

## **Treball final de grau**

**Estudi: Grau en Enginyeria Informàtica**

**Títol:** Aplicació mòbil/IoT per orientar-se a l'interior dels edificis de l'EPS

**Document:** Memòria

**Alumne:** Genís Arumí Novellas, u1933627

**Tutor:** Pere Vilà Talleda

**Departament:** Arquitectura i Tecnologia de Computadors

**Àrea:** Arquitectura i Tecnologia de Computadors

**Tutor:** Josep Blanes Gimferrer

**Departament:** Servei Informàtic de l'EPS

**Àrea:** Àrea d'estudis tècnics

**Convocatòria (mes/any):** Setembre 2019

Aplicació mòbil/IoT per orientar-se a l'interior dels edificis de l'EPS

## Contingut

<b>1. Introducció .....</b>	<b>- 6 -</b>
1.1 Motivacions i propòsit.....	- 6 -
1.2 Objectius .....	- 7 -
<b>2. Estudi de viabilitat.....</b>	<b>- 8 -</b>
2.1 Viabilitat tecnològica .....	- 8 -
2.2 Viabilitat econòmica .....	- 8 -
2.3 Conclusió de la viabilitat.....	- 10 -
<b>3. Metodologia.....</b>	<b>- 11 -</b>
<b>4. Planificació .....</b>	<b>- 13 -</b>
4.1 Pla de treball.....	- 13 -
4.2 Planificació del projecte .....	- 15 -
<b>5. Marc de treball i conceptes previs.....</b>	<b>- 17 -</b>
5.1 Participants en el Treball de Fi de Grau .....	- 17 -
5.2 Frameworks i tecnologies.....	- 17 -
5.2.1 Android .....	- 17 -
5.2.2 Android Studio .....	- 18 -
5.2.3 Balisa / Beacon.....	- 18 -
5.2.4 Apiary.....	- 19 -
5.2.5 GitHub.....	- 20 -
5.2.6 Gradle .....	- 20 -
5.2.7 Android SDK.....	- 21 -
<b>6. Requisits del sistema .....</b>	<b>- 22 -</b>
6.1 Actors .....	- 22 -
6.2 Requisits funcionals .....	- 22 -
6.2.1 Requisits funcionals en la gestió de permisos de Bluetooth i Localització.....	- 22 -
6.2.2 Requisits funcionals en l'elecció de la destinació .....	- 23 -
6.3 Requisits no funcionals .....	- 23 -
<b>7. Estudis i decisions .....</b>	<b>- 25 -</b>
7.1 Android vs Swift .....	- 25 -
7.1.1 Android .....	- 26 -
7.2 Programes & editors utilitzats .....	- 29 -
7.2.1 Android Studio .....	- 29 -

7.2.2 GitHub.....	- 29 -
7.2.3 Apiary.....	- 30 -
7.3 Llibreries .....	- 31 -
7.3.1 iBKS Plus.....	- 31 -
7.3.2 Estimote Proximity.....	- 32 -
7.3.3 Retrofit .....	- 33 -
7.3.4 Google Play Services Location .....	- 33 -
7.3.5 Llibreries suport Android .....	- 34 -
7.6 Software complementari .....	- 35 -
<b>8. Anàlisi i disseny del sistema .....</b>	<b>- 36 -</b>
8.1 Anàlisi del sistema .....	- 36 -
8.1.1 Anàlisi de la gestió de permisos de Bluetooth i Localització.....	- 36 -
8.1.2 Anàlisi de l'elecció de destinació.....	- 37 -
8.1.3 Anàlisi dels beacons .....	- 37 -
8.1.4 Anàlisi dels camins.....	- 37 -
8.2 Disseny del sistema .....	- 38 -
8.2.1 Disseny del model de dades .....	- 38 -
8.2.2 Disseny de la interfície d'usuari.....	- 41 -
<b>9. Implementació i proves.....</b>	<b>- 45 -</b>
9.1 Preparació dels beacons Estimote .....	- 45 -
9.1.1 Problemes amb la propietat dels beacons.....	- 48 -
9.2 Configuració dels beacons Estimote .....	- 50 -
9.3 Preparació d'Apiary .....	- 53 -
9.4 Preparació d'Android Studio.....	- 53 -
9.5 Desenvolupament del projecte .....	- 53 -
9.5.1 Implementació de l'aplicació .....	- 54 -
9.5.2 Problemes amb el desenvolupament de l'aplicació.....	- 58 -
9.6 Proves.....	- 60 -
9.6.1 Proves Apiary.....	- 60 -
9.6.2 Proves beacons iBKS i Estimote.....	- 62 -
9.6.3 Proves indicacions del camí a seguir.....	- 64 -
<b>10. Implantació i resultats .....</b>	<b>- 66 -</b>
10.1 Resultat d'interfície.....	- 66 -
10.2 Resultat final del projecte.....	- 71 -

10.3 Compliment amb la legislació i normatives vigents .....	- 75 -
<b>11. Conclusions.....</b>	<b>- 76 -</b>
11.1 Valoració dels objectius assolits .....	- 76 -
11.2 Desviacions de la planificació.....	- 77 -
11.3 Aprenentatge assolit amb el projecte.....	- 78 -
<b>12. Treball futur .....</b>	<b>- 79 -</b>
<b>13. Webgrafia .....</b>	<b>- 81 -</b>
<b>14. Annexos .....</b>	<b>- 84 -</b>
14.1 Codi font del projecte.....	- 84 -
14.2 Apiary utilitzat en el projecte .....	- 84 -
14.3 Índex de Figures del projecte .....	- 84 -
<b>15. Manual d'usuari i/o instal·lació .....</b>	<b>- 87 -</b>
15.1 Instal·lació.....	- 87 -
15.2 Manual d'ús.....	- 87 -
15.3 Exemple d'ús.....	- 89 -

## **1. Introducció**

Aquest projecte consisteix en desenvolupar una aplicació mòbil utilitzant tecnologies d'*Internet of Things (IoT)* [1]. L'aplicació resultant servirà per guiar als seus usuaris, mitjançant uns dispositius *IoT*, fins a destinacions concretes.

Avui en dia, les tecnologies d'*Internet of Things* s'estan escampant molt ràpidament i ja es poden trobar per tot arreu. Aquestes tecnologies es poden entendre com un concepte que es basa en la interconnexió d'objectes, els quals són objectes físics amb components elèctrics, software, sensors, connectivitat a la xarxa...

Actualment ens trobem en una situació on gairebé tothom disposa de dispositius mòbils prou moderns com per ser utilitzats per orientar-se a l'interior d'edificis, sense la necessitat de fer servir el GPS. La cobertura de GPS en edificis és molt dolenta o pràcticament nul·la, cosa que pot comportar una mala precisió en la ubicació.

### **1.1 Motivacions i propòsit**

Cada any hi ha molts alumnes, conferencians o visitants, que s'han de moure pels edificis de l'Escola Politècnica Superior i no saben on són les aules, espais o despatxos de professors que estan buscant. Per aquest motiu, des del Servei Informàtic de l'EPS es vol impulsar un servei que permeti solucionar aquest problema.

Normalment aquest problema es pot solucionar col·locant mapes o senyals dins els edificis, per tal que la gent es pugui situar i guiar, o bé demanant al personal, encara que aquest últim cas pot ser complicat si els visitants són estrangers. Tot i que les solucions anteriors en la majoria de casos són suficients per solucionar el problema, també cal dir que són solucions més clàssiques o bàsiques.

Una solució més moderna seria utilitzar el GPS, però a l'interior dels edificis de l'EPS la senyal és molt dolenta o, fins i tot, hi ha llocs on no arriba.

Tenint en compte que aquest treball és el projecte final d'una Enginyeria Informàtica, es vol aportar una solució moderna i tecnològica per al problema anterior. D'aquesta manera, espero poder posar en pràctica tots els coneixements adquirits durant el grau i aprofitar aquest projecte per aprendre noves tecnologies que cada vegada guanyen més importància.

## 1.2 Objectius

L'objectiu d'aquest Treball de Final de Grau és desenvolupar l'anàlisi, disseny i implementació d'una aplicació mòbil Android, per a orientar-se per l'interior dels edificis de l'EPS, que sigui intuïtiva, senzilla d'utilitzar i no carregui gaire el dispositiu mòbil.

En aquesta aplicació s'haurà de poder indicar quina aula (*I-01, II-07...*), espai (*Sala d'actes, Secretaria...*), despatx, nom de professor, PAS, etc. es vol trobar i l'aplicació donarà unes indicacions concretes que permeti arribar a la destinació escollida.

Juntament amb el desenvolupament de l'aplicació, un altre objectiu és utilitzar dispositius d'*Internet of Things*, concretament del tipus *beacons* o balises, de manera que aquests dispositius i l'aplicació es puguin comunicar per tal d'ubicar els usuaris.

Durant el projecte es farà ús de l'API d'Apiary per recuperar les dades necessàries que s'utilitzaran a l'aplicació i posteriorment, quan es posi l'aplicació en producció, s'afegirà un servidor per substituir Apiary.

Finalment, amb aquesta aplicació es vol fer un desenvolupament de test per demostrar-ne la seva viabilitat, utilitzant poques balises degut al seu cost important, i si funciona correctament i es veu pràctic en un futur implantar-ho a tots els edificis de l'Escola Politècnica Superior adquirint un gran volum de balisses.

## **2. Estudi de viabilitat**

Per justificar la viabilitat del projecte és necessari tenir en compte les tecnologies necessàries i l'impacte econòmic que pot suposar, per això es farà un estudi de la viabilitat tecnològica i econòmica.

### **2.1 Viabilitat tecnològica**

A continuació s'especificarà una llista amb tots els materials necessaris per al desenvolupament del projecte:

- Un ordinador amb connexió a internet
- Entorns de programació per realitzar el projecte amb les seves llicències
- Balises necessàries per desenvolupar el projecte
- Un dispositiu mòbil compatible amb les balises
- Software d'Apiary per recuperar dades durant el desenvolupament del projecte
- Posteriorment, un servidor per substituir Apiary

La majoria dels materials anteriors són de fàcil accés així que no han de suposar un problema a l'hora d'obtenir-los pel projecte.

La intenció és utilitzar balises en el desenvolupament del projecte i no hauria de suposar cap problema el seu ús, però no s'està segur si seran apropiades pel que caldrà fer degut al desconeixement d'aquestes noves tecnologies.

### **2.2 Viabilitat econòmica**

L'estudi de la viabilitat econòmica tindrà en compte aspectes com ara les hores invertides i els costos de tots els materials i software que es pretenen utilitzar en el projecte.

Abans de començar a analitzar les hores a invertir en el projecte, és necessari tenir en compte les hores estimades de desenvolupament d'aquest. Segons els 15 crèdits ECTS oficials dedicats al projecte s'haurien de dedicar, com a màxim, 375 hores en el seu desenvolupament.

En el desenvolupament del projecte es dedicaran hores a:

- Anàlisi



## Aplicació mòbil/IoT per orientar-se a l'interior dels edificis de l'EPS

- Disseny
- Estudi de les balises
- Programació
- Generació de dades a Apiary
- Proves del funcionament correcte de l'aplicació
- Creació d'un mallat de balises
- Tests dels camins generats a partir del mallat
- Documentació

El cost de les hores de dedicació per al desenvolupament del projecte no es tindran en compte, ja que el projecte es desenvoluparà dins el marc d'hores que es consideren apropiades en un Projecte de Fi de Grau, 375 hores tal i com s'ha comentat abans, i per tant, entren dins les activitats acadèmiques del Grau.

En el cas dels materials, el seu cost serà el següent:

<b>Material</b>	<b>Cost</b>
Ordinador	600€
Android Studio	Gratuït
Apiary	Gratuït
GitHub	Gratuït
Balises x5	140 €
Dispositiu mòbil	140 €

El cost del servidor no està inclòs a la llista anterior, ja que apart que s'afegirà posteriorment al projecte, encara no se sap exactament quin material serà necessari per a la seva construcció ni el tipus de servidor que s'utilitzarà.

Pel que fa als altres materials necessaris, inicialment el seu cost no es tindrà en compte perquè ja es disposa de tots aquests. En el cas de les balises, inicialment se'n disposa d'unes quantes per començar el desenvolupament del projecte, però com que no se sap si seran apropiades, és possible que se n'hagi de comprar de noves i de marques diferents i pot ser que el seu preu canviï.

### **2.3 Conclusió de la viabilitat**

Segons els estudis anteriors, es pot arribar a la conclusió que aquest projecte és viable tant tecnològicament com econòmicament perquè ja es disposa de tot el material necessari i el software que s'utilitzarà és gratuït.

L'únic inconvenient en la viabilitat econòmica és la possibilitat d'haver de comprar més balises en el cas que se'n necessitin més per desenvolupar correctament el projecte o que aquestes no siguin adequades i, per tant, se n'hagin de comprar de diferents.

### 3. Metodologia

Una metodologia és un conjunt de fases i etapes, que formen el procés de desenvolupament d'un sistema informàtic. No hi ha una metodologia única ja que segons l'entorn i condicions de treball s'ha d'acordar la metodologia que s'adapti millor a aquestes.

Com que inicialment es desconeixia el que es podia fer i el que no amb les tecnologies utilitzades durant el projecte, les balises, no es va utilitzar, per exemple, el model cascada, una metodologia molt útil en projectes on els requisits no variaran durant el desenvolupament. Per això, per desenvolupar aquest projecte no s'ha seguit una metodologia estàndard, sinó que s'ha utilitzat una metodologia iterativa que permetés anar endavant i endarrere en els passos a seguir, ja que els requisits anaven canviant o hi havia canvis en el disseny o implementació.

Els passos que s'han seguit de la metodologia són els següents:

1. Escollir les eines i llenguatge de programació.
2. Recordar i millorar el coneixement del llenguatge i de les eines.
3. Fer l'anàlisi i disseny de l'aplicació.
4. Estudi dels *beacons* i escollir la marca i tipus.
5. Comprar els *beacons*.
6. Estudiar i fer proves amb codi d'exemple dels *beacons*. Si el funcionament dels *beacons* no és adequat o l'esperat, tornar al punt 4.
7. Desenvolupar el mòdul App/Interfície i construcció del mòdul d'enllaç i control de les balises a partir de les proves fetes al punt 6.
8. Comprovar el funcionament correcte del mòdul App/Interfície. Si alguna cosa no és correcte, tornar al punt 7.
9. Ajuntar el mòdul App/Interfície amb el mòdul d'enllaç i control de les balises. Si hi ha problemes o errors, tornar al punt 7.
10. Fer proves amb els dos mòduls junts. Si hi ha problemes o errors, tornar al punt 7.
11. Crear un mallat de punts eficient, per tal d'utilitzar el mínim de *beacons* possible.
12. Completar i documentar el projecte.

Tot seguit es pot veure la metodologia utilitzada en un diagrama d'activitats:

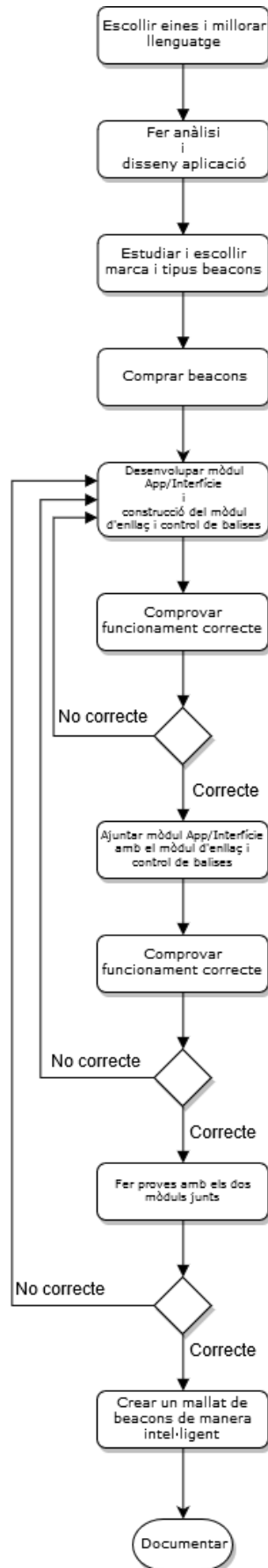


Figura 3.1: Diagrama de la metodologia utilitzada.

## 4. Planificació

En aquest apartat es comentaran els passos que s'han de seguir per desenvolupar el projecte i es complementarà amb un diagrama de Gantt per detallar el temps estimat de cada tasca fins a la finalització del projecte.

Per poder explicar millor la planificació, es separa el projecte en dos mòduls:

- Mòdul App/Interfície
- Mòdul d'enllaç i control dels *beacons*

Cal destacar que en els dos mòduls s'hi inclou la funció de comunicar-se amb Apiary, que posteriorment servirà per comunicar-se amb el servidor. També cal tenir en compte que, tot i que la major part d'aquests mòduls es poden desenvolupar seqüencialment, un darrere l'altre, certes parts també es podran desenvolupar en paral·lel, com ara el desenvolupament dels dos mòduls.

### 4.1 Pla de treball

El projecte es va començar a mitjans de març de 2019, comentant, amb el tutor i el cotutor, les idees generals de com havia de ser l'aplicació funcionalment i gràficament.

Una vegada s'havia deixat clara la idea principal de l'aplicació, s'havien d'escollir les balises que s'utilitzarien per fer el projecte. El cotutor feia poc que n'havia comprat unes quantes de la marca iBKS de l'empresa Accent Systems [2], així que es van aprofitar i se'n va començar a buscar informació i a estudiar-les per saber com funcionaven.

Mentre s'estudiaven es va trobar que es necessitava una versió mínima d'un dispositiu mòbil Android per poder treballar amb les balises, així que durant el mes d'abril es va buscar un dispositiu que complís aquests requeriments i es pogués utilitzar al llarg de tot el projecte.

Com que ja s'havia utilitzat Android el curs anterior, abans de començar a desenvolupar, es va fer un curs online d'Android a Udacity [3] per recordar-ne els conceptes principals.

A principis de maig es va començar el desenvolupament del projecte. Primer es va centrar en el mòdul App/Interfície i posteriorment en el mòdul d'enllaç i control de les balises.

A finals de maig es va acabar el desenvolupament dels dos mòduls i després de fer-ne diverses proves i d'ensenyar-li al cotutor els resultats obtinguts amb aquestes, es va arribar a la conclusió que els *beacons* iBKS no eren prou precisos i que per tant no ens servien. D'aquesta manera es va haver de buscar una altra marca de balises que es pogués utilitzar al projecte, cosa que va comportar tornar al punt 4 de la metodologia. Finalment es van trobar uns *beacons* concrets de la marca Estimote [4] que s'ajustaven molt bé al projecte.

Degut al problema anterior, no es va poder entregar i presentar el projecte al juny, tal i com es tenia previst, així que vaig acabar preparant el projecte per presentar-lo al setembre.

Mentre no es disposava dels dispositius nous, es va dedicar aquest temps a buscar informació sobre aquests i a estudiar-ne les característiques. A mitjans de juny van arribar els nous dispositius, així que fins a finals de juny es va estar desenvolupant altra vegada el mòdul d'enllaç i control de *beacons*.

Després de fer-ne totes les proves pertinents, es va comprovar que els nous dispositius sí que funcionaven correctament i que per tant, es podrien utilitzar.

Fins a mitjans de juliol, es va acabar completament el disseny de la interfície, ja que es van haver de fer algunes modificacions. També es va aprofitar per solucionar algun error que no havia aparegut fins llavors.

Finalment, quan l'aplicació va estar acabada només faltava fer un estudi per construir un mallat de *beacons* de manera intel·ligent, reduint el problema d'anar d'un punt a un altre, intentant utilitzar el mínim número de punts possible i aprofitant rutes als edificis de l'Escola Politècnica Superior.

Durant el desenvolupament de l'aplicació es va anar consultant amb el cotutor si calia variar algun aspecte de l'aplicació, com ara el disseny de la interfície i es va anar informant al tutor sobre l'estat del treball.

També durant tot el projecte es va recopilar informació i documentació sobre les decisions preses o canvis i la documentació final es va fer de mitjans de juliol a mitjans d'agost.

## 4.2 Planificació del projecte

El següents diagrames de Gantt han estat generats amb l'aplicació online *TeamGantt* [5] que proporciona una versió gratuïta per a un sol projecte.

