplicació, els usuaris van acompanyats de la seva imatge personal, per a facilitar la identificació i donar riquesa visual al contingut, així com la possibilitat d'afegir-los com a amistsats.

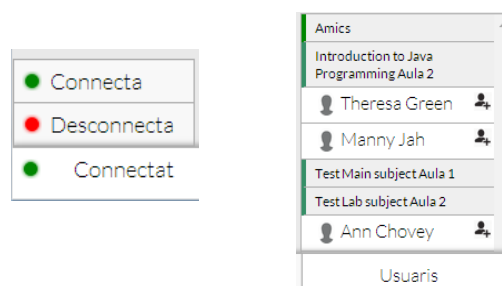


Figura 12. Detall de menús en la eina de missatgeria instantània

### 3.5.2. Adaptació a diferents resolucions

Aquesta decisió fa referència a un problema que ha existit des dels inicis d'Internet, i que sembla que ha adquirit més protagonisme en aquests últims anys degut a la popularització dels telèfons intel·ligents i tauletes electròniques. I és que avui en dia existeixen dispositius capaços d'accedir a Internet de gairebé totes les combinacions de altura i amplada possibles. Fet que dificulta molt com s'ha de mostrar la informació per tal de no deixar cap dispositiu sense poder accedir a l'aplicació.

Aquest fet s'ha tingut molt en compte a **Fòrums UOC** ja que la universitat disposa de plana web específica per a dispositius mòbils, i la naturalesa de la **UOC** implica que hi haurà molt ús de la plataforma des de qualsevol tipus de dispositiu.

No obstant dintre de la planificació del projecte no hi va trobar cabuda el desenvolupament d'una interfície especial per a dispositius mòbils, part que queda com a treball futur per a millorar la accessibilitat. De totes formes, l'aplicació es pot visualitzar correctament en pantalles més petites.

### **3.5.3. Navegació intuïtiva**

La navegació intuïtiva engloba dos factors molt importants. Que l'usuari sigui capaç d'escollir de forma correcta quin és el següent pas que ha de donar per a satisfer les seves necessitats, i que l'usuari disposi de les eines necessàries per a dur les accions que desitgi en tot moment.

En aquest concepte s'uneixen molts factors que ens ajuden a decidir tant quines eines són necessàries en cada moment, com quina aparença, posició i importància han d'adquirir.

Els elements de navegació han d'incorporar una diferenciació clara amb el contingut purament informatiu, i mantenir una coherència en tota l'aplicació. D'aquesta forma, si de bon principi identifiquem de forma única la aparença de botons i enllaços, l'usuari serà capaç de trobar-los amb més facilitat. Aquesta reflexió no indica que tots els elements d'interacció hagin de ser botons, de fet la interfície d'usuari queda molt més neta si podem estalviar-nos alguns botons de navegació i transformar blocs d'informació, en blocs duals que reaccionin al moviment de l'usuari i actuïn tant proveint informació com oferint funcionalitat.

També s'ha tingut molt en compte factors per-adquirits per factors culturals i d'educació. La nostra lectura esdevé de la part superior esquerra a la part inferior dreta d'un document, aquest fet també es pot observar en ullades ràpides a una informació concreta, on la lectura es produeix "en diagonal". Tenint en compte aquest aspecte sabem que la informació més rellevant -com podria ser un títol o una capçalera- s'ha de situar a la part superior esquerra. En canvi, un element que indiqui continuació o seguiment s'haurà de situar a la part inferior dreta ja que és on s'ha acaba la cerca ràpida o la lectura exhaustiva. Si un usuari no ha trobat la informació, ho sabrà en aquest punt i ja tindrà en un radi d'acció proper la possibilitat de seguir la seva cerca. Un element de retrocés -com un botó de tornar a un nivell jeràrquic superior de la informació- s'ha de situar a la part esquerra de la interfície per dos motius: és on comença aquesta informació, i té un símil amb la lectura de passar la pàgina d'un llibre endarrere. Aquesta funcionalitat pot situar-se a parts més superiors si el retrocés és jeràrquic o a una situació més central o inferior si es tracta d'un retrocés en un mateix nivell (pàgina anterior d'un fòrum p. ex.).

És important destacar que aquesta organització de la navegabilitat i accessibilitat és molt dependent de la cultura occidental. Un usuari àrab, acostumat a llegir de dreta a esquerra, o un usuari japonès acostumat a llegir en vertical, podrien no tenir les facilitats inconscients que disposaria un estudiant europeu.

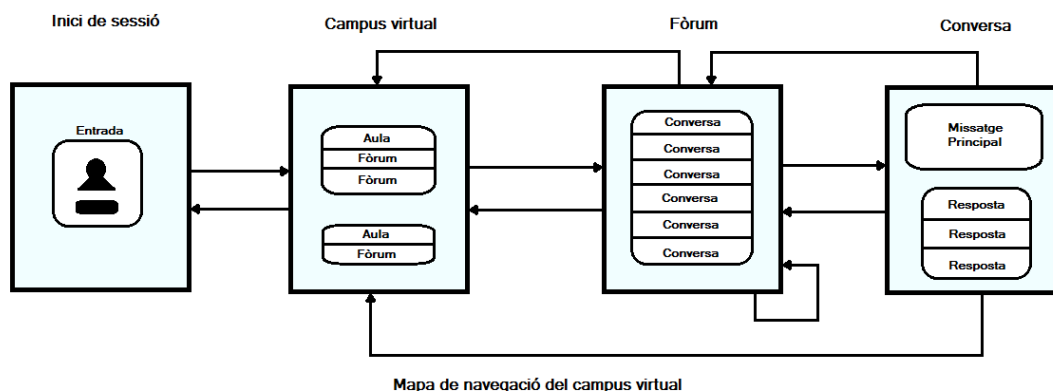


Figura 13. Mapa de navegació del campus virtual

### 3.5.4. Comunicació asíncrona

Tot i que la comunicació asíncrona pot semblar un paradigma més relacionat amb la lògica de negoci -que també ho és-, té una gran repercussió amb l'experiència d'usuari degut a que aporta un nivell d'interacció amb l'usuari que del que no podria disposar una aplicació web.

Gràcies a tècniques de *scripting* que s'executen en el navegador, som capaços de canviar la interfície d'usuari per a reaccionar a les seves accions de forma immediata i sense necessitat de refrescar la pàgina. Aquestes tècniques inclouen la comunicació asíncrona amb el servidor, que no solament ofereix interacció entre l'aplicació i el client, sinó que també permet comunicar al usuari amb el servidor i intercanviar informació sense que l'usuari perdi la visualització i la capacitat d'us de la interfície d'usuari i de la informació que aquesta conté.

A més a més, aquesta aproximació notablement més localitzada al client, obliga a dissenyar una estructura que cuidi a l'usuari en tot moment. Que la comunicació sigui asíncrona no significa que no hi hagi comunicació ni temps d'espera, sinó que aquesta espera no té perquè bloquejar l'experiència d'usuari. De totes formes no s'ha de deixar a l'usuari desemparat i s'ha de notificar l'estat de les càrregues d'informació que ha provocat el seu pas per l'aplicació, d'altra forma es podria donar el cas de que no sàpiga que està passant i es senti perdut en l'aplicació.

Si es gestiona correctament, la comunicació asíncrona és una de les úniques aproximacions que ens permeten mantenir la sensació de continuïtat en la experiència d'usuari, qualitat que redueix l'estrès en l'ús de la web i permet mantenir el centre d'atenció de l'usuari en la interfície.



## 4. Desenvolupament

### 4.1. Avaluació tecnològica

L'aplicació s'ha perfilat com una aplicació web, per tant ha de poder ser accessible des d'un navegador. És a dir, independentment de la tecnologia que serveixi el contingut, el resultat final ha de consistir en un conjunt de pàgines *HTML*, scripts i recursos (imatges, fulles d'estils...).

Ahora, al tractar-se d'una aplicació web, es necessitarà un sistema informàtic connectat a la xarxa que pugui donar servei als clients que vulguin connectar-se en aquest entorn. El servei de missatgeria instantània també dependrà d'aquest servei ja que necessitarà un servidor central que controli les converses.

La API de dades que ofereix la *UOC* és un servei que permet accedir a dades concretes de la universitat després de accedir a l'autorització d'un usuari. Aquestes dades s'obtenen a través de crides *REST*<sup>5</sup> al servidor de la universitat. La resposta pot venir en 2 formats diferents: *JSON*<sup>6</sup> o *XML*<sup>7</sup>, ambdós protocols d'intercanvi d'informació molt utilitzats en entorns web.

Per començar avaluarem les possibilitats que hi han per gestionar el sistema informàtic que proveirà el servei. Qualsevol ordinador pot ser utilitzat com a servidor a la xarxa si està correctament configurat. Malauradament, la oferta de les operadores (*IPs* dinàmiques) i les capacitats de un domicili (poca fiabilitat del sistema elèctric, poc rendiment dels ordinadors personals) fan que aquesta idea sigui poc atractiva. A la xarxa hi han moltes empreses que ofereixen diferents plans d'allotjament web tant per a empreses com per a particulars, juntament amb diferents escales de preus i serveis.

Per a les nostres necessitats, no necessitem un gran poder de processament ja que serà una aplicació amb càrrega relativament baixa de transferència de dades. No obstant, les característiques de la comunicació instantània pot requerir configuracions més avançades que no s'ofereixen als plans d'allotjament més assequibles. A més a més, s'ha de tenir en compte que seria molt adequat adquirir un domini públic per tal de facilitar l'accés al públic general.

L'aplicació s'ha de servir com a un conjunt de pàgines *HTML* amb recursos. Hi han moltes tecnologies de servidor que ens permeten la generació de documents d'aquest tipus per a ser enviats per la xarxa (*PHP, Python, Java, Ruby, ASP.NET...*). Totes serien adequades per a realitzar aquest projecte. De totes formes, no és necessària la utilització d'un llenguatge de servidor per a realitzar el projecte.

---

5 [http://es.wikipedia.org/wiki/Representational\\_State\\_Transfer](http://es.wikipedia.org/wiki/Representational_State_Transfer)

6 <http://json.org/>

7 [http://www.w3schools.com/xml/xml\\_what\\_is.asp](http://www.w3schools.com/xml/xml_what_is.asp)

Aquest projecte es pot enfocar de 2 formes principalment: que la carrega del processament la dugui a terme el servidor i envii al navegador del client un document, o que el servidor envii un document amb capacitat d'actualitzar el seu contingut mitjançant el llenguatge de *scripting* de navegador *Javascript*. En aquest cas, es percebria la pàgina com una aplicació que no es refresca, però que es modifica amb la interacció de l'usuari.

No obstant, sempre serà necessari un llenguatge de *scripting* per la secció de comunicació instantània, ja que ha de actualitzar-se sense necessitat de d'interacció de l'usuari ni cap refresc del contingut.

Seguint amb el raonament anterior, seran necessàries una infraestructura addicional per poder donar suport a la missatgeria instantània. Actualment hi han moltes aproximacions per aconseguir aquest efecte (*forever iframe, long polling...*) on la majoria es basen en fer crides asíncrones al servidor per demanar si hi ha nova informació que ens interressi. Afortunadament existeix una nova tecnologia anomenada *WebSockets* que permet treballar amb ports bidireccionals directament amb el servidor sense necessitat d'enviar tota la càrrega del protocol *HTML*. Malauradament, al ser una tecnologia relativament nova tenim els inconvenients de que no tots els navegadors suporten aquest tipus de comunicació i de que necessitem una serie de tecnologies de servidor addicionals. Respecte al problema de la compatibilitat, ja existeixen algunes llibreries que ofereixen les funcionalitats dels *WebSockets* i també incorporen altres implementacions alternatives si el navegador no els suporta (*Socket.io*<sup>8</sup>). Respecte a la part del servidor, significarà tenir disponible un servidor com *Node.js*<sup>9</sup> que permet la comunicació amb aquest tipus de ports.

Finalment haurem de decidir amb quin llenguatge ens comunicarem amb els servidors de la *UOC*. El protocol de transmissió de dades (*REST*) és natural per als entorns web, per tant serà trivial accedir a les dades tant des de servidor com des del navegador. Alhora, els 2 protocols de organització de dades que s'ofereixen també són molt comuns en entorns web, i per tant, els mateixos llenguatges de programació ja incorporen sistemes de avaluació que interpreten les dades i les organitzen en objectes.

---

8 <http://socket.io/>

9 <http://nodejs.org/>

## 4.2. Programari utilitzat

### 4.2.1. HTML5 + CSS3

*HTML* fa referència a *Hypertext Markup Language* (llenguatge de marques d'hipertext), és a dir, un llenguatge que defineix com s'estructura la informació mitjançant marques, o etiquetes.

Des de fa molts anys aquest llenguatge de marques ha sigut la cara visible d'Internet. Aquest fet és degut a que és l'encarregat de definir com es mostrarà la informació als navegadors. Des dels seus inicis ha sofert moltes modificacions fins a arribar a la definició de **HTML5**. El seu de venir està controlat per el *W3C (World Wide Web Consortium)*.

Tot i el seu nom, el *HTML5* no és una actualització del llenguatge, ni un document específic. Realment es tracta d'una idea, la idea d'una nova mentalitat respecte a com fer les coses a les pàgines web. Per tant, la compatibilitat amb *HTML5* no és una relació binària ja que un navegador pot ser compatible amb unes característiques però no a unes altres, quan totes són pròpies del mateix llenguatge.

La gran diferència que aporta als seus predecessors resideix en la forma en que es tracten les noves funcionalitats. En el *HTML* tradicional, una etiqueta tenia un ús purament estètic. En canvi en la nova especificació, les noves etiquetes venen acompanyades d'un controlador que li aporta funcionalitat. Com a conseqüència podem establir que aporta un salt en riquesa de llenguatge. Per exemple, un navegador que suporti la funcionalitat de vídeo, no sol serà capaç de reproduir un vídeo, també aportarà funcionalitats per detectar diferents formats de vídeo, oferir opcions de reproducció, transformacions de la imatge... A més a més també especifica funcionalitats que no estaven disponibles per als navegadors com pot ser l'emmagatzemament local de dades o l'ús de connexió mitjançant ports *TCP*.

