

# Memòria PFC **nineando**

---

Programació d'aplicacions per a mòbils usant HTML5

**Íñigo Garcia Colino**

PFC - Enginyeria en Informàtica  
Consultor: Roman Roset Mayals  
Desembre | 2012

# Índex

---

<b>1</b>	<b>Definició del projecte</b>	5	<b>3.2</b>	<b>WAMP 5</b>	12
1.1	Descripció del PFC	5	3.3	Google Chrome	13
1.2	Objectius del projecte	6	3.4	eclipse SDK 4.2	13
1.2.1	Objectius generals	6	3.5	Android SDK Tools	13
1.2.2	Objectius específics	6	3.6	Android Development Tools (ADT)	13
1.3	Metodologia	7	3.7	TortoiseSVN	13
1.4	Calendari i terminis	8	<b>4</b>	<b>Tecnologies, frameworks i serveis web utilitzats</b>	14
1.5	Diagrama de Gantt	9	4.1	jQuery Mobile	15
<b>2</b>	<b>Estat de l'art</b>	10	4.2	Zend Framework	15
2.1	Guia de Fiestas Infantiles	10	4.3	MySQL	16
2.2	Cult App	11	4.4	Picasa	16
<b>3</b>	<b>Entorn de treball</b>	12	4.5	Disqus	17
3.1	Netbeans IDE 7.0	12	4.6	Google Maps	17
			4.7	PhoneGap	18

---

<b>5 Anàlisi i disseny</b> .....	19	<b>6 Arquitectura del projecte</b> .....	29
<b>5.1 Històries d'usuari</b> .....	19	<b>6.1 Arquitectura física</b> .....	29
<b>5.1.1 Cercar activitats per localització</b> .....	19	<b>6.2 Arquitectura lògica</b> .....	30
<b>5.1.2 Cercar activitats per províncies</b> .....	20	<b>7 Implementació i proves</b> .....	32
<b>5.1.3 Veure fitxa activitat</b> .....	21	<b>7.1 Aplicació jQuery Mobile</b> .....	32
<b>5.1.4 Veure / Afegir comentaris</b> .....	22	<b>7.1.1 Estructura pàgina HTML</b> .....	32
<b>5.1.5 Veure imatges</b> .....	23	<b>7.1.2 Pàgines dinàmiques</b> .....	34
<b>5.1.6 Afegir imatges</b> .....	24	<b>7.2 Aplicació Zend Framework</b> .....	35
<b>5.1.7 Com arribar</b> .....	25	<b>7.3 Implementació <i>Cercar activitats per localització</i></b> .....	37
<b>5.2 Flux de pantalles</b> .....	26	<b>7.4 Implementació <i>Cercar activitats per províncies</i></b> .....	39
<b>5.3 Disseny model de dades</b> .....	27	<b>7.5 Implementació <i>Veure fitxa activitat</i></b> .....	41
<b>5.3.1 Event</b> .....	27	<b>7.6 Implementació <i>Veure / Afegir comentaris</i></b> .....	42
<b>5.3.2 Categoria</b> .....	28	<b>7.7 Implementació <i>Veure imatges</i></b> .....	44
		<b>7.8 Implementació <i>Afegir imatges</i></b> .....	46

---

---

<b>7.9</b> Implementació <i>Com arribar</i> .....	47	<b>ANNEX I - Glosari</b> .....	62
<b>7.10</b> Proves .....	49	<b>ANNEX II - Fonts consultades</b> .....	64
<b>7.10.1</b> Cercar activitats per localització .....	49	<b>ANNEX III - Índex de figures</b> .....	65
<b>7.10.2</b> Cercar activitats per províncies .....	50		
<b>7.10.3</b> Veure fitxa activitat .....	52		
<b>7.10.4</b> Veure / Afegir comentaris .....	52		
<b>7.10.5</b> Veure imatges .....	53		
<b>7.10.6</b> Afegir imatges .....	54		
<b>7.10.7</b> Com arribar .....	55		
<b>7.11</b> Implementació Aplicació Android .....	56		
<b>8</b> Manual d'ús .....	59		
<b>9</b> Conclusions .....	60		
<b>10</b> Entregables PFC .....	61		

---

# 1 Definició del projecte

---

## 1.1 Descripció del PFC

El següent Projecte de Final de Carrera (des d'ara **PFC**), denominat "**nineando**", és un projecte dins de el àrea de les aplicacions mòbils de geolocalització.

Els serveis basats en localització busquen oferir un servei personalitzat als usuaris basant-se en la majoria de situacions en informació d'ubicació geogràfica d'aquests. En el PFC que s'exposa, s'utilitza tecnologies de sistemes d'informació geogràfica, per una banda es té el costat del client, que mitjançant HTML5 resoldrà el posicionament de l'usuari, i per l'altra banda es té el costat del servidor, que mitjançant el servei de **Google Maps**, servirà per transmetre informació cap l'aplicació que pugui processar.

Concretant una mica més l'abast, el PFC ha desenvolupar té com finalitat crear una aplicació per a mòbils que sigui capaç de localitzar activitats destinades als més petits, nens entre 0 i 12 anys.

El públic objectiu d'aquesta aplicació són dos grups. Per una banda tenim els pares amb nens que en un moment donat puguin fer servir el seu telèfon mòbil per cercar alguna activitat que estigui propera a la seva ubicació. Per l'altre, tenim a les diferents associacions, professionals i institucions que volen donar a conèixer les seves activitats. Així doncs, l'aplicació pretén ser un punt d'unió entre els generadors d'activitats infantils i els consumidors de les mateixes.

Procurant que la informació que es doni sigui el més útil possible, les activitats estaran dividides en categories, així mateix per cada activitat s'haurà de facilitar informació que vagi més enllà que la seva ubicació, com ara: horaris, edats recomanades, dates, preus....



Figura 1.1 Logotip Aplicació "nineando"

També, tenint en compte la tendència col·laborativa dels usuaris de les noves tecnologies d'avui en dia, s'ha pensat donar opció a aquests usuaris a que puguin afegir contingut en relació a les diferents activitats. Aquest contingut es basa en comentaris, útils per als pares que busquen una informació més subjectiva, i també en fotografies, de gran utilitat per als usuaris que busquen fer-se una idea més completa del que els pot oferir una activitat.

Es parteix de la base que la web col·laborativa fa enriquir més la informació que es pot adquirir des d'una aplicació, els diferents punts de vista ajuden sempre a fer-se una idea més completa del que pot representar, en aquest cas, una activitat infantil.

# 1 Definició del projecte

---

## 1.2 Objectius del projecte

### 1.2.1 Objectius generals

L'objectiu general del PFC és desenvolupar una aplicació per a dispositius mòbils la qual ha de facilitar informació d'interès per al seu públic objectiu: pares amb nens de 0 a 12 anys. Això és, que els usuaris pugin obtenir i accedir a informació relacionada amb activitats infantils: horaris, adreça, preu, com arribar... Per aconseguir això, l'aplicació ha de resoldre quina és la ubicació de l'usuari i ha de mostrar-li activitats properes a aquesta ubicació. També s'ha de tenir en compte que és possible que l'usuari vulgui buscar activitats al llarg de la geografia, per això l'aplicació ha de ser capaç de permetre la navegació lliure per mapes.

Un altre dels objectius generals és que l'aplicació no es quedi en una simple aplicació informativa, sinó que ha de permetre a l'usuari interactuar amb la mateixa en forma de col·laboració, s'està referint a la possibilitat que un usuari pugui transmetre les seves opinions sobre un acte concret així com que pugui accedir a les valoracions d'altres usuaris. En aquest sentit, i per enriquir més l'aplicació, també ha de permetre poder afegir imatges relacionades amb l'activitat així com accedir a les afegides per altres usuaris.

En conclusió, es podria resumir en un objectiu final: crear un punt de trobada o xarxa social de pares consumidors d'activitats infantils i de proveïdors d'aquestes activitats on cadascuna de les parts implicades puguin treure el màxim profit de l'eina que està a la seva disposició.

### 1.2.2 Objectius específics

Els objectius específics es centren en el disseny i desenvolupament de l'aplicació amb les tecnologies determinades en el projecte.

Segons l'arquitectura de l'aplicació, aquesta estarà composta per diferents capes: presentació, negoci i dades. Aquest fet vol dir que a banda de l'aplicació mòbil, es necessitaran una sèrie de serveis proporcionats per diferents servidors. En el cas de les activitats, serà necessari disposar d'una petita base de dades d'actes, ja que queda fora de l'abast del PFC la part administrativa que permeti incloure, actualitzar o eliminar activitats. El que sí es durà a terme és el model de la base de dades, les classes i mètodes necessaris per poder comunicar-se amb l'aplicació i retornar la informació demanada en els formats requerits.

Dintre dels serveis que ha d'oferir l'aplicació, el servei de localització l'haurà de facilitar l'API de Google Maps, el servei de comentaris, l'API de Disqus i, per finalitzar, el servei d'imatges serà proporcionat per l'API de Picasa.

En definitiva, els objectius específics, a banda de desenvolupar l'aplicació amb el framework jQuery Mobile (capa de presentació), es centren també en els serveis que han de proporcionar els diferents servidors. S'han de crear aquells que no existeixen i aprofitar al màxim aquells amb mètodes ja definits per les diferents APIs anomenades.

# 1 Definició del projecte

## 1.3 Metodologia

Donada la planificació de l'assignatura, la qual està definida per 4 entregues bàsiques, es pren la decisió d'utilitzar metodologies àgils en el desenvolupament del PFC, això són quatre unitats de temps on s'anirà desenvolupant el projecte en lapses curts.

El projecte desenvolupat en cada unitat de temps (entre tres i quatre setmanes) haurà d'incloure una planificació, anàlisi de requeriments, disseny, codificació, revisió i documentació. Al final de cada iteració s'ha d'aconseguir disposar de la part pactada sense errors.

Per gestionar de manera general com es van completant les diferents tasques que configuren el projecte, s'ha pensat en triar la metodologia **Kanban**. Aquesta metodologia ens permetrà visualitzar el treball i les fases del cicle de producció, determinar el límit de "treball en curs" i mesurar el temps a completar una tasca.

Bàsicament dividim el treball en parts, i cada tasca pot estar en 3 estats diferents (per fer, fent i fet), d'aquesta manera realitzarem un desenvolupament incremental. No farem servi pissarres plenes de post-it, utilitzarem el taulell Kanban online que ens proporciona [www.trello.com](http://www.trello.com).

S'ha de tenir en compte que hi ha varies tasques a realitzar en el projecte que en un entron real d'empresa segurament no s'inclourien en la planificació, però seran necessàries degut al desconeixement d'algunes tecnologies. En primer lloc es requereix un mínim aprenentatge teòric sobre conceptes de geolocalització i posteriorment, un aprenentatge pràctic de les diferents tecnologies necessàries HTML5, CSS3, Javascript, jQuery Mobile, API de Google Maps, API Picasa i API disqus.

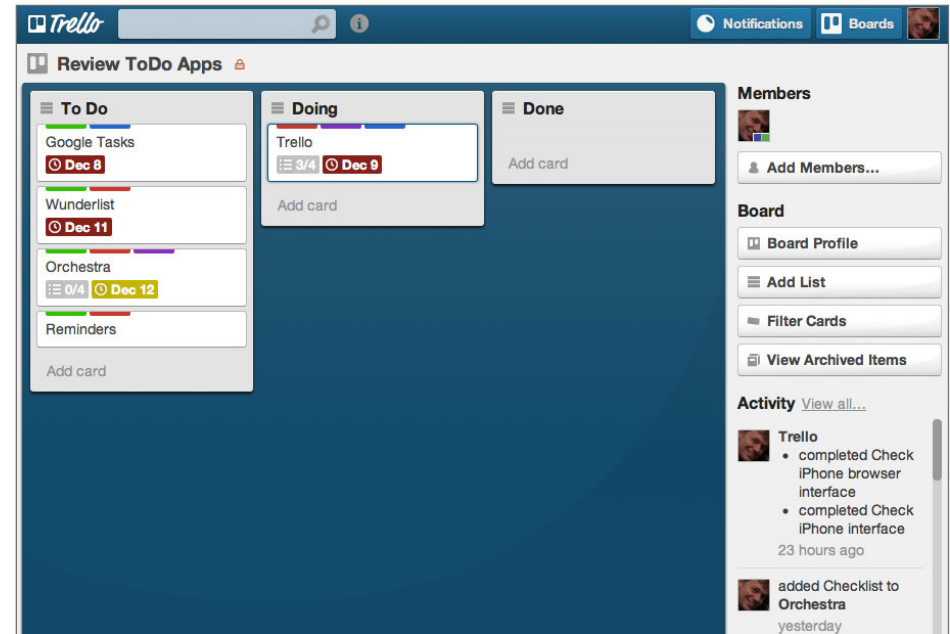


Figura 1.2 Panell Kanban facilitat per trello.com

En un principi pot semblar que es vulguin abastar moltes tecnologies, però es creu que és possible arribar als objectius marcats amb constància i una bona planificació.

## 1.4 Calendari i terminis

Per la realització del PFC es disposa d'un termini d'uns tres mesos i mig. Com s'ha definit anteriorment, la planificació d'entregues està definida per 4 entregues bàsiques o fases, i es pretén utilitzar metodologies àgils per els seu desenvolupament:

### Fase I: Preparació de l'entorn de treball i elaboració del pla del projecte

#### Del 19 de setembre al 3 d'octubre de 2012

- Preparació entorn treball: Triar el projecte a desenvolupar i descàrrega i instal·lació del programari necessari.
- Elaboració del pla del projecte: Determinar els objectius del projecte i planificació del treball a realitzar.

### Fase II: Anàlisi i disseny del projecte

#### Del 4 d'octubre al 31 d'octubre de 2012

- Estudi i aprenentatge de les tecnologies associades al projecte.
- Anàlisi: Detallar els requeriments funcionals/no funcionals de l'aplicació a desenvolupar.
- Disseny: Realització del disseny tècnic de l'aplicació necessari per dur a terme la implementació del projecte.
- Prototip: Creació del prototip de les aplicacions per validar les interfícies gràfiques.

### Fase III: Implementació

#### De l'1 de novembre al 28 de novembre de 2012

- Creació estructura fitxer XML de dades d'activitats.
- Crides als serveis web de les diferents APIs (GMaps i Disqus) per recuperar dades.
- Desenvolupament interfície gràfica aplicació: mapa, mapa cerca, fitxa activitat.
- Implementar les tecnologies necessàries per acomplir amb els objectius

### Fase IV: Proves, memòria i presentació

#### Del 29 de novembre al 4 de gener de 2013

- Testeig
- Correcció errades
- Creació del manual d'ús.
- Creació de la memòria.
- Elaboració de la presentació.



# 1 Definició del projecte

## 1.5 Diagrama de Gantt

A continuació es mostra el diagrama de Gantt de la planificació i tasques del projecte:

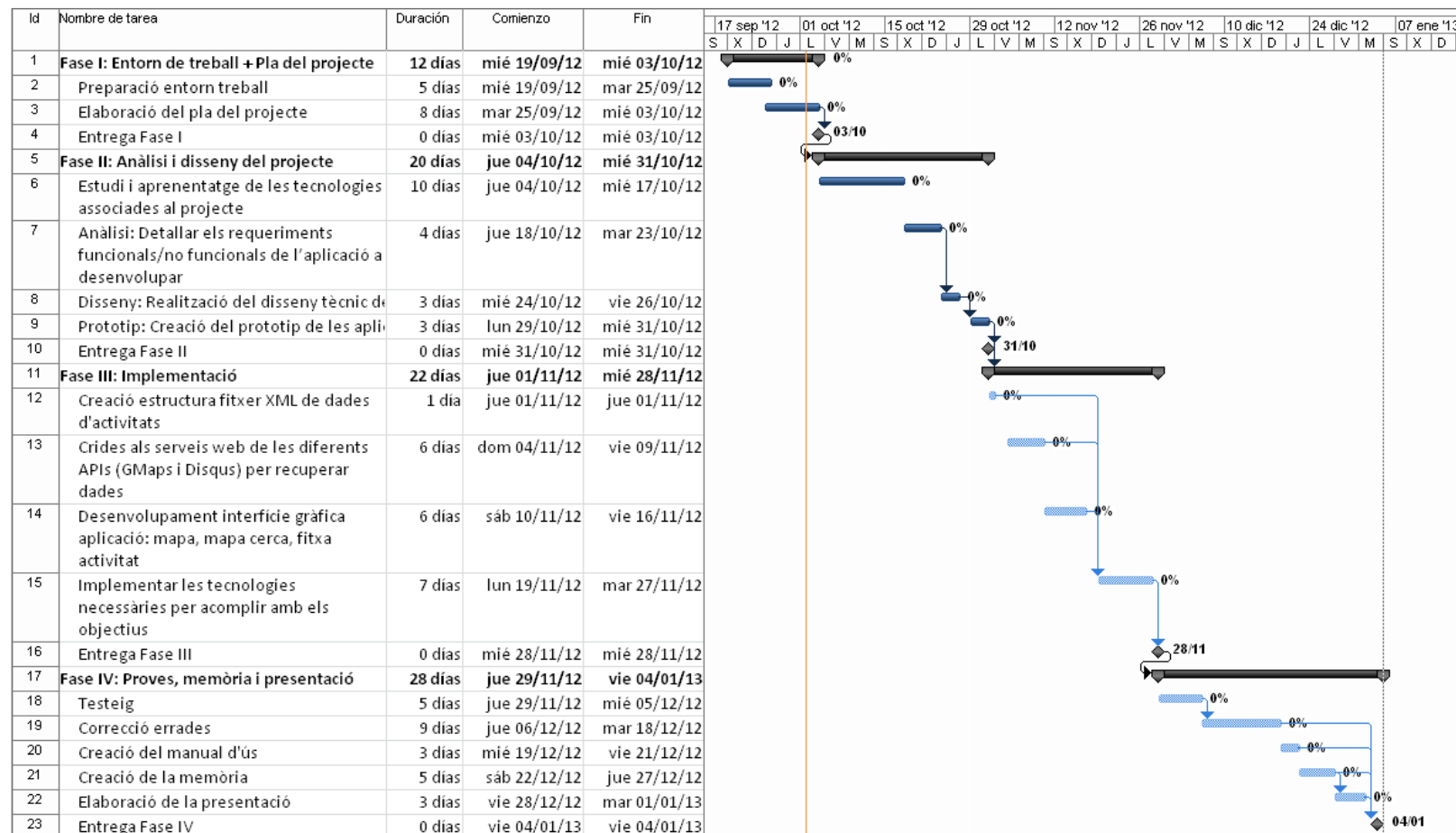


Figura 1.3 Diagrama de Gantt de la planificació i tasques del projecte

# 2 Estat de l'art

---

Entenem com a "estat de l'art", tots aquells desenvolupaments realitzats en tecnologies similars i enfocats al mateix públic que el projecte exposat. Les aplicacions trobades, en un principi han estat provades per la indústria i han estat acollides i acceptades amb més o menys èxit. Es tracta d'un estudi bàsic de mercat d'aplicacions similars a la presentada en aquest projecte, amb l'objectiu de valorar les semblances i diferències entre les diferents aplicacions.

## 2.1 Guia de Fiestas Infantiles

**Guia de Fiestas Infantiles**, és una aplicació que ofereix productes i serveis per a l'organització de festes i esdeveniments. Si bé es cert que hi ha algunes semblances, les diferències també són notables, en primer lloc el que s'ofereix no són activitats sinó una sèrie de serveis destinats al públic infantil, com poden ser empreses d'articles de festa, locals i fins i tot fotògrafs. En aquest sentit s'allunya bastant de l'objectiu de la nostra aplicació. Una altra diferència, més o menys important, és que està destinada al públic sud-americà.

Accessible des de Google Play, URL: [https://play.google.com/store/apps/details?id=org.guiafiestasinfantiles.gui&feature=search\\_result](https://play.google.com/store/apps/details?id=org.guiafiestasinfantiles.gui&feature=search_result)



Figura 2.1 Detall Aplicació "Guia de Fiestas Infantiles"



Figura 2.2 Detall Aplicació “Cult App”

### 2.2 Cult App

**Cult App**, és una aplicació gratuïta en la qual els usuaris poden consultar des del seu mòbil diferents activitats culturals, d'oci i benestar i esport que esdevenen en el seu entorn. L'aplicació proporciona informació sobre esdeveniments, concerts, teatres, cinemes, esport, activitats infantils...

