Una aplicació mòbil per a la lectura de comptadors de xarxes de reg

Nom Estudiant: Francisco Javier Signes Mulet
Màster Universitari en Desenvolupament d'Aplicacions per a Dispositius Mòbils

Nom Consultor/a: Francesc D 'Assís Giralt Queralt

Professor/a responsable de l'assignatura: Carles Garrigues Olivella

6 de juny de 2018

© Francisco Javier Signes Mulet
Reservats tots els drets. Està prohibit la reproducció total o parcial d'aquesta obra per qualsevol mitjà o procediment, compresos la impressió, la reprografia, el microfilm, el tractament informàtic o qualsevol altre sistema, així com la distribució d'exemplars mitjançant lloguer i préstec, sense l'autorització escrita de l'autor o dels límits que autoritzi la Llei de Propietat Intel·lectual.

FITXA DEL TREBALL FINAL

Títol del treball:	*Una aplicació mòbil per a la lectura de comptadors de xarxes de reg*
Nom de l'autor:	*Francisco Javier Signes Mulet*
Nom del consultor/a:	*Francesc D 'Assís Giralt Queralt*
Nom del PRA:	*Carles Garrigues Olivella*
Data de lliurament (mm/aaaa):	*06/2018*
Titulació o programa:	*Màster Universitari en Desenvolupament d'Aplicacions per a Dispositius Mòbils*
Idioma del treball:	*Català*
Paraules clau	*Lectura comptadors, Android, comunitats regants*

Resum del Treball

L'aigua és un recurs escàs. Per això a l'hora de produir aliments en els camps s'ha de gestionar d'una manera eficaç. Durant els darrers anys els sistemes de reg han avançat molt en aquest sentit i el sistema de reg per degoteig s'ha implantat per molts llocs de la geografia espanyola. No obstant això, aquest avanç no ha evolucionat de la mateixa manera pel que fa al control i gestió de l'aigua usada.

El treball d'aquesta memòria se centra en el desenvolupament d'una aplicació mòbil per a la lectura de comptadors de comunitats de regants.

Aquesta aplicació pretén fer més fàcil i automatitzar totes aquelles tasques que duen a terme habitualment els lectors. Així mateix pretén augmentar la transmissió d'informació entre aquests i els propietaris dels camps regats. Finalment, també intenta minimitzar errors i optimitzar el temps de treball.

Per al desenvolupament del projecte s'ha dut a terme una planificació, un estudi de disseny centrat en l'usuari, una explicació de l'arquitectura global i la implementació d'una primera versió de l'aplicació.

Abstract

Water is a valuable resource. For this reason, it must be managed efficiently when we grow food. Over the last years irrigation systems have advanced a lot in this issue and the drip irrigation system has been implanted in many places in Spanish. However, this advance has not evolved in the same way in the control and management of the used water.
The work of this memory focuses on the development of a mobile application to read communities of irrigators meters.

This application aims to make it easier and automate all the tasks commonly carried out by readers. Morever, it aims to increase the transmission of information between these and the owners of the irrigated fields. Finally, it also attempts to minimize errors and optimize work time.
In order to develop this project it was configured a planning, a user-centered design studio, an explanation of the global architecture and the implementation of the application.

Índex

1. Introducció 5
1.1 Context i justificació del treball 5
1.1.1 Estudi de mercat 8
1.2 Objectius del treball 10
1.3 Enfocament i mètode seguit 10
1.3.1 Metodologia de desenvolupament 10
1.3.2 Estratègia triada 13
1.3.3 Anàlisi de riscos 16
1.4 Planificació del treball 17
1.4.1 Cost estimat dins del mercat 20
1.4.2 Viabilitat financera 21
1.5 Breu sumari de productes obtinguts 22
1.6 Breu descripció dels altres capítols de la memòria 23
2. Anàlisi i disseny 24
2.1 Usuaris i context d'ús 24
2.1.1 Usuaris passius 24
2.1.2 Usuaris actius 25
2.2 Disseny conceptual 27
2.2.1 Arbre de navegació 28
2.3 Prototip 29
2.3.1 Avaluació del prototip 36
2.4 Disseny tècnic 37
2.4.1 Actors 37
2.4.2 Casos d'ús 37
2.4.3 Diagrama de l'arquitectura del sistema 48
2.4.4 Diagrama UML d'entitats i classes 52
2.4.5 Diagrama UML de la base de dades 54
3. Implementació 56
3.1 Estructura de paquets i funcions de cadascun 56
3.2 Interfície d'usuari 59
3.2.1 Login 59
3.2.1.1 Detalls d'implementació 60
3.2.1.2 Proves 60
3.2.2 Menú i pantalla principal 61
3.2.2.1 Detalls d'implementació 61
3.2.2.2 Proves 61
3.2.3 Carregar ruta de lectura 62
3.2.3.1 Detalls d'implementació 62
3.2.3.2 Proves 63
3.2.4 Exportació lectures 63
3.2.4.1 Detalls d'implementació 64
3.2.4.2 Proves 65
3.2.5 Llegir ruta 65
3.2.5.1 Botonera 65
3.2.5.1.1 Detalls d'implementació 65
3.2.5.1.2 Proves 66
3.2.5.2 Presa de lectura 66

3.2.5.2.1 Detalls d'implementació 67
3.2.5.2.2 Proves 67
3.2.5.3 Ubicació de comptador en el mapa 68
3.2.5.3.1 Detalls d'implementació 68
3.2.5.3.2 Proves 69
3.2.5.4 Incidència abonat 69
3.2.5.4.1 Detalls d'implementació 69
3.2.5.4.2 Proves 71
3.2.6 Cercar abonat 72
3.2.6.1 Detalls implementació 72
3.2.6.2 Proves 73
3.2.7 Nova avaria 73
3.2.7.1.1 Detalls d'implementació 74
3.2.7.1.2 Proves 75
3.2.8 Eixir 75
3.2.8.1.1 Detalls d'implementació 75
3.2.8.1.2 Proves 75
3.3 Proves realitzades 75
3.4 Criteris d'optimització presos durant la implementació 77
3.5 Revisió de la planificació en l'entrega de la implementació 77
4. Conclusions 78
5. Glossari 79
6. Bibliografia 80
7. Annexos 82
7.1 Qüestionari d'avaluació del prototip 82
7.2 Manual d'instal·lació 84
7.3 Manual d'usuari 90
7.4 Fotos de comptadors per a fer proves OCR 97

Llista de figures

Figura 1: Percentatge tècniques de reg per comunitat autònoma 6
Figura 2: Evolució superfície terres de cultiu de secà i regadiu 7
Figura 3: Model cascada 11
Figura 4: Model Iteratiu 11
Figura 5: Model espiral 12
Figura 6: Model prototip 12
Figura 7: Model cascada amb retroalimentació 13
Figura 8: Característiques tipus desenvolupament aplicacions mòbils 14
Figura 9: Percentatge de dispositius per sistema operatiu venuts a 4T2016 15
Figura 10: Diagrama de Gantt del treball 19
Figura 11: Preu per hora desenvolupament aplicacions Android 20
Figura 12: Viabilitat econòmica del treball 22
Figura 13: Fitxa usuari passiu 1 - Albert Alemany 24
Figura 14: Fitxa usuari passiu 2 - Mariola Ferrer 25
Figura 15: Fitxa usuari actiu - Joan Torres 26
Figura 16: Pantalla login 30
Figura 17: Menú 30
Figura 18: Exportació de lectures al servidor 31
Figura 19: Importació de lectures des de servidor 31
Figura 20: Llegir abonat 32
Figura 21: Lectura OCR del comptador 33
Figura 22: Posició comptador al mapa 33
Figura 23: Incidència abonat 34
Figura 24: Compartir incidència 34
Figura 25: Introducció avaria 35
Figura 26: Cercador abonat 35
Figura 27: Resultat cerca 35
Figura 28: Informació d'un abonat 36
Figura 29: Diagrama UML de casos d'ús 38
Figura 30: Arquitectura del sistema 48
Figura 31: MVC vs MVP 50
Figura 32: Comparativa base de dades per a Android 51
Figura 33: Diagrama classes UML llegir comptador 52
Figura 35: Diagrama classes UML login, logout, main 52
Figura 37: Diagrama classes UML OCR, base de dades, avaria i utilitats 53
Figura 34: Model classes UML importar, exportar 53
Figura 36: Diagrama classes UML cercar abonat 54
Figura 38: Model de dades rutes lectura servidor 55
Figura 39: Model relacional de base de dades local 55
Figura 40: Estructura de paquets del projecte 56
Figura 41: Implementació login 59
Figura 42: Implementació menú 61
Figura 43: Implementació pantalla principal 61
Figura 44: Implementació pantalla principal amb lectura carregada 61
Figura 45: Implementació pantalla carregar ruta de lectura 62
Figura 46: Missatges pantalla exportació lectures 64
Figura 47: Pantalles implementació exportació lectures 63
Figura 48: Implementacions de la botonera 65
Figura 49: Implementació pantalla de lectura de comptador 66

Figura 50: Captura OCR de lectura 67
Figura 51: Implementació posició comptador en el mapa 68
Figura 52: Exemple compartir incidència per email 70
Figura 53: Exemple compartir incidència per Whatsapp 70
Figura 54: Implementació cercar abonat 72
Figura 55: Resultat cercar abonat 72
Figura 56: Detall d'un abonat cercat 72
Figura 57: Pantalla avaria 73
Figura 58: Pantalla enviament avaria 73

1. Introducció

1.1 Context i justificació del treball

El sector primari és un sector fonamental per a l'ésser humà perquè ens proporciona, entre altres, els aliments que ens permeten la subsistència. A més a més, també ens proveeix de la matèria primera per a les produccions industrials.