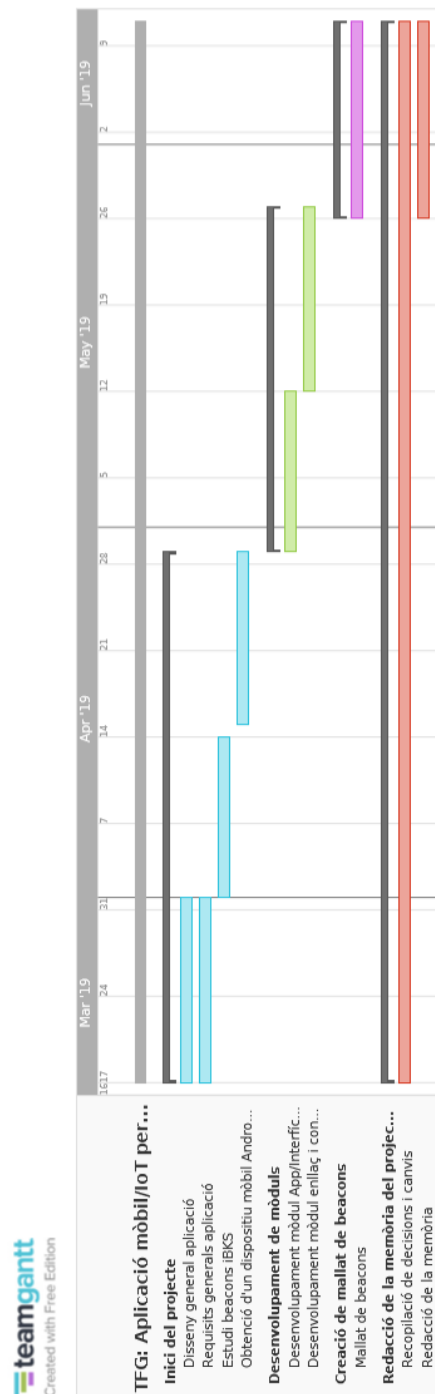


Figura 4.1: Primera planificació del projecte

A la Figura 4.1 es pot veure la planificació inicial del projecte, però degut al canvi de *beacons* es va haver de canviar i va quedar de la manera següent:

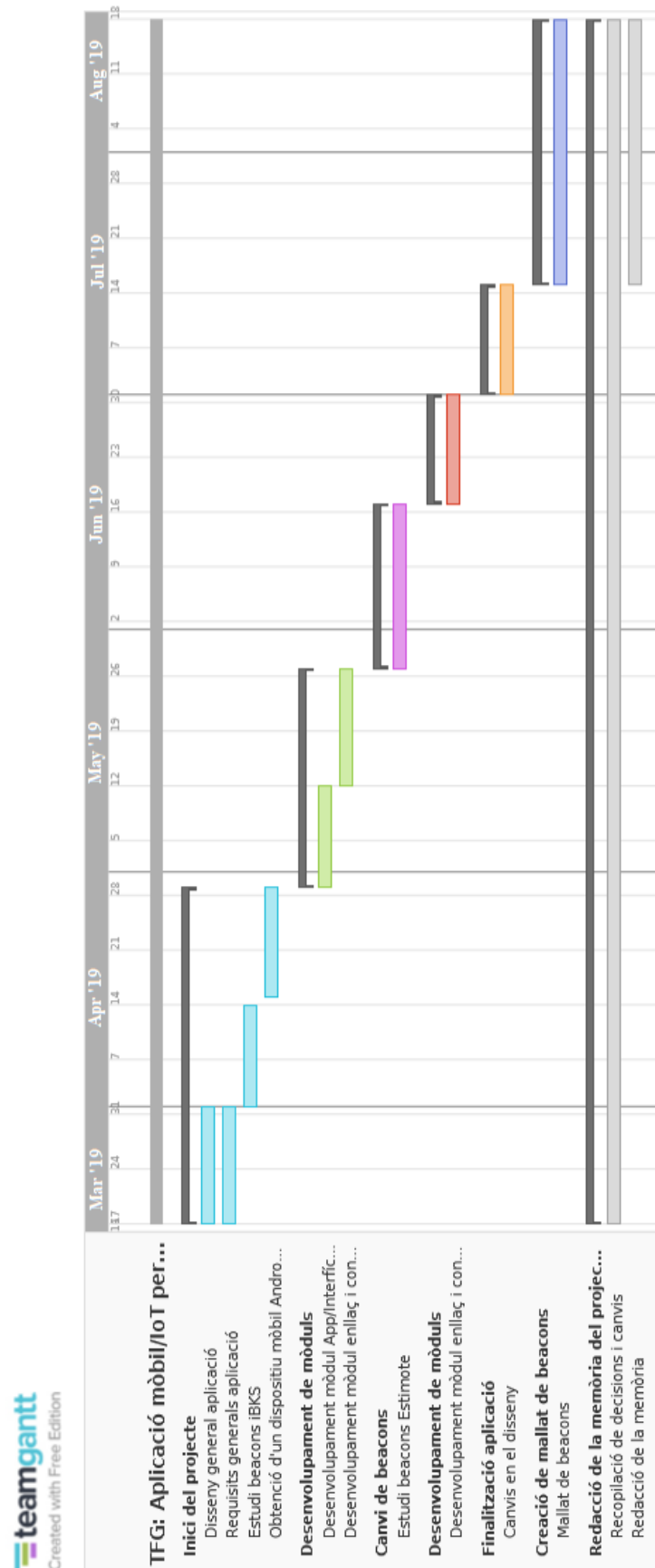


Figura 4.2: Planificació final



## **5. Marc de treball i conceptes previs**

En aquest apartat s'expliquen els diferents conceptes que apareixen al llarg de tot el document i que són necessaris per entendre el desenvolupament del projecte.

Aquest projecte s'ha desenvolupat per l'Escola Politècnica Superior de la Universitat de Girona, concretament pel departament de Servei Informàtic.

### **5.1 Participants en el Treball de Fi de Grau**

#### **Pere Vilà Talleda**

- Tutor del projecte.
- Professor del departament d'Arquitectura i Tecnologia de Computadors a la Universitat de Girona.

#### **Josep Blanes Gimferrer**

- Cotutor del projecte.
- Cap de la Secretaria Informàtica de l'Escola Politècnica Superior de la Universitat de Girona.

### **5.2 Frameworks i tecnologies**

A continuació s'explicaran els entorns de desenvolupament i tecnologies amb els que es realitzarà aquest Treball de Fi de Grau.

#### **5.2.1 Android**

Android és un sistema operatiu mòbil desenvolupat per Google. El que el distingeix de la resta de sistemes operatius per mòbils és que està basat en Linux, un nucli de sistema operatiu lliure, gratuït i multiplataforma. [6]

Aquest sistema operatiu permet programar aplicacions amb el llenguatge Java i proporciona totes les interfícies necessàries per desenvolupar aplicacions que accedeixin a les funcions del telèfon, com ara el GPS, les trucades, l'agenda...



**Figura 5.1: Logo d'Android**

### 5.2.2 Android Studio

Android Studio és l'entorn de desenvolupament oficial per la plataforma Android. Està publicat de manera gratuïta sota la Licència Apache 2.0. [7]



Figura 5.2: Logo d'Android Studio

### 5.2.3 Balisa / Beacon

Una balisa o *beacon* és un dispositiu petit basat en tecnologia Bluetooth de baix consum que constantment emet una senyal que l'identifica de manera única. Aquesta senyal sol ser rebuda i interpretada per un dispositiu mòbil. Normalment, amb la senyal, apart de saber la identitat del dispositiu que l'ha emès, també es pot saber la distància a la que es troba. Hi ha *beacons* de moltes marques diferents però les més conegudes són les següents: [8, 9]



Figura 5.3: Beacon iBKS

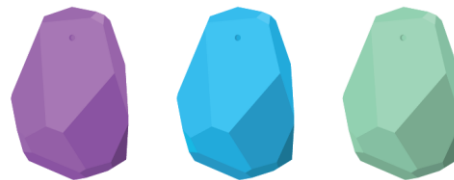


Figura 5.4: Beacon Estimote

Cal dir que en el cas dels *beacons* Estimote hi ha molts tipus diferents de *beacons* amb funcionalitats diferents. [10]

### Tipus de *Beacons* Estimote

- **Sticker beacon:** Són més petits que la resta de beacons i van equipats amb un acceleròmetre, sensor de temperatura i una ràdio bluetooth potent.
- **Proximity Beacon:** Útils per detectar quan s'està a la vora d'àrees d'interès. Es posa un identificador al beacon i a l'acostar-s'hi el detecta.
- **Location Beacon:** Útils per replicar la tecnologia GPS, però en zones interiors, on no hi ha senyal de satèl·lit. Col·locant-ne uns quants en una àrea ajuda a obtenir coordenades (x, y) precises.
- **Video Beacon:** Beacon que es pot connectar a un dispositiu via HDMI, d'aquesta manera es pot controlar què s'està veient per pantalla. Emeten una senyal bluetooth indicant la seva presència a tota mena de dispositiu que hi hagi a la vora.
- **LTE Beacon:** Poden parlar directament amb el Cloud d'Estimote, es poden posicionar a l'aire lliure a través del GPS i altres satèl·lits i poden detectar altres beacons bluetooth.

#### 5.2.4 Apiary

Apiary és un editor per APIs que permet crear documentació però també permet crear un simulador de servidor (Server Mock). Amb aquest Mock es poden testear les APIs en desenvolupament sense la necessitat d'un servidor real. Al mateix temps també analitza les peticions que rep amb el seu Inspector i permet comprovar si s'estan fent correctament o no. [11, 12]

Durant el projecte s'utilitzarà, en comptes d'un servidor, per poder recuperar les dades necessàries i per provar el funcionament de l'aplicació.



Figura 5.5: Logo Apiary

### 5.2.5 GitHub

Controlar les versions d'un projecte vol dir gestionar els canvis dels arxius i directoris en el temps. Permet revertir un arxiu a una versió anterior, comparar canvis en el temps, veure qui ha fet els canvis, etc. [13]

És molt útil gestionar projectes amb eines d'aquest tipus perquè si mai hi hagués algun problema en alguna màquina o problema amb el codi, és molt fàcil de recuperar.

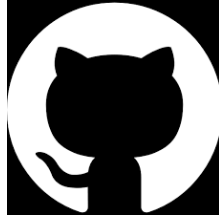


Figura 5.6: Logo de GitHub

### 5.2.6 Gradle

El Gradle és una eina que permet l'automatització de compilació de codi obert i és el sistema de compilació oficial per Android. Té una gran flexibilitat, compta amb un sistema de gestió de dependències molt estable i permet utilitzar altres llenguatges, apart de Java. [14, 15]

Des del punt de vista dels projectes, és el fitxer on s'especifiquen les dependències o llibreries que es volen utilitzar.



Figura 5.7: Logo de Gradle

### **5.2.7 Android SDK**

Un SDK (*Software Development Kit*) és un conjunt d'eines de desenvolupament de software que permeten a un desenvolupador de software crear una aplicació informàtica per un sistema concret. [16]

En el cas d'Android, inclou diverses APIs subministrades per Google que permeten controlar les funcions del dispositiu i la integració de serveis. També inclou un depurador, un emulador per poder fer proves de les aplicacions i tota la documentació necessària per a utilitzar aquesta eina.

## **6. Requisits del sistema**

En aquest apartat es descriuen els requisits necessaris del sistema, funcionals i no funcionals. Els requisits funcionals són els serveis que ofereix l'aplicació independentment de la seva implementació. Per altra banda, els no funcionals són restriccions imposades per l'aplicació. També s'indicaran els actors que interaccionen amb el sistema.

Els requeriments funcionals estaran enumerats seguint l'estil RF – X, on X és el número de requeriment.

### **6.1 Actors**

Els actors que interaccionaran amb l'aplicació seran els alumnes, visitants, conferenciants o qualsevol persona que es descarregui l'aplicació. Només hi haurà un rol i serà el d'un usuari normal.

No serà necessari que hi hagi cap rol d'Administrador ja que el que s'encarregarà del manteniment de l'aplicació és el mateix desenvolupador i tots els usuaris hauran de poder fer les mateixes accions.

### **6.2 Requisits funcionals**

Per especificar els requisits funcionals, es farà segons les diferents accions que pot fer l'usuari en l'aplicació:

#### **6.2.1 Requisits funcionals en la gestió de permisos de Bluetooth i Localització**

En el cas de l'acceptació o rebuig de permisos, se li haurà de permetre a l'usuari acceptar o rebutjar els permisos i activació, i en el cas que els rebutgi, l'aplicació es comportarà d'una manera raonada i informarà a l'usuari per què són necessaris aquests permisos, sense forçar a acceptar-los.

Els requeriments que permetran controlar el comportament anterior són:

- **RF-1** Els usuaris podran acceptar o rebutjar l'activació del Bluetooth
- **RF-2** Els usuaris podran acceptar o rebutjar que l'aplicació accedeixi a la ubicació del dispositiu
- **RF-3** Els usuaris podran acceptar o rebutjar que l'aplicació activi la Localització.
- **RF-4** En el cas que els usuaris rebutgin l'activació de Bluetooth o els permisos de Localització, aquests podran tancar una finestra informant del per què és necessari activar-los.

### **6.2.2 Requisits funcionals en l'elecció de la destinació**

L'aplicació ha de permetre als usuaris escollir, mitjançant un llistat, l'aula, despatx o espai on volen arribar. També se'ls haurà de permetre buscar en el llistat escrivint el nom del lloc o de persona que estan buscant.

- **RF-5** Els usuaris podran clicar sobre una aula, despatx o espai de la llista per indicar el lloc que estan buscant.
- **RF-6** Els usuaris podran filtrar la llista mitjançant una barra de cerca, on només escrivint ja es filtrarà automàticament.
- **RF-7** Els usuaris podran filtrar la llista, escrivint, per nom d'aula, nom d'espai, número de despatx, nom de professor o qualsevol altre element que es trobi a la llista.

### **6.3 Requisits no funcionals**

A part dels requisits funcionals de l'aplicació, també s'han de tenir en compte els requisits que garanteixin una bona experiència pels usuaris. Aquests requeriments són els següents:

- S'ha de disposar d'un dispositiu mòbil amb sistema operatiu Android amb la versió 5.0 o posterior.
- La interfície serà desenvolupada per tal que es vegi correctament en tota mena de dispositius Android.
- S'ha de tenir accés a Internet.
- El disseny serà senzill i comprensible per tota mena d'usuaris.

- L'aplicació serà fàcil d'utilitzar per qualsevol usuari, de manera que un usuari inexpert podrà aprendre a utilitzar-la amb menys de 10 minuts.
- L'aplicació s'haurà de desenvolupar tenint en compte que haurà de ser fàcil de mantenir.
- L'aplicació només es podrà utilitzar en mode portrait, o el que és el mateix, en mode vertical.
- Tot el text de l'aplicació, que sigui possible, estarà en català i anglès, excepte la llista d'aules, espais i despatxos, on s'utilitzarà el format utilitzat per l'EPS.
- Tota la interfície no contindrà faltes ortogràfiques.
- L'aplicació en cap cas demanarà dades personals.
- L'aplicació no contemplarà cap tipus de pagament.
- Cada vegada que l'usuari obri l'aplicació es comprovarà si té els permisos de Bluetooth i Localització activats.
- El temps de resposta de l'aplicació a l'hora de detectar *beacons* serà d'uns 10 segons o menys.
- Els *beacons* estaran repartits de manera òptima, per tal de cobrir la màxima zona possible amb el menor número de *beacons* possible.
- La interacció amb els *beacons* es farà automàticament, de manera que els usuaris només hauran d'anar seguint les indicacions que se'ls hi proporcionarà.
- Les indicacions que es proporcionaran als usuaris per trobar les destinacions hauran de ser clares i senzilles d'entendre.
- L'aplicació només serà funcional en els edificis de l'Escola Politècnica Superior, de manera que si s'utilitza fora aquests no servirà.
- L'aplicació utilitzarà el magnetòmetre del mòbil per saber l'orientació del l'usuari per poder-lo guiar amb més precisió.



## 7. Estudis i decisions

En aquest apartat es fa un estudi de les possibilitats que hi havia respecte les diferents tecnologies i programari a l'hora de desenvolupar el projecte. A continuació s'expliquen en apartats les decisions preses i el seu motiu.

### 7.1 Android vs Swift

A l'hora de desenvolupar aquest projecte vaig haver d'informar-me sobre el tipus d'aplicació que estava buscant en Josep. Exactament necessitava saber què volia que fes l'aplicació, saber per quin tipus de dispositiu la volia, què hi volia mostrar, com ho volia mostrar, etc.

Va sorgir el dubte de si fer una aplicació per dispositius Android o iOS. De bon principi jo ja estava més predisposat a fer una aplicació per Android ja que ja n'havia fet el curs anterior en una assignatura i en tenia un cert coneixement. En canvi, no havia utilitzat ni vist mai codi amb Swift [17], que és el llenguatge per programar aplicacions per iOS [18] i macOS [19]. Aquest va ser un dels motius principals pels que es va descartar fer l'aplicació per iOS, a part que també vam tenir en compte que Android és més accessible a l'hora de desenvolupar-hi aplicacions i que és el sistema operatiu més utilitzat en mòbils.

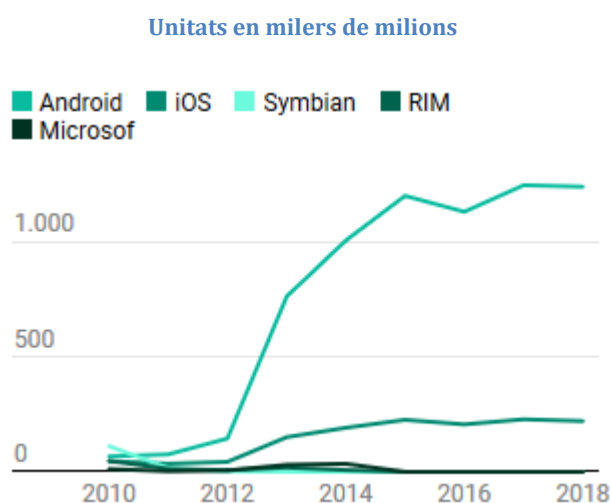


Figura 7.1: Evolució de vendes de smartphones segons el S.O. entre 2010 i 2018

### 7.1.1 Android

Com he comentat abans, Android és un sistema operatiu mòbil desenvolupat per Google, basat en el Kernel de Linux i altre software de codi obert que es programa amb el llenguatge Java.

#### Conceptes importants

- **Activity:** Una Activity és una “pantalla”. En una aplicació com a mínim n’hi ha una.
- **Fragment:** Part d’una Activity. Encapsula el codi per tal de poder-lo reutilitzar.
- **View:** Objectes que seguint una jerarquia acaben formant una activity
  - Es creen mitjançant fitxers xml o amb codi font
  - Proporcionen eines com ara: etiquetes, imatges, text d’entrada, botons...
  - Aporten events a una activity, com ara: clics, arrossegar elements, etc.
- **Fitxer AndroidManifest.xml:** Fitxer que conté les dades principals de l’aplicació, com el nom, el tema, l’icona, el nom de les activitats que hi ha a l’aplicació, l’activity inicial...
- **Service:** Component no visible que s’utilitza quan no hi ha una activity visible de l’aplicació.
  - S’utilitzen per tasques de llarga durada al fons, com ara: tocar música...
- **Intent:** Representen accions que es duran a terme per un altre component
  - Es poden enviar per començar una activity, un servei, etc.
- **Layout:** Especifica l’estructura visible d’una activity. Estan formats d’elements de tipus View.
- **Activity Stack:** Una aplicació és una pila d’activitats. Cada vegada que una nova activity comença, es col·loca a sobre de la que hi havia fins ara. L’activity que hi ha a dalt de tot és la visible. Per tornar enrere només cal prémer el botó de back. Si es prem el botó de “home” tota la pila se’n va al fons (background).
  - **Resumed:** Visible i executant-se.
  - **Paused:** Parcialment visible i no executant-se.
  - **Stopped:** Invisible i no executant-se.

A continuació afegeixo tres figures per deixar clar el cicle de vida de les activitats.