Per la seva banda, el *CSS* fa referència a *Cascade Style Sheets* (fulls d'estils en cascada). Aquest tipus de fitxers venen acompanyant als documents *HTML* en forma de recurs per a separar la implementació de les regles de disseny gràfic i de format de les pàgines web. És a dir, tot lo relacionat amb com i on es mostrarà la informació continguda en un document web.

De forma gairebé contemporània a la aparició del *HTML5*, va fer-se públic el *CSS3*. De forma totalment compatible amb les seves versions anteriors, la 3a especificació de estils afegeix regles de disseny que li mancaven en el seu protocol, així com més detall en la definició d'alguns elements ja existents. Alhora també incorpora un nou sistema de transformació i animació d'elements, característica molt interessant per oferir elements que reaccionen amb la interacció de l'usuari, sense necessitat de recórrer a llenguatges de programació que modifiquin el contingut. I si això no fos prou, aquestes animacions i transformacions són calculades per la targeta gràfica de la maquina client,

optimitzant molt el rendiment en màquines amb capacitat de processament limitat, com podrien ser dispositius mòbils.

### 4.2.3. Javascript

El *Javascript* és un llenguatge de programació interpretat, és a dir, es compila en temps d'execució. Es defineix com un llenguatge orientat a objectes, prototipat, dèbilment tipat i dinàmic. En *Javascript* tots els elements són objectes que es representen en forma de vector associatiu, per tant les seves propietats es poden enumerar utilitzant bucles iteratius. Es tracten d'objectes molt dinàmics ja que poden canviar el seu valor i propietats en temps d'execució. No solament això, sinó que el tipus d'un objecte no està associat al mateix objecte, sinó al valor que te assignat l'objecte.

```
/* aquestes dos sentències són equivalents */  
object.x = 1;  
object['x'] = 1;  
  
/* Aquestes instruccions són correctes */  
var object = { x: 1 };  
object.y = 2;  
object = "canvi de tipus";
```

Des dels seus inicis és un llenguatge de programació molt lligat al món d'Internet on el seu màxim exponent es troba lligat a la part del client (navegador). No obstant, també el podem trobar en entorns mòbils, aplicacions d'escriptori o inclús programari de servidor (com el servei que mantindrà l'eina de missatgeria instantània en aquest projecte).

Per tal de maximitzar la seva potència quan s'executa en la banda del client en aplicacions web, el navegador proveeix la definició d'un objecte especial anomenat *DOM (Document Object Model)*. Aquest objecte conté tota la informació relacionada amb la estructura i jerarquia del document *HTML* que està actiu en el navegador. A més a més, també proporciona una serie de funcionalitats per a poder interactuar amb el document i generar canvis i contingut en el document el qual s'actualitzarà en la pantalla de l'usuari. Les principals aplicacions de l'ús de *Javascript* en el navegador són:

- Transformació i animació d'elements de la pàgina per a donar sensació de vida.
- Comprovació de formularis de forma prèvia al seu enviament.
- Execució de crides *REST* asíncrones al servidor. Utilitzades per recuperar contingut nou sense necessitat de refrescar la pàgina.
- Contingut interactiu per a la reproducció d'àudio o vídeo o inclús programació de videojocs.
- Interacció amb serveis externs com integració de xarxes socials, estadístiques o anuncis.

Tot i el seu potencial, les característiques i funcionalitats per defecte del llenguatge són bastant limitades. Afortunadament existeixen molts recursos d'iniciatives particulars (tant públiques com privades) que ofereixen extensions del llenguatge per completar les seves deficiències. Tot això degut a la seva facilitat d'extensió, prototipatge i orientació a objectes.

#### 4.2.3.1. jQuery

*jQuery* és una llibreria que estén la funcionalitat del *Javascript* facilitant la manipulació del DOM i l'accés a moltes més funcionalitats de les que disposa el llenguatge de forma inicial. El seu nom es deu a la forma en que accedeix a la jerarquia del document.

La seva gran acceptació entre els desenvolupadors web es deu a la seva lleugeresa (34Kb en la seva forma minimitzada i compresa), i a la seva relació de dificultat d'aprenentatge i possibilitats, ja que és molt fàcil entendre el seu funcionament i ofereix possibilitats casi inimaginables utilitzant el *Javascript* per si sol.

El seu punt fort resideix en la gran versatilitat que ofereixen els seus selectors, que en forma de cadena de caràcters permet fer seleccions dintre del *DOM* en poques instruccions, de forma que estalvien al desenvolupador de fer varies línies de codi molt menys fàcils de mantenir. A més a més de incloure funcions de manipulació de documents noves com animacions.

Ahora també simplifica la forma en que es gestionen tasques complicades com les crides asíncrones, reduint la seva complexitat, gestionant de forma eficient els esdeveniments i reduint la extensió de caràcters utilitzats, reduint així la quantitat de dades que s'han de descarregar del servidor.

```
/* aquestes dos sentències són equivalents */
$('#id_element').height(300);
document.getElementById('id_element').height = 300;

/* selecciona la primera fila de la taula amb classe 'taula-
centrada dintre de l'element amb id 'divisio_1 */
var first = $('#divisio_1 table.taula-centrada tr:first');
```

En aquest exemple podem veure com simplifica la manipulació del document, i més important encara, com accedeix a un element molt endinsat en la jerarquia del document en una sola instrucció. Si s'hagués d'accedir a aquest element sense *jQuery* es necessitarien varies sentències i condicionals per tal de que el conjunt d'operacions fos robusta i el seu resultat satisfactori.

#### 4.2.3.2. Deer Renderer

*Deer Renderer* és una llibreria auxiliar que permet la manipulació del contingut del DOM de forma eficient i reutilitzable. És una creació personal, creada específicament per les necessitats d'aquest projecte, però que disposa d'una funcionalitat suficientment específica i necessària que mereix estar tractada com un element per si sol.

Aquest mòdul s'encarrega de representar en estructura del *DOM* objectes de *Javascript* que es carreguen de forma dinàmica i que es representen de forma repetida degut a la càrrega dinàmica de la informació mitjançant crides asíncrones.

Tant *Javascript* com *jQuery* no ofereixen ninguna funcionalitat per a incloure blocs d'elements de *HTML* de forma dinàmica. L'única forma de realitzar aquesta tasca es resumeix a crear els elements mitjançant una cadena de caràcters concatenant la informació dinàmica. Aquesta pràctica és molt tosca, poc entenedora i poc reutilitzable.

*Deer Renderer* s'encarrega de descarregar un document de forma asíncrona. En aquest document hi han les definicions dels blocs que volem utilitzar. Mitjançant *jQuery* carrega la estructura per a poder cridar els blocs de forma individual. Una vegada te la representació individual substitueix les dades de l'objecte que se li ha passat com a font d'informació en els espais que se li ha indicat a la definició dels blocs. Finalment torna una reproducció del bloc desitjat amb dades dinàmiques que podem afegir al document de forma dinàmica, utilitzant unes poques línies de codi.

```
/* un bloc vindria representat d'aquesta forma*/
<div id="bloc">
  <div id="bloc_{{id}}"> {{text}} </div>
</div>

/* la sentència per incloure el bloc al document */
var data = { id: 123, text: 'nou contingut'};
renderer.init( function(){
  $(body).html( renderer.load('bloc', data) );
});

/* resultat del dom */
<body>
  <div id="bloc_123"> nou contingut </div>
</body>
```

Com es pot veure en aquest exemple, quan tenim condicions molt més complexes, només amb una línia de codi es pot utilitzar un bloc prefabricat, donar-li un valor nou i afegir-lo al document. És especialment útil en casos on la estructura dels elements

repetits és molt similar i la estructura és massa complexa per a escriure directament al codi tots els elements.

Aquest mòdul s'ha utilitzat molt en el projecte per tal de incloure contingut asíncron com els missatge d'un fòrum o la llista d'usuaris d'una aula.

#### 4.2.3.3. JSO (Javascript OAuth library)

*JSO* és una llibreria desenvolupada per Andreas Åkre Solberg que permet la autenticació de tercers mitjançant *OAuth 2.0* sense necessitat de passar per un servidor, és a dir, gestionant l'autenticació totalment des de la banda del client.

Per explicar millor el funcionament, és necessari explicar que és i com funciona l'*OAuth 2.0*. És tracta d'un entorn de treball de codi obert que permet l'autorització de tercers a un servei web de forma limitada. El seu funcionament és molt simple, quan un client vol accedir a les dades de un tercer, fa la petició de accés i el client és redirigit a la pàgina del tercer on gestiona els passos per a l'autorització. Quan l'autorització acaba, l'usuari és redirigit a la adreça indicada al començar la petició amb el resultat de l'autorització. Si el resultat és satisfactori, el servidor pot desar les credencials i demanar les dades de l'usuari per a que el client mostri les dades del tercer.

En el nostre cas, el client és **Fòrums UOC** i el servei de tercers és el *Open-API* de la *UOC*. Malauradament, la aplicació intenta evitar al màxim les interaccions al servidor, fent que la major part de la càrrega del servidor sigui suportada per el client. En aquest punt és on entra en joc la llibreria *JSO*, que gestiona la resposta dels servidors d'autorització i guarda les credencials de forma local.

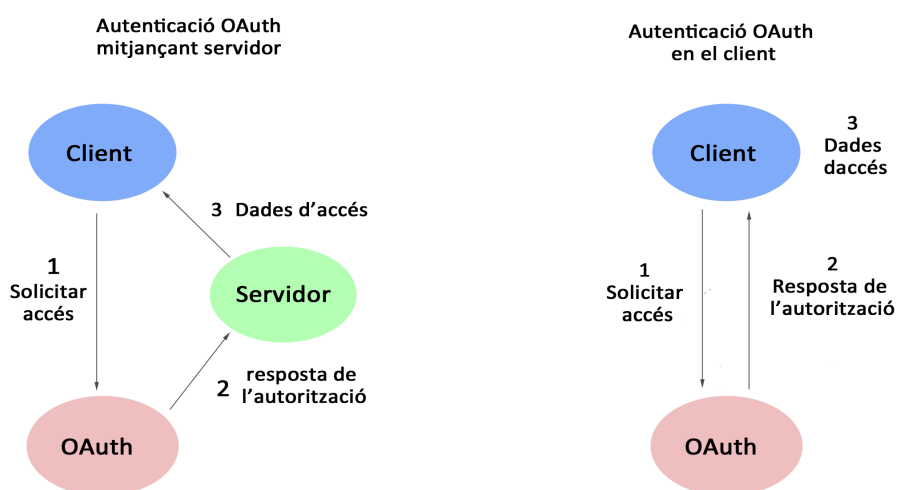


Figura 14. Esquema d'autenticació del protocol OAuth 2.0

Utilitzant aquesta llibreria aconseguim extreure molta càrrega de treball del servidor i aconseguim la nostra fita de mantenir la càrrega de treball al client, utilitzant una pàgina que no canvia, augmentant la sensació de context en tot moment.

#### 4.2.3.4. Localstorage i JSON

L'entorn del llenguatge *Javascript* canvia en cada navegador, inclús entre les versions d'un mateix navegador. Amb l'aparició del *HTML5*, les últimes versions dels navegadors més populars han començat a incorporar noves APIs molt interessants. Dues d'elles són el suport de tractament del llenguatge de serialització de dades *JSON* i el gestor de persistència local *Localstorage*.

*JSON (Javascript Object Notation)* és un conjunt de regles que regeix com s'estructuren les dades en *Javascript*. Alhora és un dels mètodes de serialització de dades més populars per la transferència d'informació a través de la xarxa. Això es deu a que té un molt bon equilibri entre la lleugeresa que aporta i la facilitat de la lectura per als humans.

```
/* exemple d'objecte JSON */
{
  id: 123,
  vector: [ 'array', 'dintre', 'un', 'objecte' ],
  objecte: {
    mes: 'objectes'
  }
}
```

En l'exemple anterior podem veure un objecte que conté un atribut numèric, un vector i un altre objecte. La estructura que ofereix simplifica molt la lectura, molt útil a l'hora de revisar la informació i de entendre possibles mals funcionaments. No obstant, no sacrifica la eficiència, ja que els delimitadors de tipus són mínims i no afegeixen una gran quantitat d'informació extra que no tingui res a veure amb la informació a transmetre.

Per la seva banda, el *Localstorage* és una interfície que afegeix capacitats d'emmagatzemament local en els navegadors. És una funcionalitat molt nova en aquest sector de la xarxa i molts navegadors encara no ho suporten. És basa en un sistema molt bàsic de clau-valor i les dades es guarden en forma de cadena de caràcters. Té una funcionalitat molt limitada (guardar, consultar i esborrar).

La forma en que el *Localstorage* guarda les dades és el que fa que s'entengui de forma molt relacionada amb el *JSON*, ja que aquest és capaç de serialitzar els objectes en forma de cadenes de caràcters, les quals es poden guardar en la memòria del navegador de forma trivial.



Tot i la desconfiança que poden sembrar a priori aquests dos grans avenços, degut a la baixa implantació que tenen als navegadors actuals, existeixen llibreries que emulen el seu funcionament i serveixen com a xarxa de seguretat en navegadors que no disposen de les eines suficients. Degut a la necessitat d'alguns elements de **Fòrums UOC** per utilitzar aquestes eines, l'aplicació disposa de les implementacions alternatives per assegurar el correcte funcionament de tot el conjunt.

#### 4.2.3.5. Node.js

*Node.js* és un entorn de *Javascript* que s'executa en la banda del servidor. Utilitza un model d'execució asíncron orientat a esdeveniments. S'executa sobre V8, el compilador de *Javascript* desenvolupat i mantingut per *Google*.