Dins dels sector primari, l'agricultura representa una activitat econòmica important. Des del paleolític [1] la nostra espècie ha aprofitat els recursos que ens ha donat el nostre entorn per tal de sobreviure.

El descobriment de l'agricultura va propiciar la Revolució Neolítica [1]. Vam passar de ser nòmades a ser sedentaris. Vam aprendre a treballar la terra perquè ens subministrara part de la nostra alimentació. Mitjançant el conreu d'arbres i plantes vam iniciar una nova forma de vida per abastar-nos de menjar.

Les plantes i els arbres, a l'igual que l'ésser humà, necessiten aigua per poder viure. Aleshores sempre ens ha preocupat com poder aconseguir-la. A l'inici creàvem les nostres llars a la vora de rius i llacs. Però després ens l'hem enginyada per poder emmagatzemar-la i canalitzar-la per poder fer que arribe a tots els indrets.

Els romans van ser grans enginyers hidràulics i aportaren noves tecnologies hídriques a les nostres ciutats. Però no fou així amb les terres de conreu perquè les espècies cultivades pertanyien a l'ecosistema mediterrani. Els cultius estaven adaptats als cicles de pluja del nostre clima mediterrani.

Per contra, els àrabs, a l'ocupar les nostres terres, van introduir noves plantes que venien d'àrees tropicals i subtropicals que necessitaven molta humitat per a poder proliferar. Aleshores van introduir tècniques de reg, canalització i emmagatzematge al camp per tal d'optimitzar l'ús d'aquest recurs tal valuós com és l'aigua [2] [3].

La seua tecnologia de reg per inundació (o també conegut com reg per gravetat – a manta o per inundació són els més coneguts [4] -) ha perdurat durant molts anys a la península. De fet encara és la forma de reg més usada en molts llocs de la nostra geografia. Però en gran part de la regió mediterrània del nostre país, i degut majoritàriament a la cada vegada més elevada mancança d'aigua, des dels darrers 20 anys s'han començat a introduir noves tècniques que permeten una millor gestió i ús dels sistemes de reg. Entre elles destaca el reg per degoteig. Aquest permet un ús més racional i òptim de l'aigua.

El 2013 en la Comunitat Valenciana, el reg per degoteig representava el 55,8% de l'aigua de reg, més de 40 punts percentuals que el 2000, la qual cosa suposa un increment del 209%. Continuar aquesta progressió o substitució del reg per gravetat (que encara acapara un percentatge elevat de l'aigua de reg, un 43,8%) és la millor alternativa d'estalvi per garantir la viabilitat del sector agrari. [5]

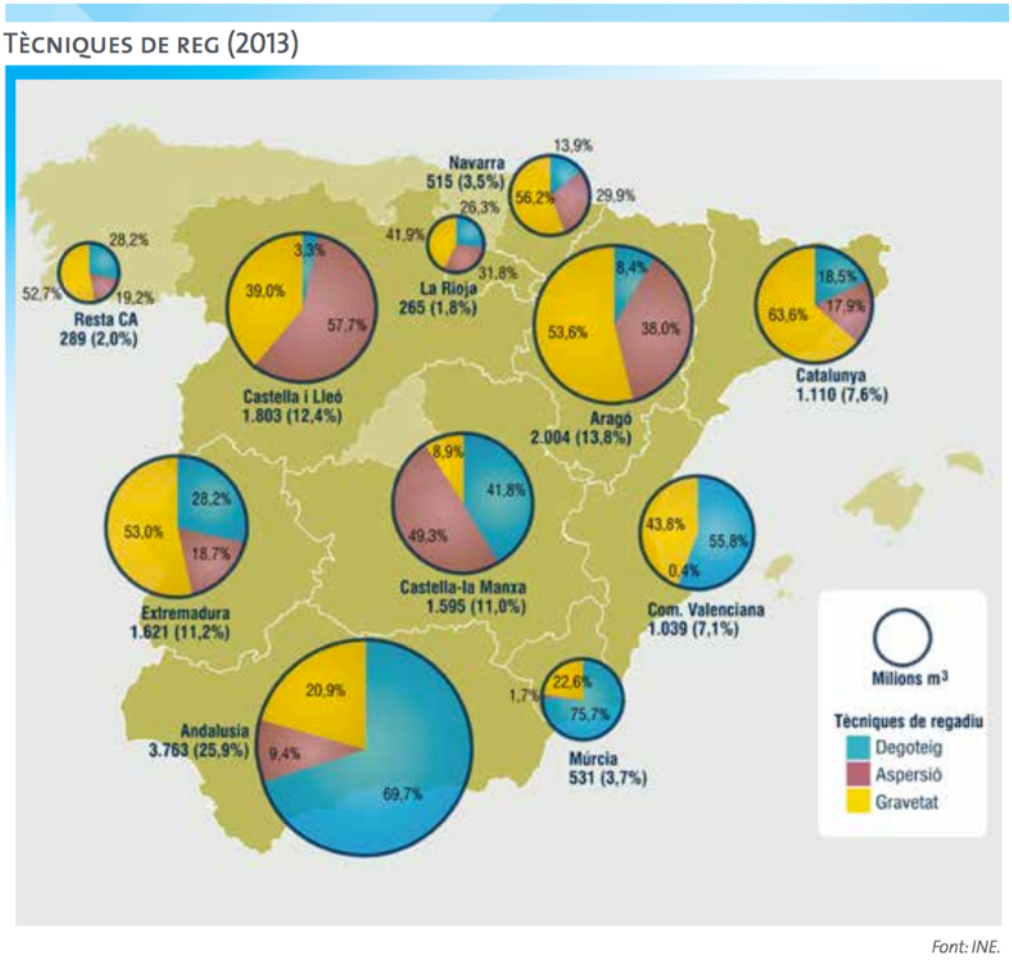


Figura 1: Percentatge tècniques de reg per comunitat autònoma

Del plànol anterior es pot deduir que moltes comunitats autònomes han implantat en els darrers anys aquest sistema de reg per degoteig i van a seguir implantant-lo.

Moltes comunitats de regants han modernitzat gran part de les seus instal·lacions i han introduït aquesta nova tècnica de reg. Això ha suposat, entre altres, la instal·lació de noves xarxes hidràuliques de tubs amb degotadors i comptadors d'aigua per tal de controlar l'ús d'aquesta per part de cada abonat o camp que es rega.

No obstant això, aquesta innovació hídrica no ha sigut acompanyada de la mateixa manera per la part de les tecnologies de la informació pel que fa al control i lectura dels comptadors. Alguns operaris encara continuen en molts llocs anant amb una carpeta amb fulls de paper o una petita llibreta anotant a mà les lectures amb els conseqüents inconvenients:

- possibilitat d'errada en l'anotació
- dificultat en trobar la ubicació física dels comptadors en un terreny si no va sempre el mateix empleat
- dependència de la persona per saber on estan situats els comptadors a l'hora de què agafe vacances, baixes mèdiques, ...
- ineficiència en el camí seguit per fer les lectures
- lentitud (i nova possibilitat d'error humà) al tornar a introduir la informació en els sistemes de facturació

És ací on **s'ha vist una veta de mercat per explotar**.

L'automatització d'aquesta part del sector representaria un factor diferencial per al mateix. Una aplicació mòbil que facilitara fer totes les tasques anomenades d'una manera més simple i ràpida suposaria un gran avanç. A més d'equilibrar la desmesurada diferència entre la gran innovació aconseguida en el reg per degoteig pel que fa a l'eficiència i estalvi d'aigua i l'obsolescència dels mètodes de presa de mesures d'ús d'aquests.

A més, les tècniques de reg més eficients que representen un factor d'estalvi i modernització del sector agrari valencià convé que altres comunitats autònomes les emulen.

De fet, el foment del reg per degoteig a les comunitats autònomes amb més recursos hídrics és una assignatura pendent que afavoriria la modernització del sector agrari espanyol.

Així mateix cal tenir present l'augment en la Comunitat Valenciana de la transformació de cultius de secà pels cultius de regadiu. Per la qual cosa es té, segons indica un estudi [6] i la gràfica que es presenta a continuació, que l'any 2017 és el segon any que els cultius de regadiu superen els de secà en aquesta comunitat i és una tendència en creixement.

