



**Universitat Autònoma
de Barcelona**

Campus 2.0 - Disseny d'una xarxa social per escoles

Memòria del projecte
d'Enginyeria Tècnica en
Informàtica de Sistemes

realitzat per

Fernando Pradas Monllor

i dirigit per

Rafael Fernández González

Escola d'Enginyeria

Sabadell, juny de 2012

El sotasignat, *Rafael Fernández González*,
professor de l'Escola d'Enginyeria de la UAB,

CERTIFICA:

Que el treball al que correspon la present memòria
ha estat realitzat sota la seva direcció per

Fernando Pradas Monllor

I per a que consti firma la present.
Sabadell, *juny* de **2012**

Signat: *Rafael Fernández González*

FULL DE RESUM – PROJECTE FI DE CARRERA DE L'ESCOLA D'ENGINYERIA

Títol del projecte: Campus 2.0: Disseny d'una xarxa social per escoles	
Autor: Fernando Pradas Monllor	Data: Juny de 2012
Tutor: Rafael Fernández González	
Titulació: Enginyeria Tècnica en Informàtica de Sistemes	
Paraules clau (mínim 3) <ul style="list-style-type: none">• Català: Xarxa social, Grafs, HTML, PHP, JavaScript, jQuery, MariaBD, OQGraph.• Castellà: Red social, Grafos, HTML, PHP, JavaScript, jQuery, MariaBD, OQGraph.• Anglès: Social network, Graphs, HTML, PHP, JavaScript, jQuery, MariaBD, OQGraph.	
Resum del projecte (extensió màxima 100 paraules) <ul style="list-style-type: none">• Català:<p>Disseny i desenvolupament d'una aplicació web de xarxa social orientada a estudiants i professors d'una escola o institució educativa. Una eina que pretén millorar la interacció entre alumnes i professors mitjançant un mateix espai concurrent i virtual on poder comunicar-se i compartir opinions.</p>• Castellà:<p>Diseño y desarrollo de una aplicación web de red social orientada a estudiantes y profesores de una escuela o institución educativa. Una herramienta que pretende mejorar la interacción entre alumnos y profesores mediante un mismo espacio concurrente y virtual donde poder comunicarse y compartir opiniones.</p>• Anglès:<p>Design and development of social network oriented to students and teachers from a school or educational institution. A tool designed to improve interaction between students and teachers by using a same virtual space where they able to communicate and share opinions.</p>	

Índex

1	Introducció	1
1.1	Descripció	1
1.2	Objectius	1
1.3	Estat de l'art	2
1.4	Motivacions	3
1.5	Organització de la memòria	4
2	Estudi de viabilitat	5
2.1	Introducció	5
2.1.1	Definicions, acrònims i abreviacions	5
2.1.2	Descripció de la situació a tractar	6
2.1.3	Situació actual i context	7
2.1.4	Parts interessades (<i>stakeholders</i>)	8
2.1.5	Objectius generals	8
2.2	Descripció del sistema	9
2.2.1	Descripció física	9
2.2.2	Perfils d'usuari del sistema	10
2.2.3	Requisits funcionals	12
2.2.4	Requisits no funcionals	13
2.2.5	Restriccions	14
2.2.6	Normatives i legislació	14
2.3	Objectius	15
2.3.1	Crítics	15
2.3.2	Prioritaris	16
2.3.3	Secundaris	16
2.4	Recursos	17
2.4.1	Hardware	17
2.4.2	Software	18
2.4.3	Humans	18
2.5	Anàlisi de costos	19
2.5.1	Cost del material	19
2.5.2	Cost del personal	20
2.5.3	Pressupost	20

2.6	Alternatives.....	21
2.7	Solució proposada.....	25
2.8	Metodologia i WBS.....	26
2.8.1	Metodologia.....	26
2.8.2	Fases i activitats.....	26
2.8.3	Diagrama WBS.....	27
2.9	Calendari i planificació.....	28
2.9.1	Taula de tasques.....	28
2.9.2	Diagrama de Gantt.....	29
2.10	Avaluació de riscos.....	29
2.10.1	Llista de riscos.....	29
2.10.2	Catalogació.....	30
2.10.3	Pla de contingència.....	30
2.11	Conclusions.....	31
3	Anàlisi de requeriments.....	32
3.1	Introducció.....	32
3.2	Requeriments funcionals.....	32
3.2.1	Inici.....	32
3.2.2	Pàgines d'usuari.....	32
3.2.3	Pàgines de grup.....	33
3.2.4	Pàgines d'assignatura.....	33
3.2.5	Taulell d'anuncis.....	34
3.2.6	Xat.....	34
3.2.7	Cerca.....	35
3.2.8	Administració d'usuari.....	35
3.2.9	Peticions.....	36
3.2.10	Administració general.....	36
3.3	Requeriments no funcionals.....	37
3.3.1	Requeriments de rendiment.....	37
3.3.2	Requeriments de disseny.....	37
3.3.3	Requeriments de seguretat.....	37
3.4	Casos d'ús.....	38
3.4.1	Inici.....	39
3.4.2	Pàgina d'usuari.....	40
3.4.3	Pàgina de grup.....	41

3.4.4	Pàgina d'assignatura	43
3.4.5	Taulell d'anuncis	44
3.4.6	Xat	45
3.4.7	Cerca.....	45
3.4.8	Administració d'usuari.....	46
3.4.9	Peticions.....	47
3.4.10	Administració general	48
4	Disseny	50
4.1	Introducció.....	50
4.2	Base de dades.....	50
4.2.1	Base de dades relacional	52
4.2.2	Base de dades orientada a grafs.....	54
4.3	Estructura de classes.....	59
4.3.1	Diagrama de classes	60
4.3.2	Interfícies	61
4.3.3	ConnexioBD	64
4.3.4	Fòrum.....	64
4.3.5	Graf	65
4.3.6	Usuari.....	66
4.3.7	Grup	67
4.3.8	Assignatura	68
4.3.9	Post.....	68
4.3.10	Comentari.....	69
4.3.11	Arxiu	69
4.3.12	Article	69
4.3.13	Chat.....	70
4.3.14	Cerca.....	71
4.3.15	Token	72
4.3.16	Tipus.....	73
4.4	Diagrames de seqüència.....	73
4.4.1	Registre usuari.....	74
4.4.2	Login.....	75
4.4.3	Publicar article usuari.....	75
4.4.4	Modificar dades usuari.....	76
4.4.5	Expulsar membre grup.....	77

4.4.6	Modificar dades grup	78
4.4.7	Cercar usuaris.....	79
4.4.8	Administrar amics	80
4.4.9	Administrar alumnes	81
4.4.10	Enviar petició d'amistat.....	82
4.5	Interfície.....	83
5	Implementació.....	88
5.1	Introducció.....	88
5.2	Eines tecnològiques.....	88
5.2.1	Llenguatges de programació	88
5.2.2	Llibreries utilitzades	90
5.2.3	Entorns de programació	92
5.2.4	Servidor web	93
5.2.5	Sistema Gestor de Bases de Dades (SGBD)	93
6	Proves.....	95
6.1	Proves d'unitat	95
6.2	Proves d'integració.....	95
6.3	Proves de regressió.....	96
6.4	Proves d'acceptació	96
6.5	Eines tecnològiques.....	96
7	Conclusions	98
8	Bibliografia.....	102

1 Introducció

1.1 Descripció

El que es vol fer és desenvolupar una xarxa social orientada a estudiants i professors d'una mateixa escola o institució educativa, i que anomenarem Campus 2.0. El registre a aquesta estaria restringit, per tant, a usuaris d'un mateix centre, tot i que això estarà controlat per un administrador. Una vegada registrats al sistema, per una banda els alumnes podrien realitzar diverses accions com publicar a la seva pàgina de perfil, enllaçar amestat amb altres usuaris i crear grups de debat entre d'altres. Per altra, els professors, a part de l'anterior, podrien a més crear i organitzar assignatures, matricular a alumnes, penjar material docent, notícies, etc.

Per tant, en l'aplicació a desenvolupar, tindrem diversos elements bàsics com pàgines de perfil, grups de debat i pàgines d'assignatures que conformarien en conjunt tota l'aplicació i que només serien accessibles a una part dels usuaris. Per exemple, només estaria permès accedir a la pàgina d'una assignatura als alumnes matriculats a aquesta assignatura, o permetre accedir a la pàgina de perfil d'un usuari només als seus amics.

En aquesta aplicació també hi haurà altres elements com un cercador, un xat o un taulell d'anuncis, entre d'altres.

1.2 Objectius

L'elecció d'aquest projecte ha estat una decisió personal i portant-la a terme es busca no només accomplir uns objectius acadèmics sinó també uns objectius personals, així com també accomplir uns certs objectius principals que es pretén que compleixi l'aplicació final.

- L'objectiu principal d'aquest desenvolupament és crear una xarxa social amb la qualitat suficient, i novetats addicionals que la facin única, tant com per poder competir amb altres xarxes socials en el mateix mercat. També es busca generar una experiència satisfactòria als usuaris que l'utilitzin, i així poder considerar-se com una eina útil, senzilla d'utilitzar i d'ús freqüent per aquests. Així com també millorar la relació de l'alumnat amb el seu centre d'estudis i servir com a eina que ajudi a l'aprenentatge.
- Els objectius personals que es volen assolir són aprendre a portar a terme un projecte real des de la idea inicial fins a la seva implementació. Accomplir amb un cert nivell de coneixements tècnics, tan a nivell de programació com de gestió de projectes. I obtenir

la capacitat de poder organitzar des del principi totes les tasques necessàries en la implementació d'un projecte i l'elecció d'un mètode concret de desenvolupament. Per al final documentar tot aquest procés en una memòria escrita.

1.3 Estat de l'art

En el món actual les xarxes socials a Internet han permès a milions de persones interactuar entre elles de forma fàcil en un espai virtual, a qualsevol moment i sense importar la distància física.

Existeixen diverses xarxes socials, tant de caràcter general com específiques, i moltes són orientades a estudiants. D'aquestes últimes han aparegut diverses en els darrers anys i la seva finalitat principal, o bàsica, és la de permetre a alumnes d'una mateixa escola o universitat compartir apunts o treballs, tot i que han anat incorporant característiques de caràcter més lúdic i social. Actualment en actiu es poden trobar alguns exemples d'aquest tipus de xarxes socials, entre les que podríem destacar les següents:

- <http://www.patatabrava.com>: És una xarxa social universitària orientada a estudiants i futurs estudiants universitaris, i on aquests poden penjar apunts, posar-se en contacte, etc. Té diversos serveis socials com el de compartir cotxe i compartir pis, entre d'altres. Per accedir es pot registrar qualsevol usuari amb accés a Internet.
- <http://cv2008.uab.cat>: És una xarxa social interna (ISN)¹ i que és utilitzada oficialment a la Universitat Autònoma de Barcelona com a medi de comunicació entre professors i alumnes. El professorat pot administrar assignatures i penjar apunts, exàmens, problemes, notes, notícies, etc. Existeixen diversos elements com la wiki², fòrum i la missatgeria que permeten als usuaris, tant professors com alumnes comunicar-se entre ells. És, en el sentit educatiu, una solució bastant completa. Per accedir no és necessari cap registre, només ser estudiant de la UAB.

¹ Una ISN és una comunitat tancada i privada que es compon d'un grup de persones dins d'una empresa, associació o organització, o d'una escola, en aquest cas.

² Una wiki és un lloc web on les diverses pàgines que el conformen poden ser editades per múltiples voluntaris o usuaris a través del navegador web.

- <http://mi.rincondelvago.com>: És una xarxa social desenvolupada pels creadors de *rincondelvago.com* i actualment en fase de desenvolupament. Està orientada a estudiants i permet registrar a alumnes d'ESO, batxillerat i cicles formatius residents en Espanya. Els usuaris poden penjar apunts de les seves assignatures, experiències, fotografies, etc. També es poden establir amistats amb companys altres usuaris.
- <http://moodle.org/> : És un Sistema d'Administració de Cursos (LMS), tot i que també pot considerar-se com a xarxa social, que s'instal·la en un servidor i que permet principalment al professorat d'una institució educativa crear i administrar cursos. Als que poden apuntar-se els alumnes i realitzar comentaris, test d'avaluació, publicar als fòrums, a la wiki, etc. A part d'això té diverses característiques addicionals més enfocades a la interacció social entre els seus usuaris. El registre està controlat per la institució educativa que n'és administradora de la plataforma.

És en aquest context, on es pretén crear una xarxa social totalment diferent a les que acabem de comentar. Aquesta nova xarxa social, com hem dit, estaria orientada tant a estudiants com a professors i diferiria diversos aspectes d'aquestes. Per una part, seria una xarxa social interna (ISN) com el "Campus Virtual de la UAB", i per altra, no només estaria orientada a ús educatiu, sinó també a ús lúdic. I on l'alumnat tindria un major nivell de llibertat per expressar les seves idees al que podem veure en altres xarxes d'aquestes característiques.

Per tant, i en general, podríem destacar dos vessants principals en l'aplicació, la vessant educativa i la vessant lúdica o d'oci, i és al voltant d'aquestes on es vol desenvolupar el projecte.

1.4 Motivacions

Amb la realització d'aquest projecte he buscat crear una eina útil i que permeti a estudiants i professors, d'una escola o qualsevol institució educativa, comunicar-se entre ells en un espai virtual mitjançant internet.

Donar un aspecte més informal a l'educació del que s'imparteix actualment com a model educatiu és una motivació addicional. Habilitar la interacció entre alumnes i professors en un mateix espai concurrent i virtual crec que enriquirà els aspectes més formals de l'educació.

També la motivació de millorar la relació de l'alumnat amb el seu centre d'estudis i també, servir com a eina d'ajut a l'aprenentatge de l'alumnat, tot en conjunt.

Tot per després, poder presentar aquesta aplicació a la societat educativa com una eina útil que permet millorar la relació entre els estudiants i el professorat i que busca també millorar els resultats acadèmics de l'alumnat.

1.5 Organització de la memòria

La memòria està organitzada en 8 capítols. Al primer capítol es fa una introducció al que es vol realitzar al projecte, les motivacions i l'estat de l'art. Després, al segon capítol es fa un estudi de viabilitat del projecte per determinar si es tracta d'un projecte realitzable i viable. Al tercer capítol es fa una anàlisi de requeriments de l'aplicació. Als capítols 4, 5 i 6 es documenten les fases de desenvolupament del projecte, que són el disseny, la posterior implementació i per últim les proves. Aquests capítols s'han anat documentant a mesura que es desenvolupava l'aplicació. Ja per acabar, al capítol 7 s'extreuen les conclusions de la realització del projecte així com possibles millores i al capítol 8 es pot veure una bibliografia dels recursos utilitzats per a la seva realització.

2 Estudi de viabilitat

2.1 Introducció

Amb l'estudi de viabilitat es pretén determinar si aquest projecte pot resultar un projecte realitzable i també viable, no només pel que respecta a la seva idoneïtat al context actual, si no també als seus beneficis econòmics. No hem d'oblidar que la realització d'aquest projecte es fa en un marc real i per tant, aquest ha d'acabar tenint beneficis, com tot projecte empresarial. És per això que, amb l'estudi de viabilitat, analitzarem els objectius que es volen assolir i les necessitats, tant de recursos humans com materials, que es requereixen per acomplir aquests objectius. Així com també els costos que impliquen aquests recursos i que repercuteixen directament en el cost final d'aquest projecte.

Per altra banda, analitzarem les alternatives que es presenten per al final acabar escollint la que més s'adapti a les necessitats del projecte, i que no necessàriament ha de ser la menys costosa. Per últim escollirem una metodologia de desenvolupament i farem una organització del projecte en tasques més concretes.

2.1.1 Definicions, acrònims i abreviacions.

En l'estudi de viabilitat s'utilitzen certs conceptes tècnics que cal conèixer:

Servidor web: És el software que s'encarrega de cedir i respondre a peticions per part d'un client, com pot ser un navegador web, i de forma adequada. Per exemple, a un usuari sol·licita accedir a una pàgina web, el servidor li envia les dades o codi necessari i és el navegador el que s'encarrega de mostrar la pàgina a partir d'aquest codi.

SGBD: Sistema de Gestió de Bases de Dades, és el software que s'encarrega de fer d'interfície entre la base de dades i les aplicacions o usuaris que l'utilitzen. Permet definir, construir i manipular una base dades amb un cert nivell d'abstracció. A més el SGBD ha de garantir la consistència i integritat de les dades i també la seguretat amb la que aquestes són emmagatzemades. Així com també la concurrència de diversos clients en un mateix moment donat.

HTML5: HyperText Markup Language 5, és un llenguatge de programació utilitzat àmpliament en l'elaboració de pàgines web. S'executa en la màquina del client i no té accés als recursos del servidor.

PHP: PHP Hypertext Pre-processor, és un llenguatge de programació que s'utilitza en l'elaboració d'aplicacions web, la seva característica principal és que s'executa en un servidor, i no en la pròpia màquina del client.

JavaScript: Llenguatge de programació interpretat i orientat a objectes. La seva característica principal és que s'executa en la pròpia màquina del client. No té accés, per tant, als recursos del servidor directament, però mitjançant altres tecnologies es poden fer peticions al servidor.

2.1.2 Descripció de la situació a tractar

El que volem desenvolupar és una xarxa social interna (ISN) on es puguin registrar alumnes i professors d'una mateixa escola. Per tant hi haurà dos tipus d'usuaris principals: alumnes i professors.

En l'aplicació, tots els usuaris han de tenir una pàgina de perfil on puguin publicar articles que han de poder ser comentats per la resta d'amics de l'usuari.

Els usuaris han de poder també pujar arxius per compartir-los a les seves pàgines de perfil.

S'han de poder crear, també, pàgines de grups de debat on els usuaris puguin publicar articles i arxius en comunitat, i on el creador del grup ha de tindre un control total d'administració sobre aquest.

Pel que respecta a la vessant més educativa, els usuaris professors han de poder crear i administrar noves assignatures. A la pàgina d'una assignatura, el professor ha de poder publicar noves notícies i material educatiu, entre d'altres. A més, les assignatures han de tenir també una wiki pròpia de l'assignatura i un fòrum on els alumnes matriculats puguin publicar i comentar articles amb resta d'usuaris, inclòs el professor.

L'aplicació ha de ser senzilla d'utilitzar i amb una interfície de qualitat que generi una bona experiència als usuaris, pretenent així convertir-se en una eina útil i d'ús freqüent.

2.1.3 Situació actual i context

Com hem analitzat anteriorment, existeixen al mercat diverses solucions amb característiques semblants a les que es volen assolir amb aquest projecte. No obstant, es considera a aquest projecte únic en els seus aspectes ja que no existeix cap altre destinat tant a ús lúdic com educatiu i que inclogui totes les funcionalitats que es volen implementar en aquest.

En el context tecnològic actual hem vist com en els últims anys, amb l'aparició del Web 2.0, un nou concepte que permetia una major interacció entre els usuaris a Internet, han aparegut les primeres xarxes socials, on aquests usuaris han trobat un punt en comú on poder compartir qualsevol cosa i interactuar amb la resta del món.

Per altra banda, també s'ha pogut veure com s'ha incrementat l'ús de les noves tecnologies per part dels estudiants i l'ús amb el que aquests les fan servir com a complement pels seus estudis. La pàgina www.wikipedia.com, una enciclopèdia lliure a Internet, n'és un exemple clar, ja que s'ha convertit en una eina de consulta freqüent per molts estudiants. Amb aquest increment no és estrany que moltes escoles i institucions acadèmiques hagin optat per implantar sistemes com *moodle* als seus centres. A l'actualitat, però, es troba a faltar alguna alternativa que a més de permetre als alumnes i als professors compartir material educatiu, com és el cas de *moodle*, tingui una vessant més social i que permeti a alumnes i professors compartir altres coses no relacionades necessàriament amb l'educació. Al voltant d'aquesta necessitat és on es pretén desenvolupar aquest projecte.

Pel que fa als requisits tècnics necessaris per executar l'aplicació web a desenvolupar no es requeriran requisits per a res elevats. I per tant, es considera que qualsevol ordinador actual, ja sigui portàtil, *netbook* i fins i tot alguns telèfons mòbils, haurien de poder executar sense problemes l'aplicació, amb l'únic requisit de disposar d'un navegador web amb suport per HTML5 i JavaScript, i d'una connexió a Internet.

2.1.4 Parts interessades (*stakeholders*)

Les parts interessades o involucrades (*stakeholders*) són totes aquelles persones que estan interessades en la realització del projecte o que formen part de la seva solució o implementació. A continuació mostrem una taula que ens indica les parts interessades i les responsabilitats que assoleixen:

Nom	Perfil	Responsabilitat
Director del projecte	Director del projecte	Supervisa la feina de l'alumne, fa el seu seguiment de desenvolupament i avalua el projecte i la seva evolució.
Programador	Analista-programador	S'encarrega tant de documentar l'anàlisi de requeriments i del disseny de l'aplicació com de la seva implementació final.
Administrador del sistema	Administrador del sistema	La seva tasca serà la d'administrar el sistema una vegada hagi estat instal·lada la versió final al servidor del client. Ha de rebre la documentació adient per tal d'assolir els coneixements necessaris per poder realitzar aquesta tasca.
Client	Client	Fa l'aprovació i el seguiment puntual del projecte. Participa en la tasca d'anàlisi de requeriments.

Taula 1: Parts interessades.

2.1.5 Objectius generals

- Que l'aplicació funcioni correctament en els navegadors web més utilitzats.
- Que l'aplicació funcioni sense cap ralentització que pugui afectar al seu rendiment.
- Consolidar-se com una eina útil tan per alumnes com per professors i per les escoles que l'utilitzin.
- Millorar la relació de l'alumnat amb el seu centre educatiu a través d'aquesta aplicació
- Consolidar-se com una eina que també ajudi a l'alumne al seu aprenentatge.
- Que la interfície tingui un bon disseny, usabilitat i on tots els elements siguin fàcilment accessibles.

- Generar una bona experiència d'ús als usuaris que l'utilitzin.
- Que l'aplicació tingui pocs o cap error de funcionament, ja que això també repercutirà en l'experiència final de l'usuari.

2.2 Descripció del sistema

A continuació es fa una descripció general sobre el sistema i es fa una primera anàlisi de requeriments per poder determinar els costos que pot tenir assolir-los.

2.2.1 Descripció física

Per a poder executar l'aplicació tan sols necessitem un servidor web amb PHP i un Sistema de Gestió de Bases de Dades (SGBD). Quan parlem de servidor web i de SGBD estem parlant de components de software, i per tant, són components que poden funcionar instal·lats en una mateixa màquina (hardware) juntament amb la base de dades en si mateix. Per altra banda, els usuaris del sistema (persones) només necessiten un ordinador amb connexió a Internet, i un navegador web amb suport per HTML5 i JavaScript.

Els clients (navegadors web) s'encarregarien de fer les peticions dels usuaris al servidor i de processar les respostes d'aquest per generar una resposta visual que és la que veu l'usuari finalment. A continuació veiem un diagrama que mostra el funcionament del sistema que acabem de descriure:

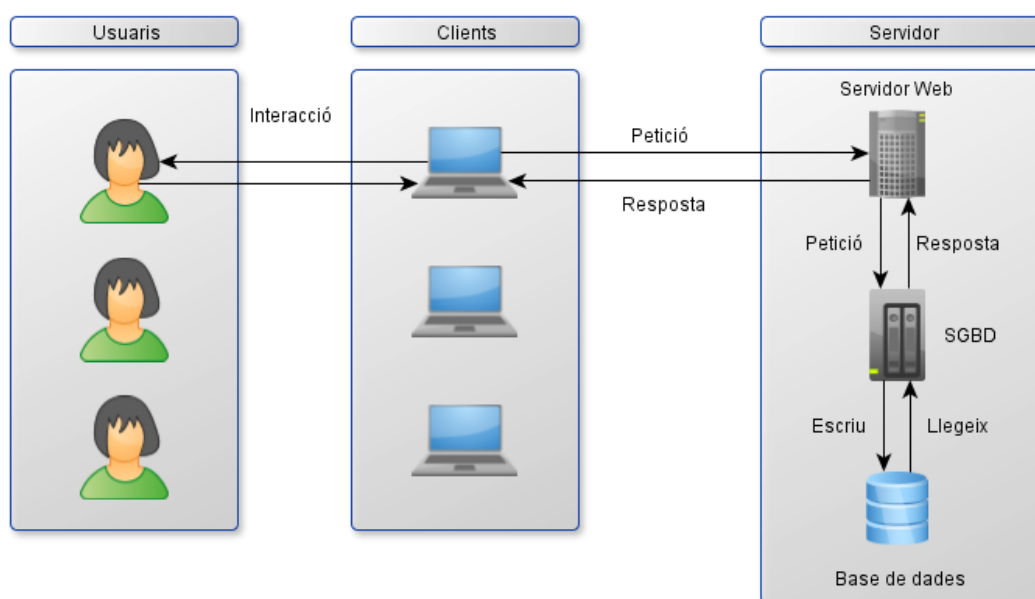


Figura 1: Descripció física del sistema

2.2.2 Perfils d'usuari del sistema

Quan ens referim a usuaris del sistema, ens referim a usuaris que utilitzaran l'aplicació web. Hi haurà diversos tipus, cadascun amb diferents permisos per realitzar certes tasques. Podríem agrupar els usuaris del sistema en els següents tipus:

- **Usuaris no registrats:** Són usuaris que no estan registrats al sistema i per tant només poden accedir a la plana principal del sistema i registrar-se, si disposen d'un codi d'invitació que només podrà expedir l'administrador del sistema. No obstant, els professors no podran ser registrats mitjançant aquest mètode i hauran de ser registrats a través de l'administrador.
- **Usuaris registrats:** Poden accedir a la plana principal del sistema i accedir amb el seu usuari i contrasenya. Una vegada hagin estat validades les seves dades d'accés accediran a la seva pàgina de perfil. Poden crear grups i assignatures, i publicar anuncis al taulell d'anuncis, a més poden establir relacions d'amistat amb altres usuaris. Els usuaris registrats poden ser tant alumnes com professors i tenen unes accions específiques pel seu tipus d'usuari, per exemple els professors són els únics que poden crear i administrar noves assignatures.

Pel que respecta al tipus de relació d'amistat que podran establir dos usuaris, s'ha definit que hi haurà dos tipus d'amistat possibles. A partir d'ara, quan parlem d'amistat completa, voldrà dir que un usuari A ha fet una petició d'amistat a un usuari B, i aquest l'ha acceptada. Per tant tots dos usuaris tenen dret d'accés a les seves respectives pàgines de perfil. Quan parlem d'amistat parcial voldrà dir que un usuari A ha fet una petició a un usuari B, però que aquest encara no l'ha acceptada. Llavors B té dret d'accés a la pàgina de perfil de l'usuari A, però A no tindrà dret d'accés a la pàgina de l'usuari B fins que aquest no accepti la sol·licitud.

En tot moment, l'usuari A podrà cancel·lar la seva sol·licitud d'amistat, revocant els drets d'accés de l'usuari B a la seva pàgina de perfil.

Podem veure el seu esquema de funcionament al següent diagrama:

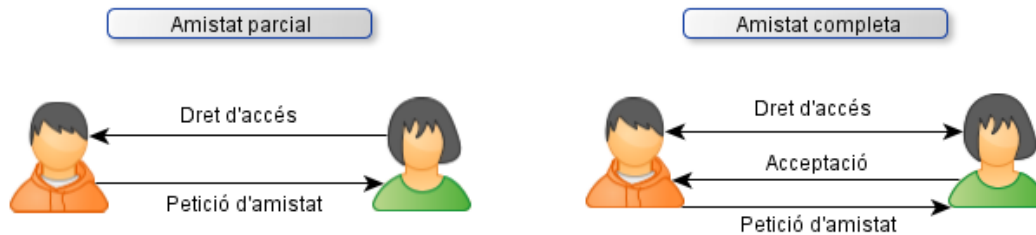


Figura 2: Tipus d'amistat

A partir d'aquí, els usuaris registrats poden agafar quatre rols més, en funció de diverses accions, que són els següents:

- **Membre de grup:**

Són usuaris registrats que són membres d'un grup i poden publicar articles i pujar arxius a aquest grup, també eliminar-los. Poden comentar articles, seus o d'altres usuaris, i fer "M'agrada"³ a articles, comentaris i arxius del grup el qual són membres.

- **Administrador de grup:**

Són usuaris registrats que han creat un grup de debat i poden realitzar les tasques d'administració d'aquest, modificar les dades del grup, administrar els seus membres i eliminar-lo. També tenen permís per eliminar qualsevol article, comentari o arxiu del grup. A part d'això poden realitzar les mateixes tasques que qualsevol altre membre del grup que han creat.

- **Matriculat a assignatura:**

Són usuaris registrats que han estat matriculats a una assignatura i en aquesta poden publicar i comentar articles al fòrum, així com també poden editar els articles de la wiki els quals el professor hagi especificat com a editables pels alumnes.

³ La funcionalitat "M'agrada" permetrà als usuaris indicar que els agrada un article, comentari, arxiu o anunci. De forma que després es podrà veure el nombre de "M'agrada" que ha tingut un element i, en alguns casos, veure quines persones han indicat que els agrada aquest. En altres parts del document es fa referència a "Votar" elements, en aquests casos es tracta igualment de la mateixa funcionalitat.

- **Professor d'assignatura:**

Són professors que han creat una assignatura i poden realitzar les tasques d'administració d'aquesta, modificar les seves dades, administrar els seus alumnes i eliminar-la. Poden publicar notícies, pujar material docent i crear o editar nous articles a la wiki, i tenen permís per eliminar qualsevol article o comentari del fòrum de l'assignatura. A part d'això poden realitzar les mateixes tasques que qualsevol altre alumne matriculat.

- **Administrador del sistema:** L'administrador és un tipus d'usuari totalment diferent que no podrà interactuar a la xarxa social, només administrar-la. Tindrà accés a una pàgina principal d'administrador introduint el seu usuari i la contrasenya d'administrador.

A aquesta pàgina ha de poder realitzar les següents tasques:

- Donar d'alta a usuaris, alumnes o **professors**.
- Modificar les dades de qualsevol usuari.
- Eliminar qualsevol usuari.
- Modificar les dades de qualsevol grup.
- Eliminar qualsevol grup.
- Modificar les dades de qualsevol assignatura.
- Eliminar qualsevol assignatura.
- Establir o trencar relacions entre usuaris, o entre usuaris i elements.
- Afegir o eliminar tipus d'arxius o de grups, ciutats o cursos.
- Eliminar qualsevol anunci del taulell d'anuncis.