Aquest entorn de programació està destinat a mantenir aplicacions i serveis web de forma lleugera, escalable i eficient. Un dels avantatges d'aquest tipus de servidor és que s'executa sobre un sol fil, cosa que fa que pugui mantenir un gran nombre de connexions simultànies sense que el temps de resposta es vegi afectat. Aquest fet és especialment important sobretot quan tenim una aplicació que ha de suportar fins a milers de peticions simultànies. No obstant, aquest protocol d'execució també es pot tornar un inconvenient, ja que al ser un sol fil d'execució, només pot executar-se en un nucli de la màquina. Per tant una important tasca de balanceig entre diferents execucions és necessària per tal de poder servir aplicacions i serveis d'alta demanda.

Alhora, la seva naturalesa fa que sigui una eina idònia per a allotjar aplicacions que necessitin d'una connexió permanent amb el servidor i així poder transmetre informació en temps real. Aquest factor fa que el seu ús sigui gairebé indispensable en un projecte com el que ens ocupa. Tenim un servidor *Apache* que serveix els documents i recursos, i una instància de *Node.js* que s'ocupa de mantenir actualitzada la eina de missatgeria instantània.

#### 4.2.3.6. Socket.io

Es tracta d'un mòdul que complementa les funcions de l'entorn de servidor *Node.js*. Els mòduls d'aquesta tecnologia de servidor serveixen per estendre la funcionalitat que aquesta no porta per defecte. Qualsevol individual o col·lectiu pot aportar els seus mòduls per a que siguin descarregats per tota la comunitat.

Concretament, *Socket.io* s'encarrega de obrir el ventall de public objectiu de la nostra aplicació amb *Node.js*. Un dels problemes del *Node.js* recau en que només suporta la connexió amb els clients mitjançant *WebSockets*. Igualment que passava amb el *LocalStorage* és una interfície present encara a molts pocs navegadors, per tant limita molt la decisió de adoptar aquesta solució. Afortunadament, aquest mòdul ofereix varies alternatives als *sockets* de navegador per a que qualsevol ordinador personal, inclús dispositius mòbils puguin disposar dels beneficis del *Node.js*.

Entre les solucions alternatives que ofereix, hi podem trobar: *WebSocket*, *Adobe Flash Socket*, *AJAX long polling*, *AJAX multipart streaming*, *Forever Iframe*, *JSONP Polling*. Tots aquests diferents mètodes de comunicació són respectivament menys eficients. El mòdul, de forma transparent a l'usuari escollirà la millor opció possible per comunicar-se amb el servidor mantenint el mateix codi d'aplicació en qualsevol dels casos. Aquesta característica fa del *Socket.io* un mòdul d'ús imprescindible quan volem desenvolupar una aplicació client-servidor de temps real amb *Node.js*.

#### 4.2.4. PHP

Fa referència a *HyperText Preprocessor*. Es tracta d'un llenguatge de programació que s'executa en el servidor. És una eina molt popular per el desenvolupament d'aplicacions web gràcies a la suavitat de la seva corba d'aprenentatge i a la gran potència que pot oferir per als programadors. Molt llocs web molt populars fan servir aquesta tecnologia, com pot ser *Facebook*.

Una de les bones qualitats de *PHP* és que s'executa en el servidor i pot portar codi *HTML* incrustades. El resultat de la seva execució és interpretada per els navegadors com *HTML*, per tant, el client mai sabrà el contingut de la programació, només tindrà noció del resultat obtingut.

Tot i voler minimitzar el ús de *PHP* en aquest projecte, s'ha volgut aprofitar les seves capacitats d'executar-se en el servidor. Aquest fet fa que pugui respondre a peticions asíncrones del client i comunicar-se amb el sistema de persistència del servidor per tal de poder emmagatzemar les dades necessàries dels usuaris que estenen les proporcionades per la *Open-API* de la universitat.

Concretament en aquest projecte s'encarrega d'atendre a les peticions *AJAX* enviades per el client, utilitzades per la gestió de les sessions de la eina de missatgeria instantània i l'emmagatzemament del sistema d'amistats, també utilitzades per el xat.

#### 4.2.5. MySQL

És un dels sistemes de gestió de bases de dades més utilitzats que es poden trobar. Està especialment utilitzat en entorns webs degut a la seva llicència de codi obert, tot i que és de propietat privada.

La seva popularització es deu a la simplicitat de la seva gestió, la facilitat de configuració (sobretot en entorns *Linux*) i el gran suport de la seva comunitat. Trets que l'han fet destacar tot i les seves mancances inicials. En les seves primeres etapes no assegurava la integritat referencial ni la atomicitat de les transferències.

Afortunadament, amb el pas del temps ha esdevingut un sistema molt complet, afegint noves característiques i funcionalitats.

Es tracta d'un sistema que utilitza el llenguatge *SQL* ampliat per interpretar les consultes de la base de dades. Respon a un model de dades relacional, per tant les seves dades estan organitzades en taules formades per atributs, i aquestes contenen registres. És un sistema multi-fil i multi usuari, fet que aporta molta potència en quant a la gestió de sessions i permisos.

El fet de que sigui una base de dades relacional ens aporta les característiques de coherència de dades i alta disponibilitat. Malauradament no és un sistema fàcil d'escalar degut a la seva estructura. Una altra característica molt important es que permet fer una selecció del motor d'emmagatzemament com per exemple *CSV*, *MyISAM* o *innnoDB*; permetent així seleccionar el que millor s'escaigui depenent del nostre tipus de dades.

En aquest projecte s'utilitza com a pont entre la gestió de recursos de comunicació de la *UOC* i la eina de missatgeria instantània. Per tant tant el codi PHP *com el servidor Node.js accedeixen a les seves dades*. Aquesta circumstància fa que la gestió de la informació sigui efectiva i fiable ja que un entorn tancat de servidor controla sempre el seu estat i coherència de dades.

### 4.3. Capes de l'aplicació

Fòrums UOC és una aplicació web estructurada en capes per a aïllar les responsabilitats de cada nivell de tasques. Es tracta d'una execució amb molta càrrega de treball en la banda del client, el qual du a terme les seves operacions amb el suport d'un entorn de servidor que és el que controla la persistència de dades.

El client te la responsabilitat de controlar la presentació de l'aplicació mentre que els servidors s'encarreguen de la persistència. No obstant, els dos participen de les regles de negoci de l'aplicació mantenint-se en comunicació mitjançant protocols d'intercanvi en xarxa.

En el següent gràfic es pot comprovar la distribució de responsabilitats de cada mòdul que participa en la execució de l'aplicació.

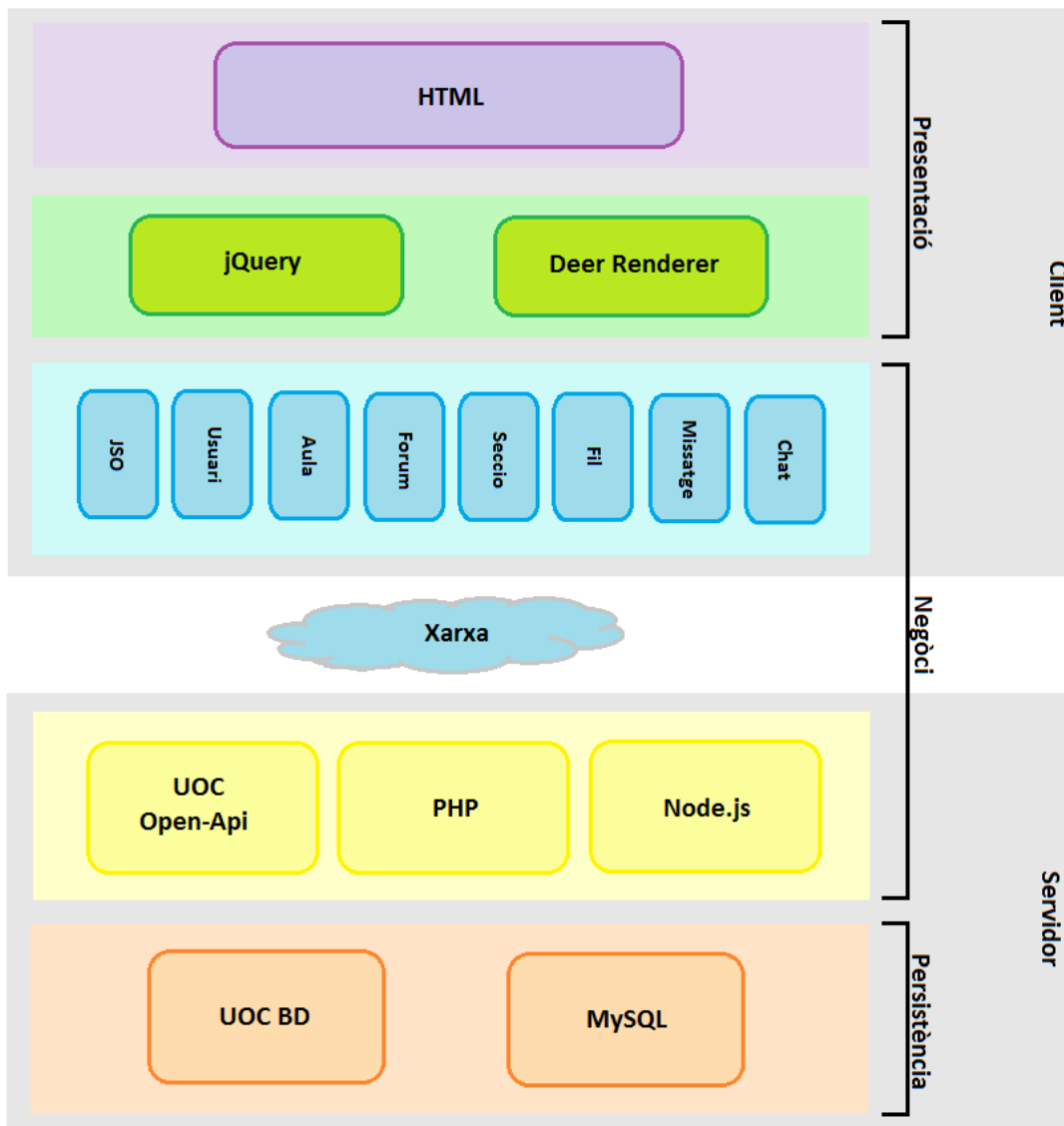


Figura 15. Esquema de capes i components de l'aplicació

### 4.3.1. Client

A la banda del client és on es centra la major part de l'activitat de l'aplicació. És qui s'encarrega de gestionar els fluxos d'informació i de reaccionar als estímuls de l'usuari.

Com s'ha pogut veure en apartats anteriors, l'aplicació es bàsicament un lloc web sense refresc. Aquest fet és possible gràcies als llenguatges de *scripting* disponibles a la majoria de navegadors moderns. Per tant, sense moure'ns de la mateixa plana som capaços de mostrar informació canviant de forma automàtica o responent als estímuls de l'usuari.

Per a gestionar aquest gran flux d'esdeveniments s'ha optat per l'ús de controladors per cada sector d'informació. Aquesta decisió ve donada degut a la independència entre els diferents grups d'informació que s'havien de tractar. La informació recollida de la *UOC* ve clarament diferenciada entre aules, fòrums, missatges i usuaris; on cadascuna d'aquestes entitats modulars necessiten tenir poder de control tant a nivell estructural com de presentació. A més a més, el disseny de la navegació per l'aplicació afavoreix la modularització d'aquestes estructures degut a la seva independència en la forma en que es presenten a l'usuari.

Cada controlador te la responsabilitat d'inicialitzar el seu estat, carregar les seves dades i encarregar-se de la mostra de la seva informació.

El ús de controladors ens permet encapsular la funcionalitat de cada unitat d'informació restringint les responsabilitats de cada unitat de programari i mantenint la cohesió i la coherència en tota l'arquitectura. Per tant ens simplificarà les tasques de manteniment i detecció d'errors ja que sabrem fàcilment d'on prové la informació, alhora permet un desenvolupament més clar i robust ja que hi han menys dependències entre classes.

Aquesta estructura brinda la oportunitat de navegar entre seccions de forma fàcil i aïllada, permetent el pas d'informació de forma ordenada que es percep de forma natural en l'usuari.

L'usuari accedeix a l'aplicació després d'autoritzar el seu ús amb els servidors de la *UOC*. De forma immediata, l'aplicació recupera la informació personal i de les seves assignatures. La informació personal serveix per personalitzar l'aplicació amb les seves dades, de forma que es sent més integrat i la percep de forma més amistosa. Les assignatures presenten el primer bloc d'informació permetent a l'usuari veure informació des del primer moment. A partir d'aquest punt la navegació és transparent per a l'usuari mitjançant la mimetització dels elements informatius amb els elements interactius, una tècnica molt utilitzada en les noves interfícies gràfiques que s'apropen a la comunicació natural.

També, a la banda del client, hi trobem l'eina de missatgeria instantània (xat), que permet establir converses en temps real amb altres usuaris. De la mateixa manera que el campus virtual mantenia una estructuració de la informació mitjançant controladors, la eina de xat disposa d'un controlador específic per a gestionar el seu funcionament. De fet és un dels controladors més aïllats amb una forta càrrega de gestió d'interfície d'usuari i de comunicació amb el servidor.

El controlador de missatgeria instantània s'encarrega de mostrar la barra de xat, un element visual i estructuralment aïllat del flux de navegació establert per la resta de l'aplicació. Aquesta barra és la encarregada d'aïllar conceptualment la eina, permetent distingir amb facilitat de les funcions habituals dels fòrums. En la barra s'organitzarà tota la informació relativa al xat: menús de configuració, llistat de converses i llistat de persones connectades. És una eina sempre visible de forma fixa a la part inferior de l'aplicació per minimitzar la molèstia que podria ocasionar a l'usuari i afavorir la interacció en tot moment.

### 4.3.2. Servidor

Tot i ser una aplicació on el moviment de la informació es situa a la banda del client, **Fòrums UOC** recupera la informació al núvol per a que sigui mostrada, independentment del dispositiu d'accés, a l'usuari.

Concretament es nodreix de tres fonts d'informació bàsiques per a aconseguir ser una eina de comunicació transversal.

