

## **RADIUS AUGMENTED REALITY**

Desenvolupament d'una aplicació de realitat augmentada per a iPhone  
per visualitzar els llocs d'interès propers a l'usuari en entorns urbans controlats

**David Arcos Gutiérrez**

Postgrau en desenvolupament d'aplicaciones per a dispositius mòbils

Carlos Caballero González

22 de juny de 2015



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

**Títol del treball:**

Radius Augmented Reality. Desenvolupament d'una aplicació de realitat augmentada per a iPhone per visualitzar els llocs d'interès propers a l'usuari en entorns urbans controlats

**Nom de l'autor:**

David Arcos Gutierrez

**Nom del consultor:**

Carlos Caballero González

**Data d'entrega:**

22 de juny de 2015

**Àrea del treball final:**

Informàtica, telecomunicació i multimèdia

**Titulació:**

Postgrau en desenvolupament d'aplicacions per a dispositius mòbils

**Paraules clau:**

Realitat, Augmentada, iPhone, iOS, Geolocalització, Orientació.

**Resum:**

Aquest projecte consisteix en la planificació i el desenvolupament d'una aplicació de realitat augmentada per a iPhone amb finalitats turístiques: *Radius AR*. L'aplicació, que pretén ser un dels mòduls d'una xarxa social que es desenvolupa a l'empresa on treballa l'autor del projecte, permet visualitzar llocs d'interès situats al voltant de l'usuari (fins a una distància de 10 km). Per a cada lloc, es mostra una etiqueta sobre la imatge de la càmera del telèfon en temps real. En tocar l'etiqueta, apareix una pantalla amb informació sobre el lloc (les coordenades del lloc i la distància respecte de l'usuari). Des d'aquesta pantalla, es pot seleccionar el lloc en qüestió com a objectiu i, en tornar a la vista de la càmera, només es conserva aquesta etiqueta. Un cop es tornen a carregar les dades, apareixen de nou tots els llocs.

D'altra banda, des de la part superior de la vista principal, es pot accedir a un menú de configuració de filtres, que permet seleccionar un lloc segons la seva naturalesa (cultura, parcs, telecomunicacions...) i la distància a què es troben. Es poden seleccionar fins a sis filtres simultàniament. D'aquesta manera, la informació que es mostra per pantalla s'adiu millor amb els interessos de l'usuari.

**Abstract:**

This project consists on planning and developing an augmented reality app focused on tourism for iPhone. This app, which is intended to be part of a social network that is being developed by the company where the project's author works, lets the user see the tourist attractions located around him (until 10 km). For each location, a label appears on the image provided by

the smartphone's camera in real time. When tapping on the tag, a new view with information about the place appears. From this view, the option "Show this site only" may be selected in order to hide all the tags but this one (this touristic attraction is considered the user's objective). After reloading the data, all the labels are shown again.

In addition, the filters menu can be accessed from the top of the screen. Using this module's selector, the attractions can be filtered according to its nature (culture, park, telecommunications...) and its distance from the user. A maximum of six filters can be chosen simultaneously. Thanks to this functionality, the on-screen information better fits the user's interests.

# ÍNDEX

<b>Índex de figures</b>	<b>6</b>
<b>Índex de taules</b>	<b>7</b>
<b>1 Introducció</b>	<b>9</b>
1.1 Context i justificació . . . . .	9
1.2 Motivació . . . . .	10
1.3 Objectius del treball . . . . .	10
1.4 Enfocament i metodologies . . . . .	11
1.5 Planificació del treball . . . . .	11
1.6 Entregables . . . . .	13
1.7 Continguts de la memòria . . . . .	13
<b>2 Anàlisi</b>	<b>15</b>
2.1 Anàlisi de l'audiència . . . . .	15
2.1.1 Públic objectiu: Abigail Steward . . . . .	15
2.1.2 Públic objectiu: Ben Müller . . . . .	16
2.2 Anàlisi del mercat . . . . .	17
2.2.1 Mercat principal . . . . .	17
2.2.2 Altres mercats . . . . .	20
<b>3 Objectius i planificació</b>	<b>22</b>
3.1 Descripció de l'aplicació . . . . .	22
3.2 Funcionalitats de l'aplicació . . . . .	23
3.3 Objectius . . . . .	24
3.4 Llista de tasques . . . . .	25
3.5 Planificació general . . . . .	28

<b>4</b>	<b>Disseny</b>	<b>30</b>
4.1	Disseny conceptual i arquitectura de l'aplicació . . . . .	30
4.1.1	L'elecció del nom . . . . .	30
4.1.2	Diagrama de casos . . . . .	31
4.1.3	Fluxos d'usuari . . . . .	33
4.2	Prototip de baixa resolució i <i>wireframes</i> . . . . .	38
4.3	Prototip d'alta resolució i <i>wireflow</i> . . . . .	40
4.4	Icona de l'aplicació . . . . .	44
<b>5</b>	<b>Desenvolupament</b>	<b>45</b>
5.1	<i>Hardware</i> i <i>software</i> utilitzat . . . . .	45
5.2	Models i diagrames dels objectes . . . . .	45
5.2.1	Classes dels objectes de <i>back-end</i> . . . . .	46
5.2.2	Classes dels objectes de <i>front-end</i> . . . . .	49
5.2.3	Controladors de vista . . . . .	52
5.2.4	Diagrama de classes . . . . .	55
5.2.5	Model relacional de la base de dades . . . . .	56
5.3	Algorismes destacats . . . . .	57
5.3.1	Mètode per obtenir les jerarquies . . . . .	57
5.3.2	Mètodes de projecció per la realitat augmentada . . . . .	60
5.3.3	Filtres per a les dades dels sensors . . . . .	63
<b>6</b>	<b>Radius Augmented Reality</b>	<b>64</b>
<b>7</b>	<b>Síntesi i conclusions</b>	<b>72</b>
7.1	Conclusions . . . . .	72
7.2	Anàlisi crítica . . . . .	73
7.3	Continuació . . . . .	73

# ÍNDEX DE FIGURES

1.1	Diagrama de Gantt general . . . . .	12
3.1	Diagrama PERT del projecte complet. . . . .	28
3.2	Diagrama de Gantt de l'etapa de disseny. . . . .	28
3.3	Diagrama de Gantt de l'etapa de desenvolupament. . . . .	29
4.1	Esquema conceptual de l'aplicació <i>Radius AR</i> . . . . .	31
4.2	Diagrama de casos de l'aplicació <i>Radius AR</i> . . . . .	32
4.3	Esquema de comunicació de l'aplicació <i>Radius AR</i> . . . . .	33
4.4	Flux dels permisos de l'aplicació . . . . .	34
4.5	Flux bàsic de l'aplicació . . . . .	35
4.6	Flux de filtratge . . . . .	36
4.7	Flux del botó “On sóc?” (extra) . . . . .	37
4.8	Flux de la vista de configuració (extra) . . . . .	37
4.9	Prototip en baixa resolució de les vistes de principals . . . . .	38
4.10	Prototip en baixa resolució de les vistes de configuració . . . . .	39
4.11	Prototip en alta resolució 1. Vista de realitat augmentada i menú de filtres . . . . .	40
4.12	Prototip en alta resolució 2. Vistes detall dels llocs . . . . .	41
4.13	Prototip en alta resolució 3. Configuració i vista “On sóc” . . . . .	42
4.14	Prototip en alta resolució 4. Informació de la vista “On sóc” . . . . .	43
4.15	Propostes d'icones . . . . .	44
5.1	Model del <i>tag</i> del servidor . . . . .	46
5.2	Model del <i>Handler</i> del <i>tag</i> del servidor . . . . .	47
5.3	Model del <i>location</i> del servidor . . . . .	48
5.4	Model del <i>Handler</i> del <i>location</i> del servidor . . . . .	49
5.5	Model del <i>tag</i> . . . . .	49
5.6	Model del lloc . . . . .	50
5.7	Model de l'objecte que es comunica amb el servidor . . . . .	51

5.8	Model de la brúixola . . . . .	52
5.9	Model de l'objecte que realitza els càlculs geomètrics . . . . .	52
5.10	Controlador de la vista principal . . . . .	53
5.11	Controlador del panell de filtres . . . . .	54
5.12	Controlador de la vista detall . . . . .	55
5.13	Controlador de la vista de configuració . . . . .	55
5.14	Diagrama de classes . . . . .	56
5.15	Model relacional de la base de dades . . . . .	56
5.16	Llista de <i>tags</i> provisional . . . . .	57
5.17	Exemples de jerarquies . . . . .	58
5.18	Exemple de jerarquia de dos nivells . . . . .	58
5.19	Algorisme per obtenir les jerarquies . . . . .	59
5.20	Exemple de cicle en la jerarquia de <i>tags</i> . . . . .	60
5.21	Factor de conversió per a l'aproximació plana de la latitud . . . . .	60
5.22	Factor de conversió per a l'aproximació plana de la longitud . . . . .	61
5.23	Angle relatiu de la direcció horitzontal a la qual es troba el lloc . . . . .	61
5.24	Projecció horitzontal de la posició del lloc . . . . .	62
5.25	Obtenció de l'angle d'inclinació a partir del vector gravetat . . . . .	62
5.26	Angle relatiu de la direcció vertical a la qual es troba el lloc . . . . .	62
5.27	Projecció vertical de la posició del lloc . . . . .	63
5.28	Projecció de les direccions a la pantalla . . . . .	63
5.29	Filtre per a les dades de l'acceleròmetre . . . . .	63
6.1	Captures. Posicionament correcte a diferents distàncies . . . . .	64
6.2	Captures. Plaça d'Espanya des de diferents angles (1) . . . . .	65
6.3	Captures. Plaça d'Espanya des de diferents angles (2) . . . . .	65
6.4	Captures. Resultat d'aplicar el filtre <i>telecommunication</i> (1) . . . . .	66
6.5	Captures. Resultat d'aplicar el filtre <i>telecommunication</i> (2) . . . . .	66
6.6	Captures. Resultat de realitzar un canvi de radi (filtres) . . . . .	67
6.7	Captures. Reposicionament de les etiquetes en temps real . . . . .	67
6.8	Captures. Resultat d'aplicar el filtre <i>culture</i> . . . . .	68
6.9	Captures. Accés a la vista detall del MNAC . . . . .	68
6.10	Captures. Llocs a llargues distàncies . . . . .	69
6.11	Captures. Efecte de canviar la inclinació del terminal . . . . .	69
6.12	Altres captures (1) . . . . .	70
6.13	Altres captures (2) . . . . .	70
6.14	Consola per crear els continguts . . . . .	71



# ÍNDEX DE TAULES

1.1	Taula resum de la planificació. . . . .	11
2.1	Taula comparativa d'algunes aplicacions de l'App Store. . . . .	19

# CAPÍTOL

## 1

# INTRODUCCIÓ

## 1.1 Context i justificació

Els dispositius mòbils d'avui en dia presenten unes especificacions tècniques increïbles. Els *smartphones* incorporen cada vegada més sensors i aquests són, al seu torn, cada vegada més precisos. Aquesta explosió tecnològica dels dispositius portables, juntament amb les capacitats creixents de les xarxes de telecomunicació i geolocalització, ha fet possible que la realitat augmentada hagi traspassat les fronteres de la ciència-ficció [9, 10, 21].

D'altra banda, la presència del nou model de software, les aplicacions mòbils, ha canviat la manera amb què els usuaris descobreixen, interactuen i aprenen a utilitzar el nou programari i, com a conseqüència, la manera amb què s'ha de desenvolupar aquest programari. Les aplicacions han de ser simples, atractives, útils, interactives i, si és possible, eficients. Així, l'experiència d'usuari i el disseny de les aplicacions són més importants que mai i la tasca dels desenvolupadors s'ha d'adaptar a aquest nou context.

Una tecnologia d'avantguarda com aquesta atreu els usuaris i els facilita l'accés a la informació que desitgen que és, en última instància, l'objectiu d'aquest projecte. A més, tot i que al mercat hi ha un munt d'aplicacions d'aquest tipus que ofereixen solucions interessants (vegeu l'apartat 2.2), com que es tracta d'una branca relativament nova, hi ha molts aspectes a millorar (tant pel que fa als continguts com pel que fa a la tecnologia): qualitat dels sensors, processament de les dades dels sensors, mètodes d'interacció amb els objectes virtuals, representació dels objectes virtuals sobre la realitat, exhaustivitat i vigència de les bases de dades, classificació de la informació, etc.

En aquest treball es busca un compromís entre representació fidedigna dels objectes i senzillesa en el mode d'interacció. Per això, els valors afegits que es persegueixen són un bon sistema (simple i escalable) per filtrar les dades i posicionar les etiquetes (accessos directes a la informació) dels llocs d'interès propers a l'espai 3D que percep l'usuari de manera natural.

## 1.2 Motivació

La motivació per realitzar un projecte tan ambiciós com aquest està recolzada en tres pilars fonamentals: el seu potencial, el repte personal que suposa i l'aportació a la feina actual.

Com s'ha comentat en l'apartat anterior, la realitat augmentada és una tecnologia d'avantguarda que promet convertir-se en un dels actors del futur mòbil. Per exemple, aplicacions com *Aurasma* [9, 10] demostren que la realitat augmentada pot estar present en àmbits ben diferents i innovadors. Aquest potencial, juntament amb l'interès personal pel tema, ha estat el motiu primordial per escollir fer una aplicació relacionada amb la realitat augmentada.

Un altre aspecte molt interessant de les aplicacions de realitat augmentada és que fan un ús exhaustiu dels sensors del mòbil (repte en l'àmbit tecnològic). A més, en aquestes aplicacions, el posicionament dels objectes que apareixen per pantalla és un problema purament geomètric (repte en l'àmbit matemàtic) [19]. Aquests dos reptes, a més d'oferir una dosi extra de motivació, coincideixen amb interessos i aficions personals.

Finalment, el motiu que ha acabat de concretar l'abast i el tipus de servei que ha d'oferir l'aplicació ha estat el benefici que pot suposar des del punt de vista laboral. L'aplicació web que actualment es troba en producció a l'empresa vol oferir nous serveis i funcionalitats als clients en les futures versions mòbils de l'aplicació. Per tant, aquesta aplicació pretén ser, de fet, una versió prèvia d'una de les funcionalitats extres de l'aplicació nativa en iOS.

## 1.3 Objectius del treball

L'objectiu d'aquest projecte és desenvolupar una aplicació nativa (exclusiva per a iOS [18]) senzilla i intuïtiva per visualitzar la informació dels llocs d'interès propers a l'usuari. Aquesta informació s'ha de mostrar sobre la imatge capturada per la càmera de l'iPhone en temps real. Per aconseguir aquest objectiu, cal desenvolupar dos tipus de software: una base de dades (servidor) i una aplicació mòbil per tractar la informació dels sensors i mostrar les dades (client).

L'aplicació del client (*front-end*) ha de comunicar-se amb el servidor per obtenir la informació dels llocs propers a la posició actual de l'usuari. Mitjançant els sensors d'orientació i la càmera del mòbil s'han de mostrar, com a mínim, unes etiquetes amb els noms dels llocs que estan visibles a la pantalla del mòbil, ja que, si s'aconsegueix això, incloure més informació hauria de ser una tasca trivial. A més, l'usuari ha de poder filtrar la informació mostrada mitjançant, d'una banda, un sistema de categories i, de l'altra, la distància radial a la seva posició actual. És important destacar que l'usuari no introdueix les dades al sistema, només les llegeix; per tant,

a banda de la tasca de programació, cal recollir les dades de les localitzacions i introduir-les a la base de dades. Finalment, tot i que l'objectiu principal d'aquest projecte és desenvolupar la part del client —és a dir, l'aplicació per a iOS—, també es pretén programar una base de dades de prova en PHP [13], senzilla i fàcilment escalable, per guardar les coordenades i la informació dels llocs d'interès (*back-end*).

## 1.4 Enfocament i metodologies

Com que la branca de realitat augmentada és relativament nova, la metodologia de recerca que s'ha utilitzat per al desenvolupament de l'aplicació és la de disseny i creació (*design and creation research*). Per la part de servidor, en canvi, com que es tracta d'una base de dades normal i corrent, s'ha utilitzat el framework *Symfony*, el qual presenta un conjunt d'eines i models que permeten implementar una base de dades robusta en relativament poc temps.

Pel que fa al procés de desenvolupament de l'aplicació, s'ha seguit una estratègia modular que consisteix a generar múltiples projectes petits que intenten donar solució a problemes parcials. Més concretament, s'ha realitzat la següent bateria de projectes individuals en Xcode:

- Projecte per testejar la comunicació amb el servidor.
- Implementació d'una aplicació brúixola per obtenir i testejar les dades de tots els sensors.
- Projecte per realitzar proves de disseny per a la interfície.
- Projecte amb els algorismes per realitzar els càlculs de realitat augmentada.
- Projecte de proves amb la càmera i les capes superposades.

## 1.5 Planificació del treball

Aquest projecte de desenvolupament té una duració aproximada de 4 mesos (incloent-hi totes les etapes del procés de desenvolupament). A més del document final i l'aplicació, consta de tres entregues parcials relacionades amb la conceptualització, la planificació i el disseny de l'aplicació.

PEC	Inici	Final	Temps aprox.	Valor
Proposta	25/02/2015	08/03/2015	2 setmanes	5 %
Anàlisi i planificació	09/03/2015	29/03/2015	3 setmanes	25 %
Disseny	30/03/2015	26/04/2015	4 setmanes	30 %
Desenvolupament	27/04/2015	21/06/2015	8 setmanes	40 %

Taula 1.1: Taula resum de la planificació.

Tot i que la planificació general i les diferents tasques es detallen al capítol 3, en aquest apartat s'ha inclòs el diagrama de Gantt complet que s'ha seguit durant el curs. A la figura 1.1 es mostra el diagrama complet amb les subtasques corresponents a cada etapa.

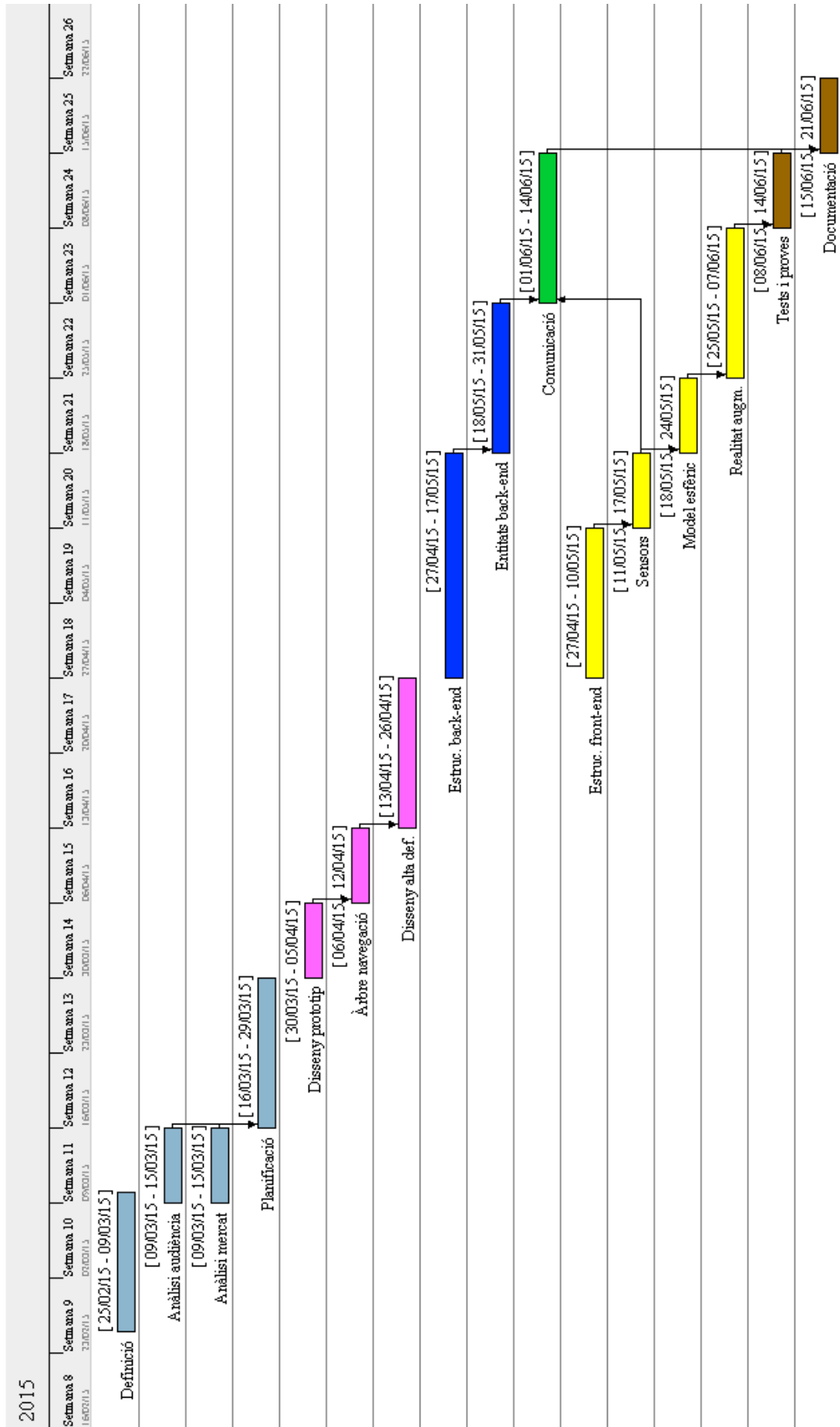


Figura 1.1: Diagrama de Gantt general

## 1.6 Entregables

Com que l'objectiu d'aquest treball és desenvolupar un software, i.e. la plataforma per l'aplicació de realitat augmentada, a més d'aquesta memòria, s'entregaran el conjunt de fitxers i projectes amb el codi d'implementació. A continuació, s'inclou una llista amb tots els entregables d'aquest projecte:

- Projecte Xcode amb el codi de l'aplicació del client.
- Mòduls *Symfony* dels objectes del servidor.
- Consola d'administració web per crear elements a la base de dades.
- Fitxer ipa de l'aplicació del client.
- Memòria del projecte.
- Projecte *Axure* amb el disseny en alta resolució de la interfície.
- Conjunt de vídeos que demostren el funcionament de l'aplicació.

## 1.7 Continguts de la memòria

La memòria d'aquest projecte s'estructura en set capítols. Cadascun dels cinc capítols centrals (del 2 al 6) pretén reflectir una de les etapes de creació de l'aplicació, mentre que el primer i l'últim capítol constitueixen el marc que cohesiona i dona forma a la memòria.

En el primer capítol, es descriuen els motius de l'elecció d'aquest tema tant des d'un punt de vista social com personal (apartats 1.1 i 1.2). A més, es comenten les metodologies emprades (apartat 1.4) per a la consecució dels objectius marcats (apartat 1.3). D'altra banda, també s'inclou la planificació del treball durant l'assignatura (apartat 1.5) i una breu introducció a la planificació del procés de desenvolupament de l'aplicació (aquesta secció s'aprofundeix en el capítol 3). Per acabar, aquest capítol també recull la llista de materials que s'entreguen com a part d'aquest projecte (apartat 1.6).

El contingut del segon capítol recull el primer pas a l'hora de desenvolupar una aplicació: l'anàlisi de l'audiència i del mercat. Per a l'anàlisi de l'audiència (apartat 2.1), s'ha utilitzat el model Persones, en què l'Abigail i en Ben representen el públic objectiu de l'aplicació que es pretén desenvolupar. Pel que fa a l'anàlisi del mercat (apartat 2.2), s'han buscat aplicacions semblants a l'App Store, se n'ha fet una breu descripció i s'ha generat una taula comparativa amb les dades més rellevants. Tot i que l'anàlisi més aprofundida s'ha fet en aquesta botiga (ja que l'aplicació que es desenvolupa és només per a iPhone), també s'han fet cerques semblants a mercats d'aplicacions d'altres sistemes operatius per tal de tenir una visió més completa de la situació real del mercat.

Tenint presents els objectius i les anàlisis de públic i mercat, es pot començar a concebre l'aplicació i, a continuació, planificar el procés de desenvolupament. Tot això es recull en el capítol 3, que té dos grans blocs. El primer bloc recull una descripció de l'aplicació (apartat 3.1) i una llista de les funcionalitats de què ha de disposar (apartat 3.2). El segon bloc constitueix pròpiament la planificació del desenvolupament de l'aplicació. D'una banda, es comenten els objectius que cal assolir perquè l'aplicació compleixi les característiques descrites en el bloc anterior (apartat 3.3) i, de l'altra, a partir d'aquestes funcions, es detallen totes les tasques que caldrà fer durant el desenvolupament (apartat 3.4). Per tancar aquest segon bloc, es fa la planificació temporal del procés utilitzant diagrames de PERT i de Gantt (apartat 3.5).

En el quart capítol, es reflecteix el procés de disseny de l'aplicació. El primer aspecte que es tracta és el disseny conceptual (apartat 4.1), que inclou tant l'elecció del nom com el diagrama de casos i els fluxos d'usuari. Un cop es té clar què ha de poder fer l'usuari, s'elaboren els primers prototips de l'aplicació: primer, de baixa resolució (apartat 4.2) i, després, d'alta resolució (apartat 4.3). Finalment, es comenta el disseny de la icona (apartat 4.4).

El capítol 5 recull tota la informació relacionada amb la programació de l'aplicació. És, doncs, el capítol més tècnic. En primer lloc, es descriuen breument el *hardware* i el *software* utilitzats (apartat 5.1). En segon lloc, es comenten les classes de tots els objectes involucrats (apartat 5.2). Per acabar, tanca aquest capítol la secció en què es comenten els algorismes més significatius de l'aplicació (apartat 5.3).

El sisè capítol mostra el fruit del treball descrit en els capítols 2 a 5, ja que inclou captures de pantalla de diverses proves de l'aplicació dutes a terme a Barcelona. A més a més, en aquest capítol també s'inclou una imatge de l'aspecte que presenta la consola que s'ha programat per introduir nous llocs d'interès a la base de dades (aquesta era una tasca que no formava part dels objectius del treball).

Finalment, el setè capítol conclou el projecte. Aquest capítol breu inclou les conclusions (apartat 7.1), una anàlisi crítica de l'aplicació obtinguda (apartat 7.2) i les possibles línies de treball per continuar aquest projecte (apartat 7.3).

