



**MÀSTER UNIVERSITARI EN SOFTWARE LLIURE**  
**DESENVOLUPAMENT D'APLICACIONS**

**SISTEMA D'INVENTARI PER LA GESTIÓ I**  
**CONTROL D'OBJECTES DELS DIFERENTS ESPAIS**  
**D'UN MUSEU**

**Autor:** Jordi Guinart Vidal

**Consultor UOC:** Gregorio Robles Martínez

**Consultor extern:** Miquel Àngel Cordon Pajuelo



Contingut protegit per una llicència Creative Commons BY-SA 4.0 que permet la còpia i difusió ilimitades, i l'ús en obres pròpies amb la mateixa llicència CC. El contingut íntegre de la llicència es pot consultar a <http://creativecommons.org/licenses/by-sa/4.0>

## Taula de continguts

1.- INTRODUCCIÓ.....	1
2.- DESENVOLUPAMENT.....	3
2.1.- Anàlisi de viabilitat.....	3
2.1.1.- Viabilitat tècnica.....	3
2.1.2.- Viabilitat econòmica.....	4
2.1.3.- Viabilitat operativa.....	4
2.2.- Anàlisi de requeriments.....	4
2.2.1.- Objectius i abast.....	5
2.2.2.- Definicions,acrònims i abreviatures.....	5
2.2.3.- Descripció general.....	6
2.2.3.1.- Perspectives del producte.....	6
2.2.3.2.- Funcionalitats del producte.....	6
2.2.3.3.- Característiques d'usuari.....	7
2.2.3.4.- Restriccions.....	7
2.2.3.5.- Requisits per a noves versions.....	8
2.2.4.- Requeriments de l'interfície.....	8
2.2.5.- Requeriments funcionals.....	9
2.3.- Disseny.....	11
2.3.1.- Diagrames de cas d'ús.....	11
2.3.1.1.- Cas d'ús inventari.....	11
2.3.1.2.- Cas d'ús de gestió d'objectes i zones.....	12
2.3.1.3.- Cas d'ús sincronització i paràmetres generals.....	13
2.3.2.- Interfícies d'usuari.....	13
2.3.3.- Mapa de navegació.....	16
2.3.6.- Arquitectura de l'aplicació.....	17
2.3.6.1.- Aplicació.....	17
2.3.6.2.- Bases de dades.....	17
2.3.6.2.1.- Base de dades Offline (SQLITE).....	17
2.3.6.2.1.- Base de dades servidor.....	17
2.3.6.2.2.- Disseny de la Base de dades.....	18
2.3.6.2.2.- Estructura de les taules de la Base de dades.....	19
2.4.- Construcció.....	19
2.4.1.- Classes principals del programa.....	20
2.4.2.- Diagrama de classes.....	22
2.5.- Proves.....	23
2.5.1.- Proves contra requeriment.....	24
2.5.1.2.- Requisits de rendiment.....	24
2.5.1.3.- Altres requisits.....	25
2.5.2.- Proves de rendiment.....	25
2.6.- Implementació.....	26
2.6.1.- Desplegament e instal·lació.....	26

3.- RESULTATS.....	27
3.1.- Accés a l'aplicació.....	27
3.2.- Accés al mòdul d'inventari.....	28
3.3.- Accés a mòdul buscar RFID.....	31
3.4.- Accés al mòdul Zones.....	32
3.5.- Accés a mòdul Objectes.....	32
3.6.- Accés a la sincronització de la Bdd i preferències del sistema.....	34
3.6.1.- Sincronització de la Bdd.....	35
3.6.2.- Preferències generals.....	36
4.- CONCLUSIONS I EXPERIÈNCIA.....	37
4.1.- Conclusions.....	37
4.2.- Experiència.....	38
Bibliografia.....	40

# 1.- INTRODUCCIÓ

---

La gestió d'inventaris en llocs on el volum d'objectes és ampli, s'ha tornat una feina molt complicada de portar a terme. Aprofitant l'incursió de la tecnologia RFID i les necessitats actuals, MEIBIT (una pyme gironina especialitzada en la gestió i control d'accés) ha decidit desenvolupar un software per dur a terme aquesta fita.

La meua motivació per participar en aquest projecte de MEIBIT era perquè tenia molt clar que en primer lloc volia desenvolupar una aplicació per el sistema operatiu Android i en segon terme volia fer les pràctiques en una empresa pròxima per poder tenir un feedback constant. Vaig buscar aquests dos trets diferencials en empreses que estiguessin a dins el parc científic de Girona on hi ha una gran concentració de tecnològiques.

Tot i que el desenvolupament de software per Android s'aparta una mica del dia a dia de l'empresa, era un repte incorporar-me a l'equip de 2 persones que l'estava desenvolupant. La seva concepció va ser fruit de les necessitats d'un client a tenir una aplicació per poder controlar l'inventari d'un museu. Mitjançant la tecnologia RFID i un telèfon mòbil, volia poder portar un catàleg de les diferents sales i objectes que disposa, d'una manera còmode i segura.

Actualment l'empresa presentarà el programa al client i decidiran si a part d'utilitzar-la per benefici propi, la comercialitzen. El client la testejaria dia a dia i en cas d'una bona funcionalitat, aportaria financiació necessària a MEIBIT per acabar de polir-la i ser partícip de la comercialització d'aquesta.

El llenguatge amb el que s'ha desenvolupat l'aplicació del sistema és JAVA juntament amb SQLite en la part de la base de dades. S'ha fet compatible per les versions d'Android Ice Cream Sandwich en endavant i es preveu en un curt termini canviar-li el disseny i passar-la a Fragments per aprofitar-ne les millores gràfiques que presenta.



L'objectiu principal del projecte és realitzar l'anàlisi, disseny i construcció d'una aplicació que com ja s'ha esmentat anteriorment, permeti portar l'inventari dels objectes dins d'uns determinats espais utilitzant un lector RFID com a .

Per a complir amb aquests objectius , en primer lloc s'han de definir exhaustivament els requeriments del sistema que el nostre client ens ha proposat i a posteriori escollir quin entorn de treball, així com les eines necessitarem per a dur a terme aquest projecte.

L'informe està dividit en tres grans estructures. En la primera veurem els aspectes general: analitzarem la viabilitat de portar endavant el projecte, requeriments, perspectives etc...

La segona part veurem els aspectes relacionats amb el desenvolupament de l'aplicació agafant com a base l'anàlisi de viabilitat, els requeriments, disseny, construcció, proves i finalitzant amb la implementació del sistema.

Pel que fa al tercer capítol s'analitzarà el resultat final de l'aplicació amb diverses captures de pantalla posant més èmfasis en els processos en els que he participat activament així com les conclusions finals i les coses que he après a partir de l'experiència.

## **2.- DESENVOLUPAMENT**

---

En aquest apartat veurem el desenvolupament del projecte des de l'anàlisi fins a la implementació. En primer lloc farem un estudi de viabilitat, detallant els tres apartats principals: tècnica, econòmica i operacional.

En segon lloc ens centrarem amb el disseny i el desenvolupament de l'aplicació. Finalment es mostrarà la bateria de proves realitzades i la posta en marxa de l'aplicació executant-se en un dispositiu mòbil.

### **2.1.- Anàlisi de viabilitat**

#### **2.1.1.- Viabilitat tècnica**

El projecte va ser desenvolupat utilitzant eines de software lliure en la seva totalitat. Començant pel sistema operatiu on es desenvolupava fins a les aplicacions, llenguatge i base de dades utilitzada.

El hardware utilitzat va ser:

- Servidor CVS per a la gestió del control de canvis de les diferents classes .java

Model: Acer veriton S661  
Processador: Core2 Duo  
HD: 250 Gb de disc  
RAM : 2Gb de memòria

- Ordinadors per al desenvolupament

Model: HP Elitedesk 800 G1  
Processador: Core i5 vPro  
Hd: 250 SSD  
RAM: 8GB de memòria  
SO: Kubuntu 64 Bits

Les eines utilitzades van ser:

Desenvolupament: Android Studio/ JAVA  
Bdd: SQLite / SqliteBrowser

Tot i ser un equipament modest és suficient per al desenvolupament. El servidor de versions requereix molt poca potència i l'Android Studio corre a una velocitat acceptable en els ordinadors de sobretaula.

### 2.1.2.- Viabilitat econòmica

Un 85 % dels costos totals, es destinen als salaris que s'han de pagar als desenvolupadors. En aquesta empresa s'ha utilitzat un cap de projectes i 3 persones en pràctiques a temps parcial. A part hem de contar la infraestructura de hardware i els costos associats com ara consums etc .. Com que són recursos compartits per altres funcionalitats a dins de l'empresa, s'engolbaran en un mateix grup: despeses varies.

Concepte	Cost
Salari mensual Cap de Projectes (incloent IRPF i SS)	2.500 €
Telèfons mòbils (test) x 2	450 €
Despeses varies proporcionals (ordinadors, llum, local, viatges)	100 €
<b>TOTAL mensual</b>	<b>3.050 €</b>

\* S'ha de tenir en compte que les 3 persones de suport només treballaran a temps parcial i no cobraran per la feina realitzada ja que estan en període de pràctiques.

Es calcula que el amb 2 mesos aquesta aplicació podria estar enllestida, per tant estariem parlant d'un cost total aproximat de **6.000 €**.

### 2.1.3.- Viabilitat operativa

El programa té una interfície molt amigable i l'ús de smartphones està molt arrelat a la societat actual, facilitant molt la feina i l'adaptació dels treballadors en un entorn on ja estan familiaritzats.

Les reunions que s'ha tingut amb l'empresa i presentació de diferents betas han estat molt ben rebudes. Cal remarcar que avui en dia no tenen aquest sistema informatitzat i ho realitzen manualment, així doncs aquest canvi els afavorirà sense cap mena de dubte.

Tant a nivell d'infraestructura i equipament, com la liquiditat per portar a terme el projecte són totalment factibles per a l'empresa i per tant el projecte no tindrà cap problema per esdevenir viable.

## 2.2.- Anàlisi de requeriments

En aquest apartat detallarem l'interfície i els requeriments que ens ha demanat el nostre client i que són necessaris que compleixi la nostra aplicació d'inventari.



### 2.2.1.- Objectius i abast

L'aplicació està concebuda per a poder portar l'inventari d'objectes d'un museu distribuïts per un número determinat d'espais mitjançant la tecnologia RFID. El programa el realitzarà l'empresa MEIBIT.

Concretament l'usuari haurà de poder inventariar els objectes que hi ha dins una sala a partir d'un lector RFID que anirà llegint els codis de cada un d'aquests elements. Un cop hagin estat enregistrats dins la Bdd de l'aplicació l'usuari els podrà consultar a posteriori per saber-ne la seva localització o bé fer alguna modificació si s'escau.

Una altre de les funcionalitats serà fer un recompte d'objectes dins una espai en concret. Això vol dir que un cop tornem a entrar a la sala, l'usuari haurà de poder comprovar que tots els objectes que inicialment es van inventariar, segueixen en el seu lloc. S'iniciarà la lectura i a mesura que el lector vagi llegint, anirà marcant els objectes com a trobats fins a la seva finalització.

A més a més servirà com a catàleg per a poder consultar la descripció de l'objecte, veure la fotografia i les característiques generals.

El programa està pensat perquè treballi en mode "sense connexió" ja que hi ha moltes parts del museu on no arriba la cobertura. Posteriorment aquestes dades es sincronitzaran amb una bdd central per a la posterior consulta des d'altres tipus de sistemes.

L'aplicació ha de funcionar amb 3 tipus de lectors RFID que s'enllaçaran amb el mòbil a partir de la tecnologia sense fils Bluetooth. Aquests dispositius són capaços d'obtenir un codi que hi ha dins una targeta RFID sempre que estiguin dins del seu radi d'acció.

### 2.2.2.- Definicions,acrònims i abreviatures

- **RFID** (Sigles de Radio Frequency Identification):

És un sistema d'emmagatzematge i recuperació de dades remotes que utilitza dispositius anomenats : etiquetes,transponedors o tags RFID. El propòsit fonamental d'aquesta tecnologia és transmetre la identitat de l'objecte , mitjançant una ID única utilitzant ones de ràdio

- **MEIBIT:**

És l'empresa on he realitzat les pràctiques. Va néixer a partir de la dissolució d'una altre empresa matriu. Un dels treballadors (l'actual propietari),va tenir la oportunitat d'agafar una part del negoci i es dedica a oferir solucions tecnològiques per a l'automatització i control de sistemes en àmbits de

la identificació intel·ligent de persones i objectes.

- **BDD:** Base de dades
- **ASSETVENTORY:** Nom que se li ha donat a l'aplicació.

