

---

# **Repositori de records**

## **Recoly - Webapp amb tecnologia HTML5, CSS3 i JavaScript**

**Sergio Cantón Gavilán**  
Enginyeria Informàtica

Consultor: **Jordi Ceballos Villach**

Juny de 2012

*A tots aquells que m'han fet costat al llarg de tots aquests anys.*

*Als que hi sou i sempre hi sereu, i als que ja han marxat.*

*A tots ells, gràcies.*

## Índex de continguts

1. Introducció.....	6
2. Motivació del projecte.....	7
3. Planificació.....	8
3.1 Fase de planificació, anàlisi de requeriments i especificació.....	9
3.2 Fase d'estudi comparatiu i disseny.....	9
3.3 Fase d'implementació i verificació.....	10
3.4 Temporització del projecte.....	11
3.5 Eines utilitzades.....	13
3.6 Productes obtinguts.....	13
4. Estudi comparatiu.....	14
4.1 Definició de framework.....	14
4.2 Frameworks per al desenvolupament d'aplicacions web mòbils.....	14
4.3 Avantatges.....	15
4.4 Criteris de selecció.....	15
4.5 Selecció dels frameworks per l'estudi comparatiu.....	16
4.6 Metodologia.....	18
4.7 Característiques a comparar.....	18
4.8 Selecció del millor framework.....	19
4.9 Resum de l'anàlisi comparatiu.....	20
4.10 Conclusions.....	22
5. Anàlisi.....	23
5.1 Introducció.....	23
5.2 Justificació.....	24
5.3 Requeriments.....	25
5.3.1 Usuaris de l'aplicació.....	25
5.3.2 Requeriments funcionals.....	25
5.3.3 Requeriments no funcionals.....	26
5.4 Funcionalitats de l'aplicació.....	27
5.5 Riscos.....	28
5.6 Diagrama de Casos d'ús.....	29
5.7 Especificació de casos d'ús.....	30
5.7.1 CU01 – Crear record.....	30
5.7.2 CU02 – Publicar contingut.....	30
5.7.3 CU03 – Cerca record.....	31
5.7.4 CU04 – Llista records.....	31
5.7.5 CU05 – Enviar petició publicació.....	32
5.7.6 CU06 – Llegir notificacions.....	32
5.7.7 CU07 – Acceptar petició publicació.....	33
5.7.8 CU08 – Rebutjar petició publicació.....	33
6. Disseny.....	34

6.1	Arquitectura.....	34
6.2	Vistes de l'aplicació.....	34
6.2.1	Vista física.....	34
6.2.2	Vista lògica.....	36
6.2.3	Vista de components.....	37
6.3	Decisions de disseny.....	38
6.4	Diagrama estàtic.....	39
6.5	Diagrama de seqüència.....	41
6.6	Disseny de la persistència.....	43
6.6.1	Model relacional de la base de dades.....	43
6.6.2	Diagrama de la base de dades.....	44
6.7	Bones pràctiques de disseny d'interfícies mòbils.....	45
6.8	Prototips de la interfície d'usuari.....	47
7.	Implementació.....	49
7.1	Controladors.....	49
7.2	Models.....	50
7.3	Vistes.....	51
7.4	Autenticació d'usuari.....	52
7.5	Darreres publicacions.....	54
7.6	Nou Record.....	56
7.7	Els meus records.....	58
7.8	Cerca de records.....	59
8.	Línies futures.....	62
9.	Conclusions.....	63
10.	Glossari.....	64
11.	Fonts d'informació.....	65

## Índex de figures

Fig 1: Esquema de fases del cicle de vida clàssic.....	8
Fig 2: Diagrama gantt amb la temporització del projecte.....	12
Fig 3: Diagrama de casos d'ús.....	29
Fig 4: Escenari de desplegament.....	35
Fig 5: Esquema webapp amb MVC.....	36
Fig 6: Diagrama de components.....	37
Fig 7: Diagrama estàtic.....	40
Fig 8: Diagrama de seqüència del CU01.....	41
Fig 9: Diagrama E/R.....	44
Fig 10: Teclat virtual qwerty a un terminal amb Android.....	45
Fig 11: Mockups del prototip d'interfícies.....	48
Fig 12: Controladors de l'aplicació.....	50
Fig 13: Models de l'aplicació.....	51
Fig 14: captura formulari inici sessió.....	53
Fig 15: captura formulari inici sessió amb error.....	53
Fig 16: Captura llista publicacions.....	54
Fig 17: Captura comentaris publicació.....	54
Fig 18: Captura menú principal aplicació.....	55
Fig 19: Captura nou record.....	56
Fig 20: Captura nova publicació.....	56
Fig 21: Captura publicació creada.....	57
Fig 22: Captura publicació comentada per usuari.....	57
Fig 23: Captura llista records.....	58
Fig 24: Captura informació del record en detall.....	58
Fig 25: Captura cerca de records.....	59
Fig 26: Captura resultats de cerca.....	59
Fig 27: Captura sol·licitud participació.....	60
Fig 28: Captura Sol·licitud participació enviada.....	60
Fig 29: Captura resposta a sol·licitud.....	61
Fig 30: Captura safata de notificacions.....	61

## 1. Introducció

L'accés a serveis web tradicionals des d'ordinadors personals s'ha vist eclipsat amb l'aparició de dispositius mòbils com smartphones, pda's, tablets, etc., que ofereixen accés als continguts de la xarxa des de qualsevol ubicació. Aquesta darrera revolució neix al voltant de l'any 2007 amb un fort creixement de dispositius mòbils com smartphones i des de l'any 2010 amb el creixement de tablets.

Per una banda, aquesta revolució tecnològica ha suposat un replantejament de la web en si mateixa tal com la coneixíem fins al moment. Ja no només calia oferir una informació actualitzada i amb un bon format per a diferents resolucions i navegadors sinó una informació amb un format adaptable a diferents grandàries de pantalla, des de petites TFT a grans LCD com les dels tablet. La informació visible requeria ser adaptada per veure's d'una manera independent al dispositiu emprat per accedir-hi.

Per altra banda, aquest nou paradigma de desenvolupament d'aplicacions web ofereix un ampli ventall d'avantatges respecte les tradicionals aplicacions natives. En primer lloc, existeix una fragmentació important en els desenvolupaments d'aplicacions natives. Qualsevol projecte de gran o mitjana escala requereix l'estudi previ del seu SDK i la posterior compilació per a la portabilitat al codi natiu del dispositiu sobre el qual ha de funcionar. En segon lloc, cal un fort esforç per al manteniment d'una aplicació a diferents plataformes (BlackBerry OS, Android, iOS, Windows Phone, etc.); qualsevol canvi o millora cal adaptar-lo a cada plataforma per a mantenir l'aplicació al dia. Finalment, el mercat està reaccionant i adaptant-se a les noves necessitats i convergint cap a un model capaç de fer front la fragmentació de dispositius a l'ecosistema de xarxa.

El projecte final de carrera que es presenta realitza per una banda un estudi acurat de diferents tecnologies multi-plataforma i per altra, el desenvolupament d'una petita aplicació web mòbil – en endavant *webapp* – emprant un dels productes analitzats a l'estudi.

Per al desenvolupament de l'aplicació web mòbil s'ha emprat el llenguatge PHP orientat a objectes seguint un patró de disseny MVC. Per a la integritat de les dades s'ha optat per MySQL per la seva robustesa, rapidesa i gestió de transaccions.

## 2. Motivació del projecte

Fem servir diferents serveis 2.0 per enregistrar l'activitat del nostre dia a dia: Flickr per les fotos, Facebook per compartir novetats amb amics, Twitter per estar informats en temps real, Foursquare per enregistrar espais i compartir-los, Pinterest per compartir imatges, etc. Al cap d'una setmana hem distribuït molta informació que forma part de la nostra vida a diferents serveis. Tota aquesta informació la trobem distribuïda a diferents ubicacions i cada porció forma part d'un moment de la nostra vida, ja sigui una fotografia de les vacances d'estiu de l'any passat com una cançó que ens recorda el dia de la nostra graduació.

A mesura que passa el temps ens resulta més difícil identificar a quin moment de la nostra vida pertany aquella fotografia, aquella cita d'aquell autor, el perquè d'aquella cançó, etc. Anem compartint moments sense seguir un fil, sense identificar a quin record pertany, quin sentiment associat vam experimentar en escoltar aquella cançó. Totes aquestes emocions associades es van difuminant a mesura que passa el temps, es perd el valor del moment i únicament tenim una acció en el temps que podem o no recordar amb exactitud.

Actualment gairebé tothom té un smartphone o tablet amb connexió permanent a Internet. És, doncs, la millor eina de què disposem per enregistrar cada moment a la nostra vida i poder-lo compartir amb els nostres.

Així sorgeix aquest Projecte Final de Carrera, amb l'objectiu d'implementar una aplicació web mòbil capaç d'executar-se sobre qualsevol plataforma i per altra banda, l'estudi de les tecnologies web que permeten aquesta portabilitat entre diferents dispositius.

Aquest Projecte Final de Carrera – en endavant PFC – s'emmarca dins l'àrea de Xarxes de Computadors de la Universitat Oberta de Catalunya, més concretament a les tecnologies web multi-plataforma. El PFC inclou per una banda un estudi exhaustiu dels principals frameworks disponibles per al desenvolupament d'aplicacions web multi-plataforma i per altra, el desenvolupament d'una aplicació web mòbil seguint un patró de disseny MVC i emprant un dels frameworks de suport per aquest tipus de tecnologies.

L'ús del framework de desenvolupament escollit haurà d'estar justificat al resultat de l'estudi previ. Aquest estudi, destacarà els trets diferenciadors així com els avantatges i inconvenients de cadascú d'ells. Per a la comparativa es tindrà en especial consideració el tipus d'aplicació que es vol desenvolupar (un repositori de records) i per tant, aquells trets que puguin afavorir o obstaculitzar el pla de treball fixat a la planificació inicial del projecte. Finalment, es fixaran uns criteris de qualitat que serveixin de model per a poder dur a terme la comparativa final.

En el darrer punt de l'estudi comparatiu es comentarà quin framework es tria per al desenvolupament del projecte i quins arguments recolzen la decisió final. Per assolir aquesta fita, primer s'explicarà breument què és un framework web per al desenvolupament d'aplicacions mòbils i seguidament els criteris que cal tenir en compte a l'hora de comparar-

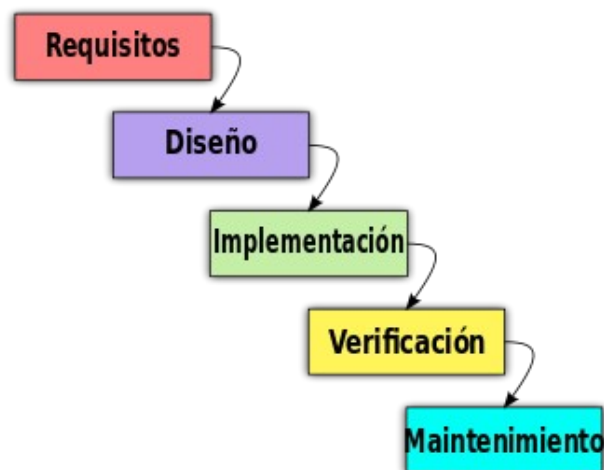
los. Finalment, es realitzarà una breu explicació de cadascun d'ells i es valoraran les funcions i característiques segons els criteris de comparació definits.

Actualment hi ha molts frameworks disponibles per al desenvolupament d'aplicacions web mòbils, tant de manera gratuïta com de pagament. Tot i això, per a la comparativa farem servir un subgrup més reduït, escollit segons uns criteris de qualitat específics, amb l'objectiu de disposar d'aquells més rellevants<sup>1</sup> en l'àmbit de desenvolupament; la resta, s'enuncien a títol informatiu però en cap cas formen part de l'estudi d'aquest document.

Deixem al lector els recursos de les fonts d'informació perquè pugui ampliar-ne informació pel seu compte.

### 3. Planificació

El projecte s'ha desenvolupat seguint les directrius del consultor i el calendari d'entregues per fases. Tot i així, s'ha seguit el cicle de desenvolupament de programari clàssic basat en cascada. Tot i no ser la millor opció a l'hora de desenvolupar un projecte d'aquest tipus, la planificació exigida pel calendari d'entregues ha condicionat l'adopció d'aquest model de desenvolupament.



*Fig 1: Esquema de fases del cicle de vida clàssic*

Tot i fixar una planificació inicial que incloïa una distribució equitativa d'hores per a cada fase, al llarg del desenvolupament del Treball Final de Carrera s'ha comprovat que la fase d'implementació englobava una distribució d'hores més important que no pas d'altres. Això es deu al fet que, per una banda, per aconseguir obtenir un producte final estable i verificable cal un calendari acord al desenvolupament, i per altra banda, el tipus de tecnologia emprada era gairebé desconeguda per l'alumne.

Resumint, les petites variacions que puguin haver afectat a les fases del projecte han estat

---

<sup>1</sup> Més endavant es descriuen quins criteris s'han considerat per a valorar la rellevància dels frameworks.



per la corba d'aprenentatge de l'alumne amb l'estudi del framework de desenvolupament.

### **3.1 Fase de planificació, anàlisi de requeriments i especificació**

En aquesta fase i un cop acordat amb el consultor el tipus de projecte que es volia desenvolupar al llarg del semestre, es va dissenyar el calendari de treball distribuït per fases. En aquest moment i degut a la desconexió del grau d'assoliment de la tecnologia emprada, es van incloure massa funcionalitats a la fase d'implementació que després s'han escurçat per poder arribar a assolir les fites del semestre.

Seguidament es van definir quins requeriments hauria de tenir l'aplicació desenvolupada tot desconexió quin seria el framework que s'empraria fruit de l'estudi previ que es realitzaria a una de les fases prèvies a la implementació. A més, es van descriure les funcionalitats de l'aplicació així com les tecnologies que, *a priori*, es podien fer servir per al tipus de desenvolupament que s'havia de realitzar.

Finalment es van enumerar les propietats d'alguna d'aquestes tecnologies, beneficis i es van identificar els riscos del desenvolupament.

### **3.2 Fase d'estudi comparatiu i disseny**

Aquesta fase constava de dues parts. Per una banda es va realitzar l'estudi comparatiu dels diferents frameworks de suport, establint quins criteris de comparació es feien servir i argumentant quina puntuació aconseguien cadascun d'ells segons aquests criteris. A més, es van descriure bones pràctiques de diferents fabricants a l'hora de dissenyar bones interfícies d'usuari per a dispositius mòbils.

Un cop realitzat l'estudi, es va escollir un framework de suport per al desenvolupament de l'aplicació web mòbil i es va justificar amb una breu descripció dels punts forts i febles de cada framework en concret.

Finalment en aquesta fase es van dissenyar un prototip d'interfícies d'usuari de les diferents possibles pantalles de l'aplicació web mòbil.

### **3.3 Fase d'implementació i verificació**

En la fase d'implementació primerament es va muntar l'entorn de treball per a poder fer funcionar l'aplicació. Seguidament es va dissenyar la base de dades i es va desenvolupar l'aplicació web mòbil seguint un patró de disseny MVC. Com a eina de suport i testing, s'ha creat una plantilla web a mode simulador que emula un dispositiu mòbil de darrera generació. Mitjançant CSS s'ha incrustat l'aplicació desenvolupada a la pantalla d'aquest simulador web.

Degut a la tipologia del Treball Final de Carrera, en concret a l'estudi previ de tecnologies web mòbils, s'ha inclòs en aquesta fase d'implementació el disseny tècnic. El motiu és que molts dels aspectes tècnics depenien del resultat de l'estudi de frameworks mòbils.

L'aplicació s'ha verificat seguint un model iteratiu on primer s'implementava una funcionalitat i seguidament es provava en l'entorn de proves. En cas de detectar un error a les proves, es tornava al codi per a implementar-ne la solució. En cas que tot funcionés correctament, es passava a la següent funcionalitat.

Finalment, un cop finalitzada la implementació, es va crear un document a mode manual d'ús de l'aplicació explicant cadascuna de les funcionalitats i les captures associades. Es va optar per aquest model degut a que l'entorn de proves es troba a un servidor local i la rèplica que es va pujar a Internet al proveïdor d'allotjament va acabar per donar problemes amb diferents versions de llibreries, en concret amb la versió de PDO (PHP Data Objects).

Per tant, calia poder ensenyar l'aplicació en funcionament fins a l'entrega de la fase final, on es realitzaria la presentació oficial en format vídeo per tal que els consultors i la resta de companys puguin veure l'aplicació en funcionament a un *screencast* realitzat per l'alumne.