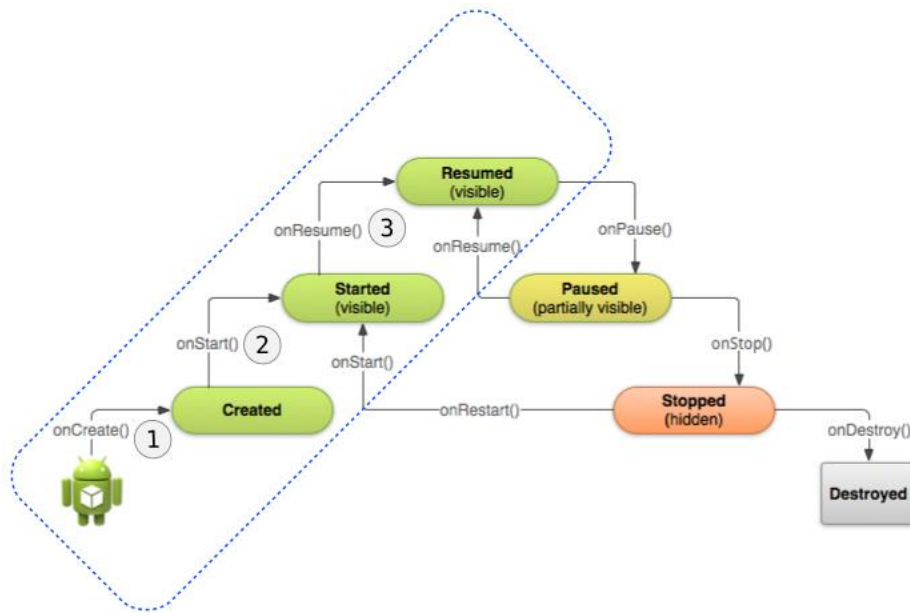


Figura 7.2: Cicle de vida de les activitats - Iniciar aplicació

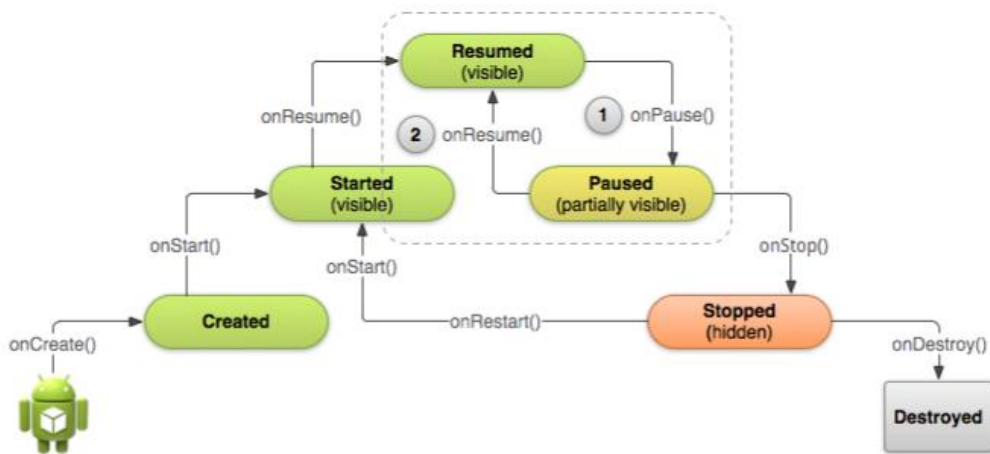


Figura 7.3: Cicle de vida de les activitats - En pausa

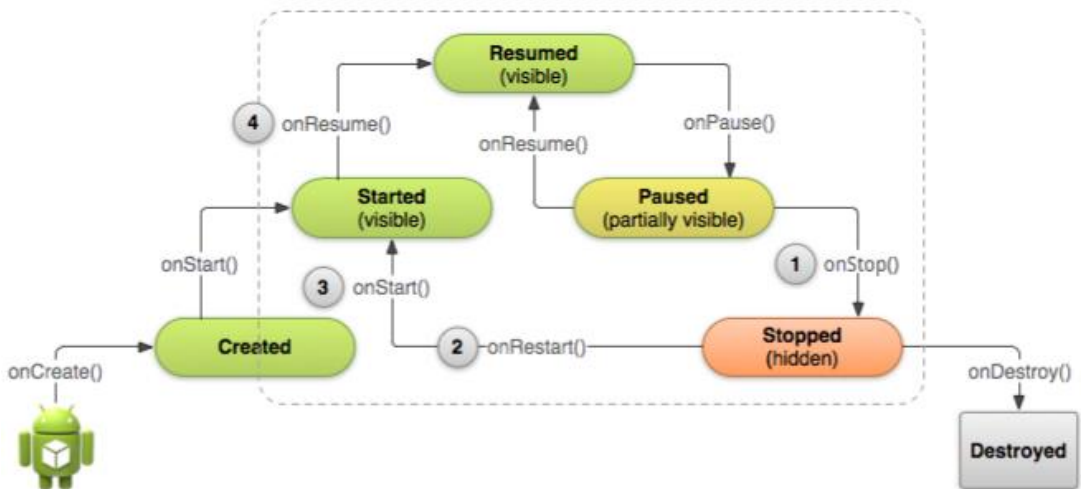


Figura 7.4: Cicle de vida de les activitats - Aturat

Cal tenir en compte que quan es prem el botó de back, l'activity es destrueix i si es torna a l'activity on s'estava abans, s'haurà perdut l'estat en què estava i es tornarà a crear de zero. Perdre l'estat vol dir que per exemple, si hi havia algun camp on s'havia d'escriure i s'hi havia escrit alguna cosa, si llavors es fa back i després es torna a l'activity on s'estava escrivint, el que s'havia escrit fins llavors s'haurà "perdut".

Per això, hi ha vegades que és molt important guardar l'estat de l'activity, per tal de no perdre informació.

L'ordre que es seguiria per guardar-lo és el següent:

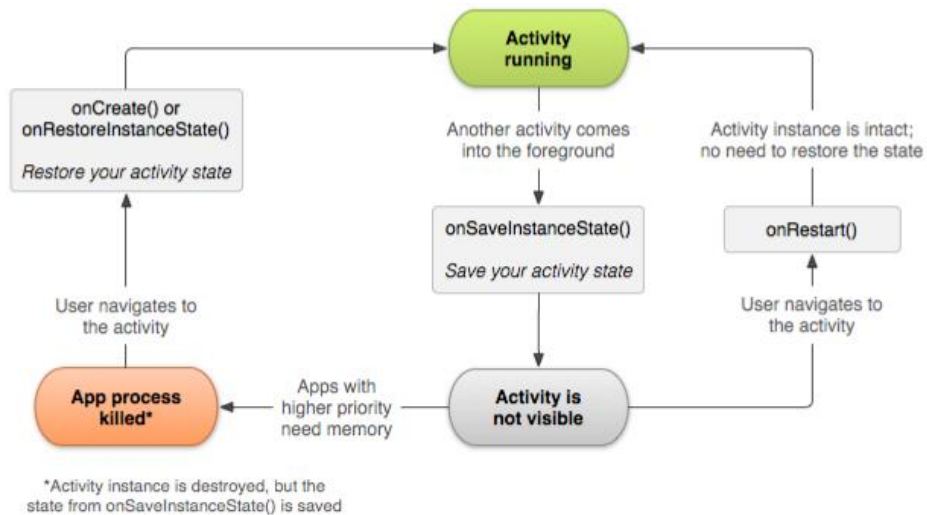


Figura 7.5: Representació de com guardar l'estat d'una activity i com recuperar-lo

## **7.2 Programes & editors utilitzats**

En aquest apartat s'explicarà el programari utilitzat al llarg del projecte el motiu pel qual es van escollir.

### **7.2.1 Android Studio**

Hi ha molts entorns per desenvolupar aplicacions per Android, però va ser fàcil escollir, ja que vaig decidir utilitzar l'oficial per aquests sistemes, que és Android Studio.

Android Studio és un entorn de desenvolupament per Android que suporta els llenguatges Java i XML.

Inclou moltes utilitats que fan que programar sigui molt més còmode, com ara compilar automàticament just després d'escriure alguna cosa, auto-completar codi o els suggeriments a l'hora d'utilitzar variables o mètodes ja existents.

També incorpora un debuggador que permet trobar errors en el codi amb facilitat.

També, mentre es té el mòbil connectat a l'ordinador, permet enregistrar moltes accions que fa el mòbil. Amb aquesta eina es pot filtrar per diferents tipus de contingut que s'extreu del mòbil, com ara informació, alertes, errors, debuggar...

En el cas del filtrat de debuggar, si mentre s'utilitza l'aplicació es tanca inesperadament, permet veure el missatge d'error que indica el mòbil i quin tros de codi ha causat el problema.

### **7.2.2 GitHub**

En projectes com aquest va molt bé utilitzar eines de control de versions pels motius que he comentat abans.

Inicialment dubtava entre fer servir Bitbucket [20] i Sourcetree [21] o GitHub [22]. Durant la carrera havia utilitzat totes les eines anteriors, però vaig acabar utilitzant GitHub ja que és el que dominava més. Concretament utilitzaré l'aplicació d'escriptori, ja que és molt pràctica de fer servir. Una altra manera de treballar amb GitHub seria mitjançant la línia de comandes utilitzant comandes de Git [23] i afegint els canvis al repositori de GitHub.

### 7.2.3 Apiary

Durant el projecte vaig utilitzar Apiary per obtenir les dades. Apiary és un editor i simulador de servidor (Server Mock) que es troba a la web i permet fer proves en una aplicació sense la necessitat d'un servidor físic. Aquest últim motiu és pel que vaig decidir utilitzar-lo, perquè permet fer proves amb una aplicació d'una manera molt ràpida i senzilla, sense haver de passar per tot el procés de muntar un servidor i base de dades. No es va contemplar la possibilitat d'utilitzar alternatives a Apiary perquè ja hi havia treballat anteriorment i sabia tot el que podia oferir.

La funció d'editor permet escriure el format i contingut de les dades i la manera de recuperar-les fent una crida a l'API d'aquest mock.

En el meu cas he guardat les dades en format JSON [24], un format de text molt fàcil de llegir que està basat en JavaScript, que utilitza convencions semblants als llenguatges de la família C però que és completament independent de qualsevol llenguatge.

Un exemple seria el següent:

```
## Beacons EPS [/beacons]
### getBeacons [GET]
+ Response 200 (application/json)
[
  {
    "id": 1,
    "codi": "4dab0ce576be4750d5ff1d5d03956c11",
    "pis": 1,
    "edifici": "P1",
    "espais": ["I-01"]
  },
  {
    "id": 2,
    "codi": "7c2ce5af34989871ffff6329187ff338",
    "pis": 1,
    "edifici": "P1",
    "espais": ["I-02", "I-03"]
  }
]
```

Figura 7.6: Exemple de JSON

A la Figura 7.5 es pot veure com està definida una crida a aquesta API amb el “/beacons”, amb el GET s’especifica el tipus de crida que s’espera i amb el “Response 200” s’especifica el tipus de resposta que s’obindrà quan es faci la crida correctament.

Amb JSON es poden utilitzar les estructures de dades que es vulguin. En el cas de la Figura 7.5 es pot veure que si la crida es fa correctament, retornarà una llista de dos objectes els quals tenen 5 atributs cada un.

Apiary té una funció “Inspector”, que permet controlar les crides que es fan en el mock i en genera un registre:

Resource	Method	Status	Source	Time
● /camins	GET	200	95.122.5.212	8 days ago
● /beacons	GET	200	95.122.5.212	8 days ago
● /espais	GET	200	95.122.5.212	8 days ago

Figura 7.7: Funció Inspector d'Apiary

Concretament el que queda registrat és:

- Si la crida s’ha fet correctament o no (Status)
- Crida que s’ha fet (/camins, /beacons...)
- Tipus de crida (GET, POST...)
- Des d’on s’ha fet la crida (Adreça IP)
- Quant fa que s’ha fet la crida

### 7.3 Lliberies

En aquest apartat es pretén explicar les lliberies utilitzades al llarg del projecte i la seva utilitat.

#### 7.3.1 iBKS Plus

La llibreria de iBKS Plus:

- *'org.altbeacon:android-beacon-library:2.9'*

Ofereix els serveis necessaris per detectar els beacons de la seva marca, mitjançant el mètode:

- *startScan()*

A més a més permet obtenir-ne la distància a la que es troba, una vegada l'ha detectat, el nom del beacon i la seva adreça MAC, amb els mètodes:

- *getRssi()*
- *getDevice.getName()*
- *getDevice.getAddress()*

### 7.3.2 Estimote Proximity

La llibreria dels beacons Estimote Proximity:

- *'com.estimote:proximity-sdk:1.0.3'*

També ofereix els serveis necessaris per escanejar balises de la seva marca, però alhora, permet escanejar zones concretes especificant un identificador de la zona. Per començar a escanejar necessita la preparació de varis components:

- *ProximityObserver*
- *ProximityZone*

Més endavant s'especificarà la configuració que necessiten per tal d'adaptar-los al projecte.

Aquesta llibreria permet recuperar dades dels beacons que se li hagin assignat prèviament durant la configuració d'aquests. Més endavant es donaran més detalls, quan s'expliqui com preparar i configurar les balises.

Per tenir-ne una idea, per recuperar-ne les dades és necessari utilitzar una mena d'*etiqueta* que s'especificarà en la configuració del *beacon*. De manera que utilitzant el següent mètode es podria recuperar el valor de la dada especificada.

- *proximityContext.getAttachments().get("orientacioEPS/nom")*

En el cas anterior, l'etiqueta especificada a la balisa seria *orientacioEps* i la dada recuperada seria *nom*.



### 7.3.3 Retrofit

La llibreria Retrofit és un client HTTP utilitzat en Android i Java. Facilita molt la feina de connectar-se a un servei web REST traduint l'API a interfícies de Java. També facilita el tractament de dades JSON o XML, ja que els transforma en objectes de Java. Es va decidir utilitzar-la ja que ja l'havia utilitzat anteriorment i és la més pràctica de fer servir.

Permet realitzar tots els tipus de peticions:

- GET
- POST
- PUT
- PATCH
- DELETE

Com que Retrofit no té un convertidor JSON integrat per convertir objectes JSON a Java, sempre va acompanyat de llibreries de suport per a solucionar-ho. Hi ha varis tipus de llibreries de suport, segons el que es vulgui fer amb Retrofit.

En aquest cas, la llibreria de suport utilitzada amb la de Retrofit és Gson:

- *group: 'com.squareup.retrofit2', name: 'converter-gson', version: '2.3.0'*

### 7.3.4 Google Play Services Location

La llibreria de Google Play Services Location:

- *'com.google.android.gms:play-services-location:16.0.0'*

Aquesta llibreria permet recopilar informació sobre la ubicació dels usuaris:

- Latitud
- Longitud
- Velocitat
- Altitud

Apart de poder obtenir la ubicació, permet als desenvolupadors saber el tipus de moviment de l'usuari, concretament es pot saber si l'usuari està caminant, corrent o anant en bicicleta o en cotxe.

Es va decidir utilitzar aquesta llibreria per poder demanar els permisos d'ubicació i activació de la localització a l'usuari, perquè és una de les més utilitzades i és la que va permetre poder activar la localització directament des de l'aplicació sense haver-ne de sortir.

### 7.3.5 Llibreries suport Android

Una llibreria de suport d'Android utilitzada en el projecte és :

- `'com.android.support.constraint:constraint-layout:1.1.3'`

Aquesta llibreria afecta al posicionament dels elements que es veuen per pantalla. A cada element que es col·loca a la pantalla se li ha d'especificar uns valors, unes *constraints* o *restriccions*, que definiran la posició on es troba. Quan hi ha més d'un element col·locat, es solen posar *constraints* entre aquests, per tal que sempre acabin col·locats a la mateixa distància. En aquest cas, si es mou un dels dos elements, l'altre també es mourà conservant els valors assignats entre ells.

A la següent figura es poden veure els valors assignats a un element d'una pantalla:

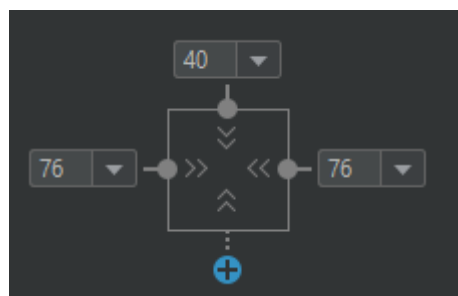


Figura 7.8: Especificació de *constraints* d'un element en una pantalla

A la Figura 7.8, es pot veure que a un element se li ha especificat una distància de 40 respecte un altre element que es troba a sobre o el límit superior de la pantalla. Respecte la distància dels costats, es pot veure que és la mateixa tant per l'esquerra com per la

dreta, per tant, indica que l'element està col·locat al mig de la pantalla. En aquest cas, aquest element no està connectat amb cap altre per la part inferior.

Aquesta llibreria és molt útil perquè permet que els elements de la pantalla es vegin ben col·locats a qualsevol dispositiu, independentment de la mida de la pantalla. En canvi, si s'utilitzen altres llibreries per col·locar els elements, poden aparèixer problemes i els elements de la pantalla es poden veure moguts en diferents dispositius. Pels motius anteriors es va decidir utilitzar-la, ja que és molt pràctic que es mantingui l'estil i disseny en varis dispositius.

## **7.6 Software complementari**

A part d'utilitzar entorns de programació, eines per testejar i llibreries, també es van utilitzar aplicacions pròpies dels beacons iBKS i Estimote per poder-los configurar i comprovar-ne el funcionament.

Aquestes aplicacions eren les oficials de les dues marques:

- iBKS Config Tool
- Estimote

Cal remarcar que d'aquestes dues aplicacions es va utilitzar la seva versió d'iOS, ja que les vaig utilitzar en el mòbil propi i no suposava cap problema a l'hora de desenvolupar l'aplicació en Android.

## **8. Anàlisi i disseny del sistema**

En aquest apartat es farà un anàlisi dels requeriments de l'aplicació per poder fer un disseny general de la interfície amb l'usuari i definir la informació necessària a guardar a la pròpia aplicació i a consultar a l'exterior a través de l'API.

### **8.1 Anàlisi del sistema**

Amb l'anàlisi del sistema es pretén fer un estudi detallat de les seves necessitats. D'aquesta manera, s'analitzaran els aspectes requerits del sistema tenint en compte les accions que podran realitzar els usuaris.

Com que tots els usuaris seran iguals i no n'hi haurà cap amb més permisos que altres, totes les accions que es podran portar a terme a l'aplicació seran iguals per tots.

#### **8.1.1 Anàlisi de la gestió de permisos de Bluetooth i Localització**

A l'iniciar l'aplicació, es comprovarà si l'usuari té activats el Bluetooth i la Localització i també si ha acceptat els permisos d'accés a la Localització. En cas que algun dels serveis anteriors no estigui activat, mitjançant una alerta, es demanarà a l'usuari que l'activi.

De l'anàlisi dels requisits funcionals, es pot extreure que els usuaris hauran de tenir la possibilitat d'acceptar o rebutjar l'activació del Bluetooth i Localització i acceptar o rebutjar l'accés a la Localització.

En cas que es rebutgi alguna activació dels serveis anteriors, mitjançant una finestra s'informarà a l'usuari del perquè són necessaris aquests permisos. L'única manera de tancar aquesta finestra serà mitjançant un botó d'acceptar.

Si en algun moment es desactiva algun dels dos serveis mentre s'està utilitzant l'aplicació, aquesta deixarà de funcionar, es podrà seguir tenint en primer pla però no reaccionarà fins que l'usuari torni a activar el servei o serveis desactivats.

### **8.1.2 Anàlisi de l'elecció de destinació**

De l'anàlisi dels requisits funcionals, es pot extreure que els usuaris hauran de poder escollir varis tipus de destinacions i filtrar-les. Podran escollir aules, que estaran especificades amb el seu nom, com per exemple: *I-01*, *II-06*, *III-12*, etc. També podran seleccionar espais que s'especificaran amb el seu nom, com ara *Secretaria*, *Sala d'actes*, etc. Finalment també hauran de ser capaços d'escollir despatxos de professors, que estaran escrits amb el número de despatx i nom de professor.

També serà necessari que els usuaris puguin filtrar la llista per facilitar la cerca del lloc que vulguin trobar.

Les dades necessàries que caldrà guardar respecte una destinació són el nom i la identificació del *beacon* més proper a aquesta, que servirà per saber el dispositiu final d'un camí i per tant, l'arribada a una destinació.

### **8.1.3 Anàlisi dels beacons**

Els usuaris interactuaran de manera indirecta amb els *beacons*, ja que serà l'aplicació mateixa qui es comunicarà amb aquests. Aquesta comunicació es farà automàticament i no serà visible per part de l'usuari i no caldrà que aquest especifiqui res.

Sobre una balisa, serà necessari poder recuperar-ne la seva identificació, l'edifici i pis on es troba i la llista d'espais que tindrà a la vora.

### **8.1.4 Anàlisi dels camins**

De la mateixa manera que la comunicació amb els *beacons* es farà automàticament, l'obtenció dels camins i de les indicacions també ho serà.

D'un camí serà necessari poder-ne obtenir totes les balises que el formaran i les indicacions per anar d'un punt a un altre.

## 8.2 Disseny del sistema

Després d'analitzar les necessitats del sistema, és necessari trobar la manera d'implementar l'aplicació que les satisfaci. El disseny del sistema té com a objectiu obtenir les solucions que permetin fer-ho.

### 8.2.1 Disseny del model de dades

Com que durant el desenvolupament de l'aplicació les dades es recuperaran d'Apiary, primer de tot es dissenyarà el sistema segons el model de dades.