## 2.2.3.- Descripció general

### 2.2.3.1.- Perspectives del producte

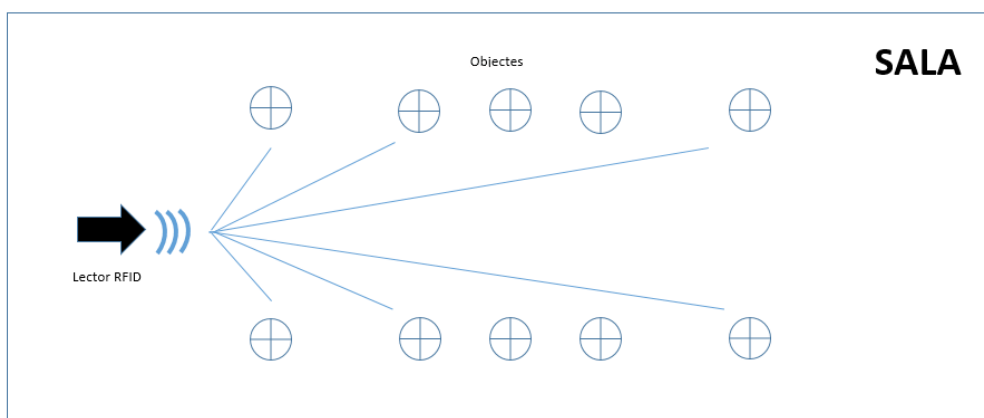
L'aplicació està enfocada a funcionar amb el sistema operatiu per a dispositius mòbils Android, per la versió Ice Cream Sandwich (versió 4) en endavant i amb qualsevol tipus de dispositiu mòbil que tingui tecnologia Bluetooth.

Es preveu incorporar-lo a la botiga d'aplicacions Google Play tot i que és un producte concebut per ús empresarial i que requereix de tot un entorn complexe per a fer-lo funcionar. Tot i això per tema d'actualitzacions e incorporació de nous requeriments, podria ser una funcionalitat interessant per l'empresa incorporar l'aplicació a la tenda de Google.

### 2.2.3.2.- Funcionalitats del producte

El programa es desenvoluparà per a la plataforma Android i l'usuari hi accedirà mitjançant una validació prèvia. Constarà de les següents parts:

- **Inventari:** Aquesta és l'opció principal del programa i serà on crearem un nou inventari o bé n'escollirem un que hàgim realitzat prèviament. Amb un lector RFID comprovarem haurem de poder inventariar tots els objectes d'una sala o comprovar que es troben en el lloc on haurien de ser.



*Il·lustració 1: Representació gràfica d'una sala del museu en procés d'inventari*

- **Buscar objecte:** Aquesta serà una funcionalitat molt simple que a partir d'una lectura d'un objecte ens mostrarà si el tenim inventariat i en cas afirmatiu ens mostrarà les característiques: descripció,localització,tipus objecte etc ...
- **Preferències:** En aquesta pantalla podrem configurar el password d'accés de l'usuari, el servidor on s'haurà de sincronitzar la bdd . Paràmetres generals.  
També hi haurà l'opció de sincronització; per quan hàgim fet una ruta per les diferents sales i vulguem bolcar la informació obtinguda o bé vulguem incorporar un nou dispositiu mòbil i necessiti obtenir les dades per primera vegada.
- **Gestió d'Objectes:** Aquí podrem consultar la informació de tots els objectes que tinguem a la bdd. Serà una llista on hi haurà l'opció de filtrar per zones,nom,tipus d'objecte etc ...  
S'han de poder fer les opcions bàsiques d'alta,baixa,modificació
- **Gestió de Zones:** Aquí podrem consultar la informació de totes les zones que tinguem a la bdd. S'han de poder fer les opcions bàsiques d'alta,baixa,modificació

### 2.2.3.3.- Característiques d'usuari

No existeixen perfils d'usuari . Només hi haurà un usuari únic que tindrà totes les opcions de l'aplicació disponibles. Al ser una aplicació molt concreta no es requereix diferents nivells de funcionalitats en funció de la persona que l'utilitzi.

### 2.2.3.4.- Restriccions

La única restricció que existeix és la versió d'Android que ha de ser la 4 en endavant. També limita quin tipus de pantalla i resolució té el dispositiu. Encara que l'aplicació vagi bé no serà massa operativa si la pantalla és molt petita. L'aplicació ha estat optimitzada per pantalles a partir de 4 ". L'aplicació funcionarà igual en pantalles inferiors però la seva usabilitat es veurà molt perjudicada.

També queda restringit els tipus de lectors que es poden utilitzar. Ara per ara s'han implementat 3 tipus de lectors amb característiques generals, rang de lectura ,composició i preus diferencials per tal que el client tingui opció d'escollir segons les seves necessitats.

Els 3 lectors implementats són:



*Il·lustració 2: Lector  
CypherLab*



*Il·lustració 3: Lector  
Embiventory*



*Il·lustració 4: Lector  
TLS*

Es preveu noves implementacions però amb aquests 3 tipus es cobreixen un ampli rang de possibilitats i preus que es poden ajustar a tot tipus de client.

### **2.2.3.5.- Requisites per a noves versions**

Es contempla passar tota la interfície de treball a *fragments* ja que aquesta nova característica que han afegit optimitza la gestió de la memòria i de recursos en detriment de l'Activity.

Es preveu introduir més models de lector RFID i fer que l'aplicació sigui compatible amb la majoria de fabricants.

Per avarcar nous mercats, s'hauria de traduir a 2 idiomes més: Francés i Espanyol.

### **2.2.4.- Requeriments de l'interfície**

- **Interfície d'usuari**

Les interfícies en les que haurà de treballar l'usuari són molt netes e intuïtives i quasi bé no existeix cap mena de corba d'aprenentatge. Qualsevol persona que hagi utilitzat un telèfon mòbil, sabrà utilitzar-la.

- **Interfície de hardware**

Diferents dispositius que utilitzin arquitectures ARM, X86, MIPS e IBM POWER i que tingui una versió d'Android a partir de la 4 podrà utilitzar el programari. També com a requeriment haurà de tenir connexió Bluetooth per a la comunicació amb el lector RFID. La pantalla es recomana per a una correcta visualització un mínim de 4"

- **Interfície de software**

El programa ha de permetre la comunicació amb la BDD del servidor per a la sincronització. Les crides es realitzaran via webservice.

- **Interfície de comunicacions**

El sistema de comunicació serà el protocol HTTP per parlar amb la BDD i el protocol Bluetooth per parlar amb els lectors RFID.

### 2.2.5.- Requeriments funcionals

<b>Descripció</b>	1.1.- Afegir,modificar i eliminar Zones
<b>Precondició</b>	Cap
<b>Entrada</b>	Dades de la zona (nom,descripció,imatge,tag..)
<b>Procés</b>	Dades gravades/modificades a la Bdd
<b>Sortida</b>	Missatge d'acceptació / error

<b>Descripció</b>	1.2.- Afegir,modificar i eliminar Objectes
<b>Precondició</b>	Cap
<b>Entrada</b>	Dades de l'objecte (nom,descripció,zona,estat,imatge,tag..)
<b>Procés</b>	Dades gravades/modificades a la Bdd
<b>Sortida</b>	Missatge d'acceptació / error

<b>Descripció</b>	1.3.- Afegir,modificar i eliminar Inventari
<b>Precondició</b>	Cap
<b>Entrada</b>	Dades de l'inventari (nom,descripció,datainici..)
<b>Procés</b>	Dades gravades/modificades a la Bdd
<b>Sortida</b>	Missatge d'acceptació / error

<b>Descripció</b>	1.4.- Buscar si un objecte està inventariat
<b>Precondició</b>	Lector i tag RFID
<b>Entrada</b>	IdObjecte
<b>Procés</b>	Cercar a la BDD si element present
<b>Sortida</b>	Missatge d'acceptació / error

<b>Descripció</b>	1.5.- Comprovació objectes inventariats en les diferents zones
<b>Precondició</b>	Inventari,Zona i lector RFID
<b>Entrada</b>	IdInventari-IdZona, IdObjecte(s)
<b>Procés</b>	Mirem si els objectes trobats coincideixen amb els inventaris
<b>Sortida</b>	Llista amb objectes trobats i/o no trobats

<b>Descripció</b>	1.6.- Sincronització Bdd offline amb Bdd online
<b>Precondició</b>	Tenir connexió amb servidor BDD online
<b>Entrada</b>	Cap (IpServidor es fa a l'apartat de paràmetres de configuració)
<b>Procés</b>	Es passen les dades de la Bdd Offline cap Online i viceversa
<b>Sortida</b>	Bases de dades actualitzades

<b>Descripció</b>	1.7.- Paràmetres de configuració
<b>Precondició</b>	Cap
<b>Entrada</b>	Paràmetres (ipservidor,password etc ..)
<b>Procés</b>	Paràmetres generals modificats a la Bdd
<b>Sortida</b>	Missatge d'acceptació/error

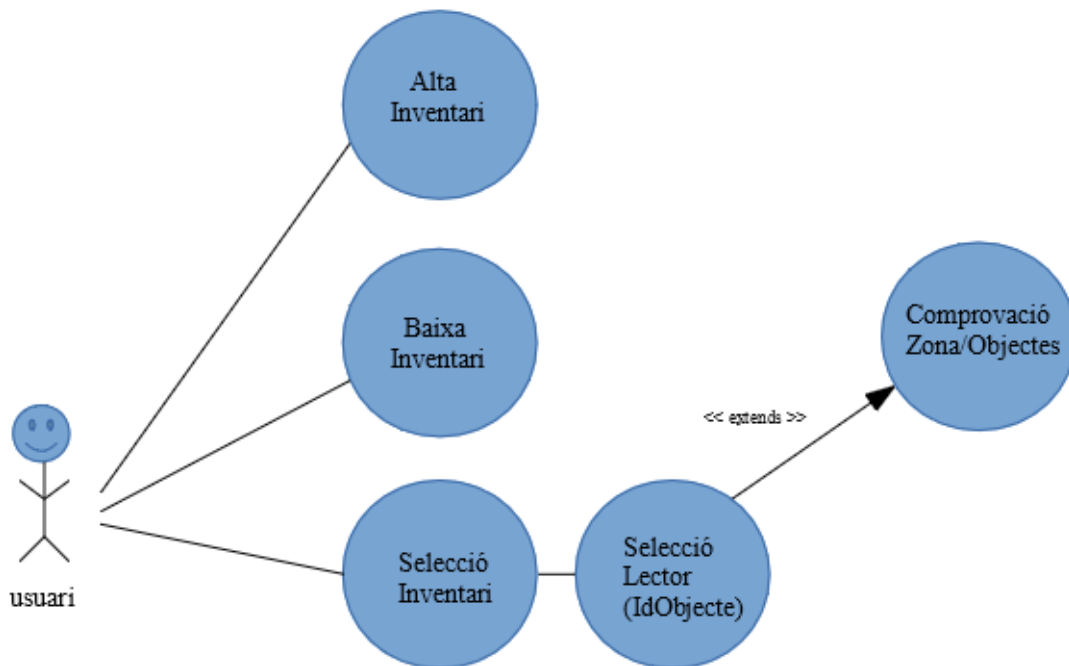
## 2.3.- Disseny

A continuació explicarem tot el que engloba el disseny del sistema. En primer lloc es mostraran els diferents casos d'ús , a continuació un diagrama de blocs generals per contextualitzar cada un dels mòduls , l'estructura i el disseny conceptual de la bdd i per últim una mosta de l'arquitectura que es farà servir per implementar el sistema.

### 2.3.1.- Diagrames de cas d'ús

A continuació passarem a analitzar els diferents casos d'ús referents a les funcionalitats que l'usuari podrà realitzar. S'analitzaran els passos o activitats que s'han de realitzar per a dur a terme una acció. En aquest cas només tenim un autor ja que no es requereixen perfils diferents amb funcionalitats especials.

#### 2.3.1.1.- Cas d'ús inventari



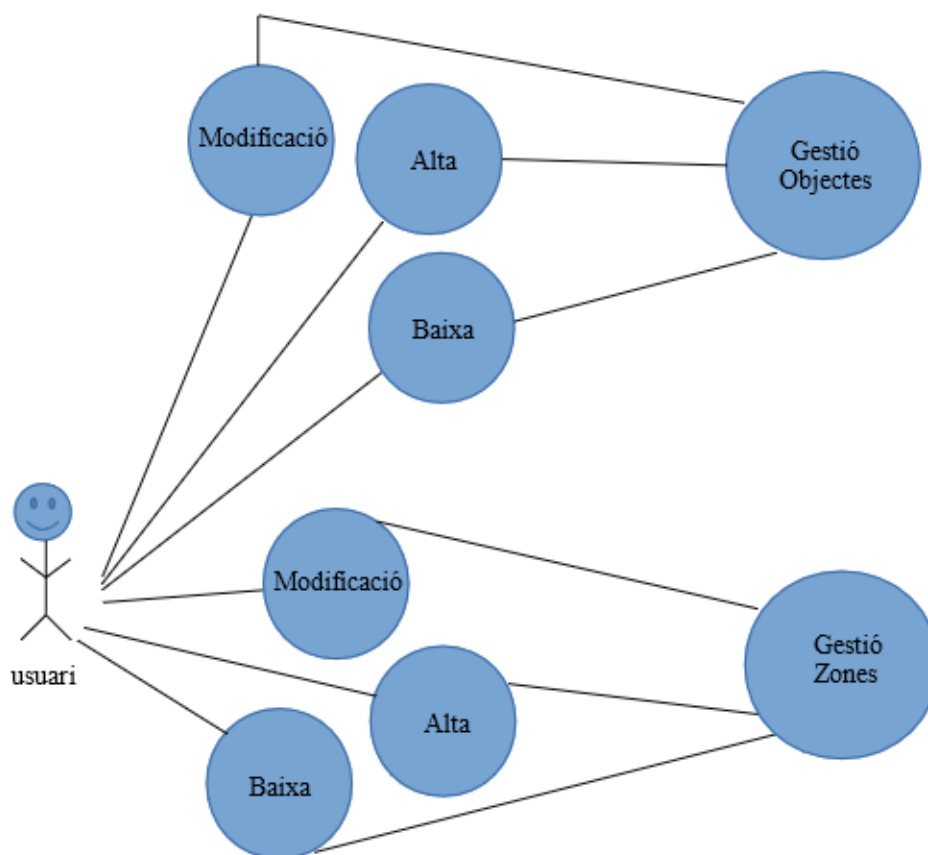
*Il·lustració 5: Cas d'ús1, inventari*

Com a funcionalitat principal, en l'inventari podrem donar d'alta o baixa inventaris o bé seleccionar un de nou i fer la comprovació dels objectes que estaven inventariats.