2.2.3 Requisits funcionals

- Els usuaris han de poder registrar-se mitjançant un codi d'invitació que rebran al seu correu electrònic i que haurà d'expedir l'administrador.
- Els usuaris han de poder establir relacions d'amistat entre ells.
- El sistema ha d'emmagatzemar totes les relacions entre usuaris, i entre els usuaris i els diferents elements de l'aplicació.
- L'aplicació ha de comptar amb un taulell d'anuncis on els usuaris puguin publicar anuncis.

- Cada usuari ha de disposar d'una pàgina de perfil on ha de poder publicar articles i arxius. També ha de poder crear noves pàgines de grups de debat i d'assignatures (si són professors)
- Les pàgines de grups de debat han de poder ser públiques o privades, les assignatures, en canvi, seran estrictament privades.
- A les pàgines de grups de debat els usuaris membres han de poder publicar articles i arxius, en canvi, en les assignatures només el professor ha de tenir potestat per pujar nous arxius.
- El sistema ha de permetre als usuaris enviar correus electrònics a qualsevol altre usuari, mitjançant una aplicació externa.
- La interfície ha de tenir una part estàtica on l'usuari pugui accedir a qualsevol de les seccions de l'aplicació o tancar la sessió en qualsevol moment.
- La interfície ha de tenir també una part estàtica on és visualitzi el títol de l'aplicació.
- S'ha d'incloure un avís sobre quin tractament se li donarà a les dades de l'usuari al formulari de registre d'usuari, i el dret de l'usuari a l'alta, modificació i baixa d'aquestes dades, tal com estableix la Llei de protecció de dades.

2.2.4 Requisits no funcionals

- Degut a que s'emmagatzemen dades personals dels usuaris, i de caràcter sensible, a la base de dades, el sistema ha de garantir el compliment de la *Llei de protecció de Dades de Caràcter Personal* mitjançant les accions necessàries per evitar el robatori de dades per terceres persones.
- Les claus d'accés dels usuaris emmagatzemades a la base de dades hauran d'estar encriptades.
- La interfície ha d'estar disponible en català.
- La interfície ha de tenir un bon disseny, usabilitat i on tots els elements han de ser fàcilment accessibles.
- La comunicació entre el sistema i els usuaris d'aquest es farà mitjançant correu electrònic.
- La interfície ha de visualitzar-se correctament en resolucions de 1366x768 i superiors. Es considera que amb una resolució inferior, com 1024x768, no es permet una visualització clara i concisa de totes les dades que volem representar a la interfície.
- L'aplicació ha de visualitzar-se correctament en els navegadors web més utilitzats.

2.2.5 Restriccions

Degut a que el sistema s'executarà en un servidor físic i amb una connexió a Internet limitada, tenim unes certes restriccions a nivell de hardware i de software que detallarem a continuació:

- L'aplicació s'executarà sobre un servidor amb unes capacitats limitades a nivell de hardware i unes capacitats limitades a nivell de software. Per tant, el nombre d'usuaris connectats alhora al servidor web en un moment donat estarà restringit tant a la capacitat del hardware (servidor físic) com al nombre màxim d'usuaris que pot gestionar el software (servidor web i SGBD). Es considera que un servidor (físic) de gamma mitja podrà complir amb les necessitats normals dels usuaris d'un centre d'estudis.
- El SGBD té una capacitat limitada de peticions concurrents en un moment donat, per tant, el nombre de peticions estarà limitat al nombre de peticions que pugui gestionar el SGBD. També es considera que un servidor (físic) de gamma mitja podrà complir amb les necessitats normals dels usuaris d'un centre d'estudis.
- Pel que fa a l'espai d'emmagatzemament del servidor per emmagatzemar les dades de l'aplicació, tant pròpies de la interfície com fotografies i arxius dels usuaris, estarà limitat al que disposi el servidor per aquests afers.
- El sistema operatiu del servidor ha de ser Microsoft® Windows o GNU Linux, que són els dos que estaran instal·lats al servidor. Per tant, tot el software necessari per executar l'aplicació ha de poder-se executar tot en un, o tot en altre.

2.2.6 Normatives i legislació.

A continuació s'especifiquen les normatives i les legislacions que s'han de tenir en compte i que ha d'acomplir el projecte:

- ***Ley de protección de Datos de Carácter Personal (Llei de protecció de dades de caràcter personal):***

Per a complir aquesta normativa hem de garantir que les dades seran emmagatzemades en una base de dades, on només nosaltres tindrem accés, i que aquesta tindrà la seguretat suficient per evitar accessos no autoritzats que provoquin un robatori o alteració d'aquestes dades.

<http://civil.udg.es/normacivil/estatal/persona/pf/lo15-99.htm> (Juny de 2012)

- **Normativa de projectes d'enginyeria tècnica:**

Aquesta normativa defineix el format que ha de tenir la memòria del projecte i d'altres aspectes de la presentació d'aquest.

http://www.uab.cat/Document/541/595/Normativa_PFCNovembre2010.pdf (Juny de 2012).

2.3 Objectius

Els objectius són les metes concretes que volem assolir durant la realització del projecte. Els podem dividir en objectius crítics, que són els que tenen una màxima prioritats i han de complir-se expressament ja que afecten al funcionament global de tot el projecte. Els objectius prioritaris, amb menys prioritats que els crítics però que, en cas de no assolir-se, poden afectar críticament a altres parts del projecte. I per últim, els objectius secundaris, que són objectius que estaria bé assolir (*nice to have*), i que són els que menys prioritats tenen i que en el cas de no assolir-se, no afectarien críticament al funcionament global del projecte.

Els objectius del projecte han de seguir la regla SMART (*intel·ligent*), paraula anglesa acrònim de *Specific* (específic), *Measurable* (mesurable), *Achievable* (realitzable), *Realistic* (realista) i *Time-Bound* (limitat en temps). És a dir, objectius concrets, que puguin ser mesurables, realistes i realitzables, i limitats en el temps, és a dir, que es poden realitzar en un temps limitat i raonable.

Per al desenvolupament d'aquesta aplicació s'han marcat els següents objectius:

2.3.1 Crítics

1. Registre de nous usuaris.
2. Creació de nous grups i assignatures.
3. Implementar les pàgines d'usuari, grup i assignatura.
4. Que els usuaris puguin publicar articles a les pàgines d'usuari, grups i als fòrums de les assignatures.
5. Que els usuaris puguin pujar arxius a pàgines d'usuari, grups o assignatures.
6. Que els administradors de grups puguin administrar el seus grups.
7. Que els professors puguin administrar les seves assignatures.
8. Que els usuaris puguin administrar el seu perfil.

2.3.2 Prioritaris

9. Implementar la funcionalitat "M'agrada" a articles, comentaris i arxius.
10. Implementar un sistema d'amistats entre usuaris.
11. Implementar un taulell d'anuncis per a tots els usuaris.
12. Que els usuaris tinguin un cercador on cercar usuaris, grups o assignatures.
13. Que l'administrador del sistema tingui una interfície on tingui accés a l'administració de tot el sistema.

2.3.3 Secundaris

14. Que els usuaris puguin comunicar-se a un xat públic.
15. Que els usuaris puguin veure el nombre de descàrregues per als arxius pujats.
16. Implementar un sistema de notificació de noves notícies o material docent a les assignatures mitjançant correu electrònic.

Objectius	Crítics	Prioritaris	Secundaris
O1	•		
O2	•		
O3	•		
O4	•		
O5	•		
O6	•		
O7	•		
O8	•		
O9		•	
O10		•	
O11		•	
O12		•	
O13		•	
O14			•
O15			•
O16			•

Taula 2: Objectius crítics, prioritaris i secundaris del desenvolupament.

2.4 Recursos

Per a la realització del projecte i la seva duta a terme disposem dels següents recursos de hardware, software i recursos humans:

2.4.1 Hardware

Es disposa d'un ordinador de sobretaula, destinat a fer la funcionalitat de servidor físic, tot i que també pot formar part activa tant en el desenvolupament com en la documentació del projecte. I també d'un ordinador portàtil destinat a ser utilitzat com a eina per provar l'aplicació des d'un entorn extern al servidor i que també pot formar part activa en la implementació i documentació. A més disposem d'una connexió a Internet i d'un mòdem-router per gestionar aquesta.

A continuació detallem les característiques d'aquests recursos:

- **Ordinador de sobretaula amb les següents característiques:**

- Processador AMD Sempron 3400+ 2.2 Ghz
- 2 GB de memòria RAM
- 750 GB de disc dur SATA (500 GB + 250 GB)
- Monitor LCD de 19" amb resolució 1440x900.
- Targeta de xarxa LAN 10/100
- Teclat QWERTY.
- Ratolí làser.
- 8 ports USB 2.0
- Lector/Gravador de DVD±RW

- **Ordinador portàtil amb les següents característiques:**

- Model ASUS Eee PC 1201N
- Processador Intel Atom 330, 1,6 GHz, 2 nuclis
- 3 GB de memòria RAM
- Pantalla LED de 12,1" amb resolució 1366x768
- Targeta de xarxa LAN 10/100
- Targeta de xarxa Wi-Fi
- 3 ports USB 2.0

- Connexió a Internet amb un ample de banda de 20 Mbps (Megabits per segon) de baixada i 1 Mbps (Megabit per segon) de pujada.
- Mòdem-router model Comtrend AR-5381u.
- A més la Universitat Autònoma de Barcelona (UAB) ens proporciona recursos tant de hardware com de servidors, detallats als següents enllaços:
<http://www.uab.cat/servlet/Satellite/els-estudis/projecte-de-fi-de-carrera-enginyeries-tecniques-en-informatica/recursos/servidors-1183702331173.html>
<http://www.uab.cat/servlet/Satellite/els-estudis/projecte-de-fi-de-carrera-enginyeries-tecniques-en-informatica/recursos/laboratori-de-hardware-1183702337252.html>

2.4.2 Software

Disposem dels següents recursos de software:

- **Sistemes operatius:** Microsoft® Windows XP, Microsoft® Windows 7 i Ubuntu GNU/Linux 11.10.
- **Servidor web:** XAMPP (Apache 2.2.21 + PHP 5.3.8)
- **SGBD:** MariaDB + OQGraph.
- **Disseny gràfic:** Adobe Fireworks CS5 i Adobe Photoshop CS5.
- **Documentació i presentació:** Suite Microsoft® Office 2010 (Microsoft® Word 2010, Microsoft® PowerPoint 2010 i Microsoft® Excel 2010) i Microsoft® Project 2007.
- **Disseny de diagrames:** Microsoft® Visio 2010, Dia i yWorks yEd
- **Entorns de programació:** PhpDesigner8 i Notepad++.
- **Testeig i proves:** Eines per desenvolupadors i consola JavaScript de Google Chrome i Firefox.

2.4.3 Humans

- **Analista-programador:** realitzarà les tasques d'anàlisi, disseny, implementació i proves, i de fer la documentació de tot el projecte, i també l'estudi de viabilitat del projecte. En aquest cas, l'alumne que realitza el projecte serà el que farà el rol d'analista-programador, i també de client.

- **Director de projecte:** supervisarà la feina de l'analista-programador, controlarà l'evolució del projecte i aprovarà les fases d'estudi de viabilitat, anàlisi, disseny, implementació i proves.

2.5 Anàlisi de costos

2.5.1 Cost del material

A la següent taula es desglossa el cost de la compra o lloguer dels recursos materials necessaris per a la realització d'aquest projecte:

Recurs	Cost d'adquisició	Cost de lloguer
Servidor web dedicat	-	50€/mes
Domini .cat	-	21,99€/any
Ordinador portàtil	0€ (Disponible)	-
Ordinador de sobretaula	0€ (Disponible)	-
phpDesigner 8 (Llicència per a ús comercial)	72,33€	-
Microsoft® Office 2010	249€ (0€ llicència universitat)	-
Microsoft® Project 2007	775€ (0€ llicència universitat)	-
Microsoft® Visio 2010	330€ (0€ llicència universitat)	-
Adobe Fireworks CS5	410€	-
Adobe Photoshop CS5	982€	-
Microsoft® Windows XP	0€ (Disponible)	-
Microsoft® Windows 7	0€ (Disponible)	-
Ubuntu GNU/Linux 11.10	0€ (Gratuït)	-
XAMPP (Apache)	0€ (Gratuït)	-
MariaDB	0€ (Gratuït)	-
Dia i yWorks yEd	0€ (Gratuït)	-
Notepad++	0€ (Gratuït)	-
Google Chrome i Firefox	0€ (Gratuït)	-

Taula 3: Cost del material.

2.5.2 Cost del personal

El cost del personal varia en funció de les hores de treball i de l'especialització d'aquest. A la següent taula es pot veure tot aquest el cost desglossat:

Recurs	Cost / Hora	Hores de treball	Cost total
Analista-Programador	30 €/h	241,8 h	7.254,00 €
Director de projecte	50 €/h	28 h	1400,00 €
Total personal			8.654 €

Taula 4: Cost del personal

2.5.3 Pressupost

Per fer el pressupost tindrem en compte la suma total dels costos de personal i a més el cost d'amortització del material adquirit. Ja que hi ha eines, de hardware i de software, que poden ser reutilitzades per al desenvolupament d'altres projectes, amb el consegüent estalvi. Per tant això ens dona un cost d'amortització que dependrà de l'ús que li hem donat al material en la realització d'aquest projecte en concret.

Recurs	Cost amortització	Cost unitari	Període amortització	Període utilització
phpDesigner8	9,04€	72,33€	24 mesos	3 mesos
Adobe Fireworks CS5	34,16€	410€	36 mesos	3 mesos
Adobe Photoshop CS5	81,83€	982€	36 mesos	3 mesos
Total d'amortització	125,03€			

Taula 5: Cost d'amortització.

Pressupost detallat	
Cost de desenvolupament del projecte	8.654,00€
1. Estudi de viabilitat (9h).....	316,00€
2. Anàlisi de requeriments (24h).	743,00€
3. Disseny de l'aplicació (26h)... ..	803,00€
4. Implementació de l'aplicació (158h).....	4.832,00€
5. Proves (12h).....	360,00€
6. Recopilació i redacció de documentació (20h).....	1600,00€
Cost d'amortització del material	125,03€
Total:	8.779,03€
<ul style="list-style-type: none"> • Atès al cost de desenvolupament de l'aplicació, es pretén vendre llicències a escoles o institucions educatives a un preu de 999€. Amb la venda de 9 llicències d'ús tindriem amortitzat el cost total del projecte. • La instal·lació de l'aplicació i el seu manteniment al client durant 3 anys seran oferts de manera gratuïta. • A més dels costos de la llicència i el manteniment, el sistema tindrà uns costos fixos mensuals de 51,83€/mes, corresponents al lloguer del servidor dedicat i al domini .cat. 	

2.6 Alternatives

Una vegada fet el pressupost, analitzarem altres alternatives que tenim a l'abast i que ens permetin assolir els mateixos objectius marcats per aquest projecte. Les compararem amb l'alternativa que ja tenim pressupostada, tot fent un càlcul del cost aproximat que tindria el projecte si agaféssim una altra alternativa com a solució per a desenvolupar-lo i veient si aquestes alternatives resulten més idònies o encertades.

Per tant, s'ha realitzat una tasca d'investigació i s'han trobat les següents alternatives:

- **Alternativa 1:**



Atès a que la plataforma *moodle* és de llicència GNU GPL, és té accés al codi font i per tant es podria modificar la plataforma per tal d'acomplir els objectius que es volen assolir amb aquest projecte. A continuació tenim una taula d'avantatges i inconvenients d'aquesta alternativa:

- **Avantatges:**

- Porta anys en desenvolupament i té darrera a una comunitat de programadors que corregeixen errors i implementen noves funcions.
- Al portar anys de desenvolupament amb un equip de moltes persones darrera, té una qualitat que difícilment podrem assolir amb els recursos humans dels que disposem per aquest projecte.
- Hi ha moltes funcions de les que volem assolir amb els objectius d'aquest projecte que ja estan implementades.
- Té moltes altres funcions addicionals de les que especifiquem a aquest projecte

- **Inconvenients:**

- Per poder arribar a l'etapa de modificació del codi font hem d'adquirir uns certs coneixements que ens permetin entendre com està programada la plataforma i poder modificar-la. (més temps)
- Hem d'implementar moltes funcions necessàries per acomplir els objectius d'aquest projecte i que no estan implementades de sèrie.
- La llicència GPL ens obliga a compartir el codi font de la nostra aplicació amb la qual cosa no podem garantir l'exclusivitat del software que estem desenvolupant, ja que el codi serà públic.

- **Cost:**

- **Adquisició del codi font de *moddle*:**

Gratuït (Amb llicència GNU GPL que ens permet el seu ús comercial)

- **Adquisició dels recursos materials per porta a terme el projecte:**

1.463,33€ (Els mateixos que a la solució proposada al pressupost)

- **Cost de desenvolupament del projecte:**

11.729,13€

- **Amortització dels recursos materials:**

366,08€

- **Cost total:** 11.729,13 + 366,08 = **12.095,21€**

- **Alternativa 2:**



Existeix una altra alternativa, anomenada “elgg” que és de codi lliure i té una llicència GPL 2, i per tant també es té accés al codi font. A més té darrera una comunitat que li dóna suport en forma de plugins⁴ que implementen noves funcionalitats. A continuació tenim una taula d'avantatges i inconvenients d'aquesta alternativa:

- **Avantatges:**

- També porta anys en desenvolupament i té darrera a una comunitat de programadors que corregeixen errors i implementen noves funcions.
- La comunitat aporta plugins que implementen noves funcions
- La plataforma està orientada principalment a xarxes socials.

- **Inconvenients:**

- Per poder arribar a l'etapa de modificació del codi font hem d'adquirir uns certs coneixements que ens permetin entendre com està programada la plataforma i poder modificar-la. (més temps)
- Hem d'implementar moltes funcions necessàries per acomplir els objectius d'aquest projecte i que no estan implementades de sèrie, ja sigui amb la implementació en forma de plugins externs o directament al codi font.
- La llicència GPL ens obliga a compartir el codi font de la nostra aplicació, amb el qual no podem garantir l'exclusivitat del software que estem desenvolupant, ja que el codi font serà públic.

- **Cost:**

- **Adquisició del codi font:**

Gratuït (Amb llicència GPL 2 que ens permet el seu ús comercial)

- **Adquisició dels recursos materials per porta a terme el projecte:**

1.463,33€ (Els mateixos que a la solució proposada al pressupost)

⁴ Un plugin (de l'anglès “endollable”) és un complement per a una aplicació informàtica que aporta una nova funcionalitat i generalment molt específica.

- **Cost de desenvolupament del projecte:**
11.729,13€
- **Amortització dels recursos materials:**
366,08€
- **Cost total:** 11.729,13 + 366,08 = **12.095,21€**

- **Alternativa 3:**



Aquesta és l'alternativa que ja tenim analitzada i pressupostada, que seria programar des de zero l'aplicació utilitzant les tecnologies PHP, HTML5 i JavaScript, entre d'altres. A continuació tenim una taula d'avantatges i inconvenients d'aquesta alternativa en comparació amb la resta:

- **Avantatges:**
 - Podem implementar la solució per aconseguir exclusivament els objectius del projecte.
 - Tenir la certesa que estem desenvolupant una cosa nova i que tenim l'exclusivitat, ja que és un treball propi al que ningú té accés.
 - Només hem d'aprendre l'ús d'algunes llibreries⁵ i llenguatges de programació, i per tant podem començar a programar més aviat (menys temps).
- **Inconvenients:**
 - Hem d'implementar totes les funcionalitats des de zero, el que ens portarà més temps d'anàlisi de requeriments i de disseny.
 - Hem d'implementar moltes funcions necessàries per aconseguir els objectius d'aquest projecte, que poden presentar més dificultats.
- **Cost:**
 - **Adquisició dels recursos materials per porta a terme el projecte:**
1.463,33€
 - **Cost de desenvolupament del projecte:**
8.654,00€
 - **Amortització dels recursos materials:**
125,03€

⁵ Una llibreria és un conjunt de subprogrames ja implementats que proporcionen serveis o funcions per al desenvolupament d'altres programes i que passen a formar part d'aquests.

- **Cost total:** 8.654,00€ + 125,03 = **8.779,03€**

2.7 Solució proposada

Per tant, d'entre les tres alternatives analitzades s'escull l'**Alternativa 3**, ja que es considera la més viable i la més idònia per aconseguir els objectius del projecte, i també la menys costosa a nivell de personal i de temps de desenvolupament. Les altres alternatives, si ben bé podrien oferir una possible qualitat superior, afronten massa riscos com per considerar-les viables per aquest projecte.

A l'**Alternativa 1** o l'**Alternativa 2**, el fet d'haver d'adquirir uns certs coneixements més complexos per desenvolupar el projecte implica uns riscos massa alts, ja que en el cas d'un endarreriment en l'aprenentatge d'aquests coneixements afectaria dràsticament a tot el projecte. Per tant resulta més viable i idoni començar el projecte des de zero que agafar un altre i haver de modificar-lo.

Una vegada decidit amb quina solució es portarà a terme el projecte, s'ha d'escollir amb quina metodologia de desenvolupament el realitzarem i hem de fer un calendari temporal de desenvolupament complint amb el pressupost del projecte.

2.8 Metodologia i WBS

2.8.1 Metodologia

L'elecció de la metodologia de desenvolupament és important ja que una mala elecció pot portar a un temps més elevat de desenvolupament, i el fet de no poder arribar a aconseguir alguns objectius. Per aquest projecte s'ha optat per utilitzar una metodologia incremental iterativa, amb una fase iterativa d'Anàlisi-Disseny-Implementació-Proves, tal i com es mostra al diagrama següent:

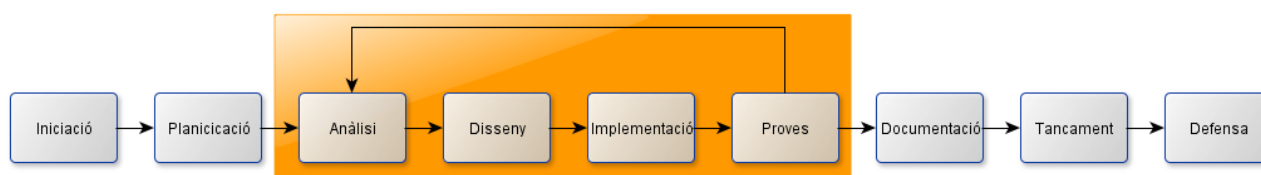


Diagrama 1: Metodologia de desenvolupament

L'elecció d'aquesta metodologia presenta alguns avantatges i inconvenients. Entre els avantatges destaquem el fet que no hem d'esperar a la fase final de la implementació, per veure i mostrar els resultats intermitjos d'aquesta al director del projecte o al client. I de fet, a cada iteració el director del projecte haurà de donar la fase per conclusa. Entre els inconvenients destaquem que poden aparèixer nous requeriments que ens obliguin a modificar part del codi que ja teníem implementat i que no havíem previst, amb el consegüent augment de temps.

2.8.2 Fases i activitats

A la següent taula podem veure una breu descripció de les tasques a realitzar en cadascuna de les fases del projecte:

Fases	Descripció
Iniciació	Fase d'iniciació. Inclou la definició del projecte, assignació matriculació.
Planificació	Inclou l'Estudi de Viabilitat i el Pla del Projecte.
Anàlisi	Anàlisi de requeriments funcionals i no funcionals.

Taula 6: Taula de fases i activitats.

Fases	Descripció
Disseny	Inclou el disseny tant de la base de dades, l'estructura de classes i la interfície.
Desenvolupament	Fase de desenvolupament de les diferents parts de l'aplicació web.
Proves	Fase de proves sobre el sistema en diferents graus.
Documentació	Fase de documentació del projecte.
Tancament del projecte	Fase de tancament. El director del projecte signa l'acceptació i tancament del projecte.
Defensa	Defensa del projecte davant la comissió avaluadora.

Taula 6: Taula de fases i activitats.

2.8.3 Diagrama WBS

WBS (*Work Breakdown Structure*) o **Estructura de Descomposició del Treball** és una descomposició jeràrquica de la gestió del projecte. La seva finalitat és organitzar i definir l'abast del projecte, segons s'ha declarat a la documentació vigent.

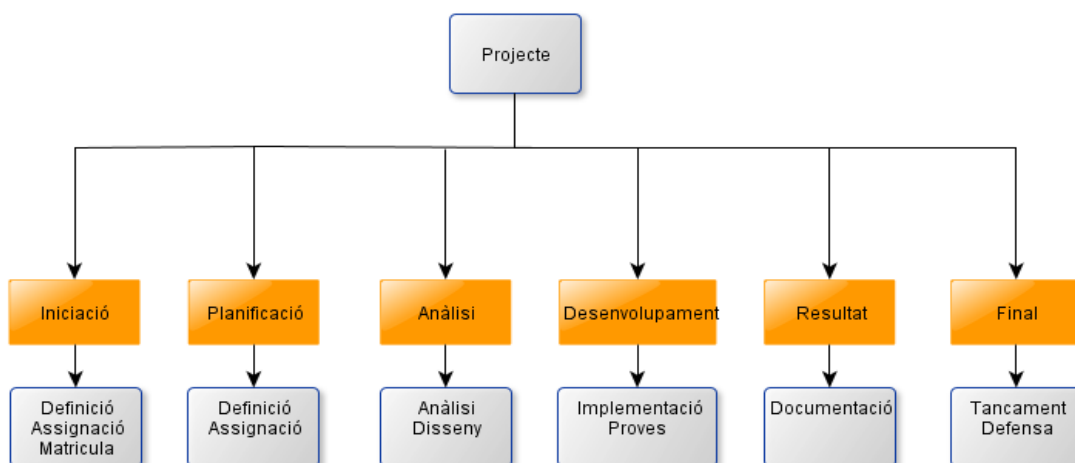


Figura 3: Diagrama WBS

2.9 Calendari i planificació

2.9.1 Taula de tasques

	Nombre de tarea	Duración	Nombres de los recursos
1	<input type="checkbox"/> Estudi de viabilitat	1,13 días	
2	Estudi de viabilitat	4 horas	Analista programador
3	Aprovació Estudi de viabilitat	1 hora	Analista programador[10%];Director del projecte
4	Pla de projecte	3 horas	Analista-programador
5	Aprovació de pla de projecte	1 hora	Director del projecte;Analista programador[10%]
6	<input type="checkbox"/> Analisi de requeriments	3 días	
7	Analisi de requisits funcionals i no funcionals	12 horas	Analista-programador
8	Analisi de dades	5 horas	Analista programador
9	Analisi de segurat i legalitat	1 hora	Analista programador
10	Documentació	5 horas	Analista programador
11	Aprovació de l'analisi	1 hora	Director del projecte;Analista programador[10%]
12	<input type="checkbox"/> Disseny de l'aplicació	3,25 días	
13	Disseny de la base de dades	5 horas	Analista-programador
14	Disseny dels diferents mòduls o pàgines	15 horas	Analista-programador
15	Disseny de la interfície	3 horas	Analista programador
16	Documentació	2 horas	Analista-programador
17	Aprovació del disseny	1 hora	Director del projecte;Analista programador[10%]
18	<input type="checkbox"/> Implementació de l'aplicació	19,88 días	
19	Instalació del SGBD	2 horas	Analista-programador
20	Implementació de la BD	10 horas	Analista programador
21	Instalació del servidor (XAMPP)	2 horas	Analista-programador
22	Desenvolupament dels mòduls de connexió amb la BD	60 horas	Analista-programador
23	Desenvolupament de la interfície d'usuari	80 horas	Analista-programador
24	Aprovació de la implementació	4 horas	Analista programador[10%];Director del projecte
25	<input type="checkbox"/> Proves	1,5 días	
26	Unitat	3 horas	Analista-programador
27	Integració	3 horas	Analista-programador
28	Regressió	3 horas	Analista-programador
29	Accepció	3 horas	Analista-programador
30	Documentació del projecte	20 horas	Director del projecte;Analista programador

Taula 7: Taula de tasques per al desenvolupament del projecte.

2.9.2 Diagrama de Gantt

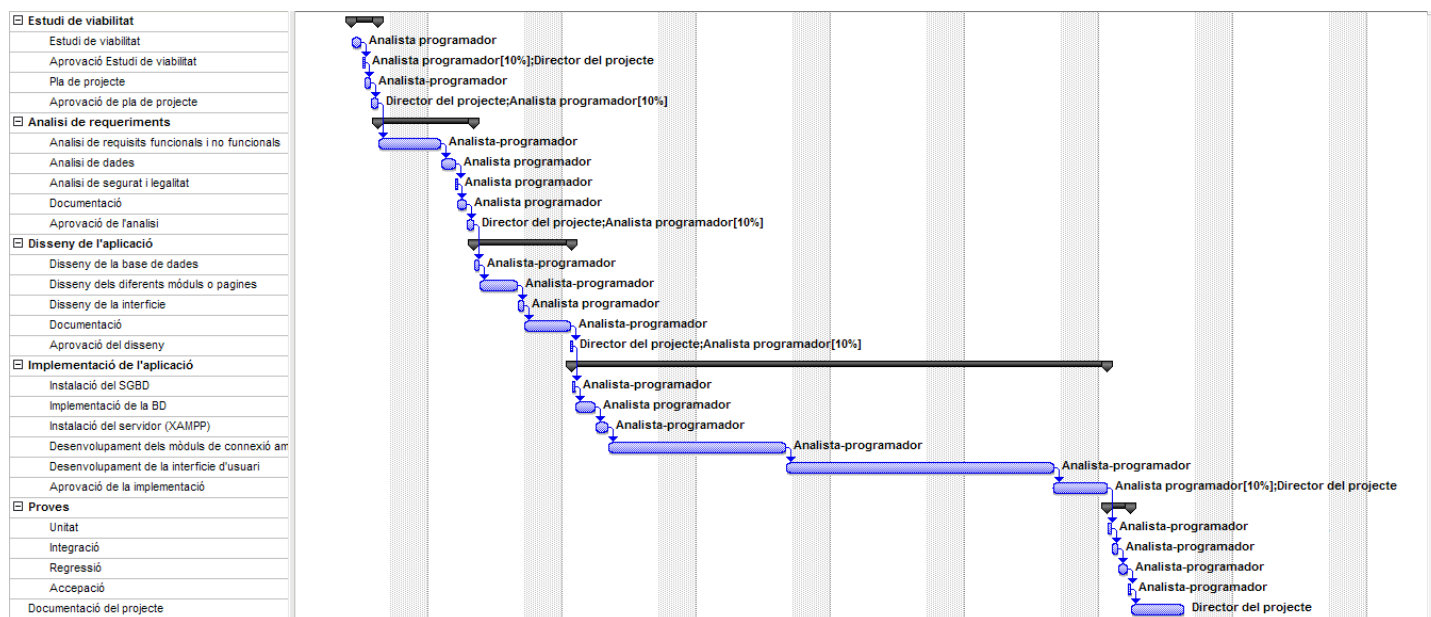


Figura 4: Diagrama de Gantt

2.10 Avaluació de riscos

2.10.1 Llista de riscos

R1. Planificació temporal optimista:

Afecta al pla de projecte. No s'acaba en la data prevista, augmenta l'ús dels recursos.