Encara que tenen punts en comú amb la nostre aplicació, ja que s'ofereixen activitats infantils, el seu públic objectiu difereix bastant ja que les activitats infantils són un petit reclam de l'aplicació.

Una altre diferència seria l'enfoc dels seus clients o proveïdors d'esdeveniments, estan encarats a empreses privades que ofereixen entrades: concerts, teatres, cinema... en canvi el nostre projecte està enfocat a institucions públiques que ofereixen espectacles infantils: ajuntaments, consells comarcals, diputacions...

Accessible des de Google Play, URL: [https://play.google.com/store/apps/details?id=com.inventiplus.cultapp&feature=search\\_result](https://play.google.com/store/apps/details?id=com.inventiplus.cultapp&feature=search_result)

# 3 Entorn de treball

Després de revisar una sèrie de paquets de programari s'han seleccionat els següents ja que ofereixen tota la funcionalitat que es necessita per dur a terme el projecte. És assequible, fàcil d'usar, i s'integra amb el maquinari del qual es disposa.

## 3.1 Netbeans IDE 7.0

NetBeans és un entorn de desenvolupament integrat lliure. Existeix un nombre important de mòduls per estendre-ho. NetBeans és un producte lliure i gratuït sense restriccions d'ús que permet crear aplicacions Web amb PHP 5 amb suport per **Zend Framework**, així com aplicacions amb **HTML**, **CSS** i **JavaScript**.

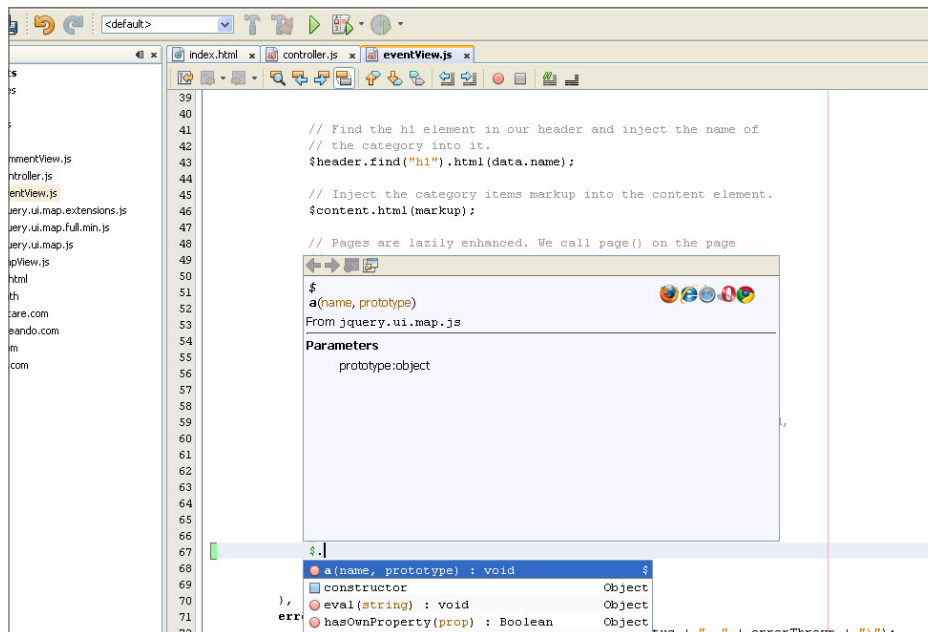


Figura 3.1 Detall Netbeans IDE 7.0



Figura 3.2 Detall WAMP 5

## 3.2 WAMP 5

WAMP és l'acrònim usat per descriure un sistema d'infraestructura d'internet que usa les següents eines:

- **Windows**, com a sistema operatiu
- **Apache**, com a servidor web
- **MySQL**, com a gestor de bases de dades
- **PHP** com a llenguatge de programació

L'ús de WAMP permet servir pàgines html a internet, a més de poder gestionar dades en elles. LAMP és el sistema anàleg que corre baix ambient Linux.

# 3 Entorn de treball

## 3.3 Google Chrome

Google Chrome és un navegador web desenvolupat per Google i compilat amb base en components de codi obert com el motor de renderització WebKit. També disposa d'eines per desenvolupadors que permeten depurar l'aplicació. Es sobre el navegador on es fara corre l'aplicació així com les proves necessaries.

## 3.4 eclipse SDK 4.2

Eclipse és un entorn de desenvolupament integrat de codi obert multiplataforma. El seu ús es necessari per construir l'aplicació nativa **Android** sobre l'aplicació web que s'ha creat al projecte.

## 3.5 Android SDK Tools

Android SDK Tools és com es coneix el conjunt d'eines de desenvolupament per programes per Android. Bàsicament s'està referint a **AVD Manager** i **SDK Manager**.

## 3.6 Android Development Tools (ADT)

ADT és el plugin requerit per l'IDE d'eclipse, està dissenyat per oferir un entorn potent, integrat amb l'objectiu de construir aplicacions Android.

## 3.7 TortoiseSVN

TortoiseSVN és un client **Subversion**, implementat com una extensió al shell de Windows. És programari lliure alliberat sota la llicència GNU GPL.

Subversion és un sistema de control de versions. És programari lliure i se li coneix també com **svn** per ser el nom de l'eina utilitzada en la línia de comandes.

El fet de poder accedir al repositori a través de xarxes, permet ser usat en diferents ubicacions. En el cas particular d'aquest projecte és una eina de gran utilitat ja que el projecte s'ha realitzat des de tres ubicacions / maquines diferents.

En aquest projecte es fa ús del servidor que facilita <http://www.sliksvn.com> amb 100 MB de capacitat.

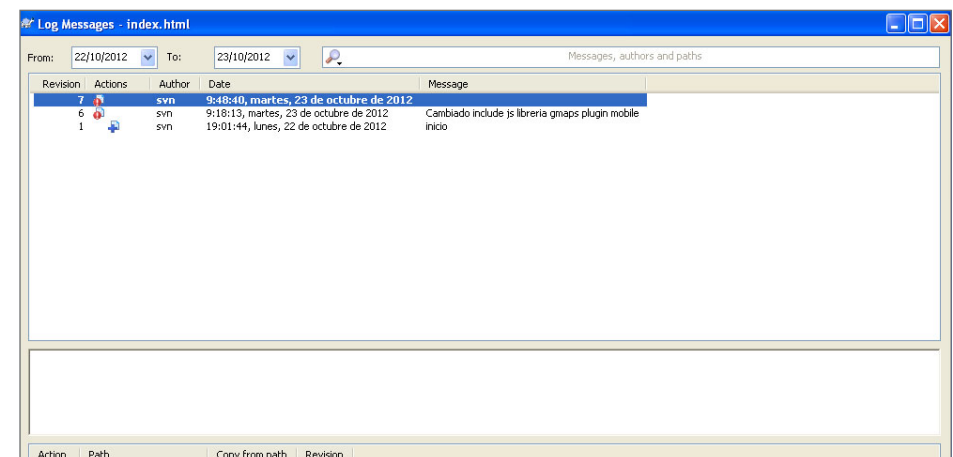


Figura 3.3 Detall TortoiseSVN

# 4 Tecnologies, frameworks i serveis web utilitzats

---

Davant el repte de crear una aplicació per a dispositius mòbils usant HTML5, el primer que es va fer va ser cercar informació per veure de quines opcions es disposava. Les opcions eren varies, però al final es va tenir que decidir entre dues: **SenchaTouch** i **jQueryMobile**. Aquestes dues opcions tenen en comú que es tracten de frameworks que es basen en la utilització de les tecnologies **HTML5**, **CSS3** i **JavaScript**. A continuació es defineixen les característiques que m'han fet decidir:

## jQueryMobile:

- Llibreria UI (interfície d'usuari) definida per manipular el DOM, Ajax i altres utilitats.
- Corva d'aprenentatge ràpida, més encara si es coneix jQuery, ja que es basa en les llibreries jQuery i jQuery UI.
- No és necessari un patró de programació, la qual cosa facilita una certa llibertat al programador i sobretot un agilitat.
- Possibilitat de fer servir un contenidor per permetre crear una aplicació gairebé nativa en sistemes Android, iOS...

## SenchaTouch:

- Es basa en la llibreria original ExtJS, solució més o menys robusta i de certs anys en el mercat.
- Es centra bàsicament en JavaScript, poc ús d'HTML5 i CSS3, més dificultat per la customització de l'aplicació.
- Corva d'aprenentatge lenta, patró molt definit en disseny i objectes.
- També disposa de la possibilitat de fer servir un contenidor per permetre crear una aplicació gairebé nativa en sistemes Android, iOS...

Un cop valorades les alternatives, el primer que es va fer va ser crear un projecte amb cada framework, aquest fet va ser definitiu per triar l'opció. El que es va trigar 5 dies per fer en SenchaTouch va ser l'equivalent d'1 dia per jQueryMobile. Tenint present l'objectiu del projecte i les funcionalitats que es volen dur a terme, es va triar l'opció que donava més rendiment: **jQueryMobile**.

# 4 Tecnologies, frameworks i serveis web utilitzats

## 4.1 jQuery Mobile

A partir de la base sòlida que proporciona jQuery i jQueryUI Foundation es va crear jQuery Mobile, un framework d'interfície unificada, amb l'objectiu de fer-lo compatible amb els navegadors de les diferents plataformes de dispositius mòbils.

La seva característica principal és que s'aconsegueixen uns resultats de gran nivell amb poc codi, i a més a més, ofereix una manera de customitzar les aplicacions de manera relativament senzilla. A banda de la navegació via Ajax, i transicions predefinides, també ofereix una sèrie de widgets (Interfície d'usuari) ja creats, com botons, barres d'eines, llistes, form elements...

Com s'especificava amb anterioritat, l'objectiu del framework jQuery Mobile és que sigui compatible amb les diferents plataformes de dispositius mòbils, com per exemple iOS, Android, RIM... sempre tenint en compte que s'està parlant d'una tecnologia web. Aquest fet determina que la sintaxis del codi HTML segueixi els estàndards necessaris per assegurar que aquesta compatibilitat sigui el més extensa possible.

Es pot accedir a descarregar el codi font a: <http://jquerymobile.com/>

## 4.2 Zend Framework

Zend Framework (ZF) és un framework de codi obert per desenvolupar aplicacions web i serveis web amb PHP 5. ZF és una implementació que usa codi 100% orientat a objectes. Cada component de ZF està construït amb una baixa dependència d'altres components. Aquesta arquitectura feblement acoblada permet als desenvolupadors utilitzar els components per separat.

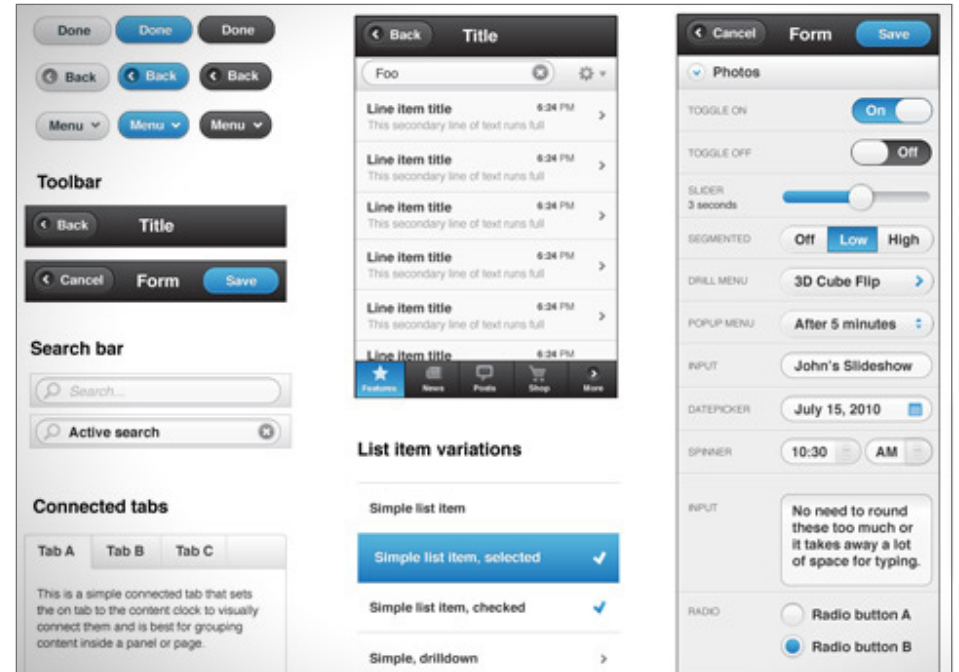


Figura 4.1 Detall Interfície d'usuari jQuery Mobile

ZF ofereix un gran rendiment i una robusta implementació MVC, una abstracció de base de dades fàcil d'usar, i un component de formularis que implementa la prestació de formularis HTML, validació i filtrat perquè els desenvolupadors puguin consolidar totes les operacions usant d'una manera senzilla la interfície orientada a objectes.

El principal patrocinador del projecte Zend Framework és Zend Technologies, però moltes empreses han contribuït amb components o característiques

# 4 Tecnologies, frameworks i serveis web utilitzats

importants per al marc. Empreses com Google i Microsoft s'han associat amb Zend per proporcionar interfícies de serveis web i altres tecnologies que desitgen posar a la disposició dels desenvolupadors de Zend Framework.

Es pot accedir a descàrregar el codi font a: <http://framework.zend.com/>

## 4.3 MySQL

MySQL és un sistema de gestió de bases de dades relacional amb més de sis milions d'instal·lacions. Des d'abril de 2009, Oracle desenvolupa MySQL com a programari lliure.

D'una banda s'ofereix sota la GNU GPL per a qualsevol ús compatible amb aquesta llicència, però per a aquelles empreses que vulguin incorporar-ho en productes privatis han de comprar a l'empresa una llicència específica que els permeti aquest ús. Està desenvolupat en la seva major part en ANSI C.

Es pot accedir a descàrregar el codi font a: <http://www.mysql.com/>

## 4.4 Picasa

Picasa és un organitzador i visor d'imatges, i una eina per editar fotografies digitals. Addicionalment, Picasa posseeix un lloc web integrat per compartir fotos.

L'aplicació va ser creada originalment per una companyia anomenada Lifescape en 2002, al juliol de 2004, Google va adquirir Picasa i va començar a oferir-la gratuïtament.

En l'actualitat s'ofereix una capacitat d'1 GB per emmagatzemar imatges en la seva versió gratuïta, si es requereix de més capacitat hi ha la possibilitat d'augmentar-la amb un servei de pagament.

Picasa disposa d'una API per PHP (tecnologia usada en el projecte per accedir al servei). Es pot consultar a: <https://developers.google.com/picasa-web/>

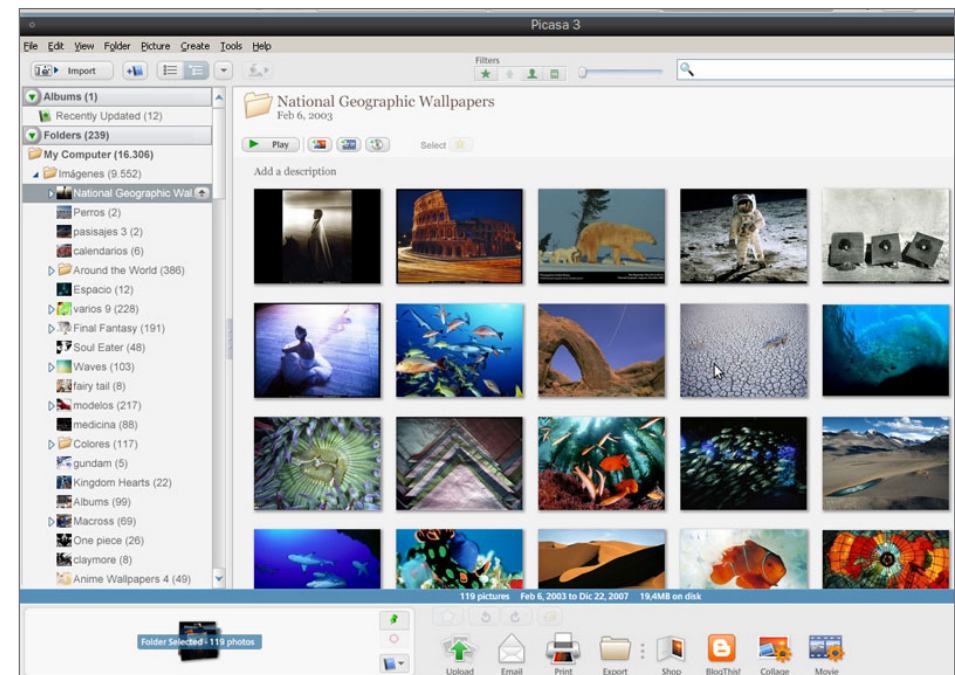


Figura 4.2 Detall Picasa



# 4 Tecnologies, frameworks i serveis web utilitzats

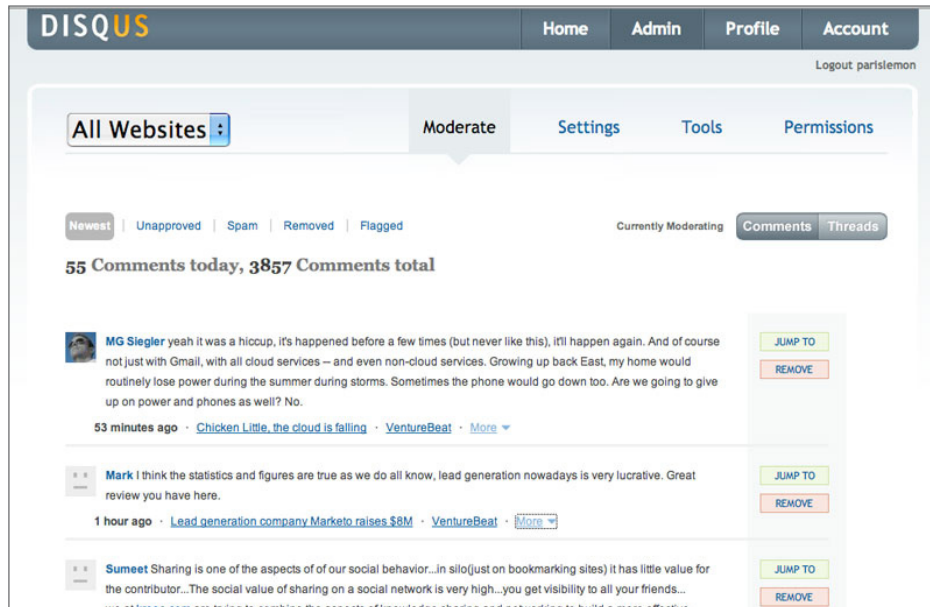


Figura 4.3 Detall panell administració Disqus

## 4.5 Disqus

Disqus és un servei de discussió en línia i de servei de comentaris per llocs web i comunitats online que utilitza una plataforma en xarxa. La plataforma inclou diverses característiques, tals com la integració en xarxes socials, perfils d'usuari, eines de moderació, anàlisi, notifikacions per correu electrònic...

Encara que hi ha serveis similars, cal destacar que és una de les plataformes amb més èxit. Està quantificat que en els EUA tenen 144 milions de visitants

únics mensuals. Entre les companyies que utilitzen el servei es pot destacar Engadget, CNN i The Daily Telegraph, a banda d'uns 750.000 blogs i milers de llocs web internacionals.