### **3.4 Temporització del projecte**

La taula següent mostra la temporització de les tasques assignades a les diferents fases de projecte:

<b>Anàlisi previ i planificació</b>	<b>29/02/2012</b>	<b>12/03/2012</b>
Preparació del projecte	29/02/2012	02/03/2012
Proposta de projecte	03/03/2012	05/03/2012
Definició de projecte	06/03/2012	07/03/2012
Planificació de projecte	08/03/2012	09/03/2012
Creació del document d'entrega	10/03/2012	12/03/2012
<b>Estudi frameworks i disseny prototip</b>	<b>13/03/2012</b>	<b>10/04/2012</b>
Estudi frameworks	13/03/2012	02/04/2012
Disseny pantalles prototip	03/04/2012	09/04/2012
Creació del document d'entrega	13/03/2012	10/04/2012
<b>Anàlisi, disseny i implementació webapp</b>	<b>10/04/2012</b>	<b>21/05/2012</b>
Disseny arquitectònic	10/04/2012	12/04/2012
Disseny de classes	12/04/2012	16/04/2012
Disseny de persistència de dades	16/04/2012	17/04/2012
Preparació de l'entorn	10/04/2012	16/04/2012
Desenvolupament webapp HTML5, CSS3 i JavaScript	17/04/2012	18/05/2012
Proves de testing	18/05/2012	21/05/2012
<b>Documentació final i presentació virtual</b>	<b>22/05/2012</b>	<b>12/06/2012</b>
Redacció memòria final	22/05/2012	11/06/2012
Creació presentació virtual amb Impress	07/06/2012	08/06/2012
Vídeo comprimit amb explicació projecte	11/06/2012	11/06/2012
Entrega final	12/06/2012	12/06/2012

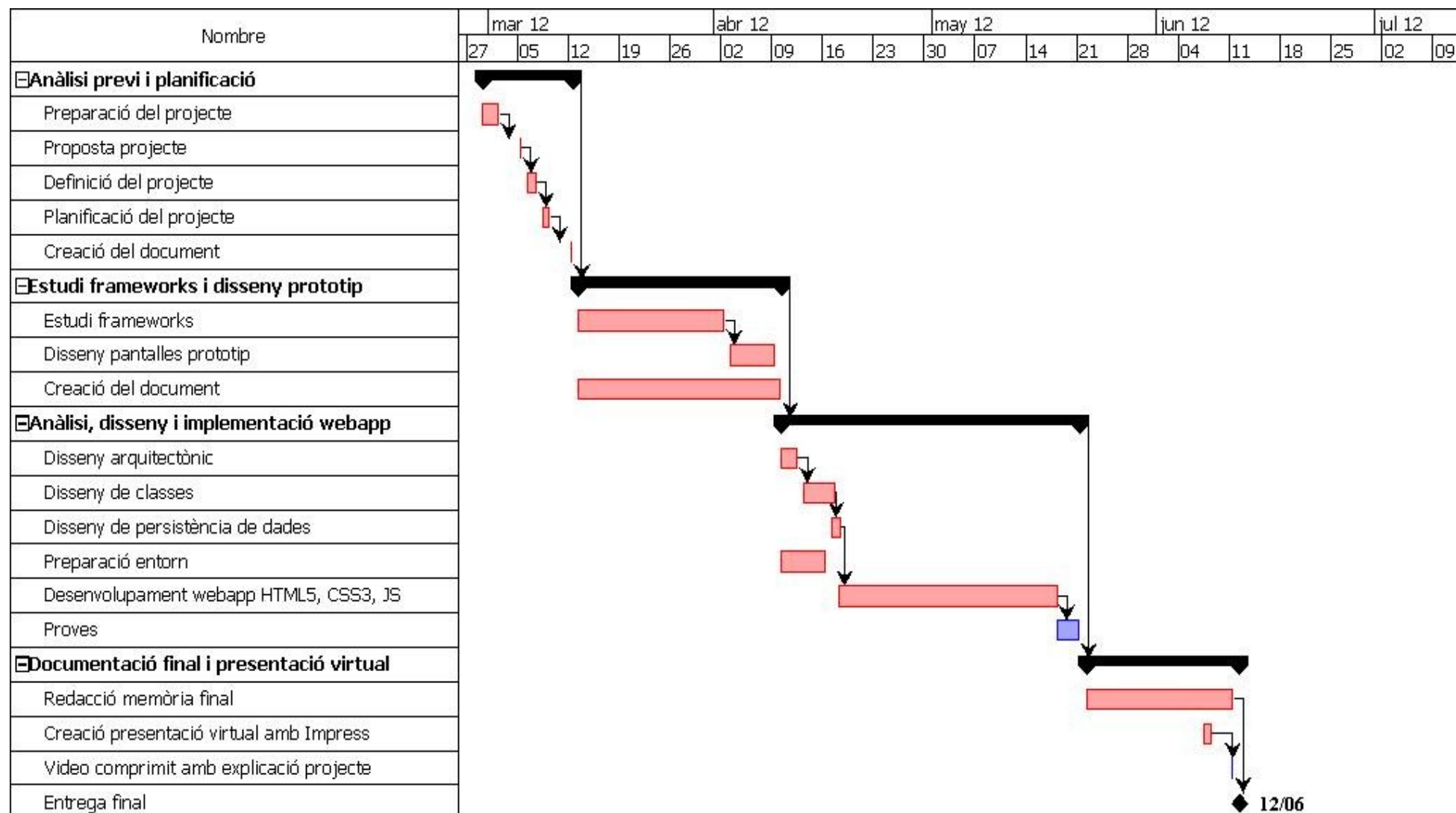


Fig 2: Diagrama gantt amb la temporització del projecte

### **3.5 Eines utilitzades**

Per a la realització del Projecte Final de Carrera s'han utilitzat les següents eines tant de desenvolupament com de suport:

- Servidor: Linux Fedora 16 3.3.7-1.fc16.i686
- Llenguatge de programació (backend): PHP 5.3.13
- Motor de base de dades: MySQL 5.5.22
- Gestor MySQL gràfic: phpMyAdmin 3.5.0
- Editor de codi: Sublime Text 2
- Framework de desenvolupament web mòbil: LungoJS 1.2
- Editor de planificació: OpenProj
- Diagrames d'anàlisi i disseny: Microsoft Visio 2003 i ArgoUML
- Prototip interfícies d'usuari: Balsamiq Mockup
- Diapositives presentació: LibreOffice Impress 3.3.0
- Vídeo presentació: gtk-recordMyDesktop
- Editor vídeo: OpenShot 1.4.2

### **3.6 Productes obtinguts**

Al llarg del desenvolupament de les diferents fases del projecte s'han generat els següents fitxers:

- Document de proposta de PFC i pla de treball.
- Document estudi frameworks i disseny possibles captures del prototip.
- Codi font aplicació web multi-plataforma que inclou:
  - Llibreries de connexió amb MySQL i variables globals.
  - Fitxers .js LungoJS amb la gestió d'esdeveniments i parametrització de l'aplicació.
  - Controladors per defecte.
  - Controlador de record.
  - Controlador d'usuari.
  - Model per defecte.
  - Model de record.

- Model d'usuari.
- Pantalles d'usuari amb navegació integrada amb efectes i icones de la llibreria de LungoJS.
- Base de dades relacional amb MySQL.
- Informe ús i funcionament de l'aplicació web.
- Memòria final de projecte.
- Presentació virtual amb vídeo .avi i diapositives amb Impress (LibreOffice).

## 4. Estudi comparatiu

### 4.1 Definició de framework

En el marc del desenvolupament de programari, un *framework* és:

*A universal, reusable software platform used to develop applications, products and solutions. Software Frameworks include support programs, compilers, code libraries, an application programming interface (API) and tools sets that bring together all the different components to enable development of a project or solution.*

Arrel la definició que ofereix Wikipedia, definim un framework com una estructura de programari de suport per al desenvolupament d'aplicacions, productes i altres solucions. És, doncs, un estalvi de feina pel programador ja que moltes de les funcions bàsiques (autenticació, operacions CRUD, disseny de layouts, memòria cau, etc.) ja es troben integrades al seu core<sup>2</sup>. De frameworks hi ha molts, per a diferents plataformes i arquitectures; és responsabilitat del desenvolupador fer ús d'aquell que satisfà els requeriments li garantirà un bon resultat final.

### 4.2 Frameworks per al desenvolupament d'aplicacions web mòbils

Un framework per al desenvolupament d'aplicacions web mòbils és una plataforma de suport per al desenvolupament d'aplicacions online per a diferents dispositius mòbils (telèfons, tablets i d'altres). Basa el seu funcionament en proporcionar una estructura mitjançant la qual poder dissenyar i executar pàgines web o serveis web per a l'usuari. El framework inclou tot el codi base necessari per al desenvolupament del projecte. Així, la llibreria del framework inclou codi:

- De suport al disseny del layout (gràfics, estructura grid del contenidor, codi JavaScript, etc.).
- D'estils i exemples (CSS3).

---

<sup>2</sup> En l'àmbit del programari: nucli que conté la totalitat del codi base d'una aplicació.

- De suport als esdeveniments tàctils (tap, double-tap, swipe, etc).
- De suport a la distribució de l'aplicació (packaging).
- De suport al tractament i intercanvi de dades (AJAX, JSON, etc.).
- De l'API, entre d'altres.

### 4.3 Avantatges

Hom es planteja la següent qüestió en iniciar un projecte de desenvolupament per a plataformes mòbils: aplicació nativa o web/híbrida?. El debat està obert. Si bé hi ha grans defensors d'un tipus de paradigma o altre, és cert que cal valorar *a priori* les necessitats del projecte per a poder escollir la millor opció. Per aquells casos en que es requereix un alt processament gràfic o accés al sistema de baix nivell és necessari un desenvolupament mitjançant les llibreries natives del dispositiu. En altres casos, però, es poden aconseguir grans resultats mitjançant aplicacions web. Tot i això, i amb l'aparició en escena d'HTML5, cada vegada és més estreta la línia que separa aquests dos paradigmes.

Els frameworks de desenvolupament web per a dispositius mòbils es caracteritzen, principalment, per permetre el desenvolupament de projectes (aplicacions, serveis web, prototipatge, etc.) per a plataformes mòbils amb la particularitat d'ésser multi-plataforma, és a dir, amb la garantia que el producte resultant funcioni sense modificacions<sup>3</sup> a diferents plataformes.

Respecte als llenguatges emprats, no deixa de ser una avantatge poder aprofitar coneixement ja adquirit prèviament com JavaScript, AJAX, JQuery, DHTML, HTML i CSS a l'hora de desenvolupar la nostra aplicació.

Degut a la independència de la plataforma, s'obvia la necessitat d'inscriure'ns (en el cas d'Apple cal registrar-nos com a desenvolupador i fer el pagament associat) per obtenir el SDK i poder accedir tant a la documentació com al canal de publicació, segons el cas.

### 4.4 Criteris de selecció

Un dels primers aspectes que cal tractar a l'hora de realitzar qualsevol estudi comparatiu és, sens dubte, determinar els atributs que descriuen els productes, atès que cada framework pot oferir característiques que el poden diferenciar de la resta. Es pretén disposar d'una llista de criteris que permeti comparar en les mateixes condicions els diferents frameworks. Així doncs, el resultat de la selecció del millor framework estarà condicionat a diversos factors com ara:

- Frameworks vius: productes vius, en desenvolupament, amb una comunitat/empresa al darrera que ofereixi suport i actualitzi el codi regularment.

---

<sup>3</sup> Si bé això no és cert al 100%. Existeixen plataformes que encara no suporten la totalitat de CSS3 i/o HTML5.

- Llicència: llicència de tipus obert que permeti la distribució del codi entre la comunitat així com la seva modificació.
- Ús de tecnologies actuals: llenguatges i tecnologies actuals, innovadores i en expansió. Aquest tipus de llenguatges garantiran una bona acceptació per part dels darrers terminals al mercat.
- Compatibilitat: amb els navegadors actuals existents a les diferents plataformes com Mozilla Firefox, Internet Explorer, Google Chrome, Opera i Safari.
- Estàndards: el codi produït amb els diferents frameworks ha de seguir els estàndards del consorci w3c.
- Comunitat de desenvolupadors: una comunitat activa de desenvolupadors que aportin coneixement tant en forma d'actualitzacions com solucions als *bugs*<sup>4</sup> que puguin aparèixer.
- Quota de mercat: nombre de dispositius i nombre d'aplicacions que fan ús del framework en concret.

Per a crear aquesta llista de criteris s'ha consultat informació publicada a mitjans tecnològics, blocs, fòrums de desenvolupadors, documentació oficial de cada framework i *screencasts* tant en anglès com en castellà.

#### **4.5 Selecció dels frameworks per l'estudi comparatiu**

Per realitzar l'estudi comparatiu d'aquesta fase del projecte i segons els criteris esmentats, s'ha fet una tria en la que s'han preseleccionat els cinc frameworks següents:

	<b>Framework</b>
<b>1</b>	Sencha Touch
<b>2</b>	JQuery Mobile
<b>3</b>	LungoJS
<b>4</b>	JQTouch
<b>5</b>	Titanium Mobile

---

<sup>4</sup> Errors de programació relacionats a un comportament no esperat de l'aplicació.



Sencha Touch és un dels frameworks web mòbils més emprats a la comunitat. Va ser un dels primers frameworks pioners en emprar HTML5 per al desenvolupament d'aplicacions web per a dispositius mòbils amb *look & feel* natiu. Fa servir tota la potència d'HTML5, CSS3 i JavaScript per a oferir a l'usuari una experiència agradable a qualsevol plataforma.

jQuery Mobile és un framework que deriva de la popular llibreria JavaScript, jQuery. Amb l'aparició de jQuery s'ofereix la possibilitat d'emprar una llibreria familiar con jQuery per al desenvolupament d'aplicacions web per a dispositius mòbils, sobretot per a desenvolupadors que ja coneguin jQuery tradicional.

LungoJS és un dels frameworks més joves del moment i tota una revolució a la comunitat de desenvolupadors d'aplicacions mòbils. Basa del seu funcionament en el que anomenen *Easy Prototype* o prototipat fàcil, o dit d'una altra manera, permet el disseny del primer prototip sense necessitat de coneixement a fons del llenguatge JavaScript. Té un control de tipus de variable fort i té un fort lligam amb HTML5 de manera que requereix que els dispositius en tinguin ple suport.

jQTouch és un framework basat en webkit amb una bona predisposició vers la web però amb molt camí per recórrer a les diferents plataformes mòbils actuals. No deixa de ser un plugin jQuery amb animacions, native *look & feel* i temes propis.

Titanium Mobile és un dels frameworks de la companya appcelerator. Titanium Mobile basa el seu funcionament en una pre-compilació de codi per portar el codi a les diferents plataformes mòbils disponibles. Inclou una extensa llibreria d'icones i temes d'usuari que possibiliten una interfície amb aparença nativa. Fa un ús extens de JavaScript per al desenvolupament de les aplicacions web mòbils.

S'ha decidit emprar una mostra reduïda per a facilitar la comprensió de l'estudi comparatiu. S'han publicat els recursos associats perquè el lector pugui ampliar-ne informació. Altres frameworks disponibles són:

- DHTMLX Touch
- The M Project
- WebAPP.Net
- Zepto.js
- DynamicX
- WinK Toolkit
- PhoneGap

## 4.6 Metodologia

Per a l'estudi comparatiu, únicament s'han tingut en compte les darreres versions dels frameworks a data de redacció d'aquest document. D'una banda, s'ha estudiat la documentació oficial associada i per altra, s'han revisat informació publicada a mitjans electrònics com blocs tecnològics, *screencasts*, articles, etc. En tot cas s'ha contrastat la informació obtinguda i en molts casos s'ha descarregat el paquet oficial amb el codi del framework per a revisar-lo a mesura que s'avançava amb l'estudi.

## 4.7 Característiques a comparar

A mesura que s'avança amb l'estudi, s'ha pogut comprovar que si bé la majoria dels frameworks tenen una estructura diferent, tots comparteixen un conjunt de funcionalitats comunes que ofereixen un resultat final semblant. A més, l'única manera de poder-los comparar és en aquells punts on coincideixen. Aquests criteris de comparació són:

- Documentació: quantitat de documentació i qualitat i claredat de la mateixa. Amb documentació tant s'inclou documentació escrita com oral (exemple *screencasts*).
- Native "look & feel": resultat final més semblant a una aplicació nativa.
- Animacions i efectes: possibilitat de realitzar animacions i efectes sobre els elements de la interfície gràfica.
- MVC: patró model-vista-controlador.
- Components externs: conjunt de funcionalitats extres fora del *core* del framework que ofereixen un valor afegit al resultat final. Tant poden ser del mateix framework com la possibilitat d'incloure components desenvolupats pel mateix programador.
- Adaptive Layouts: presentació optimitzada del contingut de la informació a la pantalla de manera independent a la resolució de pantalla, grandària, orientació, etc.
- Ajax: gestió del dinamisme de l'aplicació amb Ajax.
- JSON: suport format JSON per a l'intercanvi de dades.
- Llibreria d'icones: recursos d'icones vectoritzades, de diferents temàtiques per a incloure al projecte.
- Gestió dels esdeveniments sobre la pantalla: nombre d'esdeveniments gestionats i les possibilitats que ofereix.
- Corba d'aprenentatge: grau d'èxit durant l'inici del desenvolupament amb el framework.

## 4.8 Selecció del millor framework

S'ha optat per una escala de 0 a 10 per a la puntuació de cada característica a valorar segons els criteris de comparació definits a l'apartat anterior, de manera que 0 valora aquella funcionalitat que no respon en cap nivell la funcionalitat que se li demana i 10 la que ofereix el servei a la seva totalitat.

Framework	Sencha Touch	jQuery Mobile	JQTouch	Titanium Mobile	LungoJS
Versió	2.0.0	1.7.2	V1, beta 4rc	1.8.2	1.2
Documentació	7	7	9	8	8
Native look & feel	9	8	6	10	8
Animacions i efectes	9	8	6	9	9
Components <sup>5</sup> Externs / Plugins	10	10	8	9	7
Adaptive Layouts	8	8	6	6	8
AJAX	10	10	10	10	10
JSON	10	10	10	10	10
Llibreria d'icones	10	9	5	7	10
Touch events	10	10	6	10	10
Corva Aprementatge	6	7	9	7	10
Rendiment	9	8	4	7	9
<b>TOTAL</b>	<b>98</b>	<b>95</b>	<b>80</b>	<b>94</b>	<b>99</b>

<sup>5</sup> Tant components disponibles a la comunitat com la possibilitat de desenvolupar els propis.

## 4.9 Resum de l'anàlisi comparatiu

A continuació es presenta un resum descriptiu on s'exposa d'una manera breu i concisa els diferents punts de l'anàlisi comparatiu.