### 2.3.1.2.- Cas d'ús de gestió d'objectes i zones

En aquest cas d'ús s'analitzen les funcions bàsiques de gestió general tant per la gestió d'objectes i de Zones.

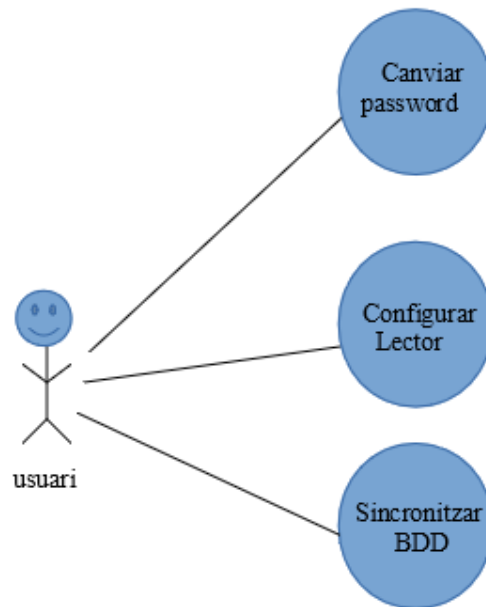
S'ha agrupat en un sol gràfic ja que les accions a realitzar són iguals en ambdós casos i d'aquesta manera queda més clar i ben explicat. Les funcions de donar d'alta, baixa i modificació conformen la gestió d'objectes i zones.



*Il·lustració 6: Cas d'ús 2, Gestió Objectes/Zones*



### 2.3.1.3.- Cas d'ús sincronització i paràmetres generals.

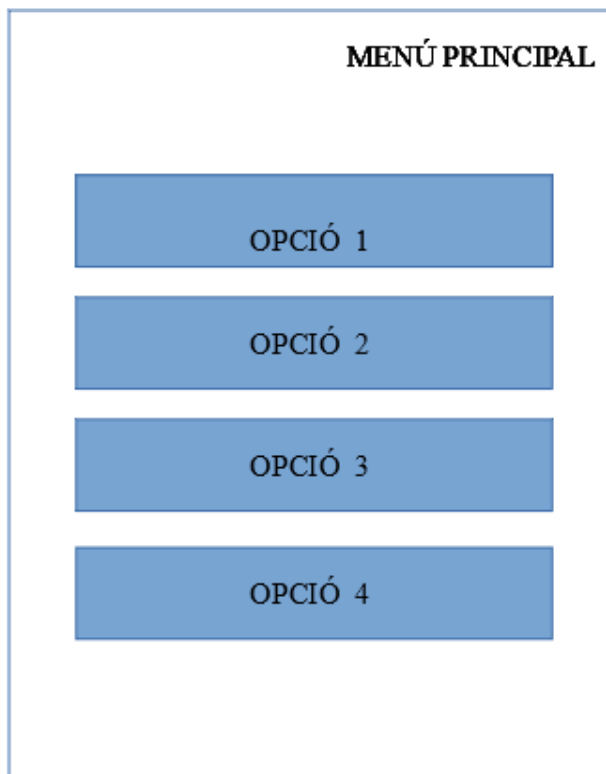


*Il·lustració 7: Cas d'ús 3, Sincronització i paràmetres generals*

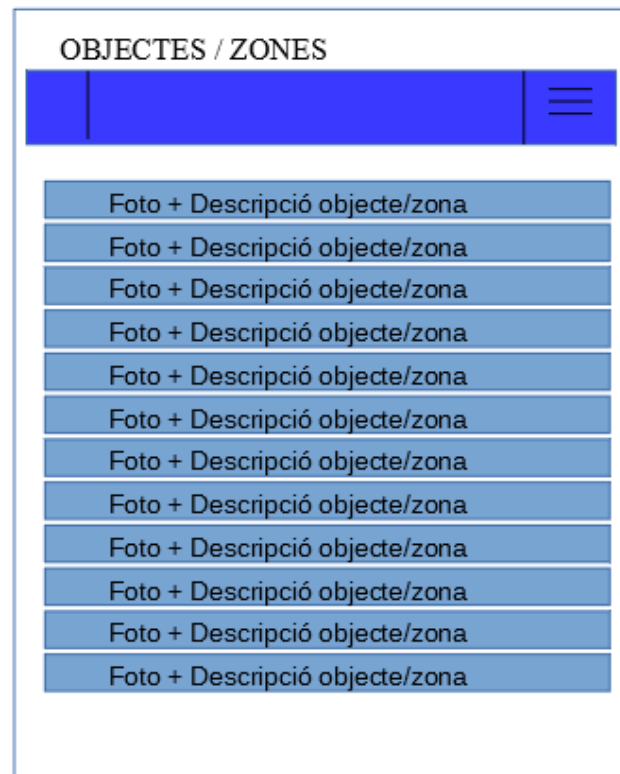
### 2.3.2.- Interfícies d'usuari

Per la fluïdesa a l'hora de treballar-hi s'han buscat dissenys minimalistes i amb pocs elements en pantalla. El fet que sigui en un dispositiu mòbil ajuda a no sobrecarregar les pantalles ja que tampoc hi cap massa informació, per tant aquesta ha de ser clara i concisa.

Degut a un gran volum d'informació, s'ha optat per aplicar filtres a les llistes, poder ordenar els elements i una eina de cerca per facilitar encara més a l'hora de trobar la informació que el client necessita.



*Il·lustració 8: Interfície del menú principal*



*Il·lustració 9: Interfícies pantalla zona/objectes*

S'han afegit dreceres a la part d'inventari per així facilitar la modificació de qualsevol zona o objecte. L'usuari no haurà d'anar a la gestió d'aquests per modificar-los i així ens assegurem d'una bona usabilitat.

Durant tota la interfície s'utilitzen controls ja més que coneguts com botons, quadres per entrar text i llistes per a mostrar elements. Qualsevol persona que disposi d'un smartphone sabrà treballar-hi d'una manera amigable. A part aquesta aplicació no es caracteritza per fer un número molt gran de funcionalitats i per tant la corba d'aprenentatge pràcticament inexistent.

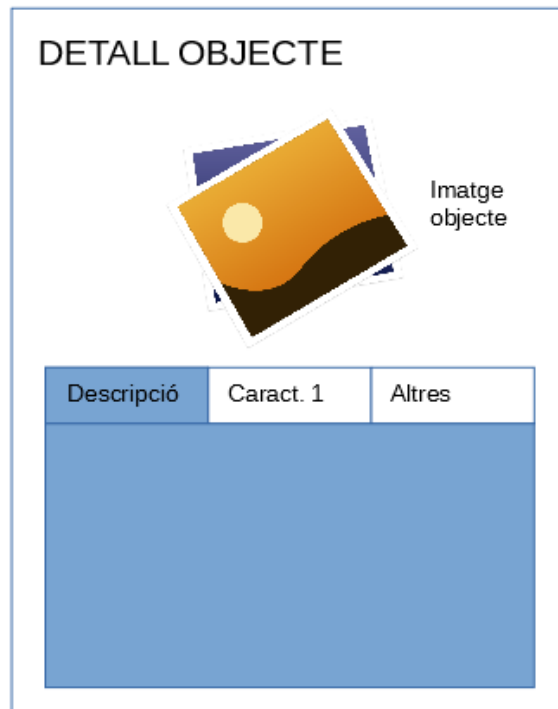


*Il·lustració 10: Interfície recompte objectes inventari*



*Il·lustració 11: Interfície buscar objecte*

La pantalla de detall serà on es mostrarà tota la informació que tenim de l'objecte. Ens permetrà també modificar-ne els valors o corregir-ne la fotografia en el cas que no es correspongui amb l'objecte.

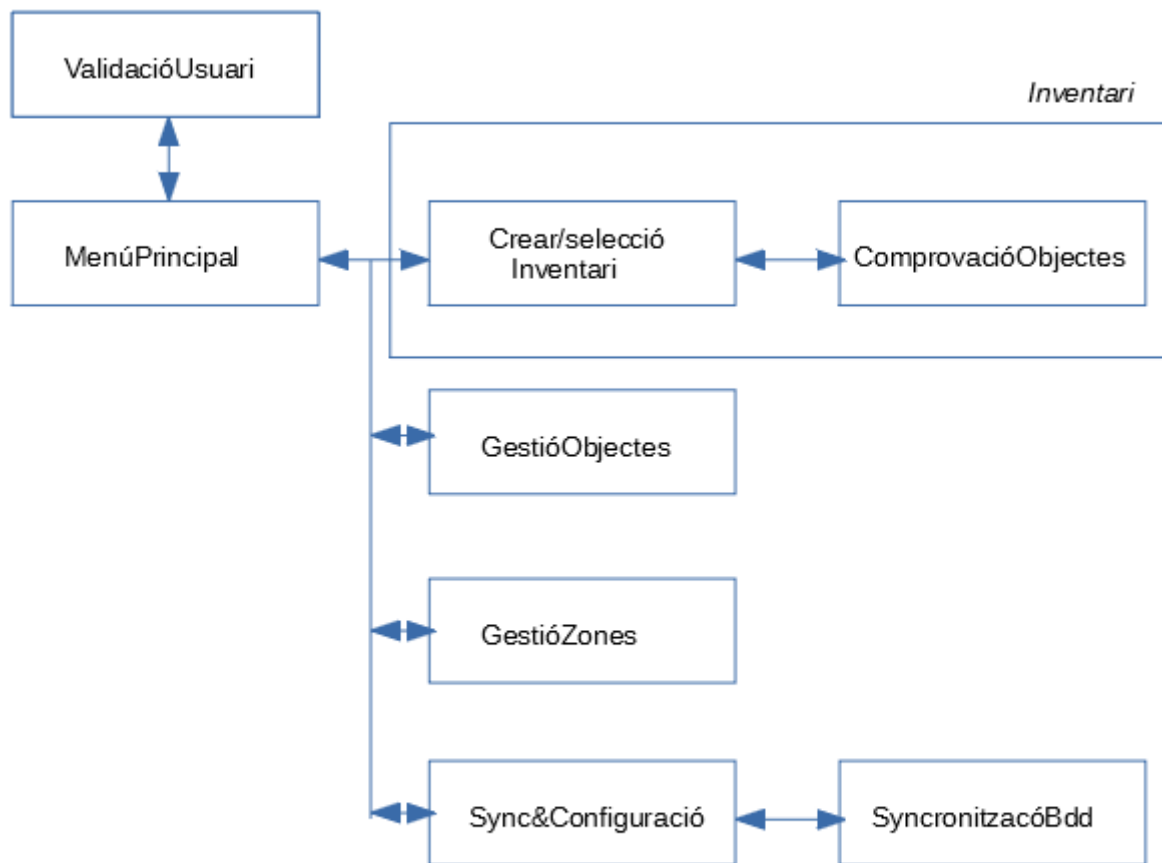


*Il·lustració 12: Interfície propietats objecte*

Per a fer més compacte la visualització de la informació, s'han utilitzat pestanyes on cada una mostra un tipus d'informació diferent. Fent lliscar cap a la dreta o esquerre podrem accedir al següent grup d'informació sense perdre la imatge de vista.

### 2.3.3.- Mapa de navegació

En aquest apartat mostrarem quines opcions estan disponibles i quin recorregut ha de fer l'usuari per accedir-hi.



*Il·lustració 13: Mapa de navegació de l'aplicació*

## **2.3.6.- Arquitectura de l'aplicació**

L'aplicació ha estat desenvolupada per a la plataforma Android i necessitem tenir instal·lat un servidor on hi haurà la BDD on es sincronitzaran els diferents dispositius i i una base de dades local per a treballar-hi de manera desatesa (SQLITE).

### **2.3.6.1.- Aplicació**

#### **Interfície d'usuari**

S'ha utilitzat el SDK d'Android per a la composició de les diferents pantalles. S'utilitza XML com a estructura de composició i s'han fet servir controls molt bàsics per interactuar amb l'usuari. Amb l'eina Android Studio es pot realitzar d'una manera gràfica i molt senzilla. És altament recomanable tenir coneixement d'XML per modificar propietats i afinar en el disseny.