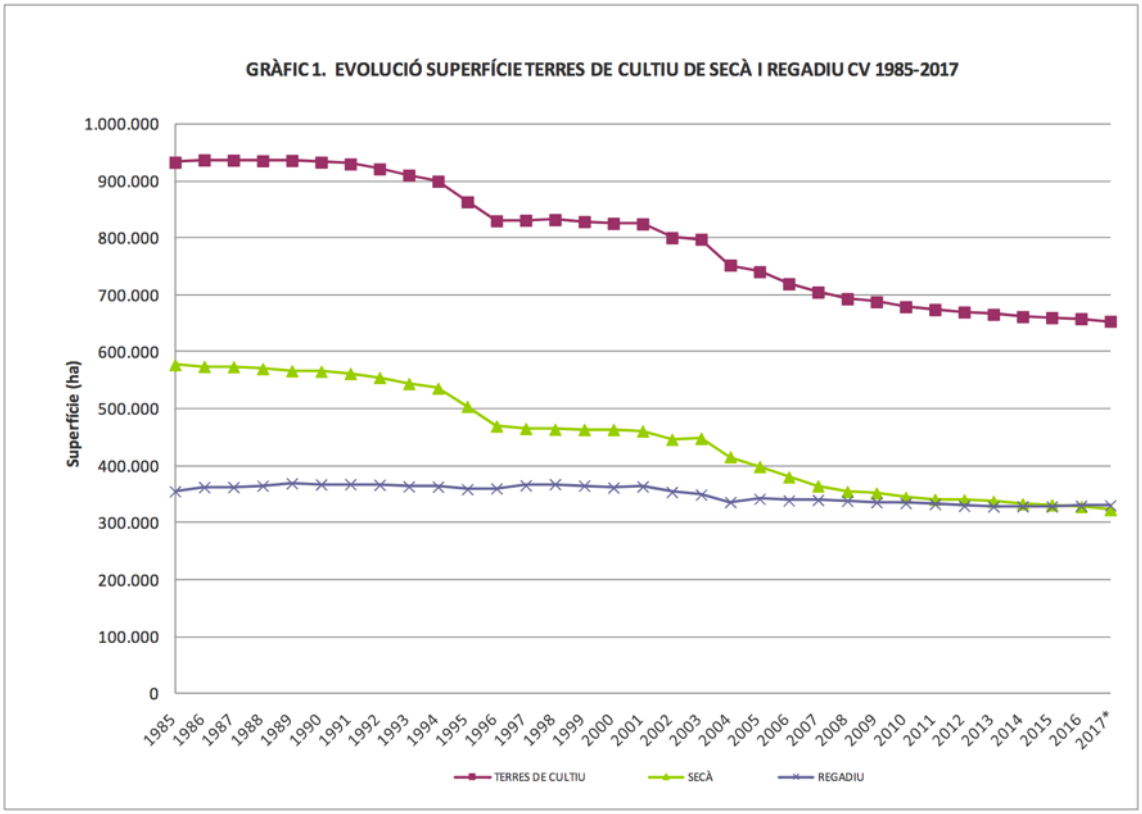


Figura 2: Evolució superfície terres de cultiu de secà i regadiu

Font: http://www.agroambient.gva.es [7]

Per tot el que s'ha indicat es veu la necessitat d'usar una aplicació mòbil per facilitar totes aquestes tasques.

1.1.1 Estudi de mercat

Identificada la mancança del mercat és fa un estudi previ per saber què existeix actualment.

Després de fer diverses tasques de recerca no s'han trobat aplicacions mòbils específiques per a lectura de comptadors de comunitats de regants. No obstant això, sí que s'han trobat algunes aplicacions que s'integren en sistemes propietaris de reg d'aigües potables. Per la qual cosa s'analitzen per tal de veure les característiques i mancances front a l'aplicació mòbil que s'espera en aquest treball.

Les aplicacions relacionades amb la lectura de comptadors d'aigua són les següents:

a) "Lectura de contadores" de l'empresa Galileo Ingeniería y Servicios [9].

Es tracta d'una aplicació privada que s'integra amb un programari propietari.
Permet la càrrega i descàrrega remota de lectura. A més també permet la introducció de la lectura.

b) "Lectura de contadores" de l'empresa Gestión Integral Ayuntamientos (GIA) [8].

A l'igual que l'anterior és una aplicació que s'integra amb un programari propietari.

Permet la introducció de lectures, veure informació d'abonats i introducció d'observacions així com la cerca d'abonats.

Finalment s'acompanya una taula resum de les funcionalitats que contenen aquestes aplicacions així com les funcionalitats esperades que hauria de tindre una aplicació ideal d'aquestes característiques:

App → Funcionalitat ↓	App Galilleo Ingenieria y servicios	App GIA	* App ideal
Importació / Exportació lectures remotes	?	SI	SI
Informació addicional abonat	SI	NO	SI
Introducció incidències	SI	SI	SI
Ubicació de comptador en mapa	NO	NO	SI
Presa de lectura automàtica a partir de foto (OCR)	NO	NO	SI
Compartir informació instantània amb abonat	NO	NO	SI
Login / Logout aplicació	NO	NO	SI
Cercar abonats	SI	SI	SI
Sistema propietari	SI	SI	NO
Informar avaria	NO	NO	SI
Possibilitat d'integració en qualsevol altre sistema de gestió d'aigües	NO	NO	SI

Com s'observa les aplicacions existents tenen funcionalitats reduïdes, són propietàries i no ofereixen possibilitat d'ús de les mateixes fora de l'àmbit d'actuació dels seus sistemes.

Per tot l'esmentat, es conclou que la tècnica de reg per degoteig és una tecnologia en augment i amb les mancances citades representa una gran oportunitat de negoci per desenvolupar l'aplicació mòbil plantejada en aquest treball de fi de màster donat que no existeix cap aplicació actualment al mercat que satisfà totes les necessitats plantejades.

1.2 Objectius del treball

El principal objectiu d'aquest treball és implementar una aplicació mòbil que permeta la lectura de comptadors en comunitats de regants així com l'enregistrament d'incidències en la xarxa d'abastiment d'aigua.

Per aconseguir aquest objectiu general s'han d'obtenir una sèrie d'objectius més concrets com ara:

1. Permetre la importació i exportació de rutes de lectura de manera telemàtica.

2. Obtenir un sistema de guiat per rutes que optimitze les tasques de lectura.

3. Proveir d'una plataforma base que serveisca per integrar les lectures amb altres sistemes de gestió i control de comunitats de regants.

4. Afavorir la possibilitat de trobar de manera senzilla sobre un mapa la ubicació determinada d'un comptador.

5. Accelerar el procés de lectura d'un abonat mitjançant tècniques de reconeixement d'imatges.

6. Permetre enregistrar incidències de la xarxa de manera immediata i localitzar-les de forma precisa en unes determinades coordenades.

7. Facilitar la comunicació amb l'abonat quan hi haja qualsevol tipus d'incident.

8. Proveir als lectors de ferramentes que els proporcionen la informació necessària i els assistisquen per dur a terme les seues tasques.

9. Integrar els coneixements que s'han adquirit durant el Màster per a implementar totes les funcionalitats que ha de realitzar l'aplicació mòbil.

1.3 Enfocament i mètode seguit

1.3.1 Metodologia de desenvolupament

Per desenvolupar projectes de tipus software existeixen diverses metodologies que ens permeten abordar la implementació del mateix [10] [11].

De les més importants es poden destacar les següents:

a) **Cascada**:

Es tracta d'una metodologia en la qual les etapes de desenvolupament del programari s'ordenen de manera que fins que no acabe una no s'avança a la següent. El desenvolupament va fluint cap avall per les etapes com si fos una cascada.

Les etapes per les que es passa són les següents:

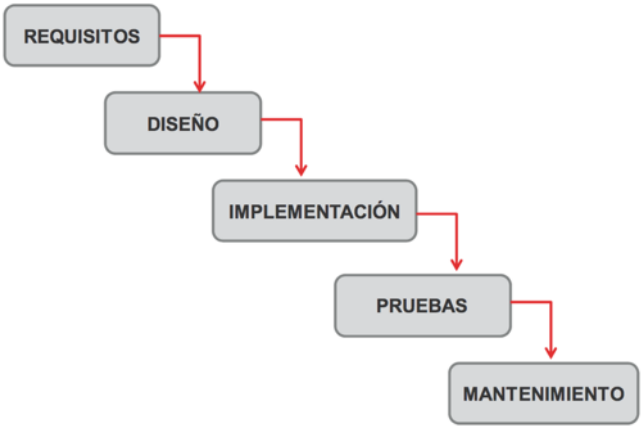


Figura 3: Model cascada

Com a avantatge cal mencionar que funciona bé en projectes on es coneguen tots el requeriments o es puguen predir a partir de les especificacions inicials.

Per contra, no es tenen resultats visibles i/o millores fins que està acabat. La qual cosa pot provocar inseguretat en el client. I també té una rigidesa pel que fa a la possibilitat de ser flexible i adaptar-se a modificacions que puga necessitar l'usuari final durant el desenvolupament del projecte.

b) Iteratiu

Es tracta d'un model que deriva de l'anterior. El que es pretén es minimitzar el risc derivat de què no s'hagen pres correctament els requeriments o no s'hagen entés de manera correcta.