- Documentació: en aquest apartat jQTouch és el framework que té una millor documentació en quant a qualitat i quantitat. Altres com Sencha Touch o jQuery Mobile tenen una documentació molt dispersa i complexa que fa minvar la corba d'aprenentatge. És per això que amb aquests frameworks els inicis dels projectes és quan es produeixen la major part dels errors. Titanium Mobile, en canvi, tot i tenir una documentació molt segmentada (una per cada plataforma (mòbil i escriptori), és d'una bona qualitat i amb un alt índex de claredat. LungoJS té una bona documentació; tot i això, únicament hi ha disponible documentació en anglès (tot i ser un projecte espanyol), segons l'autor, per poder competir en iguals condicions amb altres projectes internacionals. S'ha valorat positivament que ofereixin un canal de publicació amb screencasts en castellà.
- Native look & feel: degut al lligam amb la plataforma de Titanium Mobile i de la semblança de la versió Mobile amb la versió Desktop, l'aspecte visual final d'aquest resulta més semblant a una aplicació nativa que no pas a d'altres. Tot i això, altres com Sencha Touch i el seu gran conjunt de llibreries aconseguen un resultat final òptim sense sacrificar la seva independència de la plataforma. jQuery Mobile i LungoJS ofereixen un aspecte notable i jQTouch se l'ha penalitzat degut al seu baix rendiment en l'aspecte visual, en transicions inestables i retards en temps de resposta.
- Animacions i efectes: tots disposen d'interfícies dinàmiques amb una bona gestió d'esdeveniments i efectes sobre la pantalla. Tots els frameworks excepte jQTouch tenen ben resolta aquesta funcionalitat degut al bon ús de tecnologies com Ajax i CSS3. jQTouch, en canvi, pateix problemes de rendiment a l'hora d'interactuar amb la interfície en quant a transicions i rendiment.
- Components externs / plugins: lungojs és el framework que no conté una comunitat forta de plugins al voltant del core del framework per ampliar funcionalitats, potser degut al poc temps que porta al mercat. Tot i això, se l'ha valorat ja que és molt modular i permet a l'usuari crear-ne els seus mòduls externs, en cas que es requereixi. Tota la resta i degut a l'experiència i a la comunitat/empresa que li ofereix suport, ofereixen un bon repositori de components externs. Titanium Mobile disposa d'un repositori de plugins actiu i complet, amb plugins com Coffice (eines de Microsoft Office), Ctwitter, Cfacebook, Twitter Panel, etc. Sencha Touch ofereix un conjunt de plugins externs per proveir de funcionalitats addicionals al codi base de la llibreria Ext. Finalment, s'ha valorat amb la màxima puntuació a Sencha Touch i jQuery Mobile.

- Adaptive Layouts: en aquest apartat s'ha penalitzat a jQTouch pel seu baix rendiment a la resposta d'esdeveniments tàctils així com a les transicions. La resta (Sencha Touch, jQuery Mobile i LungoJS) tenen una bona gestió per a l'optimització de la interfície d'usuari a diferents orientacions i dispositius varis. Titanium Mobile, en canvi, és un cas tècnicament especial ja que compila el codi per generar codi natiu pel dispositiu. Per tant, cal una tasca addicional per a poder-se executar en diferents plataformes.
- AJAX i JSON: tots els frameworks tenen compatibilitat i fan ús de tecnologies com Ajax i codi que gestionen estructures de dades en format JSON. A tots se'ls hi ha puntuat amb la màxima puntuació possible.
- Llibreria d'icones: en aquest apartat Sencha Touch i LungoJS són les dues solucions que ofereixen una completa llibreria d'icones de temàtica variada com a recursos per als desenvolupadors. En el cas de LungoJS, aquestes són vectoritzats i per a fer-los servir únicament cal saber el nom semàntic. jQuery Mobile també disposa d'un bon repositori de recursos gràfics.
- Touch events: tots excepte jQTouch disposen d'una bona gestió d'esdeveniments a la pantalla tàctic com swipe, tap, double-tap, etc. jQTouch té problemes constatables de rendiment i per aquest motiu és l'únic que se l'ha penalitzat.
- Corva aprenentatge: LungoJS, amb el seu paradigma *easy prototype* (prototipat fàcil), és el framework que garanteix un alt grau d'èxit d'aprenentatge al llarg del desenvolupament del projecte. L'estil de codificació (per a realitzar un primer prototipat no cal escriure cap línia de JavaScript, únicament HTML pur) i la consulta dels screencasts en castellà garanteix un seguiment adient i una progressió constant. JQTouch, tot i no ser dels capdavanters, també té un alt grau d'èxit degut a la qualitat de la seva documentació. JQuery i Titanium Mobile seguirien una evolució normal i Sencha Mobile, en canvi, és el framework pitjor valorat en aquest aspecte degut a la seva complexa documentació.
- Rendiment: en rendiment generalment tots ofereixen un rendiment òptim que satisfà les necessitats dels desenvolupadors. Únicament comentar els problemes gràfics de jQTouch a la gestió d'esdeveniments i transicions. jQuery Mobile se l'ha penalitzat simbòlicament per la seva naturalesa versàtil (originàriament llibreria orientada a escriptori, genera un projecte massa gran i complexe degut a la compatibilitat de dispositius). Titanium Mobile compta amb avantatge degut a la seva naturalesa nativa (compilació de codi per a cada dispositiu) i és per això que s'ha cregut convenient no valorar-ho positivament sinó negativament ja que l'objectiu és obtenir una única aplicació multi-plataforma.

## 4.10 Conclusions

L'estudi ha permès veure els principals frameworks de desenvolupament de projectes per a dispositius mòbils i els seus avantatges i inconvenients.

S'ha vist que generalment la documentació associada sol ser un aspecte important a tenir en compte, lligat directament amb la corba d'aprenentatge inicial i els errors produïts al codi. Per altra, i en la mateixa línia, s'ha vist que existeix un altre tipus de documentació (multimèdia) igualment vàlida com són els screencasts interactius.

L'estudi ha permès veure que hi ha frameworks que directament estaven molt allunyats de les necessitats del projecte. És el cas de jQTouch degut als seus problemes de rendiment en transicions i efectes sobre la interfície. Per altra banda, també s'ha observat que Titanium Mobile era un cas especial degut a la seva naturalesa nativa. Per tant, en un principi el descartaríem degut a que una de les premisses del nostre projecte és garantir l'aspecte multi-plataforma amb un sol desenvolupament.

També s'ha pogut observar que existeixen tres frameworks molt complets i molts semblants entre si; aquests són Sencha Touch, jQuery Mobile i LungoJS. S'ha vist que jQuery Mobile degut als seus orígens *desktop*, comparteix (potser en excés) molt codi base de la seva versió escriptori. Per tant, i segons es pot consultar a Internet, els projectes resultants són projectes massa complexes respecte el volum de codi final.

Sencha Touch és un dels frameworks amb una comunitat més forta, una empresa forta al darrera i una versió lliure i una comercial. És, doncs, un producte complet. Tot i això, però, no ha pogut evitar que tingui una documentació massa dispersa i complexa i per tant, una corba d'aprenentatge alta. Aquest fet s'ha considerat de vital importància per a projectes a curt termini on la corba d'aprenentatge inicial ha de ser suau i progressiva.

LungoJS és el framework que té la corba d'aprenentatge més baixa degut al paradigma *easy prototype* o és el que és el mateix, dissenyar un prototip inicial sense la necessitat d'escriure cap línia en JavaScript (únicament HTML). És un projecte viu i en desenvolupament amb un resultat final molt semblant a Sencha Touch.

Finalment, i un cop realitzat l'estudi, s'ha considerat que els dos frameworks que s'adapten millor per al desenvolupament del projecte són Sencha touch i LungoJS. Tot i això, i degut al *timing* amb que disposem per a la fase d'implementació, s'ha decidit valorar positivament i així optar a la implementació a LungoJS degut a la seva senzillesa així com la qualitat del resultat final.

## 5. Anàlisi

En aquest apartat s'exposa el resultat de la fase d'anàlisi del projecte. Arrel el diagrama de casos d'ús que descriuen els requeriments funcionals de l'aplicació, seguidament es realitzarà una descripció ampliada de cadascun d'ells i es detallaran els requeriments funcionals i no funcionals a tenir en compte així com els riscos observats.

### 5.1 Introducció

L'objectiu principal d'aquest projecte és la implementació d'un repositori de records per enregistrar experiències i emocions que volem mantenir intactes al llarg del temps, moments fàcilment accessibles des de qualsevol smartphone o tableta. Entenem per record qualsevol experiència que pugui ser compartida mitjançant un text, una fotografia o un vídeo<sup>6</sup>, i que pugui ser identificada al llarg del temps.

El servei s'implementarà fent servir com a principal eina un dels diferents frameworks mòbils disponibles a dia d'avui com per exemple *Sensa Touch*, *LungoJS*, *jQuery Mobile* o *Appcelerator Titanium*, entre d'altres. Finalment es pretén desenvolupar una *webapp* capaç de funcionar sobre diferents plataformes fent servir com a tecnologia base HTML5, CSS3 i JavaScript, i amb una interfície d'usuari molt semblant a les aplicacions natives com les existents a Android, iPhone o Windows Phone.

L'aplicació ha de ser capaç de proporcionar a l'usuari d'una manera fàcil i intuïtiva la creació d'un record així com la publicació de comentaris entre usuaris.

L'aplicació final garantirà una experiència d'usuari molt propera a les aplicacions natives, amb interfície d'usuari agradable i gestió d'esdeveniments davant la interacció multi-tàctil. Al llarg del desenvolupament d'aquest Treball Final de Carrera es pretenen assolir els següents objectius:

- L'estudi de tecnologies emergents com les aplicacions web multi-plataforma.
- Posar en pràctica els coneixements adquirits a diferents assignatures dels estudis d'Enginyeria Informàtica com el desenvolupament d'aplicacions orientat a objectes, base de dades, tècniques de desenvolupament de programari, entre d'altres.
- Aplicar el patró de disseny MVC i integrar-ho amb un framework de desenvolupament d'aplicacions web multi-plataforma per a gestionar les vistes i les accions.
- Treballar amb esdeveniments sobre llenguatge client JavaScript, gestionat pel framework.

---

<sup>6</sup> En aquesta primera versió alpha no s'ha inclòs la possibilitat de publicar un vídeo associat a un record. Es deixa obert per a futures versions.

- Posar de manifest la solidesa del llenguatge PHP i la bona integració amb el sistema gestor de base de dades MySQL.
- Dissenyar la plataforma tècnica sobre la qual ha de funcionar l'aplicació desenvolupada. Plataforma formada per un servidor Linux de 32 bits, servidor web Apache, PHP 5.3.13, MySQL 5.5.22.

## **5.2 Justificació**

Atès que l'enginyeria de programari és una tasca a càrrec de persones humanes, la probabilitat de tenir-ne errors i que puguin afectar al resultat final és molt alta. A banda d'això, no sempre hom és fidel a la realitat del plantejament inicial del projecte i n'és conscient un cop entregat el projecte a l'usuari final i avaluades les funcionalitats finals (el client final o bé el tribunal avaluador en aquest cas).

A dia d'avui és imprescindible comptar amb una disciplina i una metodologia que compti amb un anàlisi funcional i tècnic, on es pugui estudiar i avaluar les funcionalitats de l'aplicació, en aquest cas. Per altra banda, les persones que interactuaran amb l'aplicació ho faran de diferents perspectives:

- L'usuari final des de la perspectiva de les funcionalitats.
- El tribunal avaluador i/o beta testers<sup>7</sup> des de la perspectiva tant tècnica com funcional.

Finalment, un bon anàlisi inicial garanteix un bon cicle de vida del projecte al llarg del temps garantint una bona adaptabilitat davant futures ampliacions i/o modificacions.

---

<sup>7</sup> Usuari amb coneixements informàtics que avalua el funcionament d'un programa per a la detecció d'errors.



## 5.3 Requeriments

### 5.3.1 Usuaris de l'aplicació

Es considera un únic tipus d'usuari a l'aplicació web mòbil. Tots els usuaris estaran al mateix nivell i tindran les mateixes funcionalitats.

### 5.3.2 Requeriments funcionals

Són aquells tipus de requeriments que expressen la naturalesa de l'aplicació, com interactua el servei amb l'entorn. Entre els requisits funcionals a l'aplicació desenvolupada trobem:

- Iniciar sessió / autenticació: els usuaris poden validar-se contra el servidor o registrar-se al sistema. El sistema s'encarrega de validar les dades d'accés i en cas de validar correctament accedeix a la pantalla principal de l'aplicació.
- Finalitzar sessió: l'aplicació requereix un navegador compatible amb HTML5 per a accedir a l'aplicació. En el moment de tancar el navegador es tanca la sessió de l'usuari. S'ha optat per aquest sistema per senzillesa i per facilitar l'ús de l'aplicació per part d'usuari sense haver de validar-se contínuament.
- Alta de record: permet a l'usuari donar d'alta un nou record al sistema. Cal haver omplert els camps correctament.
- Alta de publicació: permet a l'usuari publicar contingut associat al record. El contingut tant pot ser una imatge provinent d'una url a Internet o un text.
- Cerca de record: permet a l'usuari cerca per paraula clau i llistar els records coincidents. L'usuari pot consultar la informació de qualsevol record de la llista de resultats.
- Alta de comentari: permet a l'usuari comentar una publicació d'un usuari associada sempre a un record.
- Sol·licitud d'edició: permet a l'usuari participar en l'edició de records sobre els quals no té autoria.

### 5.3.3 Requeriments no funcionals

Són aquells tipus de requeriments en el domini de possibles restriccions sobre el sistema dissenyat. El sistema haurà de comptar amb:

- Una bona gestió dels esdeveniments multi tàctil sobre la pantalla.
- Una interfície dissenyada i adaptable a la grandària de les pantalles dels diferents dispositius sobre els quals s'accedeix.
- Una interfície ràpida i lleugera.
- Una interfície compatible amb HTML5.
- Una interfície amb aparença semblant a les aplicacions natives.

A més de complir els objectius proposats, l'aplicació web ha de satisfer els següents requeriments:

- Portabilitat: cal poder executar l'aplicació des de qualsevol plataforma. Entre les més importants hi ha Android de Google, iPhone d'Apple, Windows Phone de Microsoft i Blackberry OS de RIM.
- Rapidesa: cal que el seu funcionament sigui ràpid per garantir una bona experiència d'usuari.
- Senzilla: l'ús de l'aplicació ha de ser fàcil i intuïtiu.
- HTML5 compliant: ha de garantir la compatibilitat amb aquells navegadors amb suport d'HTML5 com el navegador d'Android, Chrome/Chromium i safari.
- Esdeveniments: ha de facilitar la interacció amb pantalles tàctils així com una bona gestió d'esdeveniments com: *tap*, *double-tap*, *swipe*, *pinch*, etc.
- Seguretat: ha de garantir una capa de seguretat robusta tant a nivell d'aplicació com d'integració.
- Distribució: possibilitat de distribució i promoció mitjançant web o *mobile stores*.
- Easy prototype: o la facilitat d'assolir un prototip d'interfície d'aplicació.

## 5.4 Funcionalitats de l'aplicació

*Recoly* és un repositori de records al núvol. Actua a mode de magatzem de records permetent a l'usuari enregistrar qualsevol moment de la seva vida i compartir-ho amb la resta d'usuaris. En aquesta primera versió *alpha* de l'aplicació es permet únicament la publicació de text i imatges dins un record. Entenem record com qualsevol experiència que pugui ser compartida mitjançant un text o imatge. La imatge, però, ha de ser un objecte *url* a Internet com poden ser fotos de qualsevol pàgina web, serveis com twitter, flickr, facebook, i en general qualsevol imatge que pugui ser accessible de manera pública mitjançant Internet i una adreça *url*.

L'aplicació mòbil ha de tenir les funcionalitats següents:

- Registrar-se al servei: l'aplicació requerirà d'un usuari i clau per a fer servir l'aplicació. Es valorarà l'ús de protocols d'autenticació com Oauth, OpenID o Facebook Connect.
- Llistar les novetats: a l'estil Facebook, es mostrarà un resum de les darreres publicacions dels contactes del cercle de confiança de l'usuari.
- Alta nou record: funcionalitat clau a l'aplicació. L'usuari requereix d'obrir un nou record en un moment donat i pot o no completar-ho en aquell moment o més endavant.
- Esborra record: els records es poden esborrar i amb ell tota la informació relacionada.
- Cerca record: per interval entre dates, per usuari i per emoció/sentiment. Es possibilita la cerca ràpida de records mitjançant filtres.
- Publicar contingut associat a un record: l'usuari va omplint el record tant amb text com fotos i vídeos.
- Comentar publicació associat a un record: es permet comentar qualsevol publicació d'un record.
- Establir permisos associats a un record: els records poden ser públics o privats. Aquest fet condiona qui pot veure el record i qui pot col·laborar en el record.
- Enllaçar un usuari a un record: el creador del record pot afegir usuaris al seu record. Aquests usuaris poden participar en l'edició del record.

## **5.5 Riscos**

Al llarg del desenvolupament del projecte s'han observat diferents aspectes que podien posar en perill la viabilitat de la tecnologia emprada (tecnologia web mòbil en aquest cas).

Entre els punts crítics observats trobem:

- Necessitat d'un alt potencial gràfic.
- Necessitat d'accés a rutines de baix nivell del SO.
- Limitacions de l'aplicació com ús eficient de la geolocalització.
- Dades emmagatzemades en local per a ús en *offline*.
- Rendiment amb ample de banda limitat.
- *Uptime* no garantit.
- *Overhead* del servei per l'èxit i punts d'histèria.

S'ha intentat, sempre que ha estat possible, valorar la viabilitat de la funcionalitat crítica i cercar alternatives per assolir l'èxit del projecte sense haver de sacrificar punts vitals de l'aplicació.

## 5.6 Diagrama de Casos d'ús

Els diagrama de casos d'ús representa el comportament del sistema mitjançant la interacció amb els usuaris en resposta a un esdeveniment.