Es pot accedir al servei així com a la documentació a: <http://disqus.com/>

## 4.6 Google Maps

Google Maps és el nom d'un servei de Google. És un servidor d'aplicacions de mapes en la Web. Ofereix imatges de mapes desplaçables, així com fotos

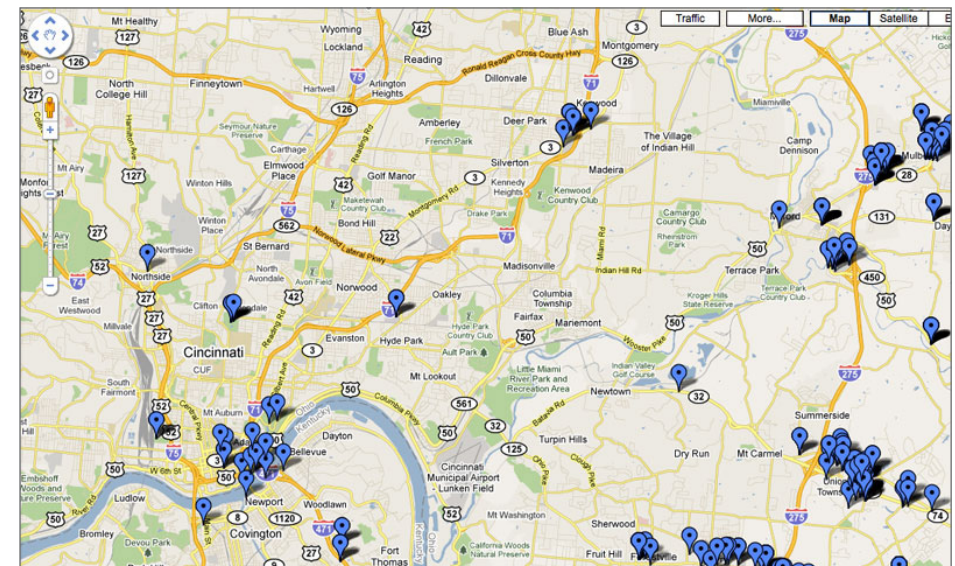


Figura 4.4 Detall Google Maps

## 4 Tecnologies, frameworks i serveis web utilitzats

---

satèl·lit del món i fins i tot la ruta entre diferents ubicacions o imatges a peu de carrer amb el seu servei Google Street View.

És idèntic a Google Earth, una aplicació d'escriptori per diferents plataformes, que ofereix vistes del globus terraquí, sigui de dia o de nit, però que no és fàcil d'integrar a pàgines Web.

Ofereix, així mateix, la possibilitat que qualsevol propietari d'una pàgina Web integri moltes de les seves característiques al seu lloc. Per gaudir d'aquestes característiques es pot accedir a la seva API: <https://developers.google.com/maps/documentation/javascript/?hl=es>

### 4.7 PhoneGap

PhoneGap (anteriorment anomenat Apache Callback, però ara Apache Cordova) és un marc de codi obert de desenvolupament d'aplicacions mòbils que permet als programadors de programari crear aplicacions per a dispositius mòbils utilitzant Javascript, HTML5 i CSS3, en lloc de llenguatges específics per dispositius, com ara **Objective-C** o **Android**.

Les aplicacions resultants són híbrids, és a dir, que no són ni veritablement natiu (perquè tota representació de disseny es duu a terme a través de visites web en lloc de marc natiu de la plataforma d'interfície d'usuari) ni purament basat en la web (ja que no són només aplicacions web, però s'empaqueten com a aplicacions per a la distribució i l'accés a les API del dispositius natius.

Es pot descarregar les llibreries necessàries i accedir a la documentació a: <http://phonegap.com/>

# 5 Anàlisi i disseny

## 5.1 Històries d'usuari

Una història d'usuari és una representació d'un requisit de programari escrit en una o dues frases utilitzant el llenguatge comú de l'usuari. Les històries d'usuari són utilitzades en les metodologies de desenvolupament àgils per a l'especificació de requisits. En aquest projecte s'utilitzarà les històries d'usuari ja que són una forma ràpida d'administrar els requisits dels usuaris sense haver d'elaborar gran quantitat de documents formals i sense requerir de molt temps per administrar-los.

### 5.1.1 Cercar activitats per localització

<b>id:</b> 1	<b>Nom:</b> Cercar activitats per localització
<b>Descripció:</b> Com a usuari vull cercar activitats que estiguin a prop de la meua localització. És necessari que m'ubiqui a mi i que em mostri en un mapa quines activitats tinc al meu voltant.	
<b>Estimació:</b> 5 dies	<b>Prioritat:</b> 8
<b>Proves d'acceptació:</b> <ul style="list-style-type: none"><li>• Comprovar que la nostra ubicació és correcte</li><li>• Comprovar que les activitats que em mostra són correctes</li></ul>	

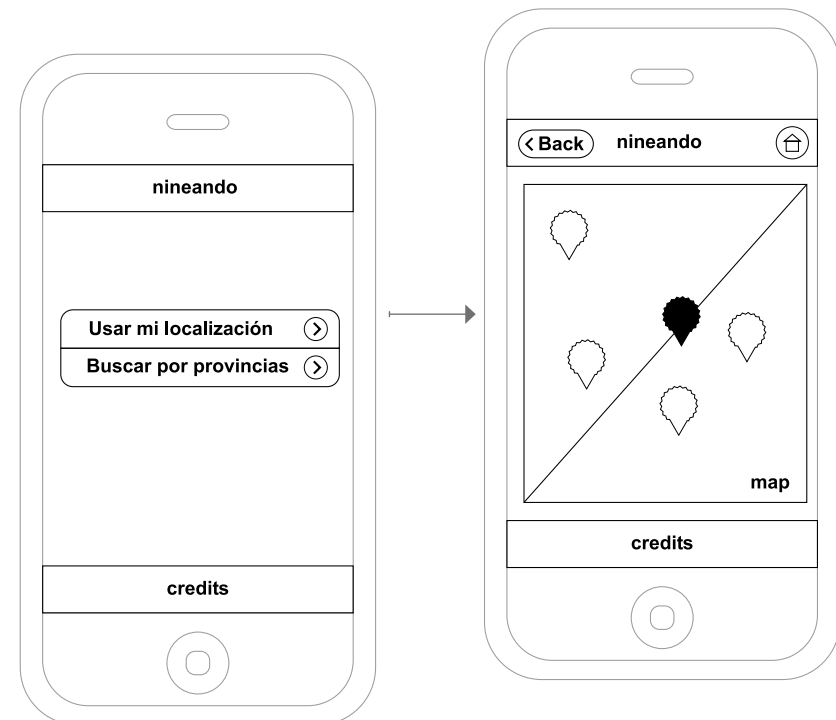


Figura 5.1 Pantalles “Cercar activitats per localització”

## 5.1.2 Cercar activitats per províncies

<b>id:</b> 2	<b>Nom:</b> Cercar activitats per províncies
<b>Descripció:</b> Com a usuari vull cercar activitats per províncies i no dependre de la meua localització. És necessari un llistat i que al fer click damunt una província em localitzi en un mapa quines activitats tinc disponibles.	
<b>Estimació:</b> 5 dies	<b>Prioritat:</b> 8
<b>Proves d'acceptació:</b> <ul style="list-style-type: none"><li>• Comprovar que el llistat de províncies és correcte</li><li>• Comprovar que es despleguen al mapa de la província les activitats correctes</li></ul>	

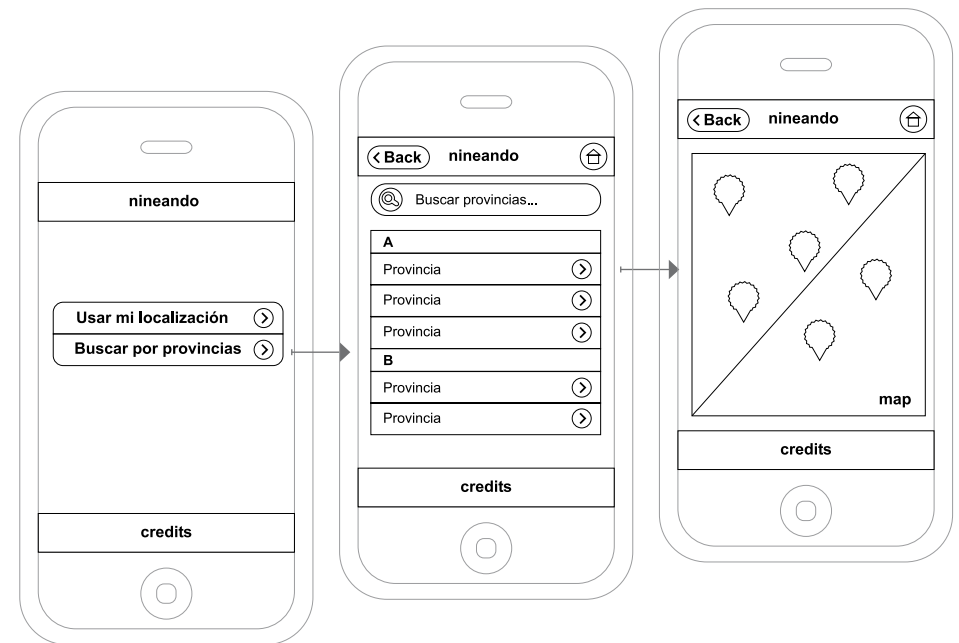


Figura 5.2 Pantalles "Cercar activitats per províncies"

## 5.1.3 Veure fitxa activitat

<b>id:</b> 3	<b>Nom:</b> Veure fitxa activitat
<b>Descripció:</b> Com a usuari, un cop tinc situat en un mapa les diferents activitats, vull accedir a la fitxa de l'activitat, i que aquesta em faciliti informació d'interès.	
<b>Estimació:</b> 4 dies	<b>Prioritat:</b> 7
<b>Proves d'acceptació:</b> <ul style="list-style-type: none"><li>• Comprovar que des d'el mapa es pot accedir a les diferents fitxes de cada activitat.</li><li>• Comprovar que les dades facilitades són correctes i en el format esperat.</li></ul>	

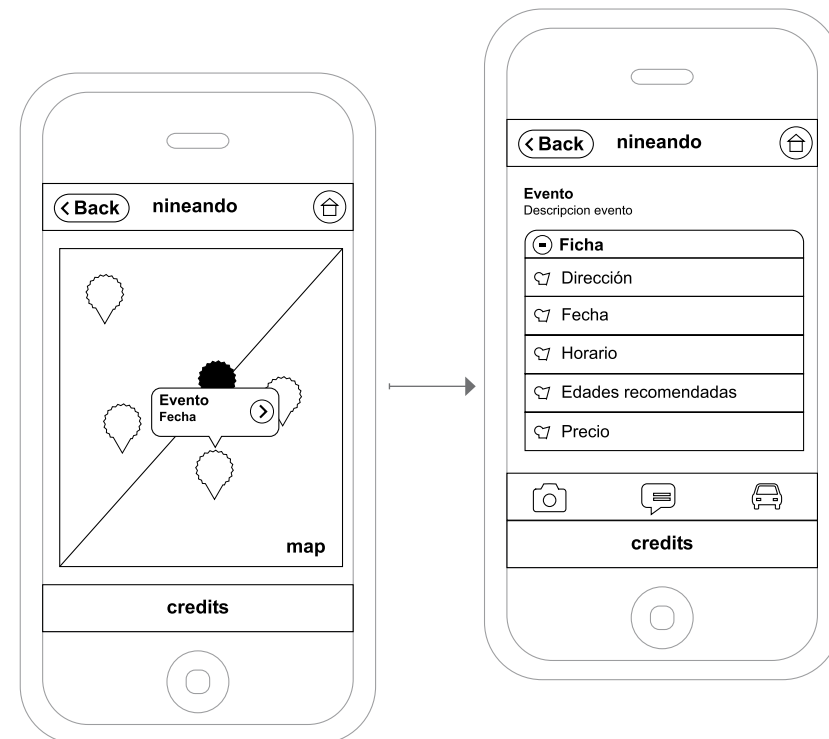


Figura 5.3 Pantalles “Veure fitxa activitat”

## 5.1.4 Veure / Afegir comentaris

<b>id:</b> 4	<b>Nom:</b> Veure / Afegir comentaris
<b>Descripció:</b> Com a usuari, vull disposar de l'opció de poder comentar el que m'ha semblat una activitat en concret, així com poder veure les opinions d'altres usuaris.	
<b>Estimació:</b> 4 dies	<b>Prioritat:</b> 3
<b>Proves d'acceptació:</b> <ul style="list-style-type: none"><li>• Comprovar que el servei web de disqus s'executa correctament.</li><li>• Comprovar que hi ha persistència de les dades (comentaris).</li></ul>	

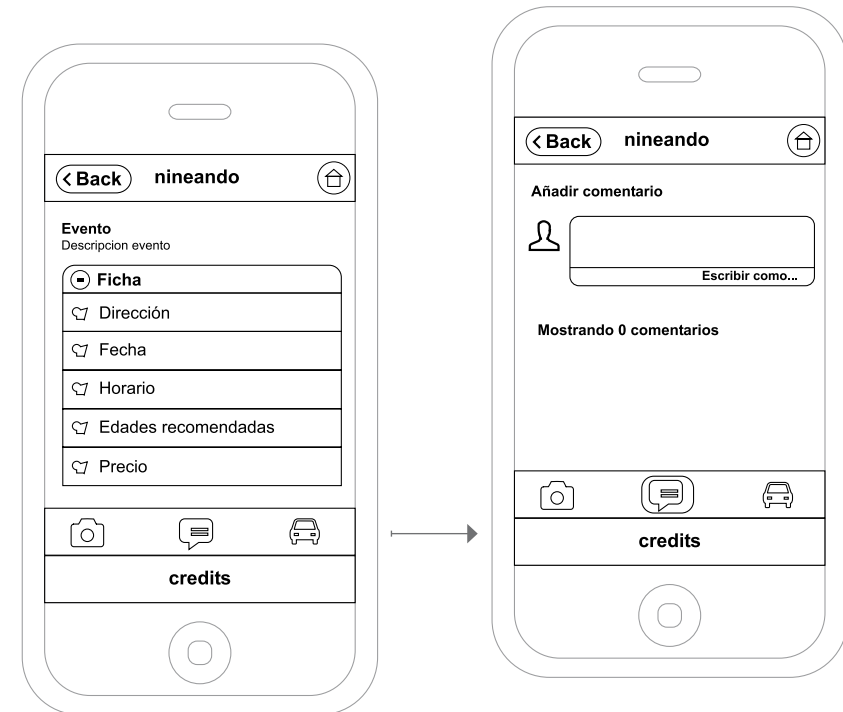


Figura 5.4 Pantalles “Veure / Afegir comentaris”

## 5.1.5 Veure imatges

<b>id:</b> 5	<b>Nom:</b> Veure imatges
<b>Descripció:</b> Com a usuari, vull accedir a les imatges que pugui tenir una activitat.	
<b>Estimació:</b> 4 dies	<b>Prioritat:</b> 3
<b>Proves d'acceptació:</b> <ul style="list-style-type: none"><li>• Comprovar que el servei web de picasa s'executa correctament.</li><li>• Comprovar que el llistat i els pop up de les imatges són correctes.</li></ul>	

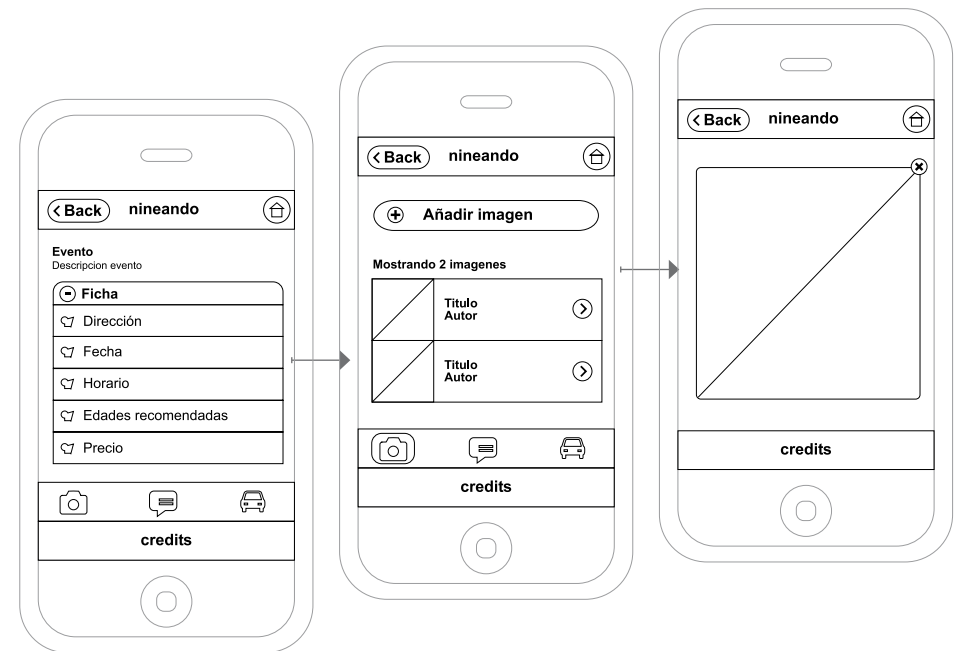


Figura 5.5 Pantalles "Veure imatges"

## 5.1.6 Afegir imatges

<b>id:</b> 6	<b>Nom:</b> Afegir imatges
<b>Descripció:</b> Com a usuari, vull tenir l'opció d'afegir imatges per les activitats que cregui convenient.	
<b>Estimació:</b> 4 dies	<b>Prioritat:</b> 3
<b>Proves d'acceptació:</b> <ul style="list-style-type: none"><li>• Comprovar que el servei web de picasa s'executa correctament.</li><li>• Comprovar que hi ha persistència de les dades (imatges).</li><li>• Comprovar que es possible carregar imatges del dispositiu i pujar-les al servidor.</li></ul>	

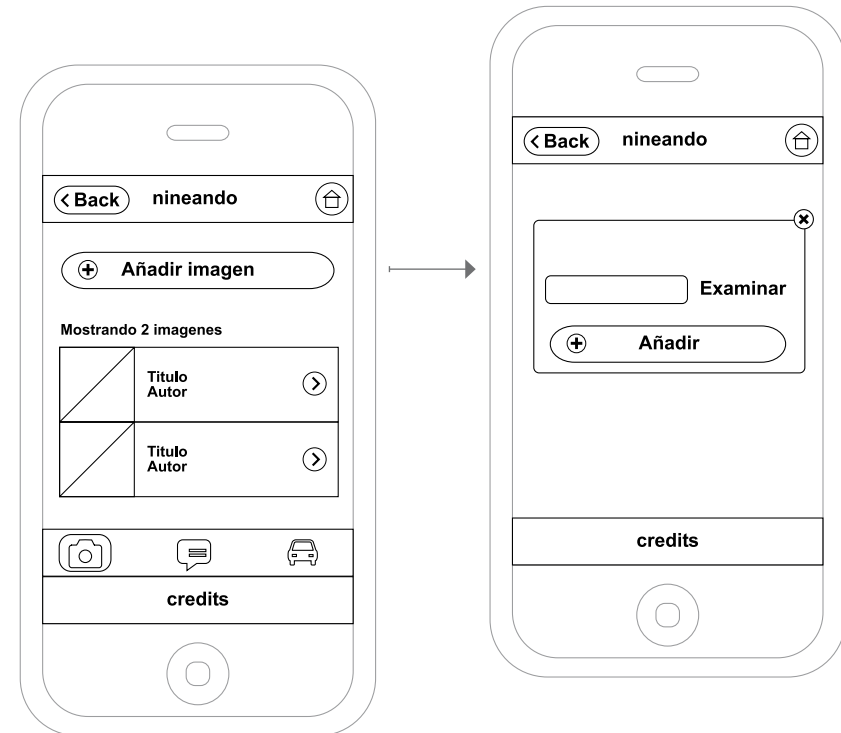


Figura 5.6 Pantalles "Afegir imatges"



## 5.1.7 Com arribar

<b>id:</b> 7	<b>Nom:</b> Com arribar
<b>Descripció:</b> Com a usuari, donada la meva localització, vull tenir l'opció de disposar d'una ruta per arribar a una activitat en concret.	
<b>Estimació:</b> 4 dies	<b>Prioritat:</b> 3
<b>Proves d'acceptació:</b> <ul style="list-style-type: none"><li>• Comprovar que el servei web de google s'executa correctament.</li><li>• Comprovar, dintre del que sigui possible, que les instruccions que es donen són correctes.</li></ul>	

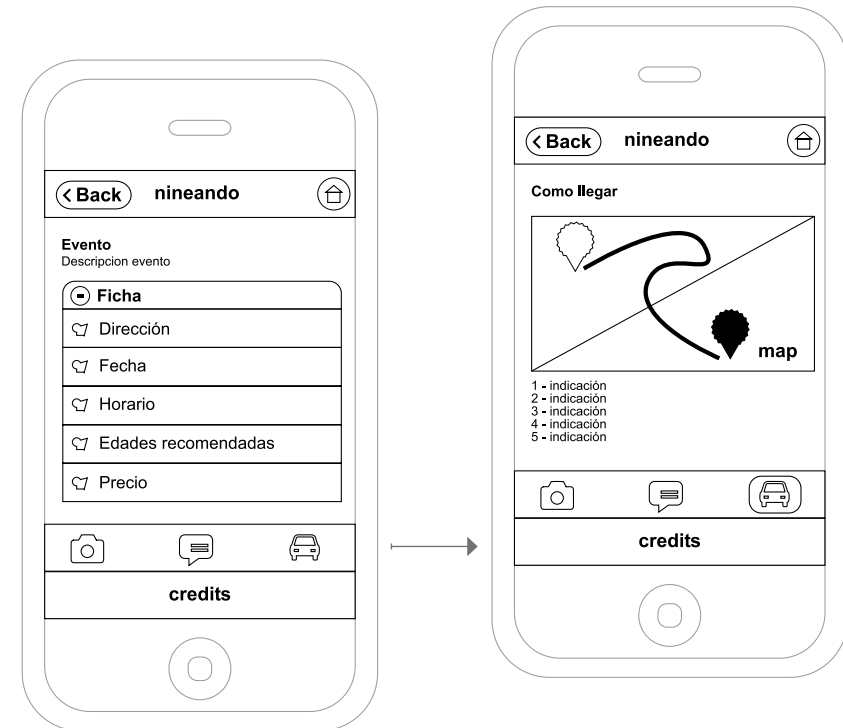


Figura 5.7 Pantalles “Com arribar”

## 5.2 Flux de pantalles

L'aplicació segueix el següent flux de pantalles. Bàsicament l'usuari té l'opció de cercar activitats per la seva localització o mitjançant un llistat de províncies. Un cop ha triat el mapa a mostrar, si té resultats pot veure la fitxa. Un cop a la fitxa té opció de veure les indicacions de com arribar, de veure les imatges i, si vol, pujar-ne una, o bé accedir als comentaris, en aquest cas, també té l'opció d'afegir-ne un.