Consisteix en la iteració de diversos cicles de vida en cascada. Al final de casdascun se li entrega al client una nova versió millorada de l'anterior. Així el client pot avaluar el producte i indicar millores.

Anem repetint iteracions fins satisfer al client.

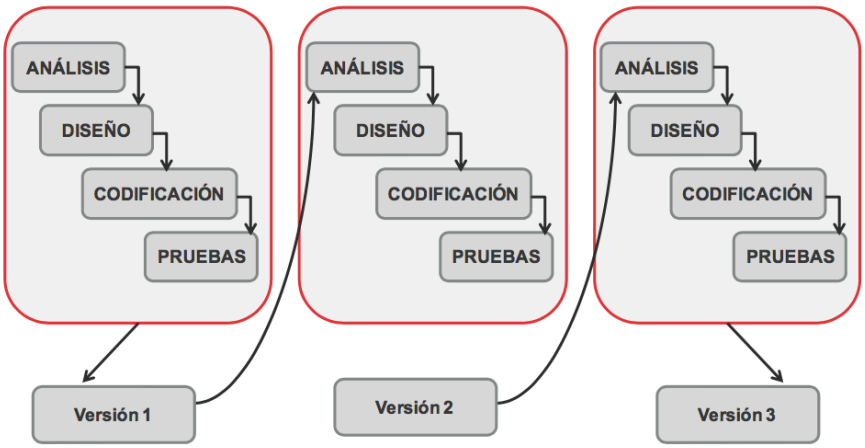


Figura 4: Model Iteratiu

S'usa normalment en projectes on no es tenen clars tots els requeriments al principi.

c) Espiral

Es tracta d'un altre model iteratiu. Però es diferència en l'anterior en què cada cicle es representa com a una espiral i en què avalua els riscos. En tots els cicles de l'espiral es diferencien 4 sectors: determinar objectius, anàlisi de riscos, desenvolupar, validar i planificar.

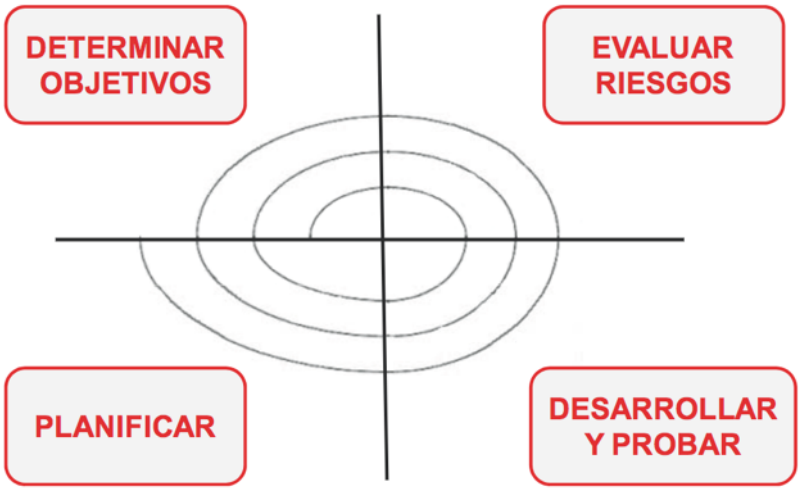


Figura 5: Model espiral

d) Prototips

Aquest model consisteix en prendre requeriments del client i anar construint prototips que el client valida.

Figura 6: Model prototip

Com a principal avantatge que té és que el client veu el producte des del primer prototip i pot experimentar amb ell. La qual cosa és millor que interpretar el que hi ha per escrit.

Per contra, el desenvolupament pot resultar lent i pot haver-hi cost en prototips que es poden descartar.

Després d'haver-se analitzar els diversos models, **el que s'emprarà per al treball serà una barreja de diverses metodologies**. **Es pot considerar com cascada però amb una variant denominada amb retroalimentació**.

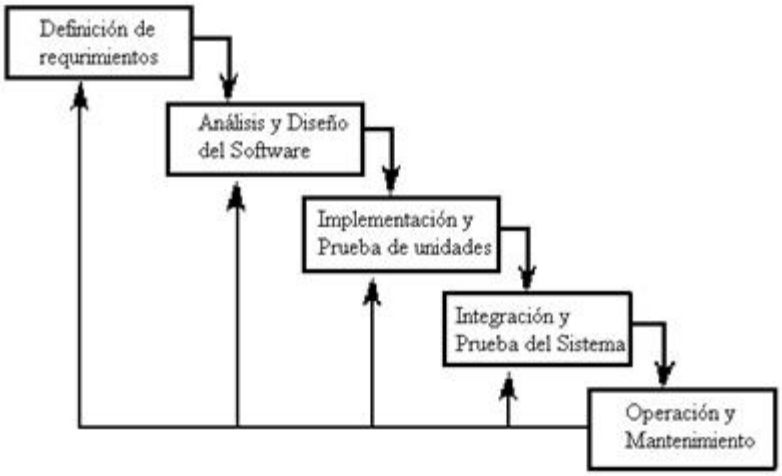


Figura 7: Model cascada amb retroalimentació

Aquest model admet la possibilitat d'iteracions, és a dir, durant les modificacions que es fan en el manteniment es pot veure, per exemple, la necessitat de canviar alguna cosa en el disseny i es podrà tornar a una etapa anterior al manteniment i recórrer la resta d'etapes.

La causa per la qual s'ha triat aquesta metodologia és perquè aquest treball compleix diversos principis que acoblen en la metodologia:

a) Es coneixen de bestreta els requeriments que ha de complir l'aplicació.
b) El projecte té una envergadura relativament curta. Aleshores les desviacions són menors.
c) Existeixen unes fites molt clares que coincideixen amb les entregues de les PACs.

1.3.2 Estratègia triada

Per tal d'abordar el desenvolupament d'una aplicació mòbil es podem tindre tres possibilitats [12], [13], [14], [15]:

a) **Aplicació nativa**: Aquesta aplicació estarà optimitzada per a un sistema operatiu determinat i la plataforma de desenvolupament del fabricant (Android, iOS, etc.). Aleshores s'hauran de desenvolupar dos aplicacions si es vol que l'aplicació funcione en Android i en iPhone**.** És la que té de totes les opcions un cost més elevat perquè caldrà fer l'aplicació dues vegades.

b) **Aplicació web**: És la més econòmica de desenvolupar donat que només s'implementa una vegada per a tots els sistemes operatius mòbils. Per contra té com a inconvenient que no pot accedir a totes les

característiques del dispositiu físic i la seguretat passa per la seguretat que tinga el navegador web.

c) **Aplicació híbrida**: Es tracta d'una solució que combina la tècnica d'aplicacions web però amb la particularitat de tindre extensions addicionals per poder accedir a característiques específiques dels dispositius mòbils (GPS, càmera de foto, etc.) com les aplicacions natives. Pel que fa al tema econòmic té un cost intermedi entre les dues anteriors.

Figura 8: Característiques tipus desenvolupament aplicacions mòbils

Es descarta una aplicació web donat que no es poden accedir a elements físics que requerirà l'aplicació com el GPS o la càmera. Aleshores queden el desenvolupament d'una aplicació nativa o una híbrida.

Tot i que l'aplicació híbrida té un cost menor de realització, pot accedir a característiques del terminal i pot fer-se servir en diversos sistemes operatius mòbils, s'ha decantat pel desenvolupament natiu pels següents motius:

a) A l'hora de fer el reconeixement d'imatges (OCR) per fer les lectures a partir de fotografies de comptadors es tenen més possibilitats de trobar APIs ja creats que permeten el seu ús i la integració en plataformes de desenvolupament natives.
b) La velocitat en aplicacions natives és més elevada que en aplicacions híbrides. I aquesta aplicació ha de ser ràpida a l'hora d'interactuar amb volum de dades de rutes de lectura.

Una vegada s'ha triat el tipus d'aplicació a realitzar cal prendre la decisió de per a quin tipus de sistema operatiu s'implementarà. Per això cal tindre presents diversos factors, com ara:

a) Sistemes operatius mòbils més usats actualment en el mercat.
   En aquest punt, fent tasques de recerca, i segons un estudi de la prestigiosa consultora de tecnologia Gardner [16], s'ha trobat la següent taula:

Table 3. Worldwide Smartphone Sales to End Users by Operating System in 4Q16 (Thousands of Units)

Operating System	4Q16 Units	4Q16 Market Share (%)	4Q15 Units	4Q15 Market Share (%)
Android	352,669.9	81.7	325,394.4	80.7
iOS	77,038.9	17.9	71,525.9	17.7
Windows	1,092.2	0.3	4,395.0	1.1
BlackBerry	207.9	0.0	906.9	0.2
Other OS	530.4	0.1	887.3	0.2
Total	**431,539.3**	**100.0**	**403,109.4**	**100.0**

Source: Gartner (February 2017)

Further information is available in the Gartner report entitled "Market Share: Final PCs, Ultramobiles and Mobile Phones, All Countries, 4Q16."