El model de dades que s'utilitzarà durant el desenvolupament es pot veure en el diagrama Entitat-Relació següent:

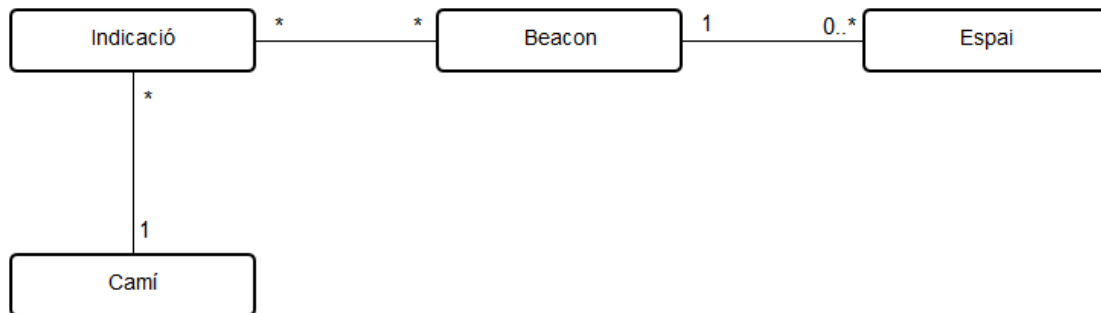


Figura 8.1: Diagrama Entitat-Relació

En aquest diagrama només s'hi ha representat les dependències que tindran les entitats entre si i, com es pot apreciar, s'ha creat una entitat diferent per cada element principal de l'aplicació.

A mesura que es van guardant camins, l'entitat *Camí* es va complicant i dificulta la distinció dels diferents camins, per això s'ha afegit l'entitat *Indicació*, per controlar d'una manera més ordenada i per facilitar l'hora d'afegir noves indicacions i camins nous.

El model de dades complet és el següent:

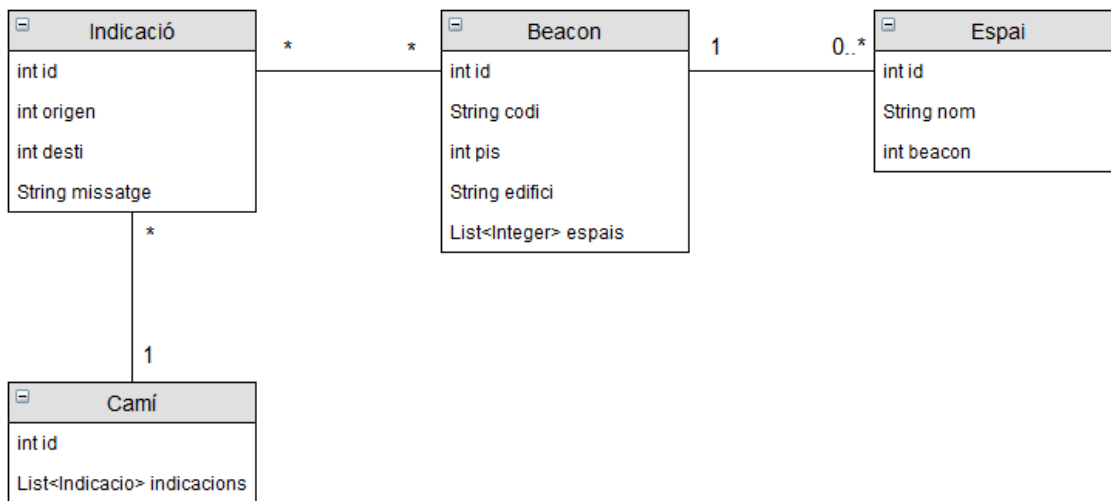


Figura 8.2: Model de dades complet

A continuació s'explicarà el contingut de cada entitat.

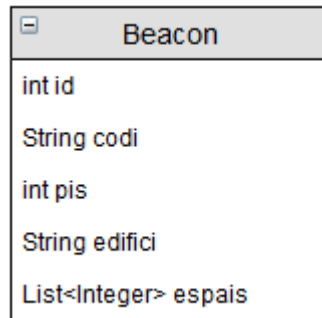
## Espai



Figura 8.3: Entitat Espai

Les dades necessàries a guardar d'un *Espai* seran la seva identificació, el nom de l'espai corresponent, per distingir entre aules, espais, despatxos i professors i es guardaran la identificació del *beacon* més proper que servirà per a guiar els usuaris fins a un espai.

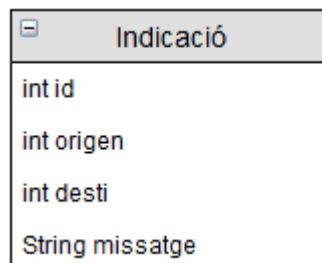
### ***Beacon***



**Figura 8.4:** Entitat *Beacon*

Les dades necessàries a guardar sobre un *Beacon* seran la seva identificació, el seu codi per poder distingir-lo quan s'escanegi, el pis i edifici on es troba per ubicar-lo i la llista d'espais que té a la vora que servirà per poder identificar les destinacions que té més a la vora.

### **Indicació**



**Figura 8.5:** Entitat *Indicació*

Les dades que es guardaran d'una *Indicació* seran la seva identificació, la identificació de les balises origen i destí i el missatge que servirà per indicar com anar del *beacon* origen al destí.



## Camí

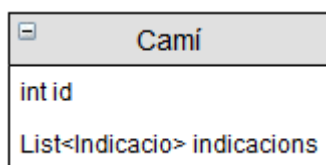


Figura 8.6: Entitat Camí

Les dades que es guardaran d'un *Camí* són la seva identificació i una llista d'indicacions que serviran per indicar a l'usuari com anar del punt on es trobi fins a la destinació que haurà escollit.

### 8.2.2 Disseny de la interfície d'usuari

Inicialment no es van especificar requisits de disseny, excepte la possibilitat de deixar escollir a l'usuari la seva destinació. La resta del disseny de la interfície es va fer després d'haver fet unes quantes proves amb les diferents tecnologies que es podien utilitzar durant el projecte, ja que d'aquesta manera es tenia una idea general del que es necessitaria representar a l'aplicació. Tot i així, el disseny es va anar comentant amb en Josep i es va anar adaptant a les necessitats que em comentava.

El disseny inicial de la primera pantalla era el següent:



Figura 8.7: Disseny de la interfície d'inici

Amb aquest disseny es pretenia que a l'obrir l'aplicació aparegués directament la possibilitat d'escollir alguna destinació i també de poder filtrar-ne el contingut.

Apart de mostrar el disseny anterior a l'iniciar l'aplicació, també es comprovava si el Bluetooth i Localització estaven activats. En cas que no ho estiguessin, mitjançant una alerta, que no es podia modificar ja que són alertes per defecte del sistema, es demanava a l'usuari per activar-los. Si l'usuari rebutjava es va dissenyar una alerta informant que Bluetooth i Localització eren necessaris pel bon funcionament de l'aplicació.

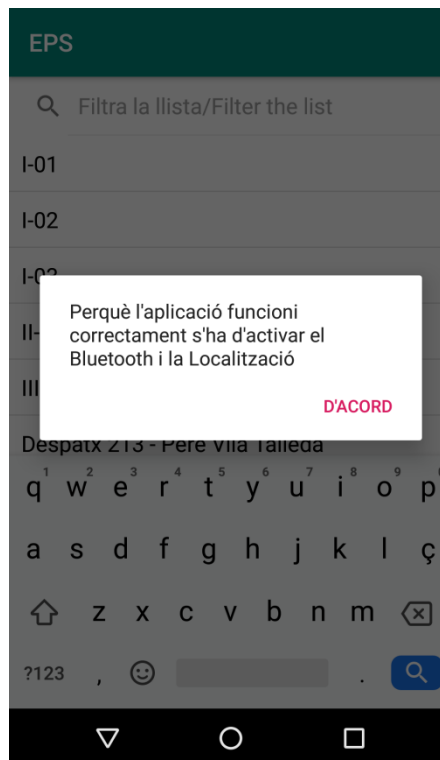


Figura 8.8: Disseny de l'alerta informant de Bluetooth i Localització

En cas de seleccionar alguna destinació a la pantalla inicial, es canviava a la pantalla on l'usuari rebia les indicacions per trobar la destinació. Abans de mostrar-la, es va pensar en afegir-hi una alerta explicant què hauria de fer l'usuari per col·locar-se en l'orientació correcta i poder seguir les indicacions correctament.

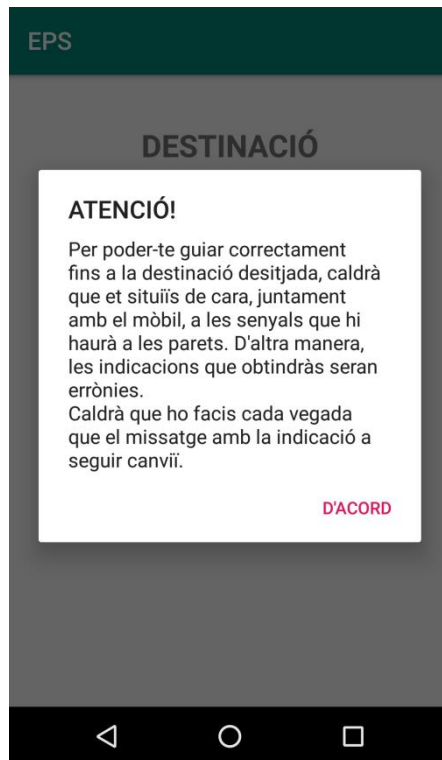


Figura 8.9: Disseny de l'alerta explicant com utilitzar l'aplicació

Després d'acceptar l'alerta de la Figura 8.9, es mostrava la pantalla on s'especificaven les direccions a seguir.

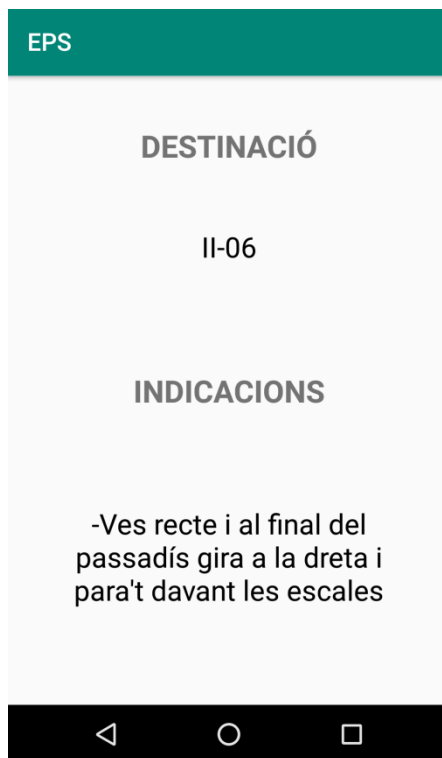


Figura 8.10: Disseny de la interfície amb les indicacions

Aplicació mòbil/IoT per orientar-se a l'interior dels edificis de l'EPS

Com que tampoc es va especificar cap tipus de requisit pel logotip de l'aplicació, provisionalment es va deixar el logotip per defecte que es genera amb l'aplicació.



**Figura 8.11: Disseny del logotip de l'aplicació**

## **9. Implementació i proves**

En aquest apartat es pretén exposar els problemes trobats durant el desenvolupament de l'aplicació i la seva solució. Els problemes principals durant la implementació han estat degut al desconeixement de la tecnologia dels *beacons* i al desconeixement de com fer peticions de permisos als usuaris de l'aplicació.

Un altre problema trobat durant el desenvolupament, va ser causat per no trobar aplicacions de brúixoles que fossin prou precises per poder-ho utilitzar.

Apart d'exposar els problemes, també s'explicarà com s'han preparat els entorns de desenvolupament.

Cal remarcar que al llarg del projecte s'han utilitzat dos tipus de balises diferents, els iBKS Plus i els Estimote Proximity Beacons.

Per utilitzar els primers no calia preparar-los ni configurar-los prèviament i com que pel desenvolupament final del projecte es van acabar utilitzant només els segons, en aquest apartat només es comentarà la preparació i configuració dels Estimote. En aquest apartat també es comentarà perquè es van acabar utilitzant els *beacons* Estimote.

### **9.1 Preparació dels *beacons* Estimote**

En un primer moment, per familiaritzar-se amb els *beacons* Estimote, es va descarregar l'aplicació mòbil que recomanava Estimote. Aquesta aplicació oferia la possibilitat d'escanejar *beacons* a través d'un radar o una llista, que indicaven els dispositius que estaven més a la vora.

Mitjançant aquesta aplicació es pretenia veure quina informació podia recuperar de cada balisa per després poder-la utilitzar a l'aplicació del projecte.

## Aplicació mòbil/IoT per orientar-se a l'interior dels edificis de l'EPS

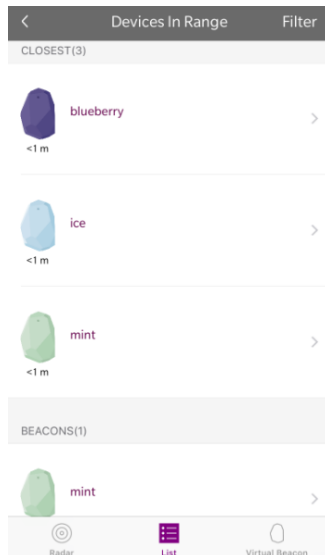


Figura 9.1: Dispositius escanejats propers en format de llista

Al fer clic sobre un *beacon* de la llista, es podia accedir a les dades del *beacon*, però no se'n podia modificar cap perquè, el *beacon* en qüestió, no estava connectat al Cloud d'Estimote [25]. Al mateix temps, tampoc podia saber quines funcions del *beacon* estaven habilitades i quines no pel mateix motiu.

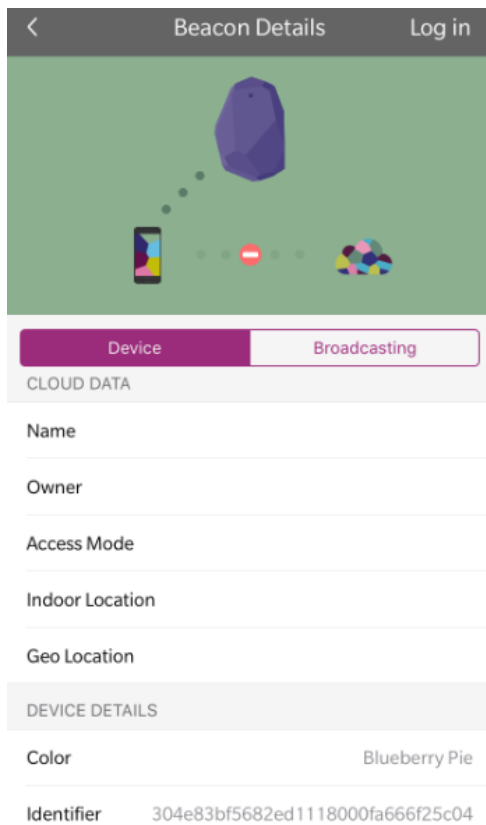


Figura 9.2: Característiques d'un *beacon* - detalls

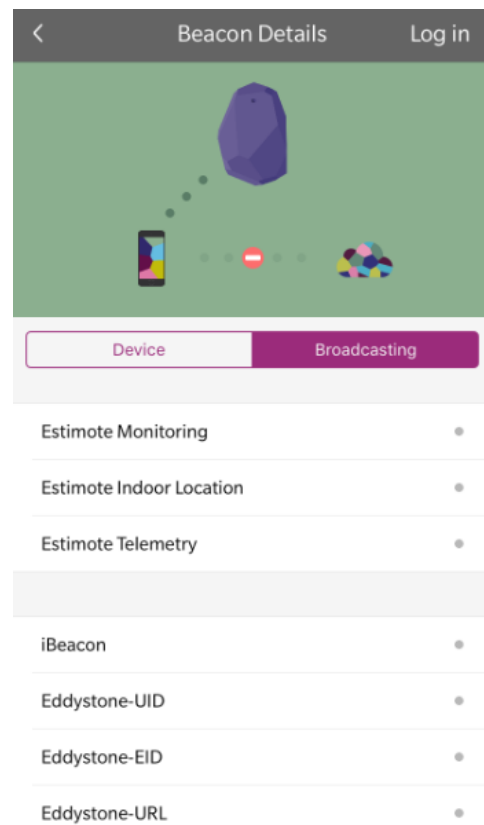


Figura 9.3: Característiques d'un *beacon* - funcionalitats

Per intentar solucionar aquest problema, calia crear un compte al Cloud d'Estimote a través de la seva web i utilitzar-lo per connectar-se a l'aplicació mòbil. Per connectar-se amb l'aplicació, com es pot veure a les Figures 9.2 i 9.3, només calia fer clic a *Log in* i entrar amb el compte d'Estimote.

Si es disposa de més d'una balisa, no cal anar seleccionant cada dispositiu i connectant-se amb cada un, només s'ha de fer *Log in* una vegada i serveix per tots.

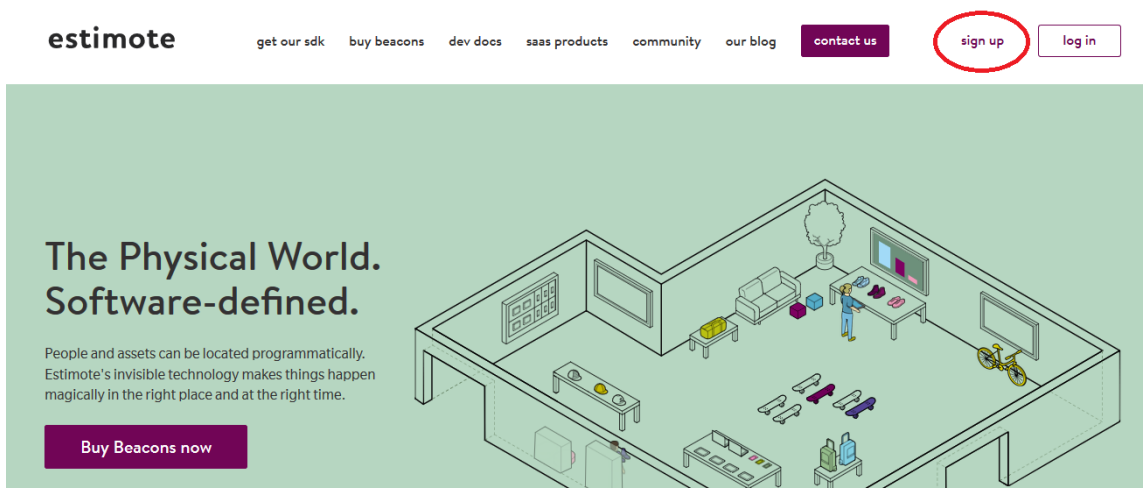


Figura 9.4: Creació d'un compte a Estimote

Una vegada connectat, es va comprovar que el problema no s'havia solucionat, ja que em seguia indicant que el *beacon* no estava connectat al Cloud, sinó que només havia connectat l'aplicació amb el Cloud, i que per tant no en podria modificar la seva configuració.

Per comprovar si es necessitava alguna configuració prèvia, es va accedir al Cloud des de la web i es va comprovar la secció de *beacons* que hi apareixia:

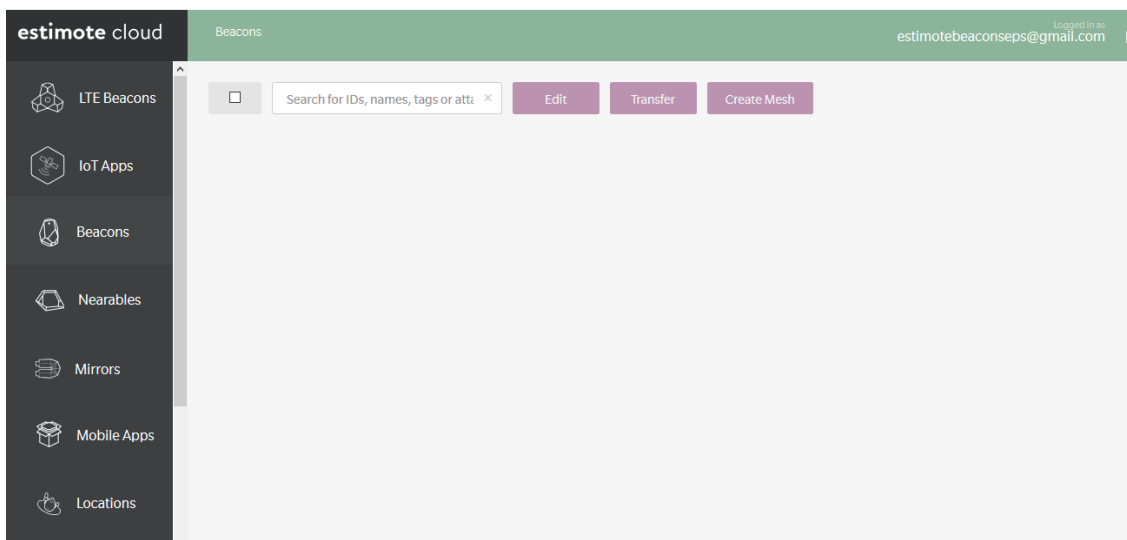


Figura 9.5: Secció de *beacons* al Cloud d'Estimote

Al veure que la secció estava buida, es va buscar informació al respecte i es va trobar que per poder veure els *beacons* al Cloud propi, calia ser-ne el propietari.