R2. Manca alguna tasca necessària:

Afecta al pla de projecte. Provocaria l'incompliment d'algun o d'alguns dels objectius del projecte.

R3. Falta de temps i/o recursos:

Afecta al pla de projecte. Provocaria un endarreriment en la finalització del projecte, afectaria al compliment d'algun o tots els objectius del projecte.

R4. Eines de desenvolupament inadequades:

Afecta al desenvolupament del projecte. Provocaria un endarreriment en la finalització del projecte, una possible menor qualitat final i més errors en l'aplicació.

R5. Dificultat per aconseguir alguns dels requeriments:

Afecta a l'estudi de viabilitat, l'anàlisi, les proves o la formació. Provocaria un endarreriment del projecte, s'hauria de refer l'anàlisi de requeriments.

R6. Incompliment d'alguna norma, reglament o legislació:

En qualsevol fase. No es compleixen els objectius, repercussions legals.

R7. Abandonament del projecte abans de la finalització:

En qualsevol fase. Pèrdues econòmiques i frustració.

2.10.2 Catalogació

	Probabilitat	Impacte
R1	Alta	Crític
R2	Alta	Crític
R3	Alta	Crític
R4	Baixa	Crític
R5	Alta	Crític
R6	Baixa	Crític
R7	Baixa	Catastròfic

Taula 8: Catalogació de riscos

2.10.3 Pla de contingència

	Solució
R1	Ajornar alguna funcionalitat, afrontar possibles pèrdues.
R2	Revisar el pla de projecte, modificar la planificació.
R3	No aconseguir algun dels objectius, o afrontar possibles pèrdues.
R4	Millorar la formació, utilitzar eines alternatives, i a ser possible gratuïtes, per reduir el cost.
R5	Intentar aconseguir els requeriments més importants i fer requeriments realistes.
R6	Revisar les normes i legislació, consultar un expert, afrontar possibles repercussions penals.
R7	No hi ha cap solució

Taula 9: Pla de contingència

2.11 Conclusions

Amb l'estudi de viabilitat s'ha determinat si el projecte resulta o no realitzable i viable, econòmicament parlant en el context de mercat actual. S'han analitzat els costos totals que tindria el projecte i fet un pressupost final. S'han analitzat també les alternatives que tenim a l'abast i s'ha escollit la més viable. S'han tingut en compte també els possibles riscos i les incidències que podria presentar el projecte i quin impacte tindria sobre el calendari final, a més, d'un pla de contingència per afrontar aquests.

En general podem destacar que el projecte presenta els següents avantatges:

- El projecte resulta viable econòmicament.
- Els riscos que presenta no són massa elevats i aquests presenten solucions.
- S'ha escollit d'entre les alternatives, la més viable i idònia.
- S'ha escollit la metodologia incremental iterativa per a desenvolupar el projecte.

El director del projecte i el client, en aquest cas l'alumne, determinen que el projecte resulta viable a data 15-10-2011, per tant la fase d'estudi de viabilitat queda per concloua.

3 Anàlisi de requeriments

3.1 Introducció

L'anàlisi de requeriments és una etapa important en l'enginyeria del software, ja que alhora de desenvolupar un projecte ens serveix per especificar que és el que es vol que faci exactament el sistema. Per una banda s'han d'analitzar els requisits funcionals, que són els que especifiquen les funcionalitats del sistema. I per altra, s'han d'analitzar també els requeriments no funcionals, que són els que especifiquen com ha de funcionar el sistema, sense entrar en el detall de com ho fa.

3.2 Requeriments funcionals

3.2.1 Inici

RF1: A la pàgina inicial, l'usuari es podrà registrar o accedir amb el seu compte.

RF2: Per poder registrar-se l'usuari tindrà abans d'introduir un codi d'invitació vàlid.

RF3: Caldrà validar el formulari de registre de l'usuari per comprovar si ha introduït tots els camps obligatoris i si les dades introduïdes són correctes.

3.2.2 Pàgines d'usuari

RF4: L'usuari propietari podrà publicar articles (amb o sense imatge associada) i comentar-los, pujar arxius i administrar el seu perfil.

RF5: Si l'usuari que accedeix és amic de l'usuari propietari podrà veure tota la informació de l'usuari. També podrà veure els seus grups i amics en comú, veure els seus articles o arxius, i fer comentaris o "M'agrada" als seus articles o arxius, i enviar-li un correu electrònic.

RF6: Si l'usuari que accedeix no és amic no podrà veure res i només ha de poder demanar una sol·licitud d'amistat a aquest.

RF7: En administració de perfil l'usuari propietari ha de poder modificar les seves dades, com el nom o el curs, i també ha de poder donar-se de baixa.

RF8: Tant al publicar un article, comentari o arxiu s'hauran de validar si les dades necessàries són correctes abans de publicar-les.

3.2.3 Pàgines de grup

RF9: A les pàgines de grup els usuaris membres podran publicar articles (amb o sense imatge associada) i arxius, així com també fer comentaris i "M'agrada" als articles dels altres usuaris, o els seus.

RF10: Els grups podran ser grups públics o no i en el cas de ser públics qualsevol usuari podrà visualitzar els seus continguts, encara que no podrà publicar res. En tot cas, els usuaris no membres podran enviar una petició de pertinença a l'administrador del grup. En els grups no públics només podran accedir els membres.

RF11: L'usuari administrador tindrà accés a les opcions d'administració del grup i permís per eliminar qualsevol article, arxiu o comentari del grup. A part d'això podrà realitzar les mateixes tasques que qualsevol altre membre.

RF12: A les opcions d'administració del grup l'usuari administrador podrà modificar les dades del grup, fer la gestió d'usuaris i també podrà eliminar completament el grup.

RF13: Al expulsar a un usuari del grup es podrà fer de dues formes, una que deixi constància de totes les seves publicacions, i una altra que elimini tota l'activitat de l'usuari al grup.

RF14: Al afegir a un membre a un grup se li ha d'enviar un correu electrònic informant-li que ha sigut afegit.

3.2.4 Pàgines d'assignatura

RF15: Les pàgines d'assignatura constaran de quatre seccions dedicades a les notícies, al material docent, a la wiki i al fòrum de l'assignatura. Els alumnes podran accedir a totes les seccions.

RF16: El fòrum de l'assignatura serà un espai on els alumnes podran publicar articles (sense imatge) i fer comentaris.

RF17: A la wiki de l'assignatura els alumnes podran editar, o inclús restaurar des d'una còpiade seguretat anterior, aquells articles que el professor hagi especificat com editables per l'alumnat.

RF18: L'usuari professor de l'assignatura podrà publicar notícies, pujar arxius al material docent, i publicar o editar articles a la wiki. I també restaurar-los des d'una còpiade seguretat. A més tindrà accés a les opcions d'administració de l'assignatura i permís per eliminar qualsevol article o comentari del fòrum de l'assignatura. A part d'això podrà realitzar el mateix que els seus alumnes.

RF19: A les opcions d'administració de l'assignatura el professor podrà fer la gestió dels alumnes matriculats, modificar el nom i curs de l'assignatura i també eliminar-la completament.

RF20: De la mateixa forma que en els grups, el professor podrà expulsar a un alumne de l'assignatura de dues formes, deixant constància dels seus articles i comentaris al fòrum, en aquest cas, o bé eliminant-los tots.

3.2.5 Taulell d'anuncis

RF21: Al taulell d'anuncis ha de poder accedir qualsevol usuari.

RF22: Els usuaris han de poder publicar i eliminar anuncis (amb o sense imatge adjunta)

RF23: Els usuaris han de poder votar (fer "M'agrada") els anuncis, i veure quants vots té un anunci. Els vots, a més, han de ser anònims, de manera que únicament s'ha de poder veure el nombre de vots d'un anunci i no qui ha votat a aquest.

RF24: L'administrador del sistema ha de poder accedir d'una forma externa al taulell d'anuncis i tindre permís per eliminar qualsevol anunci publicat.

3.2.6 Xat

RF25: Al xat ha de poder accedir qualsevol usuari.

RF26: L'usuari ha de poder enviar missatges de fins a 140 caràcters.

RF27: L'usuari ha de poder visualitzar tots els missatges que han sigut enviats al xat per la resta d'usuaris.

RF28: S'han de mostrar i emmagatzemar un màxim de 255 missatges al xat, al arribar a aquest límit s'hauran d'anar eliminant els missatges més antics.

3.2.7 Cerca

RF29: Al cercador ha de poder accedir qualsevol usuari.

RF30: L'usuari ha de poder cercar usuaris, grups o assignatures, i a més articles o arxius a les pàgines dels seus amics, grups o assignatures.

RF31: Paral·lelament se li han de mostrar a l'usuari un màxim de 10 suggeriments d'amistat. És a dir, suggeriment d'usuaris que estan relacionats amb ell o amb els seus amics, directa o indirectament.

RF32: Al cercar grups només s'han de cercar grups en els quals l'usuari ja sigui membre o grups públics, els grups no públics no han d'aparèixer en cap cas. Al cercar assignatures han d'aparèixer totes, no obstant, l'usuari només podrà accedir a aquelles en les que estigui matriculat.

RF33: Al cercar articles o arxius, apart d'en els grups de l'usuari, s'ha de cercar també en els grups públics. En cas de cercar en assignatures, només es cercarà en aquelles les quals l'usuari estigui matriculat.

3.2.8 Administració d'usuari

RF34: L'administració d'usuari ha de constar de varis llistats on l'usuari pugui veure i administrar els seus amics, grups i assignatures. Els professors, a més, tindran un llistat addicional per veure el alumnes matriculats a les seves assignatures.

RF35: En els grups o assignatures en els quals l'usuari sigui l'administrador o el professor es podrà accedir a les opcions d'administració d'aquests directament des d'aquest llistat.

RF36: L'administració ha de comptar, també, amb formularis per poder crear nous grups i assignatures (en cas que siguin professors).

RF37: Des de l'administració d'usuari també s'ha de poder accedir directament a la pàgina de perfil de l'usuari, al taulell d'anuncis, al xat i al cercador.

3.2.9 Peticions

RF38: Si un usuari demana una petició d'amistat a un altre usuari, primer s'establirà una relació d'amistat parcial amb aquest. Tot seguit se li enviarà un correu electrònic amb un enllaç per si vol acceptar-la.

RF39: Si un usuari accepta una petició d'amistat, s'establirà una relació d'amistat completa amb l'usuari que havia fet la petició.

RF40: Si un usuari demana una petició per entrar a un grup, se li enviarà un correu electrònic a l'administrador del grup amb un enllaç per si vol acceptar-la.

RF41: Si un administrador de grup accepta una petició d'un usuari per entrar al seu grup, s'establirà a l'usuari que ha enviat la petició com a membre d'aquest grup.

3.2.10 Administració general

RF42: L'administració general serà una interfície aïllada de tota la resta del sistema i per accedir es requerirà un usuari i contrasenya d'administrador.

RF43: L'administrador podrà administrar (modificar i eliminar) usuaris, grups o assignatures. I també els tipus d'arxiu, les ciutats, els cursos i els tipus de grups que estan emmagatzemats a la base de dades.

RF44: L'administrador també ha de poder registrar nous usuaris. Tant alumnes com **professors**.

RF45: L'administrador ha de poder enviar invitacions de registre a través de correu electrònic.

RF46: L'administrador ha de poder establir o trencar relacions entre usuaris, grups i assignatures a la base de dades.

RF47: L'administrador ha de poder accedir al taulell d'anuncis des de la seva interfície i eliminar qualsevol anunci.

3.3 Requeriments no funcionals

3.3.1 Requeriments de rendiment

RNF1: L'aplicació ha de poder ser accessible per múltiples usuaris al mateix moment.

RNF2: L'aplicació ha de trigar el mínim temps en carregar-se al navegador, per tant, s'ha d'optimitzar al màxim en aquest aspecte.

RNF3: Totes les imatges pujades pels usuaris hauran de transformar-se a format jpg i ocupar el mínim espai possible sense afectar notòriament a la seva qualitat.

RNF4: La mida màxima de les imatges pujades pels usuaris no pot superar els 2 MB (Megabytes), i els 5 MB per la mida dels arxius.

RNF5: Per tal d'evitar incoherències a la base de dades, s'hauran de validar tots els formularis abans de registrar les dades a aquesta.

3.3.2 Requeriments de disseny

RNF6: La interfície ha de ser clara i concisa, i fàcilment comprensible per usuaris no experts.

RNF7: L'aplicació haurà de veure's correctament en resolucions de 1366x768 i superiors.

RNF8: L'aplicació ha de visualitzar-se correctament a la majoria dels navegadors web més utilitzats.

RNF9: La mida de la lletra ha de ser l'adequada, perquè es pugui llegir correctament.

RNF10: La interfície ha de tenir un bon disseny, usabilitat i on tots els elements han de ser fàcilment accessibles en tot moment.

3.3.3 Requeriments de seguretat

RNF11: Les contrasenyes dels usuaris han d'estar encriptades a la base de dades.

RNF12: Els codis dels enllaços corresponents a peticions d'usuaris, tant peticions d'amistat com per entrar a un grup, han de ser únics, de manera que cap altre usuari malintencionat

pugui acceptar peticions que no li pertoquen. Una vegada hagi sigut acceptada una petició amb un codi únic s'haurà d'eliminar aquest per no ser utilitzable més vegades.

RNF13: Les invitacions que envii l'administrador per registrar-se a xarxa social també han de tenir un codi únic, i una vegada registrat l'usuari, aquest codi ha de quedar inservible.

RNF14: El SGBD ha de tenir la màxima seguretat per evitar el robatori de dades, que es consideraria un cas greu d'incompliment de la Llei de protecció de dades.

RNF15: Cada usuari ha de tenir iniciada una sessió al accedir al sistema, i ha de poder tancar-la quan vulgui sortir, evitant que qualsevol altra persona accedeixi al sistema i utilitzi la seva compta.

3.4 Casos d'ús

Els diagrames de casos d'ús són representacions gràfiques dels requisits funcionals. Descriuen les interaccions entre usuaris (actors) i el sistema, i expliquen que és el que fa exactament el sistema sense entrar al detall de com ho fa.

Un cas d'ús és una descripció del que fa el sistema des del punt de vista de l'usuari i que pot estar relacionat amb altres casos d'ús mitjançant les relacions <<include>> (inclusió), on un cas d'ús destí incorpora les mateixes funcionalitats que un altre, i <<extends>> (extensió), on un cas d'ús incorpora les funcionalitats d'altre cas d'ús i a més realitza alguna tasca més. Els actors són les entitats que inicien les seqüències de casos d'ús.

L'objectiu de la representació mitjançant casos d'ús és fer comprensible el funcionament del sistema tant per usuaris experts, com podrien ser programadors, com per usuaris no experts.

A continuació es mostren els següents casos d'ús:

3.4.1 Inici

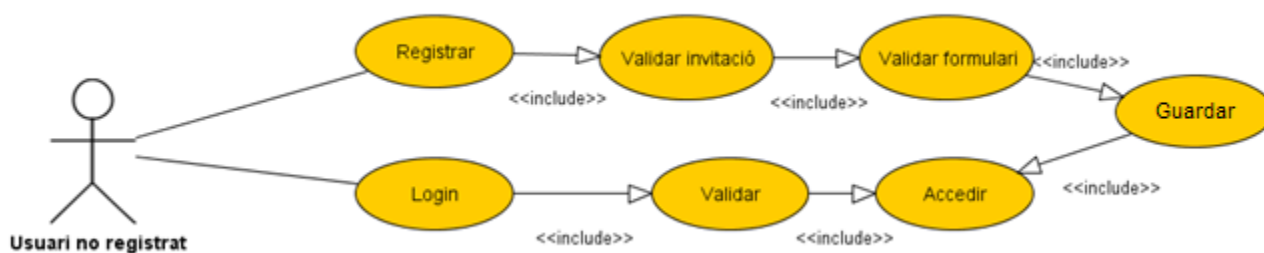


Figura 5: Diagrama de casos d'ús de la interfície d'inici

- **Registrar:** Obre el formulari de registre.
- **Validar invitació:** Valida si el codi d'invitació és vàlid.
- **Validar formulari:** Valida si l'usuari ha introduït totes les dades obligatòries, i si les dades introduïdes tenen el format correcte.
- **Guardar:** Registra les dades de l'usuari a la base de dades.
- **Login:** Obre el formulari d'accés al sistema.
- **Validar:** Valida si l'usuari i contrasenya introduïts per l'usuari són correctes.
- **Accedir:** Inicia sessió al sistema i carrega la interfície de pàgina d'usuari.

3.4.2 Pàgina d'usuari

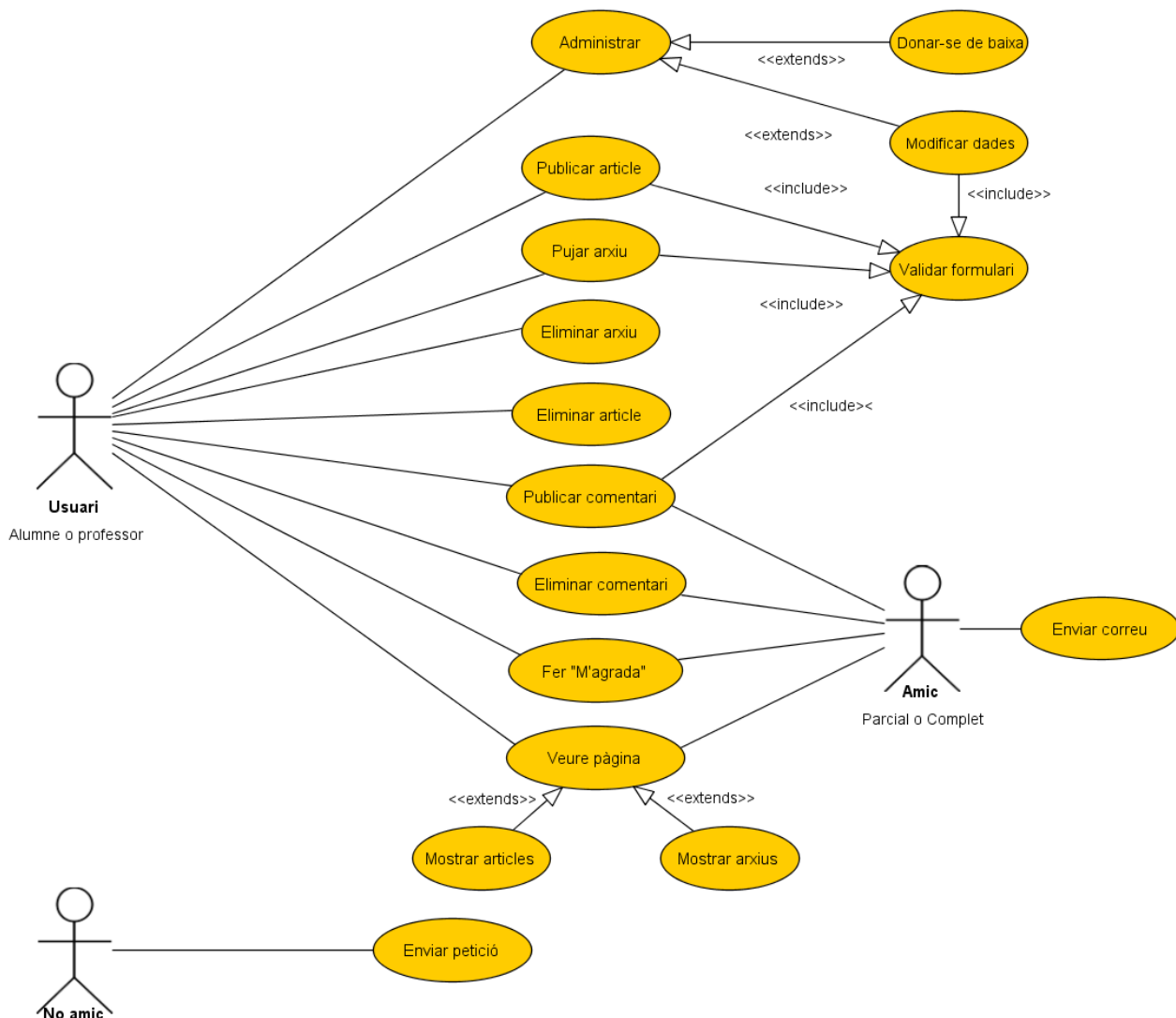


Figura 6: Diagrama de casos d'ús de les pàgines d'usuari.

- **Administrar:** Obre la interfície d'administració de l'usuari.
- **Donar-se de baixa:** Elimina a l'usuari de la base de dades i és tanca sessió.
- **Modificar dades:** Modifica les dades de l'usuari a la base de dades.
- **Validar formulari:** Valida que les dades introduïdes a un formulari són correctes i que estan introduïdes totes les que es demanen.
- **Enviar correu:** Obre la interfície externa per enviar un correu a l'usuari.
- **Publicar article:** Obre el formulari per publicar un nou article.
- **Pujar arxiu:** Obre el formulari per pujar un nou arxiu.
- **Eliminar arxiu:** Elimina un arxiu de la pàgina d'usuari.
- **Eliminar article:** Eliminar un article de la pàgina d'usuari.

- **Veure pàgina:** Es mostra tota la pàgina de l'usuari amb tota la seva informació i la secció d'articles (per defecte).
- **Mostrar articles:** Es mostra la secció d'articles de l'usuari a la pàgina d'usuari.
- **Mostrar arxius:** Es mostra la secció d'arxius de l'usuari a la pàgina d'usuari.
- **Publicar comentari:** Obre el formulari per publicar un nou comentari.
- **Fer "M'agrada":** Estableix que a un usuari li agrada un article, comentari o un arxiu.
- **Eliminar comentari:** Elimina un comentari (L'usuari propietari pot eliminar qualsevol comentari de la seva pàgina i els seus amics només els que ells hagin publicat).
- **Enviar petició:** Envia una petició d'amistat a l'usuari.

3.4.3 Pàgina de grup

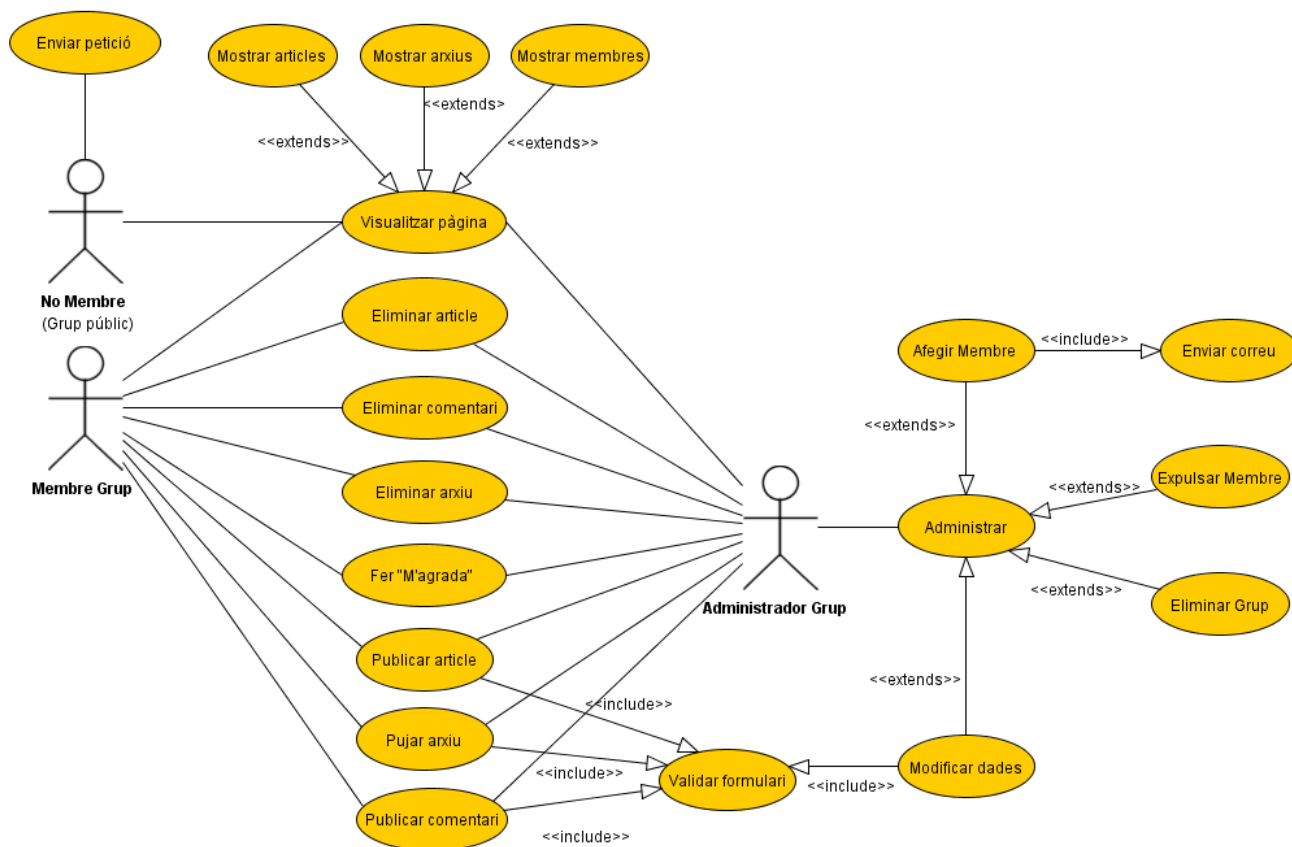


Figura 7: Diagrama de casos d'ús de les pàgines de grup.

- **Administrar:** Obre la interfície d'administració del grup.
- **Modificar dades:** Modifica les dades del grup a la base de dades.
- **Afegir Membre:** Afegeix un nou membre al grup.
- **Enviar correu:** Envia un correu a l'usuari que hagi sigut afegit al grup.
- **Expulsar Membre:** Expulsa a un membre del grup.

- **Eliminar Grup:** Elimina el grup de la base de dades.
- **Validar formulari:** Valida que les dades introduïdes a un formulari són correctes i que estan introduïdes totes les que es demanen.
- **Publicar article:** Obre el formulari per publicar un nou article.
- **Publicar arxiu:** Obre el formulari per pujar un nou arxiu.
- **Eliminar arxiu:** Elimina un arxiu del grup (Els membres del grup només poden eliminar els arxius que ells hagin pujat i l'administrador qualsevol).
- **Eliminar article:** Eliminar un article (Els membres només poden eliminar els articles que ells hagin publicat i l'administrador qualsevol).
- **Eliminar comentari:** Eliminar un comentari (Els membres només poden eliminar els comentaris que ells hagin publicat i l'administrador qualsevol).
- **Visualitzar pàgina:** Es mostra tota la pàgina del grup i la secció d'articles (per defecte).
- **Mostrar articles:** Es mostra la secció d'articles a la pàgina del grup.
- **Mostrar arxius:** Es mostra la secció d'arxius a la pàgina del grup.
- **Mostrar membres:** Es mostra la secció de membres a la pàgina de grup.
- **Publicar comentari:** Obre el formulari per publicar un nou comentari.
- **Fer "M'agrada":** Estableix que a un usuari li agrada un article, comentari o arxiu del grup.
- **Eliminar comentari:** Elimina un comentari (Els membres només poden eliminar els seus comentaris i l'administrador del grup qualsevol).

3.4.4 Pàgina d'assignatura

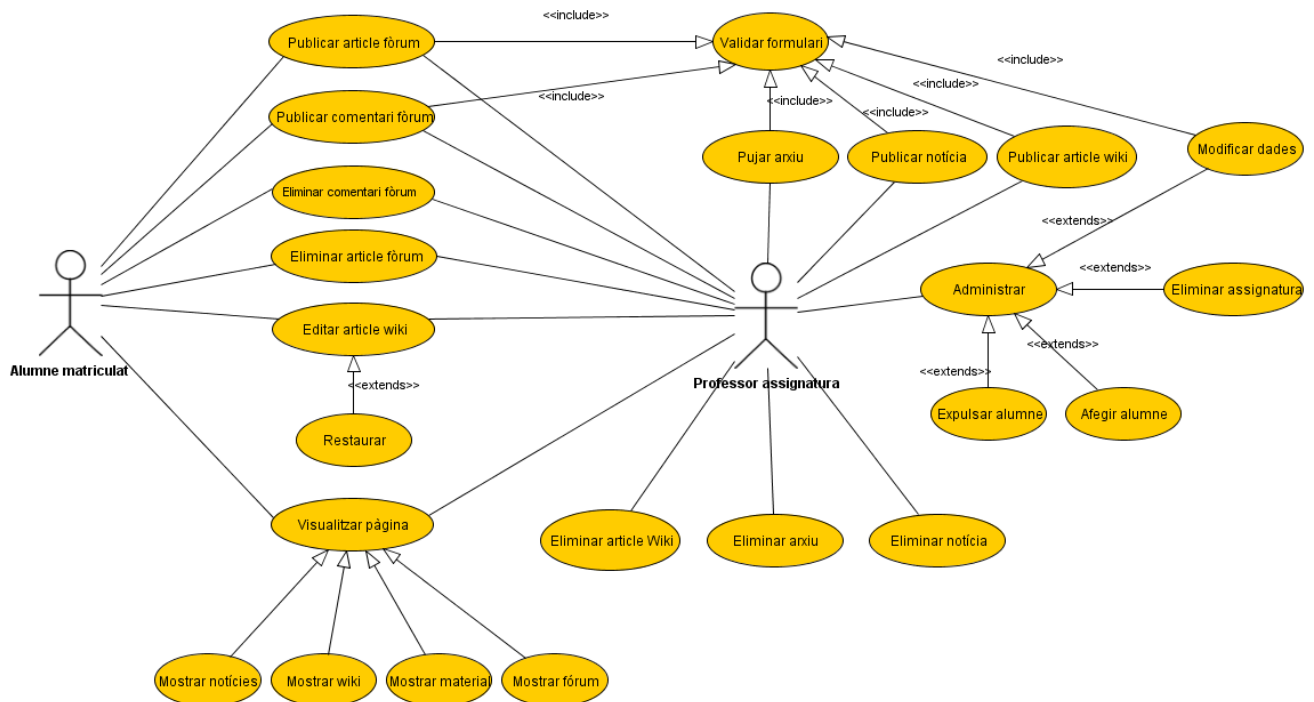


Figura 8: Diagrama de casos d'ús de les pàgines d'assignatura.