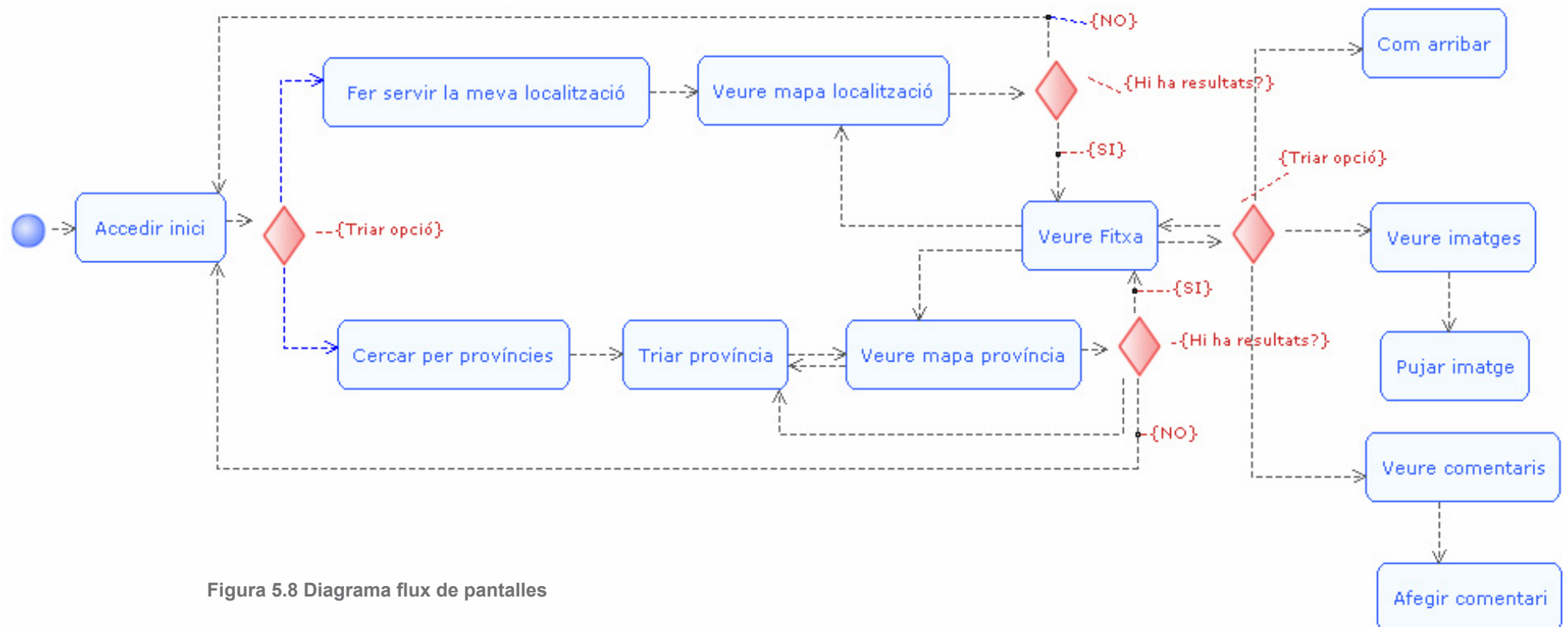


Figura 5.8 Diagrama flux de pantalles

## 5.3 Disseny model de dades

En aquest apartat es fa referència al model de dades creat que donen persistència a les activitats infantils. Queda fora de l'abast d'aquest disseny el model que fa referència a les imatges així com als comentaris.

El model és molt bàsic, però suficient per dur a terme el projecte. Hi ha 2 taules, per una banda tenim la taula **events** i per l'altre la taula **categorias**.

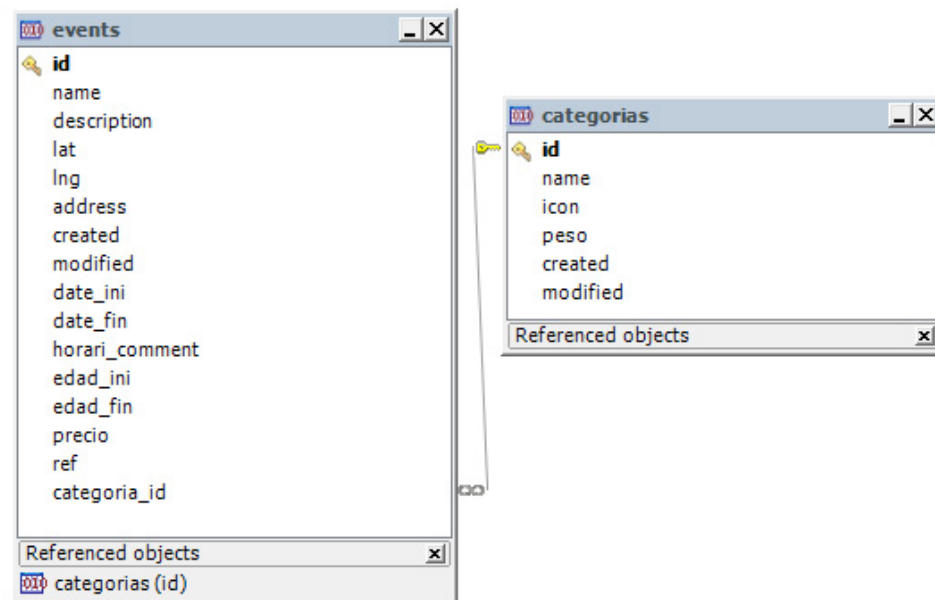


Figura 5.9 Diagrama model de dades

### 5.3.1 Events

Com a event entenem una activitat infantil. Per cada event tenim els següents atributs:

- **id**: int(10). Identificador únic de l'activitat infantil.
- **name**: varchar(255). Nom de l'activitat infantil.
- **description**: text. Descripció de l'activitat infantil.
- **lat**: float(10,6). Coordenada latitud de l'activitat infantil.
- **lng**: float(10,6). Coordenada longitud de l'activitat infantil.
- **address**: varchar(255). Adreça de l'activitat infantil.
- **created**: timestamp. Data de creació del registre de l'activitat infantil.
- **modified**: timestamp. Data d'actualització del registre de l'activitat infantil.
- **date\_ini**: timestamp. Data d'inici de l'activitat infantil.
- **date\_fin**: timestamp. Data de fi de l'activitat infantil.
- **horari\_comment**: text. Horari de l'activitat infantil.
- **edad\_ini**: int(2). Edat recomenada d'inici de l'activitat infantil.
- **edad\_fin**: int(2). Edat recomenada de fi de l'activitat infantil.
- **precio**: varchar(255). Preu de l'activitat infantil.
- **ref**: varchar(255). Referència de l'activitat infantil (pàgina web).
- **categoria\_id**: int(10). Identificador de la categoria de l'activitat infantil.

## 5.3.2 Categorias

Com a categoria entenem justament això, una categorització d'una activitat infantil. Per cada categoria tenim els següents atributs:

- **id:** int(10). Identificador únic de la categoria.
- **name:** varchar(255). Nom de la categoria.
- **icon:** varchar(255). Imatge identificativa de la categoria.
- **peso:** int(10). Pes identificatiu de l'ordre en llistats.
- **created:** timestamp. Data de creació del registre de la categoria.
- **modified:** timestamp. Data d'actualització del registre de la categoria.

Les categories contemplades són:

 Teatre	 Tallers	 Cinema	 Esport	 Colònies	 Màgia
 Música	 Fires	 Contes	 Parcs	 Pallassos	

Figura 5.10 Icones identificatives de les categories

# 6 Arquitectura del projecte

## 6.1 Arquitectura física

Per la realització del projecte, s'ha volgut fer ús de diferents serveis que es proporcionen a Internet de manera gratuïta i que tenen l'avantatge d'oferir una gran robustesa i disponibilitat degut a les àmplies infraestructures que disposen cada una de les empreses propietàries dels serveis.

Tenint en compte que hi havia una part que no es podia cobrir amb aquests serveis, que és la part que dóna singularitat al projecte (el recull d'activitats infantils), s'ha pres la decisió d'implementar a banda els serveis de la part de servidor en una infraestructura pròpia.

A continuació es mostra els diferents components de l'arquitectura física del projecte:

- **Dispositiu mòbil:** Dispositiu en el qual s'executa l'aplicació creada en HTML5, JavaScript i CSS3 mitjançant el framework jQuery Mobile.
- **Servidor propi:** Servidor propi en el qual s'han instal·lat les tecnologies necessàries per poder-se comunicar i resoldre les peticions de l'aplicació mòbil. Proporciona una part de les dades que consumeix l'aplicació, que es troben emmagatzemades en una base de dades MySQL. També està hostatjada l'aplicació PHP que es comunica amb l'aplicació mòbil i que fa la funció de proxy amb el servidor de Picasa.
- **Picasa:** Encarregat de l'emmagatzemament de les imatges pujades per l'usuari, el redimensionament d'aquestes en diverses mides i la persistència de dades com l'autor o el títol de les imatges.
- **Disqus:** Encarregat de proporcionar els serveis que fan referència a la publicació de comentaris sobre les diferents activitats i a la persistència d'aquests.

- **Google Maps:** Proporciona tots els serveis vinculats a la localització: localització de l'usuari i de les diferents activitats, creació d'itineraris, resoldre adreces en coordenades...

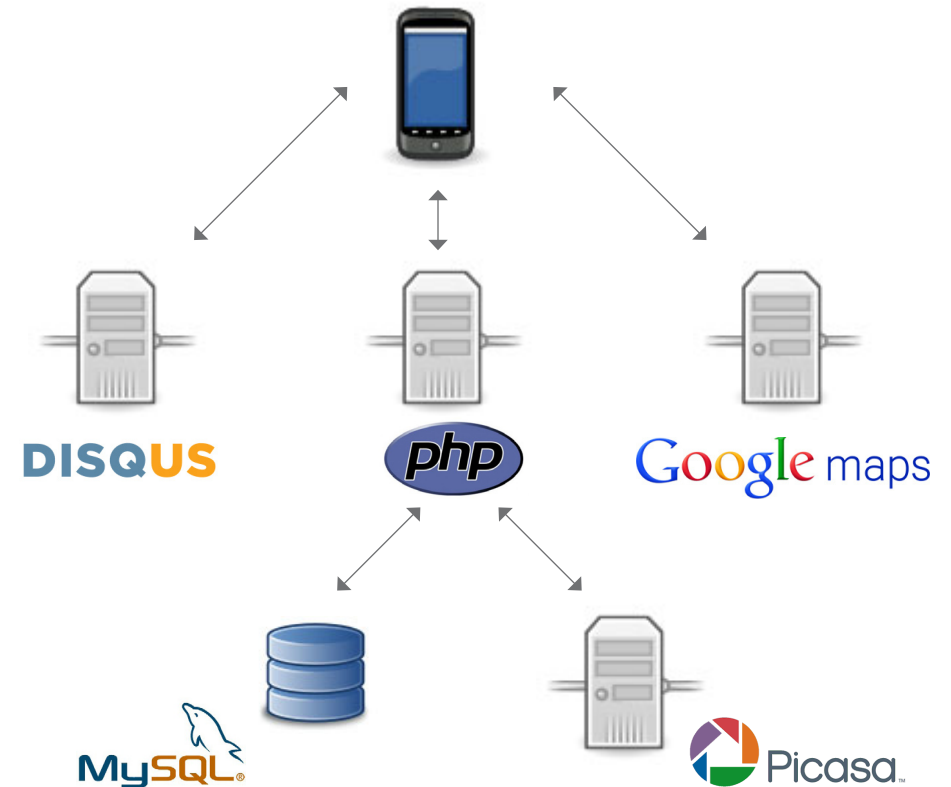


Figura 6.1 Esquema arquitectura física

# 6 Arquitectura del proyecto

## 6.2 Arquitectura lògica

El projecte exposat està estructurat en capes, això és que estan diferenciades les diferents lògiques que componen l'aplicació. Aquestes capes són les següents:

- **Capa de presentació:** Aquesta capa és la part de l'aplicació que s'executa directament en el dispositiu, és a dir, la part amb la que interactua l'usuari. Està composta per les vistes o pàgines que es visualitzen, rep les peticions de l'usuari i pinta la resposta d'aquestes peticions. És la part creada en HTML5, JavaScript i CSS3 mitjançant el framework jQuery Mobile, i serà l'encarregada de comunicar-se amb la capa de negoci.
- **Capa de negoci:** La capa de negoci és l'encarregada de processar les peticions de la capa de presentació, és a dir, aquesta capa rep uns paràmetres i el nom d'una acció a realitzar. Els diferents processos que s'hagin de realitzar, així com el resultat de l'acció demanda amb els paràmetres entrats, serà responsabilitat d'aquesta capa. Un cop ha finalitzat el seu procés, retorna el resultat a la capa de presentació, i aquesta mostra les dades. Un altre aspecte a considerar, és la necessitat de consultar una base de dades davant una petició realitzada, en aquest cas, la capa de negoci serà l'encarregada de comunicar-se amb la capa de dades per obtenir els registres sol·licitats, i amb aquests, realitzar el procés que convingui.
- **Capa de dades:** Tal com diu el seu nom, és la capa encarregada d'accedir a les dades sol·licitades per la capa de negoci i retornar-li aquestes. En el projecte exposat, les dades resideixen en una base de dades MySQL, en aquest cas, la capa de dades s'encarrega d'atacar directament la base de dades executant les querys que produeixen els resultats esperats. També s'hauria de considerar capa de dades als diferents serveis web utilitzats: Picasa, Google Maps i Disqus, ja que aquests emmagatzemen les dades

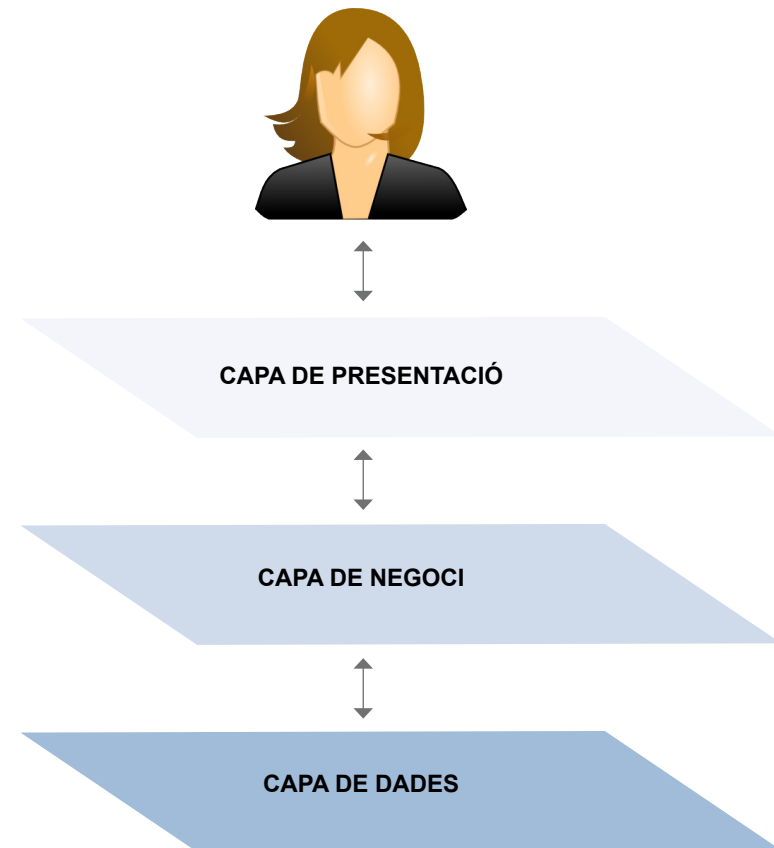


Figura 6.2 Esquema arquitectura lògica

# 6 Arquitectura del proyecto

que la capa de negoci anirà a buscar quan rebí una petició de la capa de presentació. Un exemple és el servei web de Picasa, la capa de negoci acaba fent una petició a Picasa d'unes dades, les rep, les processa i li envia processades a la capa de presentació.

Un exemple pràctic de la comunicació entre capes és el següent, per la història d'usuari **Veure fitxa activitat**, l'usuari fa click en uns dels resultats mostrats per accedir a la fitxa (Capa de presentació), la capa de presentació realitza una crida a la capa de negoci d'aquest estil:

```
http://domini/events/get-json-by-id/id/21
```

La capa de negoci rep la crida amb els següents paràmetres, una acció a realitzar (**get-json-by-id**) i uns paràmetres (**id** i **21**), processa les dades i acaba decidint realitzar una crida a la capa de dades:

```
//recuperamos param via GET
$id = $this->getParam('id');
$aEvent = $event->getById($id);
```

La capa de dades rep la petició de la capa de negoci, es comunica amb la base de dades mitjançant una query:

```
SELECT `e`.*, `ca`.`name` AS `catname`, `ca`.`icon` AS `caticon`
FROM `events` AS `e` INNER JOIN `categorias` AS `ca` ON ca.id
= e.categoria_id WHERE (e.id = '21') LIMIT 1
```

i quan obté les dades, li envia a la capa de negoci. Aquesta capa processa les dades rebudes i li envia a la capa de presentació per a que aquesta, en última instància, acabi pintant les dades en el dispositiu en forma de fitxa, la fitxa a la que accedeix finalment i de forma transparent l'usuari.