# CAPÍTOL

## 2

# ANÀLISI

## 2.1 Anàlisi de l'audiència

Aquesta aplicació està destinada a convertir-se en una funcionalitat d'una xarxa social oberta a tot el públic. Per tant, no es pot pressuposar que l'usuari final de l'aplicació tingui cap tipus de coneixement informàtic ni destresa especial en l'ús d'aplicacions. De fet, un dels aspectes més importants d'aquesta aplicació i que cal treballar amb més insistència és la simplicitat. L'usuari ha de trobar una interfície intuïtiva i una estructura de navegació coherent que li permeti entendre ràpidament el funcionament de l'aplicació. Però, abans de parlar pròpiament de les funcionalitats i de l'estructura de l'aplicació, cal donar respostes coherents a les preguntes següents: qui és l'usuari típic?, per què utilitza l'aplicació?, quins són els seus interessos?, quant de temps dedicarà a l'aplicació? Per respondre aquestes preguntes s'utilitzarà el mètode Persones [33] amb dos arquetips ben diferenciats: l'Abigail i en Ben. Aquests arquetips no es corresponen amb cap usuari real, però modelen adequadament el públic que potencialment utilitzarà aquesta funció de la xarxa social.

### 2.1.1 Públic objectiu: Abigail Steward

L'Abigail Steward és una estudiant jove de periodisme que actualment viu als Estats Units. És una noia activa, decidida i curiosa a qui li encanta viatjar i veure món. De fet, el seu somni és convertir-se en una reportera que viatgi per tot el món cobrint les notícies d'interès internacional. El seu mòbil és un iPhone 5 de color blanc, que utilitza habitualment. Ja sigui per fer fotos i compartir-les a la xarxa, ja sigui per orientar-se i consultar informació sobre els llocs que visita, el seu *smartphone* s'ha convertit en una eina indispensable en el seu dia a dia.



Disposa també de moltes aplicacions diferents —la majoria, gratuïtes— relacionades amb l'esport i l'entreteniment; tot i que, de fet, les que més utilitza són les de les xarxes socials i les pròpies del sistema operatiu. Tot i això, la seva aplicació preferida és *iTalk Recorder* [17], un eina per enregistrar veu que utilitza a vegades a la universitat (i que vol utilitzar intensivament quan treballi com a reportera).

### Abigail Steward

**Edat:** 21 anys.

**Professió:** estudiant (periodisme).

**Residència:** Boston, Massachusetts (EUA)

**Estat civil:** soltera.

**Nivell d'ingressos:** modestos (< 8.000 \$/any).

**Aficions/interessos:** viatjar, l'esport, l'art i la lectura.

**Terminal mòbil:** iPhone 5 (iOS 8.2).

**Hardware del mòbil que utilitza habitualment:** la càmera (fotos), el GPS (navegació) i el micròfon (gravar veu).

**Motiu pel qual utilitza iOS/Apple:** per la marca i l'estil.

**Motiu pel qual utilitzarà l'aplicació:** turisme i navegació.

**Xarxes socials:** Facebook, Twitter i Instagram.

El perfil de l'Abigail és un representant del conjunt de persones joves a qui els encanta viatjar i que, a més, han crescut amb les xarxes socials i l'accés ràpid a la informació. Les persones amb aquest perfil solen disposar de connexió a internet al mòbil de manera ininterrompuda, ja que l'utilitzen més per navegar que per parlar. Cal oferir una aplicació intuïtiva i vistosa, a més d'útil, per atraure aquest tipus de públic. Tot i no disposar de grans coneixements d'informàtica, volen aprendre ràpidament a utilitzar les aplicacions noves sense tutorials ni pantalles introductòries.

### 2.1.2 Públic objectiu: Ben Müller

El senyor Ben Müller és un empresari alemany que es dedica a la fabricació i a la distribució de motors de cotxe. És un home metòdic, rigurós i perfeccionista que ofereix un grau elevat de confiança i seguretat als seus clients. Està casat des de fa tretze anys i té un fill d'onze, del qual està molt orgullós. Li encanta anar al cinema i al teatre amb la família, tot i que no hi pot anar tan sovint com voldria. Fa molts anys que utilitza productes d'Apple per la marca i la seguretat i, a més, renova el seu *smartphone* quasi anualment. Actualment, el seu terminal mòbil és un iPhone 6 Plus de color gris amb 128 GB de memòria. Com que, per ell, el mòbil és una eina de treball indispensable, les aplicacions que més utilitza són les de gestió i productivitat. Tot i això, quan viatja per motius de feina, també utilitza *FaceTime* [4] per parlar amb la seva família i les aplicacions de navegació per orientar-se correctament a la ciutat que visita. Mai no utilitza aplicacions amb publicitat i no li importa pagar per les que creu que poden ser necessàries o útils.

## Ben Müller

**Edat:** 42 anys.

**Professió:** empresari (automoció).

**Residència:** Darmstadt, Alemanya.

**Estat civil:** casat (amb un fill).

**Nivell d'ingressos:** elevats (> 60.000 €/any).

**Aficions/interessos:** el cinema, el teatre, l'economia i el benestar.

**Terminal mòbil:** iPhone 6 Plus (iOS 8.2).

**Hardware del mòbil que utilitza habitualment:** la càmera (videoconferències) i el GPS (navegació).

**Motiu pel qual utilitza iOS/Apple:** per la seguretat i el *caché*.

**Motiu pel qual utilitzarà l'aplicació:** negocis i oci.

**Xarxes socials:** LinkedIn.

El perfil d'en Ben és un representant del conjunt de persones de mitjana edat que utilitzen el mòbil principalment com a eina de treball. Les persones amb aquest perfil sempre disposen de bona connexió a internet i les prestacions del seu terminal són molt superiors a les de la mitjana. Per la cultura de productivitat, aquest perfil no és tolerant ni a la publicitat invasiva ni a les esperes per carregar la informació. La prioritat per aquest tipus de persones és que l'aplicació sigui útil i ràpida.

## 2.2 Anàlisi del mercat

En l'àmbit del desenvolupament d'aplicacions mòbils, la millor manera de veure l'estat de l'art és buscar en els mercats d'aplicacions de la plataforma per a la qual es vol desenvolupar. Tot i això, també pot resultar molt útil veure solucions existents en altres plataformes per tal de detectar possibles vetes de mercat.

### 2.2.1 Mercat principal

Pel cas concret de l'aplicació que es vol desenvolupar en aquest treball, cal analitzar en primer lloc les aplicacions de navegació i realitat augmentada (Nav&AR) que estan disponibles a l'App Store, d'Apple. A continuació, se n'analitzen algunes:

#### VISION Augmented Reality (VAR) [16]

Tant el concepte com les funcionalitats d'aquesta aplicació són molt semblants als de l'aplicació que es pretén desenvolupar. Aquesta aplicació permet realitzar cerques de punts d'interès propers i mostrar-ne la informació de quatre maneres diferents: una llista ordenada per distància, el mapa tradicional amb marcadors, un conjunt de fletxes indicant les direccions lineals dels llocs respecte de la posició de l'usuari i una vista de realitat augmentada com la que es vol desenvolupar. A més, també es pot veure en detall la informació dels llocs i la valoració d'altres usuaris.

### **Junaio Navegador de Realidad Aumentada (JNRA) [22, 23, 24]**

Aquesta aplicació Nav&AR és molt completa i presenta, a més de la funció principal d'aquest projecte, un conjunt d'extres relacionats amb la realitat augmentada. Junaio és, de fet, una plataforma de l'empresa Metaio, que ofereix solucions de realitat augmentada per a empreses i particulars. Rep molt bones crítiques dels usuaris i, a més, aquesta aplicació està disponible tant per a iOS com per a Android (fins i tot, hi ha una versió beta per a *smart glasses*).

### **Augmented Car Finder (ACF) [8]**

Tot i que s'allunya una mica del concepte que es vol desenvolupar, l'aplicació ACF permet guardar la posició on s'ha aparcat el cotxe per, després, poder-lo trobar mitjançant una interfície de navegació amb realitat augmentada (tot i que, de fet, es pot utilitzar per guardar la posició de qualsevol objecte o ubicació).

### **Bus Turístic Virtual (BTV) [32, 31]**

És una aplicació Nav&AR per impulsar el turisme a Barcleona. Compleix les característiques principals de l'aplicació que es vol desenvolupar en aquest projecte, però presenta moltes limitacions tècniques i una interfície poc treballada. Es tracta d'una aplicació amb possibilitats, però que s'ha de millorar. Aquesta aplicació també està disponible per a Android.

### **Places with Augmented Reality (PAR) [7]**

Es tracta d'una altra aplicació amb les mateixes característiques que la que es pretén desenvolupar. Disposa d'un sistema de filtres i un nivell de detall de la informació dels llocs notables. També es poden desbloquejar algunes funcions extres (*premium features*) per 1,99 €, com guardar els llocs preferits o la posició on s'ha aparcat el cotxe (com en l'aplicació ACF).

### **Mountains AR (MAR) [27]**

Aquesta aplicació s'allunya una mica del concepte de navegació que es tracta en aquest treball, ja que és exclusivament per veure la posició de les muntanyes properes. Tot i així, és una bona aplicació per analitzar pel model matemàtic que s'hi utilitza. A causa de la diferència notable entre l'alçada dels pics, en aquesta aplicació s'han de posicionar bé els marcadors utilitzant geometria esfèrica o cilíndrica. Per tant, en aquest cas, la tercera coordenada (l'altitud) juga un paper molt important. La base de dades de muntanyes és completa i exhaustiva.

### **ArkSolut·CPH (AS) [6]**

Es tracta d'una altra aplicació Nav&AR amb una temàtica molt concreta, però amb un bon desenvolupament del model geomètric (esfèric). En aquest cas, es tracta d'una guia de les obres arquitectòniques daneses més significatives. Tot i que forma part d'un àmbit molt més específic que les altres, l'aplicació presenta molts extres i informació detallada. A més, la valoració dels usuaris és excel·lent.

A la taula 2.1 es mostra un resum comparatiu amb algunes característiques de les aplicacions anteriors. Algunes de les característiques són subjectives però poden resultar útils per extreure conclusions sobre quins són els punts forts i febles de les aplicacions que ja són al mercat.

App	Nav.	Simple	Filtres	Geom.	Extres	Idiomes	Preu (€)	Val.
VAR	Sí	Sí (8)	Sí	Plana	Alguns	3	Gratis	4,5 (15)
JNRA	Sí	Sí (6)	Sí	Esfèrica	Molts	12	Gratis	3 (77)
ACF	Sí	Sí (9)	No	Plana	Pocs	5	0,99 (F)	3 (94)
BTV	Sí	Sí (8)	Sí	Plana	Pocs	1	Gratis	- (0)
PAR	Sí	No (4)	Sí	Cilíndrica	Alguns	2	1,99 (F)	- (0)
MAR	No	No (2)	Sí	Cilíndrica	Molts	2	2,99	- (0)
AS	Sí	No (3)	No	Esfèrica	Molts	3	Gratis	5 (23)

Taula 2.1: Taula comparativa d'algunes aplicacions de l'App Store.

Les caracterísiques avaluades en la taula anterior es comenten a la llista següent:

- La primera columna, *Nav*, indica si l'aplicació disposa d'algun sistema de navegació.
- La segona columna, *Simple*, fa referència al nivell de dificultat que es pot trobar l'usuari objectiu que s'ha definit a l'apartat anterior a l'hora d'utilitzar l'aplicació, és a dir, si l'Abigail i/o en Ben pensarien que és una aplicació simple d'utilitzar. Es tracta d'una qualificació completament subjectiva i s'ha expressat amb un nota de 0 (dificultat excessiva) a 10 (trivial). En aquesta valoració, s'han tingut en compte el nombre d'elements que es mostren per pantalla i la complexitat de l'abre de navegació.
- La tercera columna, *Filtres*, indica si l'aplicació disposa de filtres o categories per classificar els llocs d'interès (o la informació).
- La columna *Geom.* indica el tipus de geometria que s'ha utilitzat per representar els elements del mapa. Les geometries esfèrica i cilíndrica reals es corresponen amb representacions que utilitzen tres coordenades per representar les ubicacions dels punts a l'espai. En les aplicacions la geometria de les quals s'ha descrit com a plana, només s'utilitzen les coordenades GPS bàsiques, latitud i longitud, encara que també es representen amb geometria cilíndrica.<sup>1</sup>
- La columna *Extres* indica qualitativament si l'aplicació disposa d'altres funcionalitats a més de la principal.
- En la columna *Idiomes* s'especifica el nivell de localització, és a dir, el nombre de llengües suportades per l'aplicació.
- La columna *Preu* conté el preu de venda de l'aplicació, en €, i s'ha indicat amb una (F) si aquest preu es correspon amb el d'una compra integrada (model *Premium*).
- La darrera columna, *Val.*, es correspon amb la valoració mitjana de l'aplicació a l'App Store (si ha estat valorada). També s'ha inclòs el nombre d'usuaris que han participat en la valoració.

<sup>1</sup> En algunes aplicacions de realitat augmentada que utilitzen geometria cilíndrica se sol utilitzar la distància entre l'usuari i el punt d'interès per calcular la coordenada  $z$ ; per tant, la representació cilíndrica no és pròpiament tridimensional. Amb aquest tipus de codificació, els punts més allunyats apareixen a la part superior de la pantalla i els més propers a la part inferior, però no representen l'altitud de l'ubicació.

Per últim, un aspecte que s'ha tingut en compte a l'hora d'analitzar les aplicacions anteriors és la qualitat de les bases de dades a què s'accedeix en cada cas. En aquest tipus d'aplicacions és molt important, per aconseguir una bona experiència d'usuari, que la base de dades amb la informació dels llocs d'interès sigui completa i fiable i estigui actualitzada.

## 2.2.2 Altres mercats

A continuació, s'enumeren aplicacions d'altres mercats que també s'han analitzat per tal de determinar les vetes de mercat i la retroacció dels usuaris (comentaris).

### Google Play

- metroDF Realidad Aumentada  
<https://play.google.com/store/apps/details?id=com.ngwise.MetroDF>
- My Augmented Reality  
<https://play.google.com/store/apps/details?id=com.neilneil.android.maps.stuff>
- Augmented Reality St James Way  
<https://play.google.com/store/apps/details?id=com.csversion2>
- Augmented Navigation  
<https://play.google.com/store/apps/details?id=com.justicesystems.augmented.navigation>
- Car Finder AR  
<https://play.google.com/store/apps/details?id=net.androoid.carxare>

### Windows Phone

- JobLens  
<http://www.windowsphone.com/es-es/store/app/joblens/d4491aeb-8ece-4652-8256-6626770009b7>
- MountainView  
<http://www.windowsphone.com/es-es/store/app/mountainview/40d5a5ba-17d5-4e57-aa40-edb96d8096aa>
- Cerca de mi (páginas amarillas)  
<http://www.windowsphone.com/es-es/store/app/cerca-de-mi/a80f6f41-88c6-4db6-bf78-f3b52fce4299>
- GPS Calculator PRO  
<http://www.windowsphone.com/es-es/store/app/gps-calculator-pro/48e2148a-b000-4a3e-9aab-6d5a6f763868>

En analitzar les funcions i la retroacció dels usuaris, s'ha arribat a algunes conclusions rellevants. La realitat augmentada, que es troba actualment en els seus inicis, és una tecnologia que agrada al públic i que presenta una tendència creixent. No són poques les propostes que s'han trobat en els diferents mercats i, més encara, les aplicacions de navegació són de les més populars i utilitzades en aquest sector. La major part dels comentaris negatius d'aquestes aplicacions fan referència a la base de dades i al seu estat de desactualització. Però també n'hi ha molts de relacionats amb la precisió en el posicionament dels objectes.

D'altra banda, tot i que no s'han trobat gaires comentaris negatius que hi facin referència, també s'ha observat una tendència general a sobrecarregar les vistes amb massa menús, icones i informació. Aquest excés d'elements presents en una pantalla tan petita fa incrementar la dificultat d'accedir a la informació desitjada (i la corba d'aprenentatge de l'usuari).

## CAPÍTOL

### 3

# OBJECTIUS I PLANIFICACIÓ

## 3.1 Descripció de l'aplicació

L'aplicació que es vol desenvolupar és una eina de realitat augmentada que mostri la informació dels llocs d'interès propers a l'usuari. La informació s'ha d'obtenir d'un servidor i s'ha de veure sobre les posicions reals dels llocs que es mostren per pantalla (mitjançant la càmera).

Ja s'ha vist que hi ha moltes aplicacions que realitzen funcions similars a la que es pretén desenvolupar en aquest projecte. Algunes tenen una qualitat excel·lent i inclouen moltes més funcions de les que s'han plantejat en aquest treball. No obstant això, com que el contingut de cada aplicació, que és el que interessa a l'usuari en última instància, depèn del servidor a què s'accedeix, hi ha un clar element diferenciador: la informació dels llocs d'interès. Per tant, si es disposa d'un volum d'informació prou important o la qualitat d'aquesta informació és superior a la de la competència, es pot aconseguir una aplicació exitosa per sobre de les de la competència.

Aquesta estratègia de *back-end*, tot i que s'ha tingut molt en compte a l'empresa,<sup>1</sup> no s'utilitzarà en aquesta fase de desenvolupament. L'objectiu d'aquest treball és oferir una solució de *front-end* que també millori en algun sentit les solucions de la competència.

Per començar, en la taula 1 de l'apartat anterior s'han analitzat la simplicitat, des del punt de vista de l'usuari, i el tipus de geometria que s'utilitza per representar les dades. És important destacar que la tendència observada és que les aplicacions que utilitzen les geometries esfèrica o cilíndrica presenten unes interfícies més complexes per a l'usuari. Això no hauria de

---

<sup>1</sup> La informació dels llocs d'interès que mostrarà l'aplicació final es correspon amb dades —que actualment no estan indexades— recollides per la mateixa empresa.

ser així. De fet, la complexitat en el model geomètric utilitzat per a la realitat augmentada hauria de ser transparent a l'usuari i, més encara, el fet d'utilitzar més informació per codificar els objectes, hauria d'incrementar l'experiència d'usuari. Per tant, un objectiu diferenciador que es perseguirà en aquest desenvolupament és el d'implementar una aplicació molt simple i intuïtiva, però que utilitzi un model de geometria cilíndrica o esfèrica.

Un altre aspecte que es vol cuidar en aquest treball és el sistema de classificació i filtratge de la informació. Es pretén etiquetar cada element de la base de dades (lloc d'interès) mitjançant un conjunt de categories (o *tags*) que el representin. El model que es vol utilitzar per a aquest sistema de classificació és un graf dirigit [15] codificat com a llista d'adjacència. Així, cada node del graf (categoria) estarà relacionat unidireccionalment amb les seves subcategories. Amb aquest sistema, si l'usuari busca restaurants, els llocs etiquetats com a “pizzeria” també apareixeran a la pantalla, encara que no s'hagin etiquetat com a “restaurants” (“pizzeria” serà una subcategoria de “restaurant”).

També cal facilitar l'accés a la informació que busca l'usuari. Per això, a més del filtre de categoria, l'usuari podrà seleccionar la distància radial màxima mitjançant un selector; per tant, només apareixeran els llocs propers a la posició actual de l'usuari. Amb el mateix objectiu, l'accés a la informació detallada dels llocs que apareixen per pantalla s'ha de realitzar amb un sol toc i mostrar-se mitjançant un *pop-up*, és a dir, sense vistes intermèdies ni menús desplegable.

## 3.2 Funcionalitats de l'aplicació

A continuació, es detallen les característiques principals i les funcionalitats que ha de tenir l'aplicació seguint els models de disseny de l'apartat anterior:

- Vista principal de navegació tridimensional amb realitat augmentada.
- Llista desplegable de filtres intel·ligents per seleccionar el tipus de lloc cercat.
- Selector de distància radial a la posició de l'usuari per seleccionar els llocs propers.
- Accés a la informació bàsica dels llocs que apareixen per pantalla amb un sol toc.
- Botó per marcar un lloc com a objectiu per eliminar la resta de llocs que apareixen.
- Base de dades dels llocs d'interès classificats per categories.

Finalment, una altra funció que seria interessant implementar (però que queda fora dels objectius d'aquest treball) és un botó que permetés mostrar directament la informació del lloc on es troba l'usuari. Aquest botó s'anomenaria “On sóc?” i, en prémer-lo, utilitzaria la posició de l'usuari i una certa tolerància (per exemple, entre 10 i 30 metres) per mostrar la informació detallada del lloc actual.



### 3.3 Objectius

Com que es tracta d'un projecte ambiciós, cal realitzar moltes tasques diferents i, en la major part de casos, aquestes tasques són independents de la resta. Com que el nombre de tasques és molt elevat per a un temps tan ajustat, s'assignarà una prioritat a cada una de les tasques mitjançant un nombre determinat d'etrelles (\*).<sup>2</sup> Només les tasques de prioritat elevada entraran en els objectius d'aquest projecte.

Així, basant-se en les funcions descrites en l'apartat anterior, a continuació es llisten els objectius de *front-end* i *back-end* que cal assolir per desenvolupar l'aplicació completa.

#### **Front-end (Objective-C i XCode)**

- F.1** - Dissenyar i programar el prototip de les vistes de l'aplicació (\*\*).
- F.2** - Dissenyar i programar un arbre de navegació simple i intuïtiu (\*\*).
- F.3** - Dissenyar el prototip en alta definició (\*\*).
- F.4** - Posicionar objectes tridimensionals en una vista 2D (\*\*).
- F.5** - Integrar els objectes del model 3D a la imatge de la càmera (\*\*).
- F.6** - Capturar les coordenades i l'orientació del dispositiu de l'usuari (\*\*).
- F.7** - Implementar les peticions HTTP al servidor (\*\*).
- F.8** - Implementar una llista de filtres i un selector de distància (\*\*).
- F.9** - Programar el panell d'informació bàsica (\*).
- F.10** - Vista de configuració i preferències d'usuari.
- F.11** - Implementar la funció del botó "On sóc?".

#### **Back-end (PHP i Symfony2)**

- B.1** - Dissenyar i programar l'entitat abstracta per guardar les localitzacions del mapa (\*\*).
- B.2** - Programar i mapejar la base de dades del servidor (\*\*).
- B.3** - Rebre i processar les peticions HTTP de l'aplicació mòbil (\*\*).
- B.4** - Dissenyar l'estructura de classificació de les dades (\*).
- B.5** - Dissenyar i programar l'entitat abstracta per guardar les etiquetes (*tags*) (\*).
- B.6** - Filtrar les dades mitjançant la distància i la posició enviades per l'usuari (\*).
- B.7** - Implementar un sistema de prioritats per als llocs d'interès.
- B.8** - Implementar un sistema de puntuacions per als llocs d'interès.

---

<sup>2</sup> Les tasques sense estrella són completament opcionals per a la realització d'aquest treball.

## Tasques de protocol

- P.1 - Omplir la base de dades amb informació provisional per testejar (\*\*).
- P.2 - Omplir la base de dades amb informació real.
- P.3 - Test d'usabilitat de l'aplicació amb usuaris (\*).
- P.4 - Provar la compatibilitat amb els diferents models d'iPhone disponibles (\*).
- P.5 - Realitzar proves en entorns controlats (\*\*).
- P.6 - Realitzar proves en entorns reals.
- P.7 - Realitzar l'informe i la documentació associada al projecte (\*\*).

Els objectius que s'han d'aconseguir assolir en aquest treball (primera versió de l'aplicació) són els que s'han marcat amb dues i tres estrelles (aquestes últimes són les més importants per la naturalesa de l'aplicació). Els objectius amb una estrella es desenvoluparan en funció del temps disponible en finalitzar les tasques de prioritat superior.

Com que l'aplicació de *front-end* es programarà en Objective-C, només estarà disponible per a la plataforma iOS. A més, pel tipus d'aplicació, queden exclosos els dispositius iPad (és a dir, només serà per a iPhone). Per a la part de servidor, s'utilitzarà el *framework* Symfony2 [28] de PHP i un servidor Amazon EC2 [2].

## 3.4 Llista de tasques

En aquest apartat es detallen les tasques i els terminis per assolir els objectius que s'han llistat en l'apartat anterior.

### Disseny del prototip en baixa defnició

En aquesta tasca s'han d'assolir els objectius següents:

- F.1 - Dissenyar el prototip (en baixa definició) de les vistes de l'aplicació (\*\*).

**Inici:** 30-03-15    **Finalització:** 05-04-15    **Temps:** 7 dies    **Dedicació:** TOTAL

### Disseny de l'arbre de navegació

En aquesta tasca s'han d'assolir els objectius següents:

- F.2 - Dissenyar un arbre de navegació simple i intuïtiu (\*\*).

**Inici:** 06-04-15    **Finalització:** 12-04-15    **Temps:** 7 dies    **Dedicació:** TOTAL

### Disseny del prototip en alta definició

En aquesta tasca s'han d'assolir els objectius següents:

- F.3 - Dissenyar el prototip en alta definició (★★).

**Inici:** 13-04-15    **Finalització:** 26-04-15    **Temps:** 14 dies    **Dedicació:** TOTAL

### Dissenyar l'estructura de *back-end*

En aquesta tasca s'han d'assolir els objectius següents:

- B.1 - Dissenyar l'entitat abstracta per guardar les localitzacions del mapa (★★).
- B.2 - Programar i mapejar la base de dades del servidor (★★★).
- B.4 - Dissenyar l'estructura de classificació de les dades (★).
- B.5 - Dissenyar l'entitat abstracta per guardar les etiquetes (*tags*) (★).

**Inici:** 27-04-15    **Finalització:** 17-05-15    **Temps:** 21 dies    **Dedicació:** PARCIAL

### Programar les entitats de *back-end*

En aquesta tasca s'han d'assolir els objectius següents:

- B.1 - Programar l'entitat abstracta per guardar les localitzacions del mapa (★★).
- B.5 - Programar l'entitat abstracta per guardar les etiquetes (*tags*) (★).

**Inici:** 18-05-15    **Finalització:** 31-05-15    **Temps:** 14 dies    **Dedicació:** PARCIAL

### Dissenyar l'estructura de *front-end*

En aquesta tasca s'han d'assolir els objectius següents:

- F.1 - Programar el prototip de les vistes de l'aplicació (★★).
- F.2 - Programar l'arbre de navegació (★★).
- F.8 - Implementar una llista de filtres i un selector de distància (★★).
- F.9 - Programar el panell d'informació bàsica (★).

**Inici:** 27-04-15    **Finalització:** 10-05-15    **Temps:** 14 dies    **Dedicació:** PARCIAL

### Accedir als sensors del dispositiu: posició i orientació

En aquesta tasca s'han d'assolir els objectius següents:

- F.6 - Capturar les coordenades i l'orientació del dispositiu de l'usuari (★★★).

**Inici:** 11-05-15    **Finalització:** 17-05-15    **Temps:** 7 dies    **Dedicació:** PARCIAL

### Implementar el model tridimensional esfèric

En aquesta tasca s'han d'assolir els objectius següents:

- F.4 - Posicionar objectes tridimensionals en una vista 2D (\*\*\*).