- **Administrar:** Obre la interfície d'administració de l'assignatura.
- **Modificar dades:** Modifica les dades de l'assignatura a la base de dades
- **Afegir alumne:** Matricula a un nou alumne a l'assignatura
- **Expulsar alumne:** Expulsa a un alumne de l'assignatura.
- **Eliminar assignatura:** Elimina l'assignatura de la base de dades.
- **Validar formulari:** Valida que les dades introduïdes a un formulari són correctes i que estan introduïdes totes les obligatòries.
- **Publicar article wiki:** Obre el formulari per publicar un nou article a la wiki.
- **Pujar arxiu:** Obre el formulari per pujar un nou arxiu.
- **Publicar notícia:** Obre el formulari per publicar una nova notícia.
- **Editar article wiki:** Obre el formulari per editar un article a la wiki.
- **Restaurar:** Restaura la còpia anterior d'un article a la wiki.
- **Eliminar arxiu:** Elimina un arxiu de l'assignatura.
- **Eliminar article wiki:** Elimina un article de la wiki de l'assignatura.
- **Eliminar notícia:** Elimina una notícia de l'assignatura.
- **Publicar article fòrum:** Obre el formulari per publicar un nou article al fòrum.

- **Eliminar article fòrum:** Elimina un article al fòrum de l'assignatura (Els alumnes només poden eliminar els seus propis articles i el professor tindrà permís per eliminar qualsevol article).
- **Publicar comentari fòrum:** Obre el formulari per publicar un nou comentari.
- **Eliminar comentari fòrum:** Elimina un comentari (Els alumnes només poden eliminar els seus propis comentaris i el professor tindrà permís per eliminar qualsevol).
- **Visualitzar pàgina:** Es mostra tota la pàgina de l'assignatura i la secció de notícies (per defecte).
- **Mostrar notícies:** Es mostra la secció de notícies a la pàgina de l'assignatura.
- **Mostrar wiki:** Es mostra la secció de la wiki a la pàgina de l'assignatura.
- **Mostrar material:** Es mostra la secció de material docent a la pàgina de l'assignatura.
- **Mostrar fòrum:** Es mostra la secció del fòrum a la pàgina de l'assignatura.

3.4.5 Taulell d'anuncis

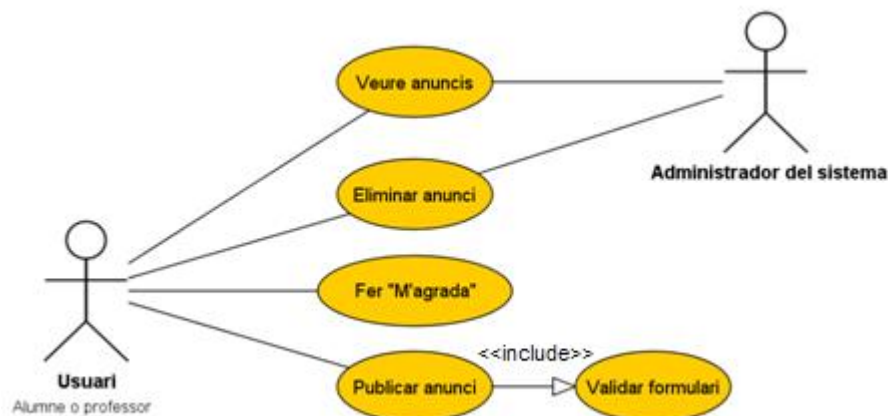


Figura 9: Diagrama de casos d'ús del taulell d'anuncis.

- **Publicar anunci:** Obre el formulari per publicar un nou anunci.
- **Validar formulari:** Valida si les dades introduïdes al formulari són correctes i que s'han introduït totes les dades obligatòries que es demanaven.
- **Veure anuncis:** Mostra tots els anuncis publicats al taulell d'anuncis.
- **Eliminar anunci:** Elimina un anunci del taulell d'anuncis (Els usuaris només poden eliminar els anuncis que ells hagin publicat i l'administrador del sistema pot eliminar qualsevol).
- **Fer "M'agrada":** Estableix que a un usuari li agrada un anunci.

3.4.6 Xat

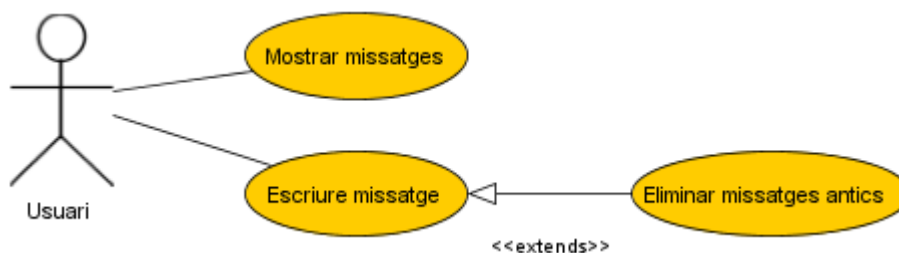


Figura 10: Diagrama de casos d'ús del xat.

- **Mostrar missatges:** Mostra els missatges publicats al xat.
- **Escriure missatge:** Emmagatzema un missatge escrit per l'usuari i el publica al xat.
- **Eliminar missatges antics:** Elimina el missatge més antic si s'ha arribat a 255 missatges publicats al xat.

3.4.7 Cerca

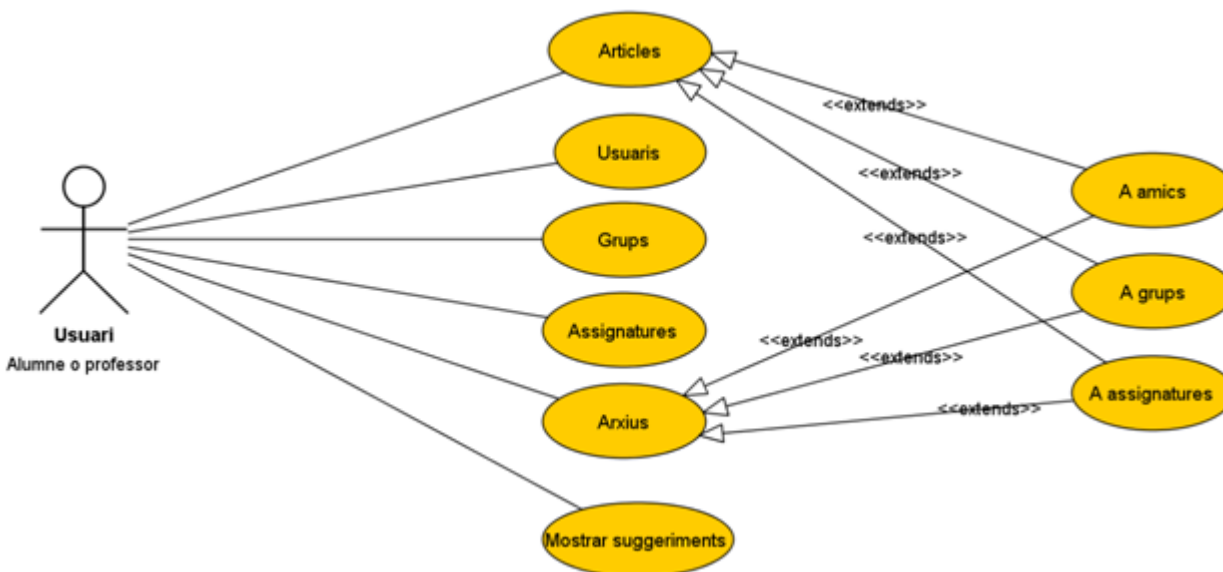


Figura 11: Diagrama de casos d'ús de la cerca.

- **Usuaris:** Busca usuaris d'acord amb el text i les opcions introduïdes per l'usuari, i els mostra per pantalla.
- **Grups:** Busca grups d'acord amb el text i les opcions introduïdes per l'usuari, i els mostra per pantalla.
- **Assignatures:** Busca assignatures d'acord amb el text i les opcions introduïdes d'acord amb el text i les opcions introduïdes per l'usuari, i les mostra per pantalla..

- **Articles:** Busca articles d'acord amb el text i les opcions introduïdes per l'usuari, i els mostra per pantalla.
- **Arxius:** Busca arxius d'acord amb el text i les opcions introduïdes per l'usuari, i els mostra per pantalla.
- **Mostrar suggeriments:** Mostra suggeriments d'amistat a l'usuari.
- **A amics:** Busca articles o arxius d'acord amb el text i les opcions introduïdes per l'usuari entre tots els amics de l'usuari.
- **A grups:** Busca articles o arxius d'acord amb el text i les opcions introduïdes per l'usuari entre els grups en els quals l'usuari sigui membre i en els grups públics.
- **A assignatures:** Busca articles o arxius d'acord amb el text i les opcions introduïdes per l'usuari entre les assignatures en els quals l'usuari estigui matriculat.

3.4.8 Administració d'usuari

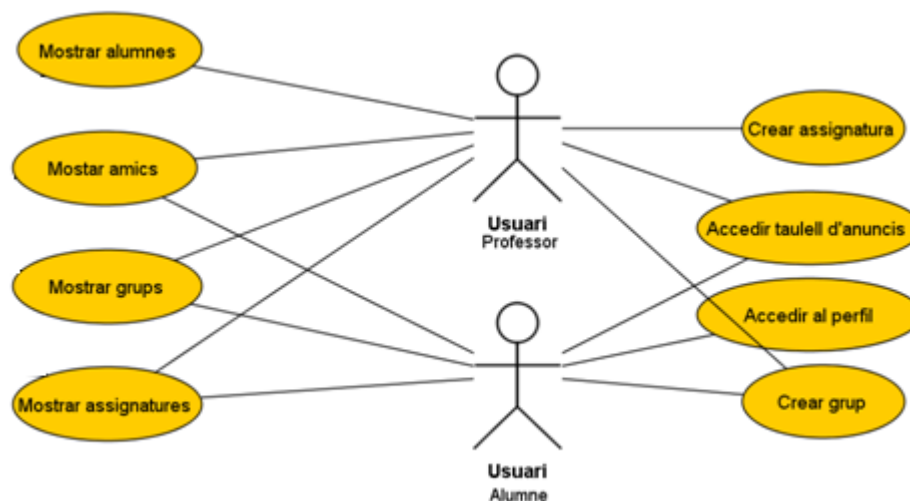


Figura 12: Diagrama de casos d'ús de l'administració d'usuari.

- **Mostrar amics:** Mostra un llistat amb els amics de l'usuari i les opcions d'administració d'aquests.
- **Mostrar alumnes:** Mostra un llistat amb els alumnes de les assignatures de l'usuari (cas que l'usuari sigui professor).
- **Mostrar grups:** Mostra un llistat amb els grups en els quals l'usuari sigui membre i les opcions d'administració d'aquests.
- **Mostrar assignatures:** Mostra un llistat amb les assignatures en les quals l'usuari estigui matriculat o sigui professor (cas que l'usuari sigui un professor), i les opcions d'administració d'aquestes.

- **Accedir al perfil:** Accedeix a la pàgina de perfil de l'usuari.
- **Accedir al taulell d'anuncis:** Accedeix al taulell d'anuncis.
- **Crear assignatura:** Mostra el formulari per crear una nova assignatura.
- **Crear grup:** Mostra el formulari per crear un nou grup.

3.4.9 Peticions

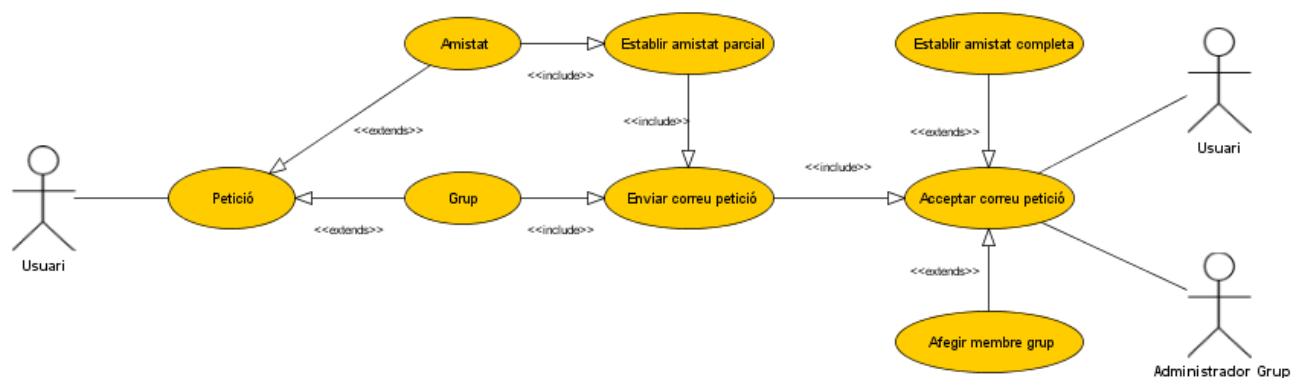


Figura 13: Diagrama de casos d'ús del sistema de peticions.

- **Petició:** Envia una petició a un usuari.
- **Amistat:** Envia una petició d'amistat a un usuari.
- **Grup:** Envia una petició per entrar a un grup a l'administrador d'un grup.
- **Establir amistat parcial:** Estableix una amistat parcial entre un usuari i un altre.
- **Establir amistat completa:** Estableix una amistat completa entre un usuari i un altre.
- **Afegir membre grup:** Afegeix a un usuari com a membre d'un grup.
- **Enviar correu petició:** Envia un correu de petició a un usuari.
- **Acceptar correu petició:** Un usuari accepta la petició enviada per altre usuari.

3.4.10 Administració general

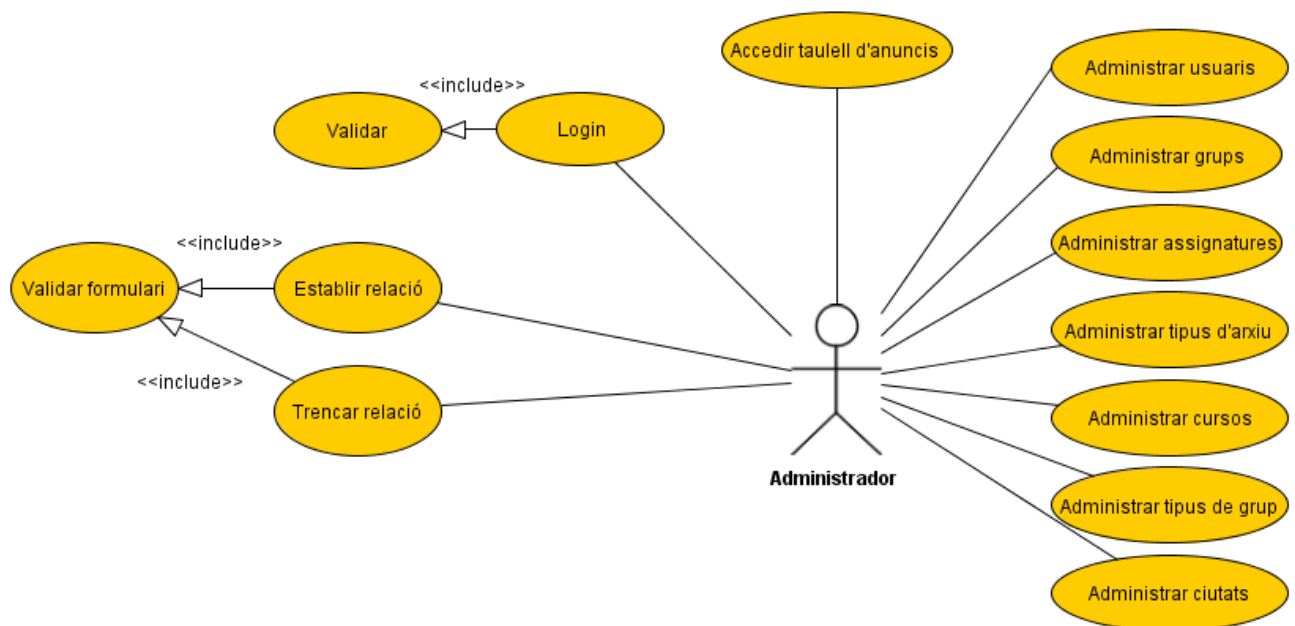


Figura 14: Diagrama de casos d'ús de l'administració general.

- **Establir relació:** Estableix una relació entre dos elements (usuaris, grups o assignatures).
- **Trencar relació:** Trencar una relació, si existeix, entre dos elements (usuaris, grups o assignatures).
- **Validar formulari:** Valida si les dades introduïdes al formulari són correctes i que estan totes les demanades.
- **Login:** Obre el formulari d'accés a la interfície d'administrador.
- **Validar:** Valida si l'usuari i la contrasenya d'administrador introduïdes són correctes.
- **Accedir al taulell d'anuncis:** Es mostra el taulell d'anuncis amb, únicament, permís per eliminar qualsevol anunci.
- **Administrar usuaris:** Es mostra un llistat dels usuaris del sistema i les seves opcions d'administració.
- **Administrar grups:** Es mostra un llistat dels grups del sistema i les seves opcions d'administració.
- **Administrar assignatures:** Es mostra un llistat de les assignatures del sistema i les seves opcions d'administració.
- **Administrar tipus d'arxiu:** Es mostra un llistat dels tipus d'arxiu del sistema i les seves opcions d'administració.
- **Administrar cursos:** Es mostra un llistat dels diferents cursos del sistema i les seves opcions d'administració.

- **Administrar tipus de grup:** Es mostra un llistat dels tipus de grups del sistema i les seves opcions d'administració.
- **Administrar ciutats:** Es mostra un llistat de les ciutats registrades al sistema i les seves opcions d'administració.

4 Disseny

4.1 Introducció

A continuació es mostra com s'ha dissenyat l'aplicació, que inclou el disseny de la base de dades, les diferents classes que es connecten amb aquesta i la interfície d'usuari. Tot ha estat dissenyat de manera que es compleixin tant els requeriments funcionals com els no funcionals. Per fer el disseny de l'aplicació ens hem d'avançar una mica i hem de pensar amb quins tipus d'eines farem la implementació, ja que depenent d'això el disseny pot variar. Com que el desenvolupament es farà amb una metodologia seqüencial iterativa, a cada iteració es pot anar modificant el disseny d'acord amb els requisits de les eines que s'utilitzen en la implementació.

4.2 Base de dades

Per dissenyar la base de dades i acomplir els requisits funcionals es va haver d'afrontar el problema d'emmagatzemar en conjunt totes les dades d'usuaris, grups, assignatures i la resta d'elements, i també les relacions que tenen lloc entre aquests mateixos. Per a resoldre això es va decidir utilitzar dos sistemes de bases de dades. Un seria un sistema de bases de dades relacional, que s'encarregaria d'emmagatzemar la informació d'aquests elements, i un altre seria un sistema de bases de dades orientades a grafs, que emmagatzemaria les relacions entre els diferents elements de l'aplicació.

Un graf és un conjunt d'objectes, anomenats nodes, units entre si per un conjunt d'enllaços anomenats arestes. Aquests enllaços o arestes poden tenir associat un pes determinat i poden ser dirigides o no dirigides. En la nostra aplicació els nodes seran usuaris, grups i d'altres i estaran relacionats entre ells per arestes dirigides amb un pes que codificarà el seu tipus de relació.

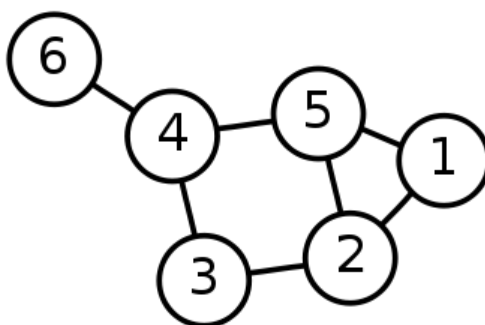


Figura 15: Exemple d'un graf amb arestes no dirigides i sense pes.

Les bases de dades orientades a grafs (BDOG) representen la informació com a nodes d'un graf i les seves relacions com a arestes del mateix, de manera que es pot utilitzar la teoria de grafs per a recórrer la base de dades.

La utilització de bases de dades orientades a grafs és una tècnica habitual en aplicacions de xarxes socials, ja que permet tot un seguit d'avantatges que no són assolibles en sistemes de bases de dades relacionals per a grans volums de dades. Podríem destacar els avantatges següents:

- Permet centralitzar totes les relacions i realitzar consultes més amples, que en un sistema relacional hauríem de fer amb diferents taules.
- No s'ha de definir un nombre determinat d'atributs, de manera que un node pot tenir relacionats només 2 atributs i un altre només 1, per exemple.
- Es pot recórrer la base de dades de forma jeràrquica, de nodes pares a nodes fills i a l'inrevés.
- Permeten unes millores considerables en el rendiment respecte a les bases de dades relacionals.

Es van investigar i provar varies solucions i SGBD orientats a grafs. Finalment es va trobar una solució que es va considerar la més viable per al projecte, per la seva facilitat d'instal·lació i administració. Que és utilitzar MariaDB (un SGBD basat en MySQL) conjuntament amb un plugin o complement per aquest anomenat OQGraph. Que incorpora un motor d'emmagatzemament que permet emmagatzemar grafs com si fossin una taula més del model relacional, i realitzar consultes SQL estàndard sense cap tipus de modificació i de manera transparent a l'usuari.

Internament, però, aquestes taules es comporten com a grafs amb totes les avantatges que això comporta. A més, OQGraph permet aplicar directament alguns algorismes de la teoria de grafs, com l'algorisme de cerca en amplada.

La base de dades del sistema haurà d'estar dissenyada, per tant, pensant que estarà implementada utilitzant el model relacional, amb la característica que aquelles taules que utilitzin el motor OQGraph es veuran des de fora com una taula però que internament realment seran grafs.

4.2.1 Base de dades relacional

A continuació es mostra el diagrama UML del disseny de base de dades que utilitza el sistema:

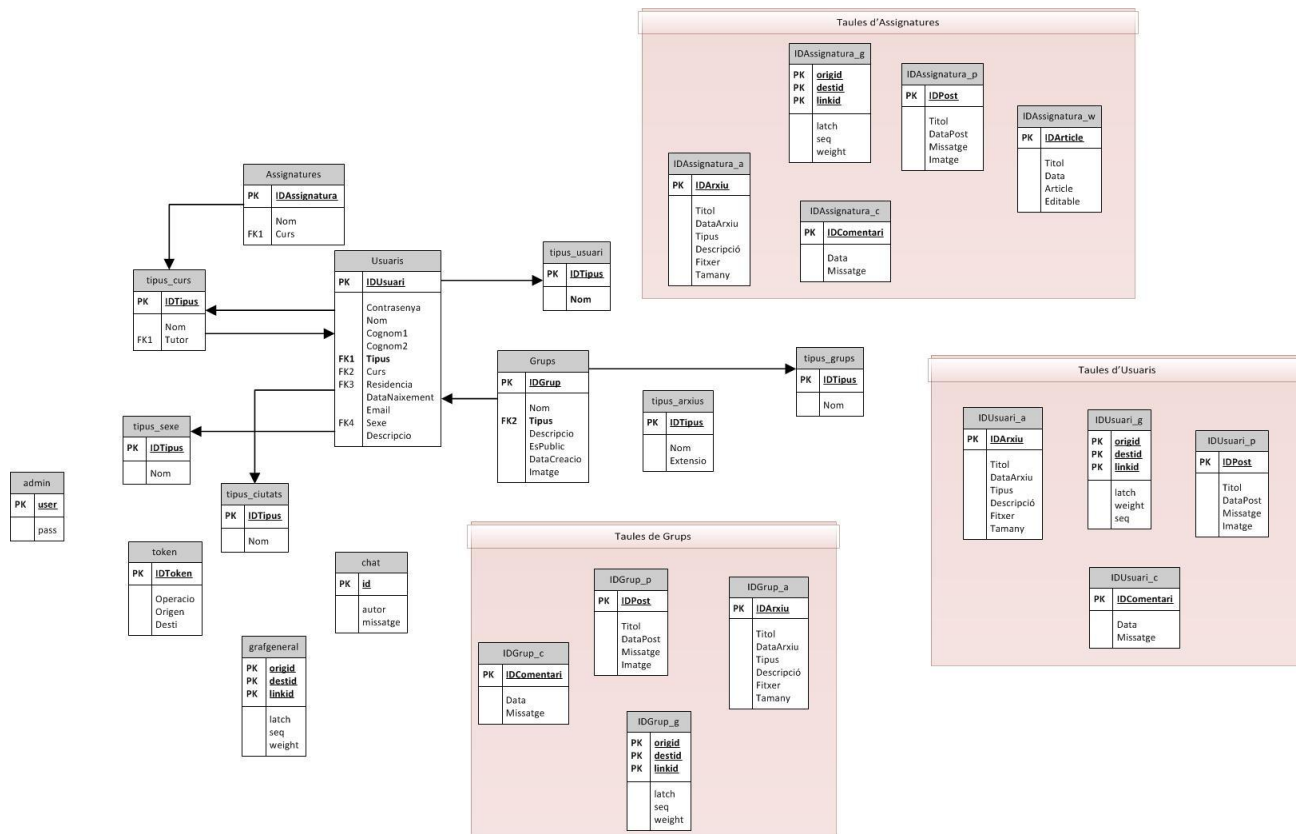


Figura 16: Diagrama UML inicial de la base de dades.

D'aquest diagrama destaquem els següents punts:

- Les taules “Usuaris”, “Grups” i “Assignatures” emmagatzemaran les dades dels usuaris, grups i assignatures, respectivament.
- La taula “admin” emmagatzemarà l'usuari i contrasenya de l'administrador del sistema.
- La taula “token” emmagatzemarà tokens, uns tipus de codis de seguretat que se li donaran als usuaris per a realitzar diferents accions.
- La taula “chat” emmagatzemarà els missatges del xat del sistema, fins a un màxim de 255.
- Totes les taules començades per “tipus_” seguit d'un nom, emmagatzemaran els diferents tipus de coses de l'aplicació (tipus d'usuari, de grup, de curs, d'arxiu, etc.)
- La taula marcada com a “grafgeneral” utilitzarà el motor OQGraph i emmagatzemarà el graf de relacions entre usuaris, grups i assignatures.

- Les taules englobades a “Taules de Grups”, “Taules d’Usuaris” i “Taules d’Assignatures” seran múltiples taules que emmagatzemaran dades relacionades amb les pàgines d’usuaris, de grups i d’assignatures..

El nom d’aquestes taules múltiples constarà d’un número identificador del grup, usuari o assignatura, seguit del identificador del tipus de taula que és. Per tal de poder així identificar cada taula individualment i les taules específiques de cada grup, usuari o assignatura en conjunt.

S’han definit els quatre tipus de taula següents:

- **ID_p**: Taula que emmagatzemarà les dades dels articles (o post) d’un grup, usuari o assignatura.
- **ID_c**: Taula que emmagatzemarà les dades dels comentaris d’un grup, usuari o assignatura.
- **ID_a**: Taula que emmagatzemarà les dades dels arxius d’un grup, usuari o assignatura.
- **ID_g**: Taula que utilitzarà el motor OQGraph i que emmagatzemarà el graf de relacions internes dins d’un grup, usuari o assignatura.

(Per exemple, un grup amb el número identificador 1000005 tindria les taules 1000005_p, 1000005_c, 1000005_a i 1000005_g)

Les assignatures, a més, incorporen un tipus de taula més, que és el següent:

- **ID_w**: Taula que emmagatzemarà les dades dels articles de la wiki d’una assignatura.

La taula d’articles de la wiki tindrà un sistema específic per emmagatzemar les còpies de seguretat dels articles de la wiki. Aquest sistema es basarà en utilitzar dos registres de la taula consecutius, el primer emmagatzemarà la còpiade l’article actual i el consecutiu la còpiade seguretat de l’article. És important garantir que quan s’afegeixi un nou article aquest dos registres siguin estrictament consecutius.

En afegit, les assignatures també incorporen la funcionalitat d’utilitzar la taula d’articles (ID_p) per emmagatzemar tant articles del fòrum com notícies del professor. Per fer això s’utilitzarà l’atribut “imatge” d’aquesta taula per “marcar” si es tracta d’una notícia o d’un article al fòrum. Així per tant, ja considerem d’avant mà que a les assignatures les notícies o els articles publicats no podran tenir imatge associada.

4.2.2 Base de dades orientada a grafs

Per emmagatzemar els grafs, hem dit, utilitzarem el motor OQGraph de MariaDB. Aquest motor ens especifica quin és el format que han de tenir les taules que emmagatzemin grafs, i que és el següent:

```
CREATE TABLE db.tblname (  
    latch SMALLINT UNSIGNED NULL,  
    origid BIGINT UNSIGNED NULL,  
    destid BIGINT UNSIGNED NULL,  
    weight DOUBLE NULL,  
    seq BIGINT UNSIGNED NULL,  
    linkid BIGINT UNSIGNED NULL,  
    KEY (latch, origid, destid) USING HASH,  
    KEY (latch, destid, origid) USING HASH  
    ) ENGINE=OQGRAPH;
```

Del format de la taula que acabem de mostrar l'única cosa que hem d'explicar són els tipus de dades que admeten els atributs "origid", "destid" i "weight" que són amb els que treballarem.

Per altra banda, aquest motor funciona de manera que nosaltres només ens hem de preocupar d'afegir les relacions entre un **node origen** (atribut "origid"), un **node destí** (atribut "destid") i el **pes de l'aresta** dirigida (atribut "weight"), o relació, que uneix aquests dos. El motor s'encarrega de fer la resta d'accions necessàries amb la resta d'atributs.

Els atributs "origid" i "destid" són de tipus "BIGINT UNSIGNED", això vol dir que en un servidor MariaDB per a sistemes de 64 bits podrien abastar valors des de 0 fins a 18.446.744.073.709.551.614 (~18 trilions de nodes), i just la meitat en sistemes de 32 bits, des de 0 fins a 9.223.372.036.854.775.807 (~9 trilions de nodes).

Per tant, amb això obtenim el nombre total de nodes diferents que podem tenir en un mateix graf.