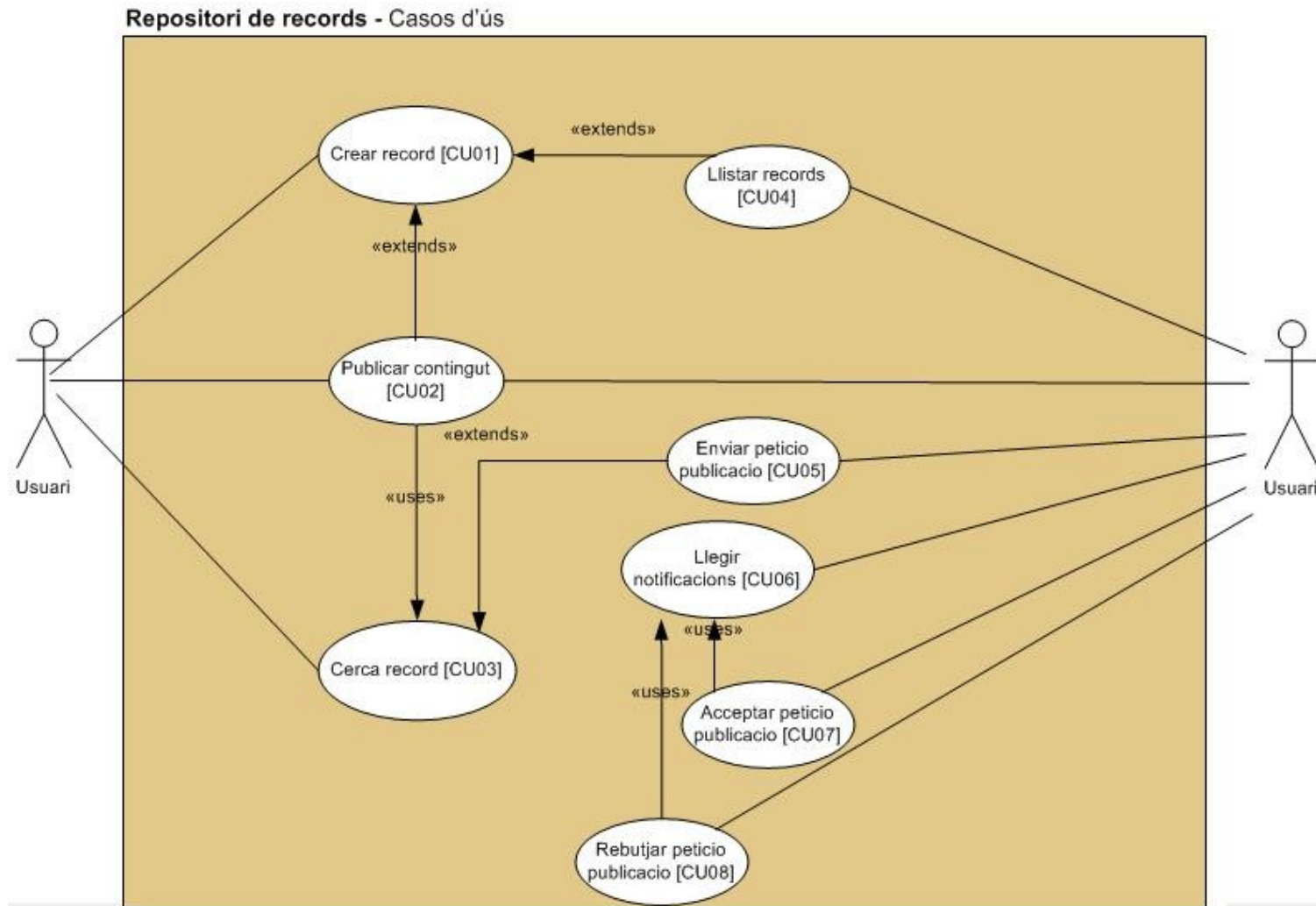


Fig 3: Diagrama de casos d'ús

## 5.7 Especificació de casos d'ús

### 5.7.1 CU01 – Crear record

Cas d'ús	Crear record [CU01]
Descripció	Aquest cas d'ús descriu com l'usuari inicia un record al sistema
Actors	Usuari
Precondició	
Flux principal	<ol style="list-style-type: none"><li>1. L'usuari s'identifica al servei amb el seu usuari i contrasenya</li><li>2. El sistema comprova les credencials i valida l'accés.</li><li>3. L'usuari desplega el menú lateral i accedeix a l'opció per a crear un nou record.</li><li>4. L'usuari omple els camps títol i descripció i crea el nou record.</li></ol>
Subfluxos	<ul style="list-style-type: none"><li>• El sistema informa que les credencials no són correctes.</li><li>• El sistema informa que hi ha camps buits per al nou record.</li></ul>
Postcondició	
Requeriments no funcionals	
Prioritat	Alta
Comentaris	

### 5.7.2 CU02 – Publicar contingut

Cas d'ús	Publicar contingut [CU02]
Descripció	Aquest cas d'ús descriu com l'usuari publica contingut relacionat amb un record.
Actors	Usuari
Precondició	Existeix un record i s'hi té permís per editar contingut.
Flux principal	<ol style="list-style-type: none"><li>1. L'usuari s'identifica al servei amb el seu usuari i contrasenya</li><li>2. El sistema comprova les credencials i valida l'accés.</li><li>3. L'usuari desplega el menú lateral i accedeix a l'opció per a cercar un record.</li><li>4. L'usuari omple el camp de cerca i accedeix a la fitxa del record.</li><li>5. Omple les dades de la publicació i envia.</li></ol>
Subfluxos	<ul style="list-style-type: none"><li>• El sistema informa que no té permís per editar el record.</li><li>• El sistema informa que no existeix cap record coincident amb els criteris de cerca.</li></ul>
Postcondició	
Requeriments no funcionals	
Prioritat	Alta

Comentaris	
------------	--

### 5.7.3 CU03 – Cerca record

Cas d'ús	Cerca record [CU03]
Descripció	Aquest cas d'ús descriu com l'usuari cerca records al sistema.
Actors	Usuari
Precondició	
Flux principal	<ol style="list-style-type: none"><li>1. L'usuari s'identifica al servei amb el seu usuari i contrasenya</li><li>2. El sistema comprova les credencials i valida l'accés.</li><li>3. L'usuari desplega el menú lateral i accedeix a l'opció per a cercar un record.</li><li>4. L'usuari omple el camp de cerca.</li><li>5. L'usuari comprova els resultats obtinguts.</li></ol>
Subfluxos	
Postcondició	
Requeriments no funcionals	
Prioritat	Alta
Comentaris	

### 5.7.4 CU04 – Llista records

Cas d'ús	Llista records [CU04]
Descripció	Aquest cas d'ús descriu com l'usuari accedeix a les darreres novetats dels usuaris que formen la plataforma de records.
Actors	Usuari
Precondició	Existeixen records al sistema.
Flux principal	<ol style="list-style-type: none"><li>1. L'usuari s'identifica al servei amb el seu usuari i contrasenya</li><li>2. El sistema comprova les credencials i valida l'accés.</li><li>3. L'usuari accedeix a la pantalla de darreres novetats on es visualitzen les darreres publicacions i comentaris dels diferents records.</li></ol>
Subfluxos	
Postcondició	
Requeriments no funcionals	
Prioritat	Alta
Comentaris	

### 5.7.5 CU05 – Enviar petició publicació

Cas d'ús	Enviar petició publicació [CU05]
Descripció	Aquest cas d'ús descriu com l'usuari envia una petició a un altre usuari per a poder publicar contingut associat al record.
Actors	Usuari
Precondició	Existeixen records al sistema.
Flux principal	<ol style="list-style-type: none"><li>1. L'usuari s'identifica al servei amb el seu usuari i contrasenya</li><li>2. El sistema comprova les credencials i valida l'accés.</li><li>3. L'usuari visualitza la fitxa informativa del record.</li><li>4. L'usuari envia una petició de publicació a l'espera que l'altre usuari li aprovi.</li></ol>
Subfluxos	
Postcondició	
Requeriments no funcionals	
Prioritat	Alta
Comentaris	

### 5.7.6 CU06 – Llegir notificacions

Cas d'ús	Llegir notificacions [CU06]
Descripció	Aquest cas d'ús descriu com l'usuari consulta la safata de notificacions d'entrada.
Actors	Usuari
Precondició	Existeixen notificacions pendents per a l'usuari.
Flux principal	<ol style="list-style-type: none"><li>1. L'usuari s'identifica al servei amb el seu usuari i contrasenya.</li><li>2. El sistema comprova les credencials i valida l'accés.</li><li>3. L'usuari consulta la seva safata de notificacions.</li><li>4. L'usuari llegeix les notificacions.</li><li>5. L'usuari accepta/rebutja la petició.</li></ol>
Subfluxos	
Postcondició	
Requeriments no funcionals	
Prioritat	Alta
Comentaris	



### 5.7.7 CU07 – Acceptar petició publicació

Cas d'ús	Acceptar petició publicació [CU07]
Descripció	Aquest cas d'ús descriu com l'usuari accepta una sol·licitud de participació per al seu record.
Actors	Usuari
Precondició	Existeixen notificacions pendents per a l'usuari.
Flux principal	<ol style="list-style-type: none"><li>1. L'usuari s'identifica al servei amb el seu usuari i contrasenya.</li><li>2. El sistema comprova les credencials i valida l'accés.</li><li>3. L'usuari consulta la seva safata de notificacions.</li><li>4. L'usuari llegeix les notificacions.</li><li>5. L'usuari accepta.</li></ol>
Subfluxos	
Postcondició	
Requeriments no funcionals	
Prioritat	Alta
Comentaris	

### 5.7.8 CU08 – Rebutjar petició publicació

Cas d'ús	Rebutjar petició publicació [CU08]
Descripció	Aquest cas d'ús descriu com l'usuari rebutja una sol·licitud de participació per al seu record.
Actors	Usuari
Precondició	Existeixen notificacions pendents per a l'usuari.
Flux principal	<ol style="list-style-type: none"><li>1. L'usuari s'identifica al servei amb el seu usuari i contrasenya.</li><li>2. El sistema comprova les credencials i valida l'accés.</li><li>3. L'usuari consulta la seva safata de notificacions.</li><li>4. L'usuari llegeix les notificacions.</li><li>5. L'usuari rebutja la petició.</li></ol>
Subfluxos	
Postcondició	
Requeriments no funcionals	
Prioritat	Alta
Comentaris	

## 6. Disseny

### 6.1 Arquitectura

L'arquitectura de l'aplicació ve determinada pel patró de disseny emprat al llarg de tot el desenvolupament juntament amb el model de persistència emprat.

S'ha seguit un patró de disseny Model-Vista-Controlador (MVC) sobre llenguatge PHP per al backend, orientat a objectes (OO) i un model de base de dades de tipus relacional com és MySQL. Si bé en un principi es va pensar en poder emprar un tipus de base de dades no relacionals, s'ha considerat adient aquest tipus per publicar la primera versió amb la possibilitat de poder canviar de motor de base de dades a futures versions.

El patró de disseny MVC garanteix un bon manteniment de l'aplicació ja que separa la lògica de negoci de la capa de presentació. En aquest projecte aquest fet és de vital importància doncs deleguem una part molt important de la capa de presentació al framework.

Al model és on hi ha la lògica de negoci i s'encarrega de l'abstracció de la lògica relacionada amb les dades de la nostra aplicació, permetent una gestió de les pantalles i accions sobre els seus components (botons, llistes, scrolls, camps de formulari, etc.) de manera independent al gestor de base de dades emprat, en aquest cas MySQL.

### 6.2 Vistes de l'aplicació

#### 6.2.1 Vista física

El model emprat per al disseny del sistema que ha d'absorbir la càrrega de l'aplicació s'ha dissenyat pensat per ser fàcilment escalable. Tal com es veu a la imatge següent, els usuaris accedeixen des de qualsevol dispositiu mitjançant un navegador compatible amb HTML5. Accedeixen a l'aplicació a través d'Internet fins al servidor web amb Apache. Aquest tipus de servidor web és fàcilment escalable simplement introduint un balancejador i creant més instàncies web. Per altra, al servidor web és on trobem el codi font de l'aplicació, en el nostre cas la combinació del tàndem PHP + framework LungoJS.

És el servidor web qui es comunica directament amb el servidor de base de dades MySQL. En aquest primer disseny s'ha considerat suficient amb un únic servidor. A futures versions i segons la càrrega a absorbir, es pot crear un clúster MySQL o migrar l'actual sistema gestor a un altre tipus (exemple Cassandra).

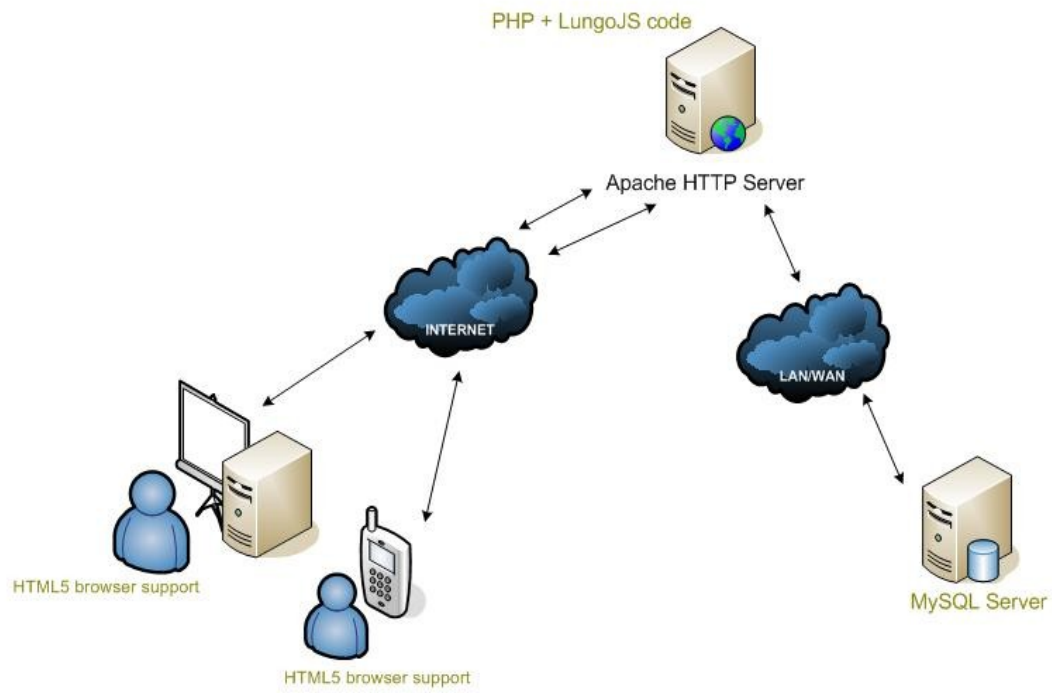


Fig 4: Escenari de desplegament

## 6.2.2 Vista lògica

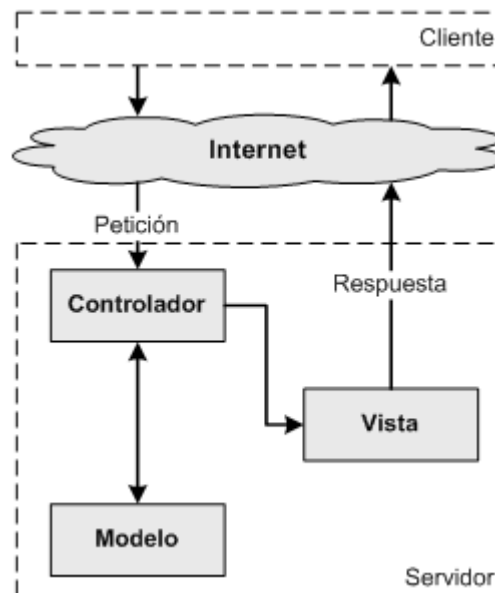


Fig 5: Esquema webapp amb MVC

- El Model s'encarrega d'interactuar amb la base de dades i és a on hi ha la lògica de negoci.
- El Controlador s'encarrega de processar les peticions que li arriben de la Vista (en el nostre cas del frontend de l'aplicació web mòbil) i envia aquestes dades al Model perquè aquest li retorni la informació necessària per a completar la petició i retornar les dades a la Vista.
- La Vista és el codi HTML que es mostra a l'usuari i sobre el qual interactua amb l'aplicació. Rep la informació directament del Controlador.

### 6.2.3 Vista de components

El diagrama següent descriu la distribució de l'estructura dissenyada des del punt de vista dels components desenvolupats per al bon funcionament de l'aplicació web mòbil:

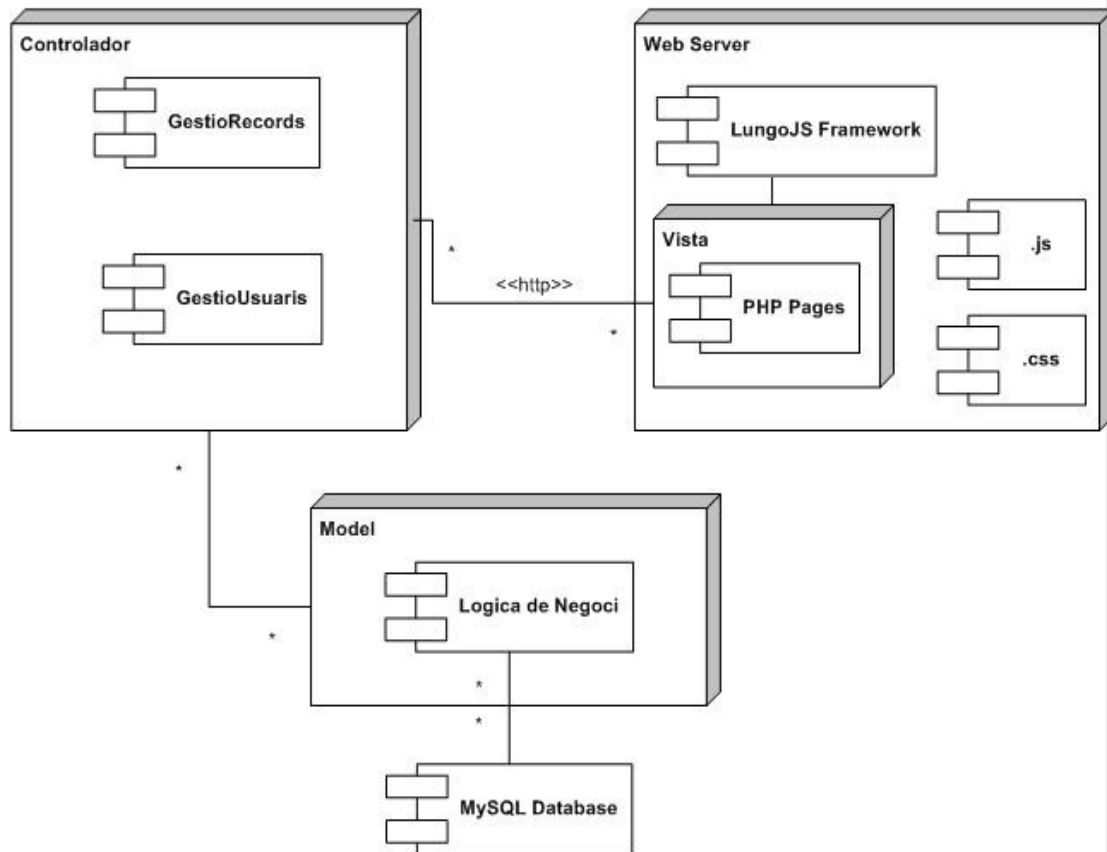


Fig 6: Diagrama de components

L'aplicació s'estructura en els components següents:

- *GestioRecords* que és el controlador encarregat de processar les peticions de la vista relacionades amb els records. En són un exemple l'alta d'un nou record, un llistat dels darrers records, etc.
- *GestioUsuarios* que és el controlador que s'encarrega de processar les peticions de la vista relacionades amb els usuaris. En són un exemple l'alta d'usuari, la petició de col·laboració per a un record d'un altre usuari, l'acceptació de participació, etc.
- *LungoJS* que és el framework encarregat de la gestió d'esdeveniments amb JavaScript, la interfície d'usuari, els elements de la interfície i el format d'estil natiu de l'aplicació web mòbil. Actua juntament amb els fitxers PHP com la Vista amb que l'usuari interactua i realitza peticions.