**Inici:** 18-05-15    **Finalització:** 24-05-15    **Temps:** 7 dies    **Dedicació:** PARCIAL

### Realitat augmentada

En aquesta tasca s'han d'assolir els objectius següents:

- F.5 - Integrar els objectes del model 3D a la imatge de la càmera (\*\*\*).

**Inici:** 25-05-15    **Finalització:** 07-06-15    **Temps:** 14 dies    **Dedicació:** PARCIAL

### Comunicació entre *front-end* i *back-end*

En aquesta tasca s'han d'assolir els objectius següents:

- F.7 - Implementar les peticions HTTP al servidor (\*\*).
- B.3 - Rebre i processar les peticions HTTP de l'aplicació mòbil (\*\*).
- B.6 - Filtrar les dades mitjançant la distància i la posició enviades per l'usuari (\*).

**Inici:** 18-05-15    **Finalització:** 31-05-15    **Temps:** 14 dies    **Dedicació:** PARCIAL

### Tests i proves

En aquesta tasca s'han d'assolir els objectius següents:

- P.1 - Omplir la base de dades amb informació provisional per testejar (\*\*).
- P.3 - Test d'usabilitat de l'aplicació amb usuaris (\*).
- P.4 - Provar la compatibilitat amb els diferents models d'iPhone disponibles (\*).
- P.5 - Realitzar proves en entorns controlats (\*\*).

**Inici:** 25-05-15    **Finalització:** 07-06-15    **Temps:** 14 dies    **Dedicació:** PARCIAL

### Documentació i memòria del projecte

En aquesta tasca s'han d'assolir els objectius següents:

- P.7 - Realitzar l'informe i la documentació associada al projecte (\*\*).

**Inici:** 15-06-15    **Finalització:** 21-06-15    **Temps:** 7 dies    **Dedicació:** TOTAL

### 3.5 Planificació general

Un cop detallades les tasques i la seva temporització, s’ha realitzat el diagrama PERT, que permet veure la planificació global del desenvolupament de l’aplicació.

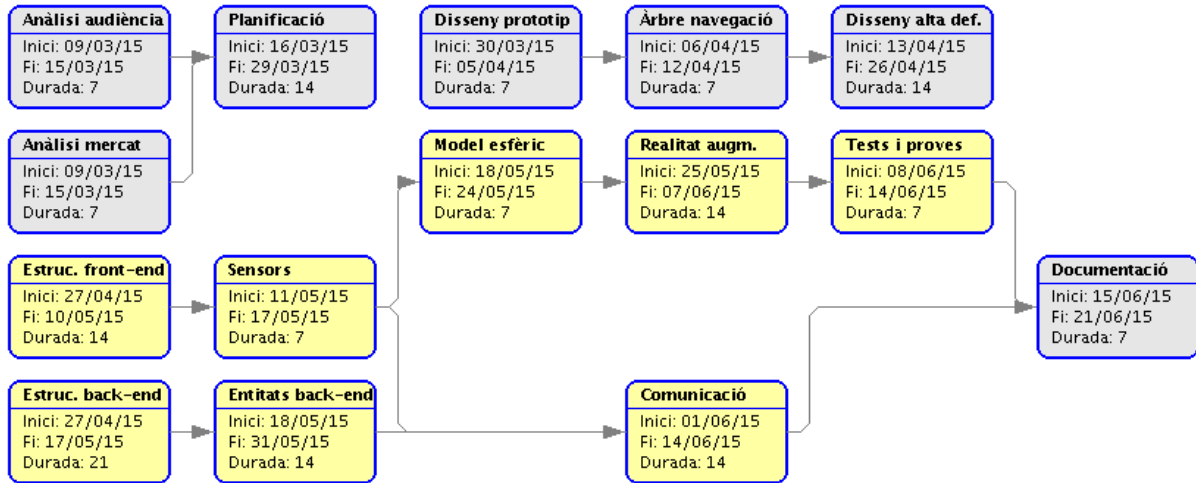


Figura 3.1: Diagrama PERT del projecte complet.

Amb el mateix objectiu, i per tal d’anitzar el camí crític del projecte, s’ha elaborat el diagrama de Gantt, en el qual s’han inclòs les etapes d’anàlisi i planificació corresponents a aquest apartat. Per facilitar la llegibilitat, s’ha separat el diagrama en dues etapes: etapa de disseny i etapa de desenvolupament.



Figura 3.2: Diagrama de Gantt de l’etapa de disseny.

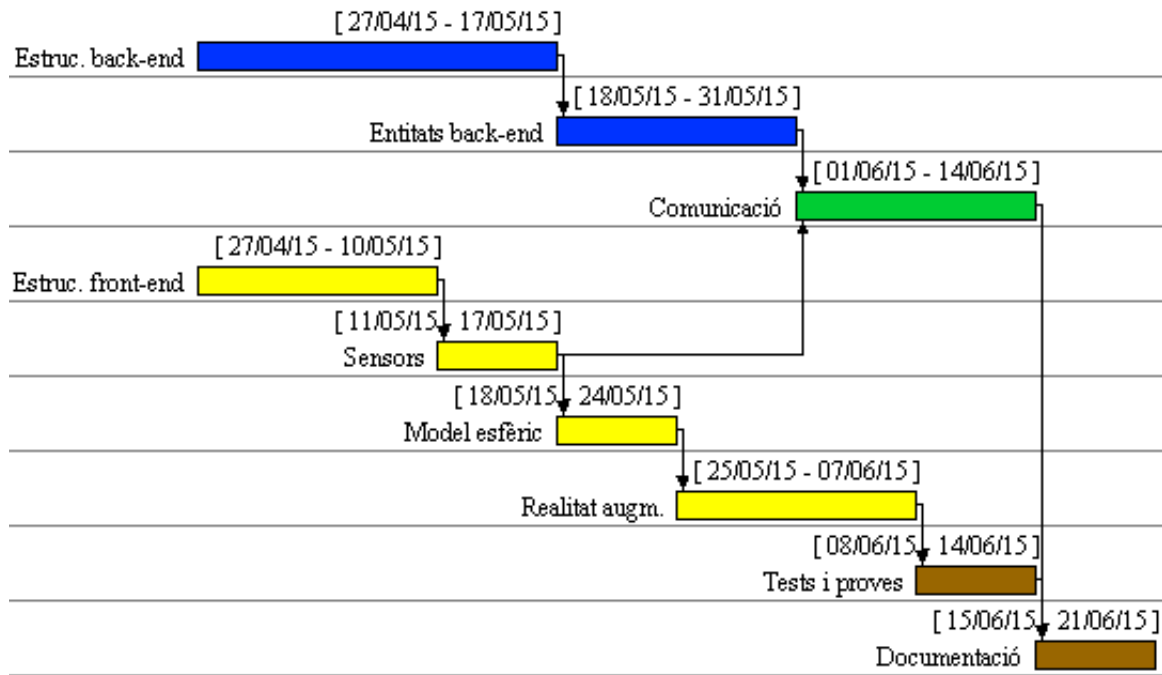


Figura 3.3: Diagrama de Gantt de l'etapa de desenvolupament.

Al llarg d'aquest apartat s'ha utilitzat una codificació de colors per distingir les diferents àrees del desenvolupament. Així doncs, el gris correspon al conjunt de punts d'aquest document; el rosa, a les tasques de disseny; el blau, al desenvolupament del *back-end*; el groc, al desenvolupament *front-end*; el verd, a la comunicació entre els dos anteriors, i el marró, a les proves i a la documentació.

# CAPÍTOL

## 4

# DISSENY

## 4.1 Disseny conceptual i arquitectura de l'aplicació

Un cop enlestida la planificació del treball, toca començar el procés creatiu que permeti donar vida a l'aplicació. Aquest procés comença amb la definició de l'arquitectura de l'aplicació mitjançant els diagrames de casos, l'esquema de comunicació i els fluxos d'usuari. A partir de tota aquesta informació, es podran realitzar els primers dissenys de les vistes de l'aplicació, pas previ al desenvolupament. En la mesura que aquest disseny estigui ben definit i ajustat a les necessitats de l'usuari, el procés de desenvolupament esdevindrà més simple i eficaç.

En primer lloc, però, com que ja s'han descrit les funcionalitats i les necessitats que es pretenen cobrir, en aquest punt sembla adequat assignar un nom provisional a l'aplicació per poder referenciar-la a la resta del treball.

### 4.1.1 L'elecció del nom

Es pretén que el nom sigui curt, fàcil de recordar i, sobretot, que representi i simbolitzi l'essència de l'aplicació. La idea bàsica de l'aplicació és aprofitar el sistema de referència “per defecte” en el qual hom és origen de coordenades i centre del seu món. Aquest sistema de referència és el més simple i intuïtiu per a la majoria de persones. Cada usuari es troba a una distància 0 respecte d'ell mateix (origen) i la resta d'objectes se situen a una certa distància en línia recta (en una direcció i sentit determinats). Així, si l'usuari es mou, l'origen de coordenades també es mou amb ell i les distàncies i direccions que defineixen la posició de la resta d'objectes canvien per adaptar-se a allò que veu.

A la figura següent es mostren l'usuari i els seus llocs d'interès propers (esquerra) i la representació de l'usuari amb el sistema de referència central (dreta).

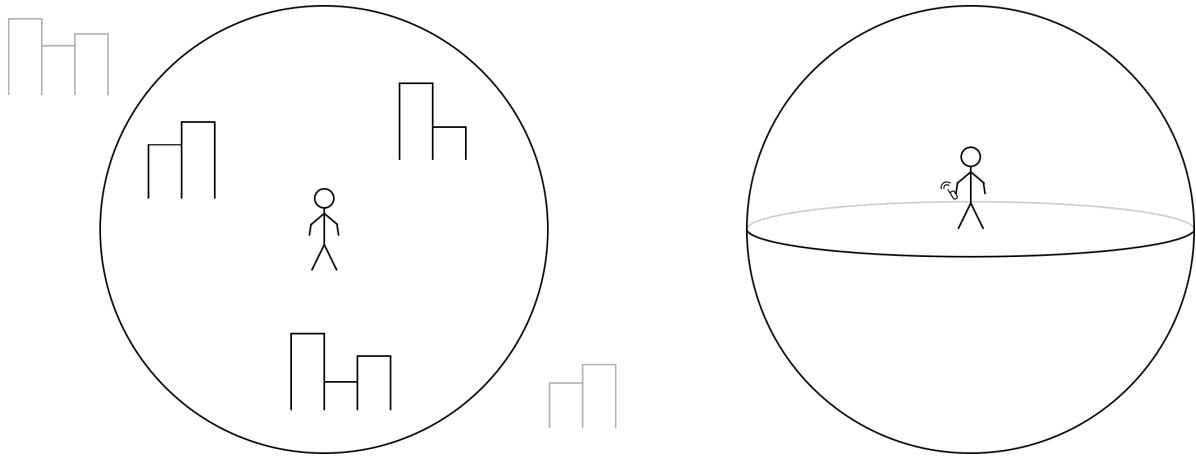


Figura 4.1: Esquema conceptual de l'aplicació *Radius AR*

Des d'aquest punt de vista, l'aplicació es comporta com una finestra que ajuda l'usuari a aconseguir informació d'allò “que té al seu voltant”. Aquest “que té al seu voltant” és, de fet, el concepte clau de l'aplicació i s'associa a les idees formals de *distància radial* i *radi esfèric*. D'altra banda, pel fet de ser una aplicació de realitat augmentada —i per tal d'aprofitar el màrqueting associat a les sigles AR—, sembla adequat utilitzar la nomenclatura habitual d'aquest tipus d'aplicacions, i.e. afegir la cua *Augmented Reality* al final. Així, el nom escollit és:

*Radius Augmented Reality (Radius AR)*

Un cop batejada l'aplicació, es pot situar l'Abigail o en Ben en el centre d'una d'aquestes esferes i, a partir de la idea conceptual que s'ha desenvolupat, començar el disseny del diagrama de casos d'ús.

### 4.1.2 Diagrama de casos

Com que l'aplicació que l'Abigail i en Ben tenen al mòbil és molt simple, en aquest apartat s'han inclòs dos tipus d'actors més a part de l'**usuari**. D'una banda, l'**administrador**, que interactuarà únicament amb el *back-end* per actualitzar la informació de la base de dades. De l'altra, la mateixa base de dades del servidor **Amazon**, que representa un actor abstracte que gestiona i ofereix les dades que necessita l'usuari.

A partir d'aquests tres actors s'ha dissenyat un diagrama de casos general, que engloba els dos tipus de sistemes que s'han de desenvolupar: el *front-end* amb Objective-C i el *back-end* amb Symphony. A la figura següent, es mostra l'esquema UML que representa aquest diagrama de casos d'ús amb el software *ArgoUML* [30].



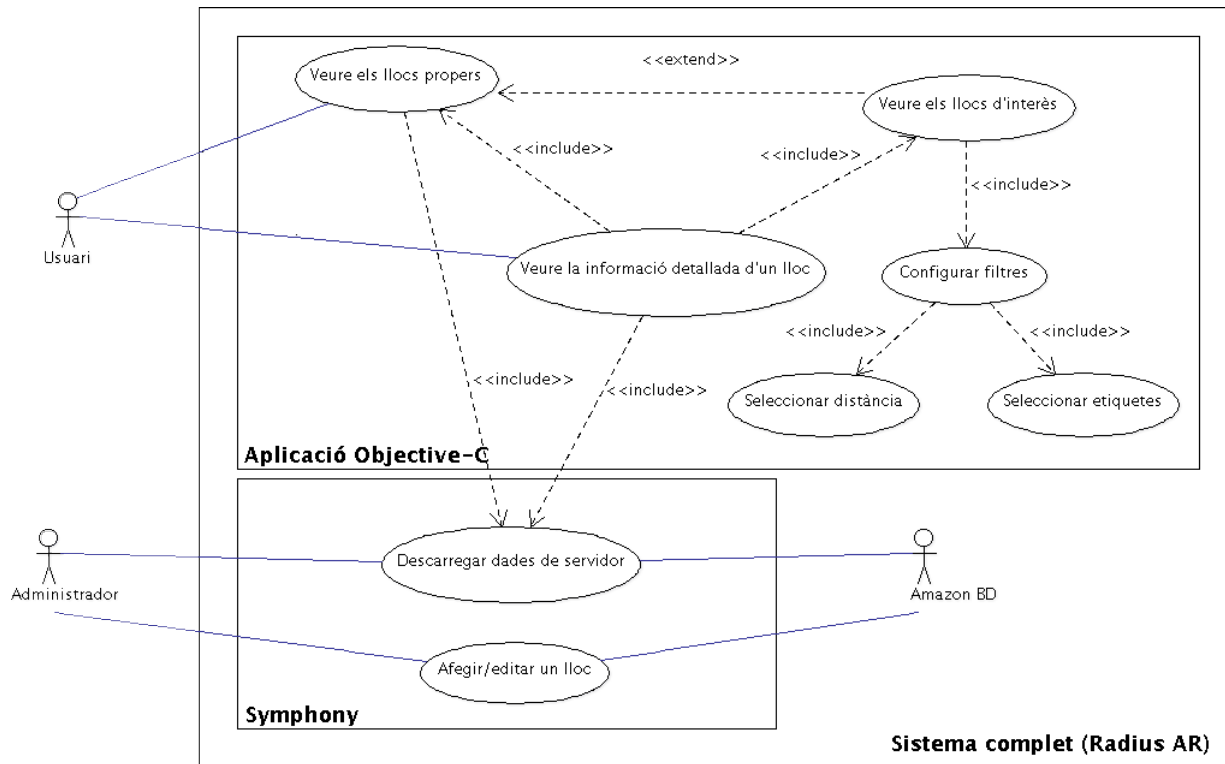


Figura 4.2: Diagrama de casos de l'aplicació *Radius AR*

Com que per utilitzar l'aplicació no cal registrar-se, les úniques accions que vol efectuar l'usuari estan relacionades amb veure allò que té al voltant a través de l'aplicació. Cal diferenciar, però, l'acció de veure *llocs propers* (llocs propers a la posició actual de l'usuari) de l'acció de veure *llocs d'interès* (tipus de llocs propers que vol trobar). Així, en l'aplicació *Radius AR*, un lloc d'interès serà un cas particular de lloc proper que apareix per pantalla com a resultat d'un filtratge de dades. Aquest procés inclou l'acció de configurar els filtres, que al seu torn inclou les accions de seleccionar la distància radial i seleccionar les etiquetes dels llocs. No obstant això, com que totes les accions relacionades amb la configuració dels filtres són mitjans per aconseguir veure els llocs d'interès, no s'han relacionat directament amb l'actor usuari. En canvi, veure la informació detallada dels llocs (tant propers com d'interès) és un altra possible acció que es relaciona directament amb l'usuari. Totes les accions que mostren informació dels llocs han de relacionar-se amb el *back-end* mitjançant un accés a les dades del servidor. Ja siguin les dades de contacte dels llocs, ja siguin les coordenades dels llocs, l'aplicació *Radius AR* necessita un accés constant a la xarxa per obtenir aquesta informació. Aquest accés s'ha englobat en una única acció, *Descarregar dades del servidor*, que relaciona al servidor Amazon amb les accions que interessin a l'usuari. A més, aquests accessos també relacionen el software Objective-C del terminal de l'usuari amb el PHP (Symphony) del *back-end*.

Finalment, l'administrador —de moment, el mateix desenvolupador— interactua amb el servidor per afegir i modificar les dades dels llocs. Com que per als objectius d'aquest projecte no s'ha de desenvolupar la consola d'administració,<sup>1</sup> aquesta acció s'ha englobat en *Afegir/editar*

<sup>1</sup> Les dades es penjaren directament al servidor utilitzant *phpMyAdmin* [29] o l'eina *Postman* del navegador *Chrome*, tot i que seria interessant, de cara a la continuació del projecte, desenvolupar una aplicació híbrida

*un lloc*, tot i que, de fet, si s'hagués d'implementar la consola, caldria representar-la mitjançant un diagrama a part.

A més de l'usuari, l'administrador i el servidor, hi ha altres "actors" secundaris que no s'han inclòs en l'esquema UML anterior. Per exemple, l'operador mòbil, el satèl·lit GPS o, fins i tot, els mateixos sensors del terminal mòbil. Aquests elements estan més relacionats amb la capacitat del sistema per oferir el servei i amb la xarxa de comunicació entre els dispositius que no pas amb el diagrama de casos d'ús. Per aquest motiu, aquests elements s'han representat mitjançant el següent esquema de comunicació.

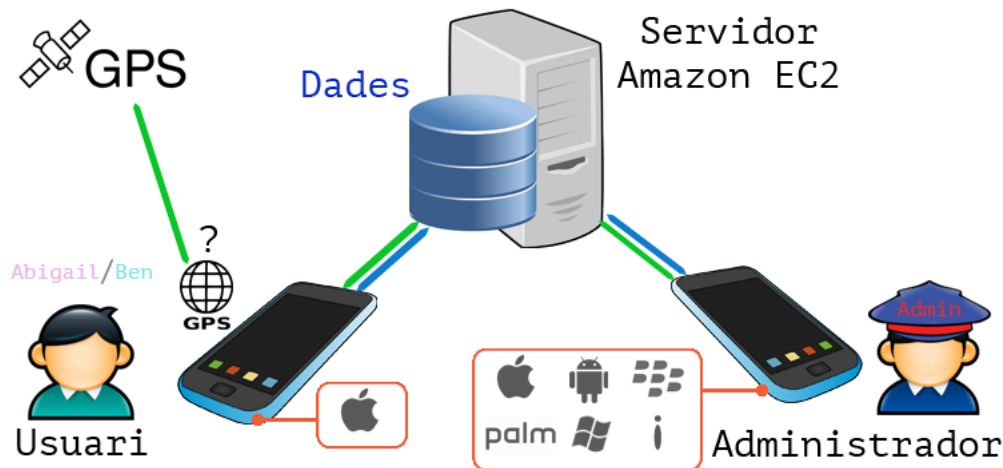


Figura 4.3: Esquema de comunicació de l'aplicació *RADIUS AR*

On les línies en verd (*downlink*) i blau (*uplink*) representen els diferents canals de comunicació entre els nodes.

### 4.1.3 Fluxos d'usuari

Un cop finalitzats el diagrama de casos d'ús i l'esquema de comunicació, es poden plantejar els fluxos d'usuari de l'aplicació del client. Aquests esquemes UML s'han desenvolupat utilitzant la mateixa aplicació que en el diagrama de la figura 2: *ArgoUML*. S'ha intentat que aquests fluxos fossin exhaustius per tal de recollir la majoria de les possibles situacions amb què es pot trobar l'usuari. Aquesta exhaustivitat pot simplificar molt el procés de disseny de la interfície d'usuari en les etapes posteriors.

El primer flux es correspon amb l'habilitació dels serveis i els sensors. Com que l'aplicació *RADIUS AR* necessita accés a la posició, a la càmera i a internet, cal que l'usuari doni els permisos corresponents i tingui accés a la xarxa abans de poder accedir a la vista principal de l'aplicació ("Es mostren els llocs propers"). Si l'usuari no pot accedir a la vista principal per algun motiu, s'ha de mostrar una vista en què s'expliqui a l'usuari la manera d'activar els permisos que no es troben actius.

---

multiplataforma per modificar les dades amb facilitat.

En la figura següent es mostra el flux general des que l'usuari obre l'aplicació fins que decideix tancar-la.

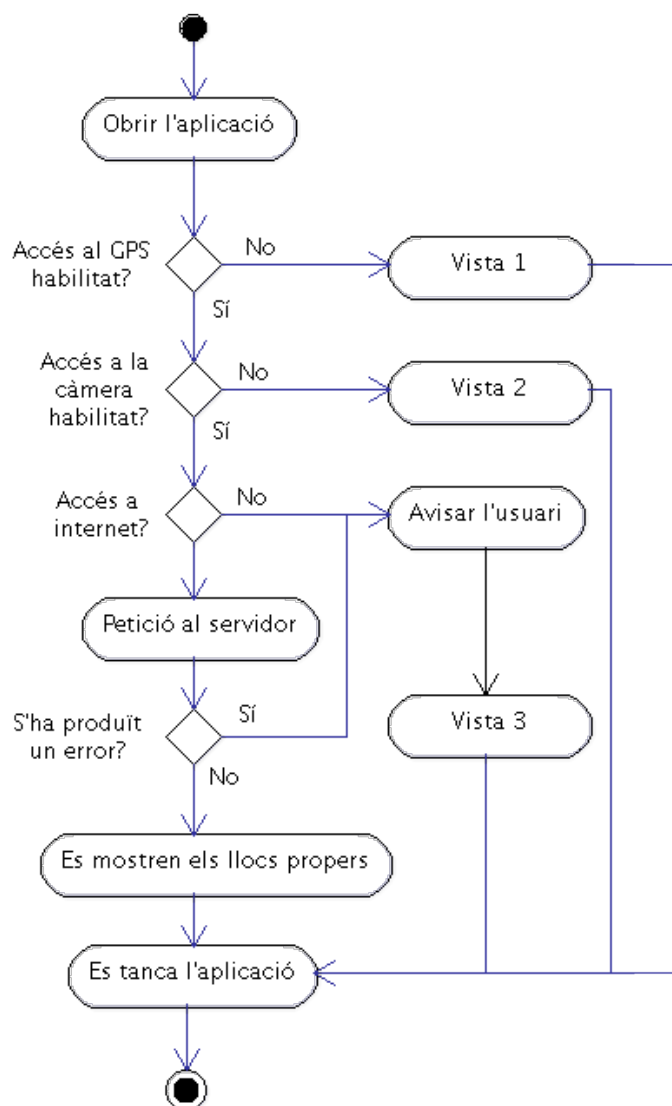


Figura 4.4: Flux dels permisos de l'aplicació

Les vistes 1 i 2 expliquen com activar el GPS i l'accés a la càmera si l'usuari contesta "No" a la pregunta que mostra el sistema operatiu per accedir als sensors. Tant si hi ha algun error amb el servidor com si no hi ha connexió a internet, cal avisar l'usuari i passar a la vista 3, en què s'expliquen els diferents motius que poden impedir l'ús normal de l'aplicació.

Si l'usuari ha arribat a la vista principal de l'aplicació, pot començar a utilitzar-la per trobar la posició i la informació dels llocs propers. Aquest flux comença amb una petició al servidor en què s'indiquen la posició i el radi d'interès (en el primer accés, s'envia el radi per defecte que hi ha a l'aplicació). A partir d'aquest punt, l'usuari pot interactuar amb els diferents elements que hi ha per pantalla i amb l'orientació del mòbil per trobar la posició dels llocs que li interessin. En prémer qualsevol de les etiquetes dels llocs que apareixen, es pot accedir a la informació detallada del lloc (vista detall).

Cada una de les etiquetes que es mostren per pantalla s'ha de poder eliminar amb un sol gest per veure les etiquetes posteriors. Per tal de desfer aquesta acció i/o per forçar una nova petició al servidor es pot prémer al botó "Refrescar". A continuació, es mostra aquest esquema general.