Per altra banda, l'atribut "weight" és de tipus "DOUBLE", això ens permet introduir números en coma flotant amb doble precisió. Encara que, no obstant, a nosaltres no ens interessa aquesta possibilitat de precisió ja que utilitzarem el pes de les arestes per codificar el tipus de relació entre dos nodes, i per tant, aquests números que codifiquin relacions sempre seran enters.

Com que en un mateix graf poden concórrer al mateix temps diversos tipus d'elements com usuaris o grups, i com els nodes han de ser nodes numèrics, com acabem de veure, hem de

dividir els diversos tipus d'elements en rangs de valors numèrics per tal de poder identificar-los individualment al graf.

Així, a la taula "grafgeneral", es relacionaran usuaris, grups i assignatures, i els identificarem, segons els rangs de valors que podem veure a la següent taula:

Tipus ID	Rang de valors
Grups	001.000.000 – 001.999.999
Usuaris	002.000.000 – 009.999.999
Assignatura	900.000 – 901.000

Taula 10: Taula d'identificadors d'usuaris, grups i assignatures.

Veiem que els identificadors estan emmarcats dins d'un rang limitat, això vol dir que tindrem un nombre limitat d'elements que podrà tenir el graf, i alhora l'aplicació, al mateix temps. Així, podrem tenir 999.999 grups, 8.999.999 usuaris i 1.000 assignatures.

Aquesta forma de codificació s'ha decidit tenint en compte l'àmbit educatiu al qual es destinarà aquesta xarxa social. Com que principalment està pensada per institucions educatives, considerem que registrar 8.999.999 usuaris és una cosa bastant improbable de superar en una institució d'aquest tipus. Així com també es considerarà bastant improbable el fet d'arribar a la xifra de quasi 1 milió de grups i de 1.000 assignatures.

Per tant no s'espera que aquests rangs puguin ser assolits ni tenir cap problema en aquest sentit. De totes maneres s'informarà degudament a l'administrador del sistema d'aquestes limitacions per tal que aquest pugui administrar l'aplicació de manera que garanteixi que això no passarà.

Per altra banda, el pes de les arestes dirigides, que especificaran el tipus de relació entre dos nodes, es codificarà segons la següent taula:

Nom Relació	Valor	Origen	Destí	Descripció
es_amic	2	IDUsuari	IDUsuari	Codifica una relació d'amistat entre dos usuaris.
es_membre	3	IDUsuari	IDGrup	Codifica si un usuari és membre d'un grup.
admin	7	IDGrup	IDUsuari	Codifica quin usuari és l'administrador d'un grup.
professor	8	IDAssignatura	IDUsuari	Codifica quin usuari és el professor d'una assignatura.
matriculat	9	IDUsuari	IDAssignatura	Codifica si un usuari està matriculat a una assignatura.

Taula 11: Taula de codificació de les relacions entre dos nodes ("grafgeneral").

Per tal de fer més clara i comprensible la taula anterior, veiem el següent exemple:

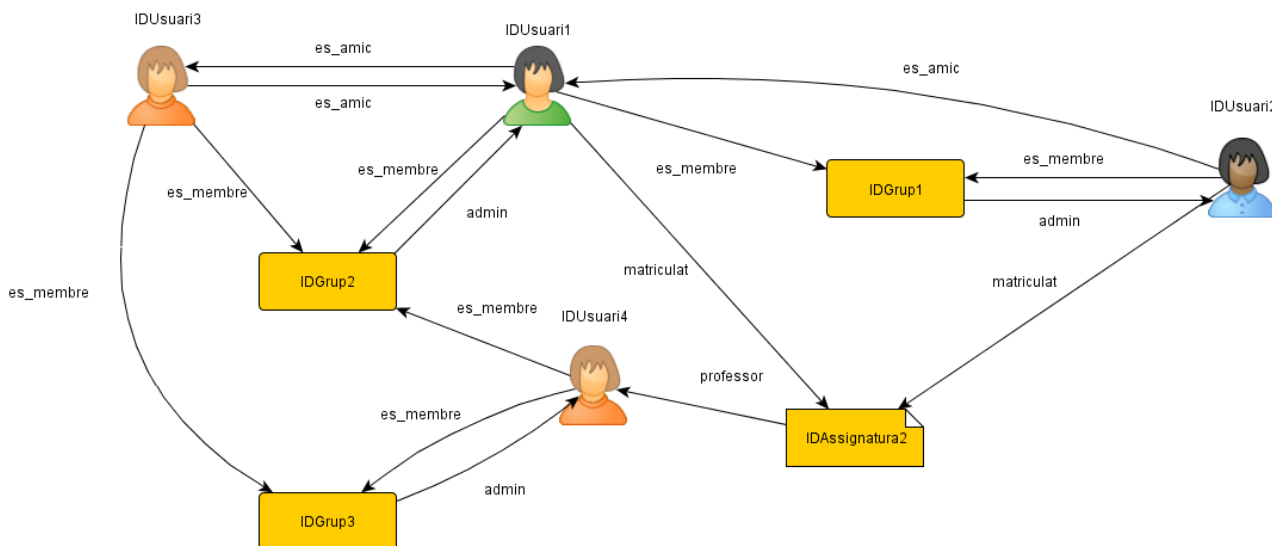


Figura 17: Exemple del graf general de la xarxa

En resum, a aquest diagrama de graf tenim que:

- L'usuari 3 té una relació d'amistat completa amb l'usuari 1, i alhora és membre del grup 2 i del grup 3.
- L'usuari 2 té una relació d'amistat parcial amb l'usuari 1, per tant l'usuari 1 pot visualitzar la pàgina de l'usuari 2 però l'usuari 2 no pot visualitzar la del 1.

- L'usuari 2 està matriculat a l'assignatura 2, que té com a professor l'usuari 4, i és administrador del grup 1.
- L'usuari 1 és administrador del grup 3, és membre del grup 1 i està matriculat a l'assignatura 2.
- L'usuari 4 és administrador del grup 3, membre del grup 2, i és professor de l'assignatura 2.

En altre lloc, tenim les taules del tipus "ID_g" on tindrem emmagatzemats el graf de relacions internes de les pàgines de grups, usuaris i assignatures. En aquestes es relacionaran els usuaris, post (articles), comentaris i arxius d'una pàgina d'usuari, grup o assignatura. Els identificarem segons els rangs de valors que podem veure a la següent taula:

Tipus ID	Rang de valors
Usuaris	002.000.000 – 009.999.999
Articles	010.000.000 – 099.999.999
Comentaris	100.000.000 – 999.999.999
Arxius	000.000.000 – 000.899.999

Taula 12: Taula d'identificadors dels elements dins una pàgina d'usuari, grup o assignatura.

Amb aquests rangs veiem que podem tenir 899.999.999 comentaris, 899.99 arxius i 89.999.999 articles (post) diferents i en una mateixa pàgina d'usuari, grup o assignatura. En aquest cas també es considera que aquests rangs difícilment seran assolibles en aquesta aplicació. Per exemple, pel que fa al nombre d'articles, en un grup podríem publicar quasi 90 milions d'articles, que repartits en 10 anys, per exemple, serien uns 24.600 articles diaris, cosa que és bastant improbable que succeeixi, pel que fa al nombre de comentaris aquest és encara major.

En tot cas, l'administrador del sistema és el que ha de garantir que no és superin aquests rangs.

Pel que fa al pes de les arestes a les taules de grups, usuaris i assignatures, que especificarà el tipus de relació entre els nodes, es codificarà segons la següent taula:

Nom Relació	Valor	Origen	Destí	Descripció
es_agrafat	0	IDPost IDComentari IDArxiu	IDUsuari	Codifica si a un usuari li agrada un article, comentari o arxiu.
es_comentat	1	IDPost	IDComentari	Codifica a quin article pertany un comentari.
Posteja	4	IDUsuari	IDPost	Codifica quin usuari ha publicat un article.
comenta	5	IDUsuari	IDComentari	Codifica quin usuari ha fet un comentari.
Puja	6	IDUsuari	IDArxiu	Codifica quin usuari ha pujat un arxiu

Taula 13: Taula de codificació de les relacions entre dos nodes en una pàgina d'usuari, grup o assignatura.

Per fer més clara i comprensible la taula anterior, tornem i veiem un exemple:

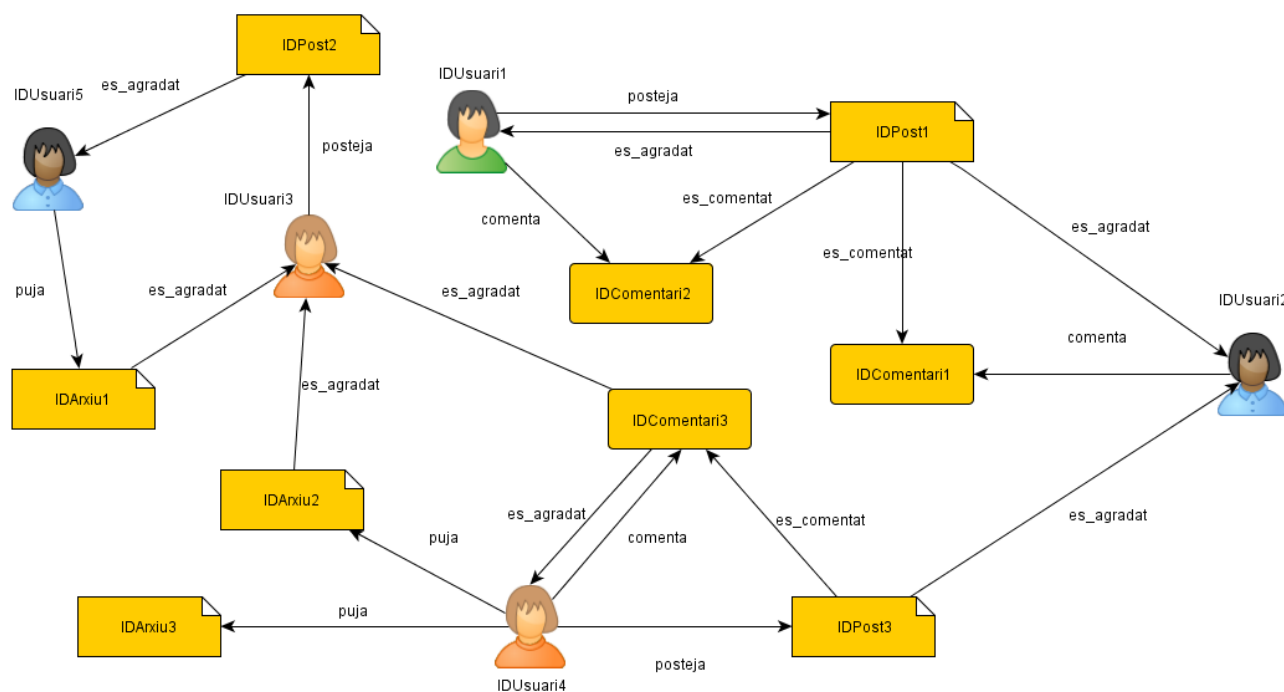


Figura 18: Exemple d'un graf de grup.

En resum, en l'anterior diagrama de graf tenim especificat que:

- L'usuari 5 ha pujat l'arxiu 1, que li agrada a l'usuari 3.
- L'usuari 3 ha publicat l'article 2 (post), que li agrada a l'usuari 5.
- L'usuari 1 ha publicat l'article 1, que li agrada a ell mateix.
- L'article 1 té dos comentaris associats, un és el comentari 1, que ha estat fet per l'usuari 2, i l'altre és el comentari 2, que ha estat fet per l'usuari 1.
- L'usuari 4 ha publicat l'article 3, que li agrada a l'usuari 2. Té 1 comentari associat, que és el comentari 3, que ha estat fet per ell mateix i que li agrada a l'usuari 3, i a ell mateix. També ja pujat l'arxiu 3 i l'arxiu 2, que li agrada a l'usuari 3.

4.3 Estructura de classes

Per a desenvolupar l'aplicació s'implementaran un conjunt de classes que són les que faran d'intermediàries entre la interfície de l'aplicació i la base de dades. Amb la separació en classes utilitzarem implícitament totes les característiques i conceptes dels llenguatges de programació orientats a objectes, com pot ser el concepte d'herència, interfície, abstracció, etc. Totes les classes estaran incloses a un mateix fitxer o biblioteca, que serà referenciat o inclòs després a la interfície per poder treballar així amb la base de dades.

S'han especificat 14 classes, que són les següents: **ConnexioBD**, **Fòrum**, **Graf**, **Usuari**, **Grup**, **Assignatura**, **Post**, **Comentari**, **Arxiu**, **Article**, **Chat**, **Cerca**, **Token** i **tipus**.

Els atributs i els mètodes de cada classe s'han declarat de tal manera que tinguin concordança amb el disseny de la base de dades i compleixin les funcions especificades als requeriments funcionals.

Per últim s'han declarat també dos interfícies, que engloben les funcionalitats, i que són implementades després per diverses classes.

Una interfície és un tipus de classe especial que serveix per definir quins mètodes s'hauran de codificar obligatòriament a les classes que les implementin. D'aquesta manera ens assegurem que totes les classes que implementin una interfície tindran definits els mètodes que són declarats a aquesta.

4.3.1 Diagrama de classes

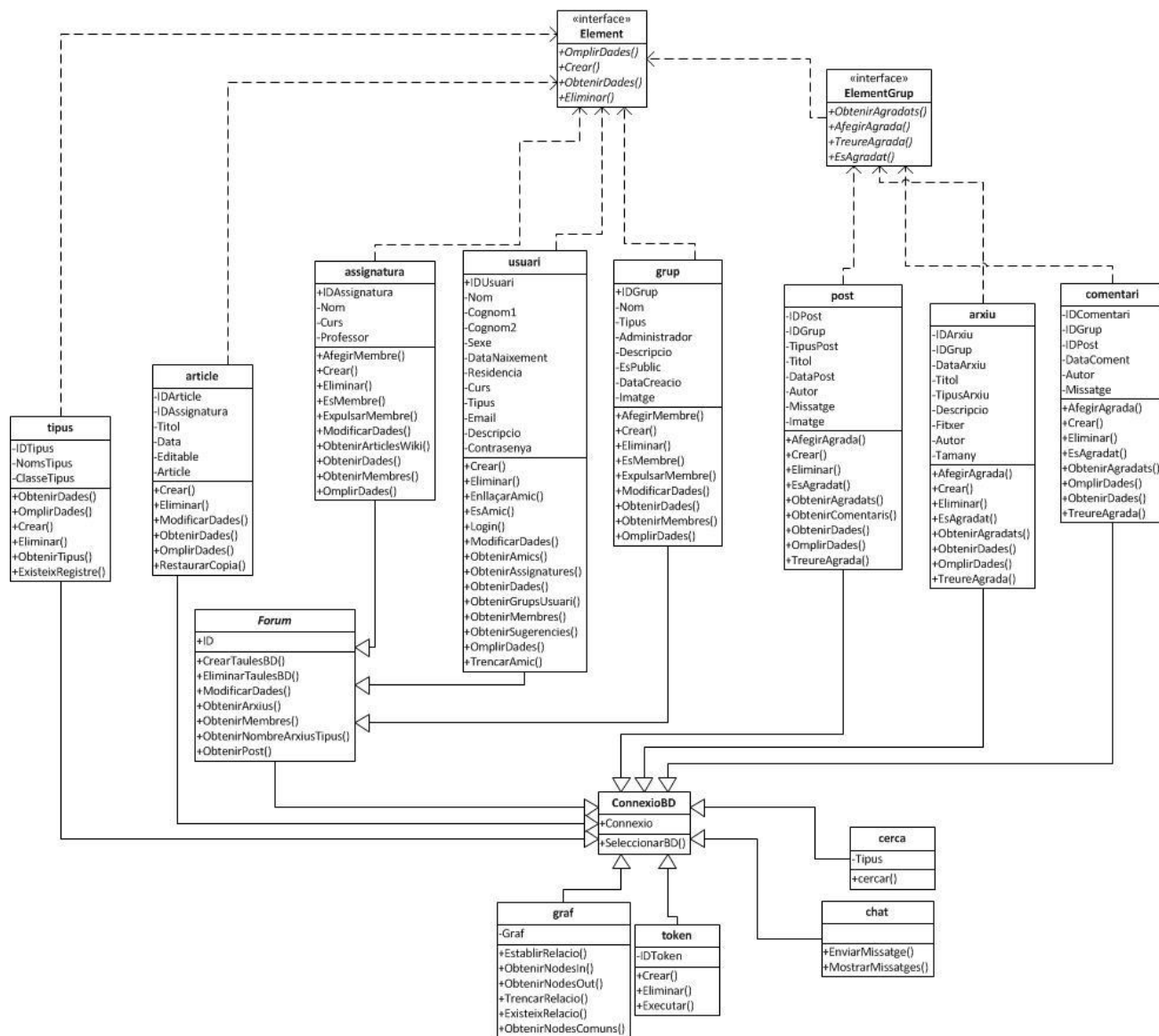


Figura 19: Diagrama de classes de l'aplicació.

4.3.2 Interfícies

Les interfícies són una part important en l'estructura de classes, ja que ens determinen els mètodes que seran codificats a les classes que les implementin. I amb això la forma amb que aquestes s'utilitzaran per realitzar diferents accions bàsiques, amb els objectes, que explicarem més endavant.

La interfície principal i més important és la interfície "Element", que és la que declara els mètodes **OmplirDades()**, **ObtenirDades()**, **Crear()** i **Eliminar()**.

Com hem dit, totes les classes que implementin a aquesta interfície hauran de definir aquests mètodes obligatòriament.

En general, a totes les classes, aquests mètodes faran el següent:

- El mètode **OmplirDades()** afegirà les dades passades per paràmetre als atributs de l'objecte que el cridi.
- **ObtenirDades()** retornarà les dades que té un element a la base de dades a partir del seu identificador.
- **Crear()** emmagatzemarà les dades d'un objecte a la base de dades i **Eliminar()** les eliminarà.

El conjunt de classes que implementen directa o indirectament la interfície "Element" són les següents: **usuari, grup, assignatura, tipus, arxiu, post, article i comentari**.

Hi haurà una seqüència d'execució específica per a realitzar els dos tipus d'accions principals que podem realitzar amb objectes d'aquestes classes.

Una acció serà la de crear o registrar un objecte a la base de dades, per exemple si volem registrar un nou usuari, un nou grup, un nou article (post) o un nou arxiu.

Per posar un exemple concret, si el que volem és crear o registrar a un usuari a la base de dades hauríem de seguir la seqüència de codificació mostrada al diagrama següent:

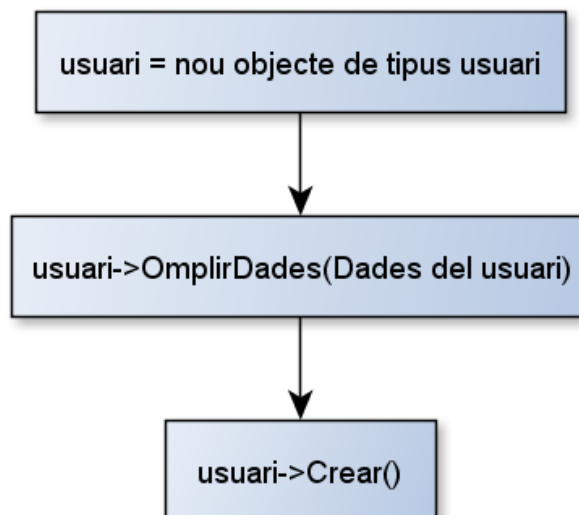


Figura 20: Exemple de seqüència d'execució per crear un nou usuari.

Que en llenguatge de programació PHP es codificaria de la següent forma:

```

$NouUsuari = new usuari();
$Dades = array(
    "Contrasenya" => md5("contrasenya"),
    "Nom" => "Fernando",
    "Cognom1" => "Pradas",
    "Cognom2" => "Monllor",
    "DataNaixement" => '1991-05-23',
    "Residencia" => 2,
    "Curs" => 1,
    "Tipus" => 0,
    "Email" => "Fernando.Pradas@uab.cat",
    "Sexe" => 0);

$NouUsuari->OmplirDades($Dades);
$NouUsuari->Crear();
  
```

En resum, primer creem un nou objecte de la classe usuari, després definim un array¹ de dades amb les dades d'aquest, i cridem al mètode **OmplirDades()**, que afegeix les dades als atributs de l'objecte. Després es crida al mètode **Crear()**, que és el que emmagatzema aquestes dades a la base de dades. Això és important ja que abans de cridar a **Crear()**, sempre hem d'haver cridat abans al mètode **OmplirDades()** per omplir els atributs de l'objecte amb les dades, sino es generarà un error.

Aquesta seqüència és exactament igual per a la creació de qualsevol objecte de les classes que hem anomenat abans, això si, cadascun amb les seves particularitats pròpies de cada classe. Cal destacar que les contrasenyes dels usuaris seran, codificades i emmagatzemades a la base de dades amb l'algorisme MD5, que ja inclou PHP.

L'altre tipus d'acció que es pot realitzar és la de cridar a qualsevol altre mètode dels que tingui implementats la classe. Abans de poder cridar a algun d'aquests haurem de seguir seqüència d'execució determinada. Per exemple, per cridar a algun dels mètodes definits a la classe usuari haurem de seguir la seqüència d'execució que es mostra al següent diagrama:

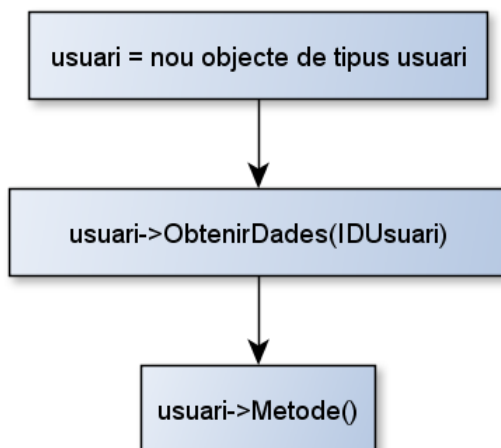


Figura 21: Exemple de seqüència d'execució per obtenir les dades d'un usuari.

En resum, abans de poder cridar a algun altre mètode de la classe usuari haurem d'obtenir les dades d'un usuari, en aquest cas. Ja que aquests mètodes realitzen algun tipus d'acció amb un usuari en concret, i per això es requereixen les dades de l'usuari amb el que es treballarà. Per obtenir les dades primer crea un objecte de classe usuari i després es crida al mètode **ObtenirDades()**, passant-li per referència l'identificador de l'usuari. Una vegada obtingudes les dades, podem cridar a altres mètodes.

Aquesta seqüència, com l'altra, també és exactament igual per qualsevol de les classes anteriors, que implementen a la interfície "Element". No obstant tot, en algunes classes existeixen excepcions on hi ha mètodes que poden ser cridats directament sense necessitat de cridar prèviament **ObtenirDades()**.

Per posar un exemple en concret, si volem afegir un amic a un usuari es codificaria de la següent forma en PHP:

```
$Usuari = new usuari();  
$DadesUsuari = $Usuari->ObtenirDades(2000003);  
$Usuari->EnllaçarAmic(2000004);
```

El que fa el codi anterior és crear un objecte de classe usuari, obtenir les dades de l'usuari amb l'identificador 2000003, per poder després cridar al mètode **EnllaçarAmic(2000004)**, que estableix una relació d'amistat de l'usuari 2000003 amb l'usuari 2000004.

Amb tot això, hem definit el mètode de funcionament bàsic dels mètodes de la interfície "Element" i que impliquen a tota la resta de mètodes de les classes.

Per altra banda, tenim la interfície "ElementGrup", que és una extensió de la interfície "Element". Per tant, totes les classes que implementin la interfície "ElementGrup" hauran d'implementar tots els mètodes d'aquesta i a més els de la interfície "Element".

Aquesta interfície ens defineix els mètodes **AfegirAgrada()**, **TreureAgrada()**, **EsAgradat()** i **ObtenirAgradats()**. Les classes que implementen "ElementGrup", i que hauran de codificar els anteriors mètodes són les següents: **post**, **arxiu** i **comentari**.

El mètode **AfegirAgrada()** ens permeten establir que a un usuari li agrada un element (post, arxiu o comentari).

Amb el mètode **TreureAgrada()** podem establir que ja no li agrada.

Amb el mètode **EsAgradat()** determinar si li agrada o no a un usuari.

I amb el mètode **ObtenirAgradats()** obtenir tots els usuaris als quals li agrada i element.

4.3.3 ConnexioBD

Aquesta classe és la que s'encarrega d'establir una connexió o enllaç amb el SGBD, té un únic mètode que és el que permet seleccionar amb quina base de dades es vol treballar, i que és **SeleccionarBD()**. Recordem que tindrem dos bases de dades, una serà la base de dades general i l'altra la base de dades de grups. Aquesta classe és heretada per absolutament tota la resta de classes, ja que de fet, totes les classes tenen com a mínim algun mètode que ha de consultar la base de dades.

4.3.4 Fòrum

Aquesta és una classe abstracta, que és un tipus de classe especial que ens permet definir mètodes abstractes. Un mètode abstracte és un mètode que es defineix, però que no s'implementa, i que serà obligatori implementar-ho en totes classes filles, de la mateixa manera que a les interfícies. També, però, podem implementar directament mètodes normals, i que podran cridats directament des de totes les classes filles.

Aquesta classe és heretada per les classes “usuari”, “grup” i “assignatura” i té implementats tot un seguit de mètodes o funcions que seran necessaris tant per poder crear nous usuaris, grups o assignatures, com per poder implementar les diferents pàgines d'aquests que formaran part de la interfície d'usuari.

Així es destaquen els mètodes **CrearTaulesBD()** i **EliminarTaulesBD()** que creen i eliminen les diferents taules necessàries per una pàgina d'usuari, grup o assignatura, respectivament.

Els mètodes **ObtenirPost()** i **ObtenirArxius()**, que retornen una llista dels articles i arxius d'una pàgina d'usuari, grup o assignatura, respectivament.

També tenim el mètode **ObtenirNombreArxiusTipus()** que ens retorna una llista amb la quantitat d'arxius que hi ha de cada tipus en una pàgina d'usuari, grup o assignatura.

I per últim es defineixen com a mètodes abstractes els mètodes **ObtenirMembres()** i **ModificarDades()** que hauran de ser codificats en totes les classes filles.

4.3.5 Graf

Aquesta classe permet treballar amb els grafs de la base de dades i és utilitzada internament en algunes de les classes per realitzar totes les accions que requereixin treballar o consultar un graf.

Implementa els mètodes **EstablirRelacio()** i **TrencarRelacio()** que ens permeten establir o eliminar relacions (o arestes) entre nodes d'un graf, respectivament.

El mètode **ExisteixRelacio()** que permet obtenir si existeix o no una relació (o aresta) entre dos nodes, tant de forma simètrica com asimètrica.

El mètode **ObtenirNodesIn()** que permet obtenir tots els nodes origen que convergeixen en un mateix node donat i amb un tipus de relació determinada (o aresta amb un pes determinat). I el mètode **ObtenirNodesOut()**, que fa exactament el contrari, retorna tots els nodes destí que divergeixen d'un node donat i amb un tipus de relació determinada.

Per últim tenim el mètode **ObtenirNodesComuns()** que retorna tots els nodes destí que tenen una relació determinada en comú amb dos nodes donats, funció que ens servirà després per calcular els amics en comú entre dos usuaris, per exemple.

4.3.6 Usuari

La classe “usuari” ens permet crear nous usuaris i obtenir les dades dels ja creats, així com també realitzar diferents accions amb aquests.

Aquesta classe hereta les funcions de la classe abstracta “Forum” i per tant, ens obliga a declarar els mètodes **ModificarDades()** i **ObtenirMembres()**.

A més tenim accés a tots els mètodes declarats i implementats a “Forum” com **ObtenirPost()** o **ObtenirArxius()** que, com hem dit, ens retornaran el llistat d'articles o d'arxius publicats en una pàgina d'usuari.

També tenim els mètodes **EnllaçarAmic()**, **TrencarAmic()** i **EsAmic()** que ens permeten establir o trencar una relació d'amistat i determinar si existeix una amistat o no, respectivament.

Els mètodes **ObtenirAmics()**, **ObtenirAssignatures()** i **ObtenirGrupsUsuari()** que ens retornen una llista dels amics, assignatures o grups d'un usuari, respectivament.

Per altra banda, el mètode **ModificarDades()** permetrà modificar les dades d'un usuari, i el mètode **ObtenirMembres()** serà declarat, per obligació de la classe “Forum”, però no implementat ja que en un usuari no compleix cap funció.

Com que la classe “usuari” també implementa la interfície “Element” i haurem d'implementar els mètodes **OmplirDades()**, **Crear()**, **ObtenirDades()** i **Eliminar()** que faran les funcions abans esmentades.

Per altra banda tenim el mètode **Login()** que ens determina a partir d'un nombre usuari i d'una contrasenya, si aquests són correctes i concorden a la base de dades.

I per últim tenim el mètode **ObtenirSugerencies()** que ens retorna una llista amb els suggeriments d'amistat d'un usuari. Per determinar això, ho farà de la manera següent:

El motor OQGraph incorpora, entre d'altres, l'algorisme de cerca en amplada en un graf, aquest algorisme el que fa és recórrer cada node i explorar tots els veïns, i per cadascun dels veïns recorre tots els seus altres veïns. Utilitzarem aquest algorisme per recórrer els amics de l'usuari, els amics dels amics de l'usuari, i així respectivament per poder mostrar els suggeriments d'amistat a aquest. Com que aquest algorisme pot tenir un cost computacional bastant alt en grafs molt grans haurem d'acotar aquest recorregut.

La primera forma d'acotar aquesta cerca és només tenir en compte els nodes que corresponen a un usuari, ja que al graf hi ha nodes que corresponen a altres coses, com grups o assignatures.

La segona forma d'acotar-ho és especificar fins a quina profunditat volem cercar. Es fa notar que en una xarxa social d'aquestes característiques es podria arribar a quasi qualsevol node, a excepció dels nodes aïllats. Per tant no ens interessin tots els nodes als quals es podria arribar, i s'opta per acotar la cerca fins a una profunditat màxima de 3, és a dir, recórrer els amics dels nostres amics, i alhora els amics dels seus amics.

L'última forma d'acotar-ho, per tal d'obtenir els millors resultats, és ordenar aquests amics pels atributs de "Curs" i de "Residència", i de forma descendent.