#### **Connectivitat**

Per enllaçar els nostres dispositius bluetooth i comunicar-nos amb els lectors RFID es faran crides directament al mateix SO mitjançant threads. No s'haurà d'utilitzar cap mena ni de controlador ni res per l'estil.

### **2.3.6.2.- Bases de dades**

Perquè el nostre aplicatiu funcioni correctament necessitem 2 motors de base de dades diferenciats que els detallarem a continuació:

#### **2.3.6.2.1.- Base de dades Offline (SQLITE)**

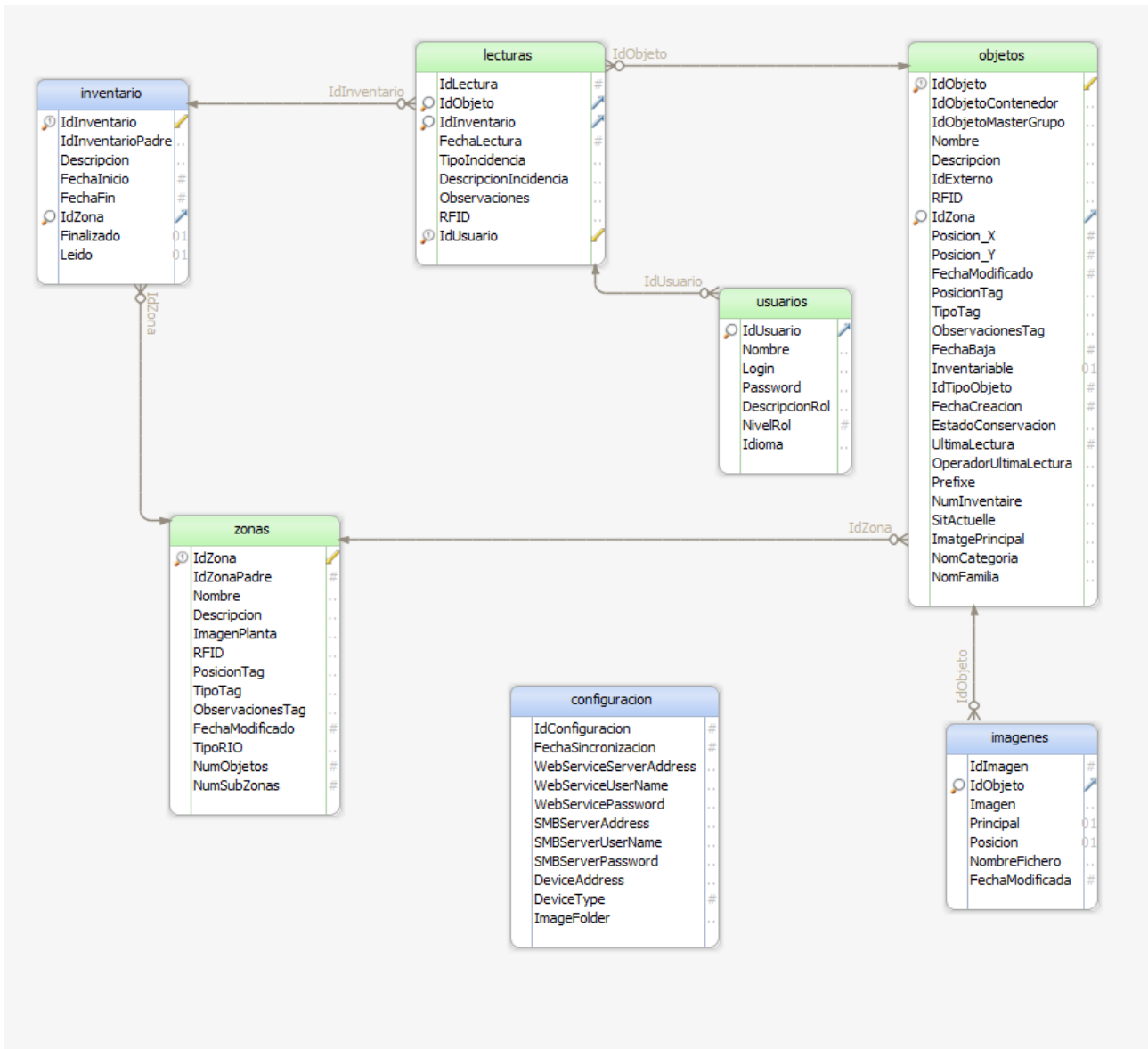
Per a la bdd interna , s'ha optat per utilitzar una bdd SQLite. Simplement consta d'un fitxer on hi ha tota la informació i no s'ha d'instal·lar cap sistema complementari. És senzilla i ràpida i per les nostres necessitats suficient. L'estructura en les taules comunes és exactament igual i a continuació es detallarà en l'apartat de la base de dades del servidor.

#### **2.3.6.2.1.- Base de dades servidor**

En aquest apartat es deixa a mans del client quin tipus de base de dades utilitzarà. L'aplicació simplement ataca un webservice que és el que li dona o rep les dades i aquest és l'encarregat de parlar i fer totes aquestes transaccions amb la Bdd. D'aquesta manera fem que independentment de l'estructura que tingui o esculli el client, l'aplicació podrà funcionar.

### 2.3.6.2.2.- Disseny de la Base de dades

L'estructura i disseny de la base de dades representat d'una manera gràfica seria el següent.



Il·lustració 14: Diagrama conceptual de la base de dades

### 2.3.6.2.2.- Estructura de les taules de la Base de dades

configuracion	objetos	usuarios
IdConfiguracion INT	IdObjeto TEXT	IdUsuario INT
FechaSincronizacion INT	IdObjetoContenedor TEXT	Nombre TEXT
WebServiceServerAddress TEXT	IdObjetoMasterGrupo TEXT	Login TEXT
WebServiceUserName TEXT	Nombre TEXT	Password TEXT
WebServicePassword TEXT	Descripcion TEXT	DescripcionRol TEXT
SMBServerAddress TEXT	IdExterno TEXT	NivelRol INT
SMBServerUserName TEXT	RFID TEXT	Idioma TEXT
SMBServerPassword TEXT	IdZona INT	zonas
DeviceAddress TEXT	IdFamilia INT	IdZona INT
DeviceType INT	IdCategoria INT	IdZonaPadre INT
ImageFolder TEXT	Posicion_X INT	Nombre TEXT
imagenes	Posicion_Y INT	Descripcion TEXT
IdImagen INT	FechaModificado INT	ImagenPlanta TEXT
IdObjeto TEXT	PosicionTag TEXT	RFID TEXT
Imagen TEXT	TipoTag TEXT	PosicionTag TEXT
Principal BIT	ObservacionesTag TEXT	TipoTag TEXT
Posicion BIT	FechaBaja INT	ObservacionesTag TEXT
NombreFichero TEXT	Inventariable BIT	FechaModificado INT
FechaModificada INT	IdTipoObjeto INT	TipoRIO TEXT
inventario	FechaCreacion INT	NumObjetos INT
IdInventario TEXT	EstadoConservacion TEXT	NumSubZonas INT
IdInventarioPadre TEXT	UltimaLectura INT	
Descripcion TEXT	OperadorUltimaLectura TEXT	
FechaInicio INT	Prefixe TEXT	
FechaFin INT	NumInventaire TEXT	
IdZona INT	SitActuelle TEXT	
IdFamilia INT	ImatgePrincipal TEXT	
IdCategoria INT	NomCategoria TEXT	
Finalizado BIT	NomFamilia TEXT	
Leido BIT		
lecturas		
IdLectura INT		
IdObjeto TEXT		
IdInventario TEXT		
FechaLectura INT		
TipolIncidencia TEXT		
DescripcionIncidencia TEXT		
Observaciones TEXT		
RFID TEXT		
IdUsuario INT		

Il·lustració 15: Detall de les taules de la base de dades

## 2.4.- Construcció

En aquest apartat analitzarem l'estructura de l'aplicació, les classes que s'han creat per aquest propòsit i els layouts que per a cada una de les pantalles.

L'aplicació s'ha programat amb llenguatge JAVA i utilitzant el SDK d'Android. La interfície de desenvolupament escollida ha estat Android Studio 2.0 i com a gestor de la Bdd Sqlite Browser. Per a la Bdd, allotjada en el servidor, s'ha utilitzat el gestor PostgreSQL.

Per a la programació de l'aplicació s'ha seguit una metodologia XP:

- Establir els requeriments i funcionalitats d'un mòdul.
- Es desenvolupaven els prototips.
- Es testeaven.
- Un cop era funcional es procedia a establir noves funcionalitats o millores.

Com que l'aplicació és molt extensa en classes, i l'espai limitat, em centraré principalment en les classes que més importància tenen i que més hi he treballat. Cal dir que l'aplicació ja estava en fase embrionària quan vaig iniciar les pràctiques i per tant s'ha seguit desenvolupant amb la mateixa línia.

Possiblement hi ha aspectes a millorar però en aquest cas un es va haver d'adaptar i ser més funcional que no pas estrictament acadèmic. S'hauria d'haver pensat una mica millor l'estructura de classes i com enfocar-ho però el dia a dia de l'empresa i la proximitat de la presentació al davant del client no va permetre reestructurar certes classes.

### 2..4.1.- Classes principals del programa

A continuació veurem una mostra de les classes més importants i quines són les tasques que implementen cada una d'elles. També mostrarem un diagrama de classes resumit per poder tenir una visió global de l'estructura.

- **m\_AssetVentoryMDB.java**

Aquesta classe extens de la classe SqlHelper i s'utilitza per a fer totes les transaccions amb la base de dades offline i online. Consultes a les diferents taules per a recuperar informació, inserts, modificacions etc .. També hi han els mètodes per a fer la sincronització amb la bdd principal via webservice. Podríem dir que és una de les classes més gran e important del programa.

- **m\_ReadRfid.java**

La classe en qüestió és l'encarregada de tota la comunicació amb els lectors RFID. Primerament cridarem a la interfície bluetooth per tal que l'usuari esculli quin dispositiu vol utilitzar tot mostrant-li una llista dels aparells bluetooth disponibles per al sistema. Després un cop establerta la comunicació, mirarem de quin tipus de lector es tracta i procedirem a cridar els mètodes de lectura mitjançant threads en les classes: *m\_LectorCiph.java*, *m\_LectorEmbi.java*, *LectorTSL.java*

- **m\_FitxeObjecteActiviy.java**

Aquesta classe és l'encarregada de gestionar tota la informació d'un objecte i mostrar-la per pantalla. Farà la consulta a la Bdd, agafarà els camps corresponents i els mostrarà en els diferents layouts de la pantalla permetent modificar dades i afegir contingut.

- **m\_FitxaZonaActivity**

Aquesta classe és l'encarregada de gestionar tota la informació d'una zona i mostrar-la per pantalla. Farà la consulta a la Bdd, agafarà els camps corresponents i els mostrarà en els diferents layouts de la pantalla permetent modificar dades i afegir contingut.

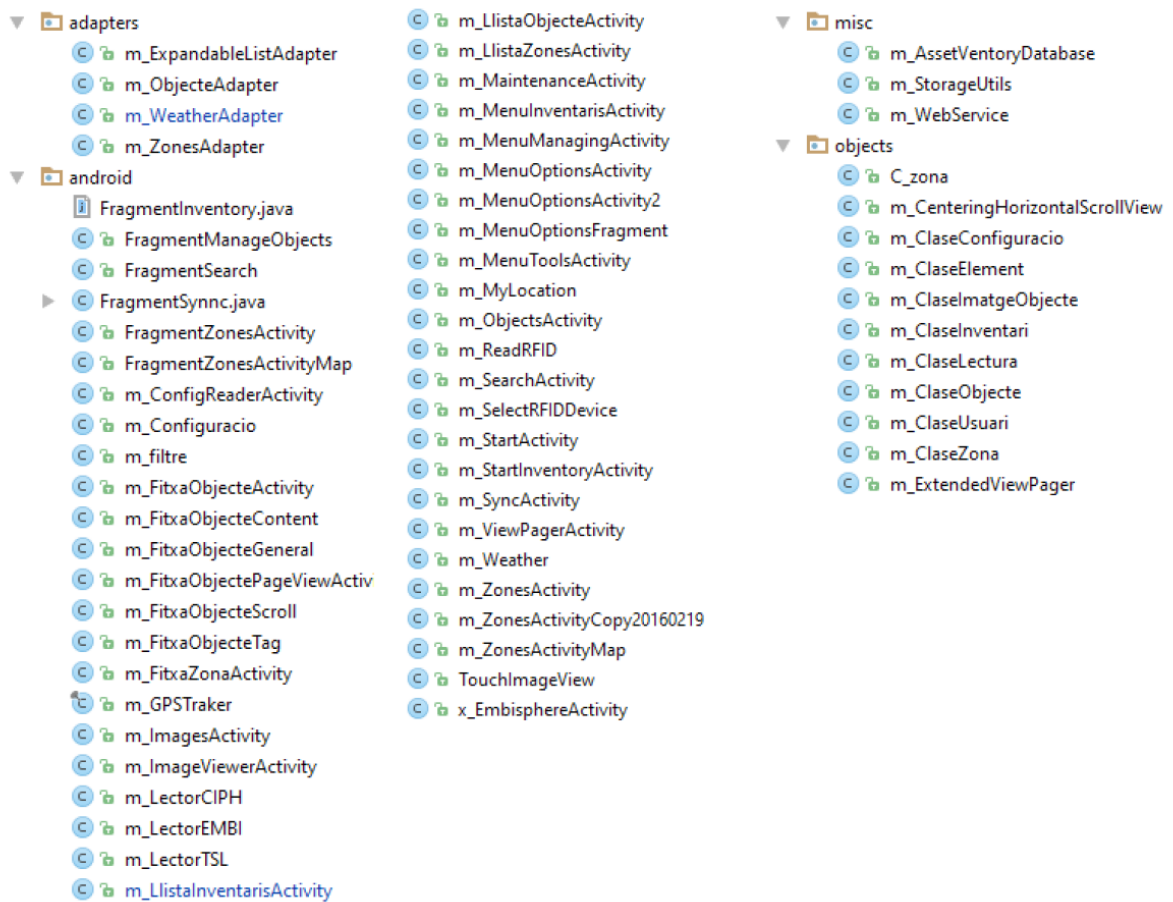


- **m\_SyncActivity**

Serà l'encarregada de sincronitzar les 2 bases de dades mitjançant la comparació de dates des de l'última sincronització. Un cop finalitza el procés, les 2 bases de dades haurien de contenir els mateixos elements.