El leitmotiv de l'aplicació, i per tant, el principal proveïdor d'informació, és el *Open-API* que ofereix la *UOC*. Es tracta d'un servei per a tercers que permet accedir a informació privada dels usuaris i serveis de la universitat mitjançant un protocol d'intercanvi d'informació assegurat per un protocol d'accés. Dit d'una altra manera, ens permet obtenir les dades personals d'un estudiant de la *UOC* per a que sigui representada en la nostra aplicació.

A més a més de la personalització dels recursos d'informació de la universitat, l'aplicació també ofereix la possibilitat de interactuar de forma directa i instantània amb altres usuaris. No obstant, aquesta comunicació s'ha de gestionar mitjançant l'ús d'un punt de unió, que anomenarem servidor de missatgeria instantània.

Finalment, per optimitzar el funcionament i la comunió de les 2 eines comunicatives, s'ha establert un servidor d'aplicació que ajudarà a mantenir la informació de la que no disposa la universitat i farà de nexa d'unió entre les eines, ajudant a mantenir un nivell òptim de seguretat i privacitat entre els dos serveis.

### 4.3.2.1. Open-API

**Fòrums UOC** ha nascut amb la intenció de modernitzar i facilitar l'ús dels recursos de comunicació que ofereix la *UOC*. D'aquest raonament es pot extreure la necessitat de interactuar amb la informació de la universitat per tal de maximitzar el profit de l'aplicació. En aquest punt és on recau la importància de l'*Open-API*.

Com s'ha explicat anteriorment, aquest servei de la universitat permet comunicar-se amb tot el contingut d'un usuari mitjançant crides de xarxa. No obstant, abans s'ha de mantenir un protocol de autenticació per mantenir la privacitat de l'usuari. Gràcies a aquest protocol, una aplicació pot sol·licitar l'accés a uns determinats permisos per a efectuar tasques d'entrada i sortida amb el sistema de persistència de la universitat.

En aquest cas, l'aplicació es centra amb la interacció amb els recursos de comunicació i les dades referents als usuaris que els envolten. La informació sol·licitada comença amb les dades de l'usuari que ha autoritzat l'aplicació per a poder oferir un servei més personalitzat i, alhora, poder conèixer les assignatures en les que participa. En aquest punt, els diferents mòduls de l'aplicació s'encarreguen de demanar la informació relacionada amb els fòrums, els missatges que contenen i els estudiants que hi participen. A més a més, l'aplicació ha d'interactuar amb la universitat amb l'enviament i resposta de missatges, per tant també executa crides per a afegir informació, no solament extreure'n.

Degut a que és un servei extern, localitzat en un context molt diferent al de l'aplicació, s'han efectuat algunes estratègies per a optimitzar el ús d'aquest servei. L'inici de sessió i la clau d'accés proporcionada una vegada s'ha autenticat l'usuari, es guarden de forma local i es comproven en cada execució per no explotar el servei i per facilitar la rutina d'accés a l'usuari. També s'ha configurat un registre d'usuaris que han aparegut al sistema, desant les seves dades principals, de forma de que només s'hagi de demanar informació de l'usuari un cop per execució. En una mateixa execució poden haver desenes de representacions d'un usuari i en molts dels casos la informació s'ha de recuperar de forma independent a la font de la referència. Per tant aquesta estratègia minimitza l'estres a l'*Open-API*. A més a més, l'eina de comunicació instantània també utilitza informació de tots els usuaris de l'aula.

### 4.3.2.2. Servidor de missatgeria instantània

Comunicació instantània entre usuaris implica transferència de dades en temps real entre dos clients concrets. Comunament en la informàtica s'ha utilitzat la transferència directa de paquets per a realitzar aquesta tasca. Malauradament, aquesta tecnologia no és una opció quan parlem d'aplicacions web.

La forma més òptima d'abordar el problema consisteix en situar un servidor mitjancer que gestioni la transferència d'informació. Actualment, una de les millors

aproximacions és configurar un servidor utilitzant *Node.js*, un servidor de *Javascript* que permet establir de forma molt eficient connexions persistents.

El que fa propici al *Node.js* per a ser utilitzat en aquest context és que esta orientat a esdeveniments, fet que simplifica la estructuració del servidor.

El primer pas que efectua el servidor és iniciar la sessió de l'usuari. És un pas molt delicat ja que no volem que cap usuari no legitimat pugui accedir al servei, i menys encara entrar suplantant la identitat d'un altre estudiant. Per tal d'assegurar que els clients que accedissin al xat fossin usuaris amb una sessió activa a la *UOC* s'ha creat un sistema de sessions lligat al paràmetre d'accés proporcionat per la *UOC*. En el moment que s'inicia l'eina de missatgeria instantània, el client envia al servidor d'aplicació el identificador d'usuari i el paràmetre de sessió de la universitat. El servidor primer utilitza el paràmetre de sessió per a demanar les dades d'usuari a l'*Open-API* i comprova que el paràmetre és vàlid i que el identificador de l'usuari correspon amb la informació rebuda de la *UOC*. Si tot és correcte, el servidor d'aplicació genera una clau única, la guarda a la base de dades i la retorna al client. Una vegada el client te aquesta clau pròpia, demana l'inici de sessió al servidor de xat, i li envia el identificador d'usuari i la clau obtinguda prèviament. El servidor de xat consulta a la base de dades si la clau és vàlida i correspon al nostre usuari, si tot és correcte inicia la sessió i notifica al client del resultat. Amb aquest procediment, ens assegurem que el client sempre demana primer una clau abans d'iniciar la sessió i per tant, que té una sessió activa vàlida amb els servidors de la universitat.

Una vegada la el client està connectat al servidor de missatgeria instantània, ambdós es queden esperant esdeveniments. Qualsevol dels dos pot iniciar un esdeveniment al qual l'altre participant ha de respondre amb una acció. Els esdeveniments normalment els inicia el client després d'una interacció de l'usuari, com la escriptura en una conversa, l'enviament d'un missatge o la petició de desconnexió. El servidor haurà d'identificar el motiu de l'esdeveniment i processar la informació, normalment per notificar a un altre client.

Des del servidor de xat es gestiona una llista de clients per a poder redirigir les accions d'un usuari a un altre. El client només ha d'enviar el identificador d'un receptor d'informació i el servidor s'encarregarà de cercar a la llista de clients i remetre la informació al destinatari.

#### **4.3.2.3. Servidor d'aplicació**

Tant la eina de comunicació instantània, com la gestió dels recursos de comunicació de la universitat esdevindrien unitats aïllades sense interacció sense l'ús intermedi d'un servidor d'aplicació que permetés la comunicació entre aquests dos serveis.



Concretament aquest servidor s'utilitza per a estendre les característiques i la informació que aporta la *UOC* per tal de fusionar-se amb la eina de xat oferint així una funcionalitat única i homogènia.

Aquest servidor es comunica amb el client de la mateixa manera que ho fa el *Open-API*, mitjançant crides de xarxa. No obstant, en aquest cas el servidor es troba en un entorn *Linux* on la tecnologia que hi dona suport és una combinació de *PHP* i *MySQL*. Les tasques que te assignades són molt clares. Oferir un sistema persistent d'amistat entre usuaris i servir de pont per l'autenticació de l'usuari en la eina de missatgeria instantània a partir del paràmetre d'accés proporcionat per la universitat.

Un xat comporta intercanvi d'opinions entre usuaris de forma ràpida i informal, per tant costaria molt d'entendre sense un sistema d'amistats que permetés accedir fàcilment als contactes que s'utilitzen més habitualment. A més a més, la llista d'usuaris que proporciona la *UOC* només conté els estudiants que estan cursant les mateixes assignatures que l'usuari actiu. Això comporta que cada semestre podrien canviar els usuaris, podent perdre contacte amb companys d'altres cursos. Per a resoldre aquest problema, s'ha creat una petita base de dades que conte la informació de parelles d'usuaris. Aquesta informació la demana el client cada cop que inicia sessió a l'eina de missatgeria instantània de mode que pot tenir accés als usuaris més afins. El servidor d'aplicació també proporciona punts d'accés per a gestionar la llista d'amistats, podent incloure estudiants des de la interfície del client.

L'altra funcionalitat que afegeix aquest servidor consta en servir de pas intermedi per assegurar la legitimitat dels usuaris que es connecten al xat. No es podria enviar directament la informació d'accés de la universitat, ja que un usuari maliciós podria falsejar la informació i el servidor de missatgeria instantània no podria comprovar la seva veracitat. De mode que el client es comunica primer amb el servidor d'aplicació per a que crei un codi únic i de curta durada, després de comprovar (des del servidor) si la informació és vàlida. I després el servidor de xat pot comprovar la informació ja que es troba a la mateixa màquina i te accés a la base de dades per a comprovar la autenticitat del codi recent generat.

## 5. Configuració del servidor i desplegament

Per a poder tenir disponible l'aplicació per a tots els usuaris, es necessitarà disposar d'un servidor que estigui connectat a la xarxa, accessible des d'una connexió ordinària a Internet. A més a més, és necessari poder accedir a la configuració del servidor per tal de configurar els seus serveis.

A continuació es detallarà com realitzar la configuració i desplegament de l'aplicació en un entorn *Linux*. No obstant, s'ha de saber que les diferents distribucions de *Linux* poden variar la forma en que s'instal·len els seus serveis, localització dels fitxers de configuració i comandes de la línia de comandament. La descripció s'ha fet basant-se en una distribució *Red Hat* modificada per *Amazon* que es pot trobar en les màquines del *Amazon Web Services EC2*, la instal·lació s'ha dut a terme des d'un PC amb *Microsoft Windows 7*.

### 5.1. Connexió al servidor

Per iniciar la instal·lació, haurem d'accedir mitjançant un programa de línia de comandes remot. Des del sistema operatiu Windows, una eina que realitza molt bé la feina és el programa anomenat *PuTTY*<sup>10</sup>. Des d'aquest programa podrem iniciar una connexió segura amb la nostra màquina.

El primer pas consisteix a esbrinar les dades de connexió de la màquina, aquestes són: IP de la màquina, port de la connexió segura (SFTP), nom d'usuari i les credencials de connexió. Les credencials poden ser un mot de pas o una clau privada -com és el cas de les màquines *EC2* de *Amazon*-. Totes aquestes dades ens les ha de facilitar el proveïdor de l'allotjament o administrador del servidor.

En cas de que el nostre mètode d'autenticació sigui mitjançant una clau privada, haurem de realitzar un pas intermedi per tal de poder-la usar amb el *PuTTY*. Aquest programa utilitza un xifrat de clau especial anomenat PPK i normalment el proveïdor ens donarà una clau de tipus PEM. Afortunadament el propi *PuTTY* ve acompanyat amb un generador de claus que resoldrà el nostre problema. Per això haurem de obtenir el programa *PuTTYgen*.

Per a transformar la clau, només és necessari obrir el programa, seleccionar "*Conversions -> Import Key*" i anar a la ubicació de la nostra clau. El programa importarà la clau i només farà falta seleccionar la opció de "*Generate Key*" per tal de generar l'arxiu PPK que ens servirà per iniciar sessió al *PuTTY*.

---

10 <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

Es molt important transportar la clau comprimida si s'envia per correu o mitjançant algun servei de disc dur al núvol ja que alguns clients de correu i de descarrega poden corrompre el la clau i que no es pugui transformar correctament.

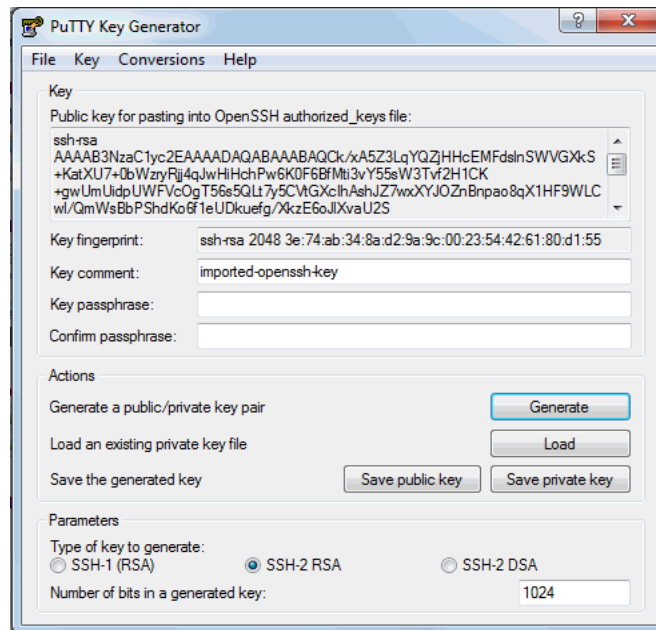


Figura 16. Generació de clau privada

En aquest moment ja es pot omplir les dades d'accés necessàries per iniciar la sessió al servidor. En la següent imatge es pot veure el menú de connexió del programa, on s'han ressaltat els camps més importants per tal d'iniciar la sessió correctament.

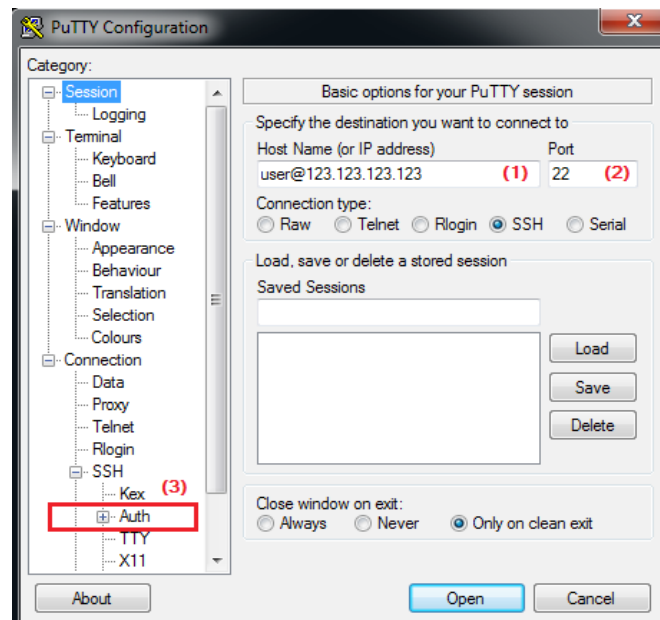


Figura 17. Connexió al servidor utilitzant SSH

## 5.2. Instal·lació dels serveis necessaris

L'aplicació **Fòrums UOC** necessita que el servidor tingui instal·lats: *PHP*, *MySQL* i *Node.js*. Alhora ha de disposar d'un servidor que pugui atendre a les peticions dels clients, en aquest document s'explicarà com configurar correctament un servidor *Apache (httpd)*.