Així aquesta cerca ens donaria primer els usuaris, amics fins a una profunditat de 3, i que van al nostre curs i viuen a la mateixa residència. Després els que van al mateix curs, encara que no visquin al mateix lloc, i després els que viuen al mateix lloc però que no vaguin al mateix curs. Es considera que aquesta forma és una bona forma de donar suggeriments als usuaris, ja que el més probable que pot succeir és que un usuari conegui més a un alumne del seu curs, i que a més visqui al seu mateix lloc, que un que no.

Per últim, limitarem aquest resultat a 10 usuaris com a màxim per no saturar a l'usuari amb molts suggeriments. S'ha d'aclarir que aquesta cerca només tindrà resultats si com a mínim es troba a un usuari, és a dir, si l'usuari no té amics no tindrà resultats.

4.3.7 Grup

La classe "grup" ens permet crear nous grups i obtenir les dades dels ja creats, també realitzar diverses accions amb aquests.

Aquesta classe, de la mateixa manera que "usuari", hereta les funcions de la classe abstracta "Forum" i per tant, ens obliga a declarar els mètodes **ModificarDades()** i **ObtenirMembres()**. A més tenim accés a tots els mètodes declarats i implementats a "Forum".

El mètode **ModificarDades()** fa la funció de modificar les dades d'un grup i **ObtenirMembres()** ens retorna un llistat amb tots els usuaris que són membres d'un grup.

La classe grup també implementa la interfície "Element" i per tant haurem d'implementar els mètodes **OmplirDades()**, **Crear()**, **ObtenirDades()** i **Eliminar()** que faran les funcions abans esmentades i que s'hauran d'utilitzar seguint la seqüència d'execució d'escripta anteriorment.

A part d'això, aquesta classe declara 3 mètodes més.

Que són **AfegirMembre()**, que afegeix un nou usuari com a membre del grup. **ExpulsarMembre()**, que expulsa a un membre del grup.

I per últim **EsMembre()**, que retorna si un usuari és membre del grup o no.

4.3.8 Assignatura

La classe “assignatura” ens permet crear noves assignatures i obtenir les dades de les ja creades, a més ens permet realitzar diverses accions amb aquestes de la mateixa forma que un grup.

Aquesta classe també hereta les funcions de la classe abstracta “Forum”.

A més incorpora el mètode **ObtenirArticlesWiki()** que, con el seu nom indica, ens retorna un llistat dels articles publicats a la wiki de l'assignatura.

Pel que fa als mètodes **AfegirMembre()**, **ExpulsarMembre()** i **EsMembre()** el que fan és afegir a un nou alumne matriculat a una assignatura, expulsar-lo i determinar si està matriculat o no, respectivament.

I per últim, **ObtenirMembres()** ens retorna un llistat amb tots els alumnes matriculats a una assignatura i **ModificarDades()** ens permet modificar les seves dades.

La classe “assignatura” també implementa la interfície “Element” i per tant haurem d'implementar els mètodes **OmplirDades()**, **Crear()**, **ObtenirDades()** i **Eliminar()** que faran les funcions abans esmentades i que s'hauran d'utilitzar seguint la seqüència d'execució d'escria anteriorment.

4.3.9 Post

Aquesta classe ens serveix tant per crear com per obtenir les dades dels articles publicats a pàgines d'usuaris, grups o assignatures.

Implementa la interfície “ElementGrup”, que ahora és una extensió de la interfície “Element”, i per tant hem de declarar els mètodes **OmplirDades()**, **Crear()**, **ObtenirDades()**, **Eliminar()**, **AfegirAgrada()**, **TreureAgrada()**, **EsAgradat()** i **ObtenirAgradats()** i que faran les mateixes funcions que hem comentat anteriorment.

A més, aquesta classe implementa el mètode **ObtenirComentaris()**, que ens retorna un llistat de tots els comentaris associats a un article.

Per altra banda, es declara que al crear un objecte de la classe “post” hem d'especificar el grup al qual pertany o pertanyerà l'article, en el cas que anem a crear-lo.

4.3.10 Comentari

Aquesta classe ens serveix tant per crear com per obtenir les dades dels comentaris dels articles publicats a pàgines d'usuaris, grups o assignatures.

Implementa la interfície “ElementGrup”, que ahora és una extensió de la interfície “Element”. I per tant, de la mateixa manera que a la classe “post”, hem de declarar els mètodes **OmplirDades()**, **Crear()**, **ObtenirDades()**, **Eliminar()**, **AfegirAgrada()**, **TreureAgrada()**, **EsAgradat()** i **ObtenirAgradats()**, i que faran les funcions abans esmentades.

Per altra banda, es declara que al crear un objecte de la classe “comentari” hem d'especificar el grup i l'article al qual està o estarà associat, en cas que anem a crear-lo.

4.3.11 Arxiu

Aquesta classe ens serveix tant per crear com per obtenir les dades dels arxius publicats a pàgines d'usuaris, grups o assignatures.

De la mateixa manera que la classe “post”, implementa la interfície “ElementGrup” i per tant hem de declarar els mètodes **AfegirAgrada()**, **TreureAgrada()**, **EsAgradat()** i **ObtenirAgradats()** a més de tots els mètodes de la interfície “Element” (**OmplirDades()**, **Crear()**, **ObtenirDades()** i **Eliminar()**).

Per altra banda, es declara que al crear un objecte de la classe “arxiu” hem d'especificar el grup al que aquest pertany o pertanyerà, en cas que anem a crear un nou arxiu.

4.3.12 Article

Aquesta classe ens serveix tant per publicar com per obtenir les dades i editar els articles publicats a la wiki d'una assignatura.

Al declarar aquesta classe hem d'especificar l'assignatura a la qual pertany o pertanyerà aquest article, en cas que anem a crear-lo.

A diferència de la classe “post”, aquesta classe implementa directament la interfície “Element”, i per tant hem d'implementar els mètodes **OmplirDades()**, **Crear()**, **ObtenirDades()** i **Eliminar()**.

A més, aquesta classe implementa un mètode anomenat **ModificarDades()** que serveix per modificar les dades d'un article a la wiki, és a dir, per editar-lo.

I un altre mètode anomenat **RestaurarCòpia()** que ens serveix per restaurar una còpia de seguretat com a còpia actual d'un article i guardar la còpia actual com a còpia de seguretat.

Aquest últim mètode funcionarà de la forma que hem especificat al disseny de la base de dades, que resulta d'utilitzar dos registres consecutius a la taula d'articles de la wiki, un per guardar la còpia actual i altre per guardar la còpia de seguretat, i per tant el que farà realment el mètode **RestaurarCòpia()** serà intercanviar aquest dos registres.

4.3.13 Chat

Aquesta classe és necessària per a la implementació del chat a la interfície i únicament implementa els mètodes **EnviarMissatge()**, que emmagatzema un missatge a la base de dades i borra els antics si és necessari, i **MostrarMissatges()**, que mostra directament els missatges emmagatzemats a la base de dades.

Tal i com es requereix als requeriments de l'aplicació, al mètode **EnviarMissatge()** haurà d'eliminar el missatge més antic, abans d'inserir un de nou, si s'arriba a la xifra de 255 missatges.

4.3.14 Cerca

Aquesta classe és necessària per a la implementació del cercador i implementa un únic mètode, anomenat **Cercar()**. Es declara que al crear un objecte d'aquesta classe hem d'especificar quin és el tipus d'element que buscarem mitjançant al mètode **Cercar()** (usuaris, grups, arxius, etc.)

Hi haurà 9 tipus de cerca definits:

- Cerca d'usuaris.
- Cerca de grups.
- Cerca d'assignatures.
- Cerca d'articles a grups.
- Cerca d'articles a assignatures.
- Cerca d'articles a pàgines d'usuari.
- Cerca d'arxius a grups.
- Cerca d'arxius a assignatures.
- Cerca d'arxius a pàgines d'usuari.

Al cridar al mètode **Cercar()**, hem de passar-li el text que es cercarà, i opcionalment com a condició una llista d'elements on s'haurà de cercar. Aquesta llista només serà utilitzada en els casos de cercar articles i arxius, i que contindrà el llistat de grups, usuaris o assignatures on s'han de cercar aquests.

Pel que respecta a la implementació d'aquesta classe es va haver d'afrontar un problema que afecta al disseny de la base de dades. Per a poder implementar un algorisme de cerca més efectiu es va decidir utilitzar la cerca de tipus **Full-Text** de **MySQL**, aquest tipus de cerca permet cercar a varies columnes de la taula, de manera que per exemple podem buscar un usuari només introduint el seu cognom, o només el nom o el segon cognom, indiferentment.

El problema rau en que aquest tipus de cerca requereix crear un índex Full-Text a la base de dades, format per les columnes a les quals es vol cercar, i que només està implementat al motor MyISAM, un altre tipus de motor d'emmagatzemament per MariaDB, com OQGraph, i que no permet l'ús de claus foranies. Així, al canviar el motor d'emmagatzemament a MyISAM a les taules a les quals es volia utilitzar la cerca Full-text, es van haver d'eliminar les claus foranies. No obstant, per altra part, es va haver d'implementar amb codi, mitjançant PHP, les

funcions que abans realitzaven les claus foranies, per evitar incongruències a la base de dades.

En concret, les taules afectades són la taula “**usuaris**” i la taula “**grups**”. També les taules encarregades d'emmagatzemar els articles i arxius enels grups i assignatures, però com que no tenien de claus foranies, el seu funcionament és exactament igual de cara al seu accés extern.

4.3.15 Token

Aquesta classe fa les funcions de generar un token de seguretat, executar l'operació associada a aquest i d'eliminar-lo. Un token és un sistema de seguretat que dóna un codi a l'usuari per autoritzar a aquest a accedir a un sistema o a un servei computeritzat. En aquesta aplicació s'utilitzaran tokens per permetre als usuaris realitzar les accions d'enllaçar amistats, afegir membres a un grup i registrar-se al sistema. Els tokens de seguretat seran emmagatzemats a la base de dades juntament amb l'operació associada a aquests i les dades necessàries per realitzar-la.

El mètode **Crear()** el que farà és generar i emmagatzemar a la base de dades un token de seguretat totalment únic i aleatori, mitjançant el protocol criptogràfic OpenSSL que implementa PHP.

El mètode **Executar()** el que farà és, a partir d'un token, buscar i executar la funció que aquest té associat.

Per últim, el mètode **Eliminar()** el que farà és eliminar un token de la base de dades, de manera que no pugui ser executat de nou.

Amb això busquem aconseguir els requisits no funcionals de seguretat, ja que aquests tokens seran enviats als correus electrònics dels usuaris i aquests seran els únics que tindran accés a aquest codi de seguretat, evitant usos indeguts per part de terceres persones. A més es considera que protocol OpenSSL té un cert nivell de seguretat com per poder ser trencat.

Al declarar un objecte de la classe “token” hem d'especificar el token que correspon, en cas que anem a executar o eliminar-ho.

4.3.16 Tipus

Aquesta classe serveix per gestionar o crear nous tipus de curs, sexe, ciutats, arxius, grups i usuaris.

Implementa la interfície “Element” i per tant ha de codificar els mètodes **OmplirDades()**, **ObtenirDades()**, **Crear()** i **Eliminar()** i que faran les funcions abans esmentades i seguint la seqüència d'execució d'escripta.

A més incorpora dos mètodes més, un és el mètode **ObtenirTipus()**, que retorna tots els identificadors i noms d'un tipus donat. I l'altre és el mètode **ExisteixRegistre()**, que retorna si existeix o no un registre a la BD a partir d'un identificador de tipus donat.

4.4 Diagrames de seqüència

Els diagrames de seqüència són un tipus de diagrames que mostren la interacció entre diferents objectes i que modelen els missatges que aquests s'intercanvien en una línia de temps. Són diagrames que detallen a un nivell més baix els diferents casos d'ús i descriuen la seqüència d'execució d'aquests. En aquest sentit, són molt útils ja que permeten veure el funcionament integrat d'un conjunt d'objectes, i no per separat. A més, amb aquests, podem veure d'una forma més detallada el funcionament dels mètodes de la interfície “Element” que hem explicat anteriorment. A continuació es detallen un seguit de diagrames de seqüència que descriuen algunes de les seqüències de casos d'ús més importants de les que es poden veure al capítol 3:

4.4.1 Registre usuari

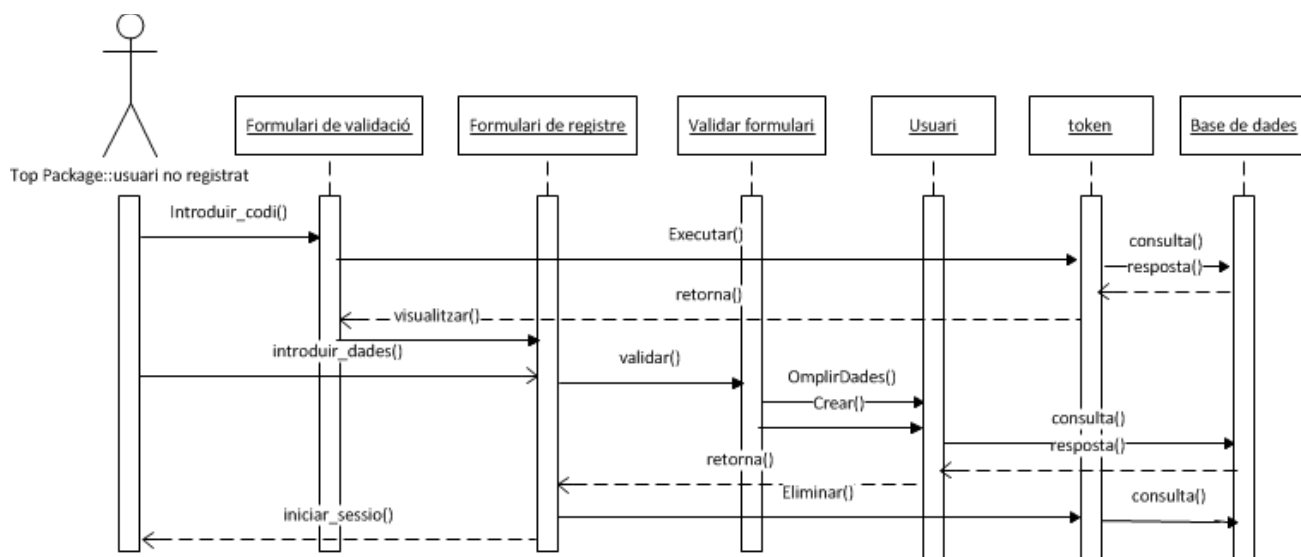


Figura 22: Diagrama de seqüència per al registre d'usuaris.

L'anterior diagrama descriu la seqüència que es segueix per a que un usuari no registrat pugui registrar-se.

Primerament l'usuari ha d'introduir el codi d'invitació a un formulari, posteriorment es crea un objecte de la classe "token" i es crida al mètode **Executar()**, que busca el token a la BD i executa l'operació associada.

En aquest cas, però, l'operació retorna si existeix o no el codi d'invitació a la BD.

Després es visualitza el formulari de registre on l'usuari ha d'introduir les dades, que són posteriorment validades per veure si estan correctament introduïdes.

Per registrar l'usuari a la base de dades es crea un objecte de la classe "usuari" i després es crida al mètode **OmplirDades()**, per omplir les dades de l'objecte. I així poder cridar al mètode **Crear()** que afegeix les dades de l'usuari a la BD. Després d'això des del formulari es crida al mètode **Eliminar()** de l'objecte "token", que elimina el token de la BD.

I per últim, s'inicia sessió i s'accedeix amb el nou usuari creat.

4.4.2 Login

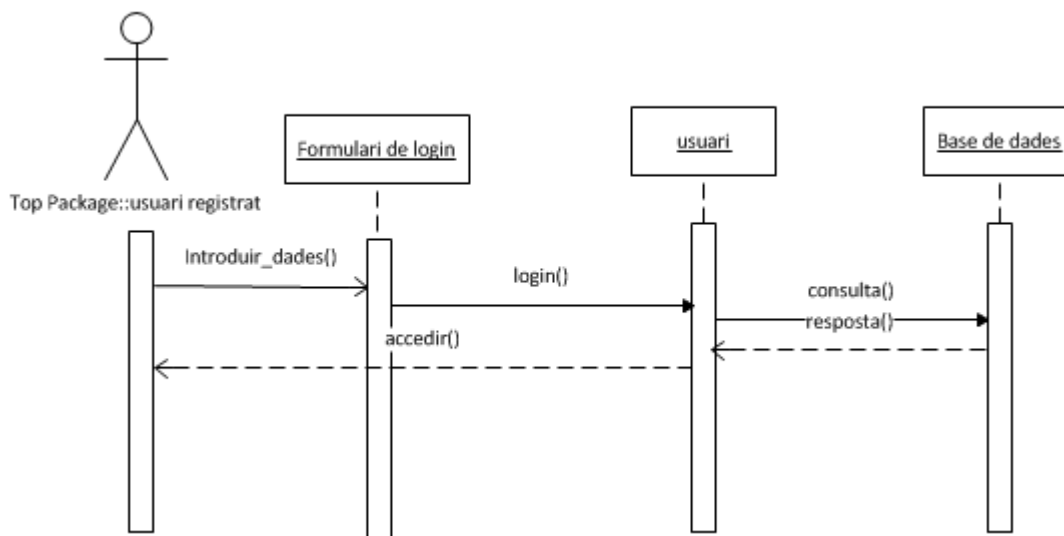


Figura 23: Diagrama de seqüència per al login d'usuaris.

Aquest diagrama descriu la seqüència que es segueix per fer el login d'un usuari. Primer s'introdueixen les dades (usuari i contrasenya) al formulari de login. Aquest crea un objecte de la classe "usuari" i crida al mètode **login()**, que consulta a la base de dades i retorna si les dades introduïdes són o no correctes i concorden.

Per últim s'inicia sessió i s'accedeix a la pàgina d'usuari.

4.4.3 Publicar article usuari

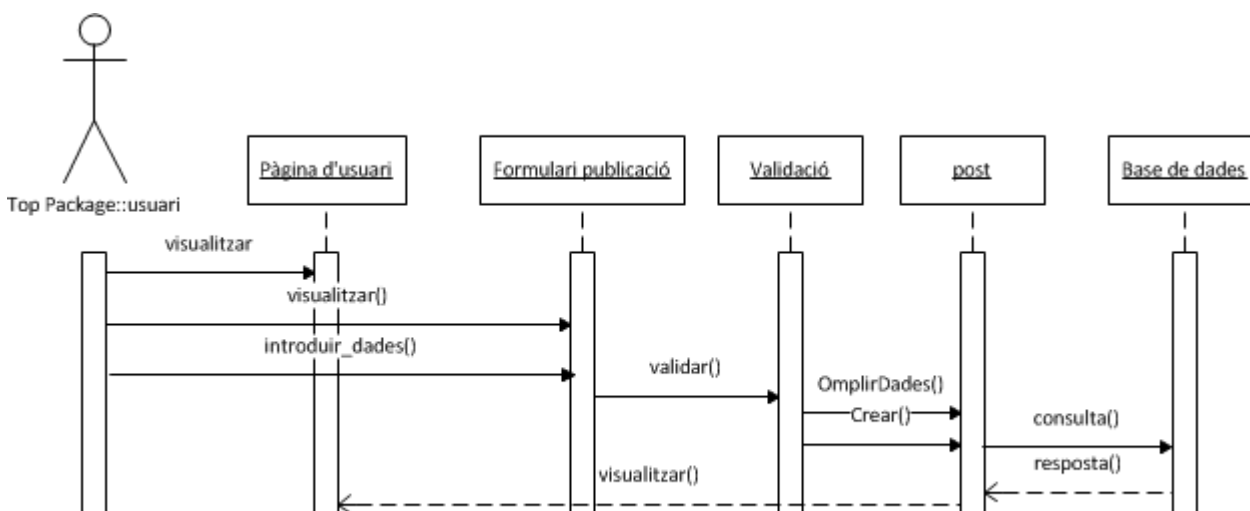


Figura 24: Diagrama de seqüència per publicar articles.

L'anterior diagrama descriu la seqüència que es segueix per a que un usuari publiqui un article a la seva pàgina de perfil. Primer l'usuari visualitza la pàgina d'usuari, després sol·licita visualitzar el formulari de publicació d'articles i introdueix les dades de l'article.

Després es valida que aquestes dades siguin correctes i si és així, es procedeix a emmagatzemar l'article a la BD. Per això, de la mateixa manera, es crea un objecte de la classe "post", es crida al mètode **OmplirDades()** i després al mètode **Crear()**.

Per últim, es torna a visualitzar la pàgina d'usuari on ha d'aparèixer aquest nou article. Aquesta seqüència és semblant per a publicar articles a pàgines de grups i assignatures.

4.4.4 Modificar dades usuari

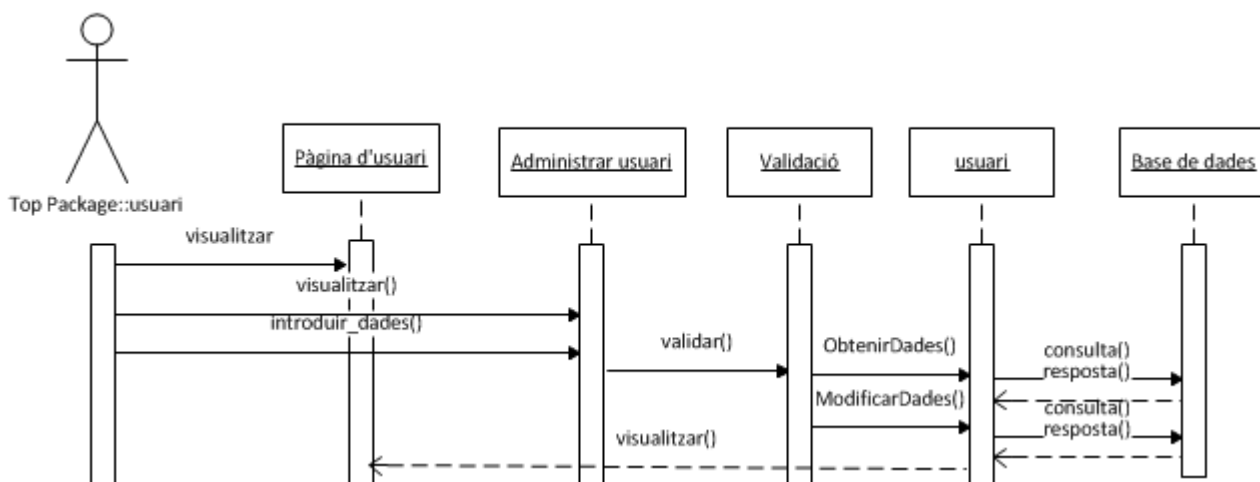


Figura 25: Diagrama de seqüència per a la modificació de dades d'usuari.

Aquest diagrama descriu la seqüència que ha de seguir un usuari per modificar les dades del seu perfil. Primer, l'usuari visualitza la seva pàgina d'usuari, després sol·licita visualitzar la pàgina d'administració d'usuari i introdueix les dades al formulari de modificació de dades. Just seguit, es valida que aquestes dades siguin correctes.

Per modificar les dades de l'usuari, a la BD, es crea un objecte de la classe "usuari" i es crida al mètode **ObtenirDades()**. Després es crida al mètode **ModificarDades()** de l'objecte "usuari", que modifica les dades de l'usuari.

Per últim es torna a visualitzar la pàgina on han d'aparèixer les noves dades que han sigut modificades.

4.4.5 Expulsar membre grup

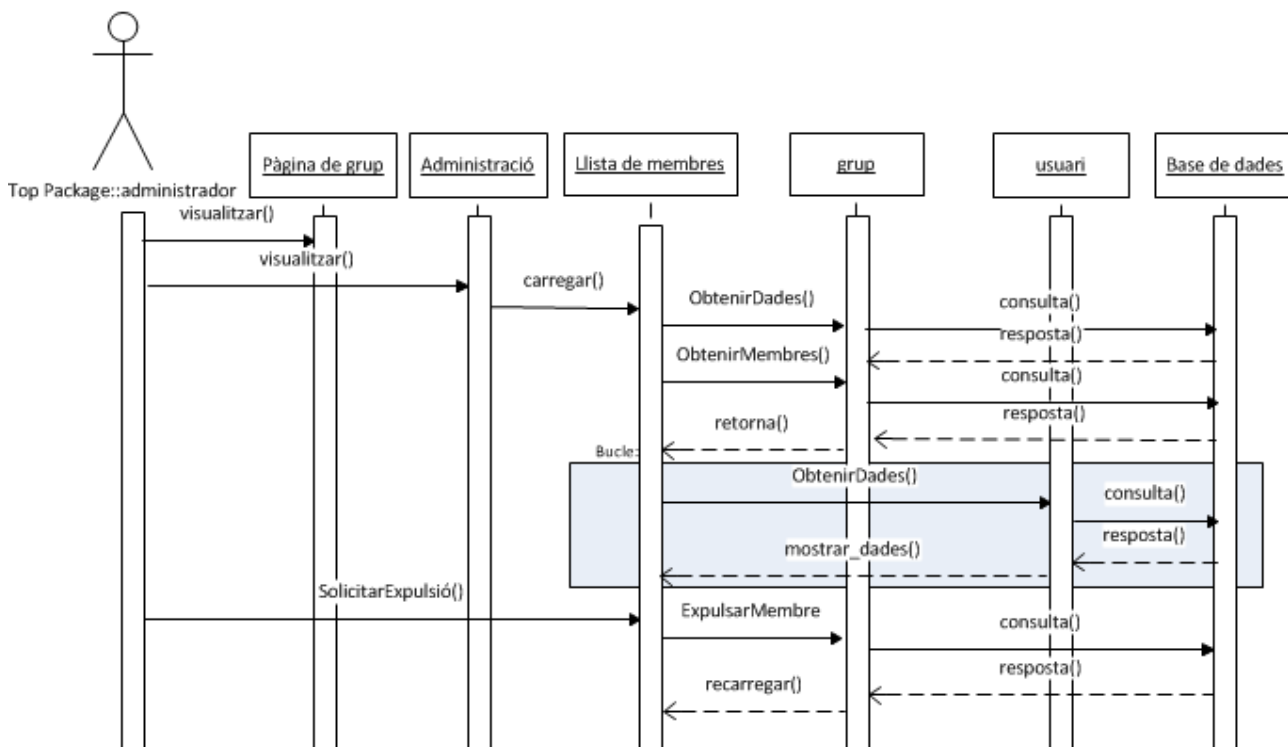


Figura 26: Diagrama de seqüència per a l'expulsió d'un membre d'un grup.

Aquest diagrama descriu la seqüència que es segueix per expulsar a un membre d'un grup. L'usuari, en aquest cas l'administrador del grup, visualitza la pàgina del grup, després sol·licita visualitzar la pàgina d'administració del grup, i en aquesta es carrega el llistat de membres del grup.

Per fer això primer es crea un objecte de classe "grup" i es crida a **ObtenirDades()**, per obtenir les dades del grup, per després cridar **ObtenirMembres()** que retorna un llistat dels membres. Després mitjançant un bucle es recorre aquest llistat, que per cada usuari membre va cridant al mètode **ObtenirDades()** que retorna les dades de l'usuari per poder mostrar-les per pantalla. Una vegada carregat el llistat de membres, l'administrador pot sol·licitar expulsar a qualsevol. Quan es fa aquesta sol·licitud es crida al mètode **ExpulsarMembre()** de l'objecte "grup", que elimina la relació de membre que té l'usuari amb el grup.

Per últim es sol·licita recarregar el llistat, on una vegada carregat no ha d'aparèixer el membre expulsat. Aquesta seqüència és similar a la que es faria en una pàgina d'assignatura.

4.4.6 Modificar dades grup

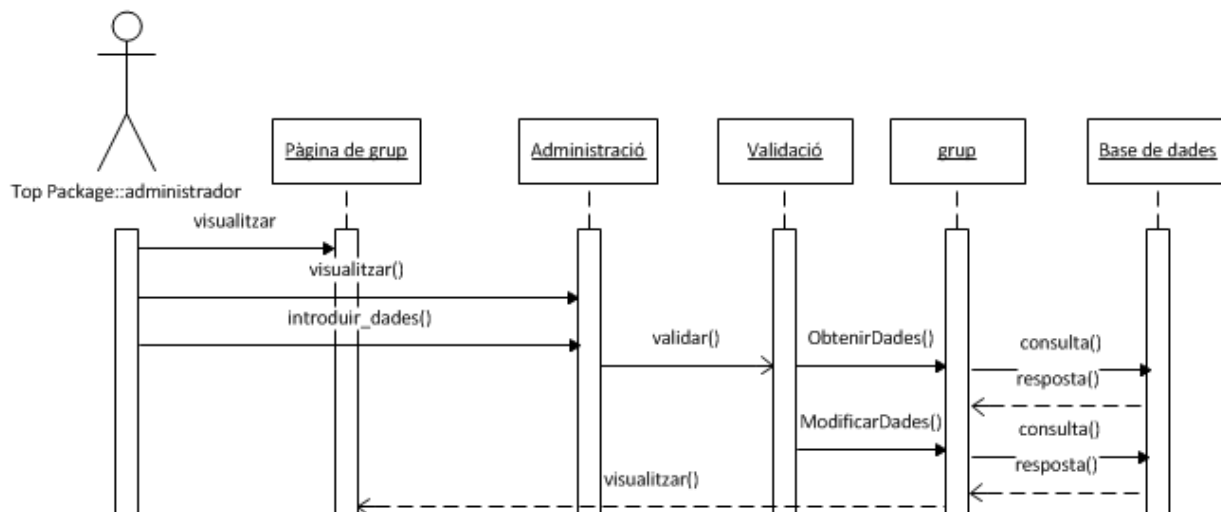


Figura 27: Diagrama de seqüència per a la modificació de dades d'un grup.

Aquesta diagrama descriu la seqüència que ha de seguir un administrador de grup per modificar les dades del grup.

L'administrador visualitza la pàgina del grup, després sol·licita visualitzar la pàgina d'administració del grup i després introdueix les dades al formulari de modificació. Just seguit, es valida que aquestes dades siguin correctes.

Per modificar les dades del grup, a la base de dades, es crea un objecte de la classe "grup" i es crida al mètode **ObtenirDades()**. Llavors es crida al mètode **ModificarDades()** que modifica les dades del grup.

Per últim es torna a visualitzar la pàgina del grup on han d'aparèixer les noves dades. Aquesta seqüència és similar a la que es faria en una pàgina d'assignatura.