- **m\_StartInventoryActivity**

Classe encarregada de totes les funcionalitats que tenen a veure amb l'inventari d'objectes i recompte



*Il·lustració 16: Captura de pantalla d'Android Studio on es veuen totes les classes*

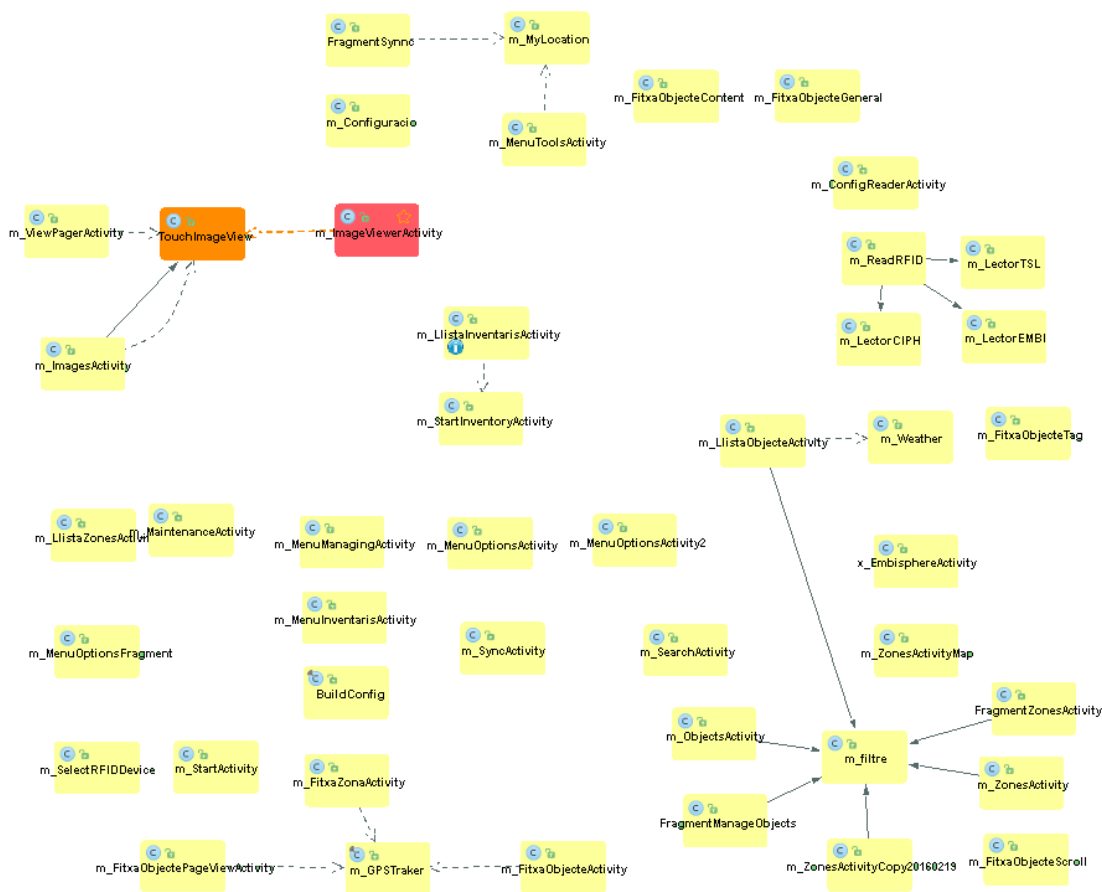
Per a poder treballar amb major comoditat, s'ha decidit fer 4 grans grups:

- **Adapters:** En aquesta secció hi hem posat adaptadors referents a les llistes que mostraran els elements de cada un dels objectes.

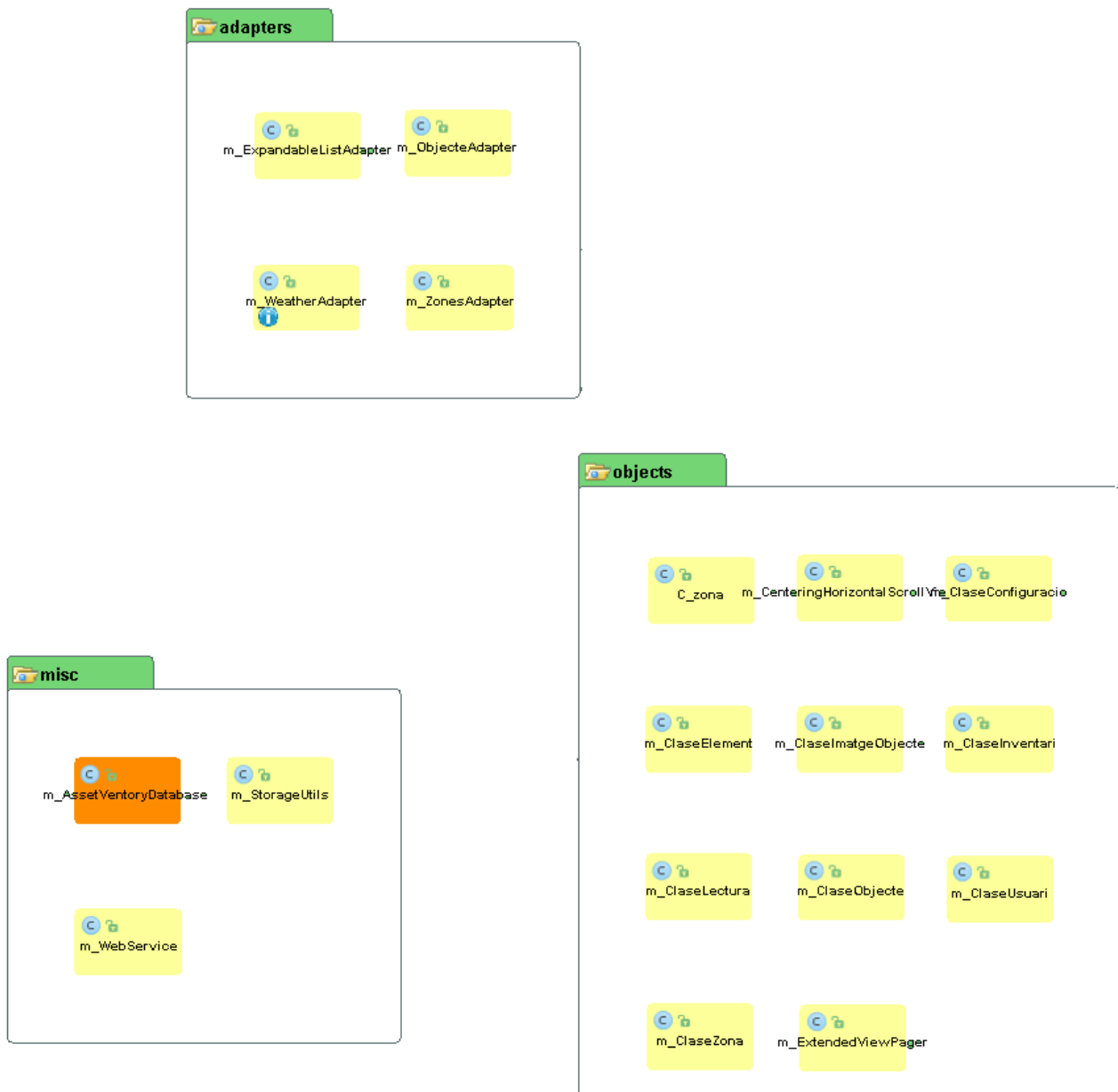
- **Android:** Totes les classes que criden als Activity , crida als lectors RFID i pantalles on l'usuari hi interactuarà.
- **Misc:** Secció que conté les classes que interactuen amb les respectives bdd i paràmetres del sistema com les diferents rutes de la carpeta SD, localització física de la Bdd etc ..
- **Objects:** Les classes dels diferents objectes de l'aplicació: objectes, zones, usuari, inventari etc ..

## 2.4.2.- Diagrama de classes

Diagrama on es pot veure la relació que hi ha entre les diferents classes que formen el programa. Degut a la dificultat de mostrar totes les relacions que hi ha en l'espai que tenim , s'ha separat en 2 diagrames .Aquests 2 diagrames es complementen entre ells:



Il·lustració 17: Diagrama classes - part 1



*Il·lustració 18: Diagrama classes -part 2*

## 2.5.- Proves

Un cop s'ha finalitzat la construcció de l'aplicació, s'ha arribat a un estat beta suficientment polit com per presentar al client. A continuació s'han fet 2 tipus de proves: contra requeriments i de rendiment.

En les proves de requeriments, s'ha utilitzat una taula per tal de verificar que realment es compleixen. En les proves de rendiment, el client ens ha facilitat a l'empresa la totalitat de la bdd amb totes els objectes i zones del museu. Així doncs, s'ha pogut testejar el rendiment que hauria de

tenir en un entorn de funcionament real.

### 2.5.1.- Proves contra requeriment

Nº	Prova	Req	Pases	OK
1	Modificar una zona	1.1	1.- Accedim a l'opció de zona 2.- Seleccionem la zona a modificar 3.- Entrem les noves dades 4.- Guardem les dades	SI
2	Canviar foto objecte	1.2	1.- Accedim a l'opció d'objectes 2.- Seleccionem un objecte a modificar 3.- Fem foto de l'objecte 4.- Guardem les dades	SI
3	Creació de nou inventari	1.3	1.- Accedim a l'opció d'inventari 2.- Apretem botó nou inventari 3.- Triem nom, zona, família 4.- Guardem	SI
4	Buscar si objecte està inventariat	1.4	1.- Accedim a l'opció de buscar objecte 2.- Connectem lector RFID 4.- Iniciem la lectura de l'objecte 4.- Mostra per pantalla descripció objecte i característiques	SI
5	Comprovar objectes inventariats zona	1.5	1.- Accedim a l'opció d'inventari 2.- Seleccionem un inventari existent 3.- Connectem lector RFID 4.- Iniciem la lectura d'objectes 4.- Guardem	SI
6	Sincronització bdd offline/online	1.6	1.- A preferències posem ip servidor 2.- Accedim a l'opció de sincronització 3.- Sincronitzem la bdd 4.- Sincronització finalitzada	SI

### 2.5.1.2.- Requisits de rendiment

El servidor on es troba la bdd principal i contra la que els diferents dispositius mòbils s'actualitzaran, només haurà d'estar disponible 24x5 (horari laboral). Els aparells poden treballar amb el mode offline durant un temps indeterminat i la funcionalitat d'inventari no és diària per tant no cal destinar molts recursos per tenir el servidor sempre funcionant.

### 2.5.1.3.- Altres requisits

Perquè l'aplicació funcioni, l'usuari haurà d'acceptar els permisos que tenim com a requeriment al AndroidManifest.xml i que es demanaran quan s'executi l'aplicació per primera vegada:

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

### 2.5.2.- Proves de rendiment

Per fer proves, s'ha utilitzat una Bdd de 500 Mb facilitada pel client amb totes les dades de que disposa el museu. Amb aquest volum s'ha vist que l'aplicació va molt justeta i les transicions entre pantalles es veuen ressentides en alguns dispositius.

Taula	Nº de registres
t_Inventari	50
t_Zones	300
t_lectures	20
t_Objectes	25.000

En el cas que s'utilitzi un telèfon mòbil amb targeta de memòria externa, degut al gran volum de dades e imatges, es recomana que aquesta tingui una velocitat de lectura ràpida.

## 2.6.- Implementació

Per la implementació del sistema, caldrà instal·lar un servidor amb una bdd que el client vulgui. Haurem d'instal·lar un webservice per tal de poder-nos comunicar amb la mateixa i que el programa utilitzarà per llegir i escriure dades.

El programa en qüestió es distribueix mitjançant un arxiu .APK que es genera a partir de l'Android Studio i que té un pes d'aproximadament 5 Mb. S'ha de tenir en compte que el que realment ocupa serà un cop s'hagi actualitzat amb la bdd i descarregat totes les imatges.