- Pàgines PHP amb el codi del core de l'aplicació web mòbil. Contenen les classes de les entitats.
- *Lògica de Negoci*, és on resideix la interacció amb les dades, generació de llistats i en general tot el processament amb les dades darrera l'aplicació.
- *MySQL Database*, és on s'emmagatzemen les dades a un tipus de base de dades relacional.

### **6.3 Decisions de disseny**

En aquest apartat enumerem les decisions de disseny preses per al desenvolupament del projecte per tal d'assolir les fites de la planificació inicial i assolir l'èxit del projecte.

Les decisions més importants són:

- Els diferents components del *backend* de l'aplicació s'han programat fent servir PHP com a llenguatge principal degut a la seva robustesa i total integració amb HTML5, CSS3 i JavaScript.
- S'ha optat per un patró de disseny basat en Model-Vista-Controlador ja que aporta flexibilitat, aïllament del codi i seguretat.
- Com a sistema gestor de base de dades s'ha decidit finalment emprar MySQL per tres motius:
  - L'alt grau de familiaritat.
  - Alta compatibilitat amb el llenguatge PHP.
  - Bon rendiment davant bloquejos, transaccions i operacions concurrents.
- LungoJS per les raons exposades al resultat de l'estudi dels diferents frameworks a l'inici del document.
- S'ha decidit prescindir d'algunes de les funcions addicionals que es varen presentar inicialment a la planificació per tal de garantir la plena funcionalitat de l'aplicació. Es registren com a futures ampliacions i/o millores per a una propera versió. Les funcions més importants són:
  - Mode *offline*, per garantir la funcionalitat de l'aplicació quan el dispositiu no disposa de connexió a la xarxa.
  - Directament relacionada amb el mode *offline*, la memòria cau permet emmagatzemar dades a l'àmbit local per quan no es disposa de connexió a la xarxa.

- La sessió d'usuari permanent que es manté al servidor fins que l'usuari tanca l'aplicació, o el que és equivalent, el navegador. El motiu de fer-ho així és per senzillesa, per facilitar l'ús de l'aplicació a l'usuari ja que les aplicacions als dispositius mòbils es mantenen residents en segon pla un cop tancades. L'usuari en tornar-les a obrir accedirà directament dins l'aplicació amb la seva sessió activa.

## **6.4 Diagrama estàtic**

El diagrama estàtic o de classes representa el domini de l'aplicació a través d'un conjunt de classes i les seves relacions.

*{Pàgina següent}*

Repositori de records - Diagrama de classes

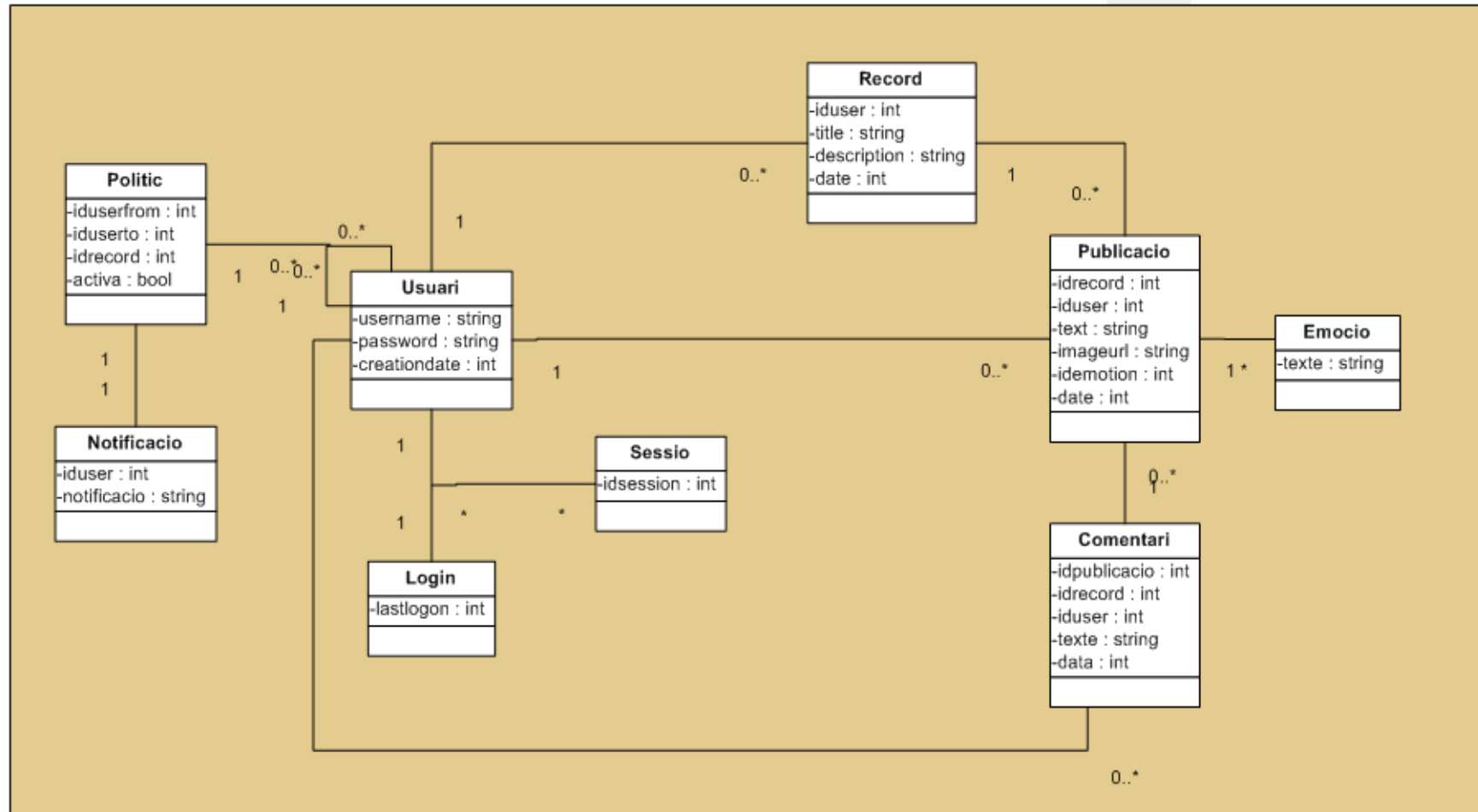


Fig 7: Diagrama estàtic



### 6.5 Diagrama de seqüència

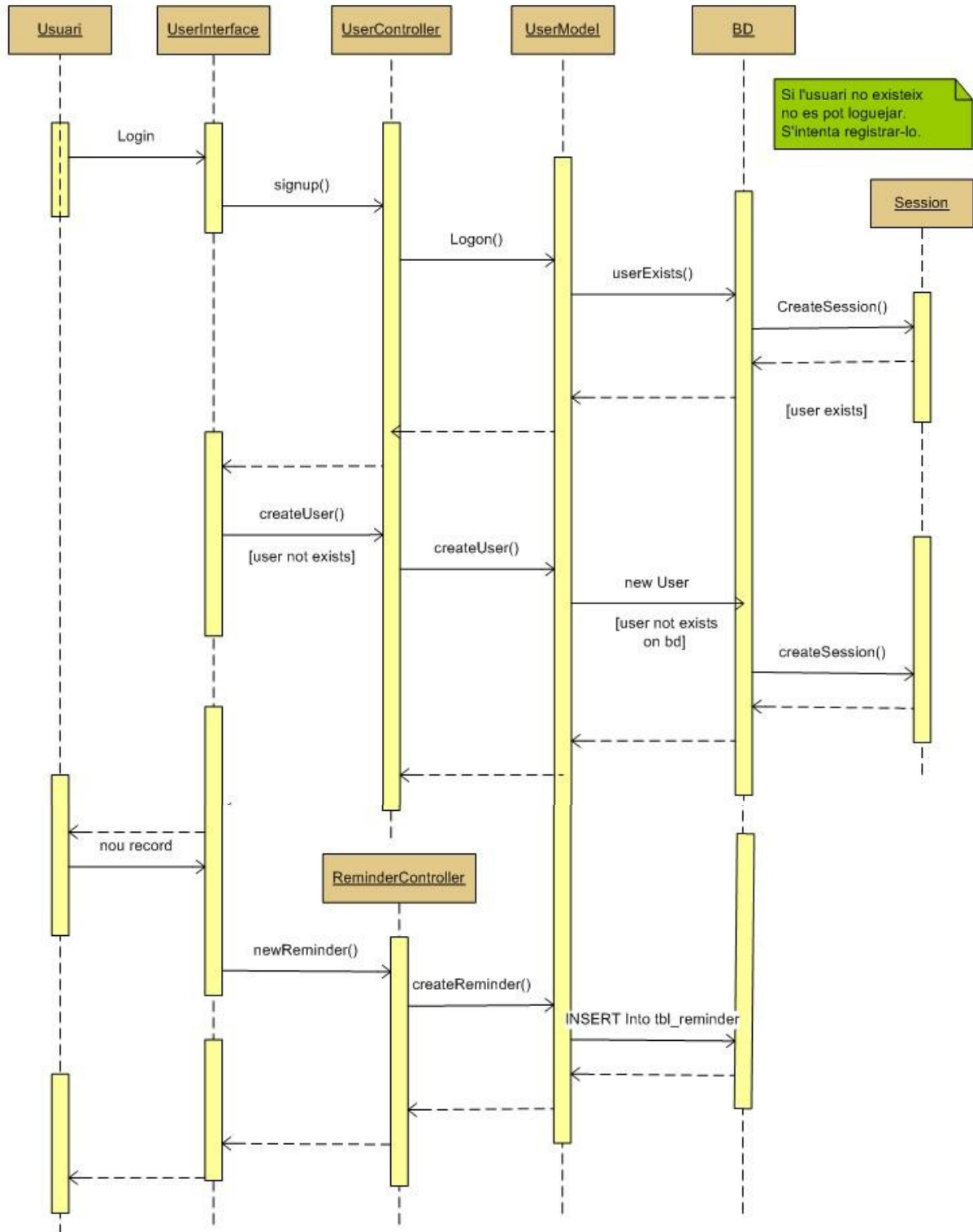


Fig 8: Diagrama de seqüència del CU01

Al diagrama de seqüència anterior s'observen les interaccions entre els objectes i com s'executen les operacions en funció del temps, quins missatges són enviats, per quin objecte, a quin i quan.

S'ha decidit mostrar únicament el diagrama de seqüència corresponent al cas d'ús CU01 (Crear record). Si bé la resta de casos d'ús tenen aproximadament els mateixos objectes, les operacions i les interaccions entre ells són diferents en cada cas.

La seqüència que s'observa és la següent:

- L'usuari obre l'aplicació i omple el formulari d'inici de sessió.
- La vista envia la petició d'inici de sessió al controlador d'usuari que trasllada la petició al model d'usuari perquè faci les comprovacions necessàries (en aquest cas l'existència de l'usuari).
- El model envia l'ordre a la base de dades per a obtenir informació útil de l'existència d'aquest usuari.
- En cas positiu (credencials vàlides), es crea la sessió associada a l'usuari.
- En cas contrari (credencials no vàlides ja que l'usuari no existeix), es retorna fins al controlador que envia l'ordre de registre d'usuari. Un cop creat l'usuari es crea la sessió associada.
- En cas de credencials no vàlides s'informa a l'usuari (usuari o contrasenya incorrecta).
- Un cop validat, l'usuari mitjançant la interfície d'usuari (vista), accedeix al formulari de creació de record i envia una petició d'alta de record.
- La petició d'alta de record arriba al controlador de record que trasllada la petició al model de registre per a procedir amb la creació del nou record.
- El model de record envia l'ordre a la base de dades per a la creació dels registres del nou record.

Al diagrama s'han obviat missatges per a fer més entenedor el gràfic. Tant en cas de credencials no vàlides com errors de validació de camps de formulari són missatges informatius retornats cap a la vista per a informar l'usuari.

## 6.6 Disseny de la persistència

### 6.6.1 Model relacional de la base de dades

tbl\_comments(commentid, rempostid, reminderid, userid, text, regdate)

*rempostid* és clau forànea de tbl\_reminderPost

*reminderid* és clau forànea de tbl\_reminder

*userid* és clau forànea de tbl\_logon

tbl\_emotions(emotionid, emotion)

tbl\_logon(userid, useremail, userpass, regdate)

tbl\_msginbox(msgid, text)

tbl\_notifiers(notifierid, userid, idmsg, read)

*userid* és clau forànea de tbl\_logon

*idmsg* és clau forànea de tbl\_msginbox

*read* admet valors 'Y', 'N' (enum)

tbl\_reminder(reminderid, userid, title, description, regdate)

*userid* és clau forànea de tbl\_logon

tbl\_reminderPost(rempostid, reminderid, userid, text, photopath, emotionid, regdate)

*reminderid* és clau forànea de tbl\_reminder

*userid* és clau forànea de tbl\_logon

*emotionid* és clau forànea de tbl\_emotions

tbl\_session(sessionid, lastlogon)

tbl\_trust(trustid, iduserfrom, iduserto, reminderid, activ)

*iduserfrom* i *iduserto* són claus foràneas de tbl\_logon

*reminderid* és clau forànea de tbl\_reminder

*activ* admet valors 'Y', 'N' (enum).

### 6.6.2 Diagrama de la base de dades

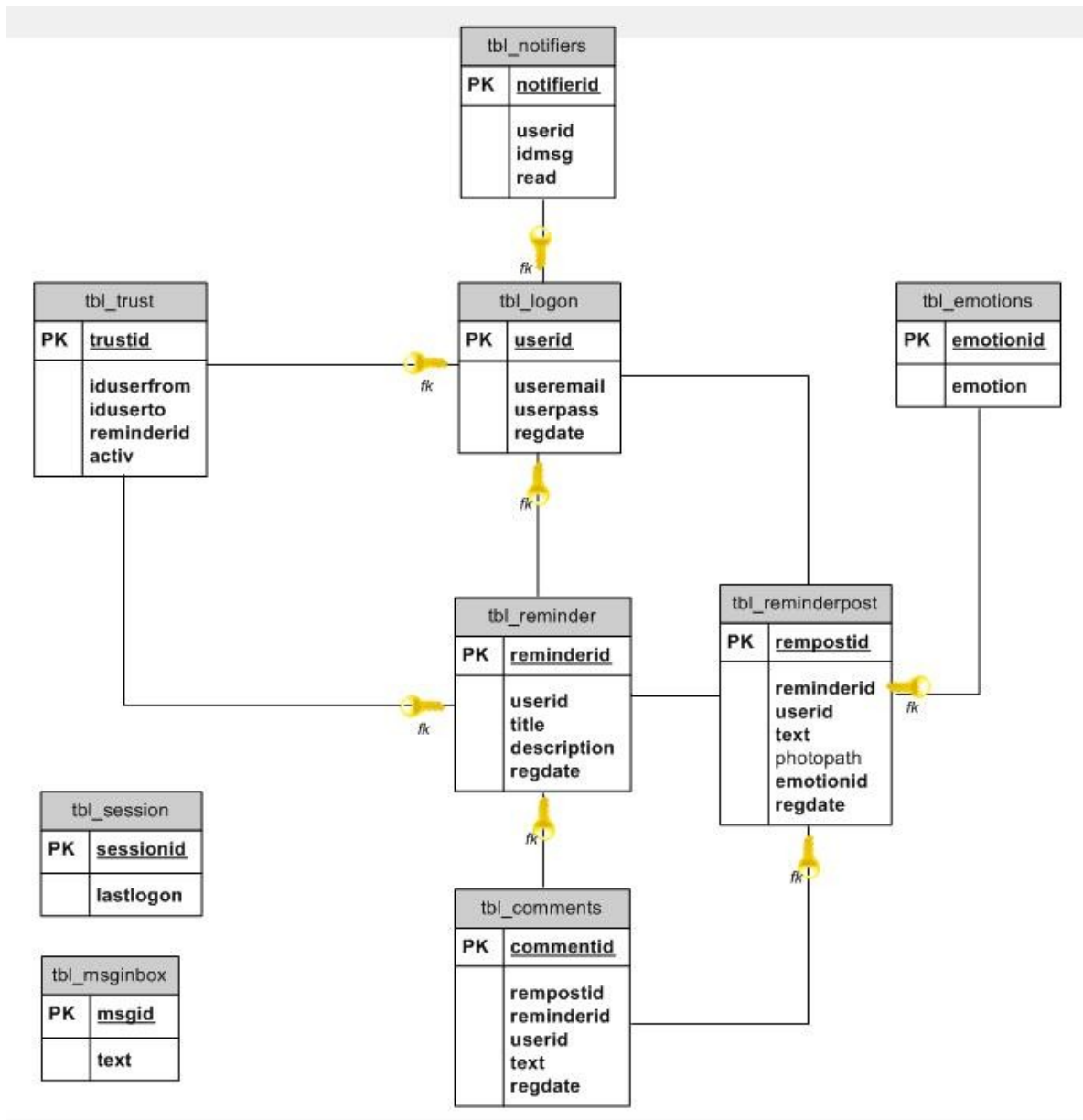
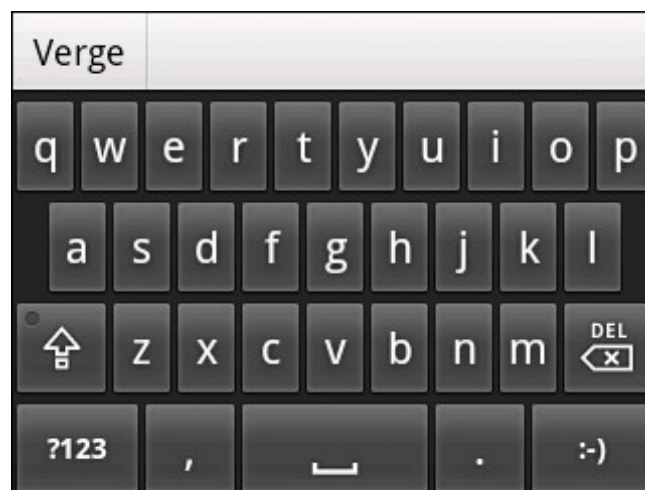


Fig 9: Diagrama E/R

## 6.7 Bones pràctiques de disseny d'interfícies mòbils

Si bé els principals fabricants han publicat documentació al respecte, sembla no haver cap estàndard amb relació a com definir els elements que cal mostrar per pantalla a la nostra aplicació.