4.4.7 Cercar usuaris

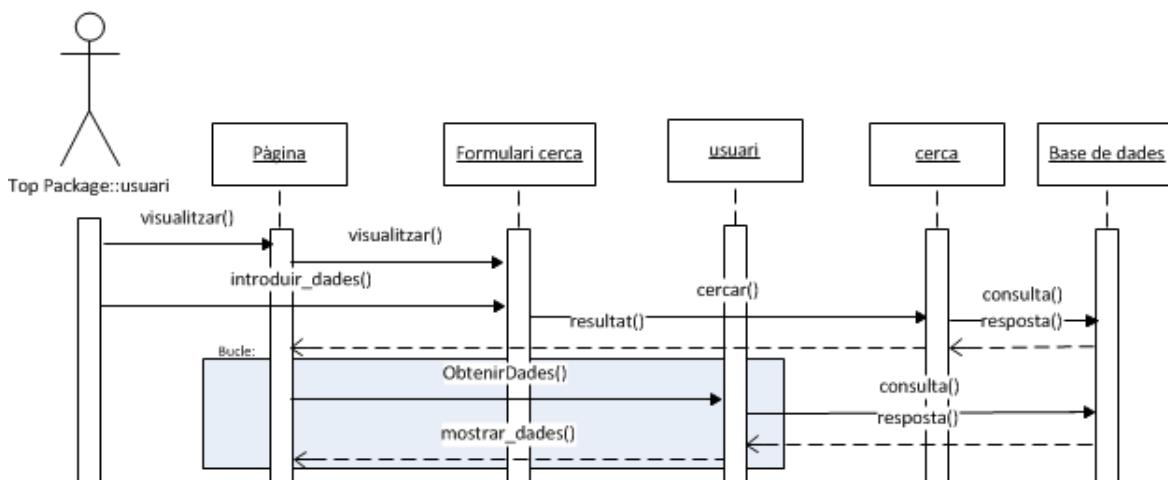


Figura 28: Diagrama de seqüència de cerca.

El diagrama anterior descriu quina seqüència es segueix per cercar usuaris.

Primerament, l'usuari visualitza la pàgina del cercador que alhora mostra el formulari de cerca.

Després l'usuari introdueix les dades a aquest formulari i al ser enviades es crea un nou objecte de la classe "cerca" i es crida al seu mètode **cercar()** amb les opcions especificades al formulari. Aquest mètode retorna un llistat amb els usuaris trobats, que són processats mitjançant un bucle, on es van mostrant les dades de tots els usuaris trobats mitjançant al mètode **ObtenirDades()** que es va cridant per a cada usuari trobat.

Aquesta seqüència és similar per a la cerca d'altres tipus d'elements.

4.4.8 Administrar amics

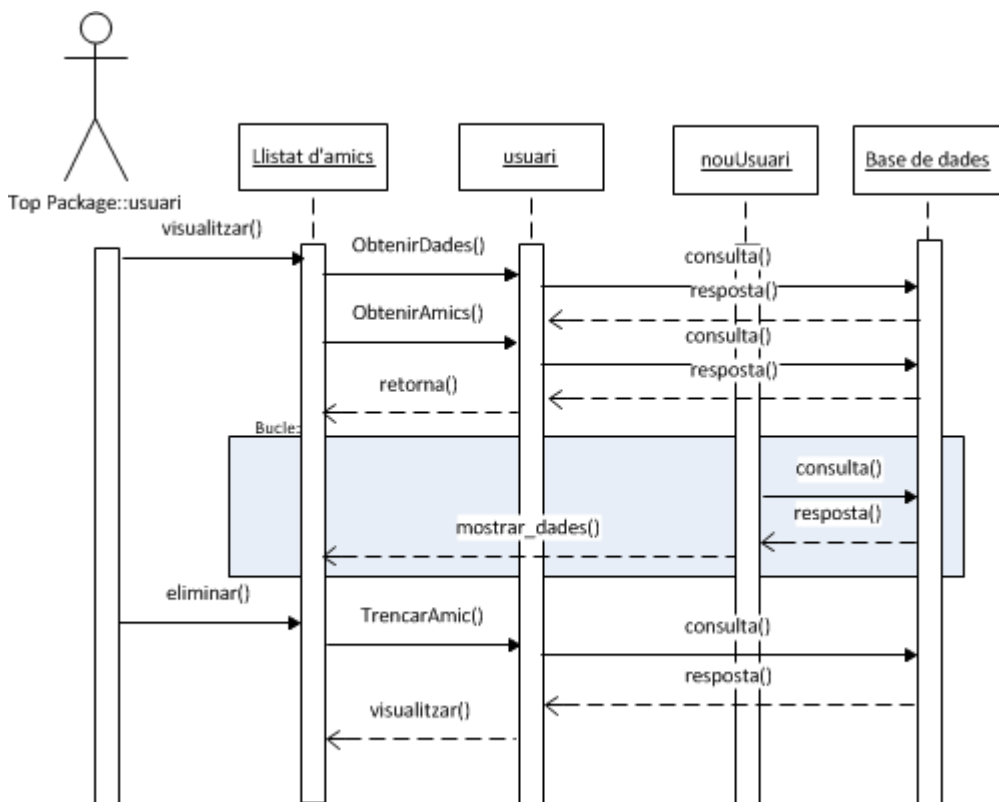


Figura 29: Diagrama de seqüència de l'administració d'amics.

El diagrama anterior descriu quina seqüència es segueix per fer l'administració d'amistats d'un usuari, en aquest cas es descriu la seqüència per eliminar a un amic.

Primerament es sol·licita visualitzar la pàgina on es mostra el llistat d'amics, després es crea un nou objecte de la classe "usuari" i es crida al mètode **ObtenirDades()**, necessari per després poder cridar **ObtenirAmics()**, que retorna un llistat de tots els amics de l'usuari.

Després mitjançant un bucle es recorre aquest llistat que per cada amic crea un altre objecte de la classe "usuari", anomenat "nouUsuari", i va cridant al mètode **ObtenirDades()**, que retorna les dades de l'usuari i que es mostren per pantalla al llistat.

Una vegada terminat el bucle i amb tot el llistat mostrat, l'usuari sol·licita eliminar a un amic, just seguit es crida al mètode **TrencarAmic()** de l'objecte "usuari", que trenca la relació d'amistat amb l'usuari seleccionat.

Per últim, una vegada eliminat es torna a visualitzar la llista d'amics on ja no ha d'aparèixer l'amic eliminat.

4.4.9 Administrar alumnes

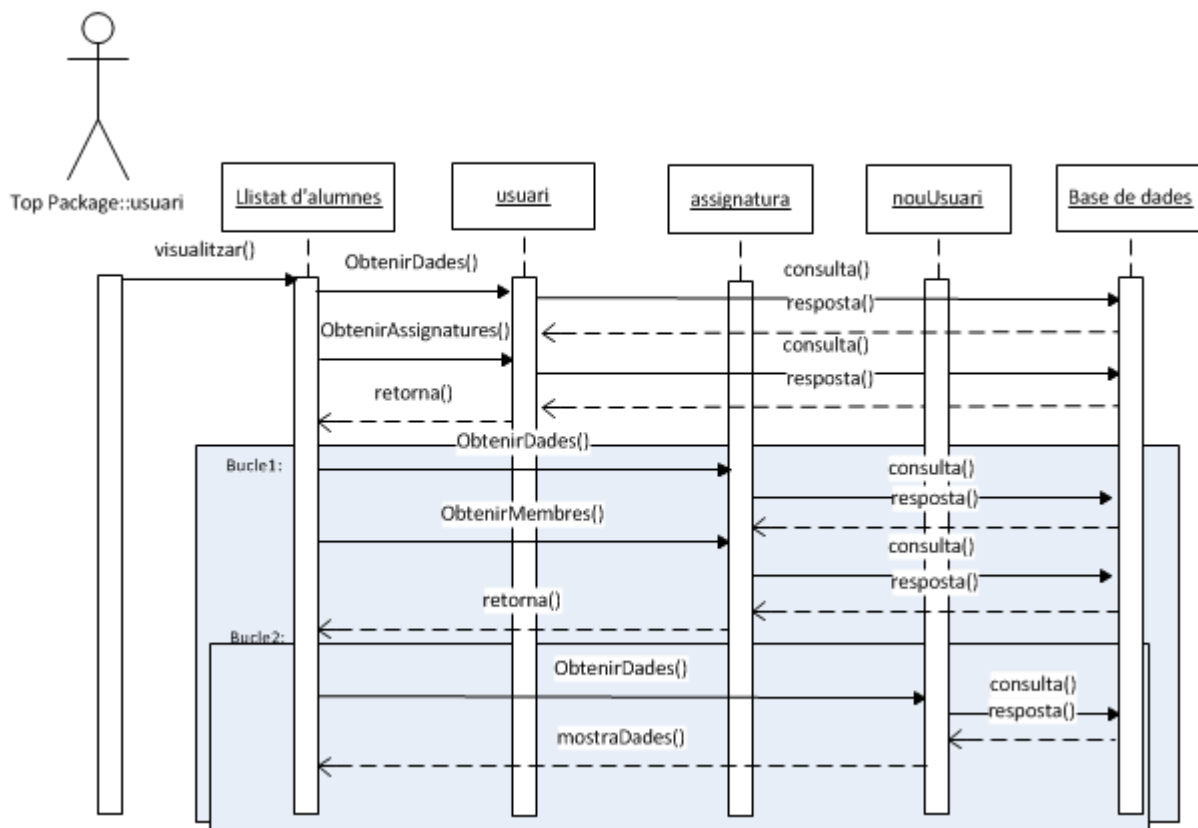


Figura 30: Diagrama de seqüència per veure el llistat d'alumnes d'un professor

L'anterior diagrama mostra la seqüència que ha de seguir un professor per visualitzar el seu llistat d'alumnes ordenats per assignatures.

Primerament l'usuari, en aquest cas el professor, sol·licita visualitzar la pàgina del llistat d'alumnes.

Seguidament, es crida al mètode **ObtenirDades()** de l'objecte de la classe "usuari". Per així obtenir les dades del professor i poder cridar després al mètode **ObtenirAssignatures()**, que retorna un llistat de les assignatures del professor.

Després, mitjançant un bucle es recorre la llista d'assignatures cridant successivament al mètode **ObtenirMembres()** de l'objecte assignatura que retorna tots els membres de cada assignatura.

Mitjançant un segon bucle niat dintre el primer es recorre la llista de membres i es van mostrant les dades dels usuaris mitjançant successives crides al mètode **ObtenirDades()**.

4.4.10 Enviar petició d'amistat

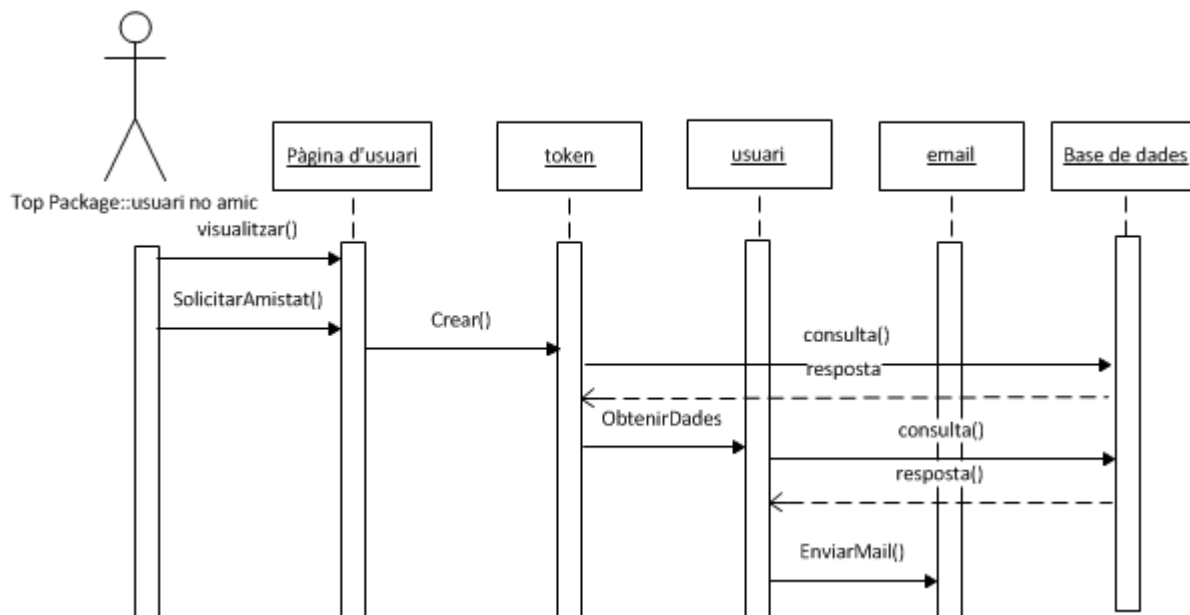


Figura 31: Diagrama de seqüència d'una sol·licitud d'amistat

El diagrama anterior mostra la seqüència que es segueix quan un usuari sol·licita una petició d'amistat a un altre.

Primerament, l'usuari visualitza la pàgina de l'usuari al que vol sol·licitar amistat, seguidament fa una sol·licitud d'amistat a aquest usuari.

Després d'això es crea un nou objecte de la classe "token" i es crida al mètode **Crear()** d'aquest. Després de crear-se el nou codi token de seguretat, es crea un nou objecte de la classe "usuari" per seguidament cridar al mètode **ObtenirDades()** d'aquest.

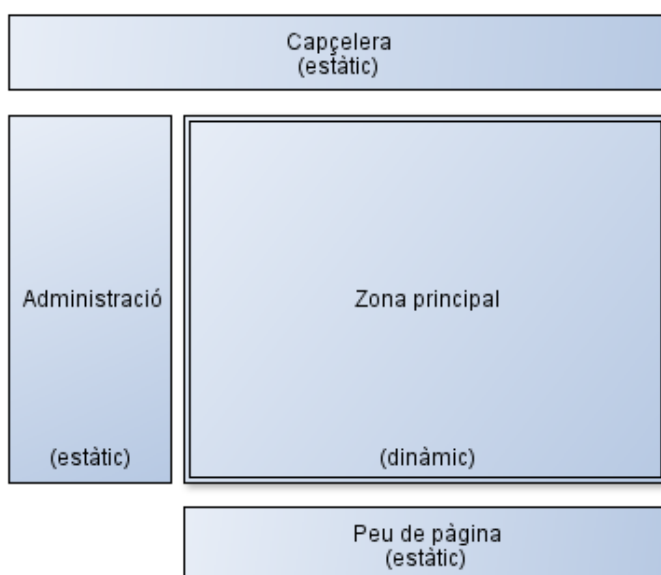
Això és necessari per obtenir totes les dades de l'usuari, i concretament el correu electrònic que és el que ens interessa.

Per últim, una vegada es té el codi token de seguretat i el correu electrònic de l'usuari, s'envia el correu electrònic a aquest amb un enllaç per acceptar aquesta petició.

4.5 Interfície

L'estructura o el disseny de la interfície s'ha dissenyat de manera es compleixin tant els requisits funcionals com els no funcionals. La interfície tindrà parts estàtiques i parts dinàmiques. Les parts estàtiques romandran tota l'estona visibles a pantalla, en canvi les parts dinàmiques, en principi, s'aniran actualitzant d'acord amb les accions que faci l'usuari.

En general, podem definir que la interfície principal seguirà el següent esquema o disseny:



a. Per tant d'afavorir la usabilitat, en tot moment serà visible un menú estàtic que permetrà les funcions d'administració d'usuari i que s'ubicarà a l'esquerra de la pàgina. Aquest menú tindrà enllaços a totes les seccions de l'aplicació.

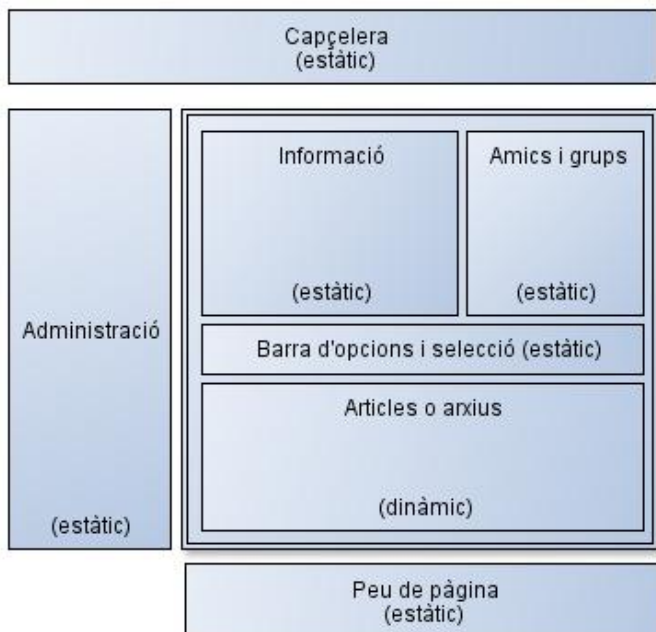
b. La capçalera inclourà el logotip de l'aplicació i un enllaç per poder tancar la sessió.

c. El peu de pàgina simplement inclourà un avís legal sobre l'aplicació i el nom de l'autor.

d. A la zona principal és on es carregaran dinàmicament totes les diferents seccions o pàgines de l'aplicació.

A continuació es mostren els dissenys de la interfície per cadascuna de les seccions.

- La pàgina d'usuari tindrà el següent disseny:



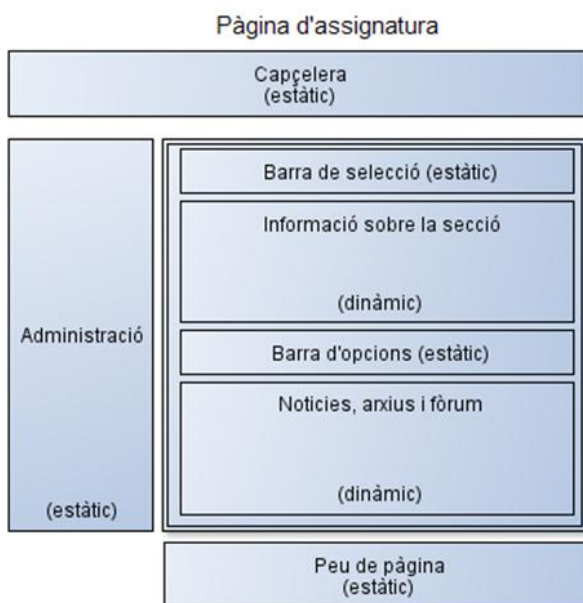
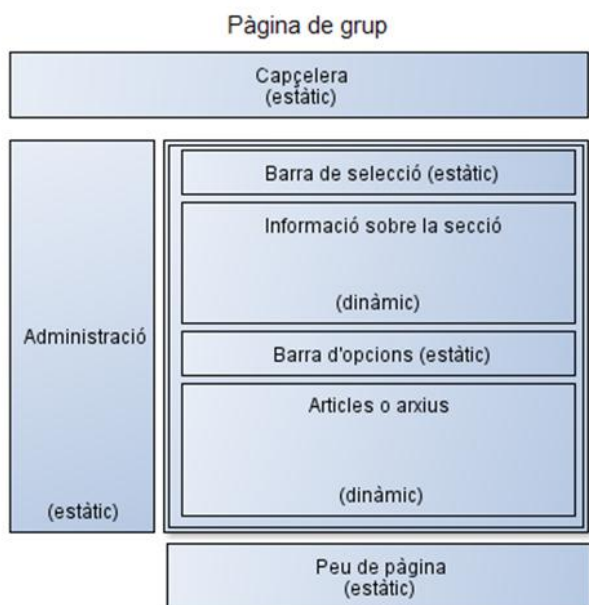
a. El bloc d'informació contindrà totes les dades d'informació sobre l'usuari així com la seva imatge de perfil.

b. Al bloc d'amics i grups es mostraran alguns, no tots, dels amics i grups de l'usuari.

c. A la barra d'opcions i selecció es mostraran les opcions de publicació d'usuari i les opcions de selecció de secció d'articles o arxius.

d. Per últim, al bloc d'articles o arxius es mostraran els articles o arxius de l'usuari segons s'hagi seleccionat a la barra d'opcions. Aquest bloc serà carregat dinàmicament i automàticament per tal d'anar mostrant els nous articles i arxius que publiqui l'usuari de forma totalment transparent a l'usuari.

- Les pàgines de grups i assignatures tindran una estructura similar que serà la següent:



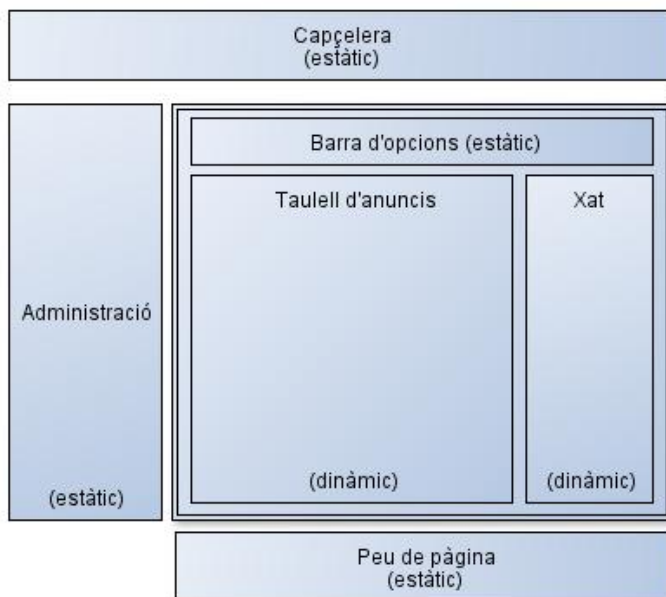
a. A la barra de selecció es mostraran les opcions per seleccionar quina secció del grup o assignatura es vol visualitzar.

b. A la barra d'opcions es mostraran les opcions per publicar articles, pujar arxius, administrar, etc.

c. Al bloc dinàmic d'informació sobre la secció es visualitzarà la informació sobre la secció carregada actualment.

d. Al bloc dinàmic d'articles o arxius (en les pàgines de grup) i el bloc de notícies, arxius i fòrum (en les pàgines d'assignatura) es carregaran els diferents elements o publicacions de la secció.

- El taulell d'anuncis tindrà el següent disseny:

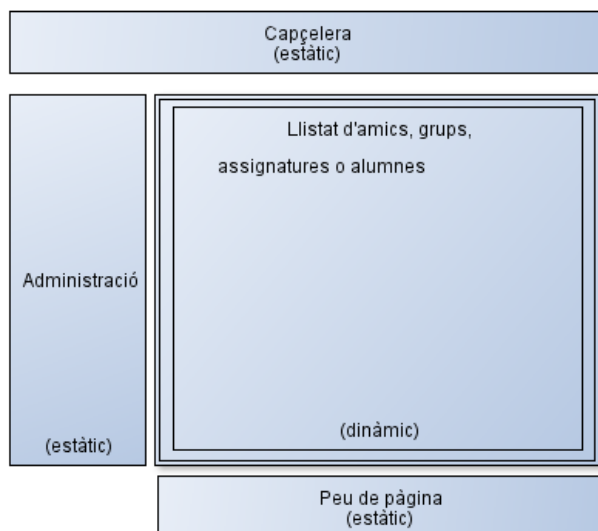


a. La barra d'opcions contindrà les opcions per publicar nous anuncis.

b. El bloc principal del taulell d'anuncis carregarà dinàmicament tots els anuncis publicats al taulell.

c. El bloc de xat contindrà el xat del sistema on l'usuari podrà comunicar-se amb la resta d'usuaris, i que serà carregat dinàmica i automàticament per tal d'anar mostrant els nous missatges de forma transparent a l'usuari.

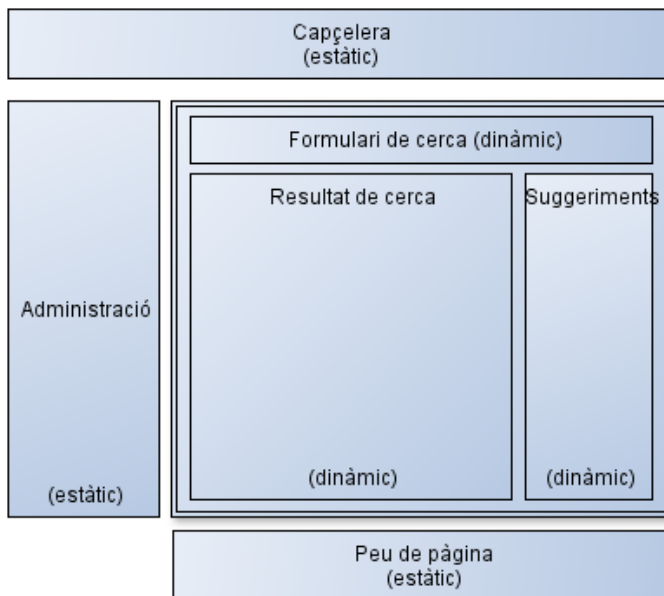
- La zona d'administració d'usuari constarà de diverses pàgines (amb la mateixa estructura mostrada a continuació) que seran carregades de forma dinàmica segons la selecció de l'usuari al menú d'administració d'usuari. Tindrà el següent disseny:



a. El bloc de llistat d'amics, grups assignatures o alumnes mostrarà un llistat d'aquests, segons la selecció de l'usuari al menú d'administració.

b. En aquest llistat l'usuari podrà realitzar diverses tasques d'administració com, per exemple, eliminar amics o sortir de grups.

- Per últim, el disseny del cercador serà el següent:



a. Al bloc de suggeriments apareixeran un màxim de 10 suggeriments que es mostraran al carregar la pàgina.

b. Al bloc del formulari de cerca apareixerà el formulari dinàmic i específic que serà enviat per fer la cerca.

c. Al bloc de resultat de cerca es mostrarà un llistat amb els resultats de la cerca.

- En altra instància, per tal de complir amb els requeriments no funcionals de l'aplicació, a la interfície se li han inclòs les següents característiques:

- S'inclourà un text al formulari de registre d'usuari, amb el compliment de la llei de protecció de dades, informant a l'usuari del seu dret d'accés, modificació i baixa de les seves dades al sistema.
- Tota la interfície estarà íntegrament en català.
- La interfície ha d'estar pensada per visualitzar-se correctament en resolucions de 1366x768 i superiors.
- Les imatges utilitzades en la interfície no han de tenir una mida excessivament gran ja que això repercutirà en el temps de càrrega de la pàgina i empitjorarà l'experiència d'usuari.

5 Implementació

5.1 Introducció

A la fase d'implementació es du a terme la codificació de l'aplicació, tot seguint el disseny que s'ha especificat abans. És important acomplir tant els requeriments funcionals com els no funcionals. A continuació expliquem el conjunt de tecnologies i eines que hem utilitzat.

5.2 Eines tecnològiques

5.2.1 Llenguatges de programació

- HTML5

HTML, acrònim de HyperText Markup Language (Llenguatge de marcat d'hipertext), és un llenguatge d'etiquetat i que és el principal i predominant en el desenvolupament d'aplicacions web. Amb HTML es programa mitjançant l'ús d'etiquetes i es permet incrustar scripts en JavaScript directament al seu codi.

L'última especificació d'HTML és l'**HTML5**, que és àmpliament suportada per tots els navegadors actuals i que s'ha utilitzat en aquesta aplicació per a desenvolupar la interfície.

- CSS

CSS, acrònim de **C**asacading **S**tyle **S**heets (Fulls d'Estil en Cascada), és un llenguatge de programació que permet descriure l'estil d'un document HTML. S'utilitza en conjunt amb HTML per determinar l'estil dels diferents blocs etiquetats amb aquest. A més, permet centralitzar tot l'estil de la pàgina en un sol fitxer, de manera que modificant aquest es pot modificar tot l'estil de la pàgina. CSS ens ha servit per aplicar estils als HTML de l'aplicació.

- PHP



PHP és un llenguatge de programació de codi obert i que és utilitzat especialment per al desenvolupament d'aplicacions web.

El principal avantatge que presenta és que és un llenguatge fàcil d'aprendre i d'utilitzar pels desenvolupadors, i a més ofereix alhora múltiples funcions avançades que permeten realitzar tasques més complexes.

El codi en PHP pot ser incrustat directament dins de fitxers HTML, amb la característica que aquest s'executa al costat del servidor i és totalment transparent al client. Quan el client fa una petició, el codi PHP s'executa primer al servidor, i aquest genera un HTML que és enviat al client. El fet de ser executat al costat del servidor ofereix molta seguretat, ja que el client no té accés al codi, com si passa amb altres llenguatges de programació utilitzats a aquesta aplicació. Una altra dada important és que des de PHP ens podem connectar al SGBD per realitzar consultes, mitjançant les funcions específiques per a MySQL que ja incorpora.

Per tant, aquest llenguatge s'ha utilitzat, en aquest projecte, principalment per a la implementació de les diferents classes especificades a l'estructura de classes, i en gran part de la interfície. La versió de PHP que s'ha utilitzat és la 5.3.8.

• JavaScript

JavaScript és l'altre llenguatge principal utilitzat en la implementació d'aquesta aplicació web. És un llenguatge que s'executa al costat del client i que és interpretat pel navegador web (client). També incorpora funcions que permeten modificar dinàmicament el contingut de la pàgina web, segons les interaccions de l'usuari o segons diferents esdeveniments i que han sigut molt útils en aquest projecte.

S'ha de tenir en compte que JavaScript s'executa després de PHP, i per tant és permet el pas d'informació, o alteració del codi, des de PHP (Servidor) a JavaScript (Client), però no a l'inrevés.

Tot i que això últim es pot resoldre utilitzant la tecnologia AJAX, que si permet el pas d'informació de JavaScript a PHP mitjançant consultes.

• AJAX

AJAX, acrònim de **A**synchronous **J**avaScript **A**nd **X**ML, és una tecnologia utilitzada en JavaScript que ens permet la comunicació en sentit Client-Servidor. Això es fa mitjançant crides, en aquest cas, a PHP (Servidor) que donen una resposta i que podem processar-la després des de JavaScript directament.

En aquesta aplicació, generalment, la transferència de dades sol·licitades al servidor es fa mitjançant objectes en format JSON (JavaScript Object Notation).

Per altra banda, l'enviament de dades des de JavaScript (Client) a PHP (Servidor) es fa mitjançant dos mètodes, que són GET i POST.

La tecnologia AJAX s'utilitza bastant en aquesta aplicació, ja que ens serveix per carregar dinàmicament els diferents blocs i pàgines de la interfície.