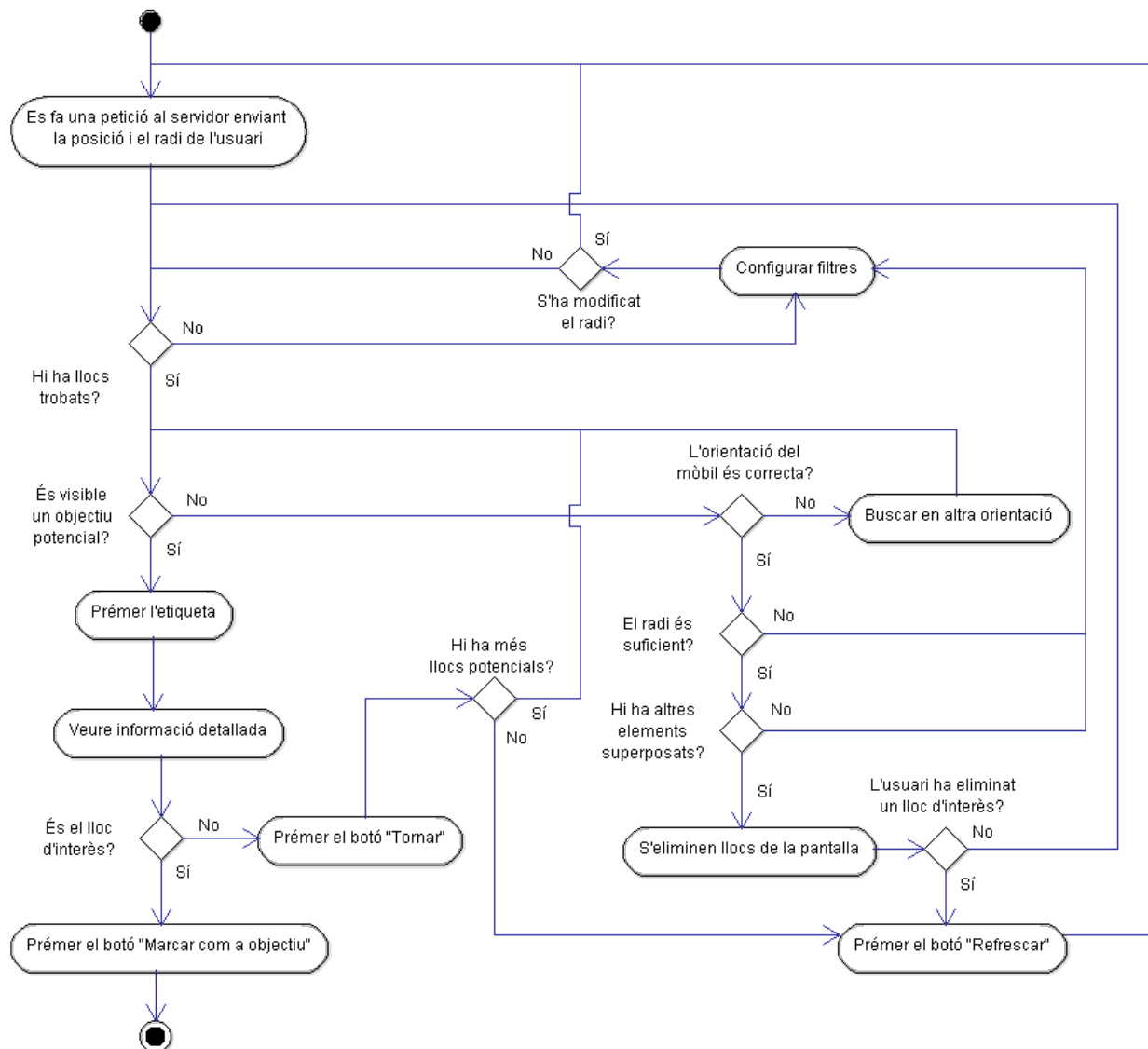


Figura 4.5: Flux bàsic de l'aplicació

En aquest flux també s'ha afegit una nova funcionalitat que pot ser interessant per als usuaris que ja han trobat un lloc on volen anar. En la vista d'informació detallada d'un lloc s'inclou el botó "Marcar com a objectiu", que fa que s'eliminin la resta de llocs que apareixen a la pantalla de realitat augmentada. Així, l'usuari només veu el lloc on vol anar.

Pel que fa al tractament de la informació, també és important destacar que la modificació dels filtres de dades només provoca un accés al servidor si s'ha canviat el radi (veure figura 5), ja que els filtres per categoria (etiquetes dels llocs) només modifiquen el nombre d'etiquetes que es mostren per pantalla, és a dir, se segueix mantenint tota la informació sense filtrar a la memòria del terminal de l'usuari. Si s'aprofundeix més en el sistema de filtres de l'aplicació s'arriba a un flux com el que es mostra a continuació.

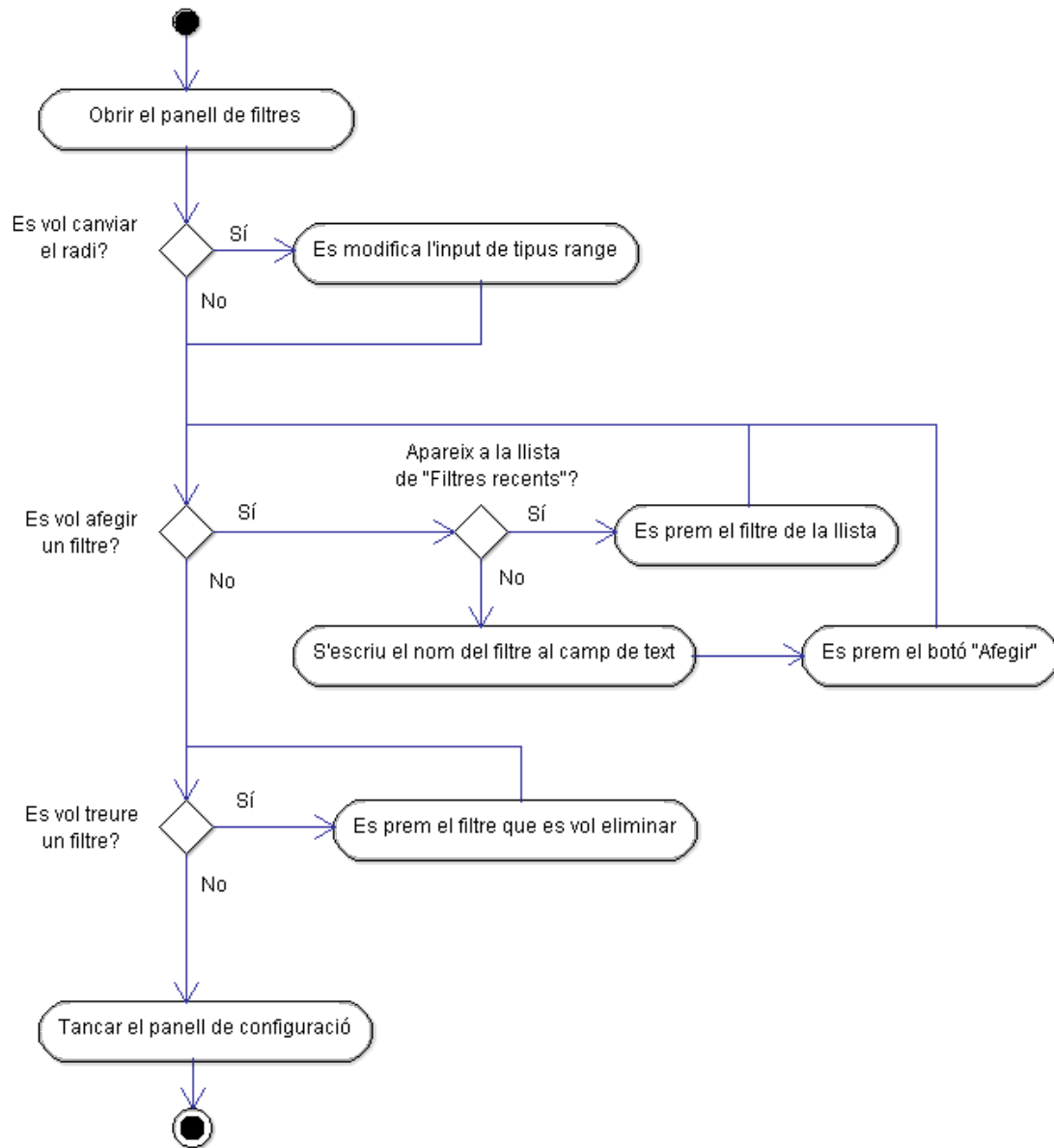


Figura 4.6: Flux de filtratge

L'usuari ha de poder afegir i eliminar etiquetes (*tags*) fàcilment per seleccionar els tipus de llocs mostrats per pantalla. A més, també pot ser molt interessant permetre a l'usuari seleccionar els filtres que utilitza habitualment en les seves cerques (llista de filtres recents). Amb aquest esquema, per afegir un filtre, l'usuari pot escriure el que vol o bé prémer un dels filtres recents per afegir-lo directament. Per eliminar un dels filtres actius, només cal prémer sobre el filtre i, un cop eliminat, el filtre s'afegeix a la llista de filtres recents (si hi ha espai). Aquests filtres han d'actuar de manera multiplicativa (AND), és a dir, si se seleccionen dos filtres, només apareixeràn els llocs que tenen simultàniament les dues etiquetes. Així, si l'usuari escull molts filtres, especifica més el tipus de lloc que busca.

A continuació, tot i que aquestes funcionalitats queden fora de l'abast d'aquest projecte, s'han inclòs els fluxos del botó "On sóc?" i la vista de configuració de l'aplicació.

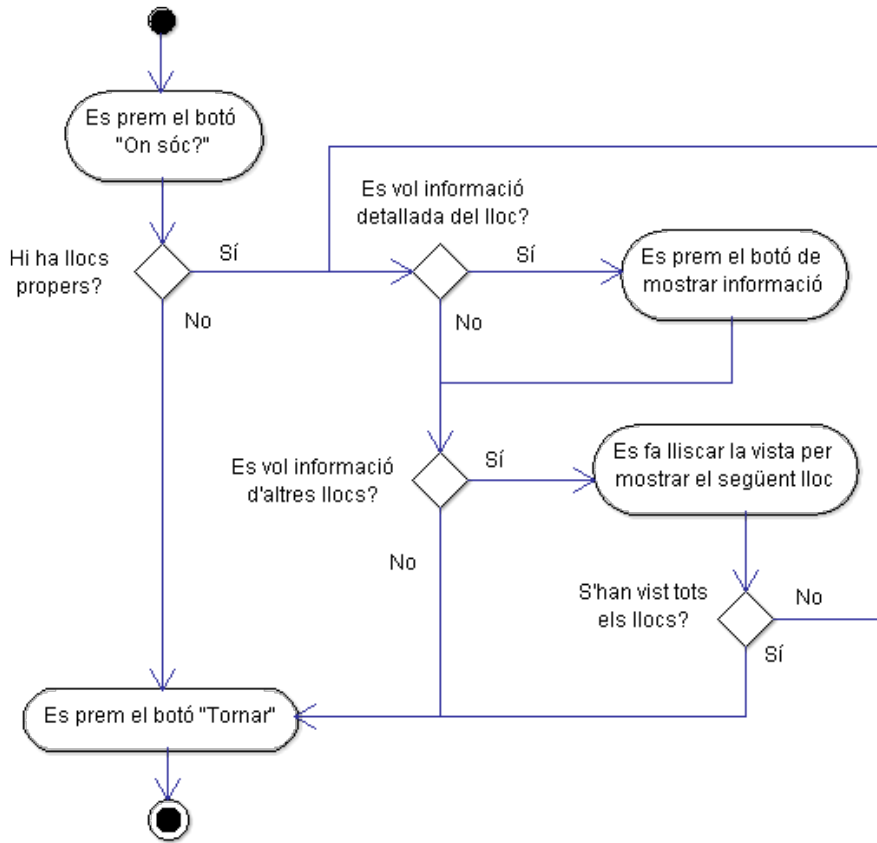


Figura 4.7: Flux del botó “On sóc?” (extra)

En la vista de configuració s’entén que hi ha tants botons de tipus *Switch* com opcions disponibles. En cas que s’hagui d’utilitzar un altre tipus de selector (per exemple, numèric o de múltiples opcions) l’esquema és el mateix.

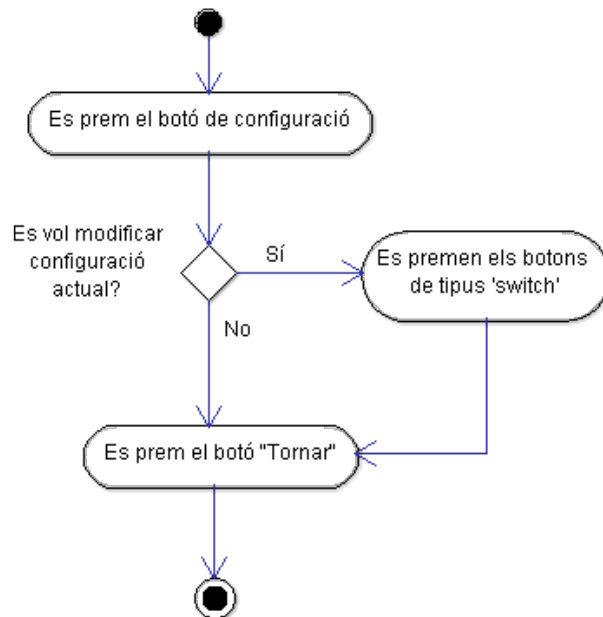


Figura 4.8: Flux de la vista de configuració (extra)

## 4.2 Prototip de baixa resolució i *wireframes*

Un cop especificats els principals fluxos, es poden començar a realitzar els primers esborranys de les vistes de l'aplicació. En aquest punt, cal tenir presents la mida i la relació d'aspecte de les pantalles en què funcionarà l'aplicació (iPhone) i dibuixar els esborranys de manera que s'hi adaptin. Per crear aquests esborranys s'ha utilitzat l'eina *Balsamiq* [12], que es va introduir a l'assignatura *Disseny d'interfícies interactives* d'aquest postgrau. A continuació, es mostren les dues vistes principals de l'aplicació *Radius AR* que es volen implementar en aquest projecte i la vista extra relacionada amb el botó "On sóc?".

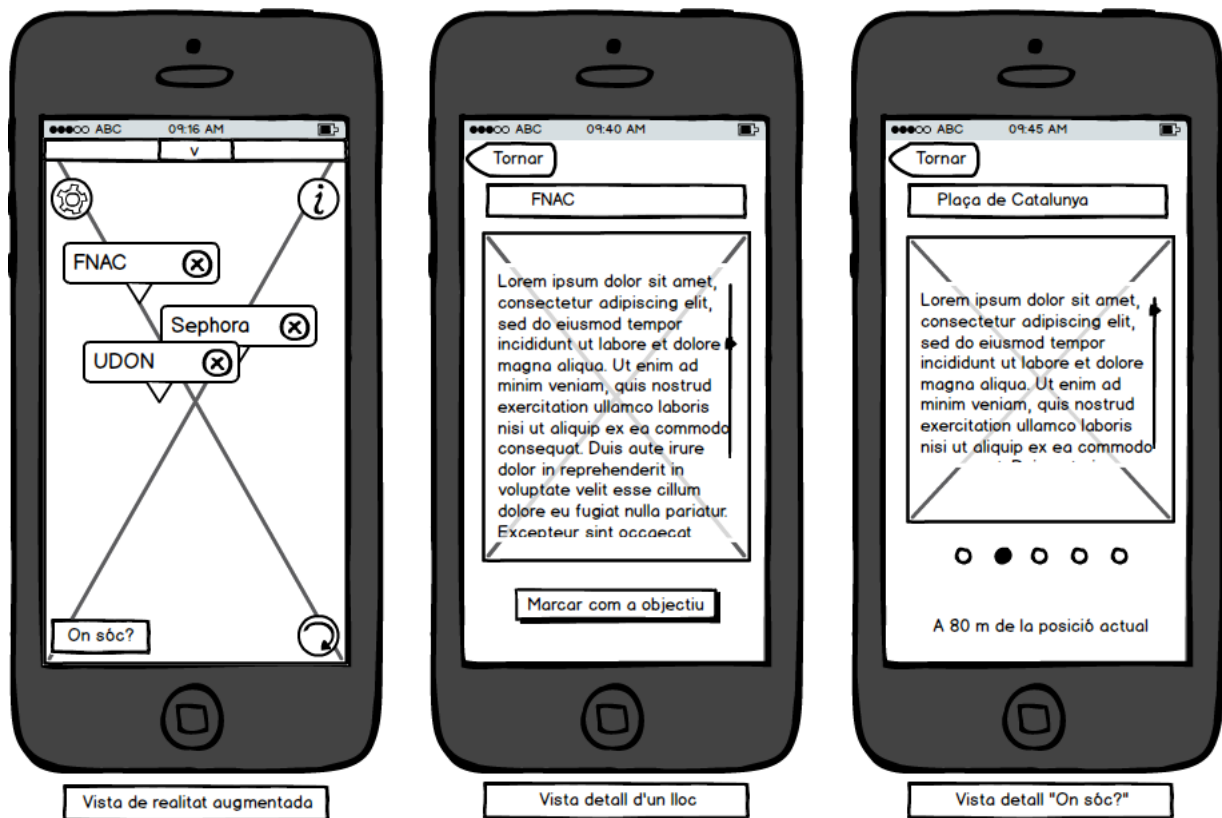


Figura 4.9: Prototip en baixa resolució de les vistes de principals

En la vista principal, apareixen les etiquetes dels llocs que es mostren per pantalla. Aquestes etiquetes són enllaços a la vista detall del lloc i, al seu torn, presenten un botó en forma de X per eliminar l'etiqueta de la pantalla. En aquesta vista també hi ha cinc botons que permeten accedir a la resta de funcions i vistes de l'aplicació. A la barra superior hi ha el botó per desplegar el panell dels filtres (figura 10a). A la part superior esquerra hi ha el botó en forma d'engrenatge, que permet accedir a la vista de configuració general de l'aplicació (figura 10b). A la part superior dreta hi ha un botó que permetria accedir a la vista d'informació i contacte de l'equip de desenvolupament (figura 10c).<sup>2</sup> Finalment, a la part inferior hi ha el botó "On sóc?" (esquerra) i el botó per refrescar (dreta).

<sup>2</sup> Aquest botó s'ha eliminat del prototip final, ja que l'aplicació s'ha de convertir en una funcionalitat de la xarxa social.

Les altres dues vistes de la figura 9 mostren informació dels llocs superposada a les fotografies que hi ha a la base de dades. La vista de detall d'un lloc mostra informació molt bàsica i dos botons: un per tornar i un altre per marcar el lloc com a objectiu (elimina totes les etiquetes llevat de la del lloc en qüestió). Per a la vista "On sóc?" s'ha utilitzat una estructura de *scroll* lateral amb paginació per veure els llocs més propers i importants<sup>3</sup> que hi ha al voltant. De fet, els llocs més propers no es mostraran per pantalla per evitar problemes de precisió en el GPS i per millorar l'experiència d'usuari; per tant, la informació d'aquests llocs "tant propers" només estarà disponible prement el botó "On sóc?".

En la figura següent es mostren l'aspecte del menú de filtres i les vistes de configuració i informació.

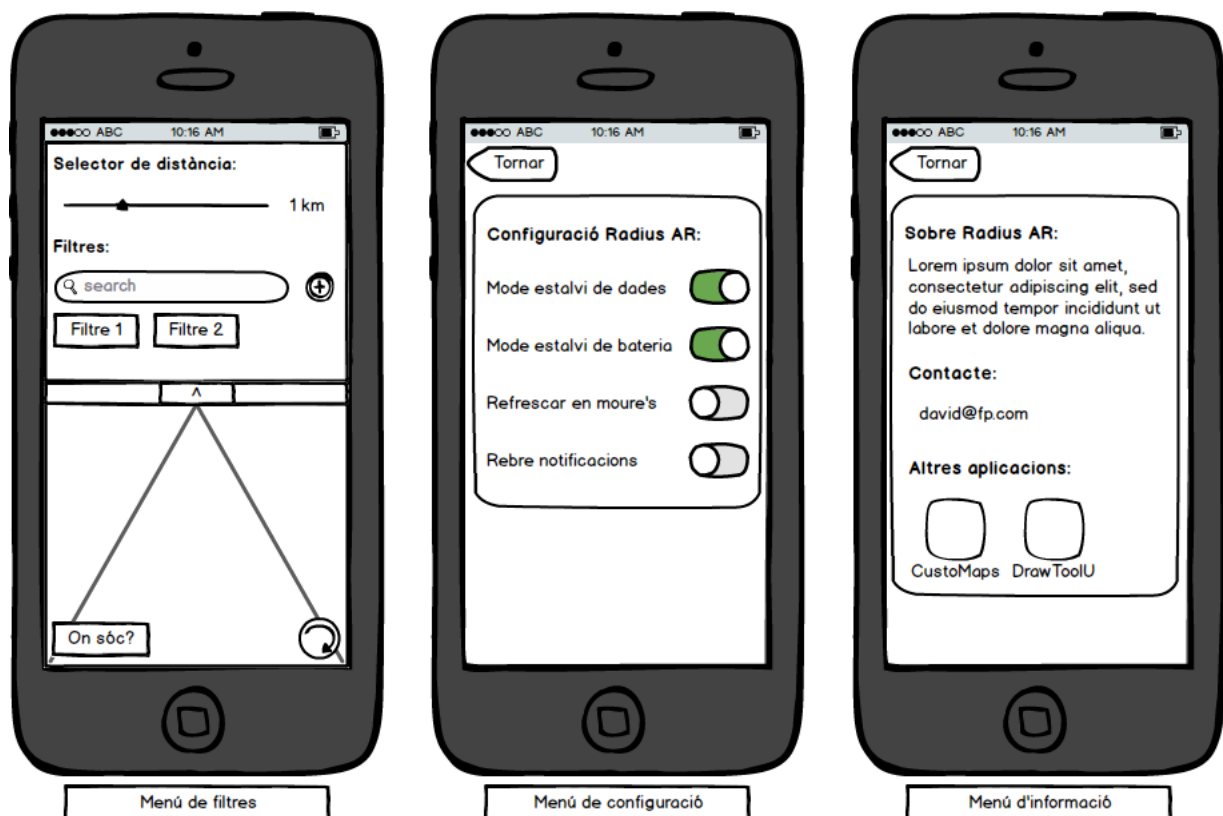


Figura 4.10: Prototip en baixa resolució de les vistes de configuració

Finalment, pel que fa a les vistes 1, 2 i 3 descrites en el flux dels permisos (figura 4), no se n'han realitzat esborranys. Aquesta decisió respon a dos motius principals: d'una banda, perquè són vistes estàtiques informatives que no tenen cap tipus d'interacció i, de l'altra, perquè es vol acotar el nombre de vistes que s'han de programar en l'etapa de desenvolupament d'aquesta primera versió. Per tant, en una primera etapa de desenvolupament, se suposarà que l'usuari ha habilitat tots els permisos necessaris i que disposa de connexió a internet.

<sup>3</sup> Això forma part del sistema de prioritats que s'ha d'implementar en el *back-end*.



### 4.3 Prototip d'alta resolució i *wireflow*

Un cop enllestit el prototip de baixa resolució, es pot procedir a dissenyar una versió més visual i interactiva de l'aplicació. El prototip d'alta resolució s'ha creat utilitzant l'eina *Axure* i un conjunt de recursos i fotografies fets per l'autor a la plaça de Catalunya (Barcelona). A continuació, es mostren algunes captures del disseny aproximat de l'aplicació *Radius AR*.



Figura 4.11: Prototip en alta resolució 1. Vista de realitat augmentada i menú de filtres

En el panell de filtres es pot veure que s'ha canviat el camp de text per escriure els filtres a afegir per un camp desplegable. També hi ha els filtres recents que es poden afegir directament a la llista de filtres aplicats. A més, com ja s'ha comentat, el botó per accedir a la informació de la vista principal s'ha eliminat en aquest prototip.

Un altre detall important sobre l'aspecte visual és l'ús de transparències en la major part d'elements presents a la pantalla. Aquest estil, juntament amb el disseny minimalista dels contenidors i els botons, respon a la nova estètica que utilitzen els sistemes operatius d'Apple [3].

En la figura 12, es veuen les vistes detall associades a les dues etiquetes que es mostren a la vista de realitat augmentada (se suposa que l'usuari ha buscat hamburgueseries mitjançant els filtres). Tot i que la informació que es mostra és molt senzilla, s'han inclòs les etiquetes, la informació de contacte i l'eslògan del lloc a banda de la descripció bàsica que apareixia en el prototip de baixa resolució. A més, s'ha inclòs una parella de colors específica per personalitzar l'aspecte del panell del títol. Aquestes millores només són una proposta de cara al futur de l'aplicació i no es tindran en compte a l'hora de desenvolupar la primera versió de l'aplicació.



Figura 4.12: Prototip en alta resolució 2. Vistes detall dels llocs

En la figura 13 es mostren la vista de configuració i la vista (extra) del botó “On sóc?”. En la vista de configuració hi ha interruptors que permeten habilitar i deshabilitar les diferents opcions. El mode d'estalvi de dades evita descarregar les imatges dels llocs (la part que consumeix el gruix de les dades), mentre que el mode d'estalvi de bateria deshabilita la càmera i el GPS en certes circumstàncies (per exemple, en les vistes detall). L'última opció de configuració serveix per cridar automàticament l'acció de refrescar les dades quan l'usuari es mou una certa distància —però sense mostrar el missatge “carregant dades”—, de manera que les etiquetes es refresquen automàticament sense molestar l'usuari.



Figura 4.13: Prototip en alta resolució 3. Configuració i vista “On sóc”

Com ja s'ha comentat, en la vista “On sóc?” hi ha el sistema de paginació per veure les imatges dels llocs molt propers a l'usuari. En aquesta versió d'alta definició s'ha inclòs un botó per mostrar i amagar la informació del lloc. La informació està amagada per defecte perquè l'usuari pugui veure d'un cop d'ull els llocs propers. A la part inferior, es mostra la distància radial aproximada a què es troba el lloc.



Figura 4.14: Prototip en alta resolució 4. Informació de la vista “On sóc”

Aquest disseny es troba disponible a la xarxa per tal de realitzar els testos d'usuari:

Sense menú de navegació: <http://cqrfam.axshare.com/#c=2>

Amb menú de navegació: <http://cqrfam.axshare.com>

## 4.4 Icona de l'aplicació

Per al disseny de la icona de l'aplicació, s'ha utilitzat l'estil minimalista que caracteritza les últimes versions del sistema operatiu d'Apple. A continuació, es mostren tres de les propostes que es van dibuixar abans d'escollir la versió final.



Figura 4.15: Propostes d'icones

Totes tres utilitzen el mateix símbol central, que representa una barreja entre la "R" de "Radius" i el dibuix del radi d'una circumferència. Pel que fa al color, s'ha escollit aquest to turquesa que recorda als tons dels paisatges paradisiacs amb vegetació, mar i un cel serè. La icona escollida ha estat la primera.

# CAPÍTOL

## 5

# DESENVOLUPAMENT

## 5.1 *Hardware i software utilitzat*

En aquest apartat es presenta el contingut pràctic del treball, és a dir, el desenvolupament de l'aplicació RadiusAR. Aquest desenvolupament, com ja s'ha comentat, consta de dues parts ben diferenciades. D'una banda, l'aplicació de *front-end*, basada en Objective-C i les llibreries que ofereix Apple, i, de l'altra, els mòduls del servidor (*back-end*), escrits en PHP i basats en el *framework Symphony2*.

El *hardware* que s'ha utilitzat per desenvolupar aquest projecte és un MacBook Air (4 GB de RAM i processador Intel Core i5 de 1,4 GHz) amb el sistema operatiu OS X Yosemite versió 10.10.3 i els terminals de proves han estat dos iPhone5 (un amb iOS 7.1.2 i l'altre amb iOS 8.1.2). Pel que fa al software, s'han utilitzat tres programes diferents. Per a la part de servidor, s'ha utilitzat PhpStorm 8.0.2; per al desenvolupament de l'aplicació, Xcode 6.1.1, i, per a les proves de comunicació i la consola d'administració, Sublime Text 2.0.2.