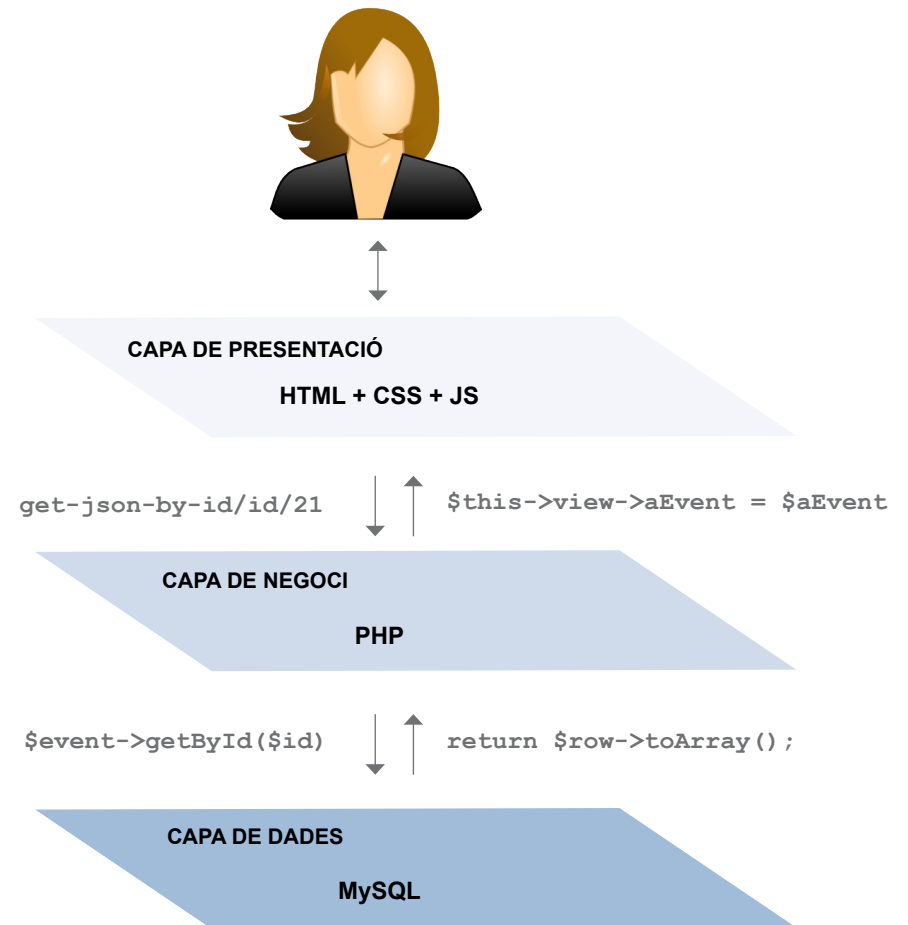


Figura 6.3 Esquema arquitectura lògica. Exemple

# 7 Implementació i proves

---

## 7.1 Aplicació jQuery Mobile

Com ja s'ha especificat, el projecte de l'aplicació mòbil es desenvolupa mitjançant el framework jQuery Mobile, que es basa en les tecnologies HTML5, CSS3 i JavaScript. S'ha triat la versió 1.2, que és la darrera versió estable a data de la realització del projecte.

### 7.1.1 Estructura pàgina HTML

Així doncs, el primer pas és crear l'estructura del fitxer HTML amb els tags necessaris, així com incloure les llibreries i fulles d'estil necessàries per fer ús del framework:

#### Declaració doctype HTML5:

```
<!DOCTYPE html>
<html>
```

#### Llibreries framework jQuery Mobile:

```
<script src="http://code.jquery.com/jquery-1.8.2.min.js"></script>
<script src="http://code.jquery.com/mobile/1.2.0/jquery.mobile-1.2.0.min.js"></script>
```

#### Fulles d'estil CSS3:

```
<link rel="stylesheet" href="http://code.jquery.com/mobile/1.2.0/jquery.mobile-1.2.0.min.css"/>
```

Una altre tag recomanable a inserir en el <head> de la pàgina és la etiqueta meta "viewport", que amb els atributs corresponents permet que l'aplicació

s'adapti a l'amplada de cada dispositiu:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Dintre de les opcions d'estructura per crear aplicacions, s'ha triat fer ús de l'estructura multi-pàgina, això és que tota l'aplicació es desenvolupa en un sol fitxer HTML, el fitxer **index.html**. No és més que la unió de diverses pàgines bàsiques en un mateix fitxer, i jQuery s'encarregarà de mostrar-les com si fossin pàgines independents.

Per crear aquesta estructura és necessari incloure en el cos de la pàgina diversos blocs amb l'atribut `data-role="page"`. Cada un d'aquests blocs representa una pàgina. També cada pàgina sol tenir unes parts que configuren la seva estructura, aquestes són: **header**, **content** i **footer**, per tant, serà necessari incloure en cada pàgina els blocs que necessiti per crear la vista. En el projecte exposat hi ha 8 vistes, el que es tradueix en la següent estructura:

```
<!-- indexView -->
<div data-role="page" id="indexView">
  <div data-theme="b" data-role="header">
    <h1></h1>
  </div>
  <div data-role="content">
```



# 7 Implementació i proves

```
        <a data-transition="slide"
href="#mapView?current=true" data-theme="c" data-
role="button" data-iconpos="right" data-icon="mappin">Usar mi
localización</a>
        <a data-transition="slide"
href="#provinciasView" data-theme="c" data-role="button"
data-iconpos="right" data-icon="search">Buscar por
provincias</a>
    </div>
    <div data-role="footer" data-theme="b" data-
position="fixed">
        <h4>nineando | actividades infantiles</h4>
    </div>
</div>
<!-- /indexView -->
<!-- mapView -->
<div data-role="page" id="mapView">
    <div data-theme="b" data-role="header"></div>
    <div data-role="content">
        <div class="ui-bar-c ui-corner-all ui-shadow"
style="padding:1em;">
            <div id='map_canvas'
style='height:200px;'></div>
        </div>
    </div>
</div>
<!-- /mapView -->
<!-- eventView -->
<div data-role="page" id="eventView">
    <div data-theme="b" data-role="header"></div>
    <div data-role="content"></div>
    <div data-role="footer" data-theme="b" data-
position="fixed"></div>
</div>
<!-- /eventView -->
<!-- commentView -->
<div data-role="page" id="commentView">
```

```
        <div data-theme="b" data-role="header"></div>
        <div data-role="content"><div id="disqus_thread"
class="disqus-ajax"></div></div>
        <div data-role="footer" data-theme="b" data-
position="fixed"></div>
    </div>
<!-- /commentView -->
<!-- photoView -->
<div data-role="page" id="photoView">
    <div data-theme="b" data-role="header"></div>
    <div data-role="content"></div>
    <div data-role="footer" data-theme="b" data-
position="fixed"></div>
</div>
<!-- /photoView -->
<!-- directionView -->
<div id="directionView" data-role="page">
    <div data-theme="b" data-role="header"></div>
    <div data-role="content"></div>
    <div data-role="footer" data-theme="b" data-
position="fixed"></div>
</div>
<!-- /directionView -->
<!-- provinciasView -->
<div data-role="page" id="provinciasView">
    <div data-theme="b" data-role="header"></div>
    <div data-role="content"></div>
</div>
<!-- /provinciasView -->
<!-- provMapView -->
<div data-role="page" id="provMapView">
    <div data-theme="b" data-role="header"></div>
    <div data-role="content"></div>
</div>
<!-- /provMapView -->
```

# 7 Implementació i proves

Cadascuna de les pàgines bàsiques de l'esquema tenen un identificador, que es posa en el div principal de la pàgina, el que té l'atribut `data-role="page"`. La navegació entre elles està realitzada per enllaços interns, a cada atribut `href` se li haurà d'indicar l'identificador de la pàgina.

S'ha de tenir en compte que la majoria dels comportaments bàsics d'aquest framework es configuren directament sobre el codi HTML, per mitjà de diversos atributs amb el prefix "data-", com per exemple els efectes de transicions entre pàgines. Es pot accedir a l'API del framework per veure totes les opcions de configuració a: <http://jquerymobile.com/demos/1.2.0/index.html>

## 7.1.2 Pàgines dinàmiques

En l'estructura del fitxer `index.html` s'ha pogut comprovar com en moltes de les pàgines no hi havia contingut aparent. Només es veuen els blocs definits de la pàgina:

```
<!-- eventView -->
<div data-role="page" id="eventView">
  <div data-theme="b" data-role="header"></div>
  <div data-role="content"></div>
  <div data-role="footer" data-theme="b" data-
position="fixed"></div>
</div>
<!-- /eventView -->
```

Això és degut a que se injectarà dinàmicament el contingut d'aquestes pàgines depenent de les crides que es realitzin als diferents serveis web. Per realitzar aquest procés s'ha separat l'aplicació en vistes, on cada una d'aquestes té el seu propi fitxer `.js` on s'especifica quina informació haurà de mostrar depenent

dels paràmetres que rebí. Així tenim 7 fitxers JavaScript per cada vista, i 1 que serà l'encarregat de cridar a la funció de la vista demanada:

- `eventView.js`, pàgina `#eventView`
- `commentView.js`, pàgina `#commentView`
- `mapView.js`, pàgina `#mapView`
- `photoView.js`, pàgina `#photoView`
- `directionView.js`, pàgina `#directionView`
- `provinciasView.js`, pàgina `#provinciasView`
- `provMapView.js`, pàgina `#provMapView`
- `controller.js`, fitxer encarregat de fer la crida a la funció corresponent de la pàgina demanada

jQuery Mobile permet injectar contingut a les pàgines de manera dinàmica en el costat del client. Mitjançant l'event `$.mobile.changePage()`, es generen una sèrie de notificacions que permet generar contingut en el moment adequat.

En el projecte exposat es farà ús de l'estat `pagebeforechange`, aquest estat, com el seu nom indica, es crea en el moment abans d'un canvi de pàgina, en aquell moment el controlador, depenent dels paràmetres de la pàgina demanada cridarà a la funció corresponent:

```
// Listen for any attempts to call changePage().
$(document).bind( "pagebeforechange", function( e, data ) {

    if ( typeof data.toPage === "string" ) {
```

# 7 Implementació i proves

Es captura la URL demanda, referenciada per una variable, i se li assigna la funció determinada:

```
var u = $.mobile.path.parseUrl( data.toPage ),
    refEventView = /^#eventView/,
    refCommentView = /^#commentView/,
    refMapView = /^#mapView/,
    refPhotoView = /^#photoView/,
    refDirectionView = /^#directionView/,
    refProvinciasView = /^#provinciasView/,
    refProvMapView = /^#provMapView/;

if ( u.hash.search(refEventView) !== -1 ) {
    showEvent( u, data.options );
    e.preventDefault();
}
if ( u.hash.search(refCommentView) !== -1 ) {
    showComment( u, data.options );
    e.preventDefault();
}

if ( u.hash.search(refMapView) !== -1 ) {
    showMap( u, data.options );
    e.preventDefault();
}

if ( u.hash.search(refPhotoView) !== -1 ) {
    showPhoto( u, data.options );
    e.preventDefault();
}
if ( u.hash.search(refDirectionView) !== -1 ) {
    showDirection( u, data.options );
    e.preventDefault();
}
if ( u.hash.search(refProvinciasView) !== -1 ) {
    showProvincias( u, data.options );
```

```
e.preventDefault();
}
if ( u.hash.search(refProvMapView) !== -1 ) {
    showProvMap( u, data.options );
    e.preventDefault();
```

Com les diferents llibreries es troben carregades, el controlador invocarà la funció i aquesta serà l'encarregada d'omplir de contingut la pàgina corresponent. Cada una d'aquestes funcions es veuran amb més profunditat posteriorment.

## 7.3 Aplicació Zend Framework

Encara que pot semblar que queda fora de l'abast del projecte l'aplicació realitzada amb el framework Zend Framework, es creu necessari realitzar una breu explicació de l'estructura de la mateixa així com dels diferents elements que la componen.

L'aplicació segueix un patró **MVC**, això és:

- **Model:** És la part que s'entén amb la base de dades, per a cada taula es genera una classe que serà el seu Model.
- **Controlador:** S'encarrega de processar l'entrada de l'usuari, en el projecte exposat, de recollir les dades via GET de les URLs demanades. Es processen aquestes dades i produeixen un resultat.
- **Vista:** És la part que s'encarrega de la part que l'usuari veurà, el cas més comú és tenir una pàgina en HTML, però en el cas exposat, la vista pinta les dades en format JSON. El controlador processa tot el necessari i envia els seus resultats a la vista.

# 7 Implementació i proves

L'aplicació té el següent model:

- **Event.php:** model que es comunica amb la taula de la base de dades **events**. Té 2 funcions, public function `getLasts($NELat, $NELng, $SWLat, $SWLng)` i `getById($id)`, la primera retorna activitats que es trobin dintre de les coordenades entrades, i la segona retorna l'activitat que té l'identificador entrat.

I els següents controladors:

- **EventsController.php:** controlador que procesa les crides de demanen dades relacionades amb les activitats emmagatzemades a la base de dades.
- **PhotosController.php:** controlador que procesa les crides de demanen dades relacionades amb les imatges de les activitats. Com es fa ús del servei de Picasa per administrar les imatges, en aquest cas no es comunica amb cap taula o model, sinó que ho fa directament amb el servei de Picasa mitjançant la llibreria PHP `Zend_Gdata`. Aquesta llibreria disposa de diverses classes i mètodes definits que faciliten la comunicació entre aplicacions.

Una vista, per cada acció declarada en els controladors, encarregades de pintar el resultat en format JSON:

- **get-json-lasts.phtml**
- **get-json-by-id.phtml**
- **get-album.phtml**
- **get-json-count-photos-by-album.phtml**

L'aplicació rebrà les crides de la capa de presentació (aplicació jQuery Mobile), en les URLs cridadades s'identifica el controlador i l'acció a executar, a més a més dels paràmetres necessaris. Un exemple:

**URL:**

`http://domini.com/photos/get-album/albumName/21`

En aquest cas el controlador és `photos` i l'acció demanada és `get-album`. A més a més, se li passa el paràmetre `albumName` amb el valor `21`. En aquest cas, s'executa l'acció demanada amb els paràmetres passats.

Si es va a la classe PHP que implementa el controlador:

```
public function getAlbumAction() {
    $this->_helper->layout->setLayout('json');
    $albumName = $this->_getParam('albumName');
    ...
    $service = $this->_getService();
    $query = new Zend_Gdata_Photos_AlbumQuery();
    $query->setUser("default");
    $query->setAlbumName($albumName);
    $query->setThumbsize('144c');
    ...
}
```

es pot veure com es recupera el paràmetre passat, s'instancia el servei (en aquest cas Picasa) i se li passen els paràmetres per obtenir les dades demanades. En aquest cas l'aplicació PHP fa la funció de proxy entre l'aplicació mòbil i Picasa.

## 7.3 Implementació *Cercar activitats per localització*

Tal com s'ha definit anteriorment, el fitxer **controller.js** recollirà la URL demanada abans de canviar de pàgina i invocarà la funció corresponent. En aquest cas el hash és igual a `#mapView`, el fitxer resol cridar a la funció `showMap(urlObj, options)`, accessible des del fitxer personalitzat per la vista, **mapView.js**. A aquesta funció no se li passen paràmetres ja que les dades necessàries per recuperar resultats les facilita la localització que possibilita HTML5 des del propi navegador.

El funcionament és el següent, un cop l'event `pageshow` ha finalitzat, es recupera l'actual posició, si la petició ha acabat amb èxit, es genera el mapa al `div` especificat amb la posició central en la localització obtinguda, amb un **marker** que ens indica la nostra posició:

```
$('#map_canvas').gmap('getCurrentPosition', function
(position, status) {
    if (status === 'OK') {
        var clientPosition = new google.maps.LatLng(position.
coords.latitude, position.coords.longitude);
        $('#map_canvas').gmap('addMarker', {
            'position':clientPosition,
            'bounds':true
        });
        $('#map_canvas').gmap('option', 'zoom', 13);
    }
});
```

Un cop creat el mapa, es capturen els **bounds** d'aquest (coordenades del mapa que es visualitza) per poder realitzar la consulta a l'aplicació PHP i que aquesta ens retorni les activitats que es troben dintre d'aquestes coordenades. La crida Ajax realitzada és del tipus JSONP, necessari quan es realitzen crides a un altre domini:

```
var bounds = map.getBounds();
var NELat = bounds.getNorthEast().lat();
var NELng = bounds.getNorthEast().lng();
var SWLat = bounds.getSouthWest().lat();
var SWLng = bounds.getSouthWest().lng();

$.ajax({
    type: "GET",
    url: 'http://www.kidevents.simplarest.com/events/get-json-
lasts/NELat/' + NELat + '/NELng/' + NELng + '/SWLat/' + SWLat
+ '/SWLng/' + SWLng,
    data: {},
    dataType: "jsonp",
    success: function(data) {
```

De la part del servidor, es crida a l'acció `getJsonLastsAction()` del controlador Events, que rep els paràmetres passats via GET. Aquesta acció crida al mètode `getLasts($NELat, $NELng, $SWLat, $SWLng)` del model Event que serà l'encarregada de realitzar la consulta a base de dades amb els paràmetres passats:

### Controlador:

```
$events = $event->getLasts($NELat, $NELng, $SWLat, $SWLng);
```

### Model:

```
public function getLasts($NELat, $NELng, $SWLat, $SWLng) {
    ...
    $query->from(array('e' => 'events'), array('e.*',
        'username' => 'u.name',
        'userfoto' => 'u.foto',
        'categoriaicon' => 'ca.icon'
    ));
}
```

# 7 Implementació i proves

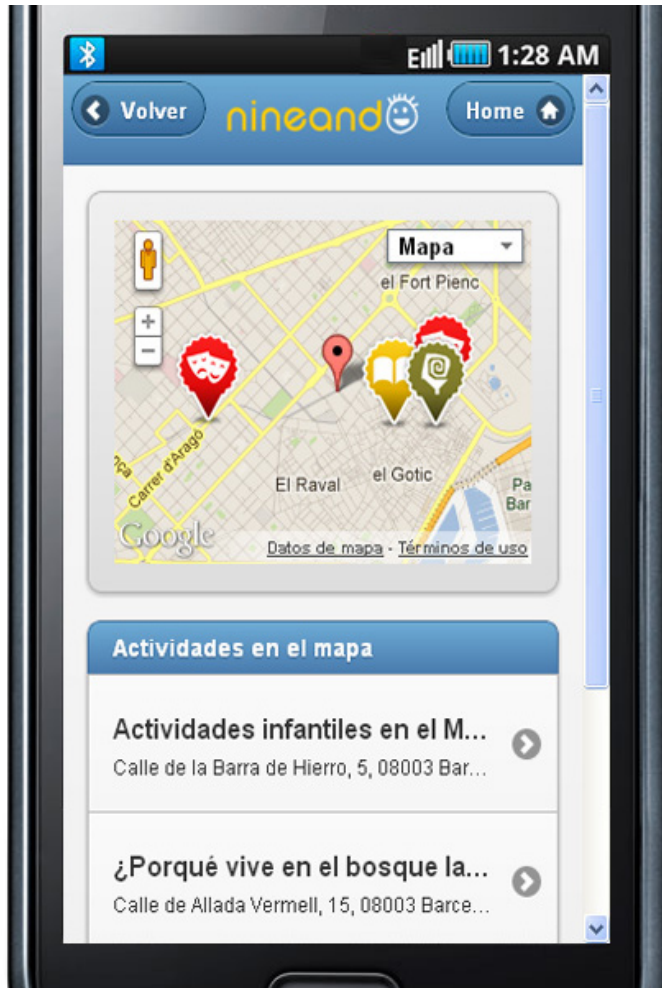


Figura 7.1 Captura de pantalla Cercar activitats per localització

```
$query->join(array('u' => 'users'), 'u.id = e.user_id', array());  
$query->join(array('ca' => 'categorias'), 'ca.id = e.categoria_  
id', array());  
$query->where('e.lat > ?', $SWLat);  
$query->where('e.lat < ?', $NELat);  
$query->where('e.lng > ?', $SWLng);  
$query->where('e.lng < ?', $NELng);  
$query->order('e.created DESC');  
$query->group('e.name');  
$query->limit($numReg);  
$rows = $this->fetchAll($query);  
return $rows->toArray();
```

El controlador, un cop rep els resultats de la query realitzada, s'encarrega de retornar els resultats codificats en format JSON. Un exemple de les dades retornades:

```
jQuery182018785386253148317_1353497714121([{"id": "4", "name": "C  
uentacientos para ni\u00f1os. El misterio del palacio embruja  
do", "description": "Escrito e interpretado por Rosa Mu\u00f1oz  
, "lat": "40.417927", "lng": "-3.689755", "address": "Calle de  
Montalb\u00e0n, 12, 28014 Madrid, Espanya", "created": "2012-09-04  
20:24:52", "modified": "2012-09-04 20:24:52", "date_ini": "2012-09-04  
00:00:00", "date_fin": "2012-09-06 00:00:00", "horari_comment": "d  
edas 4 y 6, a las 11,30 horas", "edad_ini": "3", "edad_fin": "7", "p  
recio": "0.00", "ref": null, "categoria_id": "10", "user_id": "38", "user  
name": "maria", "userfoto": "user.png", "categoriaicon": "cuentos"}]);
```

Amb les dades rebudes, es fa una iteració sobre aquestes per afegir un marker per cada activitat rebuda. Cada marker es situa en les coordenades especificades. En la mateixa iteració s'aprofita per crear dinàmicament una llista amb els mateixos resultats amb l'objectiu de facilitar la navegació:

```
$.each(data, function (i, marker) {  
    $('#map_canvas').gmap('addMarker', {
```

# 7 Implementació i proves

```
        'position':new google.maps.LatLng(marker.lat, marker.
lng) ,
        'icon':'images/markers/' + marker.categoriaicon +
'.png'
    }).click(function () {
        window.location = "#eventView?id=" + marker.id;
    });
    count++;
    //pintar listview
    mid_list+="- <a data-transition='slide'
href='\"#eventView?id=' + marker.id + '\"><h3>' + marker.name+'</
h3><p>' +marker.address+'</p></a></li>';
});

```

## 7.4 Implementació Cercar activitats per províncies

Seguint amb la lògica del fitxer **controller.js**, en aquest cas el hash recuperat és igual a `#provinciasView`, i es crida a la funció `showProvincias(urlObj, options)`. Tampoc hi ha pas de paràmetres ja que el que es fa és la lectura d'un fitxer XML.