Gairebé la totalitat dels dispositius actuals de darrera generació tenen unes dimensions de pantalla importants així com la tecnologia multi-tàctil associada que permet l'usuari interactuar amb les aplicacions. Tot i això, en quant a la introducció de dades amb el teclat virtual, encara hi ha molta feina a fer ja que tenen unes dimensions que impedeixen actuar amb precisió i pot donar lloc a confusions amb els elements a la pantalla (per exemple la confusió en activar una lletra propera a la que realment es vol activar).



*Fig 10: Teclat virtual qwerty a un terminal amb Android*

Els fabricants, no al marge d'aquesta situació, proposen guies d'estil d'interfícies d'usuari per als desenvolupadors d'aplicacions per als seus dispositius que garanteixin una interacció humana el més fluïda i àgil possible, minimitzant els errors associats a la grandària dels elements per pantalla.

En el nostre projecte, però, desenvoluparem una aplicació que haurà de funcionar sobre un navegador web i per tant, haurem d'intentar vèncer els obstacles que puguin sorgir-ne. La interacció de l'usuari i la nostra aplicació estarà limitada a l'àmbit navegador. Aquest és un dels motius de pes pels quals l'aplicació ha de garantir una bona experiència d'ús als diferents navegadors del mercat.

Tot i no desenvolupar amb cap SDK en concret, es descriu a grans trets i amb l'objectiu d'extreure una idea generalitzada d'allò que recomanen els principals fabricants de dispositius en quant a interfície d'usuari. Els principals fabricants i entorns són:

- Apple amb iOS: recomana emprar elements de **44 x 44 punts** i posa com a exemple d'interfície usable la seva aplicació de calculadora.
- Windows amb Windows Phone: manté la guia més completa en aquest àmbit. Proposen **9 mm** com la grandària ideal de l'element més petit a la pantalla per garantir – com a resultat d'hores de proves als laboratoris – que es reduïen els errors al 1,6%. Recomanen, a més, deixar un marge mínim de **2 mm** entre dos elements i definir un àrea de pulsació major que la capa visible de l'element (per evitar errors quan l'usuari no toca amb precisió el centre de l'element).
- Nokia amb Symbian<sup>8</sup>: el fabricant finlandès aposta per una visió pràctica al problema de la interacció recomanant que els elements tinguin com a mínim la grandària i espaiat equivalent a l'estàndard de la superfície d'un dit. A més, fan una distinció per als elements que cal arribar amb el polze (**8 mm** amb **2 mm** d'espaiat) i aquells amb els que interactuem amb el dit índex (**7 mm** amb **1 mm** d'espaiat).
- Canonical amb Ubuntu: recomanacions molt semblants a Nokia. Recomanen grandàries d'entre **10 i 20 mm**.
- Google amb Android: Google recomana que els elements han d'estar distribuïts dins de blocs de **48 dp** (píxels independents d'alta densitat equivalents a un píxel d'una pantalla de 160 ppp). Dins d'aquests blocs amb una grandària equivalent a **9 mm**, recomanen que l'element menor ha d'ocupar uns **32 dp** amb una separació mínima entre elements de **8 dp**.

Un estudi del MIT de l'any 2003 va obtenir la mida estàndard de la base i la punta del dit. El resultat és que un dit estàndard hauria de mesurar entre **10 i 14 mm** de base i entre **8 i 10 mm** a la punta del dit.

El nostre projecte no seguirà cap recomanació de cap fabricant en concret doncs la naturalesa del mateix (multi-plataforma) ha de permetre interactuar l'usuari independentment del dispositiu amb que es treballi. Per tant, emprarem un disseny lògic i raonable acord a la grandària de les pantalles actuals i delegant la responsabilitat al frameworks per a altres conceptes més intrínsecs en l'àmbit del disseny.

---

8 Fruit de la darrera aliança, Nokia passarà a proveir terminals per a Windows Phone.

## 6.8 Prototips de la interfície d'usuari

Es farà servir una eina per a realitzar *mockups*<sup>9</sup> per al disseny del prototip de l'aplicació. Amb aquesta eina s'obtindrà un disseny inicial de les possibles interfícies resultants i així s'obtindrà una visió global de com serà l'aplicació un cop acabada. Per al disseny del prototip no cal cap tipus de codificació ni cap disseny d'interfícies. Aquest tipus de dissenys són molt importants als tests unitaris en metodologies àgils. Farem servir la solució de Balsamiq anomenada Balsamiq Mockups, una aplicació en Adobe Air de pagament però amb una versió de prova funcional al 100%. És una aplicació *drag & drop*.

L'avantatge de realitzar un mockup abans de començar directament amb l'aplicació és que et permet experimentar diferents estructures per a la interfície així com trobar la millor fórmula per garantir una experiència final profitosa en quant a usabilitat, accessibilitat i llegibilitat.

---

9 Model bàsic de disseny d'interfícies d'usuari per mostrar un possible resultat final de l'aplicació.



Fig 11: Mockups del prototip d'interfícies



## 7. Implementació

Com ja es va descriure a l'anàlisi i el posterior estudi dels diferents frameworks mòbils, l'objectiu d'aquest projecte final de carrera era el desenvolupament d'un repositori de records.

Al llarg d'aquesta fase es realitzarà la implementació d'una plataforma d'enregistrament de records anomenada *Recoly*, concretament un repositori compartit, emprant al llarg del desenvolupament un patró de disseny Model Vista Controlador (MVC) i llenguatge PHP amb MySQL per a la persistència de les dades.

Tot i les possibilitats que ofereix LungoJS en quant a JavaScript i marcatge dinàmic, no s'ha aprofitat la totalitat de les característiques que ofereix LungoJS, doncs per una banda requeria la implementació d'un servidor RESTful i per altra, la limitació de temps per al desenvolupament ha condicionat la implementació exclusiva del core de l'aplicació. Es deixa oberta la possibilitat d'ampliar l'aplicació per afegir-hi millores a una propera versió<sup>10</sup>.

Al llarg d'aquest apartat es descriuen cadascun dels components que formen l'aplicació, les seves característiques i les decisions que s'han pres en l'àmbit de desenvolupament per a assolir el bon funcionament de l'aplicació.

### 7.1 Controladors

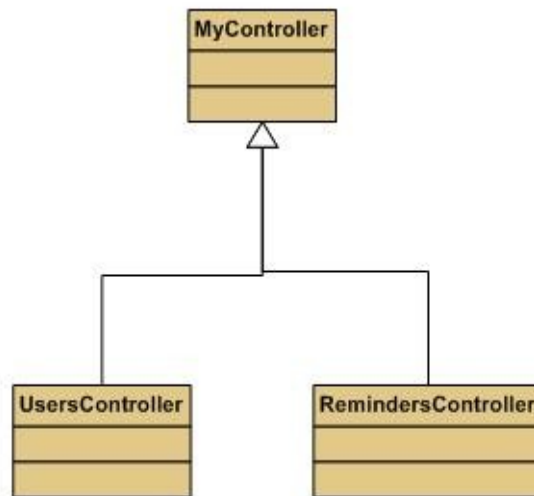
S'han implementat dos controladors per a processar les peticions de les vistes de l'aplicació:

1. RemindersController: s'encarrega de processar les peticions que li arriben de les vistes en relació amb operacions sobre el domini dels records.
2. UsersController: s'encarrega de processar les peticions sobre el domini dels usuaris.

Tots dos controladors hereten d'un controlador base anomenat *MyController*, sense cap mètode implementat.

---

<sup>10</sup> Consultar punt 8 *línies futures*



*Fig 12: Controladors de l'aplicació*

Els controladors envien les dades juntament amb les peticions cap als models de l'aplicació perquè aquests accedeixin a les dades, facin les consultes necessàries per completar la petició i li retornin les dades amb el resultat.

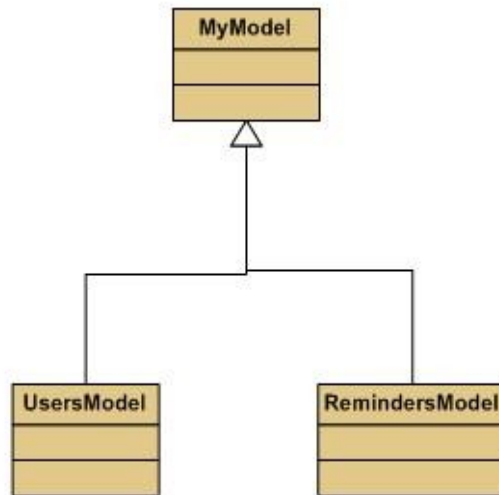
## 7.2 Models

S'han implementat dos models per a implementar la lògica de negoci dels serveis de l'aplicació. Els models creats contacten amb els components de persistència per al tractament de dades amb el SGBD adjacent.

Els models disponibles són:

3. RemindersModel: s'encarrega d'implementar la lògica de negoci dels serveis relacionats amb els records. Accedeix a la capa de persistència per al tractament de dades requerit.
4. UsersModel: s'encarrega d'implementar la lògica de negoci dels serveis relacionats amb els usuaris i la seva gestió. Accedeix a la capa de persistència per al tractament de dades requerit.

Tots dos models hereten d'un model base anomenat MyModel, sense cap mètode implementat.



*Fig 13: Models de l'aplicació*

### 7.3 Vistes

L'aplicació compta amb diferents vistes per a rebre la informació fruit de la interacció amb l'usuari:

- Index.php: pantalla principal de l'aplicació. Mostra el formulari d'inici de sessió.
- Main.php: pantalla principal de l'aplicació un cop l'usuari ha iniciat sessió. Inclou una llista de les darreres publicacions dels usuaris que interactuen amb l'aplicació. Es poden veure els records publicats i afegir-hi comentaris.
- MyNotifiers.php: mostra les notificacions pendents de l'usuari respecte les sol·licituds de participació d'altres usuaris als seus records.
- MyReminders.php: conté el llistat dels records creats per l'usuari.
- NotifierInfo.php: conté la informació de la sol·licitud de participació. Permet acceptar o rebutjar la sol·licitud.
- ReminderForm.php: formulari d'alta d'un nou record.
- ReminderInfo.php: pantalla amb la informació en detall del record seleccionat.
- ReminderPost.php: formulari d'alta d'una nova publicació associada a un record.
- Search.php: formulari de cerca de records. Es mostra una llista amb els resultats coincidents obtinguts.

## 7.4 Autenticació d'usuari

L'aplicació requereix d'una validació d'usuari inicial per poder accedir a totes les funcionalitats així com interactuar amb la resta d'usuaris. La pantalla inicial d'inici de sessió ens permet autenticar-nos davant el sistema i/o donar-nos d'alta en cas que no existeixi l'usuari. En aquesta pantalla es valida tant l'usuari i contrasenya com el format dels camps d'entrada (format e-mail, camps buits, etc.). En cas d'error s'informa a l'usuari.

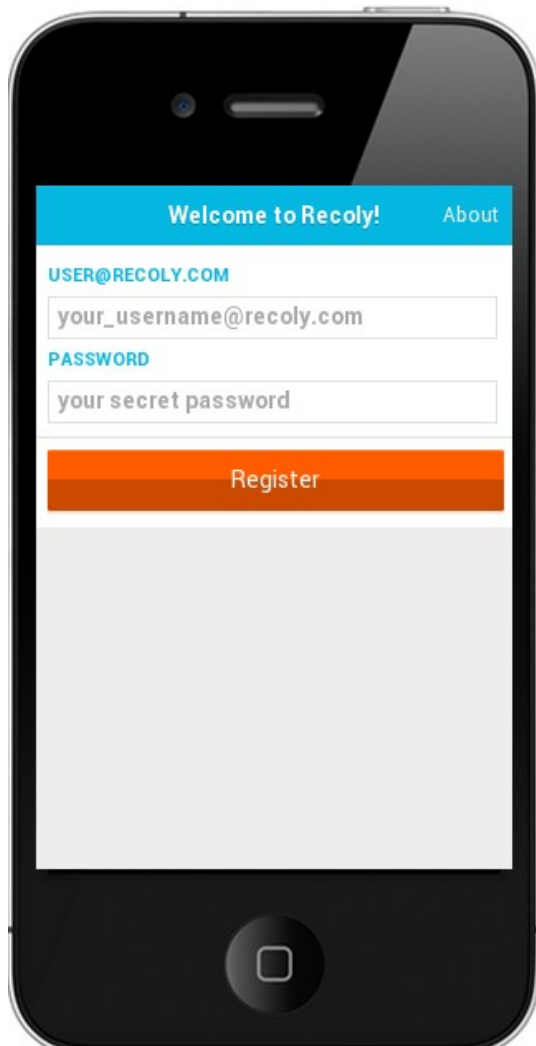
Com ja s'ha comentat a l'apartat de disseny, s'ha optat per un tipus de sessió permanent. D'aquesta forma, no hi ha cap botó *logout* on l'usuari pugui sortir de l'aplicació. L'aplicació en cas de quedar en segon pla, permet a l'usuari continuar l'ús en qualsevol moment sense la necessitat d'haver d'autenticar-se de nou.

El formulari d'entrada s'encarrega de l'autenticació de les dades d'usuari. En concret, es comprova tant si omple tots els camps, com si un dels camps no compleix el format correcte. La validació del format correu al camp usuari s'ha realitzat mitjançant l'expressió regular següent:

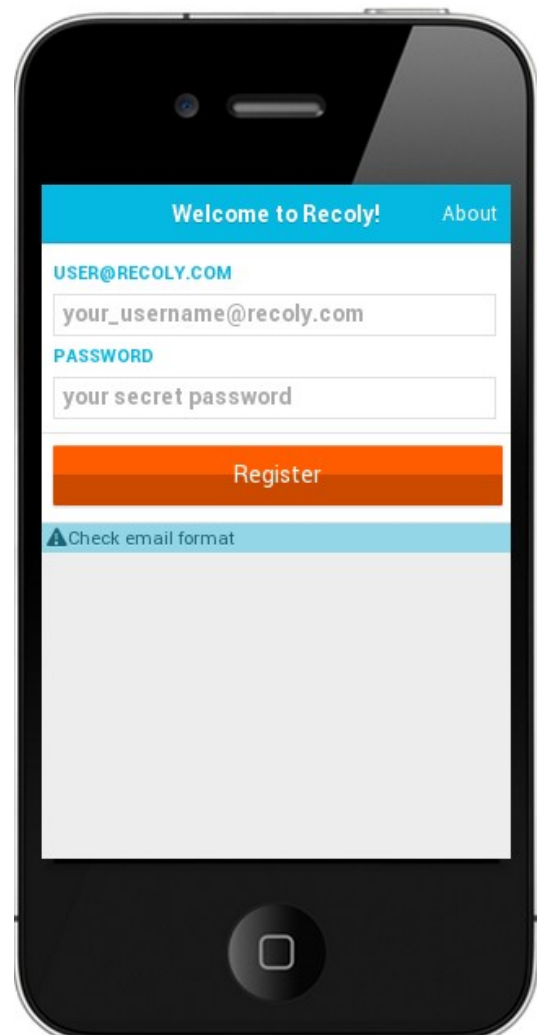
```
"^[a-z0-9-]+(\.[a-z0-9-]+)*@[a-z0-9-]+(\.[a-z0-9-]+)*(\.[a-z]{2,3})$"
```

Tant en el cas en que hi hagi errors de validació (camps buits i/o format incorrecte) com error d'autenticació (usuari existent però validació sense èxit), s'informa a l'usuari mitjançant un panell informatiu.

Un cop la vista ha rebut la petició a la vista, aquesta l'envia cap al controlador adient.



*Fig 14: captura formulari inici sessió*



*Fig 15: captura formulari inici sessió amb error*

Un cop validades les credencials d'usuari, s'accedeix a la pantalla principal de l'aplicació on es visualitzen les darreres novetats de la comunitat d'usuaris, és a dir, les darreres publicacions de records (text i fotos) d'altres usuaris.