Figura 9: Percentatge de dispositius per sistema operatiu venuts a finals 2016

   En ella s'observa clarament que els 80,7% dels mòbils venuts al quart trimestre del 2016 eren Android.

b) Sistemes operatius dels mòbils més assequibles del mercat segons la relació qualitat/preu.
   Després de fer una feina d'investigació ([17], [18], [19]) es pot concloure que tots els terminals amb el millor cost/qualitat tenen un sistema operatiu Android.

Donat que el cost de desenvolupament d'aplicacions natives per als dos sistemes operatius més usats (Android i iOS) duplicaria el cost del projecte i que els terminals amb millors prestacions/preu tenen Android, *s'ha decidit* desenvolupar l'aplicació nativa només en Android. La probabilitat de què el client potencial tinga un terminal amb Android és major que la de tindre un terminal amb iOS (80,7% vs 17,7%). A més, en cas de què aquest no tinga terminal, el cost d'adquisició d'un nou serà més econòmic si és Android. Finalment, mencionar també que si el client volgués un terminal més robust (tipus PDA amb protecció contra pols i aigua – IP68 –) seria més fàcil trobar terminals amb Android [20] com ara el terminal *PDA Mustek Android MK-6000S.*

PDA Mustek Android MK-6000S

1.3.3 Anàlisi de riscos

Tots els projectes estan sotmesos a patir riscos i aquest treball, com a projecte, també. Per això cal estar-hi preparat per poder resoldre'ls.

S'acompanya a continuació una taula amb els riscos identificats del projecte, el tipus d'impacte que provocarà en el sistema i les mesures de contingència proposades:

Risc	Tipus d'impacte*	Mesura contingència
Desconeguts	Desconegut	S'han posat coixins de temps en les tasques d'implementació per poder subsanar-los
No integració (importació i/o exportació rutes de lectura) amb servidor	Gran	Simular integració amb fitxers locals
No guiat de lectures segons una ruta preestablida	Gran	Cercar abonats segons comptador actual a llegir
Impossibilitat de reconeixement de lectures a partir d'imatges (OCR)	Mínim	Introduir per part del lector les lectures amb el teclat
No haver-se mesurat correctament el temps de les tasques	Gran	Llevar-li funcionalitat o qualitat al producte

**Tipus d'impacte*:*

- *Desconegut*: riscos que poden eixir i que no es tenen identificats al principi del projecte.
- *Gran*: l'aplicació deixaria de fer una funcionalitat considerada important en el projecte.
- *Mínim*: si l'aplicació no tingués aquesta funcionalitat no seria un factor diferencial donat que es rectificaria aquesta funcionalitat d'alguna altra manera.

1.4 Planificació del treball

Per a dur a terme el treball es dedicaran un total de dues hores diàries de dilluns a divendres i dotze hores entre dissabtes i diumenges durant les fases PAC2 i PAC4 (entrega final).

Durant la fase d'implementació (PAC3) s'augmentarà la càrrega de treball setmanal a tres hores diàries mantenint les dotze hores del cap de setmana. D'aquesta manera es podran aconseguir les funcionalitats i qualitat del producte desenvolupat.

Pel que fa als recursos humans només es treballarà en el projecte de manera individual.

Referent als recursos materials s'empraran els següents:

- IDE Android Development Kit per a desenvolupar l'aplicació.
- Servidor per importar i exportar rutes. Es decidirà en la fase de disseny la solució més adequada.
- Gitlab com a plataforma per a control de versions.
- un processador de textos per redactar la memòria.
- un programari per fer presentacions en diapositives.
- un programari per gravar la presentació en vídeo.
- un ordinador per poder implementar l'aplicació.
- Gantt Project per fer el diagrama de Gantt del projecte.
- Ferramenta de prototipat d'aplicacions mòbils (Justinmind).
- Una eina per poder fer representacions amb UML (ArgoUML).

A continuació es detallen en una taula les tasques i la durada en hores estimada. Indicar que en les tasques ja es té present un coixí per previndre imprevistos.

Fase Disseny (PAC2)	**Total: 68 hores**
Nom tasca	*Durada en hores*
Usuaris i context d'ús	5
Disseny conceptual i arbre de navegació	3
Construcció prototip	20
Definició casos d'ús	23
Diagrama UML disseny base de dades	3
Diagrama UML entitats i classes	10
Arquitectura del sistema	4
Fase Implementació i proves (PAC3)	**Total: 140 hores**
Nom tasca	*Durada en hores*
Login a l'aplicació	4
Menú aplicació	4

Importar ruta	5
Exportar ruta	3
Pantalla bàsica abonat: lectura anterior, actual, codi abonat, núm. comptador, etc.	9
Pantalla avançada abonat: telèfon abonat, email, ubicació comptador, etc.	14
Logout	3
Gestió de rutes de lectura guiades	14
Informar abonat per whatsapp o email	3
Posicionar comptador en plànol	5
Cercar abonat i resultat de cerca	11
Recerca API OCR	20
Lectura automàtica comptador a partir fotografia	4
Exportació imatges	4
Proves	22
Documentar decisions implementació	15
Fase Lliurament final (PAC4)	**Total: 66 hores**
Nom tasca	*Durada en hores*
Preparar diapositives presentació	8
Preparació vídeo	17
Manual d'usuari	11
Manual tècnic	10
Finalitzar documentació memòria	20
	Total projecte: 274 hores

Finalment acompanyem un diagrama de Gantt amb la seqüenciació de les tasques:

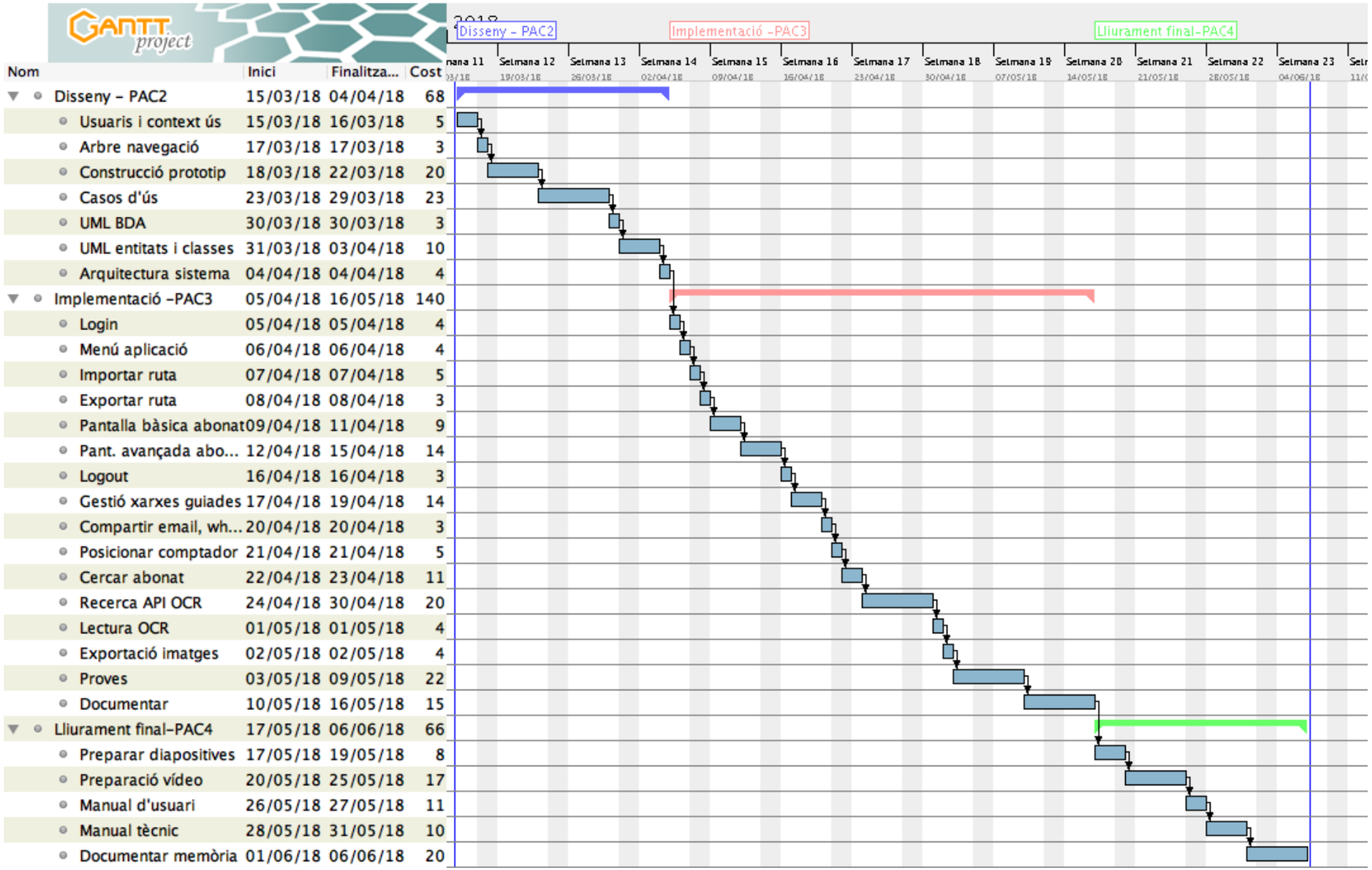


Figura 10: Diagrama de Gantt del treball

1.4.1 Cost estimat dins del mercat

Segons un estudi [21] es pot indicar que el preu per hora mitjà d'un desenvolupador freelance en Android en Espanya és de 27,5 €.