### 9.1.1 Problemes amb la propietat dels *beacons*

Per solucionar aquest problema, la documentació especificava que el propietari era l'usuari que els havia comprat, ja que en el moment de comprar nous *beacons*, la mateixa companyia Estimote, enviava un correu al client, a la direcció de correu especificada durant la compra, per tal que aquest pogués apropiar-se dels nous *beacons*. Com que els dispositius no els havia comprat jo, es va demanar a en Josep Blanes que me'ls transferís, juntament amb una explicació amb les diferents maneres de transferir un *beacon* a algú, per tal que utilitzés la que li anés millor. [26]

Les tres maneres possibles eren les següents:

- L'usuari que havia comprat els *beacons*, havia d'anar al seu Cloud i a l'apartat de *beacons*, seleccionar els que volgués transferir i clicar sobre l'opció de *Transfer*. Després calia especificar l'adreça de correu del nou propietari i, automàticament, aquest rebria un o varis correus, segons els *beacons* transferits, amb un missatge per tal de confirmar que en volia ser el propietari.

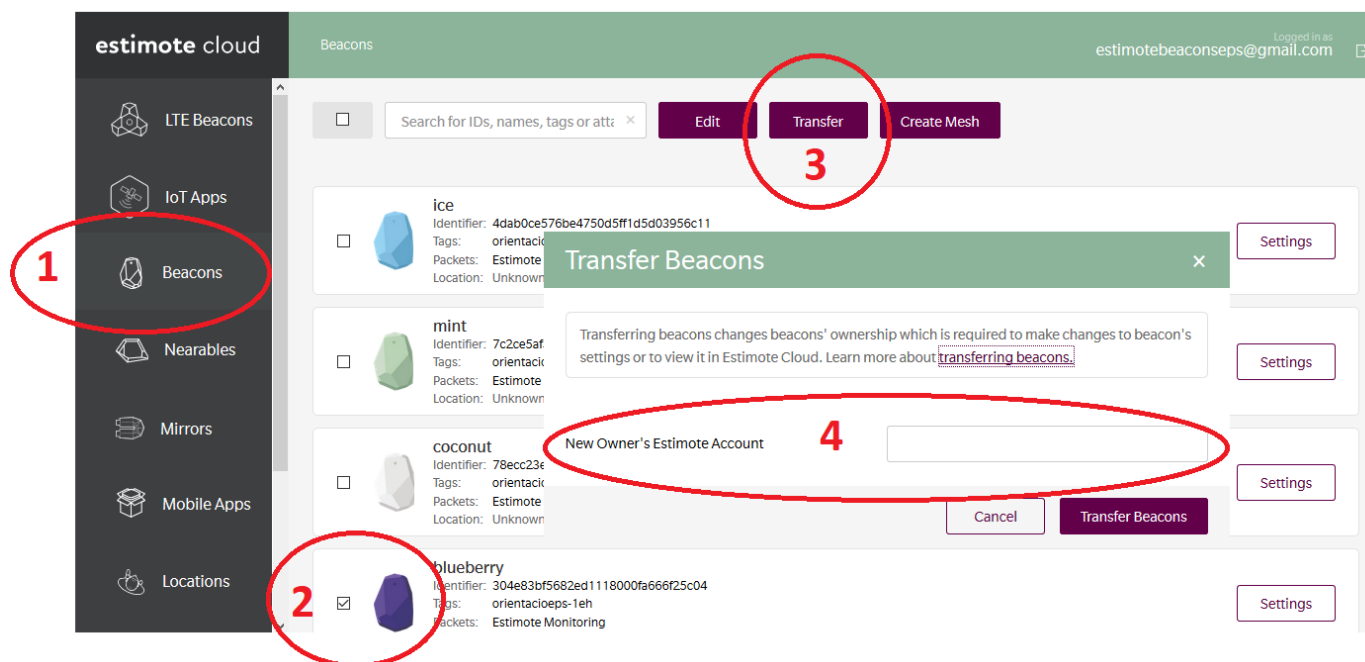
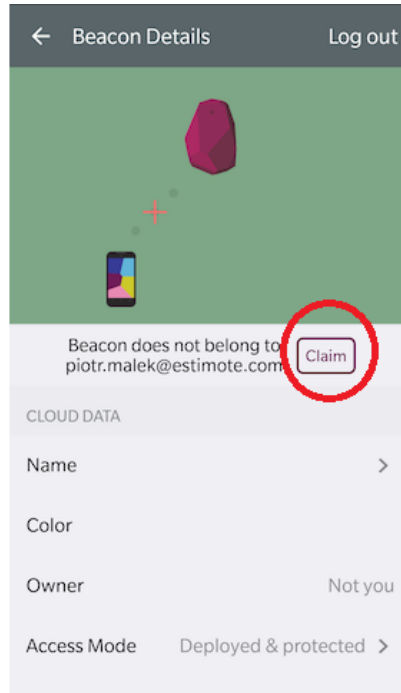


Figura 9.6: Com transferir una balisa a un usuari concret

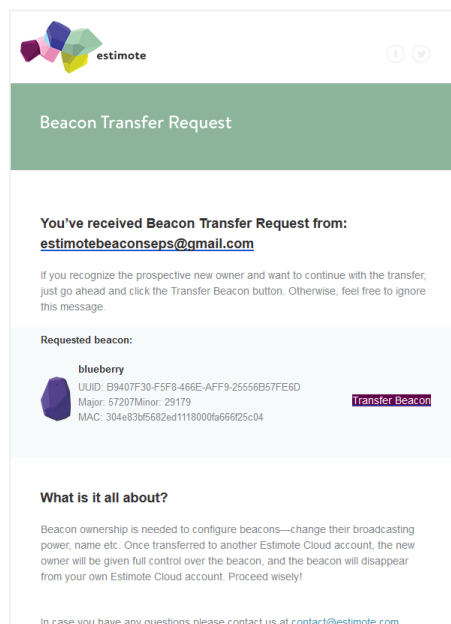


- Mitjançant l'aplicació mòbil d'Estimote, calia escanejar els *beacons* propers i seleccionar el que volguéssim. Una vegada fet això, amb un missatge s'indicava que el *beacon* no era propietat meua i amb botó *Claim* donava la possibilitat de reclamar-ne la propietat.



**Figura 9.7: Botó per reclamar la propietat del beacon**

Aquest botó envia un correu al propietari del *beacon* informant que algú l'ha reclamat i en el mateix missatge del correu dóna la possibilitat de transferir el *beacon* en qüestió a l'usuari que l'hagi reclamat.



**Figura 9.8: Missatge rebut pel propietari quan algú reclama algun dels seus *beacons***

- L'última manera possible d'obtenir la propietat d'un *beacon* era contactar amb Estimote mitjançant correu electrònic i especificant les següents dades:
  - Adreça de correu a qui pertany el *beacon*
  - Identificador del *beacon*
  - Adreça de correu on transferir el *beacon* i que estigui associada al Cloud d'Estimote

## 9.2 Configuració dels *beacons* Estimote

Una vegada ja era el propietari dels *beacons* dels que disposava i abans de configurar-los, en el mateix Cloud, calia fer un pas previ, ja que sinó no s'haurien pogut detectar els *beacons*.

Aquest pas consistia en anar a l'apartat *Mobile Apps* i a *Add New App*, després calia seleccionar el tipus d'aplicació que s'afegiria al Cloud, en aquest cas: *Proximity for Multiple Beacons*. En el següent pas s'havia de seleccionar la plataforma on s'acabaria fent l'aplicació del projecte, per tant vaig seleccionar Android desenvolupat en Java. Llavors s'havien d'especificar el nom de l'aplicació juntament amb una descripció i finalment s'acabava creant l'aplicació al Cloud.

Fer tots aquests passos era molt important ja que després de crear-se l'aplicació, es generaven una *App ID* i un *App Token*. Aquesta ID i Token, funcionen com una identificació i contrasenya que, utilitzats en el codi de l'aplicació del projecte, permetrien accedir a l'API del Cloud d'Estimote i d'aquesta manera, es podria utilitzar aquesta API per poder treballar amb els *beacons*.

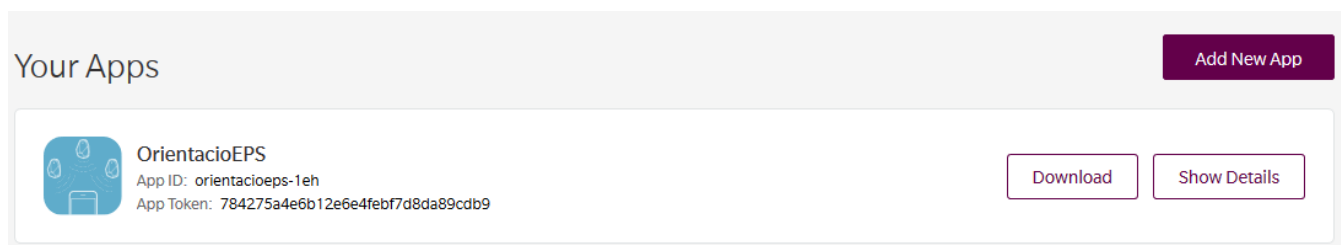
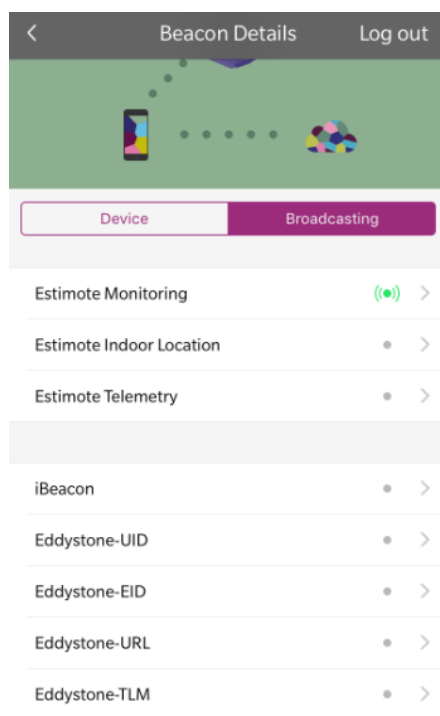


Figura 9.9: Creació d'una aplicació al Cloud, juntament amb la seva ID i Token

Una vegada creada l'aplicació al Cloud, tocava configurar els *beacons*. Per fer-ho calia anar al Cloud a l'apartat de *beacons*, tal i com es veu a la Figura 9.6, i anar a l'opció de *Settings* del *beacon* en qüestió i a *Edit Settings*.

Una vegada dins la configuració de la balisa, es va activar la funcionalitat *Estimote Monitoring* ja que permetia que la detecció de *beacons* fos més precisa, fiable i sensible. La resta de funcionalitats es van desactivar, per evitar possibles confusions a l'hora de treballar amb les balises i perquè no eren rellevants pel projecte. A part, d'aquesta manera també es va aconseguir allargar la vida dels *beacons*.

Amb l'aplicació mòbil es podia comprovar quines funcionalitats estaven activades i quines no.



**Figura 9.10: Funcionalitats d'un *beacon* – activades o desactivades**

Després d'activar i desactivar les funcionalitats, calia anar a la pestanya *Device* i canviar-ne els camps *Tags* i *Access Mode*.

The screenshot shows a configuration form for a beacon. On the left, a dropdown menu labeled 'Device' is open, listing various beacon types: Beacon Attachment, Conditional Broadcasting, Beacon Health Check, Connectivity Packet, Estimote Telemetry Packet, iBeacon Packet, Eddystone-UID Packet, Eddystone-URL Packet, and Eddystone-TLM Packet. The 'Device' label is circled in red. The main form has the following fields: 'Name' (text input with 'blueberry'), 'Geo Location' (dropdown menu), 'Tags' (text input with 'orientacioEPS', circled in red), 'Access Mode' (dropdown menu with 'Deployed & Protected', circled in red), and 'Estimote Monitoring' (toggle switch set to 'On'). At the bottom, there are three buttons: 'Hide Advanced Settings', 'Cancel', and 'Save Changes'.

Figura 9.11: Camps *Tags* i *Acces Mode* d'una balisa

Com es pot veure a la Figura 9.11, en el camp *Tags* cal especificar una etiqueta que més endavant servirà perquè l'aplicació del projecte pugui detectar el *beacon* en qüestió. Aquesta etiqueta pot contenir qualsevol valor, mentre s'utilitzi la mateixa en la balisa i a l'aplicació que la vol utilitzar.

El camp *Access Mode*, inicialment contenia el valor *Development*, de manera que qualsevol persona que estigués a la vora i utilitzés l'aplicació mòbil d'Estimote, en podria canviar les configuracions. Per evitar això, es va canviar el valor del camp a *Deployed & Protected*, de manera que només el propietari de la balisa, tingué accés a les configuracions.

Si s'anava a la pestanya *Beacon Attachment*, es podien especificar una mena d'atributs propis del *beacon*, mitjançant el *Tag* anterior indicat al *beacon* i el nom de l'atribut, de manera que quedaria de la manera següent:

- TAG/nomAtribut

Després d'especificar l'atribut, es podia assignar un valor a aquest i si es volia recuperar, només calia utilitzar el nom de l'atribut.

Després de gestionar totes aquestes configuracions, el *beacon* ja estava preparat per a poder-se utilitzar a l'aplicació del projecte.

### **9.3 Preparació d'Apiary**

Per deixar Apiary a punt per treballar no es van tenir gaires problemes ja que ja s'havia utilitzat altres vegades. Només es necessitava fer login i ja es podia començar a utilitzar. Inicialment s'havia de crear un nou projecte d'API i després a la pestanya *Editor* ja s'hi podien començar a posar les dades necessàries per l'aplicació. Els únics problemes que podien aparèixer eren problemes relacionats amb l'escriptura o el format a l'hora de definir els objectes, com ara: no posar els noms dels atributs entre comes ("), deixar-se alguna clau o parèntesis, etc.

### **9.4 Preparació d'Android Studio**

Per instal·lar Android Studio es van seguir els passos de l'instal·lador i la documentació oficial. Una vegada instal·lat, es va crear el projecte on s'acabaria fent l'aplicació mòbil. Després de crear el projecte, el mateix programa en munta l'estructura i en configura els fitxers més importants, com ara el gradle, i després demana d'instal·lar les eines SDK. Per instal·lar-les es pot fer de dues maneres:

- Descarregar les eines quan el programa ho demana i ja es configura automàticament.
- Descarregar les eines per Internet i després especificar a Android Studio la ruta on pot trobar-les, a través del menú *Tools* i a *SDK Manager*.

Amb totes les eines necessàries instal·lades, ja tenia l'entorn apunt per començar a desenvolupar l'aplicació del projecte.

### **9.5 Desenvolupament del projecte**

En aquest apartat s'explicarà com es va anar implementant el projecte i després es comentaran els problemes que es van anar trobant durant el desenvolupament.

### 9.5.1 Implementació de l'aplicació

El desenvolupament de l'aplicació es va anar fent per parts. Primer es va desenvolupar la part on es mostraria la llista d'aules, espais i despatxos a escollir pels usuaris. Per fer això, es va crear l'activity juntament amb el seu layout per definir l'estructura que tindria aquesta primera pantalla. Llavors es van afegir el format de llista en el layout per tal que el contingut d'aquesta pantalla es mostrés com una llista i un cercador a la part superior per tal de poder filtrar aquesta llista. Aquest cercador, actuava de manera intel·ligent a l'hora de fer el filtrat, ja que si s'escrivia per exemple una "a", filtraria i mostraria tots els elements de la llista que continguessin una "a".

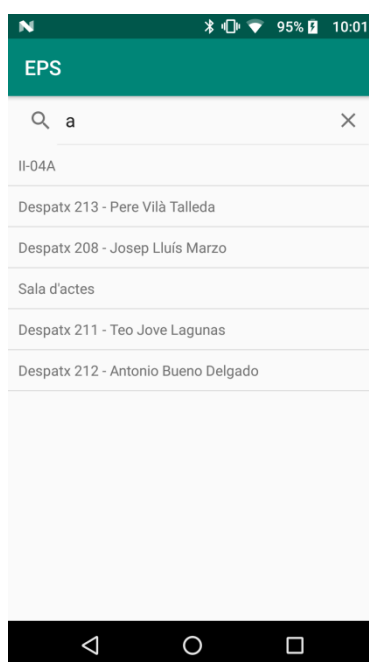


Figura 9.12: Filtrat intel·ligent

Una vegada implementat el filtrat, es va passar a desenvolupar la gestió dels permisos necessaris pel correcte funcionament de l'aplicació. Per poder treballar amb els *beacons*, l'aplicació necessitava utilitzar el Bluetooth i la Localització, mitjançant la llibreria de permisos de Google. En el cas del Bluetooth, es va implementar una alerta, que en el cas que l'usuari no tingués el tingués activat en el moment que iniciés l'aplicació, se li demanaria per activar-lo, amb la possibilitat de denegar aquesta activació.

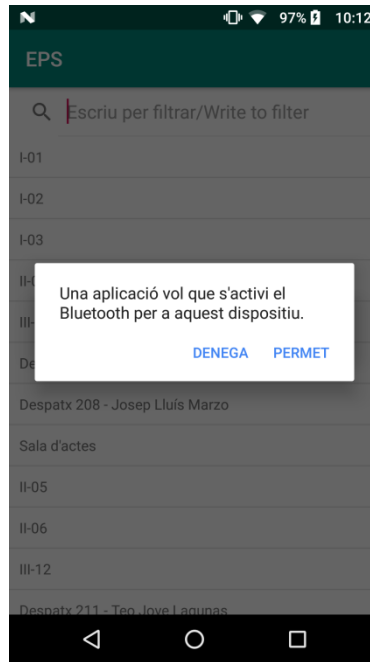


Figura 9.13: Petició d'activació de Bluetooth

En el cas de la Localització, es comportava de la mateixa manera que el cas del Bluetooth, però es demanaven dos tipus de permisos per a utilitzar-la. Aquests dos permisos eren:

- Permetre que l'aplicació tingués accés a la Localització
- Permetre que l'aplicació activés la Localització

Cal diferenciar-los, perquè si s'accepta el primer tipus de permís, l'aplicació tindrà accés a la Localització però no l'activarà, per això també és necessari demanar a l'usuari l'activació d'aquesta.

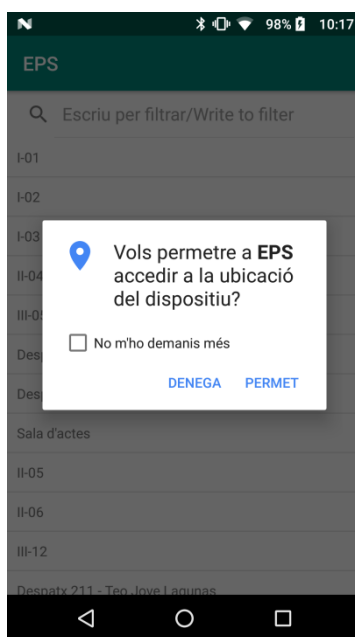


Figura 9.14: Petició d'accés a la ubicació

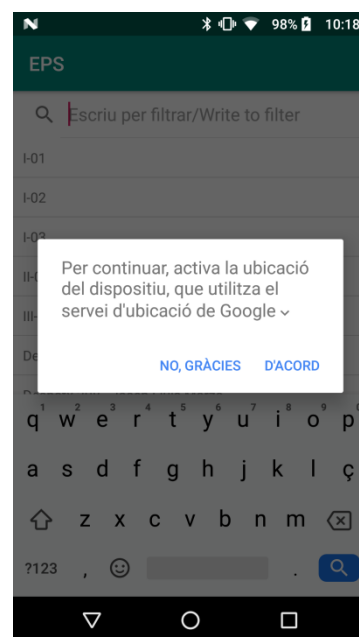


Figura 9.15: Petició d'activació de la Localització

Després d'implementar la gestió de permisos tocava desenvolupar la segona part de l'aplicació, la part on s'implementaria la interacció amb els *beacons* i posteriorment, l'orientació de l'usuari.

Inicialment, es va crear una segona activity amb el corresponent layout, on més tard es col·locarien les indicacions a seguir pels usuaris.

Primer de tot, mitjançant la llibreria d'Estimote, es van utilitzar les credencials que s'havien creat amb l'aplicació del Cloud, tal i com es comenta a l'apartat 9.2. D'aquesta manera es podia tenir accés a l'API d'Estimote i per tant, es podia començar a treballar amb els *beacons*.

El primer que es va desenvolupar va ser la detecció de *beacons* i la identificació del més proper.