### **2.6.1- Desplegament e instal·lació**

L'aplicació de moment no està disponible a la botiga d'aplicacions de Google (Google Play) i per tant només es pot instal·lar mitjançant el fitxer .APK que s'ha generat.

Per instal·lar l'aplicació simplement s'ha de copiar a la memòria del telèfon i mitjançant un explorador de fitxers executar-la. Quan s'executi, ens demanarà una sèrie de permisos necessaris i s'hauran d'acceptar ja que sinó, no funcionarà com caldria.

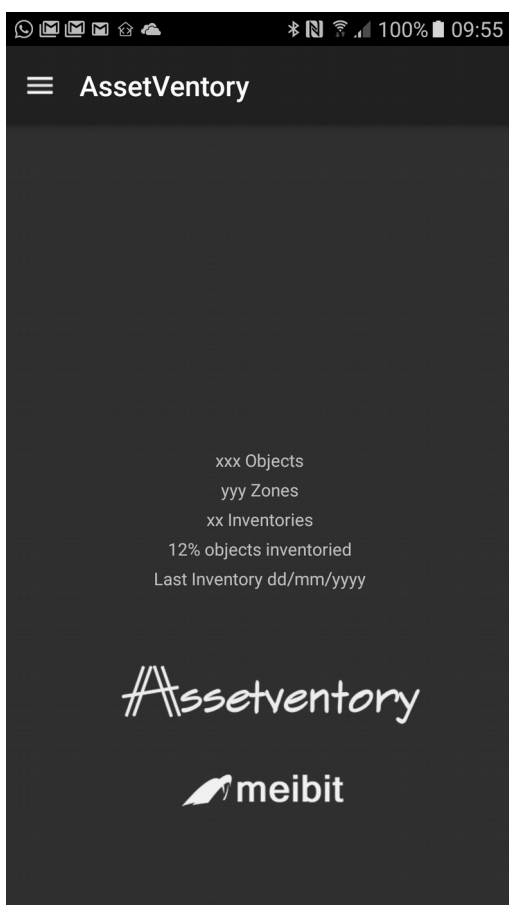
### 3.- RESULTATS

---

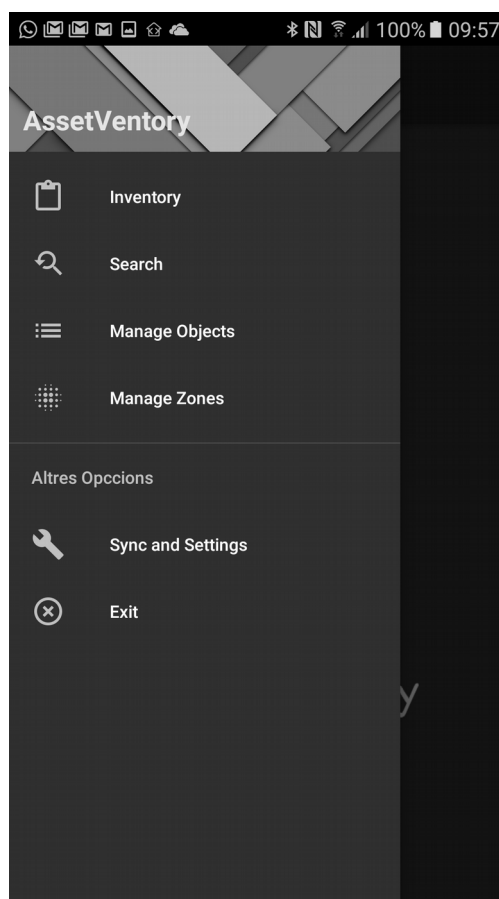
Una vegada finalitzada la fase de desenvolupament s'ha obtingut una primera versió estable de l'aplicació amb les funcionalitats requerides. A continuació mostrarem diferents captures de pantalles que mostren en detall el funcionament i que poden servir a mode de manual d'usuari

#### 3.1.- Accés a l'aplicació

Per accedir a l'aplicació ens trobarem en una pantalla on ens demanarà l'usuari i password. Aquestes dades s'han de configurar prèviament a les opcions de configuració. Per defecte, l'aplicació no en requereix i es deixa en mans de l'usuari la seva utilització. Cal recordar que el propi sistema Android té mesures de control d'accés al dispositiu i que no seria necessari posar-ne a l'aplicació. Però això ja es deixa en mans del client.



*Il·lustració 19: Pantalla d'inici*



*Il·lustració 20: Menú principal*

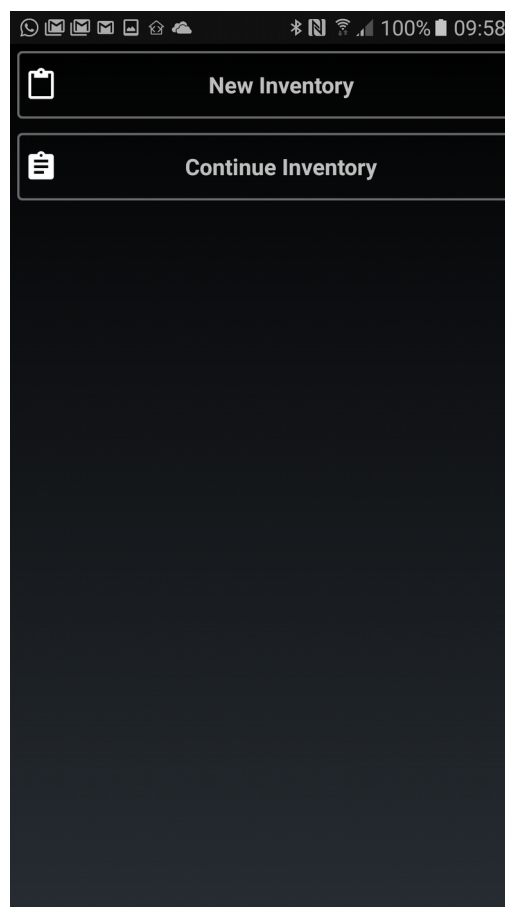
En primer lloc, es mostrarà un splash amb diversa informació general com els objectes, zones, número d'inventaris el % inventariat i la data de l'últim inventari realitzat. S'ha decidit donar aquesta informació en la pantalla principal d'inici ja que era un espai que es perdia inútilment i se li ha pogut donar una utilitat.

Mitjançant el botó de menú (a dalt a l'esquerre), es poden veure les opcions de que disposa el programa. Aquest menú és flotant i es pot accedir en qualsevol moment amagant-se o mostrant-se segons les necessitats (veure Il·lustració 20).

### **3.2.- Accés al mòdul d'inventari**

És el mòdul més important del programa i on s'han centrat realment els esforços ja que serà la part que més s'utilitzarà.

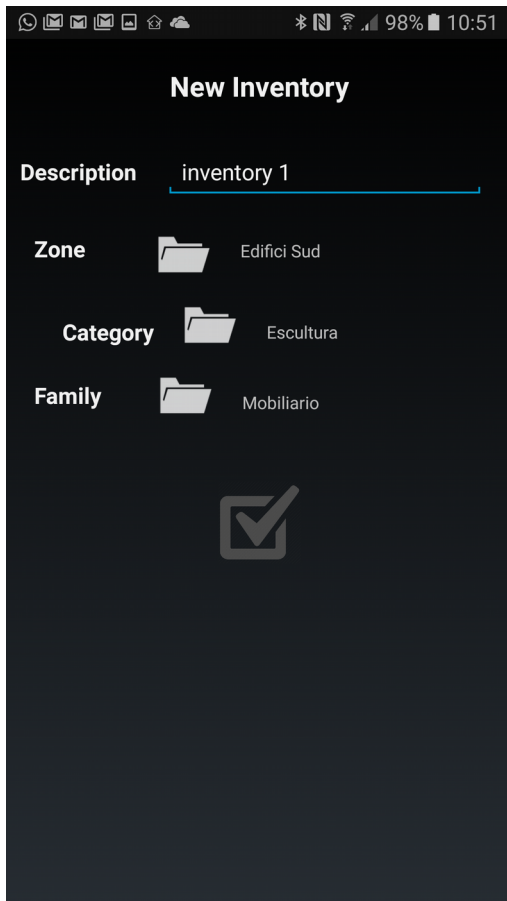
Com podem veure a la primera il·lustració, dintre a l'apartat d'inventari, tenim la opció de crear un nou inventari o bé continuar-ne un de nou.



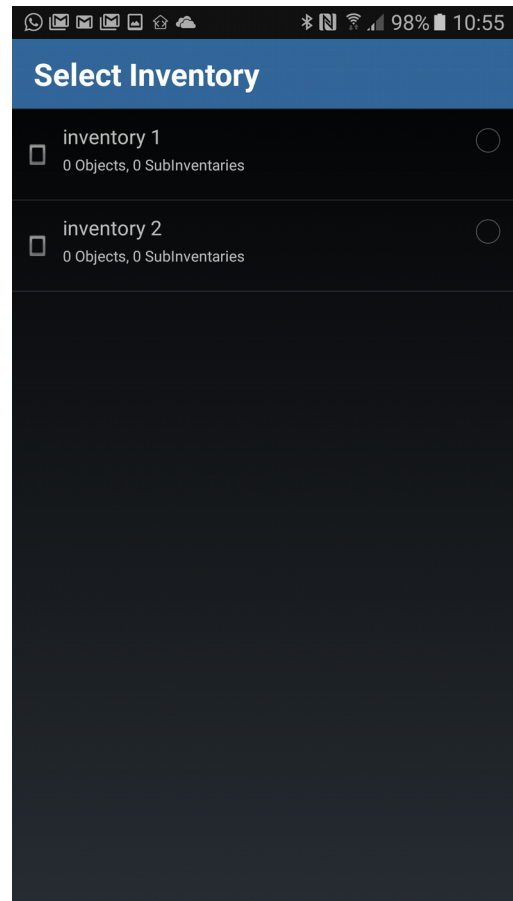
*Il·lustració 21: Submenú inventari*



Si seleccionem nou inventari , ens demanarà el nom, la zona a la que pertany aquest inventari, quina categoria d'inventari és (escultura,quadres,etc ..) i a quina família pertany. Un cop haguem entrat aquestes dades, ja ens apareixeran els objectes que hi ha dins la zona i podrem començar-hi a treballar.



*Il·lustració 22: Creació nou inventari*



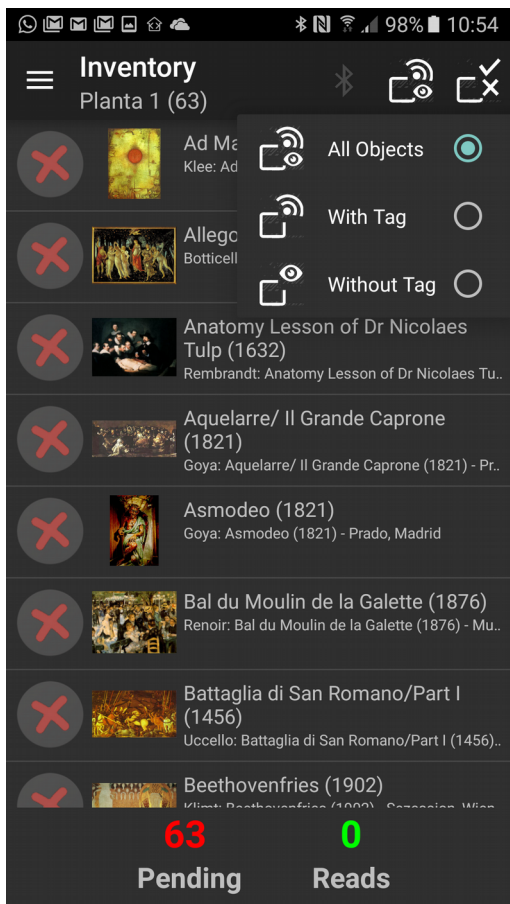
*Il·lustració 23: Selecció d'un inventari ja existent*

Per contra, si volem continuar un inventari prèviament començat, ens sortirà una llista on poder escollir-los.

Si hem seleccionat continuar inventari, podrem veure una captura de pantalla on es mostra la continuació d'un inventari que ja es va iniciar. Aquest inventari consta de 63 elements en total. En la captura de pantalla de l'esquerra, es pot veure que actualment hi ha pendents tots els objectes ja que encara no s'ha connectat cap lector RFID i s'ha iniciat la cerca d'objectes .L'indicador de la part posterior de la pantalla ens marca 63 objectes pendents i 0 de trobats.

En la captura de la dreta, es pot veure que ja s'ha iniciat la lectura i ha trobat 3 elements. Les icones de la llista canvien (es posa el tic en verd) i el comptador general mostra que hi ha 3 elements que ja

han sigut trobats.