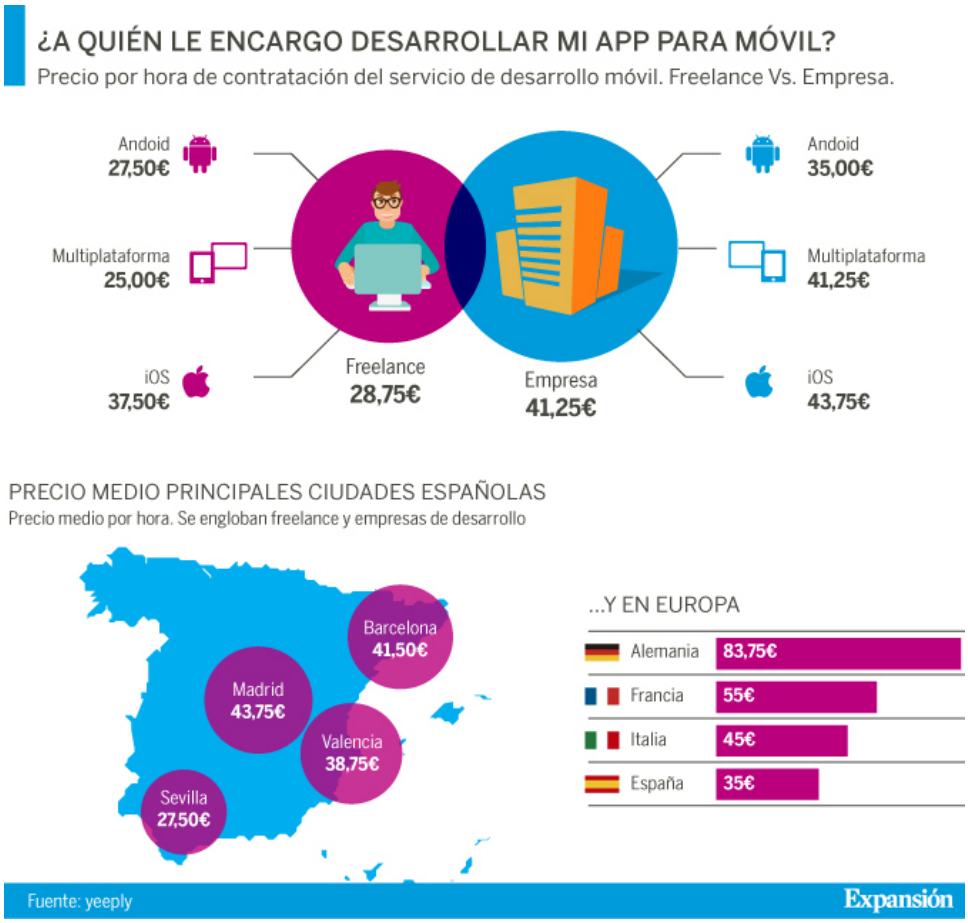


Figura 11: Preu per hora desenvolupament aplicacions Android

La durada en hores del projecte la podem extraure de l'apartat anterior al que caldria llevar la part de creació del vídeo i la part final de documentació de la memòria per no considerar-se necessàries en un projecte real. La part de la presentació en diapositives la podríem deixar per promocionar el producte.

Per altra banda caldria afegir el desplegament de l'aplicació a casa del client i la seua formació. Amb el que es pot indicar que el projecte tindria un total d'unes 275 hores. Per això el cost es pot determinar en:

275 hores x 27,5 €/hora = 7562,5 €

** Cal mencionar que no es té present la integració en plataformes de gestió de comunitats de regants. S'hauria d'estudiar en cada cas concret segons la comunitat de regants i el programari que es té implantat.

1.4.2 Viabilitat financera

Per tal de justificar la viabilitat financera del projecte s'ha hagut de fer diverses tasques de recerca.

En primer lloc indicar que s'abordaria la implantació inicial en la Comunitat Valenciana. De l'estudi [22] s'extrau que hi ha en aquesta comunitat un total de 289.000 Ha de regadiu. Si es creua aquesta informació amb l'obtinguda en la figura 1 que indica que en aquesta comunitat es tenen un 55,8% de cultius de reg per degoteig, s'obté que hi ha un total de 161.262 Ha. Si a més es creua aquesta informació amb un altre estudi [23] que indica que en aquesta comunitat la superfície mitjana per parcel.la en regadiu és de 0,52 Ha ens duu a tindre un total de 310.119 parcel·les (cada parcel·la representarà un abonat d'una comunitat de regants) de regadiu per degoteig.

Pel que fa al nombre de comunitats de regants s'ha tret informació de la Federació de Comunitats de regants de la Comunitat Valenciana [24] i s'han obtingut un total de 86 comunitat de regants que són membres d'aquesta Federació. Però no totes les comunitats de regants estan en aquest llistat donat que no tenen cap obligació de pertànyer a ella. Prova d'això ha sigut que en aquest article [25] en una comarca com La Safor hi ha un total de 41 comunitats de regants i només 2 són membres d'aquesta associació. Aleshores, extrapolant la informació es pot indicar que hi haurà un total de 1763 comunitats de regants en la Comunitat Valenciana. Segons aquestes xifres cada comunitat de regants tindrà una mitjana de 175 abonats.

Per tal de monetitzar la inversió s'ha establert el preu per ús de l'aplicació en funció d'una quota fixa (segons si les lectures són amb una freqüència mensual, trimestral, semestral o anual) més el preu per nombre d'abonats de la comunitat de regants per període de lectura. S'acompanyen unes taules amb les tarifes de preus:

	Lec. Mensual	Lec.Trimestral	Lec.Semestral	Lec.Anual
Quota fixa	40€	120€	240€	480€

Núm.abonats / Freqüència de lectura	0-99	100-499	500-999	1000-2500	Més 2500
Mensual	0,15€*	0,12€*	0,10€*	0,05€*	0,03€*
Trimestral	0,20€*	0,17€*	0,15€*	0,10€*	0,08€*
Semestral	0,25€*	0,22€*	0,20€*	0,15€*	0,13€*
Anual	0,3€*	0,27€*	0,25€*	0,20€*	0,18€*

* preus per abonat i lectura per període

Si es marca un objectiu d'abastar el primer any 6 comunitats de regants de 175 abonats amb freqüències de lectures trimestrals i el segon any abastar el doble s'obtenen les següents xifres:

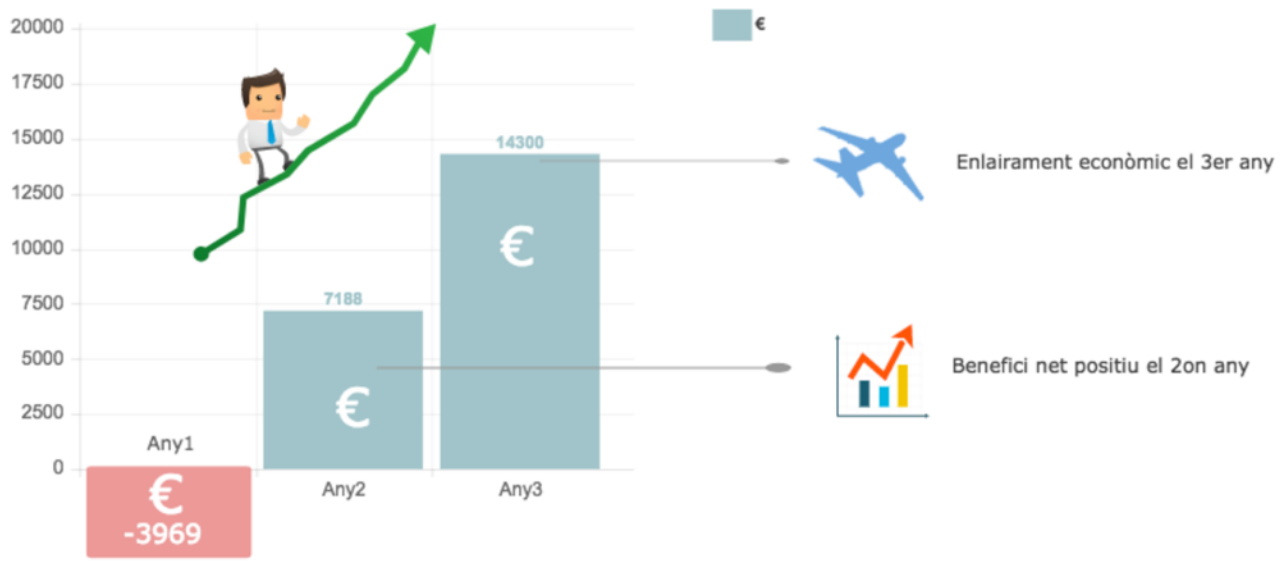


Figura 12: Viabilitat econòmica del treball

Com s'observa, a partir d'1,7 anys ja s'ha recuperat la inversió i hi ha beneficis.