Així, això ens permet, per exemple, implementar la funcionalitat de recarregar automàticament els articles d'una pàgina si hi ha de nous. Això es fa mitjançant una crida al servidor cada cert temps i preguntant si hi ha nous articles. El servidor ens respon, i si hi ha de nous es recarrega el bloc d'articles per tal de mostrar-los, en cas contrari, no es fa res.

5.2.2 Llibreries utilitzades

- jQuery



jQuery és una biblioteca, o llibreria, de codi obert per JavaScript i que ens permet simplificar molt l'ús de moltes funcions en JavaScript. Així, permet simplificar la manera d'interactuar amb els documents HTML, manipular l'arbre DOM, gestionar esdeveniments, desenvolupar animacions i afegir interacció amb la tecnologia AJAX en pàgines web, comentada abans. En aquesta aplicació utilitzem la versió 1.7.2 de jQuery, que va ser publicada el 21 de març de 2012.

- jQuery UI



jQuery UI és una biblioteca per jQuery, també de codi obert, que afegeix a aquesta un conjunt de plugins, widgets, efectes i animacions per a la creació d'aplicacions web interactives. jQuery UI és utilitzada per a desenvolupar molts dels elements de la interfície d'aquesta aplicació, també s'utilitzen algun dels seus efectes, tot i que es va decidir restringir l'ús d'efectes i animacions per tal de millorar el rendiment. jQuery UI, per altra banda, permet l'ús de diferents temes, que utilitzen CSS per definir l'estil.

- Zend Framework

Zend és un framework, o conjunt de llibreries, de codi obert i utilitzat en PHP per desenvolupar aplicacions i serveis web. A aquestes llibreries, principalment s'utilitzen en aquesta aplicació per les funcions d'enviar correus electrònics a usuaris.

- `resize-class.php`

resize-class.php és una llibreria per PHP que permet redimensionar, tallar i modificar el format i la qualitat d'imatges. S'utilitza principalment per redimensionar les imatges que pugen els usuaris, per tal que tinguin les dimensions i qualitats requerides per a l'aplicació web.

- `eIRTE`

eIRTE és una llibreria JavaScript que permet utilitzar un editor WYSIWIG, acrònim de **What You See Is What You Get** (en anglés, "el que veus és el que obtens"). Aquest és un tipus d'editor de text en HTML que permet veure el resultat final d'aquest codi

HTML. Ens permet, per exemple, ficar, a més de text, imatges, enllaços, diferents estils de fonts, etc. Aquesta llibreria s'utilitza per poder ubicar un editor d'aquest tipus al formulari per publicar o editar articles a les wikis de les assignatures. El seu aspecte és el següent:

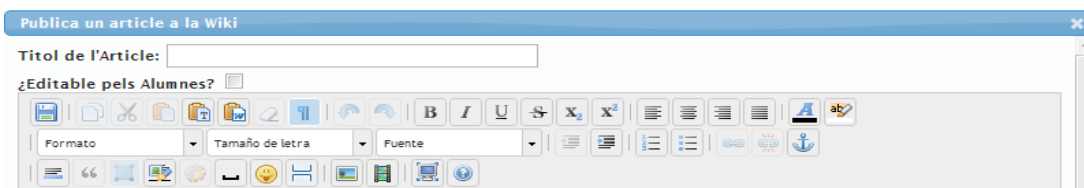


Figura 32: Editor WYSIWYG de la wiki d'una assignatura

- `Dracula Graph Library`

Dracula Graph Library és una llibreria per JavaScript que incorpora un conjunt d'eines que permeten dibuixar grafs interactius. Aquesta s'utilitza per afegir al panell d'administrador del sistema la funció de mostrar el graf general de la xarxa social. Té un aspecte com el següent:

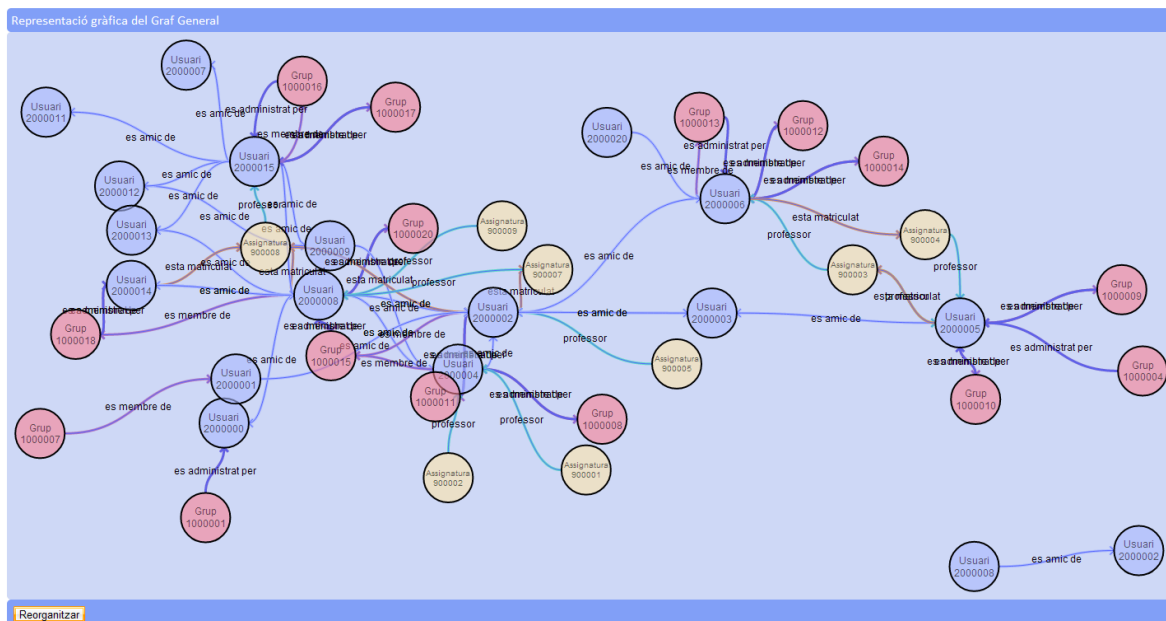


Figura 33: Diagrama interactiu del graf general

5.2.3 Entorns de programació

- **phpDesigner8**

phpDesigner8 és un entorn de programació especialment dissenyat per a la programació en PHP, tot i que també permet treballar amb altres llenguatges com HTML i JavaScript. Incorpora també un servidor web amb PHP que permet anar provant l'aplicació a mesura que es va programant.

És va decidir utilitzar aquest entorn perquè incorpora moltes funcionalitats interessants, com un quadre organitzatiu on es poden veure les classes implementades, els seus atributs i els seus mètodes. O la funcionalitat de ressaltar el codi, que permet un millor enteniment d'aquest i pot ajudar molt alhora de depurar i corregir errors.

- **Notepad++**

Notepad++ és un editor de codi obert i que suporta un conjunt de llenguatges de programació, entre ells HTML, CSS, JavaScript i PHP. Es va decidir utilitzar aquesta aplicació perquè és una aplicació lleugera, gratuïta i de codi obert, i a més incorpora la funcionalitat de ressaltar el codi.

5.2.4 Servidor web

- Apache

Apache és un servidor HTTP (o servidor web) multi plataforma i de codi obert. Permet estendre les seves funcionalitats amb la instal·lació de mòduls, com el mòdul de PHP que és el que permet l'execució de codi en PHP. Així, l'ús principal de Apache és el d'enviar pàgines web estàtiques o dinàmiques al client. Apache és el servidor web que s'utilitza en la implementació d'aquesta aplicació web, en la versió 2.2.21.

- XAMPP



XAMPP

XAMPP és un paquet d'aplicacions de programari lliure, entre les que s'inclouen el servidor HTTP Apache, la base de dades MySQL (un altre SGBD) i els mòduls necessaris per poder executar codi PHP i Perl. XAMPP s'utilitza principalment per facilitar la instal·lació d'aquestes eines en un servidor, tot i que poden ser instal·lades també de forma separada. Finalment, en el desenvolupament d'aquest projecte s'utilitzarà aquest paquet per instal·lar el servidor Apache, i mòdul PHP que incorpora, sota el sistema operatiu Microsoft® Windows XP 32 bits.

5.2.5 Sistema Gestor de Bases de Dades (SGBD)

- MariaDB

MariaDB és un SGBD (Sistema Gestor de Base de Dades), o simplement servidor de base de dades. És un fork⁶ derivat de MySQL que incorpora la funcionalitat drop-in d'aquest i a més tot un conjunt de funcions millorades, optimitzacions, nous motors d'emmagatzemament i noves actualitzacions.

Una altra cosa important és que MariaDB, a l'igual que MySQL, és de codi obert i gratuït i podem utilitzar-la lliurement. A més incorpora de sèrie el motor OQGraph, que ens serveix per emmagatzemar grafs, com ja hem comentat abans.

⁶ Un fork (paraula anglesa que significa bifurcació) és, en desenvolupament de software, la creació d'un projecte derivat d'un altre aprofitant el codi ja existent del projecte anterior.

Per altra banda, al estar basat en MySQL, té comptabilitat amb aquest i té les mateixes ordres, interfícies i llibreries. Per tant, amb les funcions per MySQL que inclou PHP 5 podem connectar perfectament amb un servidor MariaDB. Aquest és el sistema gestor de bases de dades que utilitzarem en aquest projecte.

- **OQGraph**

OQGraph, acrònim de **Open Query Graph**, és un motor de computació que permet utilitzar jerarquies i estructures de grafs més complexes en un model relacional de base de dades. És a dir, permet utilitzar taules d'una base de dades relacional, en aquest cas MariaDB, com si fossin realment grafs, i treballar amb elles mitjançant consultes amb sintaxi SQL estàndard. També incorpora la possibilitat d'aplicar directament alguns algorismes de la teoria de grafs, en concret l'algorisme de Dijkstra i el de cerca en amplada.

OQGraph ha sigut implementat com a plugin per MySQL, MariaDB 5.1 i superior, i Drizzle. I no necessita modificacions de qualsevol tipus en el servidor.

En la implementació d'aquesta aplicació s'utilitza una versió de MariaDB que ja inclou aquest motor.

6 Proves

Donada la metodologia escollida, que segueix un mètode incremental i iteratiu, la fase de proves és sempre posterior a la fase d'implementació a cada iteració. L'objectiu de les proves és comprovar que es compleixen els requisits especificats, i testejar els possibles errors. A cada iteració també s'ha de comprovar que tot el codi implementat amb anterioritat no presenta errors al afegir noves característiques.

Per donar per acabada aquesta fase ha de ser el director del projecte el que ha d'aprovar-la i poder prosseguir amb la següent iteració. Es destaquen quatre tipus de proves bàsiques que podem realitzar i que proven els components de l'aplicació a diferents nivells.

6.1 Proves d'unitat

Les proves d'unitat consisteixen en provar per separat cadascuna de les classes o parts bàsiques de l'aplicació per separat. És a dir, provar que funcionen correctament sense interacció amb altres parts, i que compleixen els requisits de l'aplicació.

Aquest tipus de prova pot ser manual o automàtica i sol consistir en petits exemples d'ús per comprovar si un mòdul falla. En aquesta aplicació les proves d'unitat van ser bastant satisfactòries, i amb poca feina de depuració.

6.2 Proves d'integració

Aquest tipus de prova consisteix en provar les diferents parts o mòduls quan s'integren en components o blocs més complexos. Es poden fer a diferents nivells des de la comunicació entre dos classes fins a la integració de tot el sistema.

A cada prova d'integració s'ha de verificar que el bloc testejat és consistent, fiable i compleix amb els requisits de l'aplicació. En la part final les proves finals d'integració es pretén abastar tot el sistema en conjunt com un sol bloc. Un dels mètodes que s'utilitzen en aquest tipus de proves és el de casos límits, que pretén provar els casos més extrems possibles per testejar si el bloc falla o no.

Pel que respecta a aquesta aplicació, les proves d'integració van donar bastants problemes i van haver d'obligar, moltes vegades, a fer una regressió i modificar blocs més petits que abans funcionaven correctament.

6.3 Proves de regressió

Aquest tipus de proves tenen lloc quan un mòdul falla una prova d'integració, havent de baixar de nivell per modificar blocs més inferiors i tornar a testejar-los. A més s'ha de comprovar que tots els blocs o mòduls modificats no afecten a la resta de mòduls que abans funcionaven correctament. Una vegada depurats els errors s'han de verificar que aquests no es tornen a produir una altra vegada.

6.4 Proves d'acceptació

Les proves d'acceptació són proves que es realitzen per part del client, en aquest cas pel mateix analista-programador. Tracten de proves funcionals sobre el sistema ja completat i finalitzat i que busquen que s'acompleixin tots els requeriments que s'esperaven de l'aplicació.

Aquestes proves es realitzen en l'última fase i després de les pertinents proves d'integració, ja que es pressuposa que a aquestes arriba un sistema en una versió final i amb pocs errors.

S'ha de destacar que aquest tipus de proves van donar pocs problemes a la nostra aplicació, en part, perquè molts dels problemes es van depurar anteriorment en les proves d'integració i regressió.

6.5 Eines tecnològiques

A continuació descrivim el conjunt d'eines que es van utilitzar en el procés de proves de l'aplicació:

- Chrome Developers Tools

Les eines per a desenvolupadors de Chrome ja venen integrades en el navegador Google Chrome. Es componen d'un conjunt d'eines entre les quals es destaquen un visualitzador d'elements, que permet veure quina part del codi HTML correspon a un element gràfic

seleccionat. I també una consola de comandes JavaScript, que permet provar, en temps real, funcions, visualitzar variables o qualsevol altra cosa que permeti fer Javascript. Aquesta eina ha estat molt útil i ha permès, sobretot, anar testejant la interfície gràfica.

- **HeidiSQL**

HeidiSQL és un client gràfic per MariaDB, que permet visualitzar les seves taules i realitzar consultes entre d'altres i de forma gràfica, comunicant-se directament amb el SGBD. La utilització d'aquest software ha estat molt útil en procés d'implementació de l'estructura de classes que realitzen una connexió amb la BD. Ja que permetia veure l'estat de la base de dades, i veure si les accions fetes des de PHP modificaven, inserien o eliminaven correctament el que els pertocava. Aquest software ve inclòs per defecte amb el servidor MariaDB.

- **Navegadors web**

L'aplicació ha estat provada, sota Microsoft® Windows 7 i Ubuntu 11.10, amb els següents navegadors web:

- Google Chrome 19.0.1084.52
- Mozilla Firefox 12.0
- Opera 11.64

Principalment s'ha provat amb el navegador Google Chrome mentre s'anava desenvolupant, i ocasionalment amb la resta Mozilla i Opera han mostrat resultats satisfactoris a l'igual que Chrome.

7 Conclusions

- **Conclusió final**

L'aplicació compleix amb tots els objectius crítics i prioritaris que es van marcar inicialment, encara que l'augment de temps i de recursos que es van requerir durant el desenvolupament, i la consegüent i inevitable desviació de la planificació original, han fet que no s'hagin assolit alguns dels objectius considerats com a secundaris.

Ha estat necessari prioritzar els objectius que es consideraven primordials per al projecte i que aquests acomplissin amb una qualitat suficient, pel que respecta al nombre d'errors de l'aplicació. Els objectius secundaris eren els objectius que hagués estat bé assolir i que no afectaven críticament a la resta del projecte. En aquest cas els objectius no acomplits podrien considerar-se com funcions de valor afegit a l'aplicació, i no com una part necessària d'aquesta, per tant, es considera que no afecten notòriament a l'experiència de l'usuari. Per altra banda, queda oberta la possibilitat d'una futura implementació en futures ampliacions.

Per tant, com a conclusió final es pot dir que s'ha desenvolupat una aplicació de qualitat que compleix els requeriments establerts inicialment.

- **Planificació inicial vs. planificació final**

Al llarg del desenvolupament del projecte han sorgit tot un conjunt de problemes que han fet que no es pogués complir la planificació inicial.

Les proves van suposar un revés en el desenvolupament i van requerir més temps del calculat a l'estudi de viabilitat, ja que no totes van resultar satisfactòries com s'esperava, i consegüentment es va requerir més temps també d'implementació, per tal de solucionar els errors que s'anaven trobant a les proves. A continuació tenim la taula de tasques original on es pot veure la duració estimada de cada tasca, que es va calcular al pressupost, i la duració real que ha suposat la seva realització:

	Nombre de tarea	Duración estimada	Duración real
1	☐ Estudi de viabilitat	1,13 días	1,13 días
2	Estudi de viabilitat	4 horas	4 horas
3	Aprovació Estudi de viabilitat	1 hora	1 hora
4	Pla de projecte	3 horas	3 horas
5	Aprovació de pla de projecte	1 hora	1 hora
6	☐ Anàlisi de requeriments	3 días	3 días
7	Anàlisi de requisits funcionals i no funcionals	12 horas	12 horas
8	Anàlisi de dades	5 horas	5 horas
9	Anàlisi de segurat i legalitat	1 hora	1 hora
10	Documentació	5 horas	5 horas
11	Aprovació de l'anàlisi	1 hora	1 hora
12	☐ Disseny de l'aplicació	3,25 días	3,25 días
13	Disseny de la base de dades	5 horas	5 horas
14	Disseny dels diferents mòduls o pàgines	15 horas	15 horas
15	Disseny de la interfície	3 horas	3 horas
16	Documentació	2 horas	2 horas
17	Aprovació del disseny	1 hora	1 hora
18	☐ Implementació de l'aplicació	19,88 días	22,12 días
19	Instal·lació del SGBD	2 horas	2 horas
20	Implementació de la BD	10 horas	10 horas
21	Instal·lació del servidor (XAMPP)	2 horas	2 horas
22	Desenvolupament dels mòduls de connexió amb la BD	60 horas	67 horas
23	Desenvolupament de la interfície d'usuari	80 horas	90 horas
24	Aprovació de la implementació	4 horas	6 horas
25	☐ Proves	1,5 días	2,6 días
26	Unitat	3 horas	5 horas
27	Integració	3 horas	8 horas
28	Regressió	3 horas	6 horas
29	Acceptació	3 horas	2 horas
30	Documentació del projecte	20 horas	20 horas

Taula 14: Comparació del temps estimat amb el temps real.

De la taula anterior es pot extreure que el projecte ens ha suposat un temps de treball de **278 hores**, que són **29 hores més** de les estimades, a les quals corresponen 19 hores més d'implementació i 10 hores més de proves. D'aquestes 29 hores més de treball, l'analista-programador ha treballat 27,2 hores més i el director només 2. Això ens suposa unes pèrdues econòmiques per valor de **916 €** respecte el pressupost inicial, i que haurem d'afrontar. Hem d'aclarir que aquesta desviació en la planificació ha sigut inevitable, ja que era necessària si es volien acomplir els objectius crítics del projecte. Per altra banda, la possibilitat d'una planificació optimista i la manca de temps o de recursos, ja s'havia considerat a l'avaluació de riscos feta a l'estudi de viabilitat. La seva conseqüència és l'augment tant de temps com de recursos, i les pèrdues econòmiques que això suposa. Per tant, són riscos que ja havien estat considerats i s'ha actuat d'acord amb el pla de contingència que es va plantejar per aquests.

● Possibles ampliacions

En una aplicació d'aquestes característiques sempre hi ha lloc per millorar i per ampliar les funcionalitats. A continuació s'ha fet un llistat de noves característiques i funcionalitats que es podrien afegir en futures versions, on hi ha algunes que es van considerar inicialment però que es van descartar perquè requerien d'un temps i d'uns recursos dels que no es disposava, i d'altres que es van suggerir durant el desenvolupament. Són les següents:

- Poder fer enquestes a les assignatures i grups.
- Que els professors puguin realitzar test autoavaluats a les assignatures.
- Crear una secció de "Dubtes al professor" a les assignatures on els alumnes puguin plantejar dubtes i els professors respondre'ls.
- Crear fòrums públics de dubtes i altres temàtiques.
- Crear una aplicació per a *smartphones* on es pugui accedir a les funcionalitats més bàsiques de l'aplicació.
- Permetre als usuaris canviar les seves imatges de perfil i les imatges dels grups.
- Implementar un sistema de notificacions de noves publicacions.
- Implementar una pàgina inicial on l'usuari pugui veure les novetats que han succeït al seu entorn (amics, grups, assignatures, etc)
- Afegir una nova secció a les assignatures on els alumnes puguin entregar treballs demanats pel professor.
- Traduir l'aplicació al castellà i altres idiomes.
- Implementar una funcionalitat que permeti a l'usuari modificar l'aspecte de la interfície de l'aplicació.

- Valoració personal

Personalment considero que aquest projecte ha estat un repte personal, que m'ha aportat bastants coneixements tècnics i molts quant a gestió de projectes. I, sobretot, la importància que té una bona gestió en qualsevol projecte i en totes les fases d'aquest.

Durant el meu pas per la universitat ha sigut habitual el fet de començar directament a programar als projectes que es demanaven a algunes assignatures. Amb el pas del temps, m'he donat compte, que en projectes d'una certa complexitat, fer això pot comportar moltes dificultats i un fracàs total del projecte.

Amb aquest projecte he après que alhora de realitzar un projecte s'ha de tenir molt clar al principi el què es vol fer, i enfocar-lo cap a un lloc en concret. Ja que intentar abastar molt donarà sempre un projecte incomplet en general, que estarà enfocat a moltes coses, però que no aprofundirà en cap d'elles. La realització d'aquest projecte m'ha servit per veure, també, que una bona planificació inicial i un bon disseny conceptual després, porten a què alhora de programar tot sigui més fàcil, ja que està ben definit tot el que s'ha de fer en cada moment.

Pot haver-hi desviacions en la planificació, és clar, però en general surt tot com s'havia previst. I també, que és important saber decidir com resoldre les incidències i dificultats que es presenten durant el desenvolupament.

Estic satisfet amb el resultat final del projecte i considero que ha estat una bona experiència, culminació final de tots els coneixements apresos durant la carrera, i que m'ha permès veure les dificultats i el temps de dedicació que requereixen els projectes d'aquestes característiques. Estic convençut que tot el que he après em serà molt útil com a primer pas a la meva vida professional.

8 Bibliografia

• Referències i tutorials:

1. <http://es.wikipedia.org/wiki/>: S'ha consultat aquesta enciclopèdia per comprendre molts dels conceptes descrits a aquesta memòria.
2. <http://www.desarrolloweb.com>: Pàgina web que inclou molts tutorials per al desenvolupament de pàgines web.
3. <http://www.graphdracula.net/documentation/>: Documentació de la llibreria Dracula Graph Library.
4. <http://www.maestrosdelweb.com>: Pàgina web amb molts tutorials de desenvolupament web.
5. <http://openquery.com/graph-computation-engine-documentation>: Documentació del motor OQGraph.
6. <http://net.tutsplus.com/tutorials/>: Pàgina web amb molts tutorials de desenvolupament web, sobretot de PHP.
7. <http://php.net/>: Manual de referència de PHP.
8. <http://www.pedroantoniioduran.es/2011/06/13/enviar-un-correo-con-zend-framework-utilizando-zend-mail-y-gmail/>: Tutorial per enviar correus electrònics amb Zend per PHP.
9. <http://dev.mysql.com/doc/>: Documentació de MySQL, que ha servit també per MariaDB.
10. http://www.davidbayon.net/index.php?mostrar=posts&post_id=91: Article que conté una font amb tots els municipis d'Espanya en format SQL i que han estat inclosos a la base de dades d'aquesta aplicació. És propietat de David Bayón Esporrín i l'autor dóna permís per reproduir el seu contingut sota una llicència de Creative Commons.
11. <http://www.phpfreaks.com/tutorial/php-security>: Tutorial de seguretat en PHP.
12. http://www.freepik.es/foto-gratis/cloudscape-ilustracion_559079.htm: Imatge que s'utilitza per al fons de la capçalera de l'aplicació, és lliure per a ús no comercial.
13. <http://www.elgg.org>: Pàgina amb informació sobre l'alternativa elgg.
14. <http://www.moddle.org>: Pàgina amb informació sobre l'alternativa moddle.
15. <http://docs.jquery.com>: Documentació i exemples d'ús de la llibreria jQuery.
16. <http://jqueryui.com/demos/>: Documentació i exemples d'ús de la llibreria jQuery UI.
17. <http://www.kickbill.com/?p=1132>: Tutorial de com validar el format d'una data en PHP.
18. <http://es2.php.net/rmdir>: Algorisme per eliminar un directori de forma recursiva.
19. <http://www.jqueryin.com/2010/04/30/a-quick-implementation-of-string-sort-in-php/>: Algorisme QuickSort en PHP.

Tots els enllaços han estat comprovats a data 22 de juny de 2012

• Llibreries de software

S'han utilitzat els següents components de software i, en alguns casos, s'han modificat d'acord amb les necessitats del projecte:

1. **XAMPP** (<http://www.apachefriends.org/es/xampp.html>)

Llicència: *XAMPP és una compilación de software libre (comparable a una distribución Linux), és gratuito y libre para ser copiado conforme los términos de la licencia **GNU General Public License**.*

2. **MariaDB** (<http://mariadb.org/>)

Llicència: *MariaDB is based on MySQL and is available under the terms of the **GPL v2 license**.*

3. **jQuery** (<http://jquery.org/license>)

Llicència: *You may use any jQuery project under the terms of either the MIT License or the **GNU General Public License (GPL) Version 2**.*

4. **jQuery highlight v3** (<http://johannburkard.de/blog/programming/javascript/highlight-javascript-text-highlighting-jquery-plugin.html>)

Llicència: Johann Burkard. MIT licenses.

5. **Sizzle** (Inclòs en la llibreria jQuery)

Llicència: *The Sizzle selector engine (which is included inside the jQuery library) is held by the Dojo Foundation and is licensed under the **MIT, GPL, and BSD licenses**.*

6. **jQueryUI** (<http://www.jqueryui.com>)

Llicència: *The jQuery UI library is currently available for use in all personal or commercial projects under both **MIT and GPL licenses**.*

7. **resize-class.php** (<http://net.tutsplus.com/tutorials/php/image-resizing-made-easy-with-php/>)

S'ha utilitzat el codi font que es presenta en l'anterior tutorial. Autor: Jarrod Oberto

Llicència: You grant a license to readers to use the knowledge and demonstrated effects shown in the tutorial in their own projects without reference to yourself, Envato or this site. They may not reproduce the tutorial itself, but they can use the techniques you teach.

8. **eIRTE** – Editor WYSIWYG (<http://elrte.org/es/>)

Llicència: *eIRTE se distribuye con la **licencia BSD***

*eIRTE utiliza y se distribuye junto con jQuery y jQuery-UI que se distribuyen con una licencia doble (**MIT y GPL**)*

9. **Chat** (<http://net.tutsplus.com/tutorials/javascript-ajax/how-to-create-a-simple-web-based-chat-application>)

S'ha seguit l'anterior tutorial per desenvolupar un chat a mida d'acord amb les necessitats del projecte.

Llicència: *You grant a license to readers to use the knowledge and demonstrated effects shown in the tutorial in their own projects without reference to yourself, Envato or this site. They may not reproduce the tutorial itself, but they can use the techniques you teach.*

10. **Dracula Graph Library** (<http://www.graphdracula.net>)

Llicència: *This code is freely distributable under the **MIT license**. Commercial use is hereby granted without any cost or restriction.*

11. **Zend Framework** (<http://framework.zend.com/>)

Llicència: *BSD.*

Tots els enllaços han estat comprovats a data 22 de juny de 2012

Relació de figures

Figures:

Figura 1: Descripció física del sistema	9
Figura 2: Tipus d'amistat	11
Figura 3: Diagrama WBS	27
Figura 4: Diagrama de Gantt	29
Figura 5: Diagrama de casos d'ús de la interfície d'inici	39
Figura 6: Diagrama de casos d'ús de les pàgines d'usuari.	40
Figura 7: Diagrama de casos d'ús de les pàgines de grup.	41
Figura 8: Diagrama de casos d'ús de les pàgines d'assignatura.	43
Figura 9: Diagrama de casos d'ús del taulell d'anuncis.	44
Figura 10: Diagrama de casos d'ús del xat.	45
Figura 11: Diagrama de casos d'ús de la cerca.	45
Figura 12: Diagrama de casos d'ús de l'administració d'usuari.	46
Figura 13: Diagrama de casos d'ús del sistema de peticions.	47
Figura 14: Diagrama de casos d'ús de l'administració general.	48
Figura 15: Exemple d'un graf amb arestes no dirigides i sense pes.	50
Figura 16: Diagrama UML inicial de la base de dades.	52
Figura 17: Exemple del graf general de la xarxa	56
Figura 18: Exemple d'un graf de grup.	58
Figura 19: Diagrama de classes de l'aplicació.	60
Figura 20: Exemple de seqüència d'execució per crear un nou usuari.	62
Figura 21: Exemple de seqüència d'execució per obtenir les dades d'un usuari.	63
Figura 22: Diagrama de seqüència per al registre d'usuaris.	74
Figura 23: Diagrama de seqüència per al login d'usuaris.	75
Figura 24: Diagrama de seqüència per publicar articles.	75
Figura 25: Diagrama de seqüència per a la modificació de dades d'usuaris.	76
Figura 26: Diagrama de seqüència per a l'expulsió d'un membre d'un grup.	77
Figura 27: Diagrama de seqüència per a la modificació de dades d'un grup.	78
Figura 28: Diagrama de seqüència de cerca.	79
Figura 29: Diagrama de seqüència de l'administració d'amics.	80
Figura 30: Diagrama de seqüència per veure el llistat d'alumnes d'un professor	81
Figura 31: Diagrama de seqüència d'una sol·licitud d'amistat	82
Figura 32: Editor WYSIWYG de la wiki d'una assignatura	91
Figura 33: Diagrama interactiu del graf general	92

Taules:

Taula 1: Parts interessades.	8
Taula 2: Objectius crítics, prioritàris i secundaris del desenvolupament.	16
Taula 3: Cost del material.	19
Taula 4: Cost del personal	20
Taula 5: Cost d'amortització.	20
Taula 6: Taula de fases i activitats.	26
Taula 7: Taula de tasques per al desenvolupament del projecte.	28
Taula 8: Catalogació de riscos	30
Taula 9: Pla de contingència	30
Taula 10: Taula d'identificadors d'usuaris, grups i assignatures.	55
Taula 11: Taula de codificació de les relacions entre dos nodes ("grafgeneral").	56
Taula 12: Taula d'identificadors dels elements dins una pàgina d'usuari, grup o assignatura.	57
Taula 13: Taula de codificació de les relacions entre dos nodes en una pàgina d'usuari, grup o assignatura.	58
Taula 14: Comparació del temps estimat amb el temps real.	99