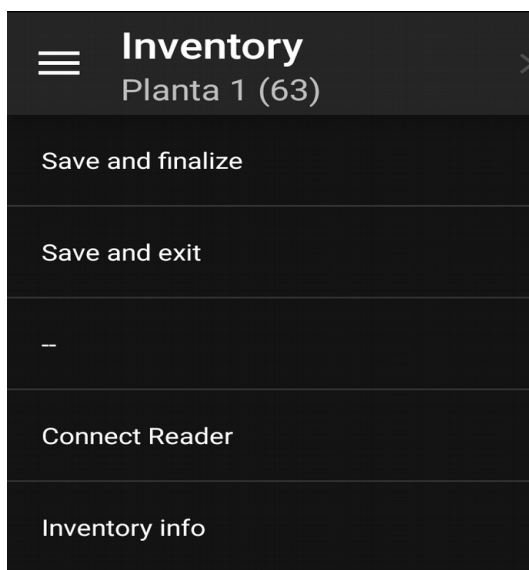


Il·lustració 24: Pantalla recompte objectes



Il·lustració 25: Pantalla recompte amb 3 objectes trobats

En la part posterior de les llistes podem veure les opcions de que disposem a l'hora de filtrar aquests objectes .Tenim diferents filtres: objectes amb tag, sense o bé tots. També podem veure només els que ja hem trobat, els que no o tots indistintament. Disposem d'un botó bluetooth per activar i desactivar el lector en tot moment.



Il·lustració 26: Detall menú inventari

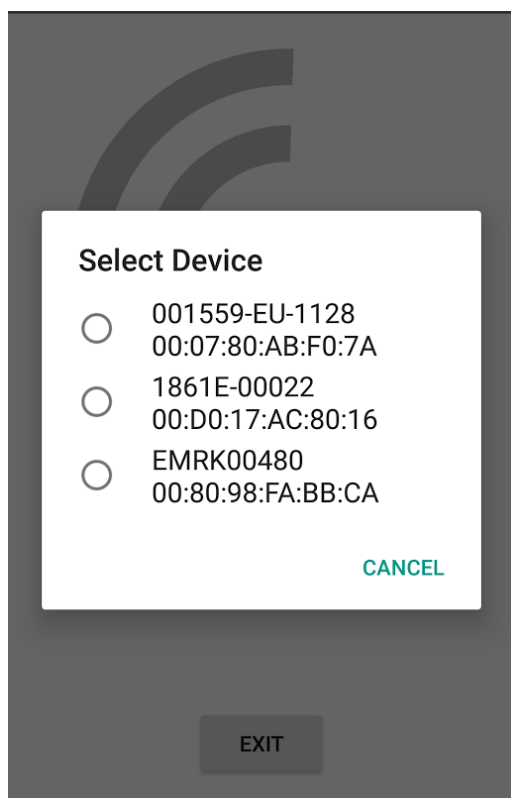
En el menú de l'esquerra superior disposarem de les típiques opcions per salvar l'inventari, connectar lector, informació , etc ..

Cal remarcar que si volem veure les propietats d'un objecte per veure'n les característiques i modificar algun paràmetre, només l'haurem de seleccionar a la llista i ens obrirà la pantalla de propietats (aquesta part es detalla més endavant en les opcions de la pantalla objectes)

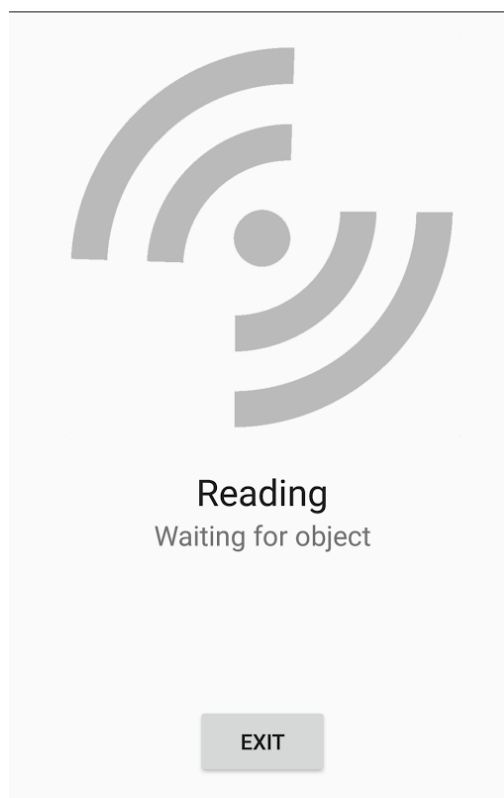
### 3.3.- Accés a mòdul buscar RFID

Aquest mòdul serà molt útil per obtenir informació ràpida d'un objecte en concret que tinguem al davant i en vulguem consultar la seva fitxa.

1. En primer lloc ens demanarà quin lector RFID volem enllaçar amb el mòbil.
2. Un cop s'hagi establert la connexió, procedirà a esperar que detecti un tag RFID
3. Un cop l'hagi llegit ens dirà que l'ha trobat dins l'inventari i ens mostrarà les seves característiques o bé ens dirà que no l'ha trobat dins la Bdd.



*Il·lustració 27: Menú selecció lector RFID*

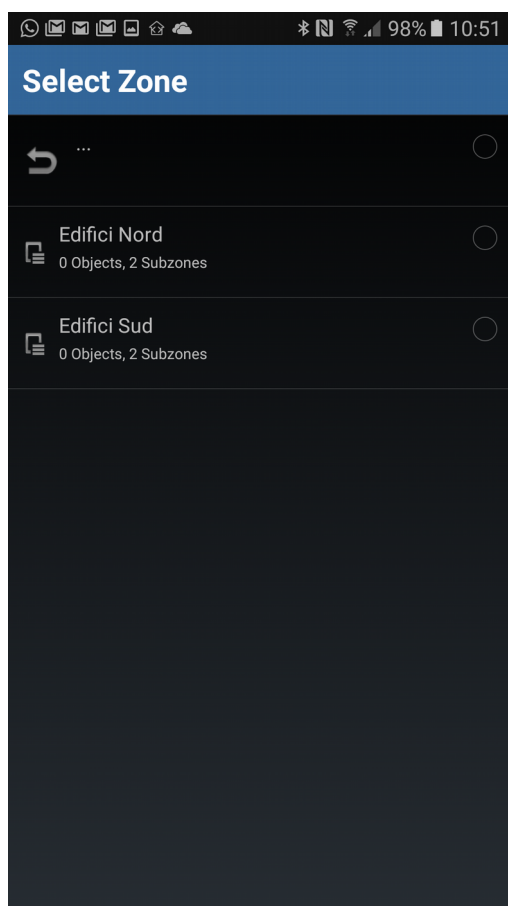


*Il·lustració 28: Pantalla espera objecte*

### 3.4.- Accés al mòdul Zones

Modul on podrem portar la gestió de les zones introduïdes.

Ens mostrarà una llista amb totes les zones que podrem seleccionar per a mirar la descripció, modificar-ne les dades o en última instància eliminar-la.

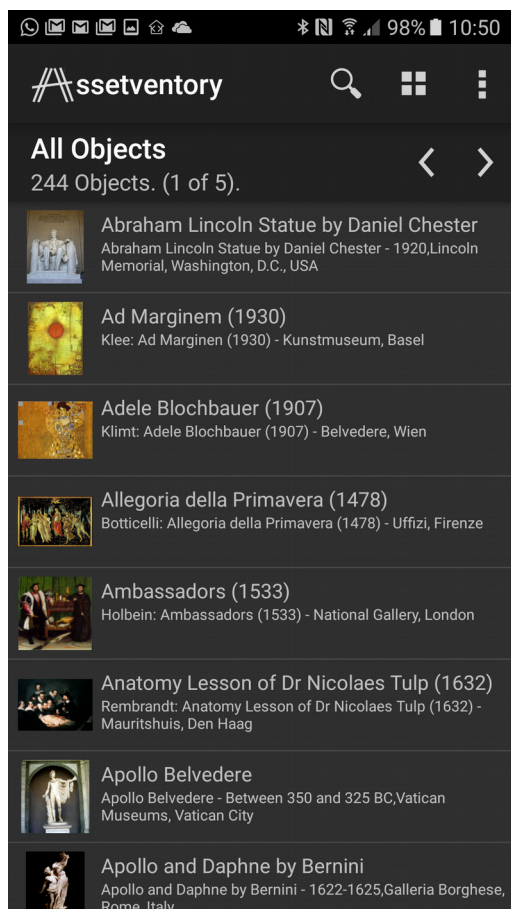


*Il·lustració 29: Selecció de zones*

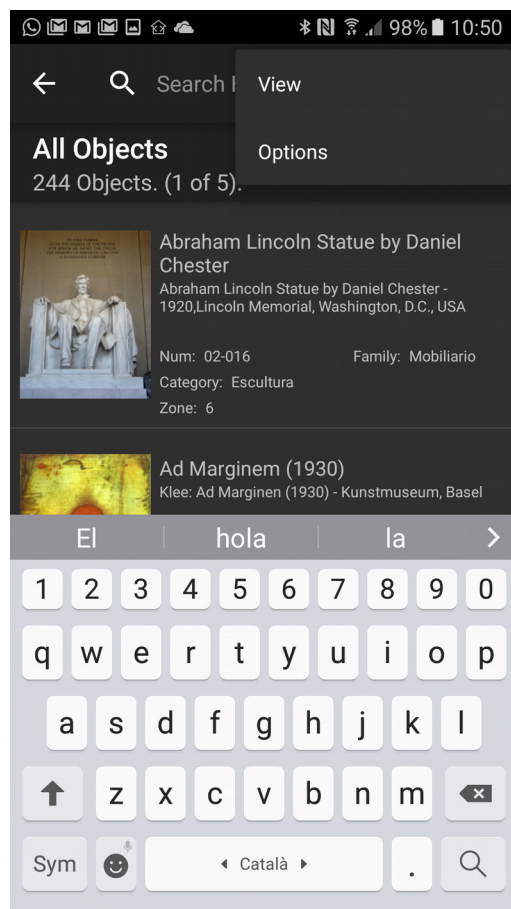
### 3.5.- Accés a mòdul Objectes

En aquest apartat portarem tota la gestió dels objectes de la base de dades. Altes, baixes i modificacions.

Primerament ens sortirà una llista dels objectes disponibles. Tenim 3 vistes de visualització diferents cadascuna amb les imatges més grans que l'anterior (mode llista, mode mosaic i mode miniatures). Per accedir al detall, quan en seleccionem un entrarem a la fitxa amb les seves característiques. Des d'allí podrem canviar el tag, modificar la fotografia, canviar-li la descripció, moure'l de zona etc ..



*Il·lustració 30: Selecció d'objectes*

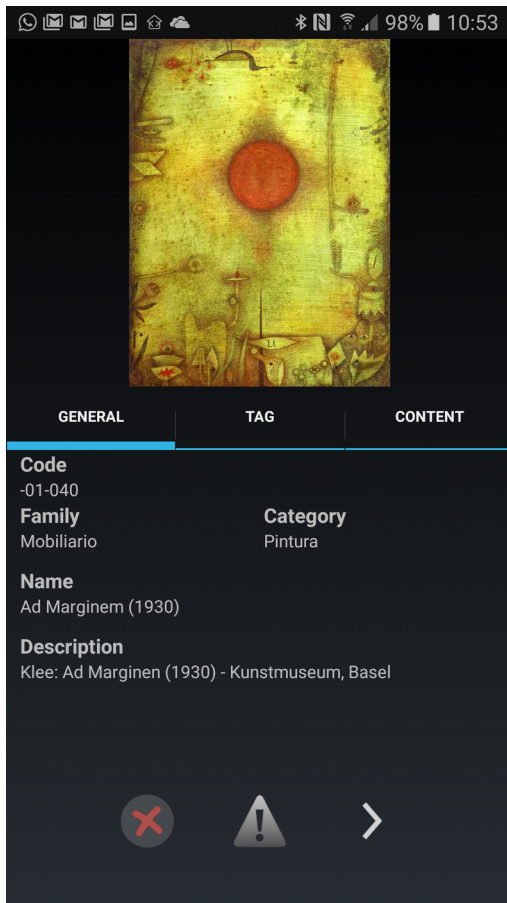


*Il·lustració 31: opcions de filtres/vista pantalla objectes*

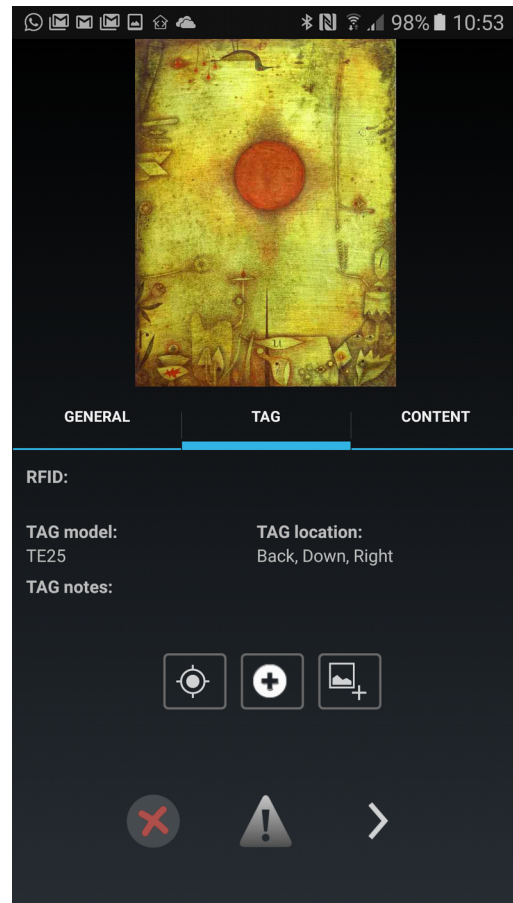
Per defecte, visualitza els 25 primers objectes per no tenir una llista massa grossa i poder moure'ns d'una manera coherent . Mitjançant les fletxes de navegació a la part de dalt podem passar a la següent pàgina i visualitzar el següent grup de 25.