** cal tindre present que no s'han considerat costos de lloguer d'oficines, llum, aigua ni despeses de servidors. Així mateix tampoc s'ha tingut present la integració amb els sistemes de gestió de rebuts de les comunitats que es facturaria apart. També cal notar que el nombre de comunitats de regants a cobrir són inicialment menudes i és un pla prou realista al ser poc ambiciós.

1.5 Breu sumari de productes obtinguts

En acabar aquest treball s'entrega la següent informació:

- Presentació del projecte desenvolupat que serveix per defensar el projecte davant d'un tribunal.
- Memòria de totes les fases del treball on s'expliquen tots els passos seguits.
- Aplicació mòbil per a Android.
- Codi font del programari.
- Manual d'usuari on s'explica com s'ha d'usar l'aplicació desenvolupada.
- Manual tècnic on s'inclouen totes les indicacions tècniques per poder compilar l'aplicació.

1.6 Breu descripció dels altres capítols de la memòria

En aquest primer apartat s'ha identificat una necessitat dins del mercat i s'ha proposat un producte per cobrir-la. S'han definit els objectius a cobrir per aquest producte així com la metodologia i estratègia que es seguirà per abordar-lo. Així mateix s'ha detallat la planificació inicial que es seguirà per realitzar la implementació del projecte. Finalment s'ha tractat el cost econòmic que pot tindre aquesta solució i la viabilitat financera del mateix.

En els pròxims apartats es tractaran altres fases que s'inclouen en tot projecte de desenvolupament de programari. Aquestes són:

1) Disseny

Ací s'aborda la definició dels usuaris i els contextos d'ús que es tindran en l'aplicació, la definició dels casos d'ús, l'arbre de navegació, el modelat de dades i la generació dels prototips de l'aplicació. Aquesta fase és necessària abans de dur a terme la programació purament dita perquè permet organitzar idees i identificar sinèrgies internes.

2) Implementació i proves

Aquest punt conté la descripció específica de la implementació real de l'aplicació amb els seus detalls. És ací on s'indiquen temes més tècnics que permeten aconseguir les funcionalitats esperades.

Paral·lelament també s'inclou els resultat de les proves que es desenvolupen per tal de garantir la qualitat del producte.

3) Conclusió

En aquest lloc s'inclouen les valoracions obtingudes després de finalitzar el projecte. És on se li donen cabuda a totes aquelles reflexions que permetran afrontar d'una manera millor els propers projectes. Perquè així, mitjançant la planificació i execució d'aplicacions, és com s'aprèn a gestionar els projectes de programari.

A més, també s'analitza el grau d'assoliment dels objectius plantejats a l'inici.

4) Annexos

En aquest apartat s'inclou un manual d'usuari on s'explica com s'usa l'aplicació. Així mateix s'inclou tota la documentació tècnica que és necessària per tal de poder fer funcionar l'aplicació. A més també conté el qüestionari d'avaluació del prototip que ha realitzat el client.

2. Anàlisi i disseny

En aquest apartat s'aborda la fase d'anàlisi i disseny de l'aplicació. Cal indicar que es farà un disseny centrat en l'usuari (DCU). Aleshores es centra el disseny de l'aplicació segons les necessitats reals dels usuaris finals. Amb el què s'aconseguirà una millor satisfacció i experiència del client amb el mínim esforç per la seua part.

2.1 Usuaris i context d'ús

Es poden classificar els usuaris de l'aplicació com a actius i passius.

Els usuaris actius són aquells que van a fer un ús directe de l'aplicació i per tant és molt important la seua opinió a l'hora de dissenyar l'aplicació.

Per altra banda estan els usuaris passius, que encara que no facen un ús directe de l'aplicació sí que van a obtenir un benefici indirecte. Aleshores també cal tenir present el seu punt de vista en aquesta fase de disseny per fer que el programari compleixi amb les màximes expectatives de tots els col·lectius implicats.

Com a usuaris actius es tenen els lectors de comptadors i com a usuaris passius es tenen per un costat els administratius que estan en les oficines usant un programa de gestió de lectures per facturar i per altra banda els abonats, que quan hi haja qualsevol incidència relacionada amb el seu comptador poden rebre una notificació immediata.

2.1.1 Usuaris passius

Figura 13: Fitxa usuari passiu 1 - Albert Alemany

2.1.2 Usuaris actius

Mariola Ferrer

"Cada cosa ha d'estar al seu lloc"

Edat: 38
Ocupació: Administrativa
Family: Casada, 1 fill
Localitat: Oliva

Personalitat

Introvertida — Extrovertida

Racional — Impulsiva

Passiu — Activa

Perfeccionista | Cuidadosa | Organitzada

Objectius

- Passar lectures al sistema informàtic amb el mínim temps possible i sense errades
- Imprimir fulls perquè els lectors puguen fer la seua feina.
- Optimitzar la seua feina per ser més eficient.
- Organitzar la feina dels lectors.

Frustacions

- Quan passa les lectures al sistema tindre errades.
- Haver d'esperar a l'arribada a l'oficina dels lectors per tal de poder avançar el seu treball.
- No poder comunicar de manera eficaç la informació d'on poden existir fugues en canalitzacions.
- No poder informar als abonats anticipadament quan tenen problemes al seus camps.

Un dia normal de treball

A les 8:00 arriba a l'oficina. Primerament imprimeix els fulls per a què els lectors puguen fer la seua feina. Després introdueix lectures que li ha portat el lector en paper al sistema informàtic. A les 15:55 arriba el lector. Arxiva la informació per poder passar-la a l'endemà al sistema.

Motivacions

Novetat

Por

Creixement

Poder

Social

Competència digital

Ordinador

Mòbil Android

Tauleta Android

Figura 14: Fitxa usuari passiu 2 - Mariola Ferrer

Joan Torres

"Treballar en contacte amb la natura emplena d'energia els nostres cossos."

Edat: 48
Ocupació: Lector
Family: Casat, 2 fills
Localitat: Gandia

Personalitat

Introvertit — Extrovertit
Racional — Impulsiu
Passiu — Actiu

Sociable | Curiós | Valent

Objectius

- Llegir els comptadors amb el mínim nombre d'errades.
- Avisar quan detecte anomalies en les canalitzacions.
- Evitar un malbaratament de l'aigua en abonats.

Frustacions

- Hi ha vegades que no recorda exactament on està la ubicació d'un comptador i perd temps per trobar-lo.
- Prendre lectures amb paper li resulta una tasca molt tediosa.
- Quan detecta una fuga i la reporta no li agrada que quan vagen a reparar-la li hagin de trucar per telèfon cada vegada perquè recorde on està ubidada.

Un dia normal de treball

A les 8:00 va a l'oficina, agafa una sèrie de fulls amb els comptadors que ha de llegir durant el dia. Després va llegint els comptadors de reg pels camps. En arribar a cada camps, obri el comptador i anota la lectura en el full.
Si detecta pel camí alguna avaria de les tuberies troncals o bé truca per telèfon a l'oficina o bé s'espera a tornar a l'oficina.
Cap a les 14:30 torna a l'oficina i li deixa els fulls amb les lectures a l'administrativa per a què passe les lectures.

Motivacions

Novetat
Por
Creixement
Poder
Social

Influenciadors

Competència digital

Ordinador
Mòbil Android
Tauleta Android

Figura 15: Fitxa usuari actiu - Joan Torres

Una vegada vistes les necessitats i objectius de cada usuari es podem extraure les funcionalitats que haurà de tindre l'aplicació. Aquestes estan directament relacionades amb el punt 1.2 on s'han fixat els objectius d'aquest treball. Per citar aquestes funcionalitats de manera breu podem dir que són:

- Automatitzar diverses tasques:
  - De càrrega i descàrrega de lectures
  - De presa de lectura mitjançant un programari
  - Integració automàtica i sense errades de les lectures al sistema de facturació
- Obtenir informació de manera immediata
  - D'avaries en les canalitzacions
  - Els abonats quan hi ha alguna incidència al seu camp

2.2 Disseny conceptual

Identificats els usuaris s'elaboren escenaris d'ús per descriure com s'usarà l'aplicació en determinats contextos. Amb aquests es determinaran necessitats del disseny i dels usuaris.