El primer que es fa és crear dinàmicament un llistat de províncies a partir d'un fitxer XML, **provincias.xml**:

```
<provincias>
  <provincia>Álava</provincia>
  <provincia>Albacete</provincia>
  <provincia>Alicante</provincia>
  <provincia>Almería</provincia>
  <provincia>Ávila</provincia>
  <provincia>Badajoz</provincia>
  ...
```



Figura 7.2 Captura de pantalla Cercar activitats per províncies

# 7 Implementació i proves

## Crida JavaScript:

```
//xml data
$.get('data/provincias.xml', function(data) {
```

Aprofitant les funcionalitats del framework jQuery Mobile, a la llista creada se li afageix l'atribut `data-filter="true"`, el qual proporciona un filtratge de resultats, això és que davant un llistat de més de 50 resultats, mitjançant un input de cerca, l'aplicació suggereix resultats coincidents amb les primeres lletres inserides, facilitant així una cerca més ràpida.

Finalment cada element de la llista enllaçarà a la pàgina `#provMapView`, passant-li com a paràmetre el nom de la província, suficient per a que el servei de Google Maps ens resolgui la creació del mapa:

```
<li><a data-transition="slide" href="#provMapView?prov=' +
arrayProv[i] + '>' + arrayProv[i] + '</a></li>
```

Quan l'usuari enllaça amb una província, s'acaba cridant a la vista `#provMapView` que el controller resol invocant la funció `showProvMap(u, data.options)`. Aquesta funció que rep el paràmetre de la província a cercar, resol la localització de la mateixa i genera el mapa corresponent:

```
$('#map_canvas_prov'+idProv).gmap('search',{
  'address':provincia,'region': 'ES'
}),function(results, status) {
  if ( status === 'OK' ) {
    //obtenemos latlng d la provincia
    var latlng=results[0].geometry.location;
    //situamos mapa
    $('#map_canvas_prov'+idProv).gmap('option',
'zoom', 9);
    map.panTo(latlng);
```



Figura 7.3 Captura de pantalla Cercar activitats per províncies



En aquest punt, l'aplicació respon igual que la vista anterior. Es capturen els **bounds** del mapa i es passen com a paràmetres per a que l'aplicació PHP resolgui quines activitats que es troben dintre d'aquestes coordenades.

La vista acaba mostrant els markers trobats en el mapa i genera una llista dinàmicament de les diferents activitats per facilitar la navegació.

## 7.5 Implementació *Veure fitxa activitat*

El hash recuperat pel controlador és `#eventView`, en aquest cas si que hi ha pas de paràmetres, l'identificador numèric de l'activitat. S'invoca a la funció `showEvent(urlObj, options)`, del paràmetre `urlObj` s'extreu l'identificador de l'activitat:

```
var idEvent = urlObj.hash.replace(/.*id=/, ""); //id del event
```

Un cop es recupera l'id, es realitza un crida **Ajax** al servei creat al servidor PHP, aquest haurà de retornar en format JSON les dades referents a l'activitat:

```
$.ajax({
  type: "GET",
  url: 'http://kidevents.simplarest.com/events/get-json-by-id/id/' + idEvent,
  data: {},
  dataType: "jsonp",
  success: function (data) {
```

De la part del servidor, es crida a l'acció `getJsonByIdAction()` del controlador `Events`, que rep els paràmetres passats via GET, en aquest cas l'identificador únic de l'activitat. Aquesta acció cridarà al mètode `getById($id)` del model

Event que serà l'encarregada de realitzar la consulta a base de dades:

**Controlador:**

```
//recuperamos param via GET
$id = $this->_getParam('id');
$aEvent = $event->getById($id);
```

**Model:**

```
public function getById($id) {
    ...
    $query->from(array('e' => 'events'), array('e.*',
        'catname' => 'ca.name',
        'caticon' => 'ca.icon'));
    $query->join(array('ca' => 'categorias'), 'ca.id =
e.categoria_id', array());
    $query->where('e.id = ?', $id);
    $query->limit(1);
    $row = $this->fetchRow($query);
    return $row->toArray();
}
```

Un cop la crida ha fet **success**, es genera dinàmicament el contingut de la pàgina amb les dades rebudes:

```
success:function (data) {
    ...
    // area of the page.
    markup = "<h2>" + data.name + "</h2>" +
        "<p>" + data.description + "</p>" +
        "<ul data-role='listview' data-theme='d' data-
inset='true'>" +
        "<li data-role='list-divider'>Ficha</li>" +
        "<li><img src='images/icons/situation.png'
```

# 7 Implementació i proves



Figura 7.4 Captura de pantalla Veure fitxa activitat

```
alt='Direcció' class='ui-li-icon'" + data.address + "</li>"
+
  "<li><img src='images/icons/fecha.png' alt='Fecha'
class='ui-li-icon'" + getDateFormatted(data.date_ini, data.
date_fin) + "</li>" +
  "<li><img src='images/icons/horario.png' alt='Horario'
class='ui-li-icon'" + data.horari_comment + "</li>" +
  "<li><img src='images/icons/edad.png' alt='Edades
recomendadas' class='ui-li-icon'">De " + data.edad_ini + " a "
+ data.edad_fin + " años</li>" +
  "<li><img src='images/icons/euro.png' alt='Precio'
class='ui-li-icon'" + data.precio + "</li>" +
"</ul>";
```

## 7.6 Implementació Veure / Afegir comentaris

Com ja s'ha especificat anteriorment, la gestió de comentaris es realitza a través del servei que ofereix Disqus.

Cal remarcar que per poder fer ús d'aquest servei, totalment gratuït, és necessari en primer lloc registrar l'aplicació, a partir d'aquí hi ha dues maneres d'identificar cada fil de discussió, una mitjançant la URL, tècnica usada en pàgines web, i l'altre mitjançant un identificador únic. Tenint en compte que s'està treballant sobre una aplicació mòbil i que per tant no disposem de URLs per cada pàgina mostrada, la manera més adient que s'ha trobat és fer servir el mateix identificador de cada activitat per identificar cada fil que a la vegada és un fil de dita activitat.

En aquest cas el hash recuperat és `#commentView`, d'aquí es passà a la funció `showComment(urlObj, options)`, del paràmetre `urlObj` s'extreu l'identificador del comentari, el mateix que el de l'activitat relacionada:



Figura 7.5 Captura de pantalla Veure / Afegir comentaris

```
var idComment = urlObj.hash.replace(/.*id=/, ""); //id del comment
```

Un cop obtenim l'identificador, s'assigna a la variable `disqus_identifier` per a que el servei s'apiga quin fil ha de carregar:

```
//disqus
disqus_identifier = idComment;
```

Tenint en compte l'arquitectura de l'aplicació, s'ha trobat un problema el qual afectava a la càrrega correcta del fil específic de cada activitat, ja que no és possible carregar en una mateixa pàgina més d'una instància de disqus. Per evitar aquest problema es fa una comprovació en cada càrrega per determinar si s'ha instanciat prèviament o no:

```
if( typeof( DISQUS ) == 'undefined' )
    (function () {
        var dsq = document.createElement('script');
        dsq.type = 'text/javascript';
        dsq.async = true;
        dsq.src = 'http://' + disqus_shortcode +
        '.disqus.com/embed.js';
        (document.getElementsByTagName('head')[0] ||
        document.getElementsByTagName('body')[0]).appendChild(dsq);
    })();

if( typeof( DISQUS ) !== 'undefined' ){
    DISQUS.reset({
        reload: true,
        config: function () {
            this.page.identifier = disqus_identifier;
        }
    });
}
```

# 7 Implementació i proves

Un cop es carrega el servei en el div especificat, per part de l'aplicació no s'ha de fer gaire més, en el sentit que tot es gestiona via Disqus: login d'usuaris mitjançant les xarxes socials amb més popularitat, comentaris anònims, la possibilitat de puntuar l'activitat... També és interessant l'opció que dona Disqus de moderació de comentaris. Al ser un usuari registrat, es disposa d'un espai web personal des d'on l'administrador pot eliminar els missatges que consideri inapropiats així com banejar als usuaris que cregui convenient.

## 7.7 Implementació *Veure imatges*

El hash recuperat pel controlador és `#photoView`, igualment com en altres vistes hi ha un pas de paràmetres, l'identificador numèric de l'activitat. S'invoca a la funció `showPhoto(urlObj, options)`, del paràmetre `urlObj` s'extreu l'identificador de l'activitat:

```
var idEvent = urlObj.hash.replace(/.*id=/, ""); //id del event
```

Un cop es recupera l'id, igualment que en la vista d'event, es realitza un crida **Ajax** al servei creat al servidor PHP, aquest haurà de retornar en format JSON les imatges vinculades a l'activitat:

```
$.ajax({
  type: "GET",
  url: 'http://kidevents.simplarest.com/photos/get-album/albumName/' + idEvent,
  data: {},
  dataType: "jsonp",
  success: function (data) {
```

De la part del servidor, es crida a l'acció `getAlbumAction()` del controlador Photos, que rep els paràmetres passats via GET, en aquest cas l'identificador únic de l'activitat:

```
public function getAlbumAction() {
    $this->_helper->layout->setLayout('json');
    $albumName = $this->_getParam('albumName');
    ...
    //atacamos picasa
    $service = $this->_getService();
    $query = new Zend_Gdata_Photos_AlbumQuery();
    $query->setUser("default");
    $query->setAlbumName($albumName);

    $query->setThumbsize('144c');
    ...
}
```

Com es pot comprovar, s'instancia el servei web de Picasa que és qui realment servei les imatges a l'aplicació. Es setea la variable `$albumName`, que no és més que l'identificador de l'activitat, assegurant així un nom únic per cada àlbum. Òbviament això ens obliga a crear un àlbum per cada activitat que s'afegeixi a la base de dades.

Un altre detall, és l'especificació de la mida de les miniatures que es mostraran a l'aplicació, dintre de les opcions disponibles s'ha triat el format quadrat de 144 x 144 píxels:

```
$query->setThumbsize('144c');
```

Les dades que es recuperen a l'acció PHP i que es parsejen posteriorment per servi el JSON que consumirà l'aplicació, són les següents:

# 7 Implementació i proves



Figura 7.6 Captura de pantalla Veure imatges

```
$entryArray = array(  
    'thumbnail0' => $thumbnail[0]->url,  
    'big0' => $big[0]->url,  
    'title' => $title,  
    'tag' => $tags->text  
);
```

Que es tradueix en JSON per un sol ítem d'imatge a:

```
{"thumbnail0": "https://lh4.googleusercontent.com/-  
TuHgGzYRmRc/UKp7ZoH7V6I/AAAAAAAAAIk/z7jcXhyPTds/  
s144-c/50aa7b662a5c3.jpg", "big0": "https://lh4.  
googleusercontent.com/-TuHgGzYRmRc/UKp7ZoH7V6I/  
AAAAAAAAAIk/z7jcXhyPTds/50aa7b662a5c3.jpg", "title": "Sin t\  
u00edtulo", "tag": "No especificado"}
```

Es pot veure com les URLs que serviran las imatges són de la **CDN** de google, en aquest cas `thumbnail0` per la miniatura i `big0` per la imatge ampliada. A més a més, també es recupera dos camps de text que serviran per mostrar el títol de la foto i el seu autor.

Un cop l'aplicació mòbil rep les dades, dinàmicament es genera el llistat de miniatures que enllacen amb les imatges ampliades. Es realitza una iteració sobre els resultats obtinguts:

```
$.each(data, function (i, photo) {  
    mid_listPhotos=mid_listPhotos + '<li><a  
href="#popupPhoto'+ i + idEvent + '" data-rel="popup"  
data-position-to="window" data-inline="true" data-  
transition="pop">' + '' + '<h3>' + photo.title + '</h3>' + '<p>Autor: <strong>' +  
photo.tag+ '</strong></p>';
```

```
popup=>popup+'<div data-role="popup"
```

# 7 Implementació i proves



Figura 7.7 Captura de pantalla Veure imatges

```
id="popupPhoto'+i+ idEvent+'" data-overlay-theme="a"
data-theme="d" data-corners="false">' + '<a href="#"
data-rel="back" data-role="button" data-theme="a" data-
icon="delete" data-iconpos="notext" class="ui-btn-
right">Close</a>'+ '' + '</div>';

});
```

## 7.8 Implementació Afegir imatges

Sense sortir de la vista #photoView, hi ha un botó que dóna accés a que l'usuari pugui afegir una imatge. El formulari, al igual que tot el contingut que es veu a la pàgina, es crea de manera dinàmica al accedir a la vista:

```
var form='<form action="http://kidevents.simplarest.com/
photos/add-photo" method="post" id="formAddPhoto'+idEvent+'"
data-ajax="false" enctype="multipart/form-data">'+
'<div style="padding:10px 20px;">'+
'<h3>Añadir foto</h3>'+
'<label for="file" class="ui-hidden-accessible">Selecciona la
foto:</label>'+
'<input type="file" name="file" id="file" value=""
placeholder="foto" data-theme="a" />'+
'<label for="title" class="ui-hidden-accessible">Título
foto:</label>'+
'<input type="text" name="title" id="title" value=""
placeholder="Título foto" data-theme="a" />'+
'<label for="autor" class="ui-hidden-accessible">Tu nombre:</
label>'+
'<input type="text" name="autor" id="autor" value=""
placeholder="Tu nombre" data-theme="a" />'+
'<input type="hidden" name="album" id="album"
value="'+idEvent+'>' +
```

# 7 Implementació i proves

```
'<input type="hidden" name="urlToRedirect" id="urlToRedirect"
value="'+urlToRedirect+'"'>' +
'<div id="messageAlert"></div>'+
'<button id="btSubmit" type="submit" data-theme="b">Añadir</
button>'+
'</div>'+
'</form>'+
'</div>';
```

Com es pot apreciar, hi ha una sèrie d'inputs que s'acabaran enviant via **POST** al servidor PHP, bàsicament la imatge carregada des del sistema de fitxers del dispositiu mòbil, un títol i el nom de l'autor. També hi ha un camp del tipus hidden, que és el que conté l'identificador de l'activitat, que com ja s'ha definit anteriorment, és el mateix que el nom de l'àlbum on s'afegirà la imatge.

Quan el servidor rep les dades, s'està cridant a l'acció `addPhotoAction()` del controlador Photos, que com s'ha dit abans rep els paràmetres passats via POST. S'instancia el servei de Picasa, i mitjançant els mètodes que disposa la llibreria, s'actualitza l'àlbum corresponent:

```
public function addPhotoAction() {
    ...

    $albumName = $this->_getParam('album');
    //imagen a upload
    $filename = $_FILES['file']['tmp_name'];
    //nombre random imagen
    $photoName = uniqid();
    //título de la foto
    $photoCaption = $this->_getParam('title');
    //autor q lo añadimos como tag de la foto
    $photoTags = $this->_getParam('autor');
    ...
}
```

```
$fd = $service->newMediaFileSource($filename);
$fd->setContentType("image/jpeg");
//Create a PhotoEntry
$photoEntry = $service->newPhotoEntry();
$photoEntry->setMediaSource($fd);
$photoEntry->setTitle($service->newTitle($photoName));
$photoEntry->setSummary($service->
    >newSummary($photoCaption));
...
}
```

## 7.9 Implementació Com arribar

Igualment com en les altres vistes, el fitxer **controller.js** recollirà la URL demanada abans de canviar de pàgina i invocarà la funció corresponent. En aquest cas el hash és igual a `#directionView`, el fitxer resol cridar a la funció `showDirection(urlObj, options)`, accessible des del fitxer personalitzat per la vista, **directionView.js**. A aquesta funció sí se li passen paràmetres, concretament l'identificador de l'activitat, necessari per recuperar l'adreça, que conjuntament amb la localització que possibilita HTML5 des del propi navegador, serviran per generar les indicacions del *Com arribar*.

Recuperem idint:

```
var idEvent = urlObj.hash.replace(/.*id=/, ""); //id del event
```

Resolem la nostra localització, que esdevindrà el punt d'origen de l'itinerari:

```
$('#map_canvas_direction'+idEvent).gmap('getCurrentPosition',
function (position, status) {
    if (status === 'OK') {
        ...
    }
});
```

# 7 Implementació i proves

Un cop tenim les coordenades d'on s'està situat, es fa servir el servei de Google Maps per resoldre l'adreça. En aquest cas s'assigna a la variable `desde`, l'adreça d'origen:

```
var desde='';
$('#map_canvas_direction'+idEvent).gmap('search',{
    'location':latlng
},function(results, status) {
    if ( status === 'OK' ) {
        desde=results[0].formatted_address;
        ...
    }
});
```

Posteriorment es realitza un crida Ajax al servei del servidor PHP, que mitjançant l'identificador de l'activitat, retornarà en format JSON les coordenades necessàries per obtenir l'adreça de destí, que s'assigna a la variable `hasta`:

```
$.ajax({
    type:"GET",
    url:'http://kidevents.simplarest.com/events/get-json-by-id/id/' + idEvent,
    data:{},
    dataType:"jsonp",
    success:function (data) {
        var latEvent=data.lat;
        var lngEvent=data.lng;
        var latlngHasta=new google.maps.
LatLng(latEvent,lngEvent);
        $('#map_canvas_direction'+idEvent).
gmap('search',{
    'location':latlngHasta
},function(results, status) {
    if ( status === 'OK' )
        hasta=results[0].formatted_address;
```



Figura 7.8 Captura de pantalla Com arribar



# 7 Implementació i proves

Un cop es disposa de les dues adreces, es realitza una crida al servei de Google Maps que resol les indicacions i genera el mapa marcant el recorregut ha realitzar:

```
$('#map_canvas_direction'+idEvent).gmap('displayDirections',  
{  
  'origin': desde,  
  'destination': hasta,  
  'travelMode': google.maps.DirectionsTravelMode.DRIVING  
},
```

Un altre punt necessari, és indicar en quin div s'hauran de generar les indicacions, aquest fet s'especifica en l'atribut `panel`:

```
{ 'panel': document.getElementById('directions') },
```

## 7.10 Proves

En aquest apartat es defineixen les proves que s'han realitzat per complir amb els requisits descrits a l'apartat de requeriments per cada història d'usuari.

### 7.10.1 Cercar activitats per localització

**Comprovar que la nostra ubicació és correcte:**

La manera de comprovar que la nostra ubicació és correcte, és assegurar-se que la localització que possibilita HTML5 des del propi navegador és exactament on es troba l'usuari.

**Comprovar que les activitats que em mostra són correctes:**

La manera d'assegurar-se que les dades que es mostren són correctes, és realitzar manualment la query que es realitza a base de dades. Les dades retornades han de ser iguals amb les mostrades en el mapa, tenint en compte que caldrà introduir les coordenades que delimiten el mapa mostrat.