Moltes versions de *Linux* optimitzades per a màquines que s'utilitzen com a servidors en xarxa, tenen molt poc programari instal·lat per minimitzar el ús de processador innecessari. Per això el primer pas serà comprovar que el servidor disposa dels serveis necessaris mitjançant les següents operacions de línia de comanda:

```
# httpd -v
# php -v
# node -v
# mysql
```

Aquestes crides han de retornar informació de les respectives aplicacions, en cas de que alguna donés error, s'ha de seguir les instruccions per instal·lar els serveis apropiats. En el cas del *MySQL* s'obrirà el monitor de *SQL*, per sortir només cal escriure *"exit"*.

La instal·lació dels serveis és molt senzilla en la majoria de distribucions de *Linux*, exceptuant la del *Node.js*. En la implantació del servidor de proves només va ser necessari executar el següent conjunt d'instruccions.

```
# sudo yum install httpd
# sudo yum install mysql mysql-server
# sudo yum install php php-pdo php-mysql php-mbstring php-
mcrypt php-gd php-xml
```

En el cas de que s'hagi descarregat el *MySQL*, és molt recomanable establir un mot de pas per al super usuari.

```
# sudo mysqladmin -u root password 'password-de-root'
```

No obstant, el *Node.js* -necessari per a servir l'eina de missatgeria instantània- no disposa d'una instal·lació tant senzilla per el moment i s'han d'executar més passos.

### Instal·lació de Node.js

Aquest servei no ofereix un instal·lador directe, per tant s'haurà de descarregar el codi font i compilar-lo per a tenir-lo disponible a la nostra màquina. Per tant, primer ens

assegurarem de que disposem de les eines necessàries per a resoldre els següents passos.

```
# sudo yum install gcc-c++
# sudo yum install git-core
# sudo yum install curl
```

Seguidament ja es pot descarregar el codi font del repositori públic. En l'exemple, es mostra com es descarrega tot en una carpeta de l'usuari actual.

```
# cd /home/ec2-user
# git clone git://github.com/joyent/node.git
```

Després de la descàrrega ja toca el torn de compilar el programa per a poder-lo instal·lar al servidor.

```
# cd node
# ./configure
# make
```

Aquest pas és molt lent, i pot durar varis minuts en acabar la seva tasca. Quan finalitzi, només caldrà executar l'instal·lador per a finalitzar. Per a comprovar que tot ha sorgit correctament, farem que el sistema ens digui quin número de versió de Node.js tenim actualment disponible al sistema.

```
# sudo make install
# node -v
```

Existeix una eina anomenada *Node Package Manager* (npm), que s'encarrega de gestionar la descàrrega i instal·lació de mòduls que el Node.js pot utilitzar, per tal de facilitar la feina d'implantació. Malauradament, aquesta eina en si mateixa tampoc és de trivial instal·lació. Per tal de fer-ho, s'han de tenir privilegis de super-usuari, cosa que fa que no es pugui accedir directament al recent instal·lat Node.js. Per tant, primer s'ha de configurar la variable de sistema PATH i després executar un fitxer de comandes que es troba a la xarxa.

```
# sudo su
# PATH=$PATH:/home/ec2-user/node
# export PATH
# curl http://npmjs.org/install.sh | sh
# exit
```

En aquest punt ja podem descarregar els mòduls de Node.js necessaris per tal de poder servir la aplicació de **Fòrums UOC**.

```
# sudo -E npm install -g socket.io mysql
```

### 5.3. Configuració del servidor

Un cop estan tots els serveis preparats per funcionar, només és necessari que funcionin. Per tant sempre es convenient saber quines són les instruccions que faran canviar l'estat d'execució dels nostres serveis.

```
# sudo service httpd start
# sudo service httpd restart
# sudo service httpd stop

# sudo service mysqld start
# sudo service mysqld restart
# sudo service mysqld stop
```

També hi ha la opció de fer que els serveis s'iniciïn de forma automàtica al iniciar el sistema. D'aquesta manera no cal recordar cada cop d'iniciar-los.

```
# sudo chkconfig httpd on
# sudo chkconfig mysqld on
```

El Node.js també és un servei que s'ha d'iniciar. No obstant, la seva execució és diferent ja que depèn de l'aplicació que estiguem executant en cada moment, i s'explicarà en un apartat posterior.

Per acabar la configuració, només cal repassar un punt important de la execució del Node.js. Tot i que es pot concebre com un servei web tradicional, en el nostre cas només s'encarrega de gestionar el servei de missatgeria instantània. Per tant, no volem que s'hi accedeixi com una pàgina web, sinó com un servei de transmissió de dades mitjançant ports del servidor.

Es pot realitzar aquesta configuració de dues formes. Obrint els ports del firewall del servidor, que és una operació delicada si no es sap el que s'està fent ja que pot exposar el servidor a fallades de seguretat. O utilitzant una redirecció interna del servidor connectant una ruta específica a un port del servidor.

Per tal de realitzar la redirecció interna, s'ha d'editar el fitxer de configuració del servei de servidor.

```
# vim /etc/httpd/conf/httpd.conf
```

I s'ha d'introduir les següents condicions al final del fitxer. Cal tenir en compte que una vegada l'arxiu ja s'hagi desat correctament, s'ha de reiniciar el servidor mitjançant la instrucció pertinent.

```
<VirtualHost *:80>
    ProxyPreserveHost Off
    ProxyRequests Off
    ProxyPass /socket_redir/ http://localhost:3123/
    ProxyPassReverse /socket_redir/ http://localhost:3123/
    ProxyPass /socket.io http://localhost:3123/socket.io
    ProxyPassReverse /socket.io http://localhost:3123/socket.io
</VirtualHost>
```

El que s'està fent en aquesta configuració és informar al servidor que quan es demani qualsevol petició al directori *socket.io* o *socket\_redir* que ho envii directament al port que farà servir l'eina de missatgeria instantània.

## 5.4. Instal·lar l'aplicació

Per a disposar del codi de l'aplicació només és necessari clonar el contingut del repositori públic de l'aplicació *UOC* que està disponible a la xarxa.

```
# cd /var/www/html/
# git clone https://drlecks@bitbucket.org/drlecks/forums-uoc.git
```

També és possible accedir mitjançant un programa de transferència *SFTP* com el *FileFox*. Però és necessari assegurar-se d'iniciar la sessió de forma segura i utilitzant la clau privada que s'ha generat anteriorment. En aquest cas, també és important que l'usuari amb el que s'accedeix tingui permisos per a modificar el directori on hi modificarem el contingut.

```
# cd /var/www/html/
# sudo mkdir forumsuoc
# sudo usermod -a -G forumsuoc ec2-user
# sudo chgrp -R forumsuoc /var/www/html/forumsuoc
```

Un cop tots els fitxers estiguin disponibles al servidor, cal configurar els paràmetres de configuració de l'aplicació. Cada sub-sistema té la seva pròpia configuració (*Javascript* client, *PHP*, *Node.js*). Per tant s'hauran d'editar 3 fitxers.

```
# cd /var/www/html/forumsuoc
# vim node/config.js
# vim scripts/chat_server.js
# vim uoc.php
```

El primer fitxer és un fitxer pur de configuració. En els altres dos només s'ha d'editar l'apartat d'edició a l'inici dels fitxers que ja està marcat de tal forma.

## 5.5. Crear la base de dades

**Fòrums UOC** utilitza una petita base de dades per ampliar les possibilitats que ofereixen les dades proporcionades per la *UOC* i per poder gestionar l'accés a l'eina de missatgeria instantània.

Per configurar la base de dades, primer hem d'accedir al monitor de SQL amb les credencials de super-usuari.

```
# mysql -u root -p password-de-root
```

Una vegada s'estigui dintre, el primer pas serà crear la base de dades.

```
CREATE DATABASE forums_uoc;
use forums_uoc;
```

El següent pas serà crear un nou usuari que serà amb el que accedirà la aplicació, de forma que no se li garantiran tots els permisos existents, només els que necessita per al seu funcionament.

```
GRANT SELECT,INSERT,UPDATE,DELETE ON forums_uoc.* TO
'uocuser'@'localhost' IDENTIFIED BY 'password-del-user';
```

En aquest punt ja podem crear les dos taules que necessita l'aplicació.

```
CREATE TABLE IF NOT EXISTS `amics` (
  `id` int(11) NOT NULL,
  `amic` int(11) NOT NULL,
  `photoUrl` varchar(150) NOT NULL,
  `nomAmic` varchar(150) NOT NULL,
  PRIMARY KEY (`id`,`amic`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```



```
CREATE TABLE IF NOT EXISTS `sessions` (  
  `id` int(11) NOT NULL,  
  `code` varchar(50) NOT NULL,  
  PRIMARY KEY (`id`,`code`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

## Instal·lació phpMyAdmin

*phpMyAdmin* és una eina que s'accedeix des de navegador i que permet accedir de forma visual a les bases de dades que hi han disponibles a un sistema. És molt útil si es necessita fer un seguiment ràpid del que està passant a una base de dades. A demés, la seva instal·lació és molt senzilla per els beneficis que aporta.

```
# cd /var/www/html  
# sudo wget  
http://sourceforge.net/projects/phpmyadmin/files/phpMyAdmin/4.0.7/php  
MyAdmin-4.0.7-english.tar.gz/download  
# sudo tar -zxvf phpMyAdmin-4.0.7-english.tar.gz  
# sudo mv phpMyAdmin-4.0.7-english phpMyAdmin
```

## 5.6. Executar l'aplicació

Finalment, ens hem d'assegurar que tots els serveis estan en marxa. Si no s'ha aturat de forma manual el servidor web i el servei de base de dades, haurien d'estar en marxa -si s'han seguit les instruccions de configuració-. I només falta la execució del servei de missatgeria instantània.

Com s'ha comentat anteriorment, el *Node.js* executa els serveis sota demanda, per tant, si volem que una aplicació s'executi en aquest servei, haurem d'especificar la ruta de l'arxiu principal de l'aplicació.

```
# node /var/www/html/forumsuoc/node/chat_server.js
```

Aquesta diferència de procediment és deguda a que el mateix node pot estar escoltant a ports diferents del servidor, i tenir una aplicació a cada port. Per tant, és més segur separar l'execució dels serveis per tal d'aïllar l'estat de cada aplicació i que la fallada d'una d'elles no faci caure a la resta. Per tal de fer que l'aplicació sempre estigui funcionant, es pot configurar un "dimoni" que s'encarregarà de comprovar que l'aplicació sempre estigui funcionant, i si no ho està, s'encarregarà d'iniciar-la de nou.

## 6. Avaluació de costos

Pot desenvolupament comporta un cost, i més com en el cas que ens ocupa que necessita mantenir una infraestructura per a servir el contingut als clients.

La primera despesa que ens trobem és la relacionada amb el desenvolupament. Tot i que el desenvolupament ha anat lligat a un desenvolupament lectiu, el cost temporal ha existit, i es convenient avaluar-lo com si es tractés d'un desenvolupament comercial.

El prototip presentat ha sigut el fruit del treball d'una persona en tres mesos. Traient la carrega de treball de la documentació, ens queden les tasques de gestió de requisits, anàlisi i disseny del projecte, desenvolupament de l'aplicació i gestió del servidor. Aquestes tasques han comportat la dedicació de 9 setmanes on s'estima que la dedicació setmanal puja a 20 hores. Això fa un total de 180 hores dedicades a la finalització del prototip. El cost horari és molt difícil d'avaluar ja que depèn de l'assignació del desenvolupador. És a dir, un treballador dintre d'una empresa tindrà un sou mitjà de 10€ a la hora, mentre que un treballador independent pot arribar a cobrar de 15€ a 30€ la hora.

Per a fer una estimació del cost de desenvolupament, partirem de 15€ a la hora que podria englobar els 2 casos. Si partim de les 180 hores, dona un total de **2.700€ de desenvolupament**. Recordar que aquest cost depèn de les qualitats del desenvolupador -o empresa desenvolupadora- i de la qualitat que en vulguem obtenir.

Un altre factor a tenir en compte són les despeses de manteniment del servei. El sistema requereix unes condicions de connexió a la xarxa que fan impossible tenir una màquina personal servint l'aplicació. La millor alternativa és contractar un servei d'allotjament web. Hi han molts tipus d'allotjament disponibles que ofereixen diferents opcions i preus. Per les característiques del projecte ha de ser un servei que permeti l'accés mitjançant consola de comandes. També seria una bona opció que estigués pròxim a Barcelona, degut a que la majoria d'estudiants de la universitat accedeixen de forma centrada a la ciutat Condal (Europa seria pròxim dintre de les característiques d'Internet).

Una bona alternativa seria el servei *EC2* que ofereix *Amazon*, i amb el que s'ha realitzat el període de desenvolupament. És un servei molt elàstic i escalable que permet pagar només per l'ús horari de les màquines que s'utilitzen. Per aquest projecte una màquina virtual d'ús general i de gama mitja costaria 0'077€ a la hora. Si contem que es vol tenir el servei disponible el 100% del temps, un mes està format per 744 hores, que fan un total de **57'3€ mensuals per màquina**. Una bona notícia és que per a clients nous, l'empresa *Amazon* ofereix un any de prova gratuïta amb 750 hores d'us per a màquines, com aquest projecte només necessita una màquina, podria estar funcionant de forma correcta durant 1 any sense cap despesa.