L'algorisme utilitzat per controlar aquest comportament és el que es pot veure a la Figura següent:

```
public void start() {
    ProximityObserver proximityObserver = new ProximityObserverBuilder(context, cloudCredentials)
        .onError(new Function1<Throwable, Unit>() {
            @Override
            public Unit invoke(Throwable throwable) {
                Log.e("app", "proximity observer error: " + throwable);
                return null;
            }
        })
        .withBalancedPowerMode()
        .build();

    ProximityZone zone = new ProximityZoneBuilder()
        .forTag("orientacioEPS") RangeBuilder
        .inCustomRange( desiredMeanTriggerDistance: 5.0) CallbackBuilder
        .onContextChange(new Function1<Set<? extends ProximityZoneContext>, Unit>() {
            @Override
            public Unit invoke(Set<? extends ProximityZoneContext> contexts) {

                for (ProximityZoneContext proximityContext : contexts) {
                    String beaconActual = proximityContext.getDeviceId();

                    if(!seguintCami) {
                        cami = obtenirCami(numBeacon(proximityContext.getDeviceId()), espaiSeleccionat);
                    }

                    if(numBeacon(beaconActual) == obtenirUltimBeacon(cami)){
                        missatge.setText("HAS ARRIBAT. L'ESPAI QUE BUSQUES ESTÀ PER LA ZONA!");
                        missatge.setTextColor(Color.rgb( red: 0, green: 180, blue: 59));
                    }
                    else mostrarIndicacions(numBeacon(beaconActual), cami, missatge);
                }
                return null;
            }
        }) CallbackBuilder
        .build();

    proximityObserverHandler = proximityObserver.startObserving(zone);
}
```

Figura 9.16: Algorisme encarregat de gestionar la detecció de *beacons*



L'algorisme anterior s'invoca quan un usuari escull una destinació. Llavors, es construeix un *ProximityObserver* que és l'encarregat d'estar pendent si l'usuari s'acosta a alguna balisa. Després es crea una *ProximityZone*, on es produiran totes les accions necessàries per guiar a l'usuari.

En aquesta zona, cal especificar-li el *Tag* que utilitzarà i en aquest cas, s'ha d'indicar el *Tag* que s'haurà assignat prèviament als *beacons* que es volen utilitzar, tal i com s'ha vist a la Figura 9.11. Amb aquest *Tag*, s'indica al *ProximityObserver* que només estigui pendent de detectar balises que el tinguin assignat. D'aquesta manera, si l'usuari es mou en zones on hi ha altres dispositius, però aquests tenen *Tags* diferents del que s'ha indicat al *ProximityObserver*, aquest els ignorarà.

Després d'especificar el *Tag*, es comprova si l'usuari està a prop d'algun *beacon*. En cas que en tingui algun a la vora i segons la destinació escollida per l'usuari, s'obté el camí que ha de seguir per arribar al lloc desitjat. Després, es comprova si la balisa que té a la vora és l'última del camí obtingut. Si és així, s'indica a l'usuari que ha arribat a la destinació escollida. D'altra banda, es mostraran indicacions per guiar-lo fins al lloc desitjat.

Una vegada finalitzat el desenvolupament de la interacció amb les balises, es necessitava saber l'orientació de l'usuari en el moment que se li indicava una direcció a seguir, ja que sinó les indicacions que rebria podrien guiar-lo incorrectament. Inicialment, la idea era utilitzar el magnetòmetre del mòbil, per saber l'orientació d'aquest, però degut a uns problemes relacionats amb la precisió, trobats durant el desenvolupament i que s'explicaran al següent apartat, es va desestimar aquesta idea.

Finalment es va optar per informar a l'usuari, mitjançant una alerta, que abans de començar a utilitzar l'aplicació, era molt recomanable que es llegís les instruccions que es podien trobar a la pantalla inicial de l'aplicació a través del botó *INFO*.

Una vegada finalitzat el desenvolupament de l'aplicació, es va portar a terme l'última part del projecte, l'estudi de la creació de mallats intel·ligents de *beacons*. En aquesta part l'objectiu era reduir el problema d'anar a un punt a un altre, intentant economitza al màxim el número de balises i aprofitar rutes.

### **9.5.2 Problemes amb el desenvolupament de l'aplicació**

En aquest apartat explicaré els diversos problemes que em vaig trobar desenvolupant l'aplicació i les diverses solucions utilitzades per resoldre'ls.

#### **Problema a l'hora d'obtenir un dispositiu mòbil amb la versió necessària**

Encara que no sigui un problema de desenvolupament en si, el primer problema que em vaig trobar en el projecte va ser a l'hora d'obtenir un dispositiu mòbil amb la versió necessària. Com he dit, no és un problema de desenvolupament però en certa manera sí que afecta a aquest, ja que sense el dispositiu no tenia manera de fer proves del desenvolupament.

El problema va ser causat perquè com que no tenia un dispositiu Android amb la versió mínima necessària per desenvolupar el projecte, vaig passar una quantitat de temps considerable buscant un dispositiu que em permetés utilitzar-lo en el projecte. Quan en vaig aconseguir un, em vaig trobar que per desbloquejar el telèfon necessitava utilitzar el compte que de l'anterior propietari, així que vaig haver de localitzar i comunicar-me amb l'anterior propietari.

No va ser un problema trivial, ja que l'anterior propietari no recordava ni el compte ni contrasenya que havia utilitzat en el mòbil. Després de donar-me unes quantes opcions sobre quin compte podria haver utilitzat i després de varis intents intentant connectar-se al mòbil, vaig poder descobrir quin era el compte que s'havia utilitzat. Al ser una adreça de Gmail, es va poder contactar amb Google i es va reiniciar la contrasenya. Finalment vaig poder entrar al dispositiu i començar a utilitzar-lo per poder fer les proves de l'aplicació.

#### **Problema amb els *beacons* iBKS**

Al principi del projecte es van utilitzar *beacons* de la marca iBKS. Abans d'integrar-los al projecte, era necessari fer-hi proves, per veure'n les seves característiques, funcionament, rang de detecció, etc. Al veure que tenien un comportament força inesperat, es va desestimar l'opció de seguir utilitzant-los. El problema principal amb aquests dispositius era que la seva precisió de detecció era molt dolenta, concretament,

la distància respecte el *beacon* anava variant sense que el dispositiu ni l'usuari es moguessin de lloc i no s'arribava a establir en un temps raonable (un temps superior als 40 segons). Aquest problema, podia generar comportaments erronis perquè al variar la distància de detecció, podria fer creure a un usuari que estava en un lloc en concret, quan realment no hi era.

### **Problema amb l'activació de la Localització**

En el desenvolupament de la gestió de permisos de la Localització, primer es va implementar la part on es demanava a l'usuari el permís per poder accedir. Una vegada fet això, es va considerar la possibilitat d'activar la Localització automàticament a través de l'aplicació per fer-ho més còmode pels usuaris. Aquí és on va aparèixer el problema, ja que si una aplicació necessita activar algun servei del mòbil, és necessari que aquesta avisi a l'usuari que activi el servei, amb la possibilitat d'acceptar o rebutjar, ja que es necessita el seu permís.

El problema principal va ser la creació de la petició d'activació de la Localització, tal i com es pot veure a la Figura 9.15, ja que totes les llibreries que trobava per poder portar a terme això, no funcionaven correctament o simplement redirigien a l'usuari a la pantalla de configuració, que era precisament el que volia evitar. També em vaig trobar amb moltes llibreries obsoletes i vaig preferir no utilitzar-les per evitar possibles problemes de seguretat i per utilitzar llibreries que encara s'estiguessin actualitzant.

Finalment vaig trobar una llibreria de Google, que permetia fer el que jo volia i vaig poder crear la petició tal i com es veu a la figura comentada anteriorment. D'aquesta manera vaig aconseguir activar la Localització demanant-ho a l'usuari sense sortir de l'aplicació del projecte.

### **Problemes amb l'orientació de l'usuari**

Després de desenvolupar tota la part de la detecció de les balises, es va discutir la manera en què s'informaria a l'usuari com arribar a la destinació que hagués escollit. Per poder guiar correctament a les persones, era necessari saber la seva orientació en el moment que rebessin les indicacions. Per això, es va pensar en utilitzar el

magnetòmetre, que porten incorporats tots els telèfons, de la mateixa manera que l'utilitzen les brúixoles.

La intenció inicial era que mitjançant aquest magnetòmetre, es construís una espècie de brúixola que es pogués utilitzar per guiar a l'usuari mentre seguia les indicacions. Per fer això, va ser necessari informar-se de com funcionen les brúixoles a les aplicacions mòbils i posteriorment es va decidir provar-ne unes quantes, per veure amb quines s'obtenien un millor resultat.

Aquí és on va aparèixer el problema principal, la precisió de les brúixoles que es van provar era força dolenta, si les comparàvem amb una brúixola normal, es podia apreciar que hi havia una desviació al voltant d'uns 20 graus.

Vam considerar que aquesta desviació no era menyspreable, perquè segons la zona on estigués l'usuari, podria causar que anés en direccions equivocades, així que es va desestimar la idea d'utilitzar el magnetòmetre i es va decidir utilitzar un sistema de col·locar senyals a les parets. D'aquesta manera, abans de començar a utilitzar l'aplicació, mitjançant una alerta, es recomanaria a l'usuari llegir-se les instruccions, que es troben a la pantalla inicial, per tal de familiaritzar-se amb el funcionament de l'aplicació i per poder seguir les indicacions de manera correcta utilitzant les senyals a les parets.

Tots els problemes anteriors van causar que el temps dedicat a cada tasca s'allargués més de l'esperat provocant retards en la planificació.

## **9.6 Proves**

En aquest apartat es pretén explicar el tipus de proves que s'han realitzat durant el desenvolupament del projecte, i abans de donar per acabada l'aplicació, per tal d'assegurar el bon funcionament de tot el software.

### **9.6.1 Proves Apiary**

Durant el desenvolupament del projecte es va utilitzar Apiary per obtenir les dades que necessitava l'aplicació. Abans d'aplicar l'ús d'aquest al projecte, calia comprovar que

les dades es podien recuperar correctament. Per comprovar-ho es podia fer de diverses maneres:

- Des d'un navegador
- Utilitzant aplicacions externes, com ara Postman

En les dues maneres anteriors, era necessària la URL del mock de l'Api de l'Apiary preparat pel projecte. Per obtenir aquesta direcció, era necessari anar a l'apartat *Inspector* i a la part superior apareixia l'adreça següent:

<http://private-ae02a7-tfgorientacioeps.apiary-mock.com>

Mitjançant aquesta adreça i especificant la crida que es volia fer, es podien recuperar les dades de l'Apiary.

A través d'un navegador, per comprovar que podia recuperar correctament la llista d'espais, calia escriure l'adreça anterior i afegir-hi */espais*, */beacons* o */camins*, per recuperar la llista d'espais, beacons o camins respectivament.

En el cas de voler comprovar si es podia recuperar la llista d'espais calia escriure:

<http://private-ae02a7-tfgorientacioeps.apiary-mock.com/espais>

Si no hi havia cap error escrit a la llista d'espais, apareixia el següent:

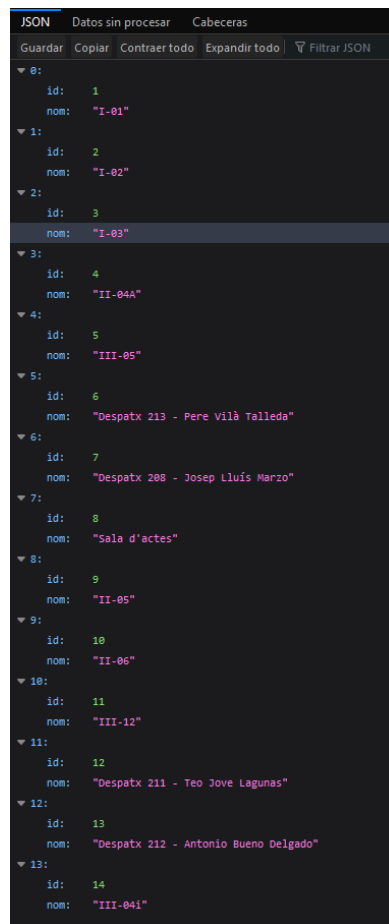


Figura 9.17: Llista d'espais recuperats d'Apiary

Una vegada s'havia comprovat que les dades es podien recuperar, es podia donar el cas que a l'aplicar l'ús d'Apiary al projecte encara hi haguessin problemes amb les crides per recuperar les dades. En aquest cas, el problema era que les dades que esperava l'aplicació i el format de les dades d'Apiary no concordaven. Per solucionar-ho, només calia corregir el format de les dades en un dels dos llocs.

### 9.6.2 Proves beacons iBKS i Estimote

A principis del projecte, es van fer varis testos de funcionament dels beacons iBKS Plus mitjançant l'aplicació oficial d'aquests. Amb l'aplicació es podien veure totes les dades de la balisa, els serveis que oferia i la possibilitat de canviar-ne la configuració. L'aspecte més important que es tenia en compte en aquests testos era comprovar si es podia obtenir la identificació d'un *beacon* i la distància a la que es trobava. Al comprovar que es podia aconseguir aquesta informació, es van començar a fer testos més específics respecte la distància de detecció d'aquests dispositius.

Després de fer aquests testos, com ja he comentat prèviament, es va poder observar que la distància de detecció no era precisa i per aquest motiu es van deixar d'utilitzar.

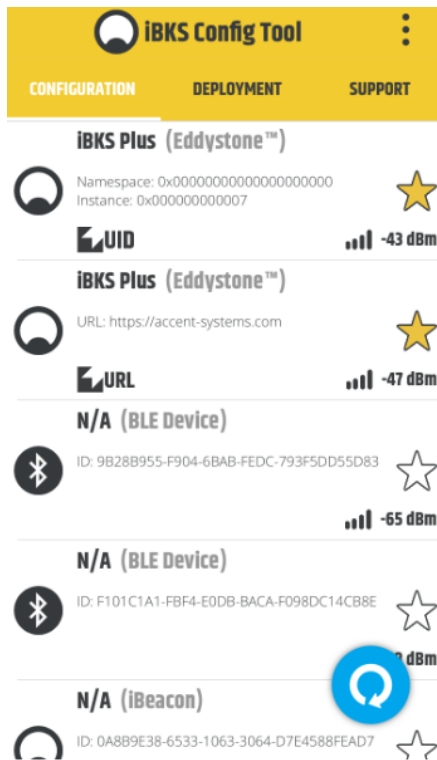


Figura 9.18: Aplicació oficial iBKS

Els següents testos que es van realitzar per comprovar el funcionament dels *beacons* ja van ser sobre els Proximity Beacons de Estimote. Amb aquests també es va utilitzar l'aplicació oficial per estudiar les mateixes característiques que amb les balises anteriors i més endavant, es va utilitzar una aplicació per a desenvolupadors per poder-ne estudiar més precisament la detecció d'aquests.

Amb aquesta aplicació, d'aspecte molt senzill com es veu a la figura següent, es comprovava la distància a la que els *beacons* apareixien o desapareixien de la llista. D'aquesta manera es va poder apreciar que els *beacons* eren molt precisos, ja que fent una simple passa endavant o endarrere, es podia detectar o deixar de detectar una balisa sense que en cap moment aparegués o desaparegués de la llista sense haver-se mogut.



Figura 9.19: Aplicació per a desenvolupadors d'Estimote

### 9.6.3 Proves indicacions del camí a seguir

Per comprovar que la funcionalitat principal de l'aplicació correctament calia preparar diverses coses:

- Primer de tot calia estudiar la zona on voldria fer la prova.
- Després caldria tenir en compte els llocs on col·locaria els *beacons*.
- Caldria construir un camí, especificant l'ordre dels *beacons* que el formarien i les indicacions que caldria seguir per anar d'un punt a l'altre. Finalment, s'hauria de guardar aquest camí a Apiary.
- Amb l'aplicació caldria escollir la destinació per la qual s'havia muntat el camí.
- Seguir les indicacions, comprovar que aquests canvien en el moment adequat i que condueixen a la destinació escollida sense dificultats.

Una vegada s'havia arribat a la destinació, calia repassar qualsevol dels punts anteriors per veure si calia millorar alguna cosa.



Aplicació mòbil/IoT per orientar-se a l'interior dels edificis de l'EPS

També, per comprovar si l'aplicació s'entenia i era fàcil d'utilitzar per tothom, vaig voler comprovar si altres persones eren capaces d'utilitzar-la i si hi trobaven alguna dificultat, m'ho anotava per tal de canviar-ho posteriorment.

## 10. Implantació i resultats

Aquest apartat té com a objectiu exposar el resultat obtingut de l'anàlisi, disseny i implementació del projecte.

### 10.1 Resultat d'interfície

En aquest apartat es mostraran les interfícies finals de l'aplicació. La majoria de desviacions de les interfícies respecte els dissenys generals han estat comentades amb en Josep i la resta de desviacions han estat modificacions que no afectaven al comportament general de l'aplicació. Totes les modificacions de disseny respecte l'original han estat fetes al final del projecte, quan l'aplicació ja estava acabada i funcionant correctament.

A la interfície final tot el text de l'aplicació apareix tant en català com en anglès, excepte el text creat pel sistema que es mostra en l'idioma que està configurat al telèfon i no es pot modificar.

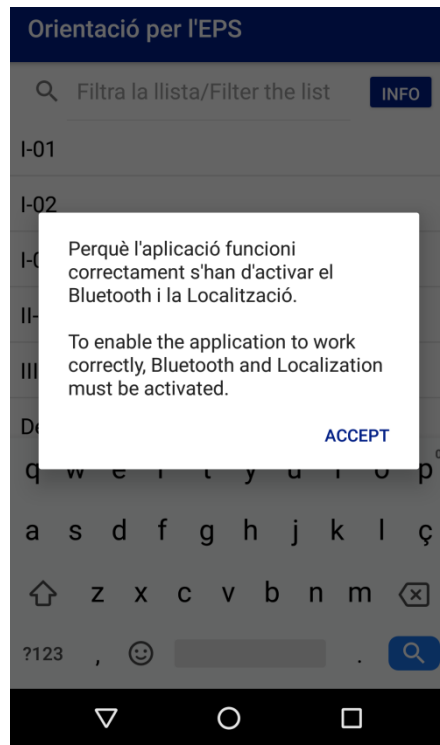
A l'entrar a l'aplicació apareix la interfície inicial, amb la llista d'espais tal i com s'havia dissenyat en el procés de disseny de l'aplicació:



Figura 10.1: Disseny final de la interfície d'inici

Tot i que al principi aquesta interfície tenia el color verd d'Android, s'ha canviat pel color blau propi de la UdG. També s'ha augmentat la mida de la lletra per millorar-ne la llegibilitat i s'ha afegit un botó amb el text *INFO* que comentaré després.

L'alerta que es mostra quan es rebutja l'activació del Bluetooth i/o Localització també ha canviat i resultat final és el següent:



**Figura 10.2: Disseny final de l'alerta informant de Bluetooth i Localització**

El missatge segueix sent el mateix que en el disseny original, però se li ha afegit la traducció en anglès i s'ha canviat el text i color del botó per tancar l'alerta.

Una altra canvi que s'ha fet a la interfície és a l'alerta que informava del funcionament de l'aplicació.

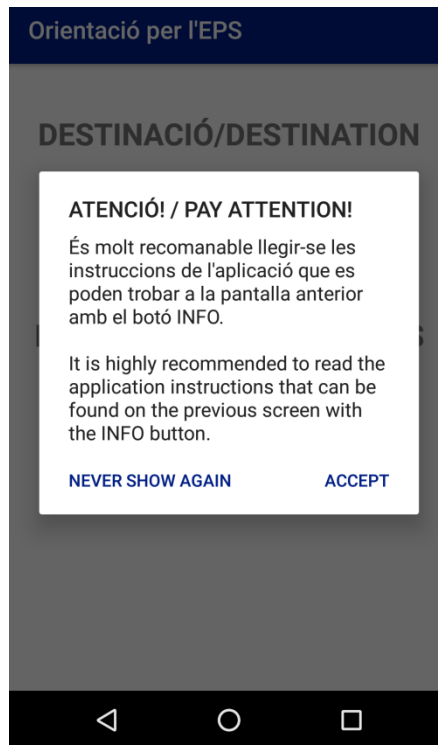
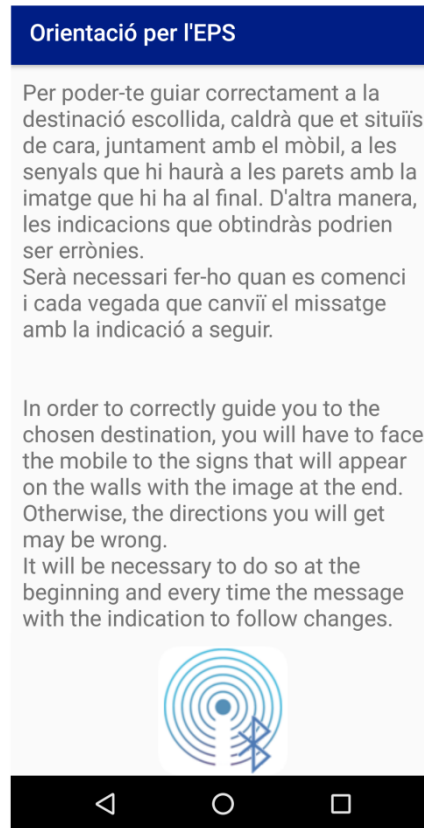


Figura 10.3: Disseny final de l'alerta explicant el funcionament de l'aplicació

Com a la interfície anterior, també s'ha fet el mateix canvi al botó d'acceptar i se n'ha afegit un de nou, un botó per no tornar a mostrar més aquesta alerta.