## 7.5 Darreres publicacions

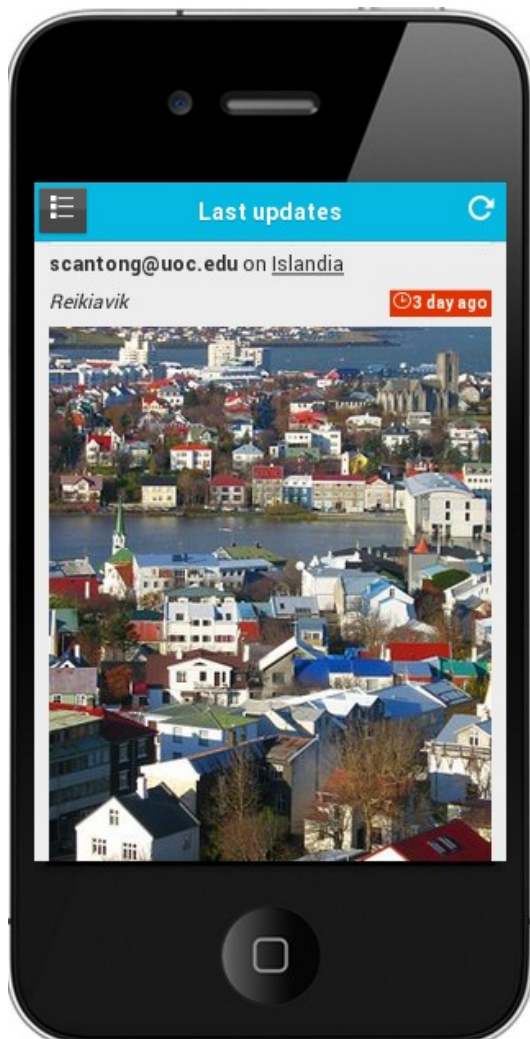


Fig 16: Captura llista publicacions

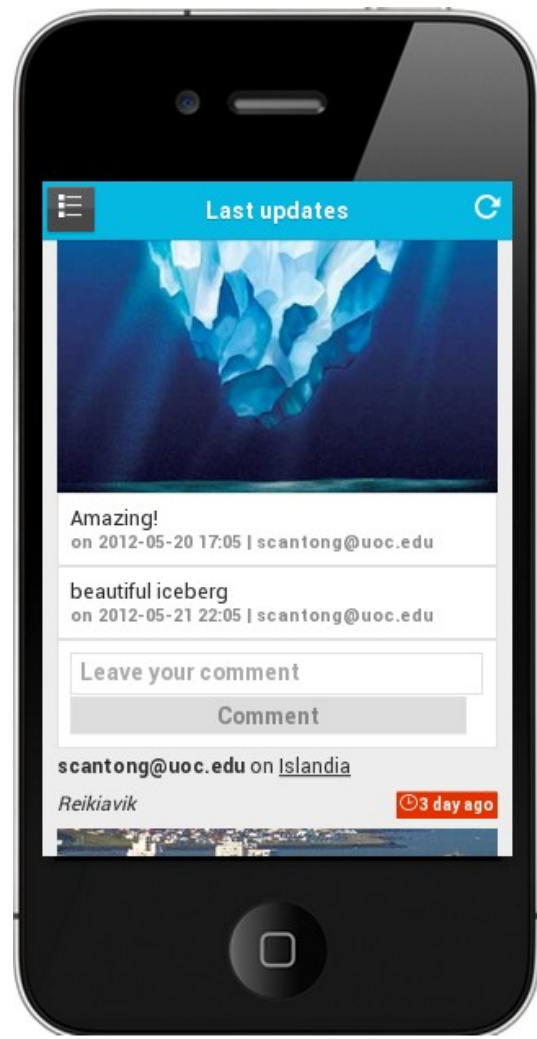
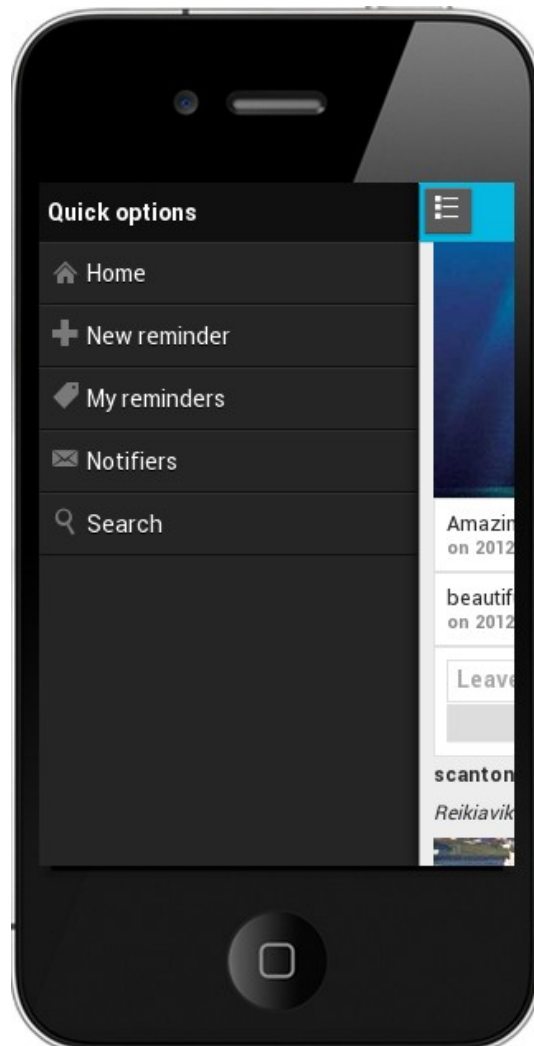


Fig 17: Captura comentaris publicació

En aquesta pantalla de novetats es mostren les darreres publicacions de records dels usuaris. Es permet a l'usuari comentar les publicacions així com actualitzar la vista per obtenir més novetats en cas que un altre usuari estigués publicant. Es mostra l'usuari que publica, el record on es publica la foto/text, la descripció i el temps que fa que es va publicar.

A la part superior tenim un botó tipus *aside* molt emprat a desenvolupament per a dispositius mòbils com smartphones i tablets que permet mostrar un panell lateral a mode menú que desapareix segons la mida de la pantalla o *viewport* del dispositiu sobre el qual s'està obrint la nostra aplicació.

Es mostra una petita transició en obrir i tancar el panell lateral, i es pot fer servir en qualsevol moment ja que la pantalla activa en aquell moment queda en segon pla.



*Fig 18: Captura menú principal aplicació*

## 7.6 Nou Record

Es mostra un formulari per omplir els camps títol i descripció del nou record. Es valida que els camps no estiguin buits. Un cop creat el nou record es redirigeix a l'usuari automàticament a la pantalla de nova publicació ja que es considera que són les publicacions les que donen sentit a un record.

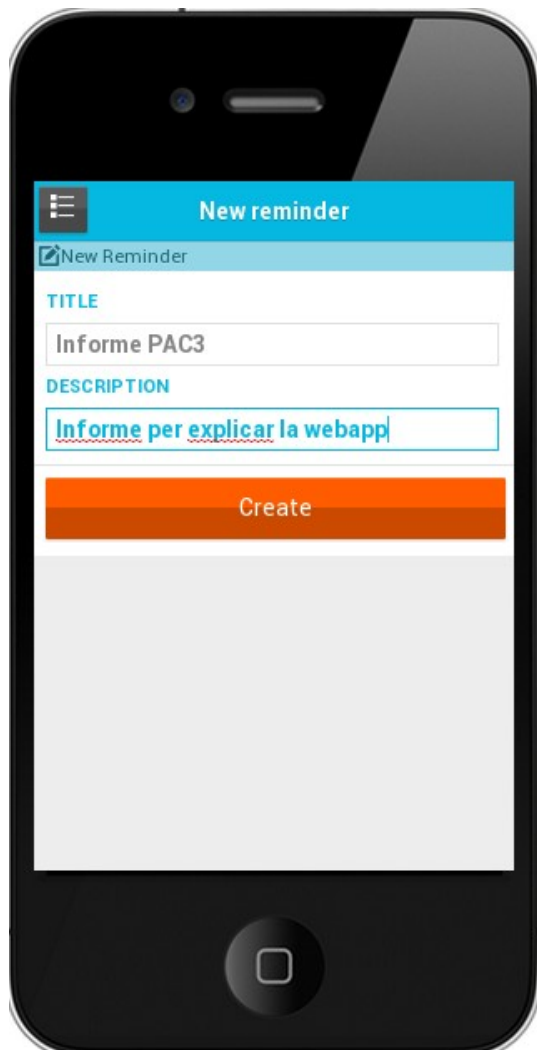


Fig 19: Captura nou record

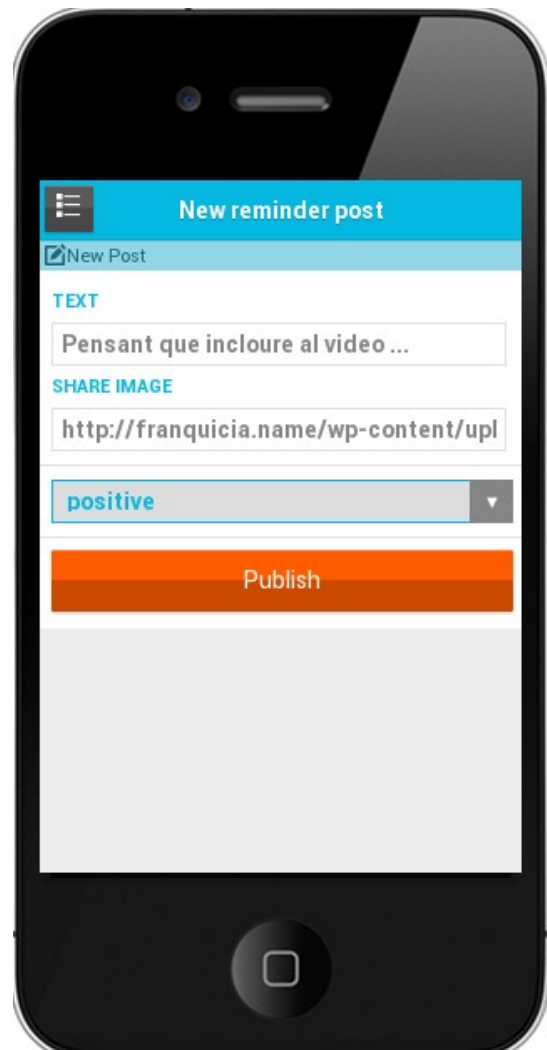


Fig 20: Captura nova publicació

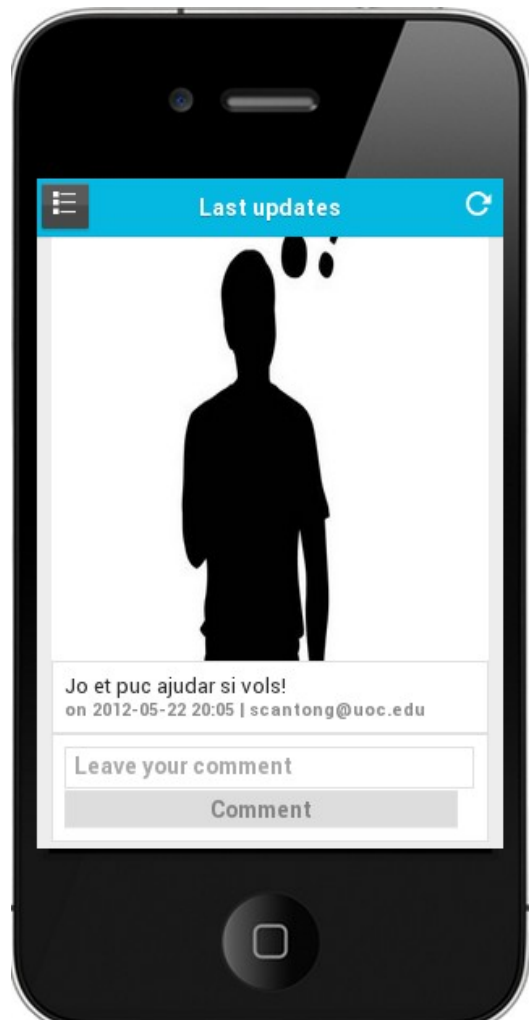
Al formulari de publicació dins el record es pot escriure un text i adjuntar una imatge (en aquesta versió únicament es permet imatges mitjançant un recurs a la xarxa. A una versió posterior es pot incloure el suport per accedir al dispositiu des d'on s'executa). L'usuari ha d'escollir una emoció d'entre la llista predefinida que li ofereix l'aplicació. Al moment de publicar es validen les dades. Un cop publicat, s'envia a l'usuari a la pantalla inicial de



novetats on pot veure la seva darrera publicació i on pot comentar-la així com observar com la resta d'usuaris comenten la seva publicació.



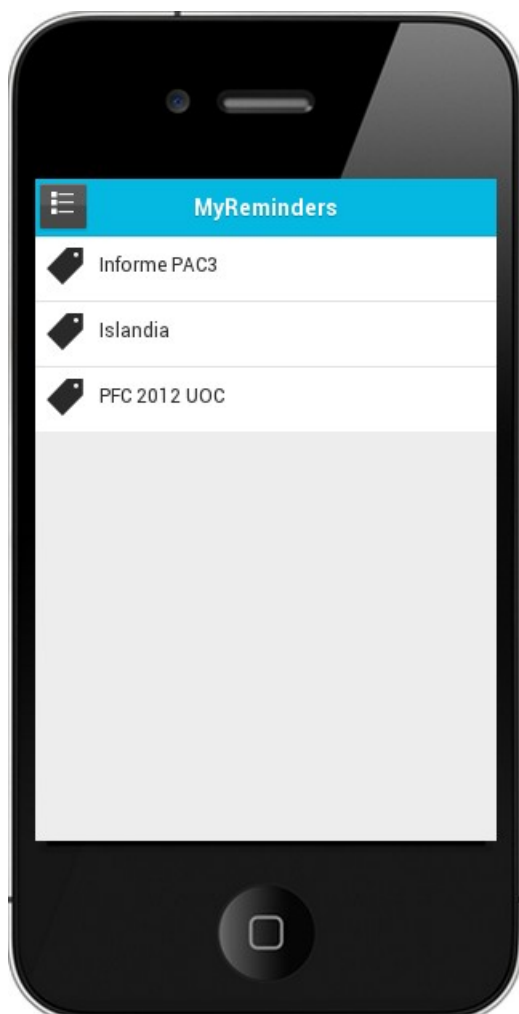
*Fig 21: Captura publicació creada*



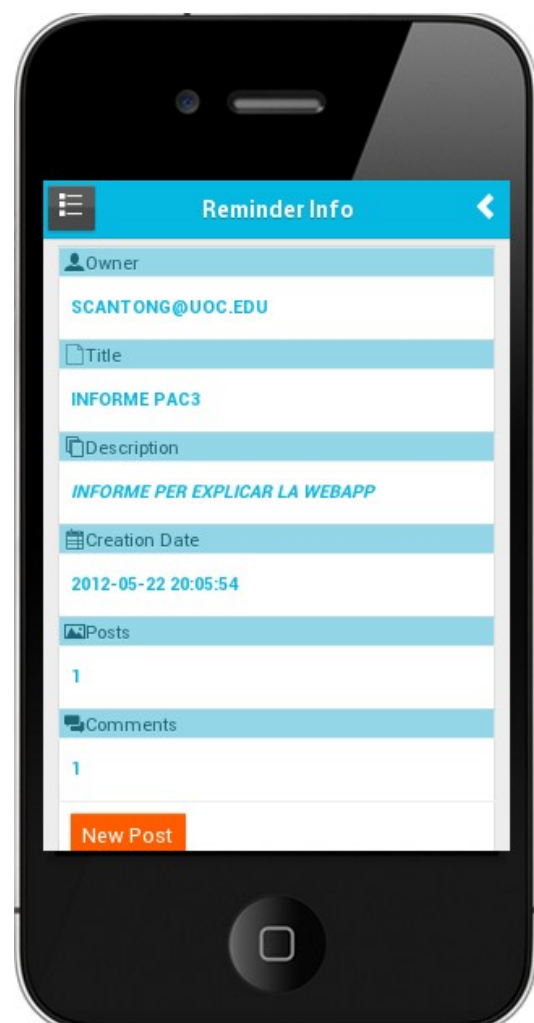
*Fig 22: Captura publicació comentada per usuari*

## 7.7 Els meus records

En aquesta pantalla es mostren els records que pertanyen a l'usuari. Apareixen a una llista dinàmica amb scroll vertical. Es pot accedir a cada record i veure la seva fitxa en vista detall. A la pantalla detall es mostra un resum del record com el propietari, el títol, la descripció, la data de creació, les publicacions i els comentaris. És des d'aquí on l'usuari pot tornar a afegir una publicació en el record.



*Fig 23: Captura llista records*



*Fig 24: Captura informació del record en detall*

## 7.8 Cerca de records

En aquesta pantalla es permet a l'usuari realitzar cerques per a trobar records (tant seus com d'altres usuaris) i accedir a la fitxa amb la informació del record i publicar contingut sempre que tingui drets sobre aquell record, és a dir, tingui la propietat del record o l'usuari propietari del record li hagi habilitat l'edició sobre el seu record. És a la pantalla de fitxa del record on l'usuari que vol publicar envia una sol·licitud a l'usuari propietari del record. L'usuari propietari rebrà a la seva safata de notifikacions la sol·licitud de l'usuari en qüestió. Pot acceptar o denegar la sol·licitud.