## 5.2 Models i diagrames dels objectes

Com ja s'ha comentat, per al desenvolupament d'aquest projecte s'han utilitzat els llenguatges PHP (servidor) i Objective-C (client). S'ha utilitzat la programació orientada a objectes a tots dos costats. Això vol dir que s'han hagut de programar dues versions dels objectes que s'emmagatzemen a la base de dades: *location*, que representa el lloc, i *tag*, que representa una etiqueta que classifica el lloc. En aquest apartat, es presenten les classes completes de no més aquests quatre objectes, sinó de tots els que s'han desenvolupat per a l'aplicació Radius AR.

### 5.2.1 Classes dels objectes de *back-end*

Per a cada classe del servidor (*Bundle*) s'han desenvolupat dos objectes: l'entitat (*Entity*), que representa l'estructura de dades i permet mapejar la base de dades, i el procés que captura les peticions i les gestiona (*Handler*).

El primer objecte involucrat, l'etiqueta del lloc, representa un node d'un graf dirigit codificat com a llista d'adjacència.[15, 5] Els mètodes d'aquest objecte són només de consulta i modificació. Els atributs, en canvi, són molt interessants i importants per classificar els llocs.

A continuació, es fa una llista d'aquests atributs:

- **id:** identificador de l'etiqueta.
- **enabled:** determina si l'etiqueta està activa.
- **cih:** enter que codifica el tipus d'etiqueta (categoria, interès i/o hipervincle) mitjançant un sistema de *flags*.
- **name:** nom de l'etiqueta.
- **mySons:** codifica les arestes del graf, és a dir, les relacions entre etiquetes, i genera una nova taula relacional a la base de dades. Aquest atribut representa una llista d'etiquetes filles que pengen d'aquesta (jerarquia).
- **users:** (no s'utilitza) relació amb l'usuari. Aquest atribut només s'utilitzarà en la versió final de l'aplicació de l'empresa.
- **interestPage:** (no s'utilitza) punter a un objecte amb dades més detallades sobre el *tag*.

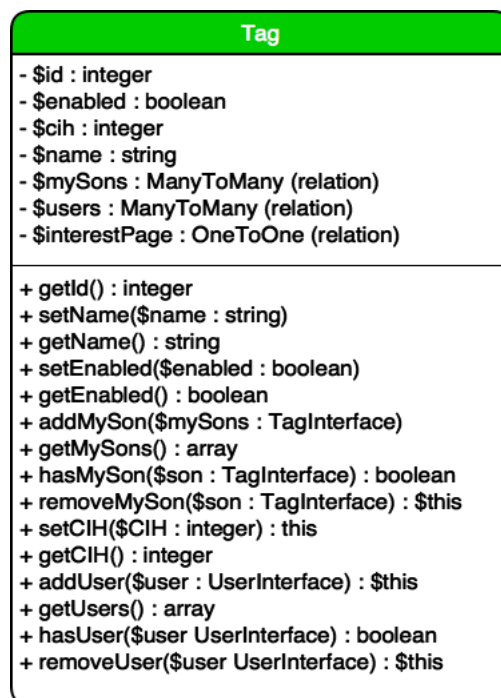


Figura 5.1: Model del *tag* del servidor

Pel que fa al *Handler* del *tag*, el que resulta interessant són els mètodes. Aquest objecte és l'encarregat de gestionar les peticions i accedir a la base de dades per seleccionar la informació de resposta. En concret, el mètode *getHierarchy* és molt important per obtenir tota la descendència d'una etiqueta. Per la seva importància, més endavant s'explica aquest algorisme amb més detall.

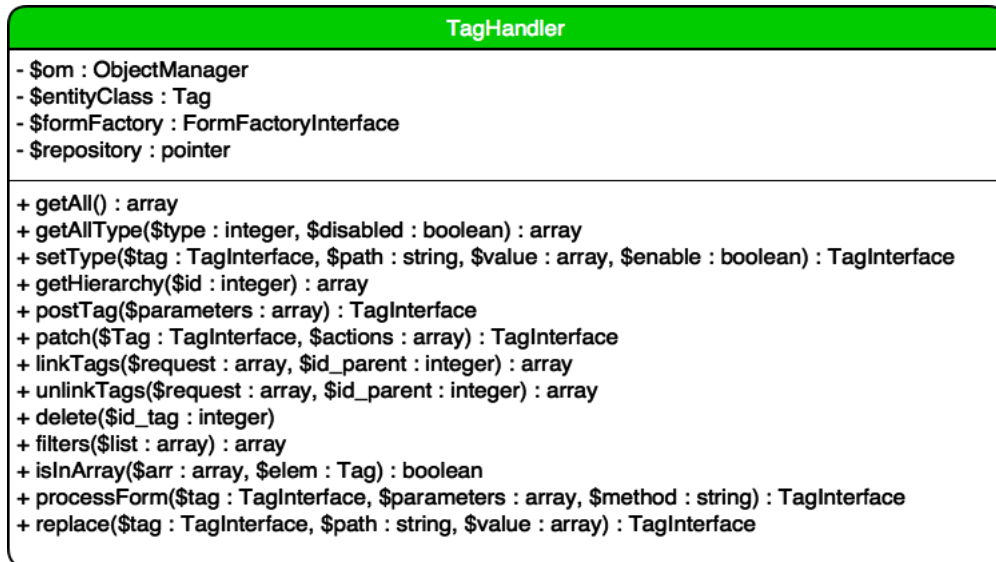


Figura 5.2: Model del *Handler* del *tag* del servidor

Un cop programat l'objecte classificador —l'etiqueta—, ja es pot presentar l'objecte classificat, és a dir, el lloc. Un lloc es pot veure com un conjunt de dades que en descriuen les característiques i unes coordenades que en determinen la ubicació. Aquest objecte és, de fet, només una part d'aquest lloc: les coordenades. La part que conté la informació més detallada és l'objecte *Page*, el qual es troba encara en desenvolupament. Tot i això, per als objectius d'aquest projecte, amb la informació de les coordenades (i amb les etiquetes) n'hi haurà prou.

De nou, com que es tracta d'un objecte abstracte per mapejar la base de dades, només s'analitzen els atributs.

- **id:** identificador del lloc.
- **enabled:** determina si el lloc està actiu i és visible.
- **name:** nom curt del lloc.
- **icon:** enter que codifica l'icona (o el color) del que es mostra en la vista de navegació.
- **coordinateLongitude:** coordenada longitud.
- **coordinateLatitude:** coordenada latitud.
- **altitude:** coordenada altitud del centre o del punt més alt del lloc.



- **floor:** coordenada altitud del terra on es troba el lloc.
- **priority:** atribut representat en coma flotant que codifica la prioritat (o importància) del lloc. Per defecte, tots els llocs tenen prioritat 1. Una prioritat entre 0 i 1 indica una prioritat per sota i una prioritat més gran que 1 indica que és un lloc destacat.
- **distance:** atribut virtual que indica la distància lineal en metres a què es troba el lloc des de la posició de l'usuari en el moment de realitzar la consulta.
- **tags:** codifica la llista d'etiquetes que descriuen el tipus de lloc que és i genera una nova taula relacional a la base de dades.
- **page\_id:** (no s'utilitza) punter a l'objecte amb les dades detallades del lloc.

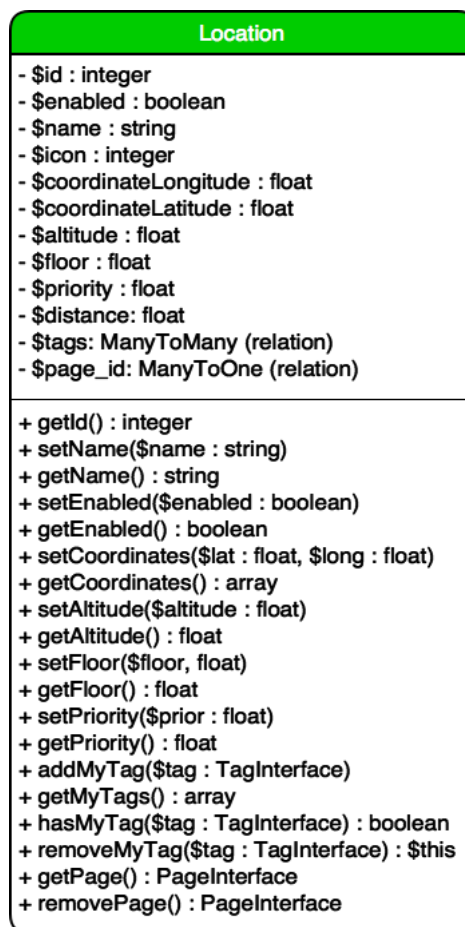


Figura 5.3: Model del *location* del servidor

El *Handler* d'aquest objecte conté mètodes per filtrar i ordenar la informació que retorna de la base de dades en funció dels paràmetres que envia l'usuari (posició i distància radial). Els mètodes per calcular les distàncies són els mateixos que s'utilitzen en el *front-end* per refrescar la informació quan l'usuari es mou. Aquests algorismes utilitzen l'aproximació de terra plana a la rodalia de la posició de l'usuari (distàncies menors a 10 km). Més endavant es detallen aquests algorismes i aproximacions.

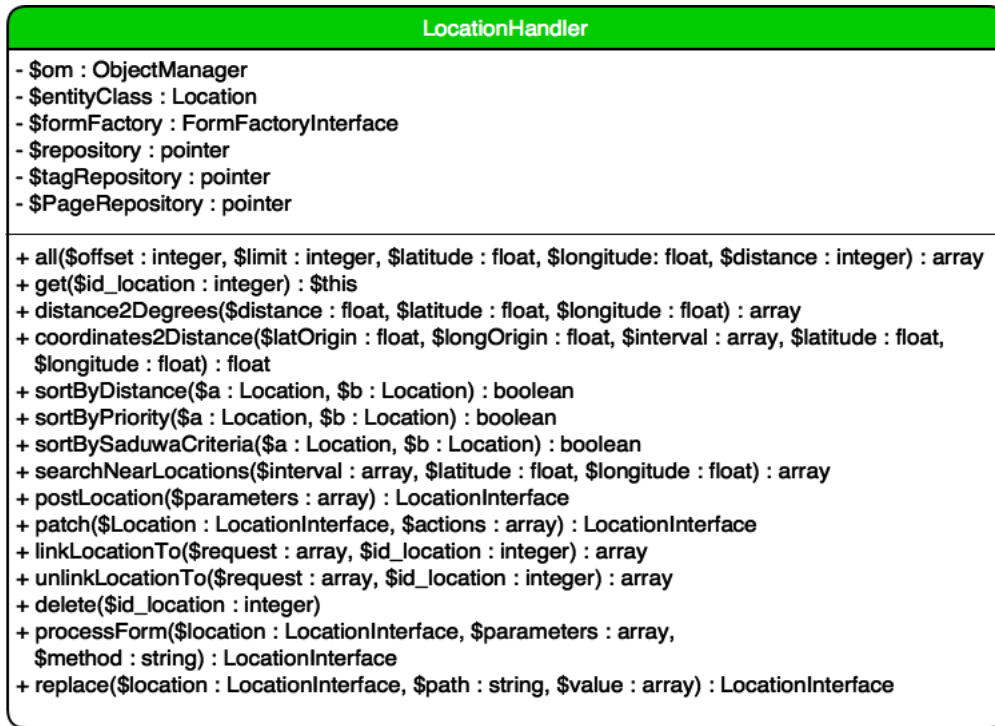


Figura 5.4: Model del *Handler* del *location* del servidor

Pel que fa al sistema per ordenar les dades dels llocs, primer es realitza una preselecció dels llocs que es troben dins del quadrat subscrit a la circumferència amb centre l'usuari i radi la distància especificada. A continuació, es calculen les distàncies a què es troben els llocs i es registren els valors a les variables *distance*. Finalment, s'ordena la llista de resultats obtinguts. Primer, per distància i, després, per prioritat.

### 5.2.2 Classes dels objectes de *front-end*

Un cop descrita la part del servidor, en aquest apartat se centra l'atenció en les classes d'objecte bàsiques que manipula l'aplicació del client. Per tal de relacionar l'aplicació amb la base de dades, els primers objectes que cal implementar són els que representen les mateixes entitats del servidor. Com en l'apartat anterior, la primera entitat presentada és la representant de les etiquetes per classificar els llocs, el *Tag*.

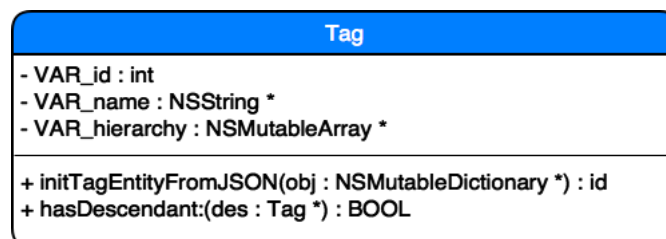


Figura 5.5: Model del *tag*

Tant l'atribut “nom” com l'atribut “identificador” són els mateixos que presenta la versió del servidor. L'atribut *VAR\_hierarchy*, en canvi, és exclusiu de la versió del client i es correspon amb un vector de *Tags* descendents (no només fills directes). Aquesta classe presenta un mètode constructor que inicialitza la classe a partir d'un *NSMutableDictionary* generat a partir del JSON que retorna el servidor. També disposa d'un mètode que determina si un *Tag* en concret es troba a la llista de descendents.

El següent objecte model és el lloc, que també conté molts atributs en comú amb el seu homònim de PHP. La latitud i la longitud s'han unit en un sol atribut de tipus *CLLocationCoordinate2D*, propi d'iOS. En aquest objecte, s'han inclòs dos tipus de constructor: un de bàsic, per inicialitzar llocs des de la pròpia aplicació, i un altre molt semblant al de l'objecte *Tag* que utilitza el JSON del servidor. Com que un dels atributs d'aquest objecte és la llista d'etiquetes del lloc, el mateix constructor s'encarrega de crear instàncies d'objectes *Tag* a partir de les dades del servidor i omple la llista. A continuació, es mostra l'esquema d'aquesta classe.

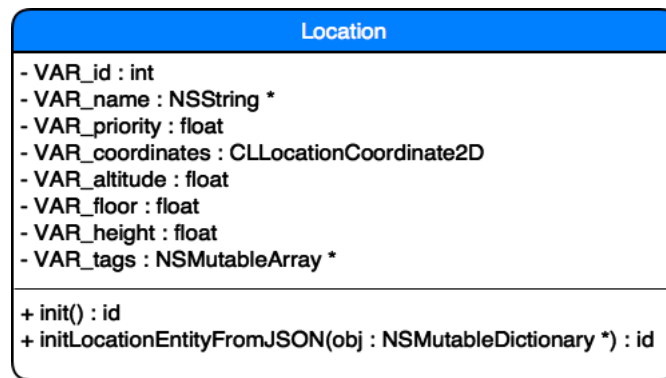


Figura 5.6: Model del lloc

En el costat del client s'han utilitzat, a més dels models de dades, altres tipus d'objecte com els gestors del *hardware* (sensors), els gestors de comunicació amb el servidor o els controladors de vista. A continuació, es presenten alguns d'aquests objectes començant per un dels més importants: el que s'encarrega de comunicar-se amb el servidor.

En efecte, el *ServerHandler* encapsula totes les funcions d'accés i comunicació amb el servidor. Així, la resta d'objectes i controladors es poden despreocupar d'aquesta part. Quan un objecte vol obtenir dades del servidor realitza una petició (envia un missatge) a aquest objecte i es posa en espera. Quan aquest objecte obté resposta, actualitza les dades dels seus atributs i avisa la resta d'objectes mitjançant el protocol *NSNotification*.

El constructor d'aquest objecte, el mètode *init()*, realitza un conjunt de peticions automàtiques de manera consecutiva per obtenir les dades del servidor.<sup>1</sup> Tot i que en aquesta aplicació no és necessària una autenticació per part de l'usuari, també s'han inclòs els mètodes i atributs necessaris per poder realitzar una petició del *token* quan formi part de la xarxa social.

<sup>1</sup> Per ordre, les etiquetes, la llista de jerarquies i la llista de llocs propers a l'usuari (10 km).

A continuació, es mostra l'esquema d'aquesta classe:

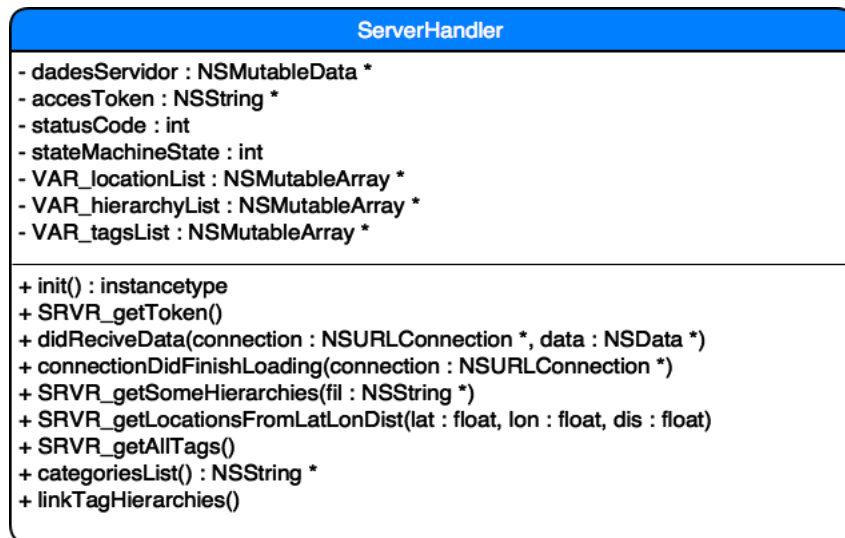


Figura 5.7: Model de l'objecte que es comunica amb el servidor

On els mètodes començats per “SRVR” es corresponen amb les peticions HTTP al servidor i els atributs començats per “VAR”, amb les llistes d'objectes obtinguts.

La següent classe representa el gestor dels sensors que s'ha batejat com “Brúixola” (*Compass*), tot i que obté informació sobre la posició GPS i l'acceleròmetre a més de la posició del nord magnètic. Aquestes dades, obtingudes pels gestors de *hardware*,<sup>2</sup> s'emmagatzemen als atributs següents:

- **VAR\_myLocation:** coordenades GPS del lloc (inclou l'altitud).
- **VAR\_accelX:** projecció del vector unitari “gravetat”<sup>3</sup> sobre l'eix *x* del dispositiu.[25]
- **VAR\_accelY:** projecció del vector unitari “gravetat” sobre l'eix *y* del dispositiu.
- **VAR\_accelZ:** projecció del vector unitari “gravetat” sobre l'eix *z* del dispositiu.
- **VAR\_north:** angle a què es troba apuntant el mòbil respecte del nord magnètic.
- **VAR\_rho:** angle a què es troba apuntant el mòbil respecte de l'eix vertical.

Els gestors de *hardware* dels objectes d'aquesta classe realitzen peticions constants per actualitzar les dades obtingudes dels sensors i, per aquest motiu, aquest objecte és un gran consumidor de bateria. Tot i això, es podria reduir la freqüència de les peticions per millorar l'autonomia del dispositiu. De fet, s'han realitzat algunes proves amb diferents valors per obtenir un bon compromís de precisió/consum.

<sup>2</sup> Els encarregats de gestionar el *hardware* són els objectes de tipus *CMMotionManager* i *CLLocationManager*.

<sup>3</sup> Representa el vector de mòdul 1 que apunta al centre de la Terra. En l'aproximació de Terra plana, aquest vector és perpendicular al pla del terra.

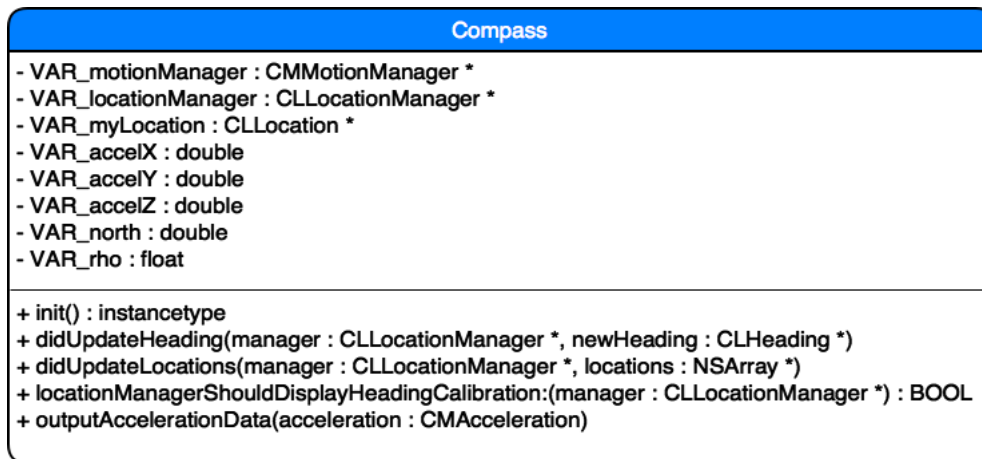


Figura 5.8: Model de la brúixola

L'última classe d'aquest apartat s'anomena *Maths* i s'utilitza exclusivament per realitzar els càlculs geomètrics necessaris per implementar la realitat augmentada (és més una llibreria que un objecte). L'únic atribut que té és un objecte de tipus brúixola que s'utilitza per obtenir les dades dels sensors (necessàries per realitzar certs càlculs).

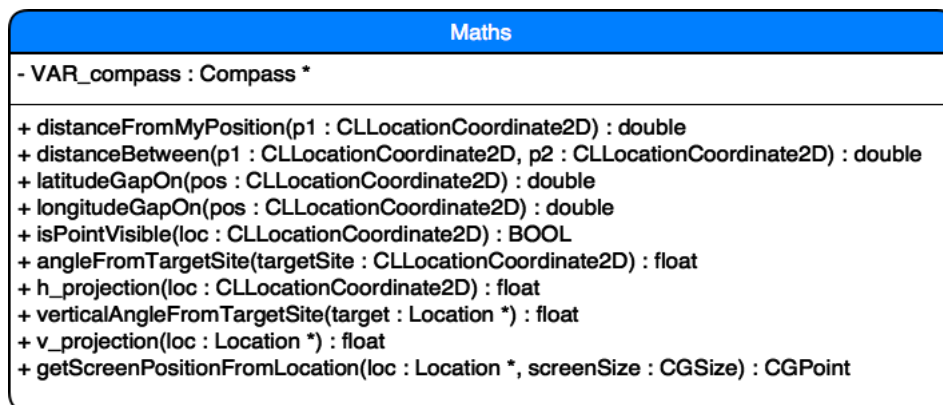


Figura 5.9: Model de l'objecte que realitza els càlculs geomètrics

Com que els mètodes d'aquesta classe són interessants, s'explicaran amb més detall en l'apartat d'algorismes.

### 5.2.3 Controladors de vista

A continuació, es presenten les quatre classes, corresponents als controladors de vista bàsics de l'aplicació. Per a cada una d'aquestes classes s'ha creat un fitxer *nib* per dibuixar el *layout* mitjançant l'eina *Interface Builder*. Els objectes creats amb *Interface Builder* que s'han volgut referenciar estan indicats amb l'etiqueta *IBOutlet*. A més, alguns mètodes i atributs que hereten de la superclasse *UIViewController* s'han sobreescrit per mantenir el control en la creació i en la càrrega de les vistes. Més concretament, d'una banda, s'han sobreescrit els mètodes

*ViewDidLoad* i *ViewWillAppear*, de l'altra, s'ha utilitzat l'atribut *view* per accedir als objectes i les capes de la vista.

El primer controlador que es presenta és el de la pantalla principal, la vista de realitat augmentada que mostra la finestra de la càmera i les etiquetes dels llocs. Aquest controlador és creat i inicialitzat per l'objecte *AppDelegate* quan s'inicia l'aplicació i pot accedir a la resta d'objectes. Aquest objecte és també l'encarregat d'inicialitzar la càmera, la brúixola i l'objecte *ServerHandler* que es connecta al servidor. Quan s'obté la llista de llocs es guarda a l'*array mySites*. Per a cada objecte *Location* d'aquest *array* es crea un objecte de classe *UIView* que representa l'etiqueta del lloc i, després d'afegir l'etiqueta com a subvista de l'atribut *view*, es guarda a l'*array myLabels* amb el mateix ordre que els objectes de *mySites*. Mitjançant un temporitzador, es crida el mètode *updateData* cada 50 ms (20 cops per segon), el qual s'encarrega de posicionar cada etiqueta al lloc que li correspon a la pantalla (en funció de l'orientació del dispositiu). A continuació, es mostra la llista de mètodes i atributs d'aquest controlador.

ViewController
<ul style="list-style-type: none"> <li>- servidor : ServerHandler *</li> <li>- indicador : IBOutlet UIActivityIndicator *</li> <li>- serverReady : BOOL</li> <li>- filterVC : FilterViewController *</li> <li>- filterContainer : IBOutlet UIView *</li> <li>- filterHeader : IBOutlet UIView *</li> <li>- cameraSession : AVCaptureSession *</li> <li>- cameraLayer : AVCaptureVideoPreviewLayer *</li> <li>- cameraV : IBOutlet UIView *</li> <li>- compass : Compass *</li> <li>- math : Math *</li> <li>- LBL_numFilters : IBOutlet UILabel *</li> <li>- LBL_numSites : IBOutlet UILabel *</li> <li>- configVC : ConfigViewController *</li> <li>- mySites : NSMutableArray *</li> <li>- myLabels : NSMutableArray *</li> <li>- detailVC : DetailViewController *</li> </ul>
<ul style="list-style-type: none"> <li>+ BTN_openFilters(sender : id) : IBAction</li> <li>+ performAllFilters()</li> <li>+ startCamera()</li> <li>+ createAllLabels()</li> <li>+ deleteAllLabels()</li> <li>+ updateData()</li> <li>+ opacityAtDistance(d : float) : float</li> <li>+ getSiteLayoutFromLocation(loc : Location *, pos : int) : UIView *</li> <li>+ refreshLabelsInformation()</li> <li>+ BTN_refresh(sender : id) : IBAction</li> <li>+ BTN_clearLocation(sender : id) : IBAction</li> <li>+ BTN_openDetail(sender : id) : IBAction</li> <li>+ NOTE_serverReady(note : NSNotification *)</li> <li>+ NOTE_finishUpload(note : NSNotification *)</li> <li>+ NOTE_finishFiltering(note : NSNotification *)</li> <li>+ BTN_openConfig(sender : id) : IBAction</li> </ul>

Figura 5.10: Controlador de la vista principal

On els mètodes començats per "BTN" representen accions vinculades als botons de la vista i els botons començats per "NOTE", els mètodes que s'executen quan es reben notifikacions d'altres objectes (el servidor i el controlador dels filtres).