Un exemple de dades retornades. En aquest cas, s'obtenen 4 resultats:

<http://www.domini.com/events/get-json-lasts/NELat/41.38843215410564/NELng/2.28733438183599/SWLat/41.36266944881427/SWLng/2.07859414746099>

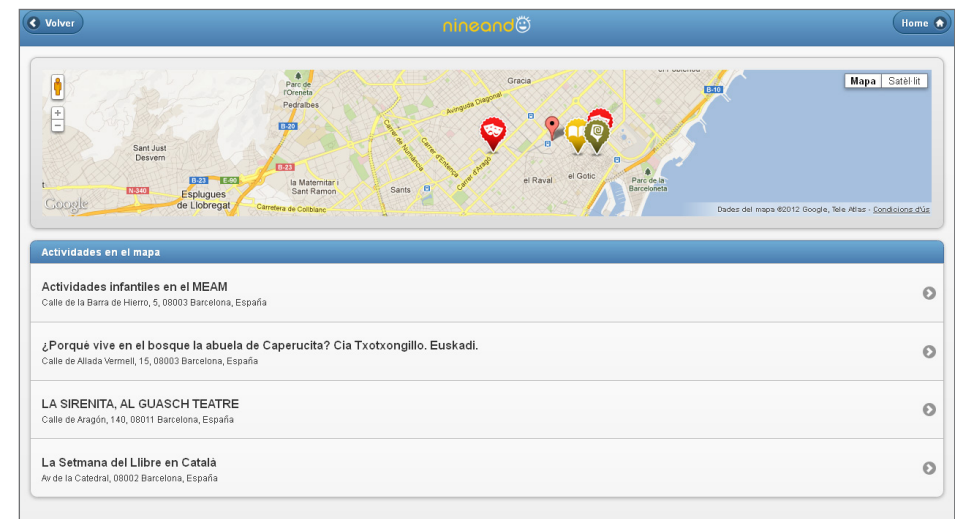


Figura 7.9 Proves resultats Cercar activitats per localització

# 7 Implementació i proves

Si s'executa la mateixa query que fa servir l'aplicació PHP per retornar les dades, els resultats tindrien de ser els mateixos:

```
SELECT `e`.`*`,`u`.`name` AS `username`,`u`.`foto` AS `userfoto`,`ca`.`icon` AS `categoriaicon`
FROM `events` AS `e`
INNER JOIN `users` AS `u` ON u.id = e.user_id
INNER JOIN `categorias` AS `ca` ON ca.id = e.categoria_id
WHERE (
e.lat > '41.36266944881427'
)
AND (
e.lat < '41.38843215410564'
)
AND (
e.lng > '2.07859414746099'
)
AND (
e.lng < '2.28733438183599'
)
GROUP BY `e`.`name`
ORDER BY `e`.`created` DESC
LIMIT 20
```

I efectivament, els resultats del **SGBD** són els mateixos com es pot comprovar en la captura de pantalla.

## 7.10.2 Cercar activitats per províncies

**Comprovar que el llistat de províncies és correcte:**

El llistat de províncies que es desplega en la vista no és un codi fixe que es pugui trobar en la pàgina HTML, sinó que es carrega d'un fitxer XML, concretament es pot accedir a ell: <http://domini/data/provincias.xml>

id	name	description	lat	lng	address	created
21	Actividades infantiles en el MEAM	Desde el Museu Europeu d'Art Modern se quiere pote...	41.384933	2.180397	Calle de la Barra de Hierro, 5, 08003 Barcelona, E...	2012-09-16 09:38:08
20	¿Porqué vive en el bosque la abuela de Caperucita?...	Xabi, un chaval de 8 años no está de acuerdo con l...	41.386803	2.181202	Calle de Allada Vermell, 15, 08003 Barcelona, Espa...	2012-09-09 12:30:15
19	LA SIRENITA, AL GUASCH TEATRE	L'Ariel és una jove sirena que viu amb els seus am...	41.385342	2.156027	Calle de Aragón, 140, 08011 Barcelona, España	2012-09-09 12:25:40
17	La Setmana del Llibre en Català	A la 30a Setmana del Llibre en Català hi trobaràs ...	41.384991	2.175923	Av de la Catedral, 08002 Barcelona, España	2012-09-09 12:08:13

Figura 7.10 Proves query Cercar activitats per localització

En aquest sentit la comprovació que s'ha realitzat és assegurar-se de que el llistat de la vista coincideix exactament amb el fitxer XML:

```
<provincias>
<provincia>Álava</provincia>
<provincia>Albacete</provincia>
<provincia>Alicante</provincia>
<provincia>Almeria</provincia>
<provincia>Ávila</provincia>
```

# 7 Implementació i proves

```
<provincia>Badajoz</provincia>
<provincia>Balears (Illes)</provincia>
<provincia>Barcelona</provincia>
...
```

I si accedim a la vista:



Figura 7.11 Proves llistat Cercar activitats per provincies

**Comprovar que es despleguen al mapa de la província les activitats correctes:**

Un cop s'accedeix al mapa de la província, les activitats que es mostren en ell, provenen del mateix servei que en la vista exposada anteriorment, això vol dir que la query a executar és la mateixa amb idèntics paràmetres d'entrada, les coordenades del mapa.

El que si quedava era un punt a comprovar, que realment si es demana una província concreta, el mapa que es mostri sigui d'aquella província. Per exemple, davant aquesta URL: <http://domini/#provMapView?prov=Albacete>, el mapa ha de mostrar la província d'Albacete (cal tenir en compte el zoom que s'ha definit i la mida de la pantalla del dispositiu, ja que el que fa realment és centrar en el mapa la localització definida).



Figura 7.12 Proves mapa Cercar activitats per provincies

# 7 Implementació i proves

## 7.10.3 Veure fitxa activitat

**Comprovar que que des d'el mapa es pot accedir a les diferents fitxes de cada activitat:**

Quan un mapa pinta els resultats trobats, s'ha comprovat que cada marker, així com cada element llistat, enllacen a la fitxa corresponent.

**Comprovar que les dades facilitades són correctes i en el format esperat:**

Un altre cop, la manera d'assegurar-se que les dades que es mostren són correctes, és realitzar manualment la query que es realitza a base de dades.

Un exemple:

<http://www.domini.com/#eventView?id=21>

The screenshot shows a mobile application interface for 'Actividades infantiles en el MEAM'. The header includes a 'Volver' button and the 'nineando' logo. The main content area contains a description: 'Desde el Museo Europeo d'Art Modern se quiere potenciar la imaginación y creatividad de nuestros hijos. Un domingo al mes, todos aquellos niños de 5 a 12 años en un taller coordinado por monitores de Biribots.' Below this is a 'Ficha' section with a blue header, listing details: location (Calle de la Barra de Hierro, 5, 08003 Barcelona, España), dates (Del 16/9/2012 al 16/12/2012), time (De 12:00 a 12:30), age group (De 5 a 12 años), and price (€ 5€).

Figura 7.13 Proves fitxa *Veure fitxa activitat*

Si executem la query:

```
SELECT `e`.*, `ca`.`name` AS `catname`, `ca`.`icon` AS `caticon` FROM `events` AS `e` INNER JOIN `categorias` AS `ca` ON ca.id = e.categoria_id WHERE (e.id = '21') LIMIT 1
```

The screenshot shows a SQL query execution interface. The query is: `SELECT `e`.*, `ca`.`name` AS `catname`, `ca`.`icon` AS `caticon` FROM `events` AS `e` INNER JOIN `categorias` AS `ca` ON ca.id = e.categoria_id WHERE (e.id = '21') LIMIT 1`. Below the query, there are controls for 'Mostra: 30', 'Files començant des de 0', and 'en mode horizontal'. The results are displayed in a table with columns: id, name, description, lat, lng, address, created, modified, date\_ini, date\_fin, and horari\_comment. The first row shows data for id=21.

id	name	description	lat	lng	address	created	modified	date_ini	date_fin	horari_comment
21	Actividades infantiles en el MEAM	Desde el Museo Europeo d'Art Modern se quiere potenciar la imaginación y creatividad de nuestros hijos. Un domingo al mes, todos aquellos niños de 5 a 12 años en un taller coordinado por monitores de Biribots.	41.384933	2.180397	Calle de la Barra de Hierro, 5, 08003 Barcelona, E...	2012-09-16 09:38:08	2012-09-16 09:38:08	2012-09-16 00:00:00	2012-12-16 00:00:00	De 12:00 a 12:30

Figura 7.14 Proves query *Veure fitxa activitat*

## 7.10.4 Veure / Afegir comentaris

**Comprovar que el servei web de disqus s'executa correctament:**

La manera de comprovar aquest requeriment es veure que realment es carrega correctament el servei en l'apartat de comentaris d'una fitxa d'activitat.

**Comprovar que hi ha persistència de les dades (comentaris):**

Lligat amb el requeriment anterior, és necessari publicar un comentari i quan

# 7 Implementació i proves

s'accedeix novament a la fitxa, el comentari ha de sortir recollit en el fil de discussió.



Figura 7.15 Proves Veure / Afegir comentaris

## 7.10.5 Veure imatges

**Comprovar que el servei web de picasa s'executa correctament:**

La manera de comprovar aquest requeriment es veure que realment es carreguen les imatges en l'apartat de fotos d'una fitxa d'activitat.



Figura 7.16 Proves llistat Veure imatges

**Comprovar que el llistat i els pop up de les imatges són correctes:**

En aquest cas, disposem d'una eina molt útil. Picasa permet accedir via Web als àlbums d'un usuari, si accedim a l'àlbum de l'activitat en concret, les imatges que surtin han de coincidir amb les que es mostra en l'aplicació.

# 7 Implementació i proves



Figura 7.17 Proves Picasa Veure imatges

## 7.10.6 Afegir imatges

**Comprovar que el servei web de picasa s'executa correctament:**

Tenint en compte que s'utilitzen llibreries per l'ús del servei, s'ha comprovat que en cap moment s'ha llençat cap excepció que pugui invalidar el procés.

**Comprovar que hi ha persistència de les dades (imatges):**

La millor manera de comprovar aquest requeriment és pujant una imatge, sortir de la fitxa, tornar a entrar i comprovar que la imatge afegida anteriorment sur en el llistat de fotos.

**Comprovar que es possible carregar imatges del dispositiu i pujarles al servidor:**

En aquest cas, el que s'ha fet és accedir des d'un dispositiu mòbil a l'aplicació via navegador. Cal apuntar que el dispositiu ha de tenir el SO Android, ja que iOS, no permet l'accés al sistema de fitxers. La prova ha sigut totalment satisfactòria.



Figura 7.18 Proves Afegir imatges

# 7 Implementació i proves

## 7.10.7 Com arribar

### Comprovar que el servei web de google s'executa correctament:

La manera de comprovar aquest requeriment es veure que realment es carrega correctament al mapa el trajecte esperat, així com que es llisten numerades les indicacions a seguir.

### Comprovar, dintre del que sigui possible, que les instruccions que es donen són correctes:

Aquest punt té una dificultat extra en la seva comprovació, ja que no és possible comprovar un a un els itineraris proposats, però tenint en compte les activitats que es disposen a prop de la nostra posició, es pot assegurar amb bastanta fiabilitat que les indicacions són del tot correctes.



The screenshot shows the 'nineando' mobile application interface. At the top, there are navigation buttons for 'Volver' and 'Home'. The main area displays a Google Map of Barcelona with a blue route starting from a green pin labeled 'A' at Plaça de Catalunya. The route passes through several streets, ending at a red pin labeled 'H' at Hospital del Mar. Below the map, there is a section titled 'Indicaciones' with a list of five directions. At the bottom, there is a blue bar with a car icon and the text 'Cómo llegar'.

Indicaciones		
A	Plaça de Catalunya, 11, 08002 Barcelona, Espanya	
1.	Direcció <b>sud-oest</b> de <b>Plaça de Catalunya</b> cap a <b>Rambla de Catalunya</b>	0,1 km
2.	Continueu per <b>Ronda de la Universitat</b>	0,2 km
3.	Gireu a l' <b>esquerra</b> per <b>Carrer de Balmes</b>	0,1 km
4.	Gireu cap a l' <b>esquerra</b> per <b>Carrer de Pelai</b>	0,2 km
5.	Continueu endavant per <b>Plaça de Catalunya</b> .	0,2 km

Figura 7.19 Proves *Com arribar*

# 7 Implementació i proves

## 7.10 Implementació Aplicació Android

Com ja s'ha definit amb anterioritat, per realitzar la implementació de l'aplicació en format Android, es farà ús de la tecnologia PhoneGap que permet crear aplicacions natives amb tecnologies web. Es pot descarregar en la següent adreça: <http://phonegap.com/download>

En primer lloc, mitjançant l'IDE Eclipse, es crearà un nou projecte de tipus **Android Application Project**.

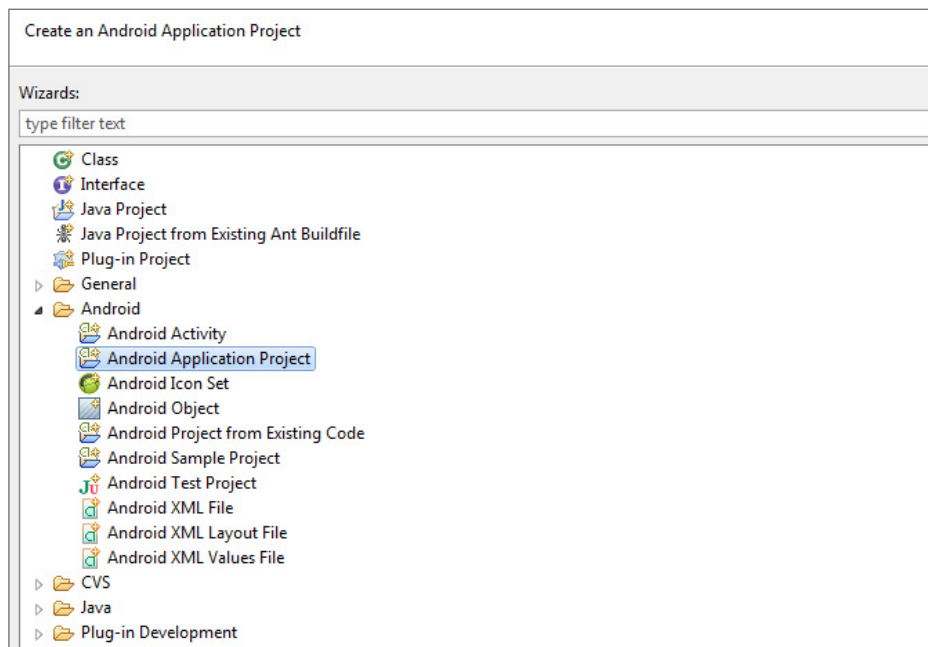


Figura 7.20 Creació projecte Android

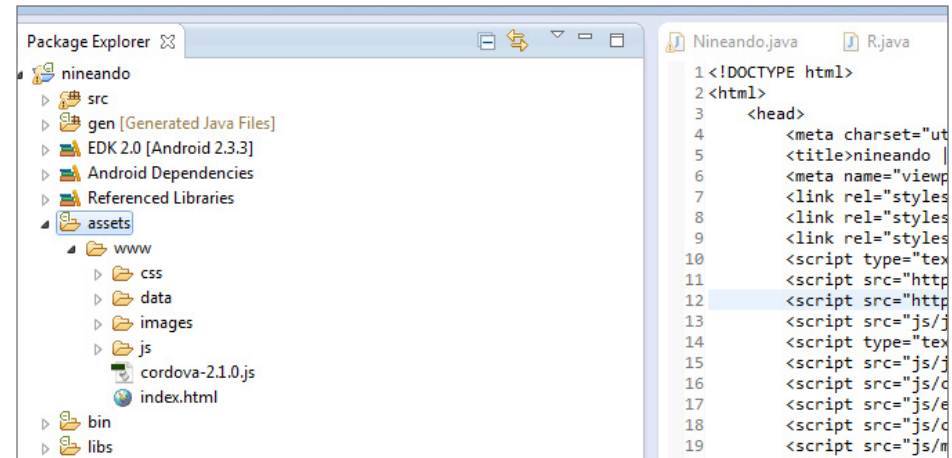


Figura 7.21 Directoris projecte Android

En el directori **assets**, es crearà un nou directori anomenat **www**, on inclourem el nostre projecte creat amb jQuery Mobile. També s'inclourà la llibreria JS de PhoneGap, anomenada Cordova. Caldrà modificar el fitxer **index.html** del nostre projecte per tal d'incloure la llibreria:

```
<script type="text/javascript" charset="utf-8" src="cordova-2.1.0.js"></script>
```

També és necessari copiar els següents fitxers de PhoneGap al projecte Android: **cordova-2.2.0.jar** al directori **libs** i el directori **xml** dintre del directori **res**. Seguidament caldrà afegir al **Build Path** del projecte la llibreria **jar** de Cordova.



# 7 Implementació i proves

Un cop tenim l'entorn preparat, caldrà modificar la classe principal del projecte. En primer lloc, afegir una sentència import per **org.apache.cordova.\***, després, canviar la classe base de **Activity** a **DroidGap**, i per últim, reemplaçar la crida **setContentView()** per una referència a la carrega del fitxer **index.html** de l'aplicació:

```
import org.apache.cordova.*;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;

public class Nineando extends DroidGap {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        super.loadUrl("file:///android_asset/www/index.html");
    }
    ...
}
```

Per últim, caldrà configurar les metadades del projecte. En el fitxer **AndroidManifest.xml** s'afegeixen els següents nodes:

```
<supports-screens android:largeScreens="true"
android:normalScreens="true" android:smallScreens="true"
android:resizeable="true" android:anyDensity="true" />

<uses-permission android:name="android.permission.CAMERA"
/> <uses-permission android:name="android.permission.
VIBRATE" /> <uses-permission android:name="android.
permission.ACCESS_COARSE_LOCATION" /> <uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION"
/> <uses-permission android:name="android.permission.
```

```
ACCESS_LOCATION_EXTRA_COMMANDS" /> <uses-permission
android:name="android.permission.READ_PHONE_STATE" /> <uses-
permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.RECEIVE_
SMS" /> <uses-permission android:name="android.permission.
RECORD_AUDIO" /> <uses-permission android:name="android.
permission.MODIFY_AUDIO_SETTINGS" /> <uses-permission
android:name="android.permission.READ_CONTACTS" /> <uses-
permission android:name="android.permission.WRITE_CONTACTS"
/> <uses-permission android:name="android.permission.WRITE_
EXTERNAL_STORAGE" /> <uses-permission android:name="android.
permission.ACCESS_NETWORK_STATE" /> <uses-permission
android:name="android.permission.GET_ACCOUNTS" /> <uses-
permission android:name="android.permission.BROADCAST_STICKY"
/>
```

Al node **<activity>**, s'afegeix el següent atribut:

```
<supports-screens android:largeScreens="true"
```

I seguidament, s'incorpora un altre node **<activity>**:

```
<activity android:name="org.apache.cordova.DroidGap"
android:label="@string/app_name" android:configChanges="ori
entation|keyboardHidden"> <intent-filter></intent-filter> </
activity>
```

# 7 Implementació i proves

Un cop realitzades totes les configuracions, es prova l'aplicació en un dispositiu compatible amb les especificacions triades. En aquest cas concret, com es disposa d'un dispositiu físic amb SO Android, es tria aquest per fer les proves pertinents:

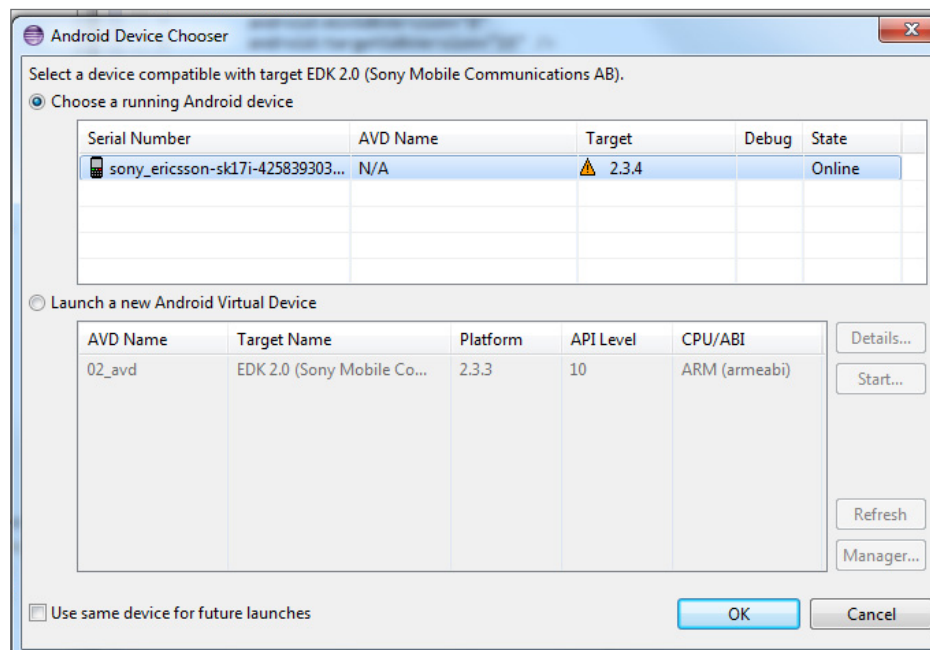


Figura 7.22 Selecció dispositiu Android

D'aquesta manera s'aconsegueix provar l'aplicació en un entorn totalment real:

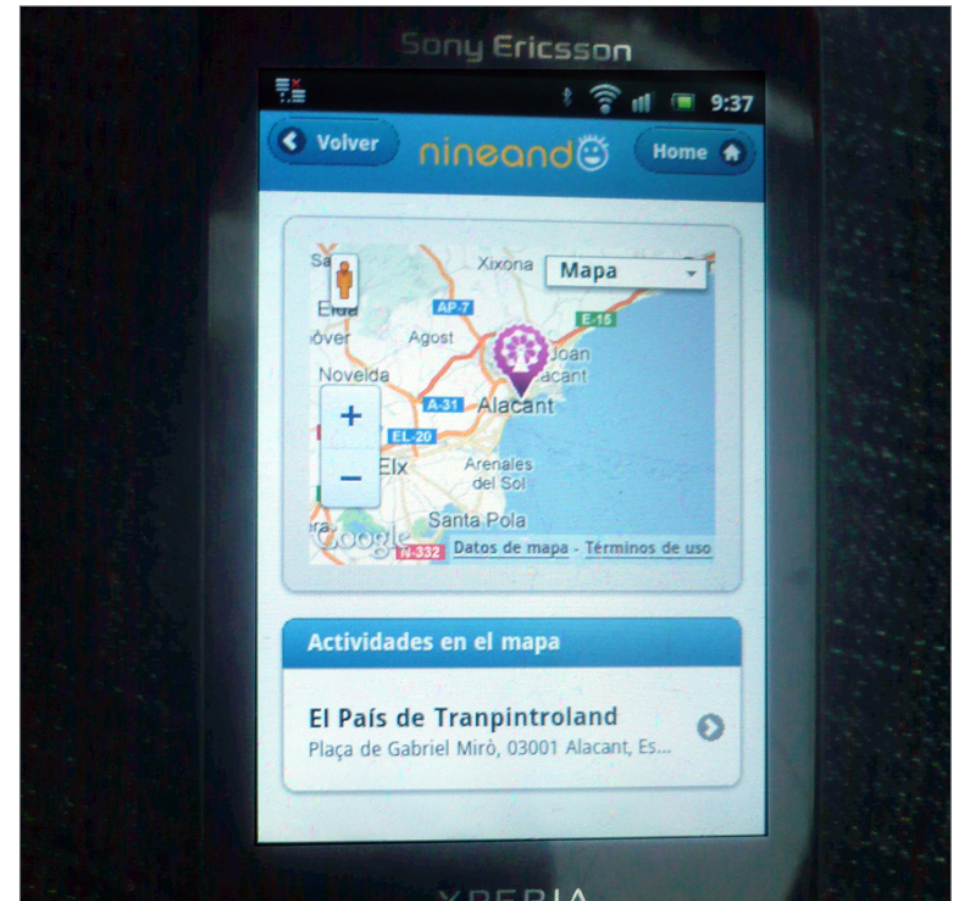


Figura 7.23 Aplicació en dispositiu Android

# 8 Manual d'ús

---

Per l'elaboració del manual d'us, s'ha triat fer-ho de la manera que s'ha cregut més entenedora així com més visual, per aquest motiu s'ha realitzat en format vídeo. Per la confecció del vídeo s'ha utilitzat el software **Camtasia Studio**.

Un altre aspecte que es vol deixar present, és la narració del mateix. En un primer moment s'ha optat per incloure la pròpia veu de l'estudiant, però la qualitat obtinguda amb els mitjans disponibles era molt deficient, en aquest sentit s'ha decidit fer servir una veu sintetitzada per narrar el vídeo. Per disposar d'aquesta veu s'ha fet mitjançant el software **ivona**.

S'ha optat per penjar el vídeo a **Vimeo**, per una qüestió de millora d'accessibilitat així com per facilitar la descàrrega d'entregables del PFC. Actualment està disponible a la següent adreça: <http://vimeo.com/54351599>. S'aconsella per la seva visualització activar el format HD, disponible en la barra de control del player.

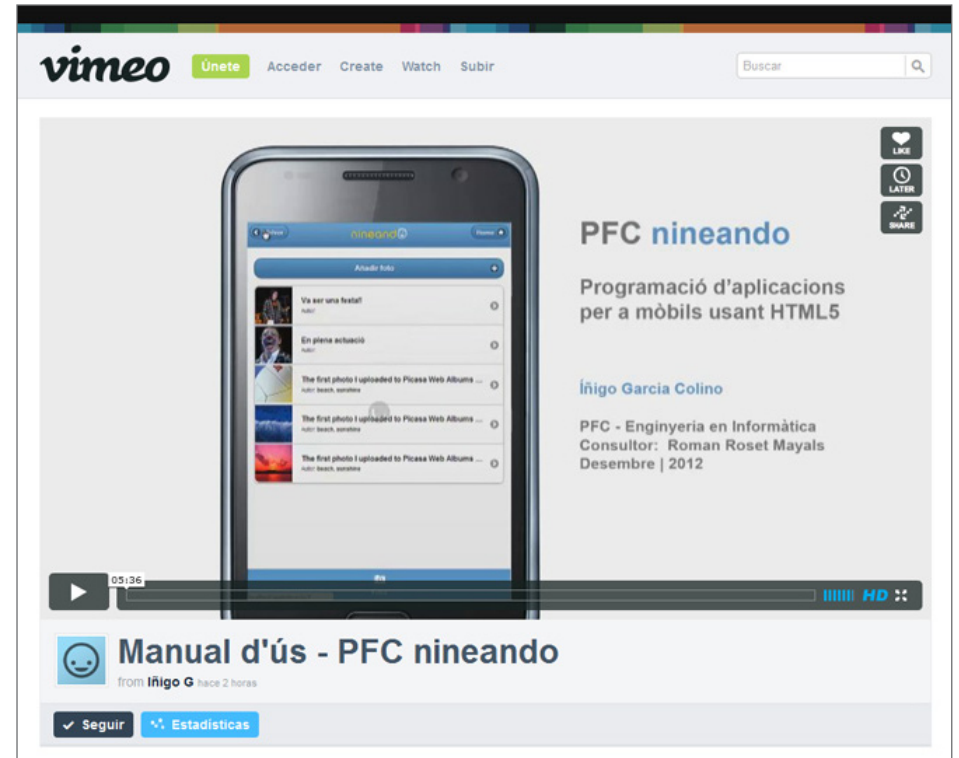


Figura 8.1 Captura de pantalla Vimeo

# 9 Conclusions

---

Després de la realització del projecte, s'ha arribat a diverses conclusions en temes relacionats amb el desenvolupament d'aplicacions per a dispositius mòbils i amb les diferents tecnologies emprades en el PFC.

## Aplicacions realitzades amb tecnologies web

Partint de la base que era la primera aplicació per a mòbils que es realitzava, i tenint en compte que aquest desenvolupament s'havia de realitzar amb tecnologies web (HTML5, CSS3 i JS), en un principi es desconeixia si el resultat seria satisfactori degut a que els diferents dispositius disposen de tecnologies pròpies per la realització d'aplicacions natives depenent del seu SO.

La conclusió que s'ha extret és que realment és viable realitzar aplicacions amb tecnologies web i que a més a més es tracta d'una solució avantatjosa per diversos motius. En primer lloc s'aconsegueix una aplicació multi-plataforma, vàlida per la majoria de dispositius, i en segon lloc, personalment es considera una tecnologia més senzilla que la necessària per desenvolupar aplicacions natives, tenint en compte la varietat de SOs presents en el mercat.

## Serveis web

Un altre dels reptes era poder fer ús de diferents serveis web disponibles gratuïtament a Internet i que ofereixen a l'aplicació una sèrie de recursos que la fan més completa i més atractiva per l'usuari final.

Alguns d'aquests serveis s'haguessin pogut implementar amb certes limitacions, però mai el resultat hagués sigut el mateix, com per exemple la gestió de comentaris, però d'altres, com Google Maps, són indispensables per l'aplicació.

Les avantatges són clares, a banda del que és més evident, fer ús dels serveis, s'ha de tenir en compte que les rutines dels diferents serveis s'actualitzen de forma transparent per al programador i que la càrrega que suposa l'execució de cada rutina desapareix i es reparteix sobre els servidors que ofereixen el servei web.

En definitiva, s'ha quedat força satisfet pel resultat obtingut, i es considera que el projecte podria posar-se en fase d'exploració, ja que donaria un servei d'interès i utilitat per als usuaris interessats.

# 10 Entregables PFC

---

La entrega final del PFC consisteix en un fitxer comprimit zip, **pfc\_igarcia.zip**, on hi ha el següent contingut:

## **memoria\_pfc\_igarcia.pdf**

Memòria del projecte.

## **presentacio\_pfc\_igarcia.html**

Presentació del projecte en vídeo. És necessari tenir accés a Internet per reproduir el vídeo.

## **codi\_pfc\_igarcia**

Directorio on es pot accedir al codi font del projecte. Dintre d'aquest directorio, trobem 2 directoris més un fitxer amb extensió **.apk**:

- **appkidevents**: Directorio amb la part de l'aplicació mòbil. Per executar-la s'ha d'accedir al fitxer **index.html** des d'un navegador. És necessari que el navegador triat usi el motor webkit: Chrome, Safari...
- **kidevents**: Directorio amb la part dels serveis creats en PHP que serveixen les dades a l'aplicació mòbil. Aquesta part està allotjada en un servidor propi, per tant no és necessari allotjar-la en local, de fet l'aplicació mòbil apunta al servidor extern. S'inclou el codi degut a que forma part del projecte. En el subdirectorio **docs**, es troba la base de dades feta servir en l'aplicació.
- **nineando-debug.apk**: Paquet per al sistema operatiu Android.

# ANNEX I - Glosari

---

**AJAX:** Acrònim de *Asynchronous Javascript And XML*, és una tècnica de desenvolupament web per crear aplicacions interactives. Aquestes aplicacions s'executen en el client mentre es manté la comunicació asíncrona amb el servidor en segon pla.

**API:** De l'anglès *Application Programming Interface*, és el conjunt de funcions i procediments que ofereix certa biblioteca per ser utilitzat per un altre programari com una capa d'abstracció.

**CSS:** De l'anglès *Cascading Style Sheets*, és un llenguatge usat per definir la presentació d'un document estructurat escrit en HTML.

**DIV:** Etiqueta HTML que s'empra per definir un bloc de contingut o secció de la pàgina, per poder aplicar-li diferents estils i fins i tot per realitzar operacions sobre aquest bloc específic.

**Framework:** La paraula anglesa *framework* (marc de treball) defineix, en termes generals, un conjunt estandarditzat de conceptes, pràctiques i criteris per enfocar un tipus de problemàtica particular que serveix com a referència, per enfrontar i resoldre nous problemes d'índole similar.

**GET:** Mètode d'enviament de les dades inserides en un formulari. Envia les dades en una cadena "visible". Convenient per enviar poques dades.

**HD:** Alta definició en anglès *High Definition*.

**HTML:** Sigles de *HyperText Markup Language*, fa referència al llenguatge de marcat predominant per a l'elaboració de pàgines web que s'utilitza per descriure i traduir l'estructura i la informació en forma de text.

**HTTP:** *Hypertext Transfer Protocol*, és el protocol usat en cada transacció de la *World Wide Web*. HTTP defineix la sintaxi i la semàntica que utilitzen els elements de programari de l'arquitectura web (clients, servidors, proxies) per comunicar-se.

**IDE:** Sigles en anglès de *Integrated Development Environment*. Un entorn de desenvolupament integrat, és un programa informàtic compost per un conjunt d'eines de programació. Pot dedicar-se en exclusiva a un sol llenguatge de programació o bé pot utilitzar-se per a varis.

**iOS:** Sistema operatiu mòbil d'Apple. Originalment desenvolupat per l'iPhone, sent després usat en dispositius com l'iPod, iPad i l'Apple TV.

**JS:** Un arxiu \*.js, és un arxiu de text pla que conté scripts de Javascript. És executat generalment per un navegador web.

**JSON:** Acrònim de *Javascript Object Notation*, és un format lleuger per a l'intercanvi de dades. JSON és un subconjunt de la notació literal d'objectes de Javascript que no requereix l'ús de XML.

**JSONP:** JSONP o "*JSON with padding*" és un complement per al format de base de dades de JSON. Proporciona un mètode per sol·licitar dades d'un servidor en un domini diferent, prohibit pels navegadors web habituals a causa de polítiques de seguretat.

**META:** Etiqueta HTML que s'incorpora en l'encapçalat d'una pàgina web i que resulta invisible per a un visitant normal, però de gran utilitat per a navegadors o altres programes que puguin valer-se d'aquesta informació.

**MVC:** Model Vista Controlador. És un patró o model d'abstracció de desenvolupament de programari que separa les dades d'una aplicació (model), la interfície d'usuari (vista), i la lògica de negoci (controlador) en tres components diferents.

**PHP:** PHP és un llenguatge de programació d'ús general de script del costat del servidor, originalment dissenyat per al desenvolupament web de contingut dinàmic.

**POST:** Mètode d'enviament de les dades inserides en un formulari. Envia les dades en forma "invisible". Convenient per enviar una gran quantitat de dades.

**SDK:** Sigles en anglès de *Software Development Kit*, és un conjunt d'eines de desenvolupament de programari que permet al programador crear aplicacions per a un sistema concret.

**SGBD:** Sistema de gestió de bases de dades, és un tipus de programari molt específic, dedicat a servir d'interfície entre la base de dades, l'usuari i les aplicacions que la utilitzen.

**SO:** Sigles de Sistema operatiu.

**SQL:** Llenguatge de consulta estructurat o SQL (per les seves sigles en anglès *Structured Query Language*) és un llenguatge declaratiu d'accés a bases de dades relacionals que permet especificar diversos tipus d'operacions en elles.

**URL:** Sigles en anglès de *Uniform Resource Locator*, és una seqüència de caràcters, d'acord a un format modèlic i estàndard, que s'usa per nomenar recursos en Internet per a la seva localització o identificació.

**XML:** Sigles en anglès de *Extensible Markup Language*, és un llenguatge de marques que permet definir la gramàtica de llenguatges específics per estructurar documents grans.

**Zend:** Principal patrocinador del projecte *Zend Framework*, framework de codi obert per desenvolupar aplicacions web i serveis web amb PHP.

# ANNEX II - Fonts consultades

---

*Android SDK | Android Developers* [en línia]  
< <http://developer.android.com/sdk/index.html> >

*Apache Cordova API Documentation* [en línia]  
< [http://docs.phonegap.com/en/2.2.0/guide\\_getting-started\\_android\\_index.md.html#Getting%20Started%20with%20Android](http://docs.phonegap.com/en/2.2.0/guide_getting-started_android_index.md.html#Getting%20Started%20with%20Android) >

*API - DISQUS* [en línia]  
< <http://disqus.com/api/docs/> >

*Developer's Guide: PHP - Picasa Web Albums Data API — Google Developers*  
[en línia]  
< [https://developers.google.com/picasa-web/docs/1.0/developers\\_guide\\_php](https://developers.google.com/picasa-web/docs/1.0/developers_guide_php) >

FIRTMAN, Maximiliano. "jQuery Mobile: Up and Running". *O'Reilly Media* (2012).

*jQuery API* [en línia]  
< <http://api.jquery.com/> >

*jQuery Mobile: Demos and Documentation* [en línia]  
< <http://jquerymobile.com/demos/1.2.0/> >

*jQuery Mobile Docs - Phonegap* [en línia]  
< <http://jquerymobile.com/test/docs/pages/phonegap.html> >

*Recently Active 'jquery-mobile' Questions - Stack Overflow* [en línia]  
< <http://stackoverflow.com/questions/tagged/jquery-mobile> >

ROSET, Roman. "Els enfocaments per escollir un framework".

ROSET, Roman. "Introducció a les aplicacions per a telefonia mòbil".

*Versión 3 de Google Maps JavaScript API - Google Maps API — Google Developers* [en línia]  
< <https://developers.google.com/maps/documentation/javascript/?hl=es> >

*Wikipedia, la enciclopedia libre* [en línia]  
< <http://es.wikipedia.org/wiki/Wikipedia:Portada> >

*Zend Framework* [en línia]  
< <http://framework.zend.com/manual/1.12/en/manual.html> >



# ANNEX III - Índex de figures

---

Figura 1.1 Logotip Aplicació “nineando” .....	5	Figura 5.7 Pantalles “Com arribar” .....	25
Figura 1.2 Panell Kanban facilitat per trello.com .....	7	Figura 5.8 Diagrama flux de pantalles .....	26
Figura 1.3 Diagrama de Gantt de la planificació i tasques del projecte .....	9	Figura 5.9 Diagrama model de dades .....	27
Figura 2.1 Detall Aplicació “Guia de Fiestas Infantiles” .....	10	Figura 5.10 Icones identificatives de les categories .....	28
Figura 2.2 Detall Aplicació “Cult App” .....	11	Figura 6.1 Esquema arquitectura física .....	29
Figura 3.1 Detall Netbeans IDE 7.0 .....	12	Figura 6.2 Esquema arquitectura lògica .....	30
Figura 3.2 Detall WAMP 5 .....	12	Figura 6.3 Esquema arquitectura lògica. Exemple .....	31
Figura 3.3 Detall TortoiseSVN .....	13	Figura 7.1 Captura de pantalla Cercar activitats per localització .....	38
Figura 4.1 Detall Interfície d’usuari jQuery Mobile .....	15	Figura 7.2 Captura de pantalla Cercar activitats per províncies ..	39
Figura 4.2 Detall Picasa .....	16	Figura 7.3 Captura de pantalla Cercar activitats per províncies ..	40
Figura 4.3 Detall panell administració Disqus .....	17	Figura 7.4 Captura de pantalla Veure fitxa activitat .....	42
Figura 4.4 Detall Google Maps .....	17	Figura 7.5 Captura de pantalla Veure / Afegir comentaris .....	43
Figura 5.1 Pantalles “Cercar activitats per localització” .....	19	Figura 7.6 Captura de pantalla Veure imatges .....	45
Figura 5.2 Pantalles “Cercar activitats per províncies” .....	20	Figura 7.7 Captura de pantalla Veure imatges .....	46
Figura 5.3 Pantalles “Veure fitxa activitat” .....	21	Figura 7.8 Captura de pantalla Com arribar .....	48
Figura 5.4 Pantalles “Veure / Afegir comentaris” .....	22	Figura 7.9 Proves resultats Cercar activitats per localització .....	49
Figura 5.5 Pantalles “Veure imatges” .....	23	Figura 7.10 Proves query Cercar activitats per localització .....	50
Figura 5.6 Pantalles “Afegir imatges” .....	24	Figura 7.11 Proves llistat Cercar activitats per províncies .....	51
		Figura 7.12 Proves mapa Cercar activitats per províncies .....	51

---

## ANNEX III - Índex de figures

---

Figura 7.13 Proves fitxa Veure fitxa activitat .....	52
Figura 7.14 Proves query Veure fitxa activitat .....	52
Figura 7.15 Proves Veure / Afegir comentaris .....	53
Figura 7.16 Proves llistat Veure imatges .....	53
Figura 7.17 Proves Picasa Veure imatges .....	54
Figura 7.18 Proves Afegir imatges .....	54
Figura 7.19 Proves Com arribar .....	55
Figura 7.20 Creació projecte Android .....	56
Figura 7.21 Directoris projecte Android .....	56
Figura 7.22 Selecció dispositiu Android .....	58
Figura 7.23 Aplicació en dispositiu Android .....	58
Figura 8.1 Captura de pantalla Vimeo .....	59