Com s'ha comentat anteriorment, també existeixen altres proveïdors, però els millors preus per aquest tipus de màquines -virtuals- no difereixen molt del servei comentat. Per exemple, l'empresa alemanya *Hezner*, ofereix servidors dedicats a 49€ mensuals. La diferència resideix amb les característiques de la màquina i la facilitat de configuració que ofereixen.

Aquests càlculs s'han realitzat per a nivells baixos d'usuaris simultanis, si l'aplicació es popularitza podria donar-se el cas de que les despeses de allotjament es dupliquessin. Aquesta situació té poques probabilitats de donar-se. En aquests moments la *UOC* té 50.000 estudiants, dels quals no més de 30.000 deuen d'estar cursant assignatures en aquest moment. La taxa de connectats simultàniament, estimant molt a la alça, no supera el 10% en qualsevol moment del dia. Això fa que en els moments més atapeïts del dia amb més connexions la pàgina oficial de la *UOC* té de 3.000 a 3.500 usuaris connectats simultàniament. El ús que fa l'aplicació del servidor que s'ha de contractar no és molt intensiu i podria suportar de 500 a 1.500 usuaris simultanis sense problemes. Per tant, es podria donar servei sense problemes a la meitat d'estudiants de la universitat.

També seria convenient tenir en compte que el servidor necessita un manteniment. El manteniment no ha de ser constant, però ha d'atendre a possibles mal funcionaments de la màquina, caigudes dels serveis que suporten l'aplicació, necessitat d'ampliar el nombre de màquines... Per tant, es necessitarà que una persona realitzi les tasques de administració del sistema. No són tasques suficients per emplear a un treballador ja que seria una despesa desproporcionada per la càrrega de treball, però s'ha de realitzar tant subcontractant un manteniment parcial, com assignant un treballador de l'empresa que serveixi l'aplicació, com del particular que decideixi encarregar-se del projecte.

## 7. Retorn de la inversió

Aquest és un dels punts més conflictius del projecte degut a les poques opcions que existeixen per, ja no treure rendibilitat del producte, sinó tan sols recuperar la inversió inicial.

Les característiques de **Fòrums UOC** fan que no sigui una bona opció utilitzar l'aplicació com un bé comercial. Per una banda, està dirigit a estudiants de la universitat oberta de Catalunya. Per tant, tenim que els clients potencials no siguin més de 50.000, sense marge d'errors, degut a que la universitat no té més estudiants. Aquest nombre és molt reduït si tenim en compte les taxes de conversió que es contemplen a les aplicacions en xarxa.

Per traduir aquesta idea en xifres, s'hauria de considerar dur a terme una acció comercial i de màrqueting per atreure clients. El cost de la mateixa no hauria de baixar de 1.500€ si vol que sigui efectiva. També seria convenient dedicar més hores de treball per afegir funcionalitats a l'aplicació i millorar el valor de la mateixa. Es podria considerar una taxa de conversió del 10% com a tot un èxit. Si es considera el cost de desenvolupament, el cost de treball futur i la despesa en publicitat situem la despesa a recuperar al voltant dels 6.000€ el primer any, i un cost de manteniment de 750€ de manteniment més el cost de publicitat els següents anys. En el millor dels casos s'hauria de comercialitzar l'aplicació per 1,50€ per treure rendiment el primer any, inclús fer que el preu sigui a mode de subscripció per pagar la quota de forma anual. També es pot combinar amb la oferta d'un període de prova gratuït per mostrar els avantatges del producte.

De totes formes aquestes dades estan calculades sobre una base molt més optimista que realista. Un 10% de conversió en un entorn tant limitat i parlant d'una aplicació de pagament que principalment no aporta contingut nou, sinó una visió més moderna i usable, és molt poc probable. Per tant no es recomana fer una inversió tant gran en model de negoci que té poques probabilitats de ser rendible.

Una altra opció, la qual s'està popularitzant molt en aplicacions informàtiques i mòbils, és el model *freemium*. Consisteix en oferir un producte de forma gratuïta però amb les seves funcionalitats més importants bloquejades per a usuaris de pagament. Aquesta aproximació ajuda a popularitzar el ús del producte ja que és molt més accessible al ser de lliure accés. També propicia que l'usuari creï dependència amb el ús de l'aplicació i afavoreixi les intencions de compra per part del client.

Malauradament, tampoc sembla una bona opció per al cas que ens ocupa. La despesa de manteniment augmentaria de forma exponencial comparada amb la taxa de conversió. Alhora implica deixar funcionalitats obertes que no es podrien incloure dintre del preu global del producte, fent que el valor del preu per usuari baixés.

La següent opció que s'ofereix és la d'oferir l'aplicació gratis amb anuncis. És a dir, establir publicitat en varies seccions de l'aplicació. La forma en que es canvia la

informació ajudaria a millorar el *CTR* (*Clicks Through Ratio*, o proporció de clics), ja que la pàgina només es carrega un cop, i la navegació esdevé de forma local, fet que redueix els refrescos de la pàgina. La unitat de mesura del *CTR* és pulsacions d'anuncis dividit entre el nombre de impressions d'anuncis, i en l'aplicació es reduiria molt el dividend, fent créixer el valor de conversió que es tradueix en diners.

La opció dels anuncis es pot combinar, de forma bastant natural, amb el model *freemium*, mostrant publicitat als usuaris que no generen despesa al sistema o oferint un producte de pagament per a eliminar els anuncis de la pàgina.

El *CTR* mitjà actualment es situa sobre el 0,10%, en el cas d'aquest projecte podria arribar a 0,15% degut a les seves característiques. Tot i això, es pot establir que de forma mitjana es paga a 0'10€ la pulsació a un anunci.

Aproximem que tenim 6 anuncis i es conta, en el millor dels casos, que existeixen 5.000 usuaris actius, que entren un de cada tres dies a l'aplicació, podem deduir que hi hauran una mitjana de 1.660 visitants únics al dia, els quals imprimiran 10.000 anuncis al dia. En aplicar la proporció de clics per pàgines vistes, tenim que existiran 15 pulsacions a anuncis que generaran 1.5€ al dia. Aquest fet suposa uns ingressos de 46'5€ mensuals, fet que no cobreix ni les despeses d'allotjament. Per tant, la base d'usuaris segueix sent molt baixa i per aconseguir recuperar la inversió es necessitaria una base d'usuaris de l'ordre de 2 o 3 vegades més gran.

Després de fer un anàlisi de les diferents opcions que existeixen per rendibilitzar es pot treure la conclusió de que és una aplicació amb poques garanties de ser rendible per si sola. No obstant, és una conclusió que cabia esperar degut a les característiques del projecte. **Fòrums UOC** no s'ha concebut com una aplicació comercial, més aviat s'esdevé com una eina de suport per a millorar la forma en que es comuniquen els estudiants dintre de la universitat. Per tant, podem concloure que per a que els usuaris de la *UOC* puguin gaudir dels seus beneficis, la universitat mateixa ha de subvencionar el seu manteniment i encarregar-se de gestionar el seu futur, ja que el motiu de la realització d'aquest projecte no va ser treure rendibilitat econòmica, sinó contribuir amb una eina de comunicació social i fluida per als estudiants de la universitat.

## 8. Treball futur

Tot i que el treball realitzat en aquest projecte ha donat lloc a un prototip plenament funcional en els requisits que es van plantejar inicialment, durant la realització d'aquest projecte han sorgit idees interessants per a millorar les funcionalitats que ja inclou i també per estendre-les i poder proporcionar una eina més transversal que pugui arribar a esdevenir d'ús gairebé obligat per a qualsevol estudiant de la *UOC* per a realitzar les tasques més quotidianes de la universitat.

La primera planificació que s'hauria d'abordar en un futur hauria de consistir en millorar les eines que ja s'ofereixen i afegir les funcionalitats que no s'han pogut implementar que ja estaven incloses en la planificació inicial. A continuació es pot trobar una llista dels requisits mínims per a que **Fòrums UOC** es pugui considerar una eina completa, llesta per a facilitar la vida dels estudiants de la universitat:

- **Millorar la gestió d'amistats.** En aquest moment, el sistema d'amistats consisteix en la possibilitat d'assignar usuaris com a amics i veure el seu estat de connexió en la llista de usuaris connectats a la barra del xat. També seria molt útil disposar d'una secció on veure tots els estudiants que han sigut marcats com a amistats per poder fer un seguiment i realitzar algunes tasques senzilles com eliminar una amistat.
- **Optimitzar l'ordenació de converses.** En aquest moment, les converses s'ordenen de forma cronològica tenint en compte l'ordenació que proposa l'Open-API. No s'han pogut fer més proves degut a les dades limitades que ofereix l'entorn de proves. No obstant, ordenar les converses per data d'última actualització milloraria el seguiment de les mateixes.
- **Millorar la informació menys rellevant.** Cada fòrum mostra la llista de converses, cada conversa mostra les dades del missatge inicial, el seu autor i la data d'inici. També seria útil poder visualitzar si hi han missatges no llegits i les dades de l'últim missatge que ha actualitzat la conversa.
- **Converses instantànies de grups.** Actualment només es poden iniciar converses entre dues persones. Seria molt interessant poder iniciar converses de grup amb la possibilitat d'invitar persones per afegir-se a la conversa.
- **Sales de xat per assignatures.** Els fòrums de les assignatures són molt utilitzats per consultar dubtes en moments concrets. El problema és que es pot tardar moltes hores a rebre una resposta. Per tant, poder disposar d'un espai on poder trobar companys d'assignatura i consultar dubtes de forma instantània seria una millora a tenir en compte.
- **Notificacions en viu.** En aquesta aplicació els esdeveniments es perceben de forma instantània degut a la seva estructura: connexió al xat, un usuari s'ha connectat, s'ha rebut un missatge nou... U sistema de notificacions emergents podria avisar de les últimes actualitzacions de l'estat de l'aplicació sense molestar a l'usuari i permetent que sempre estigués informat.

Si s'afegeix tot aquest gruix de característiques, es pot considerar ja una aplicació completa. No obstant, una aplicació completa que gestiona només comunicacions de cada assignatura juntament amb la incorporació de l'eina de missatgeria instantània. Amb tot aquest potencial, és obligatori plantejar-se la possibilitat de augmentar les funcionalitats inicials que s'han plantejat per a oferir molt més recursos a l'usuari. En la següent llista es pot trobar un recull d'algunes extensions que aportarien característiques noves a l'aplicació per oferir un servei molt més complet:

- **Eina de cerca.** És una característica que es va eliminar del plantejament inicial degut als problemes tècnics que van produir-se durant el desenvolupament del prototip. Tot i no ser una eina essencial, pot ser molt valuosa en el cas de voler cercar missatges antics entre un gran volum d'informació.
- **Bustia personal.** L'aplicació gestiona recursos de comunicació de la *UOC*, i la bustia personal n'és un d'ells. Tot i no estar tant relacionat amb les assignatures de l'usuari, seria una molt bona aportació al projecte.
- **Mur d'esdeveniments.** Tot i que un calendari tradicional no tindria una cabuda. Es podria transformar en un tauler d'esdeveniments pròxims per guiar de forma ràpida les necessitats de l'estudiant.
- **Eines de professorat.** L'aplicació està centrada en l'estudiant. La falta d'informació respecte de les eines de les que disposen i la informació que necessiten no ha fet possible adaptar la funcionalitat a les seves necessitats. No obstant, també seria interessant que poguessin desenvolupar les seves tasques mentre gaudeixen de l'ús de l'aplicació.

També cal valorar quins són els passos que s'haurien de seguir en un futur per tal de posar l'aplicació a disposició de tots els usuaris de la universitat.

El primer pas seria acabar de polir tots els aspectes del desenvolupament, això implica afegir les funcionalitats que s'han descrit com a imprescindibles per l'aplicació. Més endavant s'hauria d'escollir quin model de negoci es vol utilitzar per finançar almenys el manteniment de l'aplicació. Depenent del model de negoci, s'hauran d'efectuar alguns canvis per a traduir en beneficis el ús del servei. Una vegada tota la eina estigui adaptada, s'hauria d'escollir un pla d'allotjament que s'adeqüi a les necessitats del projecte per, finalment, ficar-la a disposició del públic general.

Idealment, la millor opció seria que la universitat integrés els serveis de forma pròpia per a que tots els usuaris poguessin gaudir dels seus beneficis, canviant la forma en que es mostren els fòrums de cada assignatura i incorporant l'eina de missatgeria com a element global del campus virtual.

## 9. Conclusions

Després de finalitzar aquest projecte focalitzat a realitzar un prototip funcional pensat per a millorar la forma en que es comuniquen els estudiants de la *UOC* es pot arribar a la conclusió de que hi ha possibilitats d'assolir una millora satisfactòria, funcional i que aportarà beneficis a l'estudiant.

L'estudi sobre la reorganització de la informació continguda en els recursos de comunicació existents demostra que és possible mostrar la informació agrupada per facilitar la comprensió. És més, les dades disponibles en els sistemes de persistència i de servei d'informació de la universitat disposen de les eines suficients per a satisfer els objectius marcats.

Alhora la elecció de tecnologies de client per a gestionar la navegació i obtenció d'informació han demostrat ser efectiva per a realitzar la tasca exigida i han complert amb els objectius de millora de la experiència d'usuari, reduint les esperes entre transicions de pàgines i aportant dinamisme davant de la interacció de l'usuari. No obstant, tot i demostrar un solvència, no destacaria la elecció d'aquest paradigma de control com a l'èxit de l'aplicació. Per tant, no supeditaria la necessitat de seguir amb aquesta elecció per a realitzar una modernització total, si la col·lisió amb les tecnologies actuals fos massa violenta.

Crec que l'èxit de l'aplicació recau sobre la filosofia d'estructura dels fòrums més que en el conjunt de tecnologies escollides, ja que segurament una altra elecció tecnològica hauria pogut resoldre el mateix problema.

En aquest projecte no solament s'ha realitzat un estudi de com millorar les eines existents, sinó que s'ha realitzat un estudi de com millorar la forma en que es comuniquen els estudiants. I per a complir aquest objectiu s'ha creat l'eina de missatgeria instantània. Sorprenentment, aquesta eina ha sigut la vida d'aquest projecte ja que li ha donat immediatesa. Crec que el fet de plantejar la eina com una característica més de l'aplicació, integrada en la mostra d'informació provinent de la universitat, li ha donat molta potència. És un recurs que es troba molt a faltar per part dels estudiants, que han de cercar eines externes per a poder treballar en grup, situació molt comú en les assignatures de la *UOC*.