El següent controlador, *FilterViewController*, és el responsable del menú per filtrar. Aquest controlador és inicialitzat pel controlador anterior i s'encarrega de gestionar tot el sistema de filtres. Per aquest motiu, és delegat de les classes *UITableView* i *UISearchBar*. El sistema per seleccionar els filtres s'ha treballat amb cura per obtenir un resultat que sigui còmode i intuïtiu per a l'usuari. Es poden seleccionar els filtres amb un simple toc a la llista desplegable (taula). Els elements d'aquesta taula es poden filtrar fàcilment amb el camp de cerca *UISearchBar*. Els filtres seleccionats es poden eliminar amb un sol toc i passen a la llista de filtres recents. Quan se surt del menú, s'apliquen automàticament tots els filtres seleccionats sobre les etiquetes dels llocs (s'amagen sense realitzar noves peticions al servidor) i s'indica el nombre de llocs que queden a la barra superior de la vista principal. A continuació, es mostren les dades de la classe.



Figura 5.11: Controlador del panell de filtres

A més dels mètodes associats a la taula i al camp de cerca, també són interessants els algorismes que s'encarregen de redibuixar els filtres aplicats i els filtres recents en les vistes contenidor. Això es correspon amb els mètodes començats per *refresh* i els atributs acabats per *ContainerV*.

El tercer controlador és el de la vista bàsica de detall d'un lloc: *DetailViewController*. Quan es prem un dels llocs que apareixen per pantalla s'accedeix a una vista amb la informació bàsica del lloc. Com que no s'ha implementat l'objecte *Page* que conté la informació detallada i les imatges, en aquesta vista només es mostren les coordenades i la llista d'etiquetes del lloc.

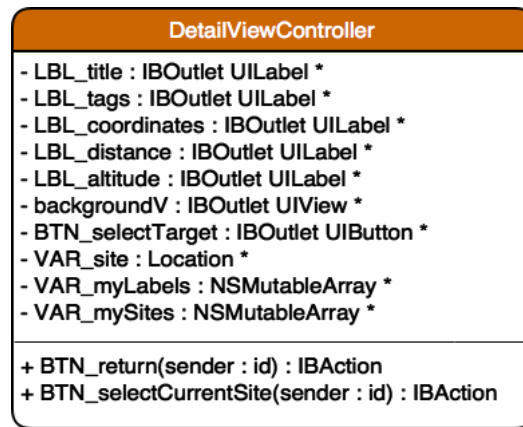


Figura 5.12: Controlador de la vista detall

L'únic mètode interessant d'aquesta classe és el que es correspon amb la funcionalitat extra “marcar el lloc com a objectiu”. Si es prem aquest botó, s'amagen totes les etiquetes dels llocs (llevat d'aquesta) per facilitar la navegació.

Finalment, l'últim controlador es correspon amb la vista extra de configuració. De moment, aquest controlador només presenta un selector de tipus *UISwitch* que indica al sistema si ha d'apagar la càmera, o no, quan s'obre el menú de filtres. Tot i això, aquesta estructura es pot escalar molt fàcilment per afegir altres opcions de configuració en el futur.

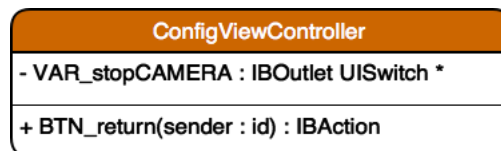


Figura 5.13: Controlador de la vista de configuració

Amb aquests controladors, ja s'han presentat totes les classes amb què treballa l'aplicació. El següent pas és presentar, d'una banda, el diagrama de classes que relaciona totes aquestes entitats i, de l'altra, el model relacional de la base de dades.

## 5.2.4 Diagrama de classes

En aquest apartat, es presenta el diagrama de classes dels objectes del costat del client. Aquest diagrama ofereix una visió global de tots els objectes i de les relacions que existeixen entre ells. Per simplificar i compactificar aquest esquema, s'han obviat tots els atributs que no estan directament involucrats en les relacions. Pel mateix motiu, s'han eliminat els mètodes de totes les classes. Per a les relacions s'han utilitzat fletxes que comencen en l'objecte que posseeix el punter i acaben en l'objecte “apuntat”. Quan un objecte conté un *array* d'objectes d'una altra classe, s'ha utilitzat la notació estàndard (1...\*).



En la figura següent es mostra aquest diagrama que relaciona les classes del *front-end*.

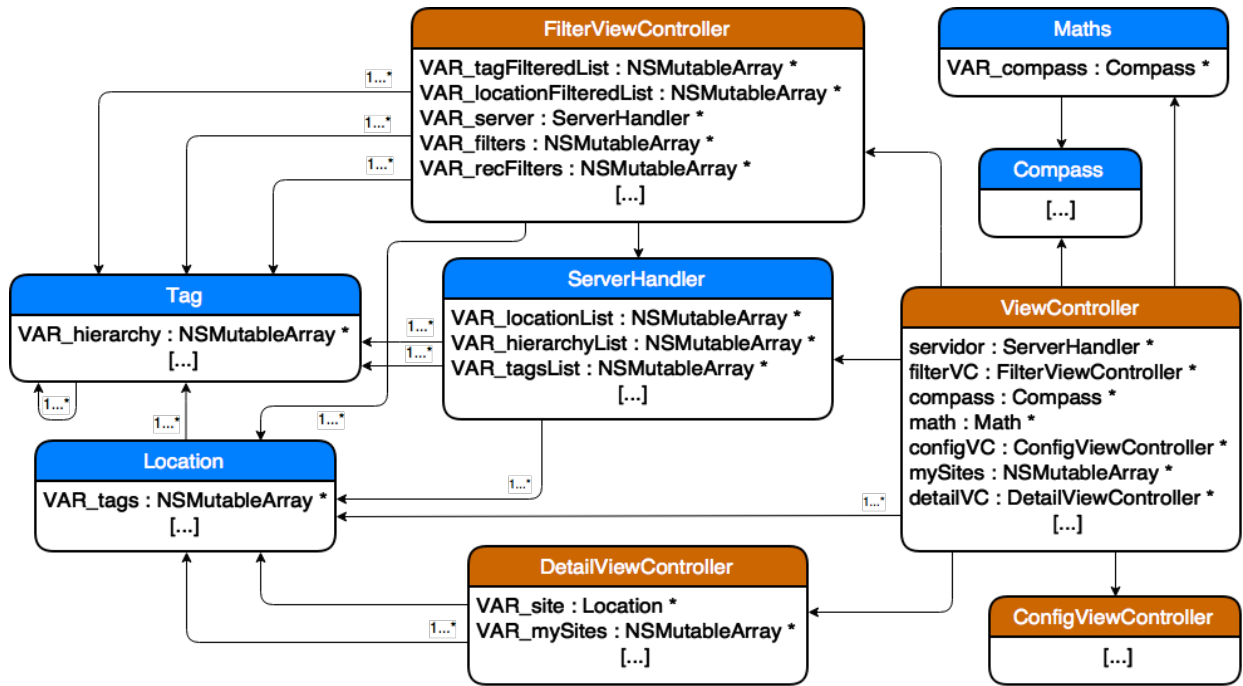


Figura 5.14: Diagrama de classes

### 5.2.5 Model relacional de la base de dades

Pel que fa a l'estructura de la base de dades, com que en aquest projecte només s'han inclòs dos mòduls *Symfony*, a diferència del diagrama de classes, es pot representar en un diagrama relacional complet sense obviar cap dels camps. Així, a partir dels models presentats del *Location* i el *Tag*, es generen quatre taules relacionades mitjançant els identificadors (*id*) d'aquestes entitats. Dues de les taules representen les entitats en si; la taula *LocationTags*, la llista d'etiquetes que classifiquen al lloc, i la taula *TagHierarchy*, la relació de descendència entre les etiquetes.

A continuació, es mostra el model que relaciona aquestes quatre taules.

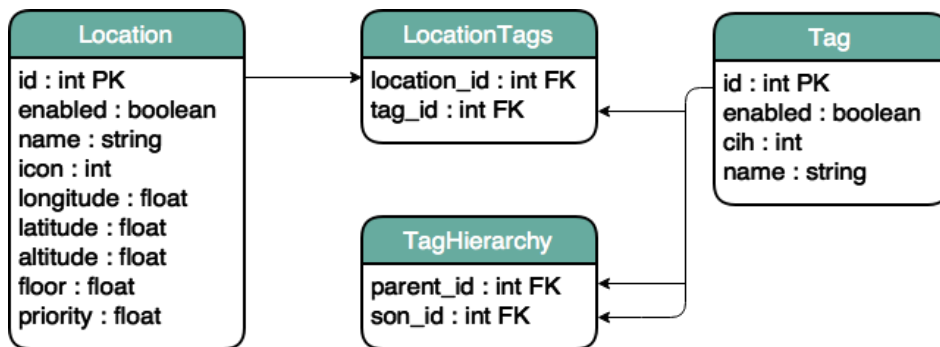


Figura 5.15: Model relacional de la base de dades

Aquestes taules es poden consultar i modificar mitjançant el software *PHPmyAdmin*.

## 5.3 Algorismes destacats

En aquest apartat es presenten i s'expliquen alguns dels algorismes més importants que són necessaris per entendre el funcionament de l'aplicació. Tot i això, si queden dubtes sobre la implementació d'algun dels aspectes de l'aplicació, es poden consultar els fitxers dels projectes, ja que el codi s'ha comentat profusament<sup>4</sup> per facilitar-ne la lectura i agilitzar la tasca de desenvolupament de futures actualitzacions.

### 5.3.1 Mètode per obtenir les jerarquies

Tot i que aquest mètode no és gaire complicat, és interessant per entendre com es guarda la informació de les relacions entre els *Tags* i quins errors poden aparèixer si no s'introdueix informació coherent. Per començar, es mostra la llista provisional d'etiquetes que s'han introduït al servidor (mitjançant la consola d'administració) per realitzar les proves.

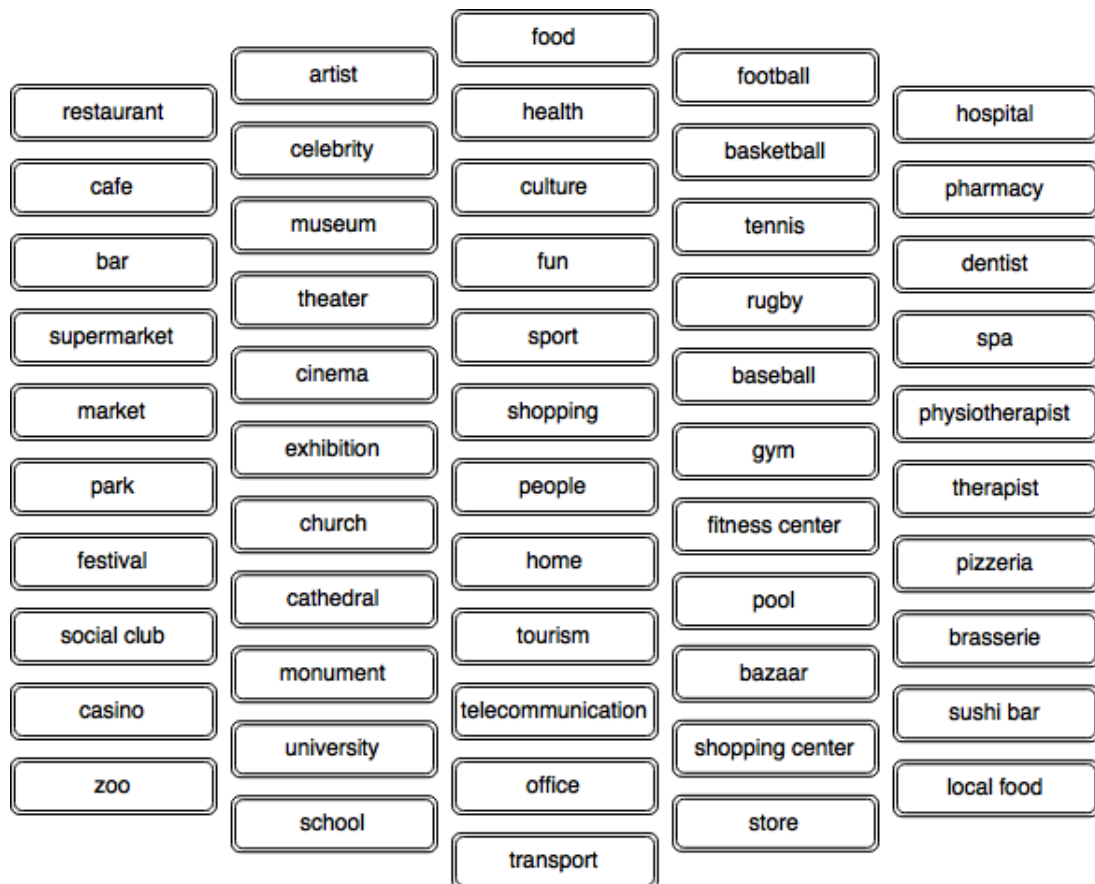


Figura 5.16: Llista de *tags* provisional

En aquesta llista de 54 etiquetes, tot i que no és gaire exhaustiva, es poden trobar tant termes genèrics (*food, health, sport...*) com termes més específics (*brasserie, dentist, therapist...*).

<sup>4</sup> El codi de la part de servidor s'ha comentat en anglès per mantenir la coherència amb la resta de mòduls que hi ha al servidor. Els comentaris del codi de la part del clients són en català.

És clar que els termes específics són molt més descriptius que els termes genèrics, però, si un usuari té gana, és probable que decideixi buscar una braseria? No. De fet, el més probable és que l'usuari realitzi una cerca mitjançant l'etiqueta *food*, que és molt més genèrica. Aleshores, sembla raonable incloure aquesta etiqueta a un lloc que sigui braseria, però si s'inclou a una braseria, s'hauria d'incloure a totes les altres braseries, pizzeries, hamburgueseries, marisqueries... de fet, a tots els restaurants. Però és evident que en una braseria (o en qualsevol altre restaurant) hi ha menjar. El terme *brasserie* ja hauria d'incloure aquesta connotació. No cal omplir la base de dades amb etiquetes redundants que no fan res més que ocupar recursos. Aquesta és la idea que hi ha darrere del model de jerarquies dels *Tags*. A la figura següent, es mostren dos exemples d'etiquetes amb les etiquetes filles.



Figura 5.17: Exemples de jerarquies

On, per exemple, l'etiqueta *hospital* és filla (cas particular) de l'etiqueta *health*. Aquesta relació estableix que si un usuari busca llocs relacionats amb la salut, s'han de mostrar els hospitals. Tot i que a la taula de la base de dades només es guarden els fills directes (primer nivell de jerarquia), aquest esquema permet, conceptualment, crear jerarquies de més nivells. Per exemple, una jerarquia com la de l'etiqueta menjar, que es mostra a continuació.

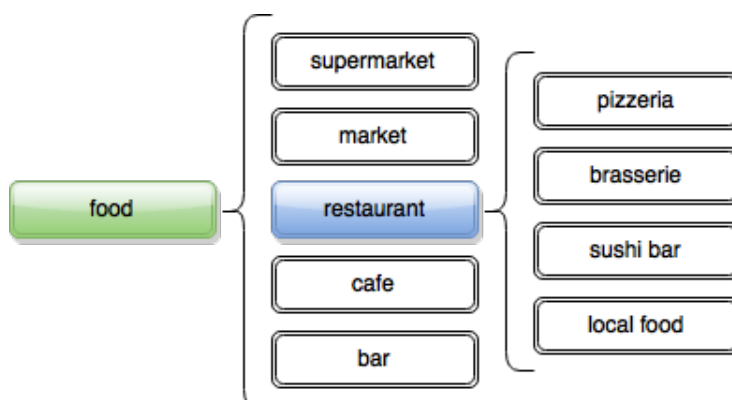


Figura 5.18: Exemple de jerarquia de dos nivells

Segons aquesta estructura, *brasserie* no és fill directe de *food* però sí n'és un descendent (nét), ja que és fill de *restaurant*. Si es pretén que les braseries, un tipus concret de restaurant, apareguin

quan l'usuari busca *food*, cal obtenir tota la jerarquia de l'etiqueta o, el que és el mateix, la llista de descendents (incloent-lo a ell mateix). Això és precisament el que realitza el mètode *getHierarchy*, que es mostra a continuació.

```

public function getHierarchy($id)
{
    //We get the first one
    $tags = array($this->repository->findOneById($id));
    //Current object in study (number of checked objects)
    $checked = 0;
    //Total number of objects that should be checked
    $counter = 1;

    //While there is an object that has not been checked yet
    while($counter > $checked){
        //We take current object's sons
        $aux = $tags[$checked]->getMySons();
        //For each son
        foreach($aux as $son){
            //If the son is a new object, we put it into array
            if(!$this->isInArray($tags, $son)){
                array_push($tags, $son);
                $counter++;
            }
        }
        //Object checked
        $checked++;
    }
    return $tags;
}

```

Figura 5.19: Algorisme per obtenir les jerarquies

L'únic paràmetre que s'envia a aquest mètode és l'identificador de l'etiqueta amb nivell més alt. El vector d'etiquetes, *\$tags*, s'inicialitza amb l'objecte obtingut a partir de l'identificador. A continuació, s'inicialitzen dues variables per comptar l'estat de la cerca. La variable *\$counter* s'inicialitza a 1 i compta el nombre d'etiquetes que s'han introduït al vector. La variable *\$checked* s'inicialitza a 0 i compta el nombre d'etiquetes de les quals s'han comprovat els fills. En cada iteració, s'agafa una etiqueta que no ha estat comprovada i se'n capturen els fills. Per a cada fill, es comprova si encara no s'ha inclòs al vector. En cas afirmatiu, s'afegeix al final i s'incrementa el comptador d'etiquetes. Quan s'han comprovat tots els fills, s'incrementa el comptador de comprovats. Aquest algorisme sempre acaba<sup>5</sup> i genera una llista robusta a repeticions i cicles. Per exemple, una jerarquia com la que es mostra a la figura 5.20 conté dos cicles. En efecte, els fills de *local food*, *pizzeria* i *food*, són, respectivament, germà i avi de *local food*. Com que en el moment de comprovar els fills ja es troben al vector de *\$tags*, l'algorisme els ignora (a efectes pràctics, és com si *local food* no tingués fills). Això vol dir que es poden anar creant relacions sense vigilar que a la jerarquia no es formin cicles? Si volem un sistema de filtratge coherent, la resposta és no. Continuant amb l'exemple anterior, què passa si un usuari està interessat en un restaurant de menjar autòcton? Si l'usuari selecciona l'etiqueta corresponent, *local food*, com que *food* n'és un fill, automàticament s'afegeix tota la jerarquia que es mostra a la figura 5.20. Això inclou pizzeries, bars, supermercats, etc., cosa que queda molt lluny del comportament desitjat per l'usuari.

<sup>5</sup> En el pitjor dels casos, aquest algorisme retorna tots els tags de la base de dades.



és molt petit. En canvi, a prop de l'equador, el perímetre dels paral·lels i els meridians és aproximadament igual. El mètode següent calcula aquesta variació en funció de la latitud a la qual es troba l'usuari en forma de factor de conversió local.

```
// Mètode per obtenir el factor de conversió per la longitud (aproximació plana)
- (double) longitudeGap0n:(CLLocationCoordinate2D)pos
{
    double lonGap = 180*sqrt(1-ECC_sq*(sin(pos.latitude*Deg2rad) * sin(pos.latitude*Deg2rad)));
    lonGap = lonGap/(M_PI*EARTH_radius*cos(pos.latitude*Deg2rad));

    return fabs (lonGap);
}
```

Figura 5.22: Factor de conversió per a l'aproximació plana de la longitud

Amb aquests dos mètodes ja és possible representar els objectes propers a l'usuari mitjançant un sistema de coordenades cartesià. Per exemple, es pot prendre el meridià en el qual es troba l'usuari (en sentit nord) per l'eix *abscisses* i el paral·lel en què es troba l'usuari (en sentit oest) com l'eix d'ordenades. El següent mètode s'encarrega de calcular l'angle respecte de l'eix *x* d'aquest nou sistema de coordenades a la qual es troben els objectes.

```
// Mètode que retorna l'angle (respecte el nord) a la que es es troba el targetSite
- (float) angleFromTargetSite:(CLLocationCoordinate2D)targetSite
{
    // Es calculen les coordenades cartesianes de l'aproximació plana
    float x = (targetSite.latitude - VAR_compass.VAR_myLocation.coordinate.latitude);
    float y = (VAR_compass.VAR_myLocation.coordinate.longitude - targetSite.longitude);

    // S'apliquen les dilatacions
    x = x/[self latitudeGap0n:targetSite];
    y = y/[self longitudeGap0n:targetSite];

    // La primera coordenada nul·la indica un angle de 90 o 270 graus
    if (x == 0) {
        return (y > 0) ? 90:270;
    } else {
        // En cas contrari, se'n calcula l'arc tangent
        float temp = Rad2deg*(atan2f(y, x));
        if (temp < 0){
            return temp + 360;
        } else {
            return temp;
        }
    }
}
```

Figura 5.23: Angle relatiu de la direcció horitzontal a la qual es troba el lloc

Com que la brúixola ofereix l'angle al qual apunta el mòbil de l'usuari respecte del nord magnètic, per saber la posició relativa dels objectes respecte del centre de la càmera només cal restar el resultat obtingut pel mètode anterior. Coneixent l'angle horitzontal d'obertura de la càmera del terminal es poden projectar les línies que uneixen l'usuari amb els llocs sobre el pla de l'horitzó visible a la pantalla del mòbil. El mètode següent realitza aquesta funció.

```

// Mètode que calcula el ràtio de projecció (de -1..1) a la que es troba la coordenada X respecte a
// l'obertura de la càmera del dispositiu
- (float) h_projection:(CLLocationCoordinate2D) loc
{
    float angle = [self angleFromTargetSite:loc];
    float myAngle = 360.0-VAR_compass.VAR_north;

    return (angle - myAngle) / (HorApperDeg_5s / 2.0);
}

```

Figura 5.24: Projecció horitzontal de la posició del lloc

Si un objecte obté una projecció igual a 0, es troba al centre de la pantalla i, si obté una projecció amb mòdul 1, en un dels extrems. Així, tots els valors de projecció del rang [-1,1] representen llocs visibles per pantalla.

El mateix principi s'utilitza per a la secció vertical de la pantalla a partir de l'altitud relativa entre l'usuari i el lloc. En primer lloc, per saber l'angle d'inclinació del mòbil, s'utilitza l'atribut *rho*, obtingut de l'objecte brúixola que es calcula a partir de les projeccions del vector gravetat.

```

float mod_ZY = sqrtf(VAR_accelY*VAR_accelY + VAR_accelZ*VAR_accelZ);
float incl = -180*asin(-VAR_accelZ/mod_ZY)/M_PI;
VAR_rho = 90 - incl;

```

Figura 5.25: Obtenció de l'angle d'inclinació a partir del vector gravetat

La coordenada *z* del sistema de referència és ortogonal al pla del terra i, per aquest motiu, en aquest cas, s'utilitzen l'obertura vertical de la càmera i l'alçada de la pantalla per determinar la projecció. Els mètodes següents són els encarregats de modificar l'alçada a la qual es troben els objectes a partir de les variables anteriors i l'atribut *rho* de la brúixola.

```

// Mètode que retorna l'angle d'elevació (respecte la teva altitud) a la que es troba el targetSite
- (float) verticalAngleFromTargetSite:(Location *)target
{
    float d = target.VAR_distance;
    float h = target.VAR_altitude;

    if (VAR_compass.VAR_myLocation.altitude != 0) {
        h = h - VAR_compass.VAR_myLocation.altitude;
    }else{
        h = h - target.VAR_floor;
    }

    // Es calcula rho
    if (d == 0) {
        return (h > 0) ? 0:180;
    } else {
        return 90 - Rad2deg*(atan2f(h, d));
    }
}

```

Figura 5.26: Angle relatiu de la direcció vertical a la qual es troba el lloc

```
// Mètode que calcula el ràtio de projecció (de -1..1) a la que es troba una coordenada Y respecte a
// l'obertura de la càmera del dispositiu
- (float) v_projection:(Location *) loc
{
    float angle = [self verticalAngleFromTargetSite:loc];
    float myAngle = VAR_compass.VAR_rho;

    return (myAngle - angle) / (VerApperDeg_5s / 2.0);
}
```

Figura 5.27: Projecció vertical de la posició del lloc

A partir de les dues projeccions, vertical i horitzontal, es poden situar els objectes a la pantalla del dispositiu. El mètode següent s'encarrega de generar els *CGPoint* pels centres dels llocs a partir dels resultats dels mètodes anteriors.

```
// Mètode que proporciona la posició d'una vista (associada a unes coordenades) dins dels marges de
// la pantalla del dispositiu
- (CGPoint) getScreenPositionFromLocation:(Location *)loc withScreenSize:(CGSize) screenSize
{
    float x_angleRatio = [self h_projection:loc.VAR_coordinates];
    float y_angleRatio = [self v_projection:loc];

    float x_p = (screenSize.width * (-x_angleRatio))/2 + screenSize.width / 2.0;
    float y_p = (screenSize.height * (-y_angleRatio))/2 + screenSize.height / 2.0;
    return CGPointMake(x_p, y_p);
}
```

Figura 5.28: Projecció de les direccions a la pantalla

Aquest conjunt de mètodes situen correctament els llocs en la mesura que les dades dels sensors són bones i només quan el mòbil està orientat verticalment (*portrait*). Una última millora que no s'ha arribat a implementar per falta de temps consisteix a compensar la inclinació lateral del mòbil mitjançant una rotació de la vista que conté les etiquetes. Amb aquesta millora ja es podrien utilitzar les altres orientacions sense problemes.