Abans amb aquest alerta s'explicava a l'usuari el funcionament de l'aplicació, però com que amb un sol idioma ja quedava un missatge molt llarg i encara s'hi havia d'afegir la traducció, es va canviar de lloc. En canvi, ara recomana a l'usuari llegir les instruccions del funcionament de l'aplicació, les quals es poden trobar a la interfície d'inici mitjançant el nou botó *INFO* que he comentat abans.



**Figura 10.4: Disseny de la nova interfície d'informació**

Amb aquesta nova interfície, l'aplicació ha quedat més organitzada i apart d'haver afegit el missatge en anglès, també s'ha afegit una imatge per especificar el tipus de senyal que ha de buscar l'usuari per les parets per tal d'orientar-se correctament abans de seguir les indicacions.

La interfície final on s'especifiquen les indicacions ha resultat quasi igual que l'original, excepte que s'hi ha afegit les traduccions i s'ha canviat una mica l'estil i mida de les lletres:

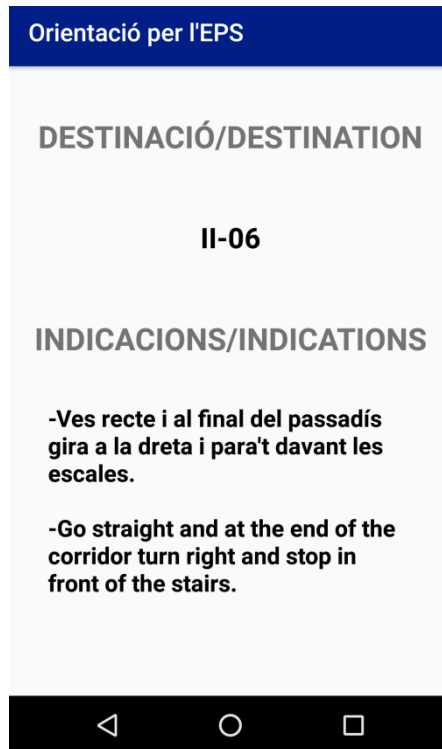


Figura 10.5: Disseny de la interfície amb les indicacions

L'únic canvi que ha patit aquesta interfície és el moment d'informar a l'usuari que ha arribat a la destinació, ja que es ressalta el missatge amb color verd, per distingir-lo de la resta d'indicacions.

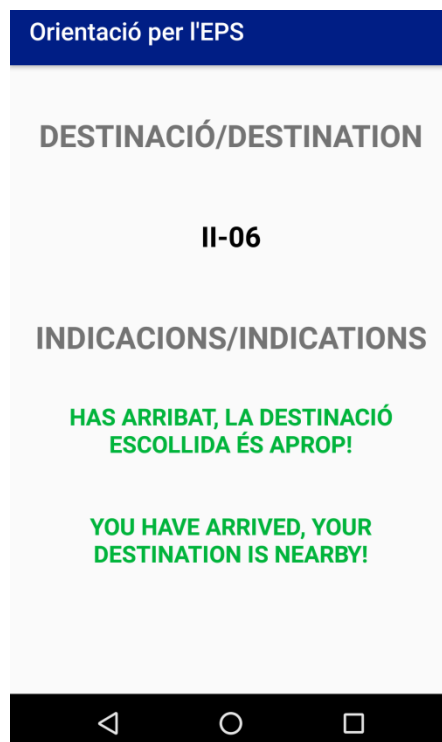


Figura 10.6: Disseny de la interfície amb indicacions amb el missatge final

Finalment, com que el logotip de l'aplicació no tenia cap disseny i s'havia deixat el logotip per defecte, he utilitzat el mateix dibuix que els usuaris han de buscar a les parets, el que apareix a la interfície d'informació, com a icona per a l'aplicació i d'aquesta manera també s'afavoreix a que als usuaris els hi resulti familiar quan estiguin buscant la senyal a la paret i així la puguin distingir més fàcilment.

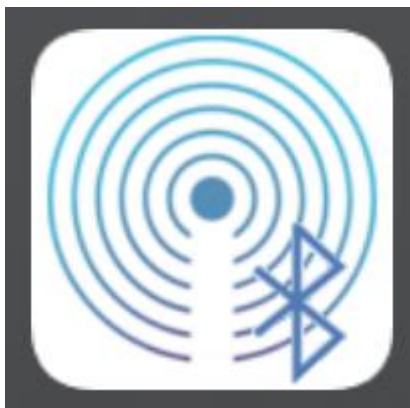


Figura 10.7: Disseny final del logotip de l'aplicació

## 10.2 Resultat final del projecte

A l'apartat *Implementació i proves* s'ha explicat tot el procés de desenvolupament i s'ha pogut veure part del funcionament final de l'aplicació. En el manual d'usuari es pot veure el funcionament complet de l'aplicació, mitjançant exemples.

El resultat final de la interfície es pot veure a l'apartat anterior, on s'han explicat els diferents dissenys.

Per fer un estudi final del resultat de l'aplicació es pot fer un anàlisi del grau d'assoliment dels requisits funcionals inicials de l'aplicació:

Requisit funcional	Grau d'assoliment
<b>RF-1</b>	✓
<b>RF-2</b>	✓
<b>RF-3</b>	✓
<b>RF-4</b>	✓
<b>RF-5</b>	✓
<b>RF-6</b>	✓

<b>RF-7</b>	✓
-------------	---

Apart d'aquests requisits, al final del desenvolupament, se'n va afegir un de nou. Aquest requisit permetia que, des de la pantalla inicial de l'aplicació, l'usuari pogués anar a una pantalla amb la informació d'ús de l'aplicació.

En el cas dels requisits no funcionals, encara que no siguin tan rellevants com els funcionals, es pot destacar que s'han complert tots excepte un. L'únic requisit no funcional que no s'ha pogut complir és el d'utilitzar el magnetòmetre, pels motius que s'han explicat en els apartats anteriors.

### Model de dades final

Al final del projecte, quan s'estava completant la informació dels camins a Apiary i fent la creació del mallat de punts, es va trobar una nova manera per millorar l'eficiència de l'aplicació i reaprofitar millor els camins existents.

Per poder implementar aquest canvi es va haver de canviar el model de dades de la manera següent:

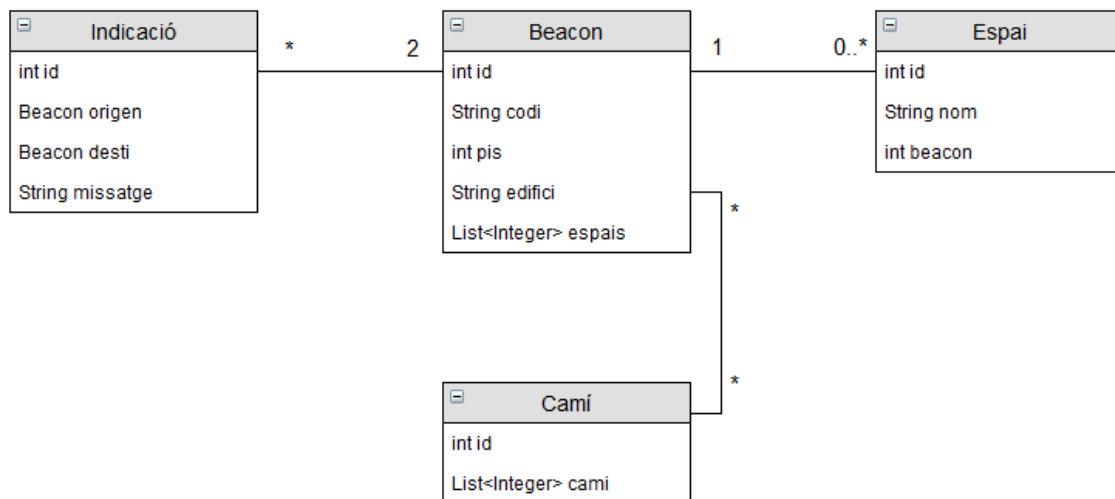


Figura 10.8: Model de dades complet final



Com es pot veure a la Figura 10.8, amb aquest canvi es va canviar la manera de tractar els camins i indicacions. A continuació s'explicarà cada canvi fet.

### Indicació

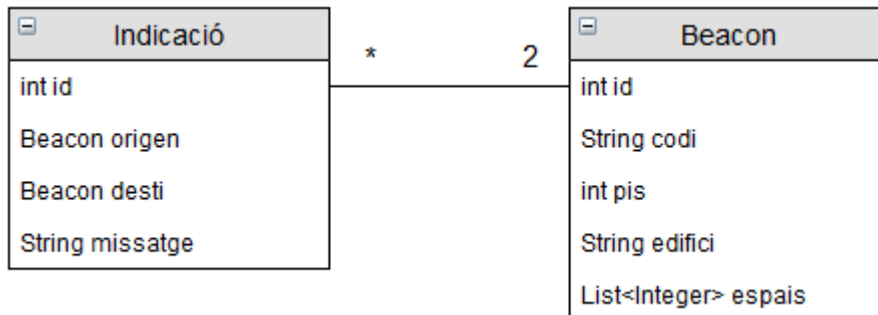


Figura 10.9: Entitat Indicació final i relació amb Beacon

Primer de tot es va canviar la relació entre *Indicació* i *Beacon*, de manera que una *Indicació* ara només està formada per 2 beacons i un *Beacon* pot aparèixer en diverses *Indicacions*. També, una *Indicació* abans només es guardava la id del beacon però ara, el guarda tot, de manera que si necessita accedir al pis o edifici el pot recuperar directament.

### Camí

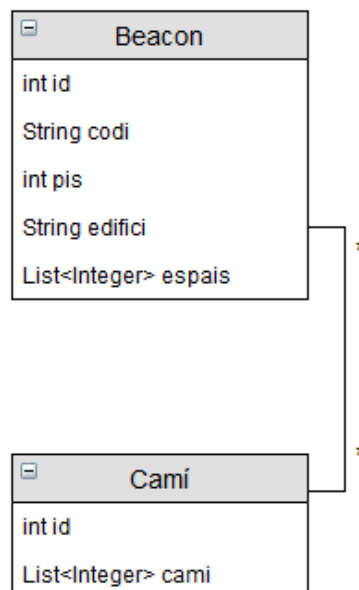


Figura 10.10: Entitat Camí i relació amb Beacon

Abans un *Camí* estava relacionat directament amb *Indicació*, ja que estava format per llistes d'indicacions. Ara un camí està format per llistes d'enters, que són les ids dels beacons, o les claus foranes, de manera que la llista de números forma el camí que es seguirà. La relació entre *Beacon* i *Camí* indica que un *Beacon* podrà estar en varis camins i que un *Camí* està format per varis beacons.

Amb aquests canvis s'ha aconseguit reaprofitar les indicacions i no s'hauran de generar tants camins com abans, de manera que es reduirà la càrrega de dades, ja que no se'n generaran tantes i augmentarà el rendiment de l'aplicació.

A continuació s'explicarà amb un exemple la millora que s'ha aconseguit amb el canvi.

Si tenim en compte el camí següent:



**Figura 10.11: Exemple de camí**

Amb el camí de la Figura 10.11 es poden formar 3 camins diferents:

- El camí per anar de l'1 al 3 (1 → 2 (A), 2 → 3 (B))
- El camí per anar de l'1 al 2 (C)
- El camí per anar del 2 al 3 (D)

A cada camí se li ha assignat una lletra per poder referenciar-lo millor a l'explicació següent.

Amb el model de dades anterior hauria estat necessari crear aquests 3 camins per separat i per cada camí s'hauria hagut de crear una indicació pròpia. D'aquesta manera s'haurien repetit indicacions, ja que en el camí per anar de l'1 al 3, primer s'ha de fer el camí A i en el camí per anar de l'1 al 2, s'ha de fer el camí C. Tant el camí A com el camí C haurien tingut el mateix missatge però serien indicacions diferents.

Amb el model de dades nou s'ha solucionat el problema d'haver de crear indicacions diferents però que tenen el mateix missatge, ja que ara són úniques.

També es reaprofiten els camins existents, sense la necessitat d'haver de crear els sub camins que hi pugui haver dins d'un camí més llarg, com en el cas de la figura anterior.

De manera que si una persona ha de fer el camí C, s'utilitzarà el camí de la Figura 10.11, ja que inclou el sub camí que es vol fer. D'aquesta manera només serà necessari crear els camins llargs i es podran aprofitar per si és necessari utilitzar els camins més petits que puguin tenir.

### **10.3 Compliment amb la legislació i normatives vigents**

L'aplicació desenvolupada en aquest projecte, no demana cap mena de dades als seus usuaris, de manera que no ha de fer cap tractament de dades privades.

Des del punt de vista de l'aplicació, es compleixen les normes relacionades amb els permisos i activació de serveis, ja que es demana a l'usuari l'accés a la ubicació i l'activació del Bluetooth i Localització i en cap cas s'hi accedeix sense el seu permís. En cas que l'usuari rebutgi algun d'aquests permisos, l'aplicació reacciona de manera adequada, informant a l'usuari el per què són necessaris aquests permisos, sense obligar a activar-los.

## **11. Conclusions**

Aquest apartat té la intenció de valorar els objectius que s'han assolit durant el Treball de Final de Grau. Al mateix temps també s'explicaran les desviacions que ha patit el projecte durant el desenvolupament i l'aprenentatge realitzat durant el projecte.

### **11.1 Valoració dels objectius assolits**

Com s'ha vist a l'apartat anterior s'han pogut complir pràcticament tots els requisits que s'havien demanat inicialment i si ens fixem en els objectius, també podem veure que s'han complert tots. Tenint en compte que s'ha pogut complir tot el que s'havia demanat i posat com a objectiu inicialment, es pot donar l'aplicació per finalitzada i per tant, que ja es pugui utilitzar.

Com s'ha comentat anteriorment, la intenció de l'aplicació era desenvolupar-la i fer proves en un entorn reduït, degut al poc nombre de balises del que es disposava. Tot i així, s'ha deixat l'aplicació preparada per a funcionar correctament quan es pugui disposar de més *beacons*, de manera que no s'haurà de fer cap canvi perquè funcioni en un entorn més extens.

Cal destacar que al final del projecte, mentre es feia un estudi per la creació de mallats a l'EPS es va arribar a la conclusió que les zones on es necessitaven més balises, eren zones on hi haguessin més destinacions o zones amb escales, ja que llavors es necessitaven més punts per poder guiar als usuaris de manera més precisa. Com per exemple el passadís del P3 o els del P4.

Les zones on es necessitaven menys balises, eren zones obertes o amb menys destinacions, ja que amb indicacions més generals es podia trobar fàcilment el punt final.

## 11.2 Desviacions de la planificació

Com s'ha pogut veure al llarg del document, durant el projecte hi ha hagut desviacions que han causat el canvis a la planificació o canvis a la manera de desenvolupar el projecte.

Les desviacions que han aparegut al llarg del projecte són les següents:

- Als inicis del projecte, la idea general de l'aplicació era tenir una pantalla amb una fletxa per indicar la direcció a seguir i que aquesta s'anés movent, mentre l'usuari es mogués, per tal de guiar-lo a la destinació. Aquesta idea es va desestimar ja que requeria construir-se una sistema de navegació en temps real, cosa que és molt complicat. Un exemple de sistema de navegació seria *Google Maps*.
- Una altra idea inicial del projecte, era construir un servidor amb la seva base de dades per tal de recuperar la informació necessària per l'aplicació. La idea es va desestimar parcialment, ja que el cotutor em va comentar que el servidor ja se'l construiria ell una vegada s'acabés l'aplicació perquè d'aquesta manera se'l podria fer a la seva manera i adaptar-lo a les seves necessitats. Per això, durant el projecte, es va utilitzar Apiary, perquè amb la seva API era suficient per comprovar el funcionament correcte de l'aplicació del projecte i una vegada acabat el projecte, només seria necessari migrar totes les dades d'Apiary al servidor.
- Una altra desviació respecte el servidor va ser que inicialment els càlculs per obtenir els camins i indicacions a seguir es farien al servidor. Finalment això no va ser així, ja que totes les dades juntes del servidor acabaven ocupant uns quants Kb. Com que les dades ocupaven molt poc, es va decidir fer els càlculs al dispositiu mòbil i d'aquesta manera també es va contribuir a augmentar la rapidesa de l'aplicació.
- Com ja he comentat abans, durant el desenvolupament del projecte van sorgir problemes amb l'ús del magnetòmetre. Com que l'ús d'aquest no era precís, degut a una gran desviació en la orientació del telèfon i com que en tots els dispositius mòbils hi ha peces diferents, es va desestimar la possibilitat d'utilitzar-lo. En el seu lloc es va decidir utilitzar senyals, com ja he comentat

anteriorment, a les parets de manera que es podria saber l'orientació de l'usuari en el moment de donar-li les indicacions.

Els problemes trobats amb el magnetòmetre, reforcen el fet que construir un sistema de navegació és molt complicat, tant pel seu desenvolupament com per tot els elements que intervenen en aquest sistema.

### 11.3 Aprenentatge assolit amb el projecte

El desenvolupament d'aquest projecte m'ha permès consolidar coneixements i conceptes tractats durant el Grau d'Enginyeria Informàtica. Concretament, els coneixements que he assolit amb el projecte són:

- Consolidació dels coneixements d'Android vistos durant el Grau.
- Reforç del coneixement d'Android Studio.
- Aprenentatge des de zero a planificar, analitzar i dissenyar un projecte real.
- Consolidació de l'ús de llibreries mitjançant el gradle d'Android.
- Aprenentatge des de zero de la llibreria *Google Play Services Location*.
- Consolidació de l'ús de la llibreria *Retrofit*.
- Aprenentatge des de zero de l'ús i gestió de *beacons*.
- Aprenentatge bàsic del funcionament del magnetòmetre d'un dispositiu mòbil.
- Consolidació de l'ús de GitHub.
- Consolidació de coneixements en la creació d'una aplicació mòbil.
- Consolidació dels coneixements per dissenyar una API REST.

La conclusió final de la realització d'aquest Treball de Final de Grau és molt positiva, ja que he pogut aprendre i consolidar coneixements que s'han vist durant el Grau i d'altres que no. Un altre aspecte molt positiu és el d'haver pogut dissenyar i desenvolupar una aplicació que resol el problema d'orientació dins la universitat. Cal destacar que l'aspecte més important del treball ha estat utilitzar tecnologies modernes com són els *beacons*. Aquest projecte es podria haver resolt d'una manera més clàssica, mitjançant cartells o senyals amb indicacions, però llavors no hauria estat interessant ja que un objectiu principal del projecte era resoldre el problema mitjançant noves tecnologies.

## 12. Treball futur

Hi ha diverses maneres de millorar l'aplicació desenvolupada. La llista següent mostra funcionalitats o característiques que es poden afegir a l'aplicació:

- 1. Orientació de l'usuari automàtica:** Seria interessant trobar la manera de saber l'orientació de l'usuari en un moment concret perquè aquest no hagi de buscar les senyals a les parets i així saber-ne la seva orientació. Tot i així, detectar automàticament l'orientació de l'usuari, podria comportar un problema nou a resoldre, que seria la possibilitat d'oferir missatges amb indicacions dinàmiques, o sigui, que el missatge amb la indicació canviï mentre l'usuari varia la seva orientació.
- 2. Desenvolupar l'aplicació per iOS:** Tot i que la majoria d'usuaris disposen d'un mòbil amb el sistema operatiu Android, estaria bé desenvolupar l'aplicació pels usuaris d'iOS ja que s'ampliaria el nombre d'usuaris que utilitzaran l'aplicació per orientar-se, de manera que una gran majoria la podria acabar fent servir.
- 3. Mostrar el text de l'aplicació en l'idioma del dispositiu mòbil:** El que tracta aquesta millora és que a l'hora d'obtenir i mostrar les dades de l'aplicació, només es faria amb l'idioma en què estaria configurat el dispositiu mòbil. D'aquesta manera, només es mostrarien els missatges en l'idioma que li interessa a l'usuari i quedaria una interfície més lleugera.
- 4. Fer estadístiques d'ús de l'aplicació:** Aquesta millora es basaria en recollir dades d'ús de l'aplicació, com ara:
  - Destinacions més escollides
  - Camins més utilitzats
  - Controlar si els usuaris tornen enrere
  - Etc.