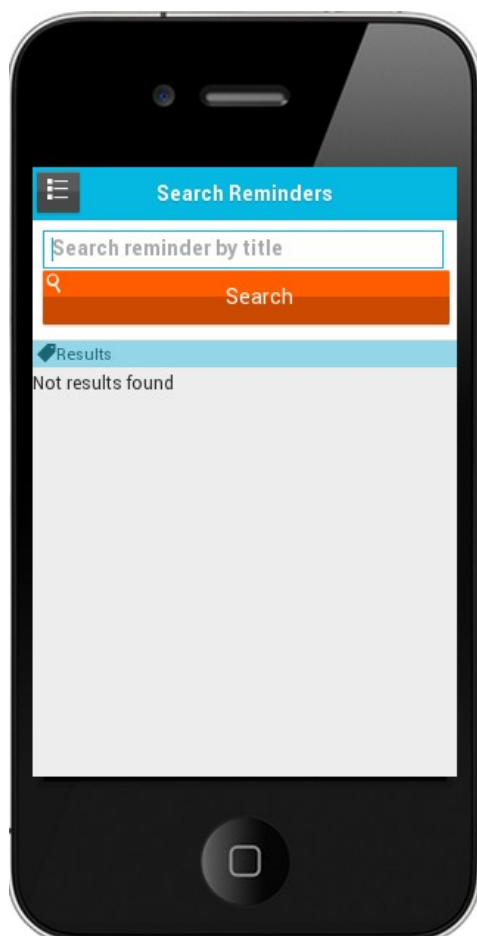


Fig 25: Captura cerca de records

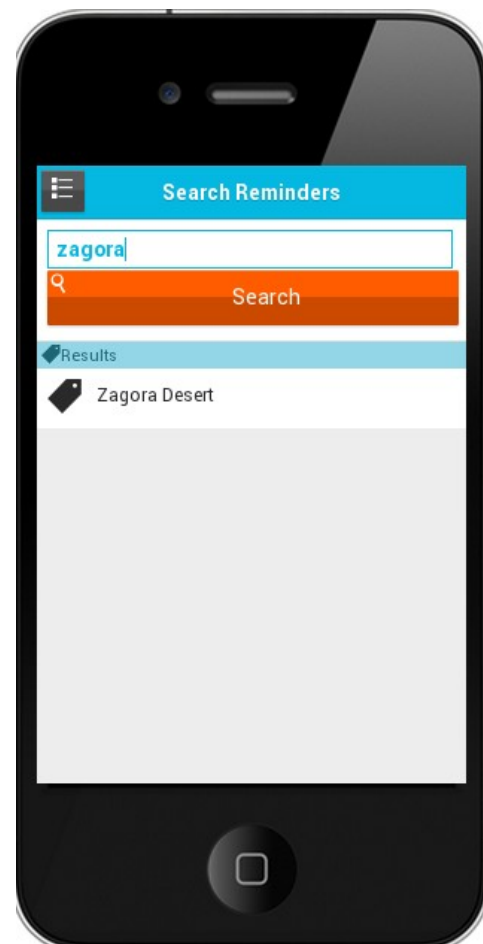


Fig 26: Captura resultats de cerca

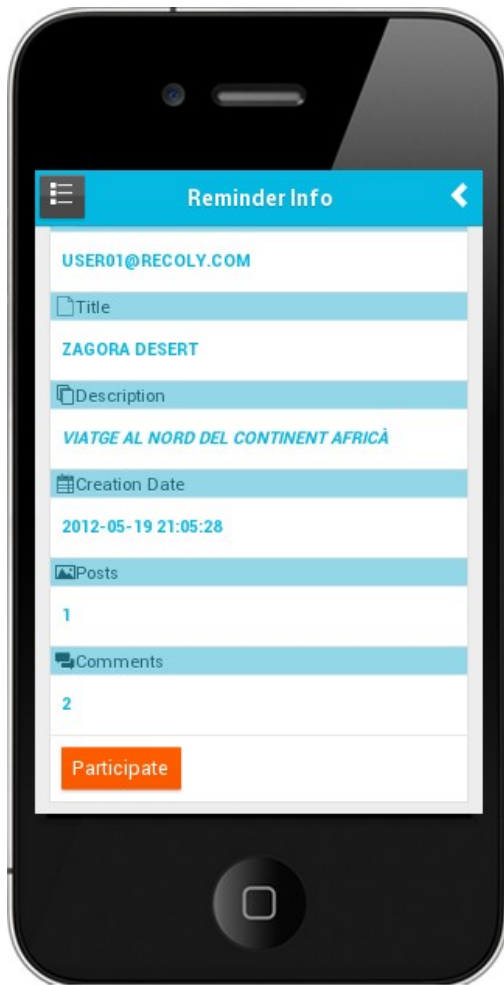


Fig 27: Captura sol·licitud participació

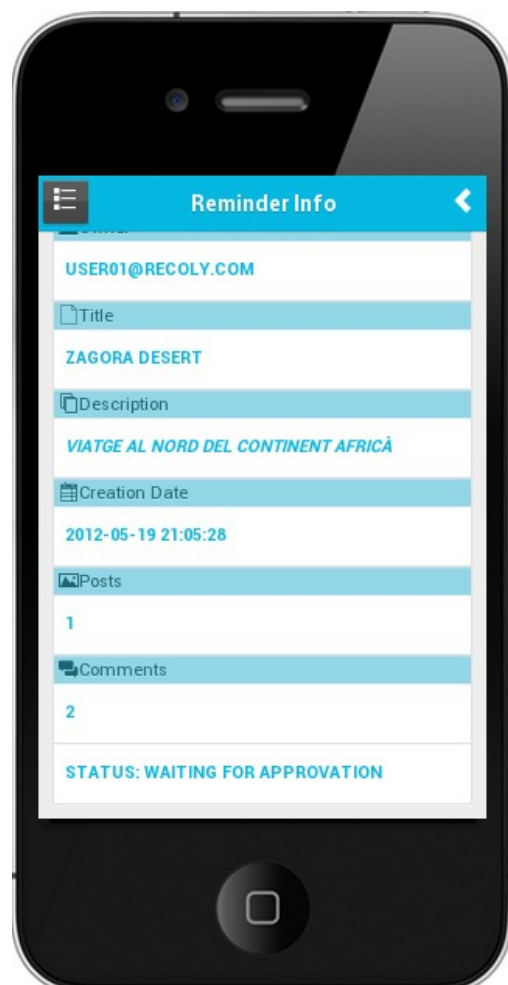


Fig 28: Captura Sol·licitud participació enviada

Observem que en el moment d'enviar la sol·licitud de participació, l'estat canvia a *pendent d'aprovació*. Ara l'usuari en qüestió rebrà a la seva safata de notificacions la sol·licitud per a aprovar-la o rebutjar-la.

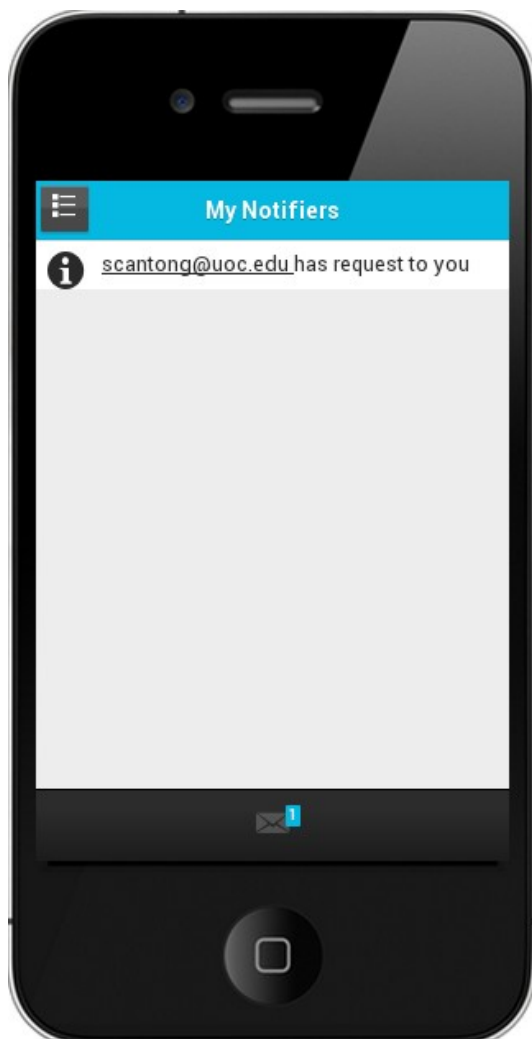


Fig 30: Captura safata de notificacions

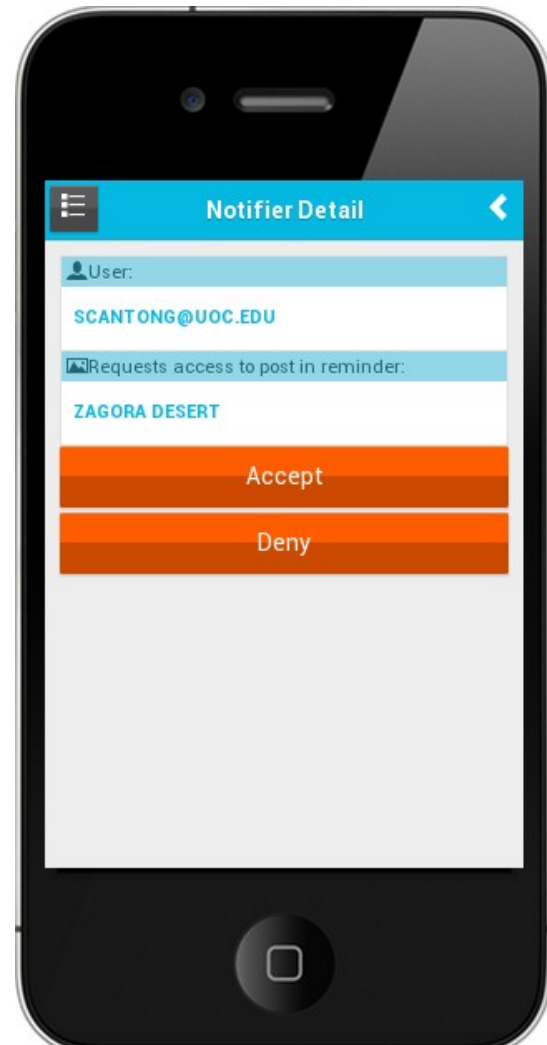


Fig 29: Captura resposta a sol·licitud

Si l'usuari accepta la meua sol·licitud, en tornar a la pantalla de detall del record anomenat *Zagora* veuré que s'ha activat l'acció de publicació i em permetrà publicar contingut al voltant d'aquest record (creat prèviament per un altre usuari). Si l'usuari no accepta ni rebutja la sol·licitud, l'altre usuari no pot enviar cap altra més i ha d'esperar a que es resolgui. En cas que es denegui, es podrà tornar a intentar enviar una altra sol·licitud de participació a l'usuari.

Una possible ampliació de l'aplicació seria la possibilitat d'enviar missatges interns entre usuaris per a facilitar la comunicació directa.

## 8. Línies futures

Al llarg del Treball Final de Carrera s'han analitzat els diferents frameworks disponibles per al suport en el desenvolupament d'aplicacions web per a dispositius mòbils. Fruit d'aquest estudi, s'han destacat aquells punts forts de cadascun d'ells i s'han contrastat vers la resta per obtenir coneixement útil a l'hora de triar quin framework aporta més valor afegit al desenvolupament segons uns criteris de qualitat definits prèviament.

Seguidament s'ha dissenyat i desenvolupat una aplicació web mòbil per enregistrar records i compartir-lo amb la resta d'usuaris. S'ha seguit un desenvolupament basat en un patró de disseny MVC degut als diversos beneficis que aporta al desenvolupament com per exemple l'aïllament de capes, el manteniment, etc.

S'han integrat tecnologies de desenvolupament com PHP i MySQL per al backend, i JavaScript, HTML5 i CSS3 per al frontend delegant la gestió d'aquestes tecnologies al framework LungoJS.

La metodologia emprada per a la fase d'implementació ha estat un model basat en iteracions doncs a cada iteració s'intentava disposar d'una funcionalitat. Respecte la planificació inicial, hi ha hagut lleugers canvis en quant a funcionalitats inicials doncs la corba d'aprenentatge de les tecnologies emprades ha estat més accentuada que no pas es va planificar inicialment.

Per a poder fer una bona verificació de l'aplicació, s'ha dissenyat un entorn de proves aïllat format per un emulador web on s'ha incrustat l'execució de l'aplicació desenvolupada. L'aplicació ha respost perfectament al joc de proves i a més tant des del mateix emulador com des d'un terminal Android de l'alumne.

El següent pas seria l'optimització de l'aplicació amb millores que aportin un valor afegit fruit de l'experiència adquirida al llarg del desenvolupament del projecte.

Per a millorar el rendiment i fluïdesa de l'aplicació es podria incorporar l'ús d'un mode offline fent servir la memòria cau d'HTML5. Aquest mode requeriria d'un emmagatzemament local al dispositiu per l'accés a les dades quan no s'hi està connectat a la xarxa.

També seria interessant millorar la gestió de les dades amb que treballa l'usuari com és el seu perfil d'usuari (modificar nom, canviar correu associat, contrasenya, etc.). A més, seria bo afegir la possibilitat d'editar un record existent, permetre l'alta d'emocions per part de l'usuari per tal de personalitzar més els records associats.

Per optimitzar una mica més la solució, es podria implementar la geolocalització associada a cada publicació. D'aquesta forma, els usuaris podien disposar de la ubicació des d'on es registra cada record.

Més difícil però molt interessant seria introduir la gestió de cercles de confiança a l'estil google+ per tal que l'usuari pugui definir perfils d'usuaris amb qui vol compartir informació. En aquesta primera versió únicament hi ha usuaris de confiança sobre un record.

## 9. Conclusions

En aquest apartat s'exposen les diferents conclusions que es deriven dels diferents treballs realitzats durant la realització del projecte.

La realització d'aquest projecte final de carrera m'ha servit per arribar a una sèrie de conclusions tècniques que es presenten a continuació:

- La integració de LungoJS amb el llenguatge PHP tot seguint un patró de disseny MVC no és del tot perfecte. LungoJS sembla estar dissenyat per treballar segons el seu model intern basat en esdeveniments i serveis amb crides asíncrones. Tot i això, crec que el resultat final ha estat satisfactori.
- El mòdul d'autenticació presenta certs interrogants en quant a com cal mantenir la sessió de l'usuari durant el temps el qual l'aplicació es troba en execució, ja sigui en segon pla o no. En el nostre cas no s'ha optat per cap botó logout tradicional doncs s'ha volgut copiar el funcionament de les aplicacions tradicionals natives que guarden l'estat permanentment per evitar l'usuari autenticar cada cop que obre l'aplicació. En aquest cas es proposa com a futura millora la creació de galetes com a mesura addicional o una base de dades reduïda a la part client.
- Tot i que la premissa de LungoJS garanteix que no calen coneixements avançats de JavaScript per al disseny del prototip, per al funcionament intern i consum de serveis externs sí que cal un bon nivell de JavaScript.

Finalment, i a títol personal, comentar que ha estat una experiència molt profitosa en quant a l'estudi de tecnologies. El sector web mòbil encara és un territori jove i emergent que acaba d'arribar per instaurar-se. Tot i la poca experiència amb aquest tipus de tecnologies, crec que hi ha una bona feina al darrera que s'ha vist reflexat amb el bon funcionament de l'aplicació.

## 10. Glossari

La següent taula conté aquella terminologia tècnica, tant acrònims, abreviatures o nomenclatura originària de l'anglès amb la seva definició, necessària per a la comprensió del document:

<b>SGBD</b>	Sistema gestor de base de dades
<b>Webapp</b>	Aplicació web
<b>Screencast</b>	Gravació de la sortida de la pantalla d'un ordinador amb la locució del narrador
<b>Timing</b>	Interval de temps en que s'han distribuït unes determinades tasques per assolir l'èxit a un projecte
<b>Testing</b>	Fase de proves per obtenir informació vers la qualitat del producte estudiat
<b>Teclado qwerty</b>	Distribució de les tecles als teclats, més comuna a la majoria de dispositius
<b>Layout</b>	Esquema de distribució d'elements
<b>Grid</b>	Estructura de distribució en dues dimensions per estructurar la informació
<b>Api</b>	Interfície de programació d'aplicacions
<b>Packaging</b>	Procés de preparació del codi d'una aplicació per a la seva distribució
<b>Look &amp; feel</b>	Aparença
<b>Plugin</b>	Complement d'un programari base que realitza una funció concreta
<b>Alpha</b>	Versió inicial d'un programa que no conté la totalitat de funcionalitats que s'han previst per la versió final
<b>Uptime</b>	Temps que un servei es troba actiu
<b>Overhead</b>	Càrrega, saturació
<b>Frontend</b>	Part del programari que interactua amb l'usuari
<b>Backend</b>	Part del programari que processa les dades des de l'entrada al frontend
<b>Dp</b>	Unitat abstracta que es basa en la densitat de píxels en pantalla
<b>Restful</b>	Serveis de transferència d'estats basat en el protocol HTTP i els seus mètodes (GET, POST, PUT, DELETE)
<b>Viewport</b>	Regió rectangular de visió de dispositius gràfics com pantalles de smartphones, pda's, tablets, etc.



## 11. Fonts d'informació

Per a la realització d'aquest Treball Final de Carrera s'han consultat les fonts d'informació següents:

[1] World Wide Web Consortium (W3C)

[en línia] <http://www.w3.org>

[2] Sencha. HTML5 Framework for Desktop and Mobile Devices

[en línia] <http://www.sencha.com/>

[3] jQuery Mobile Framework

[en línia] <http://jquerymobile.com/>

[4] jQTouch. A Zepto/jQuery plugin for mobile web development on the iPhone, Android, iPod Touch, and other forward-thinking devices

[en línia] <http://jqtouch.com/>

[5] Titanium Mobile Platform | Appcelerator

[en línia] <http://www.appcelerator.com/products/titanium-mobile-application-development/>

[6] LungoJS. HTML5 Mobile Framework... and Stuff

[en línia] <http://www.lungojs.com/>

[7] The-M-Project

[en línia] <http://the-m-project.net/>

[8] DHTMLX Touch. JavaScript Mobile Framework for HTML5 Web Apps

[en línia] <http://www.dhtmlx.com/touch/>

[9] WebApp.Net. A light weight, powerful javascript framework

[en línia] <http://webapp-net.com/>

[10] Zepto.JS. The aerogel-weight jQuery-compatible JavaScript framework

[en línia] <http://zeptojs.com/>

[11] Wink Toolkit. A Mobile JavaScript framework to build great webapps

[en línia] <http://www.winktoolkit.org/>

[12] Phonegap. The only open source mobile framework that supports 7 platforms

[en línia] <http://www.phonegap.com/>

[13] Wikipedia EN

[en línia] <http://www.wikipedia.org>

[14] Quora

[en linia] <http://www.quora.com>

[15] Balsamiq Mockups

[en linia] <http://www.balsamiq.com/products/mockups>

[16] geeks.ms | Llegada de nuevos frameworks web móviles

[en linia] <http://geeks.ms/blogs/gperez/archive/2011/02/05/opini-243-n-llegada-de-nuevos-frameworks-web-m-243-viles.aspx>

[17] Genbeta:dev | Tamaño ideal de los elementos gráficos para crear aplicaciones móviles fáciles de usar

[en linia] <http://www.genbetadev.com/desarrollo-aplicaciones-moviles/tamano-ideal-de-los-elementos-graficos-para-crear-aplicaciones-moviles-faciles-de-usar>

[18] Operation Project | Comparing Mobile Web (HTML5) Frameworks

[en linia] <http://operationproject.com/2011/adventures-in-html5-part-one/#.T3VOYWE9Vtg>

[19] 51elliot | Phonegap and jQuery Mobile First Impressions

[en linia] <http://51elliot.blogspot.com.es/2011/02/phonegap-and-jquery-mobile-first.html>