### 5.3.3 Filtres per a les dades dels sensors

Per acabar l'apartat d'algorismes, es presenta una estratègia que s'ha utilitzat per equalitzar els errors de l'acceleròmetre causats per les fluctuacions (que es pot estendre fàcilment a la resta de sensors). Aquesta estratègia consisteix a implementar un filtre passa baix mitjançant una ponderació entre les dades obtingudes i les registrades. Més concretament, cada vegada que arriba una nova dada de l'acceleròmetre, per calcular un valor més ajustat a la realitat, s'utilitza el 20% de la nova dada i el 80% de la dada anterior. Aquest sistema ofereix uns valors molt més estables i és robust als canvis bruscos causats per l'entorn; però, com a contrapartida, ofereix una resposta lleugerament més lenta que la que s'obté sense filtrar.

```
// Filtre passa baix per suavitzar les dades
VAR_accelX = VAR_accelX*0.8 + acceleration.x*0.2;
VAR_accelY = VAR_accelY*0.8 + acceleration.y*0.2;
VAR_accelZ = VAR_accelZ*0.8 + acceleration.z*0.2;
```

Figura 5.29: Filtre per a les dades de l'acceleròmetre



## CAPÍTOL

### 6

# RADIUS AUGMENTED REALITY

Aquest capítol pretén il·lustrar els resultats del desenvolupament de l'aplicació. Per aquest motiu, es presenten un conjunt de captures de pantalla que s'han anat realitzat durant el procés de testeig de l'aplicació. Aquestes proves s'han realitzat a Barcelona captial, a partir de les dades reals de 25 llocs coneguts introduïdes al servidor d'Amazon.

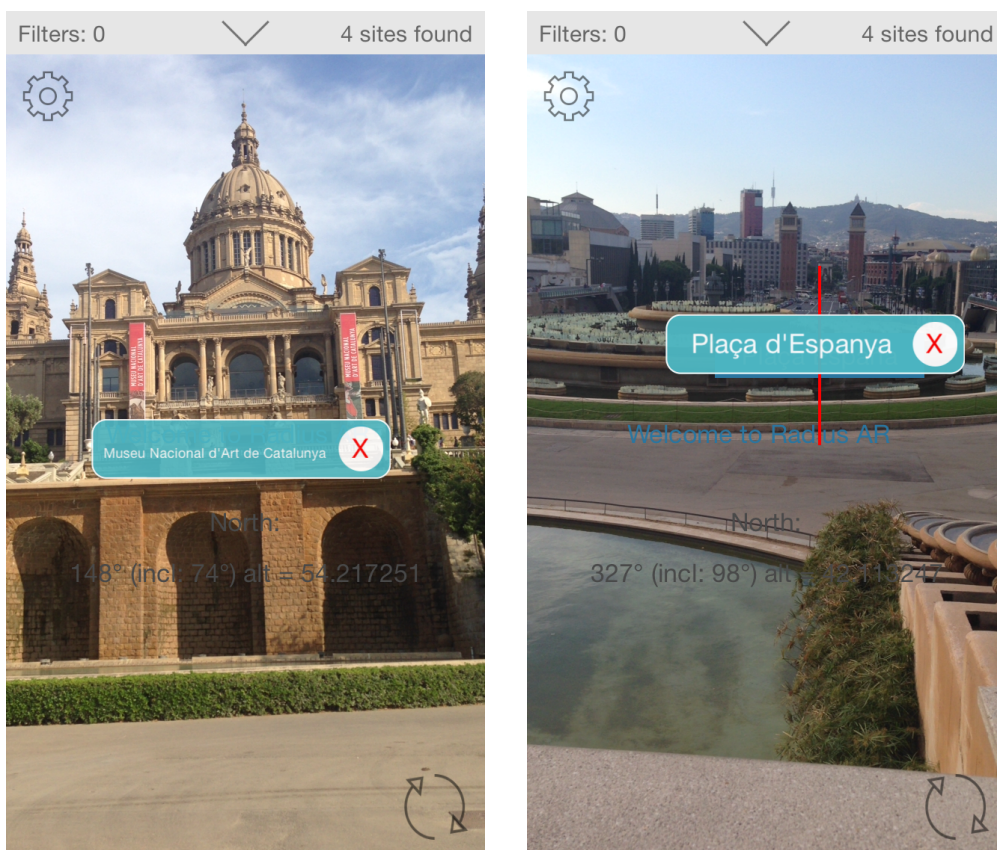


Figura 6.1: Captures. Posicionament correcte a diferents distàncies

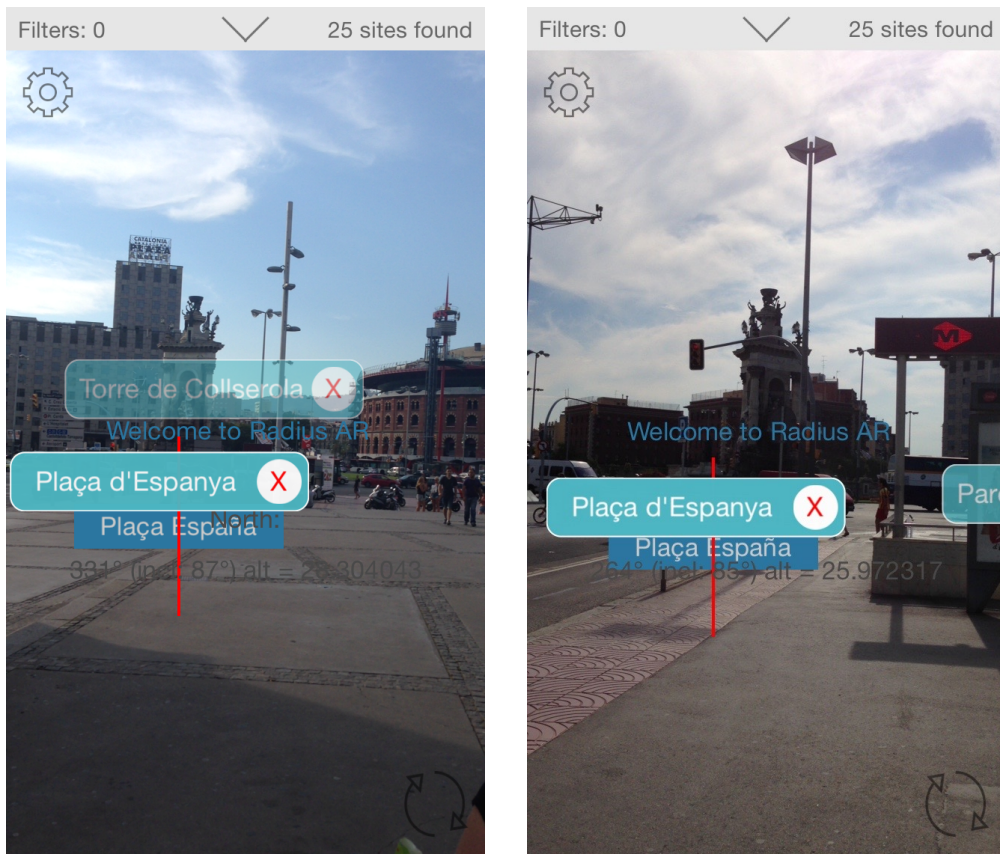


Figura 6.2: Captures. Plaça d'Espanya des de diferents angles (1)

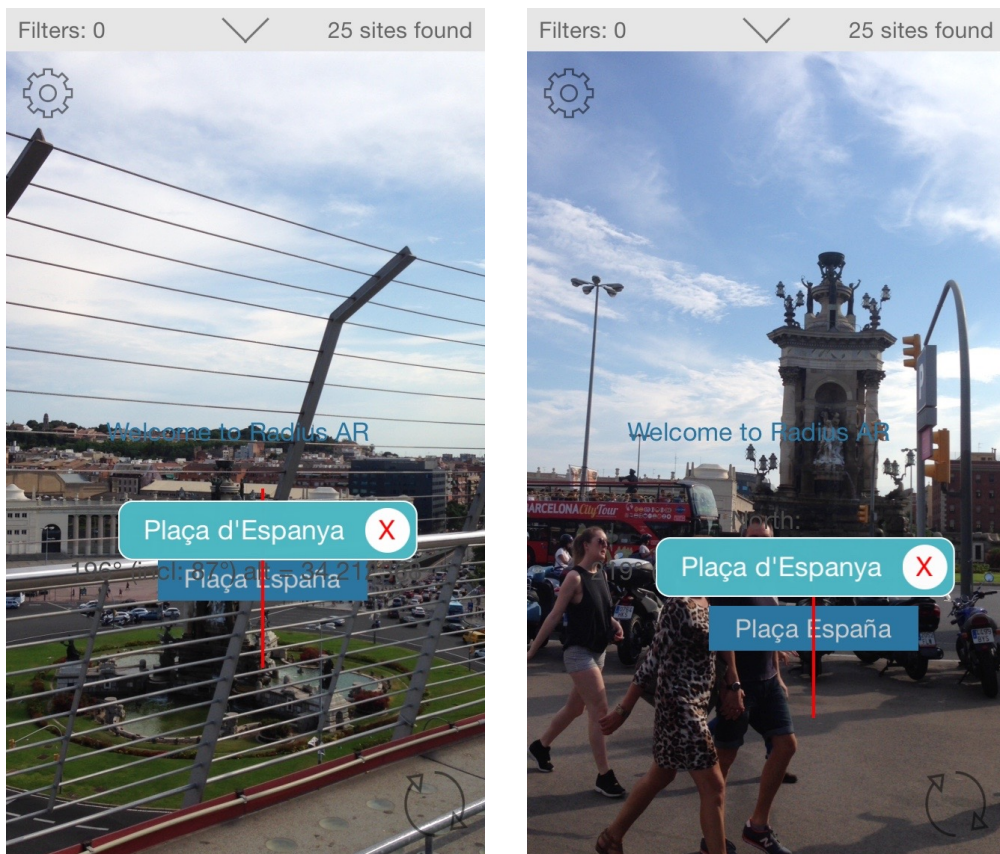


Figura 6.3: Captures. Plaça d'Espanya des de diferents angles (2)

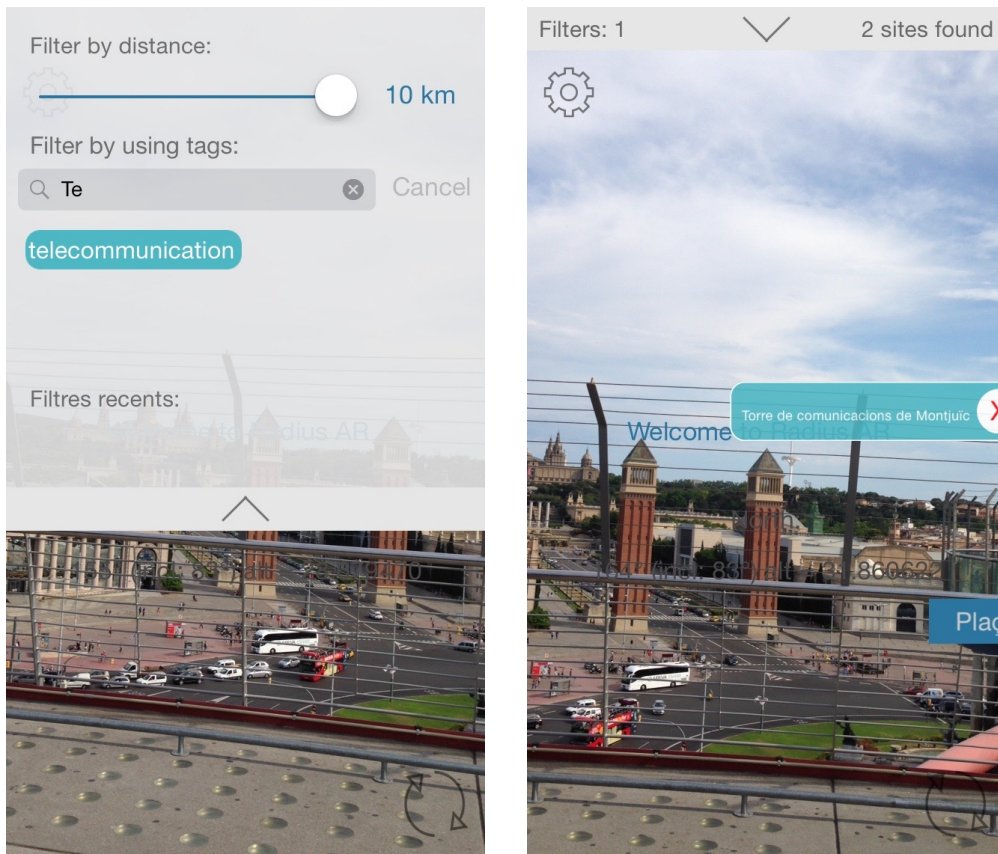


Figura 6.4: Captures. Resultat d'aplicar el filtre *telecommunication* (1)

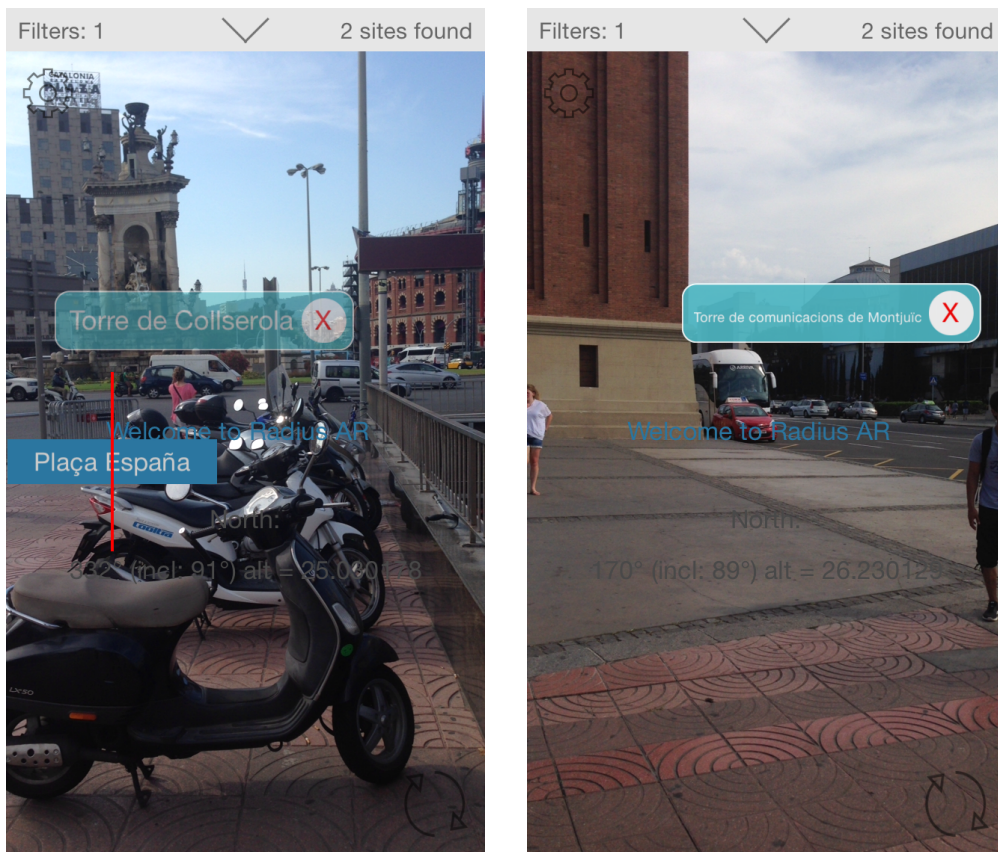


Figura 6.5: Captures. Resultat d'aplicar el filtre *telecommunication* (2)

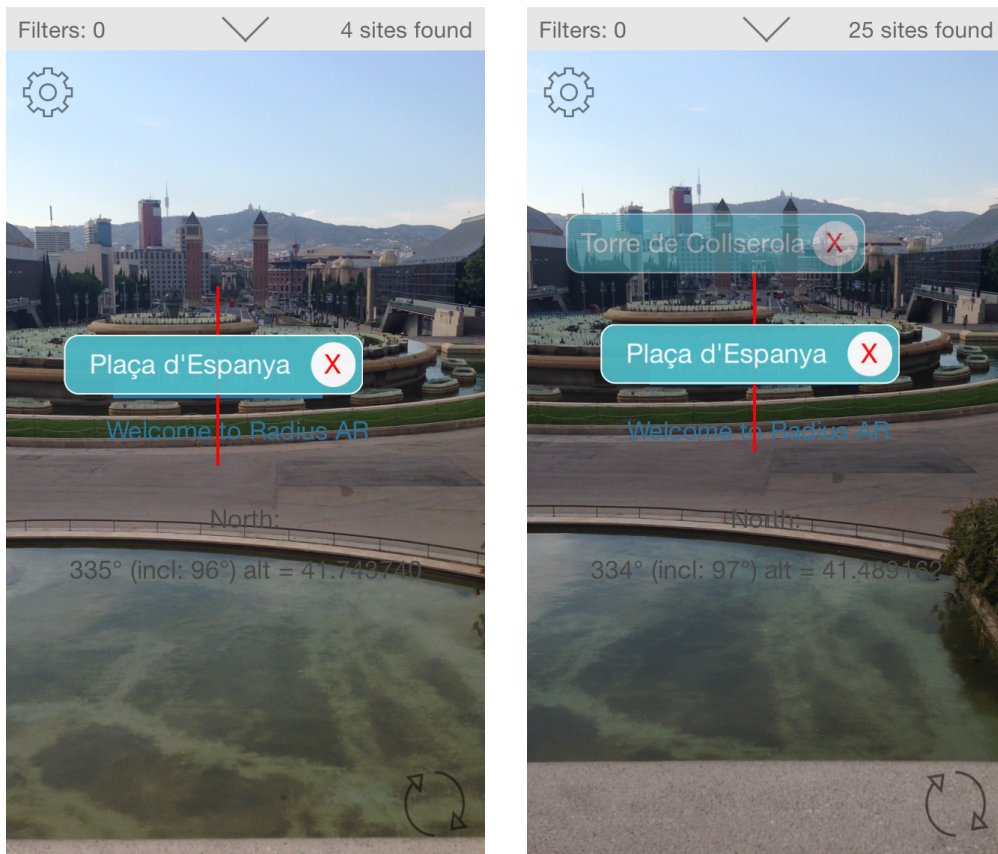


Figura 6.6: Captures. Resultat de realitzar un canvi de radi (filtres)