Per tant, si recollim el objectiu inicial d'aquest projecte: "Millorar el mode en que els estudiants de la *UOC* es comuniquen", podem dir que s'ha assolit l'objectiu marcat, i que la filosofia es pot adaptar a les eines de la universitat.

Referent a les tecnologies utilitzades, he après a confiar una mica més amb els navegadors i la potència de *Javascript*. He comprovat que si es comencen els projectes de forma planificada, utilitzant unes llibreries de base adequades al projecte i s'utilitzen tècniques poc habituals en *Javascript*, com el prototipatge o la inclusió modular d'elements *HTML*, esdevé un llenguatge molt potent i capaç de gestionar projectes de gran envergadura. La meva principal por estava relacionada a com



gestionaria la quantitat d'operacions d'entrada/sortida, tement que les esperes penalitzessin la gestió de la interfície de l'usuari. Però no tan sols ho gestiona molt efectivament, sinó que aporta ajudes per a facilitar l'enviament constant d'informació d'estat a l'usuari per a que no es senti perdut en cap moment.

La gran majoria de tecnologies treballades eren conegudes, però no és el cas de *Node.js*. Era una eina amb la que tenia moltes ganes de treballar i mai havia tingut la oportunitat. M'ha sorprès molt la facilitat de desenvolupar aplicacions que semblen molt complexes a priori. També és molt interessant la orientació a esdeveniments, un paradigma diferent amb el que treballar però fàcil d'entendre i d'aprendre. Com a punt negatiu, remarcar que és una tecnologia molt jove (Fins a inicis de Juny de 2014 no ha tret la seva versió 1.0) que encara és complicada de gestionar des del punt de vista d'administració de sistemes.

Un dels conceptes més importants que he après amb **Fòrums UOC** és el fet d'entendre la planificació. Durant tota la meua etapa lectiva he tingut tasques i he après nocions sobre planificació i organització de projectes. Alhora, en la meua curta vida professional també he viscut la planificació per la seva absència. Però aquest projecte ha esdevingut una oportunitat per realitzar una planificació i poder gestionar el seu seguiment.

Estranyament, puc treure dues conclusions tant absolutes com contradictòries referents a la organització. Per una banda, he pogut observar que una bona planificació, inclús de varis mesos, ajuda molt a distribuir les càrregues de feina i a marcar petites fites que permeten arribar al final de la planificació sense desviar-se molt dels objectius marcats. No obstant, també he observat que, tot i tenir marcades unes fites, s'ha de saber ser flexible i poder reorganitzar les càrregues de feina en moments puntuals del calendari.

El fet de que la planificació es regia per blocs temporals amplis -normalment no menys d'una setmana- i que les fites marcades guardaven molta relació amb les entregues exigides, ha fet que, de forma global, s'hagi pogut seguir la planificació correctament. Però quan ens referim a que s'ha seguit correctament, vol dir que s'han complert les fites marcades dintre de la planificació. Fet que no vol dir que s'hagi seguit el pla setmanal fidelment.

Tot això és degut a que s'han trobat molts problemes d'estabilitat i de disponibilitat en el servei extern *Open-API* de la universitat. Com es comenta en l'apartat de desajustaments en la planificació, un terç del temps planificat per a realitzar el desenvolupament del projecte va ser inhàbil degut a que el servei no estava disponible. Aquest fet ha provocat greus reajustaments en un desenvolupament amb el temps tant ajustat. En alguns casos s'han tingut que intercanviar períodes dedicats a la documentació, cosa que no sempre ha sigut possible degut a la manca de recursos per una planificació diferent. En altres casos, simplement s'ha tingut que ajornar les tasques planificades, fet que ha obligat a modificar el nombre de característiques planificades inicialment.

Tot i que els imprevistos que han afectat a la planificació, han deixat algunes idees inicials sense cabuda al prototip final, també hi han altres funcionalitats que s'havien descartat anteriorment degut als terminis dels que es disposaven. A l'apartat de treball futur s'aborda amb més detall quines són les característiques concretes que s'haurien de desenvolupar per a completar l'aplicació.

Per tant no és erroni afirmar que l'aplicació no està llesta per a poder-se ficar a disposició dels usuaris en mode de producció. Encara seria necessari incorporar funcionalitats que completessin el producte. No gens menys, també seria recomanable explorar altres fonts d'informació que augmentarien el valor de l'aplicació mitjançant una oferta més transversal de funcionalitats del campus de la *UOC*.

Malauradament, com es pot observar a l'estudi de viabilitat, l'autofinançament de l'aplicació, sigui quin sigui el nivell de funcionalitats afegides, és clarament insuficient. Degut als pocs clients potencials, a la dificultat de popularització i a les despeses de manteniment, no esdevé una aplicació comercialitzable en si mateixa.

Tot i això si que podria existir com a mòdul afegit a la universitat. Una gestió més moderna dels recursos de comunicació i d'incorporació d'una eina de missatgeria instantània ajudaria a modernitzar la imatge de la *UOC* de cara a l'exterior. Alhora ajudaria a millorar els lligams entre els estudiants mitjançant el augment i millora de la comunicació entre ells, fet que augmentaria la retenció i fidelització de l'estudiant actiu. Econòmicament parlant, una major retenció de clients i l'allargament de la vida activa del mateix, faria augmentar el nombre de matricules al semestre, rendibilitzant el reduït impacte que suposaria la incorporació de les eines de **Fòrums UOC**, ja que la universitat ja disposa de la infraestructura necessària.

Des d'un punt de vista més personal, després d'haver estat cinc anys estudiant a la *UOC* de forma activa i d'estar a les portes d'acabar un Grau superior, m'hauria agradat poder disposar de la organització de fòrums projectada i de la eina de comunicació instantània creada.

## 10. Glossari

- **HTML**. Llenguatge de marques que defineix com s'estructuren les pàgines web. Pàgina web està formada per aquest llenguatge.
- **CSS**. Format de suport per al *HTML*, serveix per definir propietats estètiques dels elements continguts a una pàgina web.
- **JSON**. Format d'intercanvi de dades molt lleuger. Utilitza la notació d'objectes de *Javascript* S'utilitza molt per intercanviar dades per la xarxa degut a la seva lleugeresa i la facilitat de lectura.
- **Javascript**. Llenguatge de programació que utilitzen els navegadors web per executar comandes a les pàgines web.
- **PHP**. Llenguatge de programació molt utilitzat per servir pàgines web dinàmiques. Permet generar marques de *HTML* personalitzades de forma dinàmica.
- **MySQL**. Sistema de gestió de bases de dades. És molt utilitat degut a les seves possibilitats, i la seva gratuïtat.
- **Node.js**. Plataforma de desenvolupament que permet executar serveis web de forma fàcil i ràpida. Utilitza *Javascript* com a llenguatge de programació.
- **Socket.io**. Extensió de *Node.js* que millora i facilita l'establiment de connexions persistents tipus client-servidor.
- **WebSockets**. Implementació de connexions persistents per a navegadors web. El seu ús no està molt estès avui en dia.
- **WebStorage**. Sistema d'emmagatzemament local per a navegadors. Desa les dades en mode clau-valor.
- **Apache**. Servidor web de programari lliure.
- **OAuth**. Protocol d'autenticació que permet a una aplicació accedir a dades protegides de tercers. Alhora permet a un servei, donar accés a les seves dades de forma segura.
- **API**. Interfície de programació de aplicacions. Conjunt de funcions que permet la comunicació entre diferents components de programari.

- **Open-API.** Servei de dades per a tercers que ofereix la Universitat Oberta de Catalunya.
- **Campus Virtual.** Plataforma de la *UOC* on l'estudiant pot trobar tota la informació relativa a les seves assignatures i tasques relacionades amb la universitat.
- **Stakeholders.** Tot aquell conjunt de persones o entitats que tenen un interès amb una activitat.
- **Front-End.** Part del programari que s'executa a la banda del client.
- **Usabilitat.** Facilitat amb la que les persones poden utilitzar una eina.
- **Comunicació asíncrona.** Tipus de transferència de dades que no bloqueja l'execució d'un programa.
- **REST.** Qualsevol interfície de transferència de dades que utilitza el protocol *HTTP*.
- **Serialització de dades.** Procediment que consisteix en representar en una cadena de caràcters, el contingut i definició d'un objecte.
- **AJAX.** Tècnica de desenvolupament web per a realitzar comunicacions asíncrones.
- **SFTP.** Protocol de nivell d'aplicació que proporciona la funcionalitat necessària per a la transferència y manipulació d'arxius de forma fiable i segura.

## 11. Bibliografía

1. [1] *OAuth Community Site*, <http://oauth.net/>. [Consulta: 02-03-2014]
2. [2] *HTML5 Web Storage*, [http://www.w3schools.com/html/html5\\_webstorage.asp](http://www.w3schools.com/html/html5_webstorage.asp). [Consulta: 02-03-2014]
3. [3] *WebSocket.org, About WebSocket*, <http://www.websocket.org/aboutwebsocket.html>. [Consulta: 02-03-2014]
4. [4] *Wikipedia, la enciclopedia libre, Diagrama de Gantt*, [http://es.wikipedia.org/wiki/Diagrama\\_de\\_Gantt](http://es.wikipedia.org/wiki/Diagrama_de_Gantt). [Consulta: 03-03-2014]
5. [5] *Wikipedia, la enciclopedia libre, Representational State Transfer*, [http://es.wikipedia.org/wiki/Representational\\_State\\_Transfer](http://es.wikipedia.org/wiki/Representational_State_Transfer). [Consulta: 10-03-2014]
6. [6] *JSON*, <http://json.org/>. [Consulta: 10-03-2014]
7. [7] *XML Introduction, What is XML?*, [http://www.w3schools.com/xml/xml\\_what\\_is.asp](http://www.w3schools.com/xml/xml_what_is.asp). [Consulta: 10-03-2014]
8. [8] *Socket.IO*, <http://socket.io/>. [Consulta: 10-03-2014]
9. [9] *Node.js*, <http://nodejs.org/>. [Consulta: 10-03-2014]
10. **Huges, Tom**. *A Million Connections...and Beyond! - Node.js at Scale*, 2011, <http://www.youtube.com/watch?v=AH7kw8sKefg>. [Consulta: 02-05-2014]
11. **Ubl, Malte. Kitamura, Eiji**. *HTML5 Rocks, Introducing WebSocket: Bringing Sockets to the Web*, 2010, <http://www.html5rocks.com/es/tutorials/websockets/basics/>. [Consulta: 02-05-2014]
12. *How to node, Node.js*, <http://howtonode.org/>. [Consulta: 05-05-2014]
13. *HTML5 Rocks - Un recurso para desarrolladores de HTML5 para una Web abierta*, <http://www.html5rocks.com/es/>. [Consulta: 15-05-2014]
14. *1 Introduction — HTML: The Living Standard - Edition for Web Developers*, <http://developers.whatwg.org/introduction.html>. [Consulta: 15-05-2014]

15. **Pilgrim, Mark.** *Dive into HTML5*, <http://diveintohtml5.info/> . [Consulta: 15-05-2014]
16. *Javascript*, Mozilla Developer Network, <https://developer.mozilla.org/es/docs/JavaScript> . [Consulta: 19-05-2014]
17. *Wikipedia, la enciclopedia libre* , *Javascript*, <http://es.wikipedia.org/wiki/JavaScript> . [Consulta: 19-05-2014]
18. *jQuery*, <http://jquery.com/> . [Consulta: 19-05-2014]
19. **Solberg, Åkre.** *Javascript Oauth Library*, 2012 , <https://github.com/andreassolberg/jso> . [Consulta: 20-05-2014]
20. **Geisendörfer, Felix.** *Debuggable, Node.js Consulting, Understanding node.js*, 2010, <http://debuggable.com/posts/understanding-node-js:4bd98440-45e4-4a9a-8ef7-0f7ecbdd56cb> . [Consulta: 20-05-2014]
21. **Stoimen.** *Diving into Node.js – Introduction & Installation*, 2010, <http://www.stoimen.com/blog/2010/11/16/diving-into-node-js-introduction-and-installation/> . [Consulta: 20-05-2014]
22. *Manual de PHP, ¿Qué es PHP?*, <http://www.php.net/manual/es/intro-what-is.php> . [Consulta: 20-05-2014]
23. *MySQL, The world's most popular open source database* , <http://www.mysql.com/> . [Consulta: 20-05-2014]
24. *Wikipedia, la enciclopedia libre*, *MySQL*, <http://es.wikipedia.org/wiki/MySQL> . [Consulta: 20-05-2014]
25. *Wikipedia, la enciclopedia libre* , *Publicidad en Internet*, [http://es.wikipedia.org/wiki/Publicidad\\_online](http://es.wikipedia.org/wiki/Publicidad_online) . [Consulta: 01-06-2014]
26. *Wikipedia, la enciclopedia libre*, *SSH File Transfer Protocol*, [http://es.wikipedia.org/wiki/SSH\\_File\\_Transfer\\_Protocol](http://es.wikipedia.org/wiki/SSH_File_Transfer_Protocol) . [Consulta: 02-06-2014]
27. **Cabrera, Àlex.** *All you must know about mobile games*, 2013, <http://www.slideshare.net/powdercode/all-you-must-know-about-mobile-games-28843443> . [Consulta: 01-06-2014]

## **12. Annexos**

## 12.1. Annex 1. Manual d'usuari

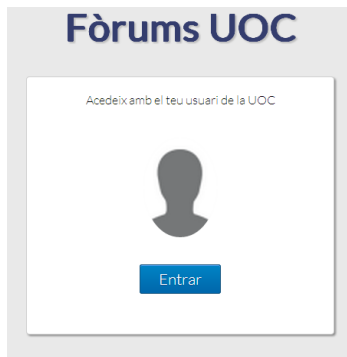
**Fòrums UOC** és una eina externa a la universitat que accedeix a les dades del campus virtual per a mostrar la informació continguda als recursos de comunicació als que te accés l'estudiant.