En la captura de l'esquerra, veiem una vista de llista mentre que la de la dreta, és el mode mosaic. Podem cercar els elements per nom o bé ordenar-los alfabèticament o per zones per tal de facilitar la navegació.

Un cop hàgim seleccionat l'element que ens interessi, accedirem a la pantalla de propietats de l'objecte. Aquesta pantalla ens mostrarà la foto en gran i a la part de sota ens mostra les característiques de l'objecte. Estan agrupats en 3 pestanyes : General (Descripció, nom, categoria, etc..), Tag (informació tipus de tag, codi del tag) i Content (localització, zona on es troba etc ..)



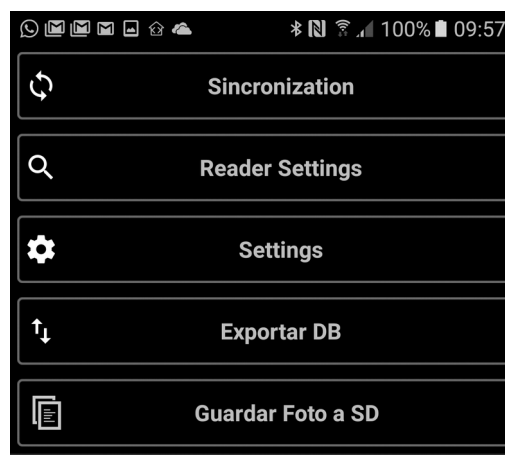
*Il·lustració 32: Pestanya general (detall objecte)*



*Il·lustració 33: Pestanya tag (detall objecte)*

### 3.6.- Accés a la sincronització de la Bdd i preferències del sistema

Aquí és on podem modificar els paràmetres generals de l'aplicació així com sincronitzar les dades amb la Bdd del servidor principal.

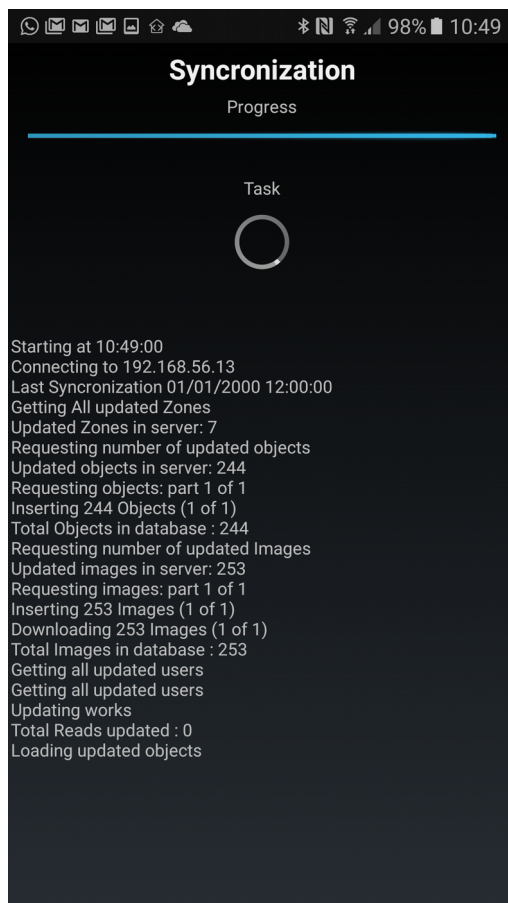


*Il·lustració 34: Submenú de preferències*

### 3.6.1.- Sincronització de la Bdd

És important que abans d'utilitzar l'aplicació es posi a actualitzar la bdd per tenir-la amb la versió més nova possible. Si aquesta opció es passa per alt, estarem treballant amb dades que possiblement ja s'hagin introduït anteriorment i s'acabi duplicant la feina.

El procés d'actualització acostuma a ser força llarg, variant segons el número d'objectes que s'hagin de descarregar / pujar i de la velocitat del servidor.



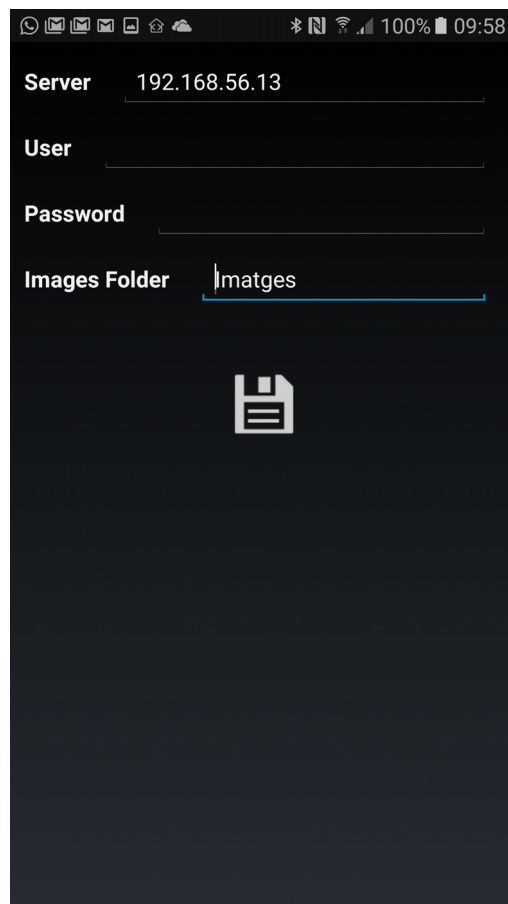
*Il·lustració 35: Sincronització amb la Bdd principal*

Mentre sincronitza, ens anirà detallant quines accions està fent en tot moment fent que l'usuari estigui informat que el procés va segons el previst.

### 3.6.2.- Preferències generals

En aquesta pantalla podrem introduir paràmetres com :

- Ip del servidor (Servidor on hi ha la bdd principal)
- Usuari
- Password
- carpeta on vulguem guardar les imatges (pel possible volum s'aconsella treballar a la SD)



*Il·lustració 36: opcions generals de configuració*



## **4.- CONCLUSIONS I EXPERIÈNCIA**

---

### **4.1.- Conclusions**

- S'ha desenvolupat l'aplicació AssetVentory per a dispositius mòbils a partir dels requeriments que va proposar el client a l'empresa MEIBIT.
- Al treballar dia a dia en un entorn com el Parc Científic i tecnològic de Girona i tenir la oportunitat d'intercanviar punts de vista amb altres empreses i persones del sector, puc afirmar que un gran nombre d'empreses utilitzen programari lliure tant a nivell de sistema operatiu com d'aplicacions i llenguatges per al desenvolupament.
- Tant el llenguatge JAVA com l'entorn de programació Android Studio ha ajudat a desenvolupar l'aplicació amb una certa rapidesa ,degut a la simplicitat i la bona conjunció entre el llenguatge i l'entorn de treball.
- La metodologia XP aplicada tant en el disseny com a la construcció de l'aplicació ha estat ideal per implementar en una empresa petita com aquesta ja que normalment no es disposa massa temps per planificar ni documentar les coses. Les persones que hi hem treballat s'han adaptat bé i per un projecte petit com aquest i treure una versió beta del producte ha estat una decisió ben presa.
- Després de fer diferents proves amb unes bdd que ens havien passat per part del client i on l'aplicatiu corria perfectament, un cop s'ha testejat amb la bdd real, s'ha vist que el rendiment baixava en picat degut al gran nombre d'elements. Per tant es va haver de posar a treballar en segon pla tota la càrrega de dades des de la crida de la bdd fins que s'omplien les diferents llistes i desplegable ja que deixava l'aplicació congelada durant 2 o 3 segons que feien que l'experiència d'usuari fos dolenta. Crec que treballar amb l'entorn real el més aviat possible et pot evitar un endarreriment en la planificació d'entrega.
- També es va haver de pujar el timeout de d'espera dels dispositius RFID ja que sinó es paraven automàticament massa aviat (per estalviar bateria) i després s'havia de tornar a repetir altra vegada el procés d'emparallament. Les sales són grosses i no s'havia tingut en compte per un temps d'espera més alt. Això vindria a complementar el punt anterior.
- Treballar amb un sistema de control de versions és indispensable en projectes on es col·labora conjuntament amb altre gent. Sense aquest tipus de funcionalitat s'haguessin fet molt complicat tot i ser una aplicació petita i un nombre de persones (3) relativament petit.
- Absolutament tot el desenvolupament de l'aplicatiu, bdd i redacció de la documentació ha

estat desenvolupat 100 % per eines de programari lliure. Constató una vegada més que a part de ser molt potents i amb una gran varietat de funcionalitats, en la majoria dels casos han superat les meves expectatives.

- Actualment l'empresa encara no està decidit amb quin tipus de llicència sortirà al mercat aquest aplicació. He arribat a la conclusió que encara la gent no té massa consciència de quin tipus de llicències hi ha ni quines avantatges e inconvenients suposen unes i altres.
- Treballar en una empresa petita té coses bones i dolentes:
  - **Bona:** No t'especialitza en una sola cosa i treballes en un ventall més ampli de fent més amena la feina.
  - **Dolenta:** Les coses no estan massa planificades i a vegades es cau en un cert bucle de treure's del cim la feina del dia a dia, sense organitzar-se .
- La desfragmentació del sistema operatiu Android així com la contínua sortida al mercat de noves versions i tecnologies fa que les aplicacions quedin "obsoletes" ràpidament i suposin un cost extra per a les empreses en dedicació i recursos humans.

## 4.2.- Experiència

L'experiència de les pràctiques ha estat molt positiva. Realment amb les 160 hores que hi he dedicat no he pogut fer massa feina efectiva.

En primer lloc, he hagut d'aprendre des de 0 tot l'entorn de treball. L'Android Studio i les seves particularitats no són massa complicades però requereixen unes hores per sentir-te còmode i al corrent de totes les opcions que té. Les constants actualitzacions del gradle i dels diferents SDK's fa que l'entorn estigui constantment canviant i et genera neguit ja que a vegades es requereix un parell de reinicis perquè tot torni a la normalitat. Un altre punt que no està massa ben resolt és la debugació ; a vegades costa localitzar en quin punt concret està corrent l'aplicació.

L'altre dificultat ha estat aprendre tota l'estructura de l'Android. Saber que és una Activity, com cridar-la, que és un layout, com s'instancien els objectes, les crides a diferents classes d'Android ... ha resultat una mica complicat. Al ser un aplicació no massa grossa en volum m'ha ajudat tot i que algunes funcionalitats hagués estat millor crear-les des de 0.

On he treballat més ha estat amb tota la part de lectors RFID . He hagut d'agafar cada un d'ells i mirar el seu SDK per tal d'adaptar-lo a l'aplicatiu. No ha estat gens fàcil ja que són molt extensos i no estava massa familiaritzat amb els threads i crides al sistema. Al final veus que tot i que cadascú utilitza metodologies diferents, el funcionament és força semblant.

La classe de la bdd també hi he treballat àmpliament tot i que en aquesta part he estat molt més còmode ja que tinc més experiència. Ha estat una feina més mecànica però haver de treballar contra un webservice que feia d'interpret entre la bdd i el programa ha estat una cosa que no havia fet mai i m'ha agradat.

Treballar amb un servidor de versions primerament m'ha costat ja que el concepte de branques i commits no el tenia massa clar i em feia por . Un cop agafats els conceptes i amb l'interfície de l'Android Studio configurada , he vist que és una eina indispensable.

Esmentar que la persona de l'empresa que s'encarregava de tutoritzar-me la pràctica va plegar després de 3 setmanes de començar. Des d'aquell moment es va fer complicat continuar ja que el cap va assolir el nou rol i el dia a dia de l'empresa feia complicat poder-li consultar els dubtes. Això ha fet que m'hagués de passar molt més temps per localitzar possibles solucions als problemes i haver de buscar informació per la xarxa.

Remarcar que gràcies a MEIBIT finalment he pogut finalitzar el període de pràctiques. L'anterior empresa amb la que havia de fer-les, em va deixar plantat pocs dies abans de començar i el temps de reacció va ser mínim. Això també ha estat un handicap per tothom. Encara que sigui d'una manera simbòlica vull agrair personalment amb en Miquel Àngel (gerent) que m'hagi ajudat en una situació realment compromesa.

## **Bibliografia**

- The Java Web Services Tutorial. <http://java.sun.com/webservices/docs/2.0/tutorial/doc/StAX.html>
- Android developers <https://developer.android.com/samples/index.html?hl=es>
- SQLite introduction [http://www.tutorialspoint.com/sqlite/sqlite\\_java.htm](http://www.tutorialspoint.com/sqlite/sqlite_java.htm)
- Creative commons licensing <http://creativecommons.org/>
- Cipherlab SDK <http://www.cipherlab.com/default.asp>
- Embisphere SDK <http://www.embisphere.com/en/rfid-products/embinventory>
- Tls SDK <https://www.tsl.com/>