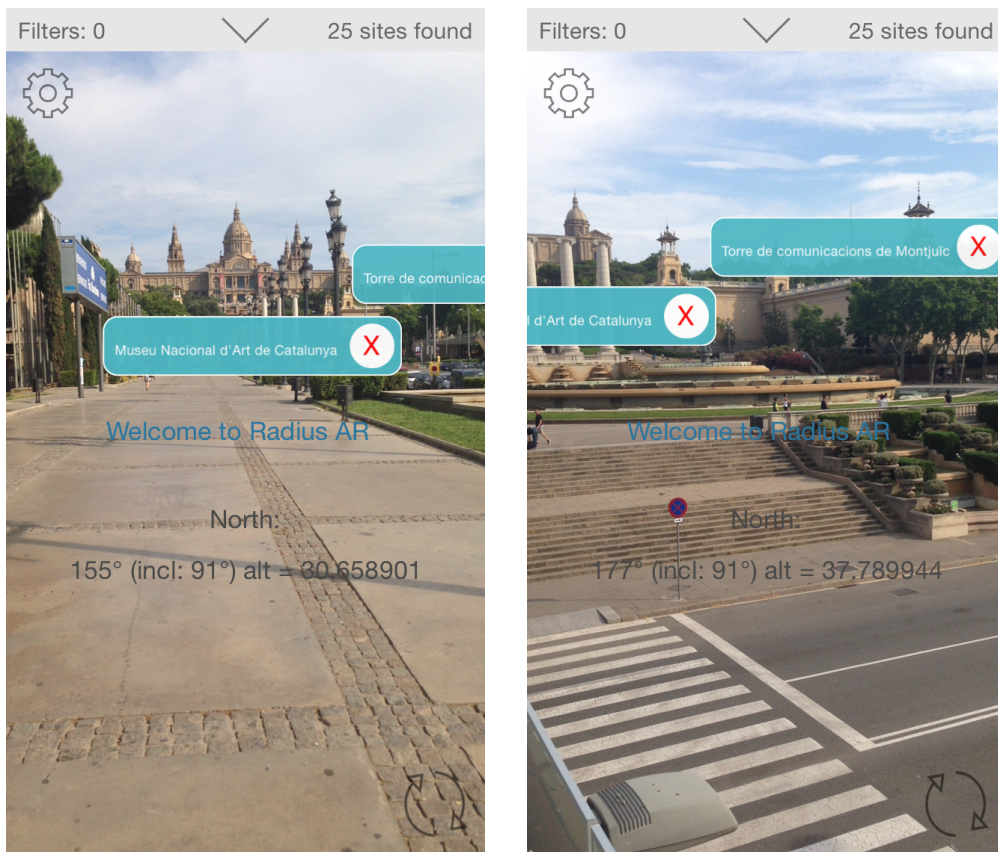


Figura 6.7: Captures. Reposicionament de les etiquetes en temps real

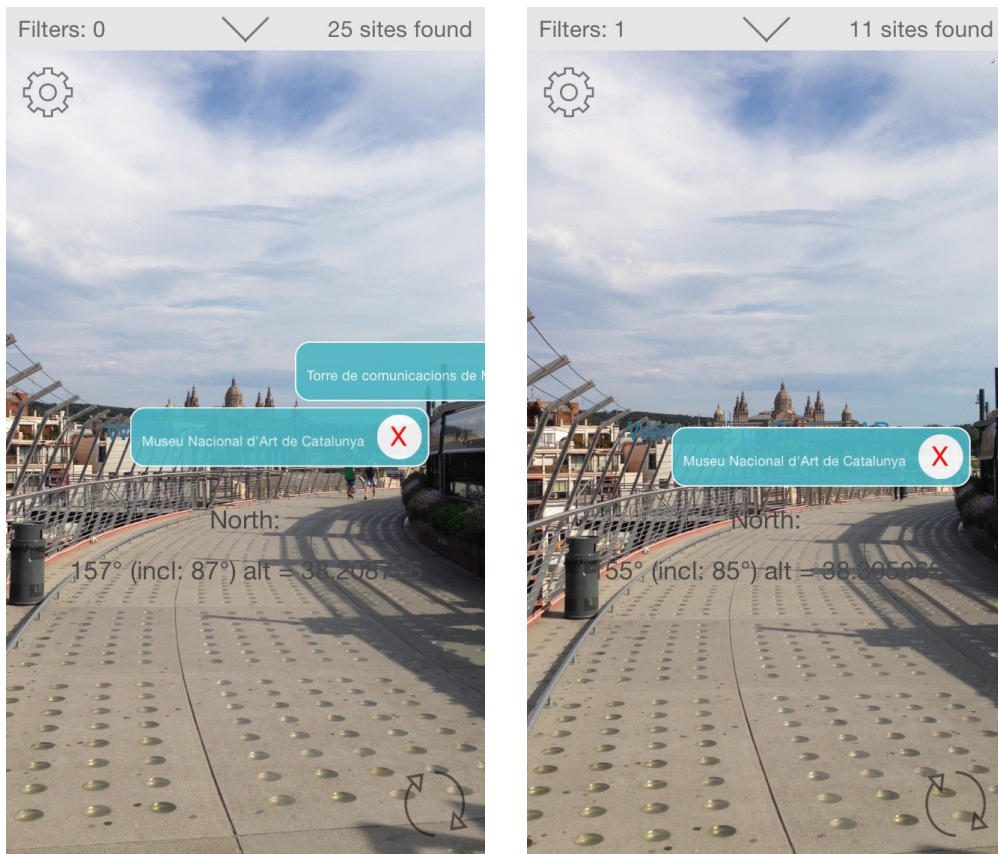


Figura 6.8: Captures. Resultat d'aplicar el filtre *culture*

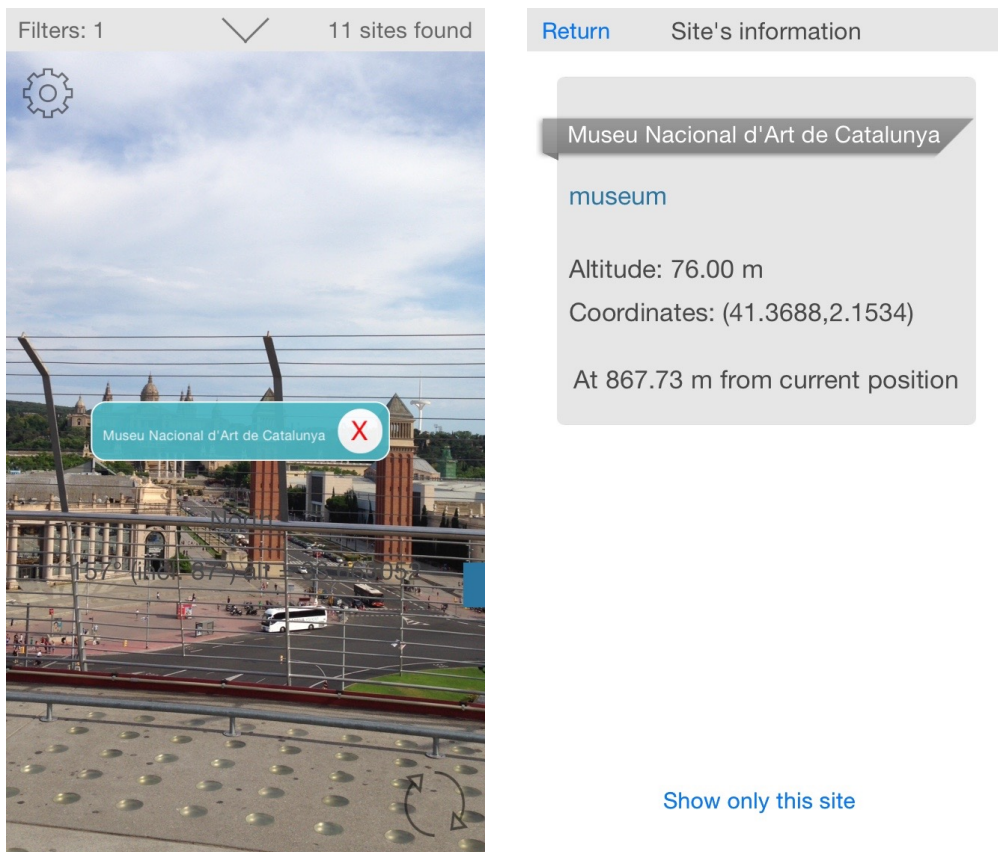


Figura 6.9: Captures. Accés a la vista detall del MNAC

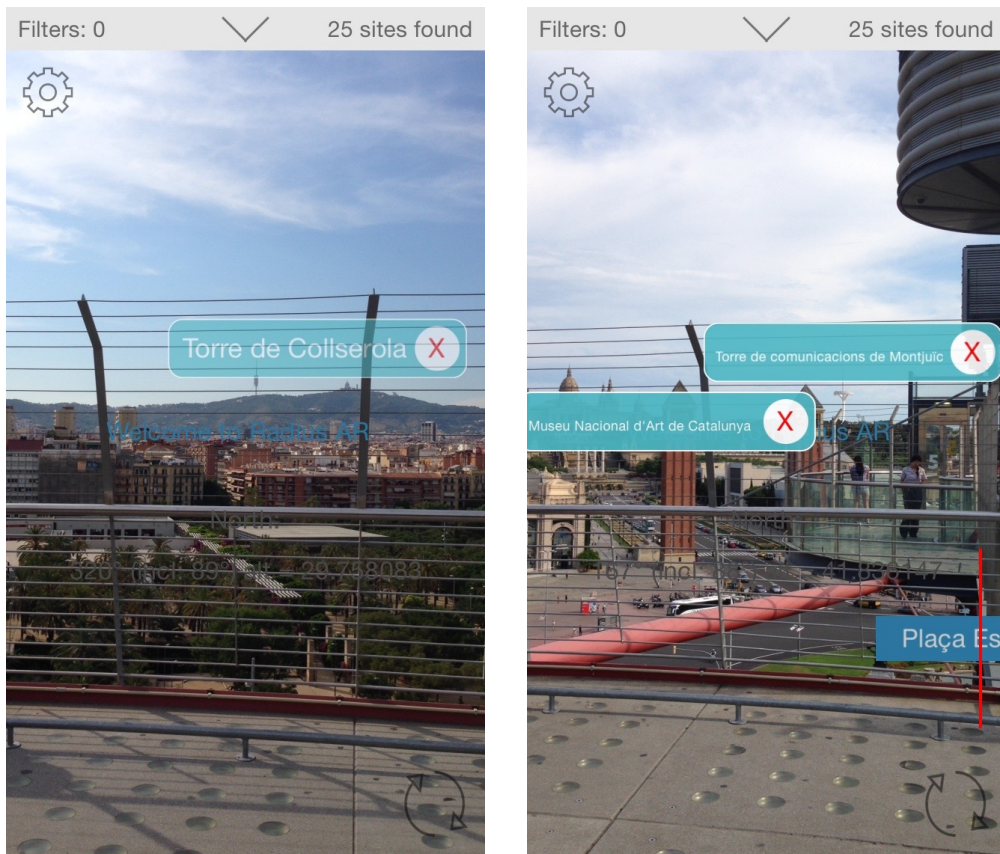


Figura 6.10: Captures. Llocs a llargues distàncies

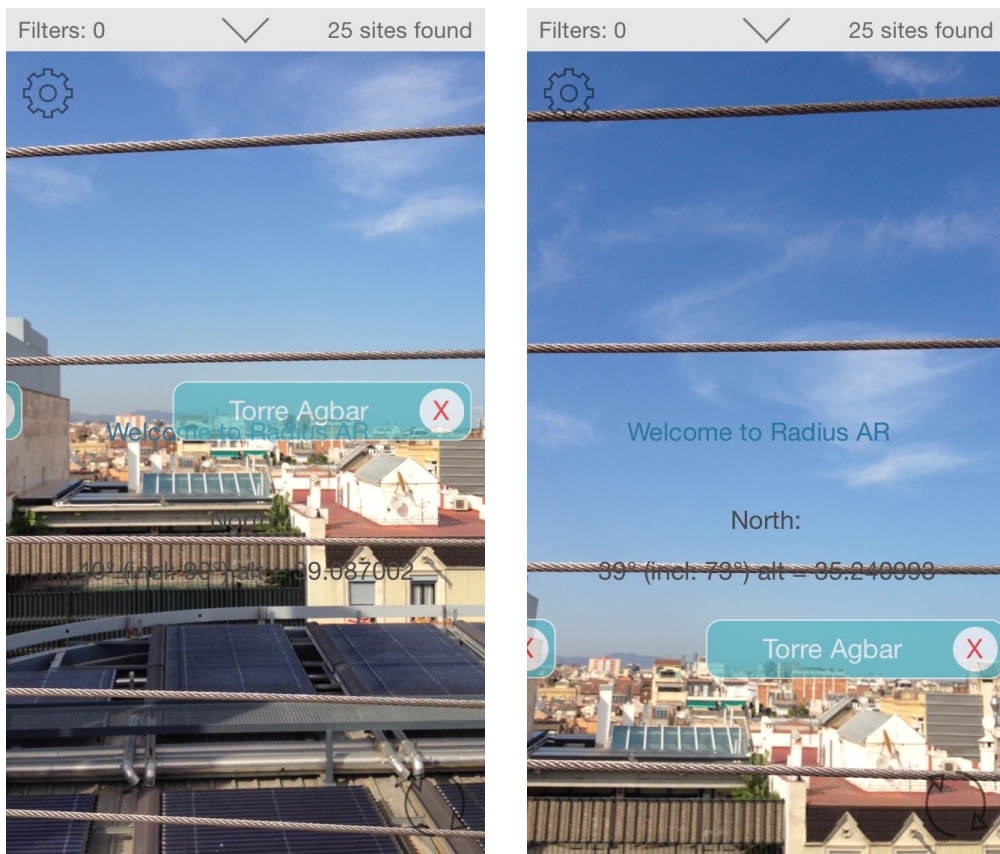


Figura 6.11: Captures. Efecte de canviar la inclinació del terminal

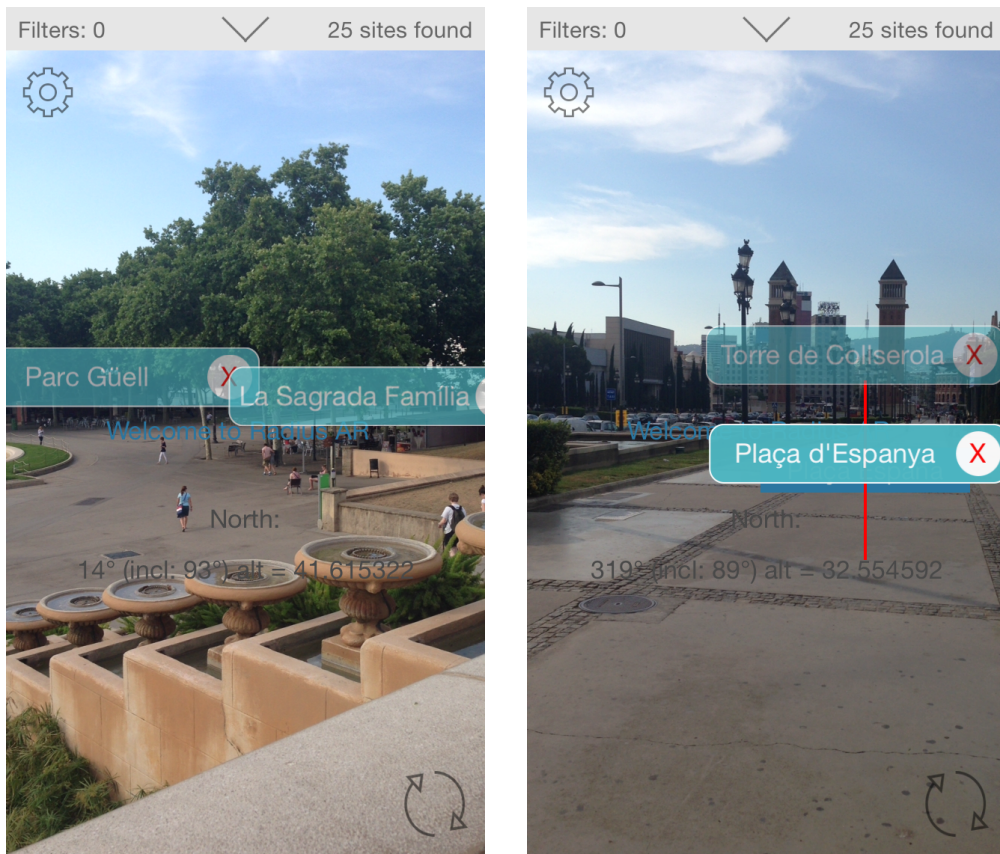


Figura 6.12: Altres captures (1)

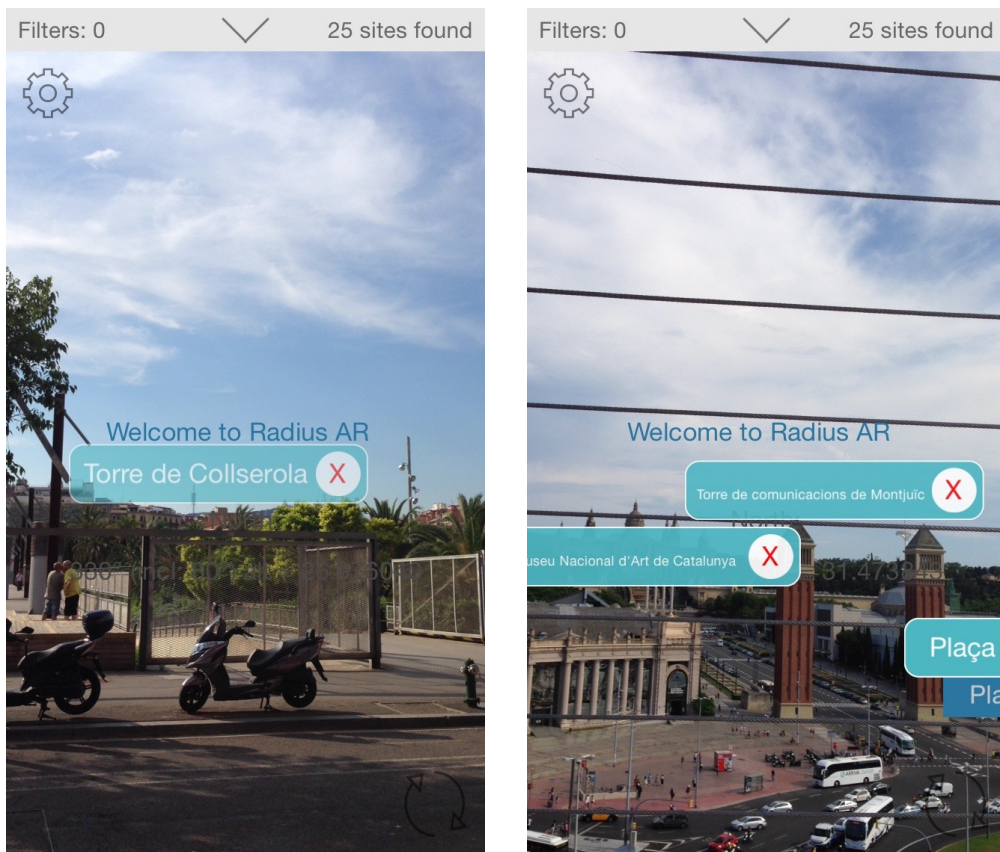


Figura 6.13: Altres captures (2)

Cal destacar que, a l'aplicació, s'ha introduït una etiqueta provisional permanent que mostra la posició de plaça d'Espanya amb un estil diferent. L'objectiu principal d'aquesta etiqueta ha estat ser el punt de referència del testeig durant el procés de proves.

Finalment, i tot i que no formava part dels objectius inicials d'aquesta fase del projecte, s'ha desenvolupat una petita aplicació web basada en *jQueryMobile* per introduir les dades al servidor amb més facilitat. L'aspecte d'aquesta consola es mostra en la figura següent.

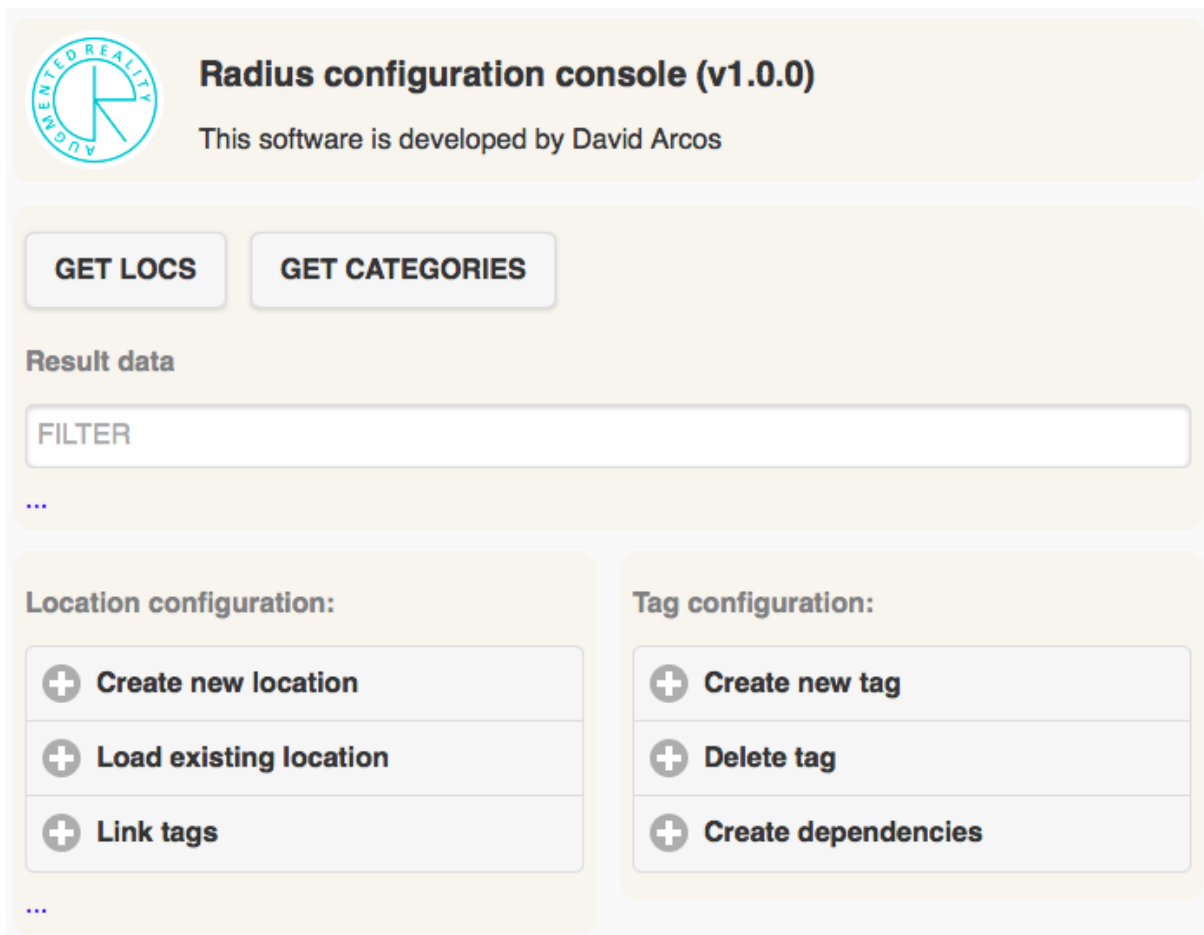


Figura 6.14: Consola per crear els continguts



# SÍNTESI I CONCLUSIONS

## 7.1 Conclusions

Durant l'elaboració d'aquest treball, s'han après moltes coses sobre la realitat augmentada en general i, més concretament, sobre com implementar-la utilitzant Objective-C i Xcode. També s'han millorat molt les destreses en la programació mòbil i, en particular, en l'accés als sensors dels dispositius. A més, s'han pogut aplicar moltes de les tècniques que s'han après al llarg del postgrau (i també moltes de noves).

Del resultat del procés de disseny i desenvolupament de l'aplicació, s'han extret un munt de conclusions i lliçons interessants de les quals, per no allargar el text innecessàriament, només es comentaran les més importants.

- Els sensors del mòbil encara han de millorar molt perquè la realitat augmentada (sense utilitzar algorismes de reconeixement d'imatge) pugui ser realment precisa. Més concretament, cal obtenir dades del giroscopi, del GPS i de l'acceleròmetre amb un nivell de precisió superior.
- La bateria del mòbil és el gran punt feble de les aplicacions de realitat augmentada (pel fet d'utilitzar tants sensors simultàniament). A més, s'ha observat que els dispositius s'escalfen després d'un llarg període d'ús de l'aplicació.
- Les eines de prototipatge encara no estan preparades per representar els modes d'interacció més avançats. Aquesta limitació ha estat un suplici en fer el disseny en *Axure*.
- Cal minimitzar el nombre d'accessos al servidor (o realitzar-los de manera transparent) per millorar l'experiència d'usuari. En les primeres proves d'ús amb usuaris reals, els

temps d'espera eren massa llargs i propiciaven que els usuaris evitessin utilitzar el botó “refrescar”. En la última versió, aquest aspecte s’ha millorat molt.

- La diferència d’altituds no és prou significativa quan els objectes es troben a grans distàncies. Per aquest motiu, la majoria d’etiquetes es troben apilades a la línia de l’horitzó. Per evitar aquest problema es podrien utilitzar tècniques més sofisticades per representar els llocs.

## 7.2 Anàlisi crítica

Tot i que en general s’han assolit tots els objectius del projecte, el tractament d’errors per informar l’usuari no s’ha arribat a implementar. Aquest aspecte és fonamental en una aplicació d’aquest tipus, ja que les fonts d’error són molt diverses i la comunicació amb el servidor és crítica per utilitzar l’aplicació. Aquesta part del desenvolupament no es va plantejar en la planificació de les tasques (ja que no es pretenia implementar-la), però, com que es tracta d’una part tan important, s’hauria d’haver inclòs.

Pel que fa a la planificació de les tasques, la que pitjor s’ha dimensionat és la redacció de la memòria. Com que es tracta d’un projecte tan gran, hi ha moltes coses que s’han d’explicar i justificar. El temps que s’ha planificat per elaborar aquest document ha resultat completament insuficient per reflectir tot el treball que s’ha realitzat. Les tasques de desenvolupament, en canvi, tot i que s’han allargat una mica per culpa dels cicles de implementació-testeig-correcció, s’han assolit amb èxit i sense massa incidents.

## 7.3 Continuació

Com que es tracta d’una versió bàsica d’una l’aplicació amb molt potencial, per continuar aquest projecte hi ha molts camins possibles. A continuació, s’enumeren algunes d’aquestes opcions.

- Implementar la funció del botó “On sóc?” i millorar les dades disponibles dels llocs amb l’objecte *Page*.
- Millorar la compatibilitat d’aquesta aplicació amb tots els dispositius iOS disponibles al mercat. Adaptar les resolucions, les orientacions de pantalla, les diferents obertures de la càmera, etc.
- Desenvolupar un versió Android pels dispositius de gamma alta que disposen de sensors iguals o superiors als iPhones.
- Implementar un sistema perquè els usuaris puguin crear i gestionar els seus propis llocs d’interès.
- Realitzar un estudi sobre la qualitat dels sensors disponibles i establir els millors algorismes possibles per filtrar les dades dels sensors.

- Implementar mètodes per estalviar bateria sense sacrificar en excés les funcionalitats i l'experiència d'usuari.
- Generar una jerarquia completa d'etiquetes basada en un estudi de comportaments de cerca en usuaris reals per optimitzar el sistema de filtres.
- Implementar el disseny de les etiquetes amb els noms dels llocs per evitar acumulacions a la línia de l'horitzó.

# BIBLIOGRAFIA

- [1] Amazon Web Services, Inc. (2015). *Amazon S3* [en línia]. Seattle (EUA): Amazon. <<http://aws.amazon.com/es/s3/>> [Consulta: febrer 2015].
- [2] Amazon Web Services, Inc. (2015). *Amazon EC2* [en línia]. Seattle (EUA): Amazon. <<http://aws.amazon.com/es/ec2/>> [Consulta: febrer 2015].
- [3] Apple, Inc. (2015). *Designing Great Apps* [en línia]. Cupertino (EUA): Apple, Inc. <<https://developer.apple.com/design/>> [Consulta: març 2015].
- [4] Apple, Inc. (2015). *FaceTime* [en línia]. Cupertino (EUA): Apple, Inc. <<https://www.apple.com/ios/facetime/>> [Consulta: març 2015].
- [5] Arcos, D. (2013). Quantum Apps: Simulació dels algorismes de Shor i Grover.
- [6] Arkspai2g SLP (2011). *ArkSolut·CPH - libro y guía de arquitectura* [App Store]. <<https://itunes.apple.com/es/app/arksolut-cph-libro-y-guia/id373541782?mt=8>> [Consulta: març 2015].
- [7] AugmentedWorks (2013). *Places with Augmented Reality* [App Store]. <<https://itunes.apple.com/us/app/places-augmented-reality/id497262224?mt=8>> [Consulta: març 2015].
- [8] AugmentedWorks (2014). *Encuentra tu coche con AR: Augmented Car Finder* [App Store]. <<https://itunes.apple.com/es/app/encuentra-tu-coche-con-ar/id370836023?mt=8>> [Consulta: març 2015].
- [9] «Aurasma Demo». A: *Youtube* [en línia]. Disponible a: <https://www.youtube.com/watch?v=GBKy-hSedg8> [Consulta: març 2015].
- [10] «Aurasma y... ¡Aumenta el mundo!». A: *En la nube TIC* [en línia]. Disponible a: <http://www.enlanubetic.com.es/2013/05/aurasma-y-aumenta-el-mundo.html#.VP1fCofji9Y> [Consulta: març 2015].

- [11] Axure Software Solutions, Inc. (2015). *Axure RP 7* [en línia]. San Diego (EUA): Axure <<http://www.axure.com>> [Consulta: abril 2015].
- [12] Balsamiq Studios, LLC. (2015). *Balsamiq* [en línia]. Sacramento (EUA): Balsamiq Studios, LLC. <<https://balsamiq.com>> [Consulta: març 2015].
- [13] Cabezas, L. M.; González, F. J. (2014). *Desarrollo Web con PHP y MySQL*. Madrid, Anaya Multimedia. ISBN: 978-8441536588.
- [14] «CMMotionManager Class Reference». A: *iOS Developer Library* [en línia]. Apple Inc., Infinite Loop, Cupertino (USA). Actualitzada: 18-09-2013. Disponible a: [https://developer.apple.com/library/ios/documentation/CoreMotion/Reference/CMMotionManager\\_Class/](https://developer.apple.com/library/ios/documentation/CoreMotion/Reference/CMMotionManager_Class/) [Consulta: febrer 2015].
- [15] Exposito López, D.; García Soto, A.; Martín Gomez, A. J. (1999). *Grafos* [en línia]. Granada: Universitat de Granada. <<http://decsai.ugr.es/~jfv/ed1/tedi/cdrom/docs/grafos.htm>> [Consulta: març 2015].
- [16] Geomobile (2012). *VISION Augmented Reality* [App Store]. <<https://itunes.apple.com/es/app/vision-augmented-reality/id462762414?mt=8>> [Consulta: març 2015].
- [17] Griffin Technology (2011). *iTalk Recorder* [App Store]. <<https://itunes.apple.com/en/app/italk-recorder/id293673304?mt=8>> [Consulta: març 2015].
- [18] Hillegass, A.; Conway, J. (2011). *Desarrollo de aplicaciones para iPhone & iPad*. Madrid, Anaya Multimedia. ISBN: 978-8441529328.
- [19] «iOS - Calculating distance, azimuth, elevation and relative position (Augmented Reality)». A: *stackoverflow* [en línia]. Stack Exchange Inc., USA. Disponible a: <http://stackoverflow.com/questions/14070931/ios-calculating-distance-azimuth-elevation-and-relative-position-augmented/14071335#14071335> [Consulta: març 2015].
- [20] «iTunes en un vistazo». A: *iTunes* [en línia]. Apple Inc., Infinite Loop, Cupertino (USA). Disponible a: <https://itunes.apple.com/es/genre/ios/id36?mt=8> [Consulta: març 2015].
- [21] «Las 5 mejores aplicaciones de Realidad Aumentada para celulares». A: *Hipertextual* [en línia]. Hipertextual, Madrid (Espanya). Disponible a: <http://hipertextual.com/2010/01/las-5-mejores-aplicaciones-de-realidad-aumentada-para-celulares> [Consulta: febrer 2015].
- [22] Metaio GmbH (2014). *Junaio Navegador de Realidad Aumentada* [App Store]. <<https://itunes.apple.com/es/app/junaio-navegador-realidad/id337415615?mt=8>> [Consulta: març 2015].
- [23] Metaio GmbH (2014). *Junaio Navegador de Realidad Aumentada* [Google Play]. <<https://play.google.com/store/apps/details?id=com.metaio.junaio>> [Consulta: març 2015].
- [24] Metaio GmbH (2014). *Junaio* [en línia]. <<http://www.junaio.com>> [Consulta: març 2015].

- [25] «Motion Events». A: *iOS Developer Library* [en línia]. Apple Inc., Infinite Loop, Cupertino (USA). Actualitzada: 18-09-2013. Disponible a: [https://developer.apple.com/library/ios/documentation/EventHandling/Conceptual/EventHandlingiPhoneOS/motion\\_event\\_basics/motion\\_event\\_basics.html](https://developer.apple.com/library/ios/documentation/EventHandling/Conceptual/EventHandlingiPhoneOS/motion_event_basics/motion_event_basics.html) [Consulta: maig 2015].
- [26] Rogers, Y.; Sharp, H.; Preece, J. (2011). *Interaction design*. John Wiley & Sons, Inc. ISBN 978-0-470-66576-3.
- [27] Saliola, G. P. (2014). *Mountains AR the augmented reality to discover mountains, peaks and summits* [App Store]. <<https://itunes.apple.com/us/app/mountains-ar-augmented-reality/id884969067?mt=8>> [Consulta: març 2015].
- [28] SensioLabs (2015). *Symfony* [en línia]. París: SensioLabs. <<http://symfony.com>> [Consulta: febrer 2015].
- [29] SourceForge.net (2015). *phpMyAdmin. Bring MySQL to the web* [en línia]. <<http://phpmyadmin.net>> [Consulta: abril 2015].
- [30] Tigris.org, Open Source Software Engineering Tools. (2009). *Welcome to ArgoUML* [en línia]. <<http://argouml.tigris.org>> [Consulta: abril 2015].
- [31] Transports Metropolitans de Barcelona (2010). *Bus Turístic Virtual* [Google Play]. <<https://play.google.com/store/apps/details?id=com.geomobile.tmbmobile.bbs>> [Consulta: març 2015].
- [32] Transports Metropolitans de Barcelona (2012). *Bus Turístic Virtual* [App Store]. <<https://itunes.apple.com/en/app/bus-turistic-virtual/id390642423?mt=8>> [Consulta: març 2015].
- [33] Universitat Oberta de Catalunya (2013). «Te presentamos a Jordi y Martina, las “Personas” de la UOC». *El blog de las aulas* [en línia]. Barcelona: Universitat Oberta de Catalunya. <<http://aula.blogs.uoc.edu/2013/10/23/et-presentem-al-jordi-i-la-martina-les-persones-de-la-uoc/>> [Consulta: març 2015].