La forma en que organitza la informació de l'estudiant és el que la fa una aplicació especial, ja que distribueix els missatges en converses, facilitant la lectura i el seguiment del progrés de les assignatures.

Ahora estén les possibilitats de comunicació del campus mitjanant una eina de missatgeria instantània, que permetrà als estudiants d'una mateixa aula mantenir conversacions en temps real, així com la possibilitat de marcar altres estudiants com a amistats per facilitar la comunicació amb els mateixos.

### Iniciar sessió

El primer pas que ha d'efectuar l'usuari consisteix a iniciar la sessió. En la pantalla de presentació es pot trobar al centre un botó amb el text **entrar**. Pulsant aquest botó l'usuari serà redirigit als servidors de la **UOC** per tal d'iniciar la sessió.



Un cop s'arriba als servidors de la **UOC**, l'usuari ha d'introduir les seves credencials de la **UOC**. Aquest sistema impedeix que **Fòrums UOC** gestioni en cap moment informació secreta de l'estudiant, millorant la seva privacitat. Més endavant s'ha d'autoritzar l'ús de l'aplicació i introduir un codi de seguretat. Finalment es mostren una llista de permisos, on s'han preseleccionat un subconjunt d'ells per a l'ús de l'aplicació. És molt important **acceptar** com a mínim **aquests permisos** per poder gaudir de totes les funcionalitats de l'aplicació.





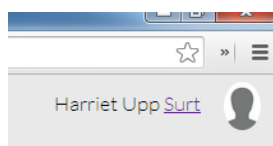
Els passos a seguir en l'entorn de la universitat poden ser lleugerament complicats, afortunadament cada pantalla està acompanyada d'informació i ajuda suficient per solucionar tots els passos.

Un cop acceptem els permisos, es redirigirà a l'usuari a l'aplicació on es mostrarà el contingut del campus virtual.

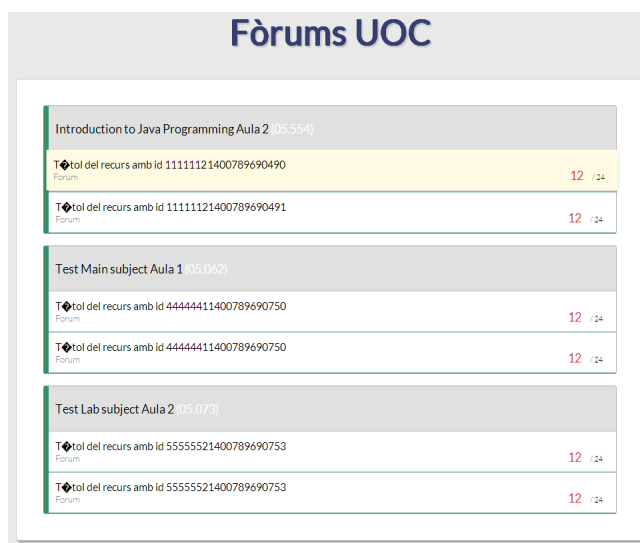
## Campus virtual

El campus virtual es compon d'una sèrie de pantalles encarregades a mostrar tota la informació continguda en els recursos de comunicació de totes les aules a les que té accés l'estudiant.

En tot moment, apareix situat a la part superior, un resum de l'usuari que ha iniciat la sessió amb un enllaç que ofereix la possibilitat de **sortir del sistema**.

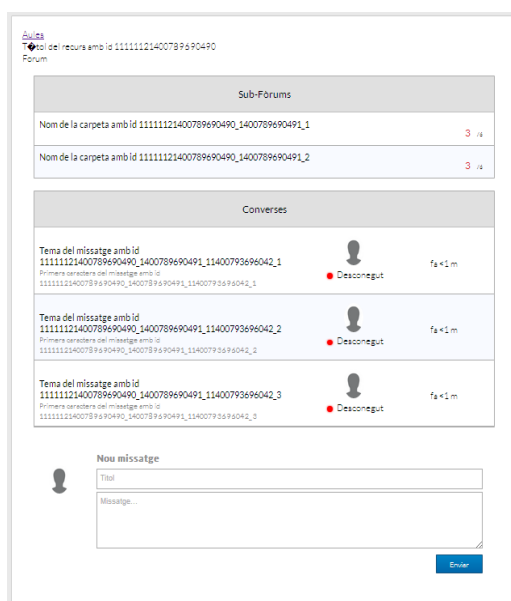


Inicialment es mostren les aules a les que té accés l'estudiant, separades en bloc i identificades per el nom de l'aula, el seu codi identificador i el color que la distingeix. Cada aula conté una llista on apareixen tots els recursos de comunicació on té accés l'estudiant (Fòrums, taulers i debats). Cada fila conté el nom del recurs, una petita descripció i un resum del nombre de missatges i els missatges no llegits que apareixen destacats. Per **accedir a un recurs de comunicació** en concret, només cal fer clic a sobre de la seva fila.



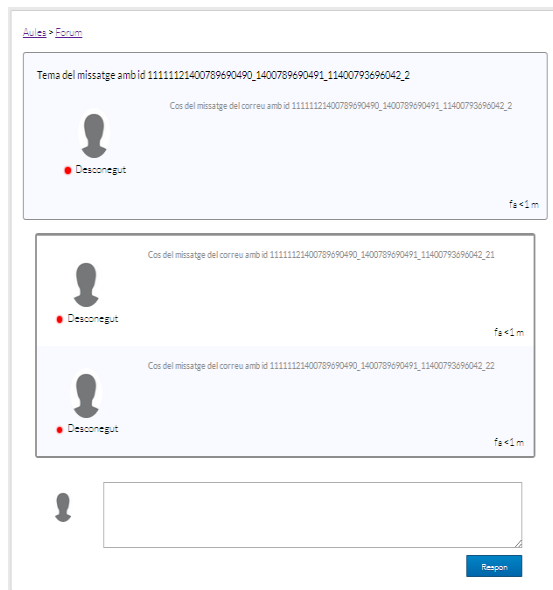
Com es podrà apreciar amb tots els elements interactius (files seleccionables, botons, enllaços...), la fila ofereix una interacció que reacciona al pas del ratolí per a indicar que aquell element pot realitzar una acció i facilitar la comprensió del seu comportament.

Un cop seleccionem un recurs de comunicació s'obrirà la informació que conté. Pot estar format per una secció de sub-fòrums, una llista de converses i un formulari per l'enviament d'un nou missatge. Aquesta informació pot canviar depenent del tipus de recurs ja que no es poden enviar missatges (com a estudiant) als taulers, i els taulers no acostumen a tenir sub-fòrums.



- La **llista de sub-fòrums** te un comportament molt similar a la de una aula, per tant, al accedir a un sub-fòrum veurem el seu contingut com si fos un fòrum normal.
- La **llista de converses** està formada també per files, però en aquest cas hi trobarem un resum del missatge, un detall de l'usuari que l'ha iniciat i el temps que fa que ha estat actualitzat. Cada fila també és un element seleccionable que ens portarà al contingut de la conversa seleccionada.
- Si el recurs ho permet, també podrem trobar al final de la secció, un formulari per enviar un **nou missatge**. En aquesta secció podrem incloure un títol i un text per a que s'envii com una conversa nova dintre del fòrum que s'està visualitzant.

La visualització de les converses està formada per un detall destacat del **missatge inicial** seguit d'una **llista de respostes**, per finalitzar amb un formulari per a enviar una resposta nova. Tant el detall com les respostes, contenen la informació necessària que necessita el lector: detall de l'usuari emissor, cos del missatge i data de publicació. Ahora, el missatge inicial també mostra el títol de la conversa. Al final de la secció es



pot trobar el **formulari de resposta**. Aquest només consisteix en un camp de text on escriure el missatge ja que el títol està relacionat amb el missatge inicial. En totes les seccions més endinsades a la jerarquia, apareix un **menú de navegació** que ens permetrà tornar a estadis anteriors per tal de facilitar la navegació.

Tots els formularis d'enviament de missatges envien automàticament el missatge i el mostren com una resposta/missatge més en la llista.

Els detalls dels usuaris estan formats per la seva imatge de perfil, nom d'estudiant, estat de connexió en el xat (connectat o desconnectat) i una icona per afegir com a amestat (si no ho és encara).

## Eina de missatgeria instantània

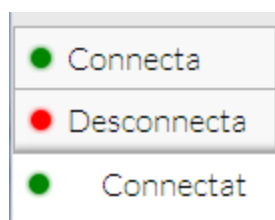
A la part inferior de la finestra del navegador es situa l'**eina de missatgeria instantània**. Està formada per una barra fixa amb tres seccions diferenciades. A la esquerra de tot trobarem el menú de connexió. A la dreta es situa la llista d'usuaris connectats i al centre un espai per a les converses obertes.

L'eina de missatgeria s'inicia de forma automàtica quan l'aplicació rep a l'usuari després de autenticar-se als servidors de la *UOC*. Passaran però uns quants segons fins que la connexió estigui llesta. No obstant, podrem seguir en tot moment l'estat de la connexió gràcies al menú de connexió.

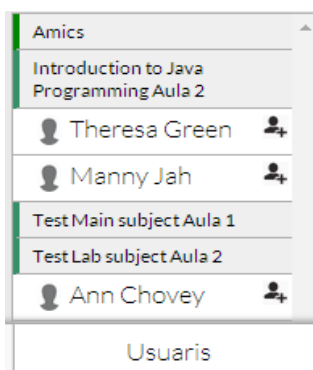
En cas de no aparèixer la barra inferior, significa que el servidor no està funcionant correctament o que el servei de xat està aturat.

Al **menú de connexió** podrem observar 2 seccions: l'estat actual i la configuració:

- L'estat actual pot tenir 3 valors: connectat, desconnectat i connectant. Depenent de l'estat en el que ens trobem, podrem utilitzar l'eina de missatgeria instantània i apareixerem com a connectats, o no.
- La configuració permet canviar de connectat a desconnectat i viceversa.

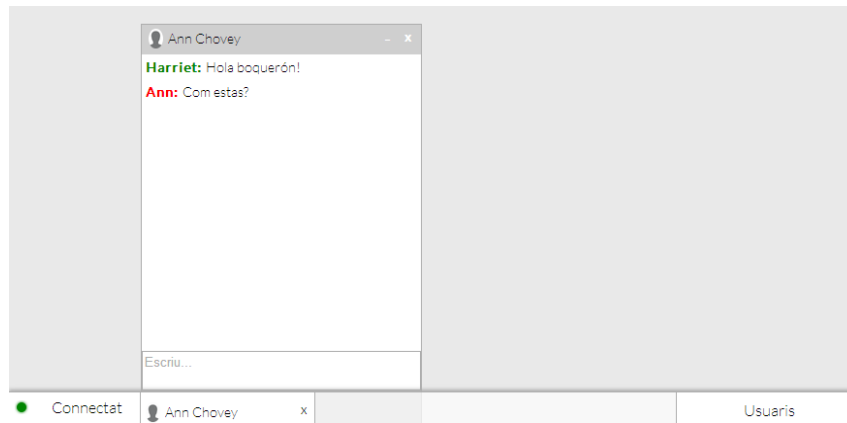


L'usuari pot desplegar la **llista d'usuaris connectats** polsant el botó "Usuaris" situat a la dreta de la barra. S'obrirà un menú que conté tots els usuaris connectats, distribuïts per aules en les que participen. En aquesta llista també hi ha una secció que conté la llista de les amistats de l'estudiant. Cada usuari d'aquesta llista és seleccionable, i al polsar en un company s'obrirà una finestra de conversa per a poder parlar en temps real. Els usuaris que no són amics, també tenen la opció de convertir-los en amics.



Les converses apareixen situades per sobre de la barra de xat, juntament amb una pestanya inferior, situada a la part central de la barra de xat, que serveix com a ancora per a controlar totes les converses disponibles. Per tant, la barra inferior té un ús similar a la barra d'inici del Windows.

Les **finestres de conversa** tenen un encapçalament on apareix informació del usuari amb el que s'està parlant i un indicador que mostra si està escrivint en aquest moment, òbviament només apareixerà quan l'altre usuari estigui escrivint. Seguidament podem trobar el cos de l'intercanvi de missatges que estarà format de files amb missatges dels dos usuaris. A la part inferior es troba un camp de text per introduir els missatges. Aquest camp de text no té botó d'enviar, l'enviament dels missatges es produeix polsant la tecla "Enter".



Les **pestanyes** ofereixen una forma de controlar la visualització de les converses actives. Al mateix temps, només una finestra pot estar visible. Per tant, si es selecciona una pestanya diferent, s'amagarà la conversa anterior per a que es pugui mostrar la conversa actual. En el cas que es rebí un missatge extern en una conversa que no estigui activa, es mostrarà de forma destacada, la pestanya que contingui informació actualitzada.

Les converses es poden obrir, tancar i minimitzar utilitzar diferents elements:

- Es poden obrir converses: seleccionant un usuari de la llista d'usuaris connectats, seleccionant el nom d'usuari quan aquest és l'emissor d'un missatge, seleccionant una pestanya que estava minimitzada, o rebent un missatge nou extern per primer cop.
- Es pot minimitzar una conversa: seleccionant el botó minimitzar que apareix a la capçalera de la conversa o seleccionant una pestanya que es visible.
- Es pot tancar una conversa seleccionant els botons de tancar que apareixen a la pestanya de totes les converses i a la capçalera de les mateixes.

Tot i que les pestanyes poden semblar molt grans i que hi ha poca possibilitat de mostrar converses, el sistema ha estat pensat per adaptar-se a pantalles més grans i més petites. Això implica que les pestanyes redueixen la seva mida horitzontal (fins a un mínim) per a poder mostrar les màximes pestanyes possibles i que alhora disposin del màxim d'espai possible per mostrar la seva informació.

## 12.2. Annex 2. Diagrama de Gantt