D'aquesta manera es podrien fer estudis per millorar el funcionament de l'aplicació i oferir un millor servei amb indicacions més precises o afegir més balises en punt conflictius.

- 5. Completar els camins al servidor:** Durant el desenvolupament del treball s'ha disposat només de 4 balises, de manera que generar molts camins era complicat. En un futur, quan l'aplicació es posi en producció serà necessari afegir més

camins, juntament amb les seves indicacions, i connectar-los entre ells per oferir un bon servei.

- 6. Desenvolupar un client web:** Durant el desenvolupament de l'aplicació, les dades guardades a Apiary s'han anat afegint a mà, d'aquesta manera es podia fer proves més fàcilment. Quan es comenci a utilitzar el servidor, estaria bé implementar un client web que permetés afegir o modificar espais al servidor, d'aquesta manera no s'hauria de modificar directament la base de dades, sinó que es faria mitjançant aquest client web.
- 7. Descarregar l'aplicació:** Actualment l'aplicació no es pot obtenir de cap manera que no sigui executant el projecte directament en un dispositiu mòbil. Per això, quan es posi en producció, serà necessari afegir els mitjans per poder-la descarregar, sigui via *Google Play* o mitjançant un codi *QR*.



### 13. Webgrafia

[1] Wired, Internet of Things [en línia]. Consulta: 17 de març de 2019. Disponible a:

<https://www.wired.co.uk/article/internet-of-things-what-is-explained-iot>

[2] iBKS, Bluetooth beacons iBKS family [en línia]. Consulta: 1 d'abril de 2019. Disponible a:

<https://accent-systems.com/es/beacons/>

[3] Udacity, Developing Android apps [en línia]. Consulta: 14 d'abril de 2019. Disponible a:

<https://eu.udacity.com/course/new-android-fundamentals--ud851>

[4] Estimote, Indoor location with bluetooth beacons [en línia]. Consulta: 26 de maig de 2019. Disponible a:

<https://estimote.com/>

[5] TeamGantt, Online Gantt Chart Software [en línia]. Consulta: 3 d'agost de 2019. Disponible a:

<https://www.teamgantt.com/>

[6] Xataka, Android [en línia]. Consulta: 28 de març de 2019. Disponible a:

<https://www.xatakandroid.com/sistema-operativo/que-es-android>

[7] Wikipedia, Android Studio [en línia]. Consulta: 14 d'abril de 2019. Disponible a:

[https://es.wikipedia.org/wiki/Android\\_Studio](https://es.wikipedia.org/wiki/Android_Studio)

[8] Wikipedia, Beacon [en línia]. Consulta: 1 d'abril de 2019. Disponible a:

<https://es.wikipedia.org/wiki/Beacon>

[9] The Valley, Qué son los Beacons [en línia]. Consulta: 1 d'abril de 2019. Disponible a:

<https://thevalley.es/blog/que-son-los-beacons-y-cual-es-su-potencial>

[10] Estimote, Estimote products [en línia]. Consulta: 26 de maig de 2019. Disponible a:

<https://estimote.com/products/>

[11] Apiary, Understanding Apiary [en línia]. Consulta: 5 de maig de 2019. Disponible a:

[https://help.apiary.io/api\\_101/understanding-apiary/](https://help.apiary.io/api_101/understanding-apiary/)

[12] Apiary, How Apiary works [en línia]. Consulta: 5 de maig de 2019. Disponible a:

<https://apiary.io/how-apiary-works>

[13] GitHub, Software development platform [en línia]. Consulta: 14 d'abril de 2019. Disponible a:

<https://github.com/>

[14] OpenWebinars, Qué es el Gradle [en línia]. Consulta: 14 d'abril de 2019. Disponible a:

<https://openwebinars.net/blog/que-es-gradle/>

[15] Wikipedia, Gradle [en línia]. Consulta: XX de XXXXXXXX de XXXX. Disponible a:

<https://es.wikipedia.org/wiki/Gradle>

[16] AndroidPIT, Android para Principiantes [en línia]. Consulta: 14 d'abril de 2019. Disponible a:

<https://www.androidpit.es/sdk-android>

[17] Swift, Swift – Apple Developer [en línia]. Consulta: 28 de març de 2019. Disponible a:

<https://developer.apple.com/swift>

[18] iOS, Qué es iOS [en línia]. Consulta: 28 de març de 2019. Disponible a:

<https://es.wikipedia.org/wiki/IOS>

[19] macOS, Qué es macOS [en línia]. Consulta: 28 de març de 2019. Disponible a:

<https://www.apple.com/es/macOS/what-is/>

[20] Bitbucket, Bitbucket features [en línia]. Consulta: 14 d'abril de 2019. Disponible a:

<https://bitbucket.org/product/features>

[21] Sourcetree, Free GIT GUI for Mac and Windows [en línia]. Consulta: 14 d'abril de 2019. Disponible a:

<https://www.sourcetreeapp.com/>

[22] GitHub, The world's leading software development platform [en línia]. Consulta: 14 d'abril de 2019. Disponible a:

<https://github.com/>

[23] Git, Git [en línia]. Consulta: 14 d'abril de 2019. Disponible a:

<https://git-scm.com/>

[24] JSON, JSON [en línia]. Consulta: 14 d'abril de 2019. Disponible a:

<https://www.json.org/>

[25] Estimote, Estimote Cloud [en línia]. Consulta: 10 de juny de 2019. Disponible a:

<https://cloud.estimote.com/>

[26] Estimote, Someone bought the beacons for me [en línia]. Consulta: 10 de juny de 2019. Disponible a:

<https://community.estimote.com/hc/en-us/articles/115001822832>

## 14. Annexos

A continuació es poden trobar els enllaços per veure el codi font del projecte a GitHub i les dades utilitzades a Apiary per provar-ne el seu funcionament. També es podrà veure un índex amb les Figures del projecte.

### 14.1 Codi font del projecte

[https://github.com/Rhandom7/Treball\\_Fi\\_Grau](https://github.com/Rhandom7/Treball_Fi_Grau)

### 14.2 Apiary utilitzat en el projecte

<https://app.apiary.io/tfgorientacioeps/editor>

### 14.3 Índex de Figures del projecte

<b>Figura 3.1: Diagrama de la metodologia utilitzada.</b> .....	- 12 -
<b>Figura 4.1: Primera planificació del projecte</b> .....	- 15 -
<b>Figura 4.2: Planificació final</b> .....	- 16 -
<b>Figura 5.1: Logo d'Android</b> .....	- 17 -
<b>Figura 5.2: Logo d'Android Studio</b> .....	- 18 -
<b>Figura 5.3: Beacon iBKS</b> .....	- 18 -
<b>Figura 5.4: Beacon Estimote</b> .....	- 18 -
<b>Figura 5.5: Logo Apiary</b> .....	- 19 -
<b>Figura 5.6: Logo de GitHub</b> .....	- 20 -
<b>Figura 5.7: Logo de Gradle</b> .....	- 20 -
<b>Figura 7.1: Evolució de vendes de smartphones segons el S.O. entre 2010 i 2018</b> .....	- 25 -
<b>Figura 7.2: Cicle de vida de les activitats - Iniciar aplicació</b> .....	- 27 -
<b>Figura 7.3: Cicle de vida de les activitats – En pausa</b> .....	- 27 -
<b>Figura 7.4: Cicle de vida de les activitats - Aturat</b> .....	- 27 -

Figura 7.5: Representació de com guardar l'estat d'una activity i com recuperar-lo .....	- 28 -
Figura 7.6: Exemple de JSON .....	- 30 -
Figura 7.7: Funció Inspector d'Apiary .....	- 31 -
Figura 7.8: Especificació de <i>constraints</i> d'un element en una pantalla .....	- 34 -
Figura 8.1: Diagrama Entitat-Relació .....	- 38 -
Figura 8.2: Model de dades complet .....	- 39 -
Figura 8.3: Entitat Espai .....	- 39 -
Figura 8.4: Entitat Beacon .....	- 40 -
Figura 8.5: Entitat Indicació .....	- 40 -
Figura 8.6: Entitat Camí .....	- 41 -
Figura 8.7: Disseny de la interfície d'inici .....	- 41 -
Figura 8.8: Disseny de l'alerta informant de Bluetooth i Localització .....	- 42 -
Figura 8.9: Disseny de l'alerta explicant com utilitzar l'aplicació .....	- 43 -
Figura 8.10: Disseny de la interfície amb les indicacions .....	- 43 -
Figura 8.11: Disseny del logotip de l'aplicació .....	- 44 -
Figura 9.1: Dispositius escanejats propers en format de llista .....	- 46 -
Figura 9.2: Característiques d'un beacon - detalls .....	- 46 -
Figura 9.3: Característiques d'un beacon - funcionalitats .....	- 46 -
Figura 9.4: Creació d'un compte a Estimote .....	- 47 -
Figura 9.5: Secció de beacons al Cloud d'Estimote .....	- 47 -
Figura 9.6: Com transferir una balisa a un usuari concret .....	- 48 -
Figura 9.7: Botó per reclamar la propietat del beacon .....	- 49 -
Figura 9.8: Missatge rebut pel propietari quan algú reclama algun dels seus beacons .....	- 49 -
Figura 9.9: Creació d'una aplicació al Cloud, juntament amb la seva ID i Token .....	- 50 -
Figura 9.10: Funcionalitats d'un beacon – activades o desactivades .....	- 51 -
Figura 9.11: Camps <i>Tags</i> i <i>Acces Mode</i> d'una balisa .....	- 52 -
Figura 9.12: Filtrat intel·ligent .....	- 54 -
Figura 9.13: Petició d'activació de Bluetooth .....	- 55 -
Figura 9.14: Petició d'accés a la ubicació .....	- 55 -
Figura 9.15: Petició d'activació de la Localització .....	- 55 -
Figura 9.16: Algorisme encarregat de gestionar la detecció de beacons .....	- 56 -
Figura 9.17: Llista d'espais recuperats d'Apiary .....	- 62 -
Figura 9.18: Aplicació oficial iBKS .....	- 63 -
Figura 9.19: Aplicació per a desenvolupadors d'Estimote .....	- 64 -

<b>Figura 10.1: Disseny final de la interfície d'inici</b> .....	- 66 -
<b>Figura 10.2: Disseny final de l'alerta informant de Bluetooth i Localització</b> .....	- 67 -
<b>Figura 10.3: Disseny final de l'alerta explicant el funcionament de l'aplicació</b> .....	- 68 -
<b>Figura 10.4: Disseny de la nova interfície d'informació</b> .....	- 69 -
<b>Figura 10.5: Disseny de la interfície amb les indicacions</b> .....	- 70 -
<b>Figura 10.6: Disseny de la interfície amb indicacions amb el missatge final</b> .....	- 70 -
<b>Figura 10.7: Disseny final del logotip de l'aplicació</b> .....	- 71 -
<b>Figura 10.8: Model de dades complet final</b> .....	- 72 -
<b>Figura 10.9: Entitat Indicació final i relació amb Beacon</b> .....	- 73 -
<b>Figura 10.10: Entitat Camí i relació amb Beacon</b> .....	- 73 -
<b>Figura 10.11: Exemple de camí</b> .....	- 74 -
<b>Figura 15.1: Pantalla inicial</b> .....	- 87 -
<b>Figura 15.2: Senyal iBeacon per les parets</b> .....	- 88 -
<b>Figura 15.3: Mapa d'exemple del P1</b> .....	- 89 -
<b>Figura 15.4: Selecció de l'aula I-03</b> .....	- 90 -
<b>Figura 15.5: Primera indicació</b> .....	- 90 -
<b>Figura 15.6: Segona indicació</b> .....	- 90 -
<b>Figura 15.7: Última indicació</b> .....	- 90 -

## 15. Manual d'usuari i/o instal·lació

### 15.1 Instal·lació

En el moment de la finalització del projecte, l'aplicació no es podrà instal·lar d'una manera trivial. Per posar-la als dispositius serà necessari connectar-los a un ordinador i executar el codi font per tal que s'instal·li. Quan l'aplicació es posi en producció es facilitarà una manera perquè els usuaris la puguin instal·lar, en forma de QR o mitjançant Google Play.

### 15.2 Manual d'ús

En aquest manual d'ús, primer s'explicarà com es fa servir l'aplicació per guiar-se d'un punt a un altre i després es mostrarà un exemple amb un mapa de la planta baixa del P1 de l'Escola Politècnica Superior.

A l'iniciar l'aplicació, es demanarà l'activació del Bluetooth, la Localització i accés a la ubicació a través d'unes alertes. Es podran activar i permetre l'accés acceptant aquestes alertes. Per poder utilitzar l'aplicació per guiar-se, serà necessari tenir activats els serveis anteriors i haver permès l'accés a la ubicació.

A la pantalla inicial s'hi pot trobar una barra de cerca a la part superior (1), a la dreta d'aquesta un botó amb el text *INFO* (2) i a la part inferior una llista (3).

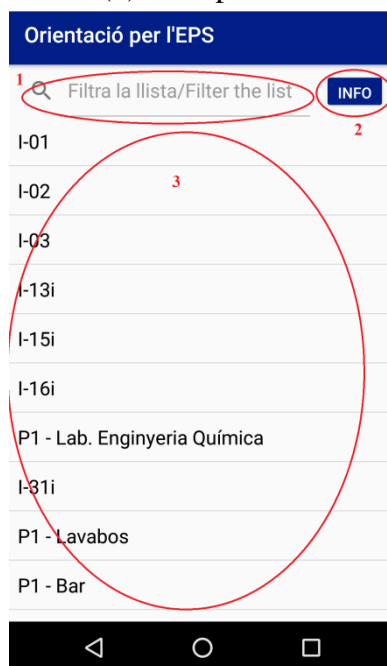


Figura 15.1: Pantalla inicial

La llista conté el conjunt de llocs on es pot arribar i que els usuaris poden escollir fent-hi un clic. La barra de cerca superior serveix per filtrar aquesta llista i a mesura que s'escriu, la llista es va filtrant automàticament sense la necessitat d'haver de prémer el botó de cercar. A l'hora de buscar els espais de la llista fent servir la barra de cerca, no és necessari escriure'ls exactament tal i com estan escrits, sinó que escrivint-ne un tros és suficient. Per exemple, si s'escriu una "p", es filtrarà la llista i només apareixeran els espais que tinguin alguna "p". En el cas d'escriure més d'un caràcter, com per exemple "des", passarà el mateix que abans i només apareixeran els espais que tinguin els caràcters "des".

Si es prem el botó *INFO* de la part superior dreta, es podran llegir les instruccions de funcionament de l'aplicació.

Estant a la pantalla inicial, si es selecciona un dels espais on es vol arribar, es canvia de pantalla. A la nova pantalla apareix una alerta, informant que és molt recomanable que els usuaris es llegeixin les instruccions de l'aplicació que es poden trobar a la pantalla anterior amb el botó *INFO*. Aquesta alerta es pot tancar amb el botó *ACCEPT* o si es prefereix, es pot fer que no aparegui més amb el botó *NEVER SHOW AGAIN*.

Amb l'alerta tancada, es pot veure que a la part superior apareix la destinació escollida i a la part inferior apareix un missatge amb unes indicacions.

Abans de començar a seguir les indicacions, és necessari situar-se de cara, juntament amb el mòbil, a les senyals que hi haurà a les parets amb la imatge de la Figura 15.2. D'aquesta manera, l'usuari es podrà orientar i seguir les indicacions correctament, altrament les indicacions podrien ser errònies.

Serà necessari col·locar-se d'aquesta manera cada vegada que el missatge amb les indicacions canviï.



Figura 15.2: Senyal iBeacon per les parets



Finalment, només cal seguir les indicacions que es van obtenint i s'acabarà arribant a la destinació escollida.

### 15.3 Exemple d'ús

A continuació es mostrarà un exemple d'ús mitjançant un mapa de la planta baixa del P1 i imatges del contingut de l'aplicació. En el mapa de la Figura 15.3 s'han marcat els punts on es troben les senyals a les parets i se'ls hi ha assignat un número per tal d'identificar-les millor.



Figura 15.3: Mapa d'exemple del P1

L'explicació següent servirà per mostrar com arribar a l'aula I-03 des de la senyal 1. Per arribar a la destinació desitjada, en aquest cas, serà necessari passar per la senyal 2 i s'acabarà a la 3.

Inicialment, serà necessari escollir l'aula I-03 de la llista:

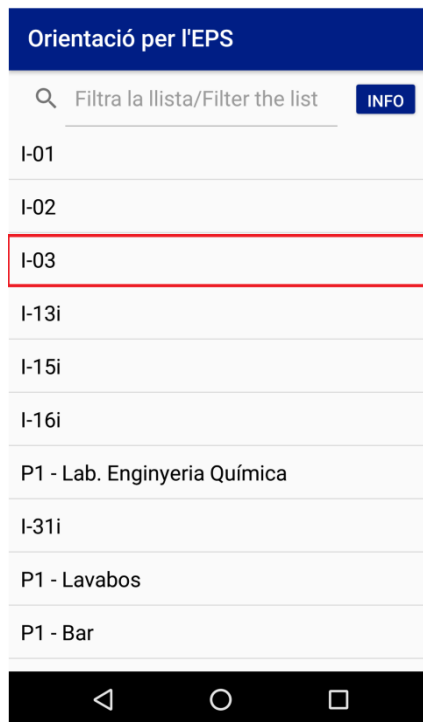


Figura 15.4: Selecció de l'aula I-03

A continuació, a la següent pantalla caldrà acceptar, per tancar l'alerta, o fer que no es mostri més. Una vegada tancada l'alerta, serà necessari posar-se de cara a la senyal amb la imatge de la Figura 15.2.

A la part inferior de la pantalla apareixerà la indicació a seguir. En aquest cas és la següent:

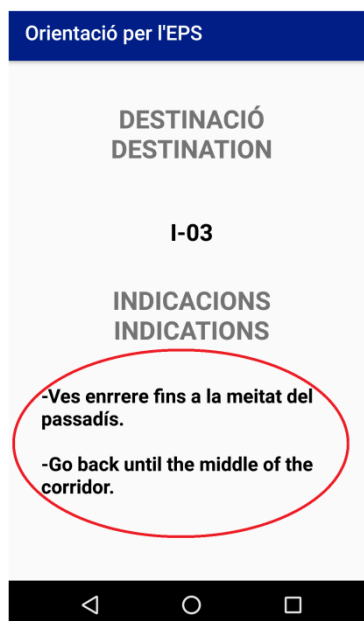


Figura 15.5: Primera indicació

Seguint la indicació, s'arriba al punt 2 de la Figura 15.3. Allà, caldrà fer el mateix i situar-se de cara a la senyal i es podrà seguir la indicació que en aquest cas és:

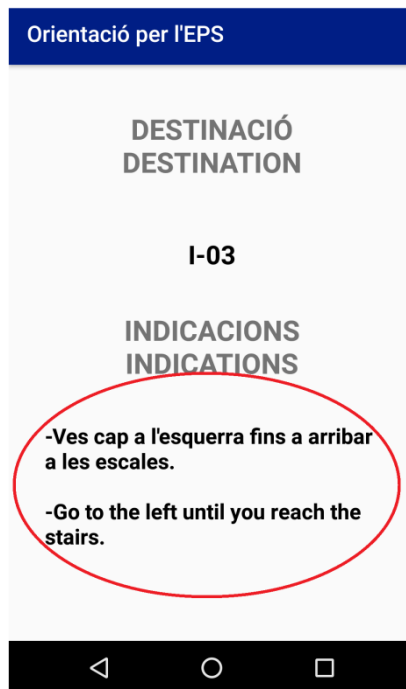


Figura 15.6: Segona indicació

Seguint la indicació s'acaba arribant a l'últim punt i com que la destinació que s'ha escollit està a la vora d'aquest, la indicació que apareix és diferent a la resta, ja que indica a l'usuari que s'ha arribat al lloc que havia escollit.

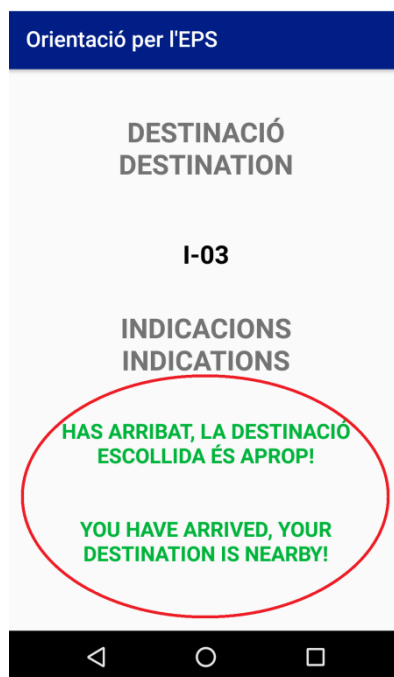


Figura 15.7: Última indicació