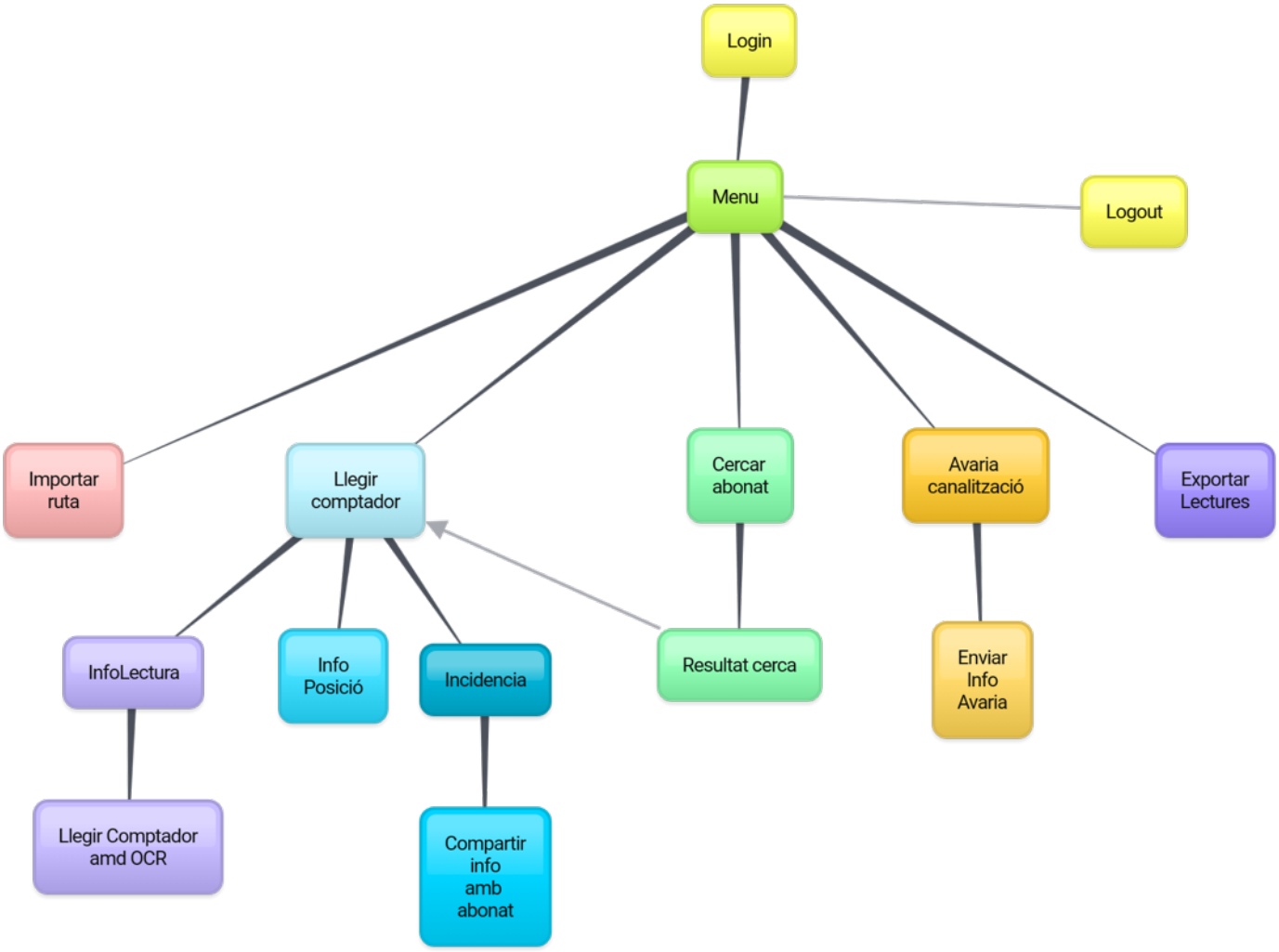
Escenari d'ús 1: Usuari actiu Joan Torres (Lector)
Són les 7:50 del matí. En Joan obri el mòbil de la feina, obri l'aplicació de lectura de comptadors, es valida i descarrega la ruta de lectura que ha de fer avui. Consulta quin és el primer abonat que ha de visitar, punxa en l'apartat "Ubicació" i veu en el plànol on està ubicat identificant la zona. Agafa el cotxe i va cap al lloc d'origen per iniciar les lectures. Una vegada en el primer comptador, treu el mòbil amb l'aplicació de lectura oberta i prem el botó que té una càmera per llegir el comptador. S'obri la càmera del mòbil i polsa damunt un rectangle blanc que està al damunt de la lectura del comptador. L'aplicació li posa automàticament la lectura. En Joan prem el botó "següent" abonat per saber quin és el pròxim comptador a llegir. Consulta novament en l'apartat ubicació i veu que és el comptador del camp que està a prop en una caseta. En la pantalla veu que posa "Alerta té un gos solt". Aleshores amb cautela va a llegir el comptador. Tot i que el gos està dins i el comptador es lleig des de fora pot ser s'espante. Una vegada presa la lectura detecta una "Possible fuga". Aleshores Joan tanca la clau de pas, li fa una foto al comptador i en l'aplicació polsa l'opció "compartir" per enviar-li per whatsapp a l'abonat la foto amb el text automàtic "Possible fuga". A mig matí anant cap al pròxim abonat veu una fuga en la canalització. Agafa el mòbil i tria en l'aplicació "Nova avaria". Treu una foto i marca l'opció enviar ubicació. Acte seguit envia la incidència per a què l'oficina estiga assabentada amb imatges i la posició GPS exacta de l'avaria. Són les 15:55 i Joan acaba de llegir l'últim comptador d'avui. En acabar amb l'aplicació tria l'opció d'enviar les lectures a l'oficina i marxa cap a casa. De camí a casa l'aplicació automàticament envia la informació.

Escenari d'ús 2: Usuari passiu Mariola Ferrer (Administrativa)
Són les 8:00 del matí. Mariola acaba d'entrar a l'oficina. El primer que fa es importar les lectures del dia anterior. Comprova que tota la informació està correcta i la integra en el sistema de facturació. Com que no ha de passar les lectures a mà no genera cap errada tipogràfica, evita tindre reclamacions derivades d'aquestes i dedica el seu temps a fer altres feines. A mig matí rep una avaria de la canalització que ha enviat el lector des de l'aplicació mòbil. Agafa el telèfon i avisa a la brigada de manteniment de les coordenades exactes facilitant-los les imatges i coordenades GPS. A les 15:30 prepara la pròxima ruta de lectura per a Joan. Quan acaba puja aquesta ruta per a què Joan l'endemà des de la seva casa puga baixar-se-la i anar directament a fer les lectures sense haver de passar per l'oficina.

Escenari d'ús 3: Usuari passiu Albert Alemany (Agricultor)
Són les 9:30 i Albert para a esmorzar després d'haver estat ruixant el tarongers de la partida de l'Espigola. Abans d'agafar l'entrepà mira el mòbil i veu un missatge de whatsapp que indica "Possible fuga" i una foto del lloc. Són les 16:30 de la vesprada Albert passa pel camp que li havien indicat al whatsapp i revisa la instal·lació. Malauradament un porc senglar n'havia fet de les seves: un forat a terra que havia soltat un tub de reg i l'aigua se n'eixia. Aleshores ho ha pogut detectar-la a temps gràcies a la informació rebuda per part del lector.

2.2.1 Arbre de navegació

Tenint present els usuaris i els contextos d'ús s'obté el disseny de quines seran les principals pantalles de l'aplicació així com l'accessibilitat que existirà entre elles. A continuació es detallen.



2.3 Prototip

Emprant els fluxos d'interacció definits en els escenaris d'ús s'ha realitzat el prototip de l'aplicació.

Per a dur a terme aquest s'ha usat la ferramenta anomenada "Justinmind". La causa per la qual s'ha utilitzat aquesta ferramenta és perquè ens permet el prototipat a alt nivell amb una semblança molt exacta de com quedarà l'aplicació desenvolupada per a Android. A més a més, aquesta ferramenta, també permet la publicació del prototip a una plana web i la navegació entre les diverses pantalles del prototip. Aquesta funcionalitat permet que l'usuari final puga veure realment com queda l'aplicació i donar la seua opinió al respecte abans de desenvolupar-se la mateixa.

Les pantalles de les que es composa el prototip es detallen seguidament.

En primer lloc tenim la pantalla de login.

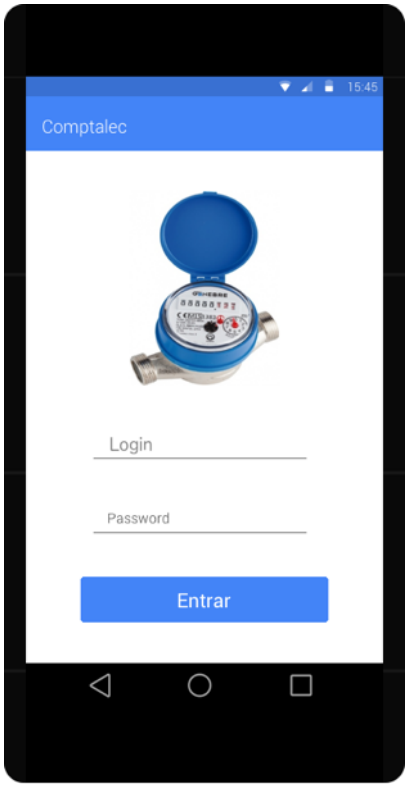


Figura 16: Pantalla login

Aquesta pantalla permet la validació de l'usuari per a què puga fer ús de l'aplicació. El disseny que s'ha realizat és el típic d'una pantalla de validació que té qualsevol aplicació. Amb açò s'aconseguiex que l'usuari final de manera intuïtiva puga accedir al sistema.

En segon lloc tenim la pantalla del menú principal de l'aplicació. Aquest s'accedeix des de la part superior de l'aplicació amb el clàssic menú de tipus "hamburguesa" que està present en moltes aplicacions mòbils.

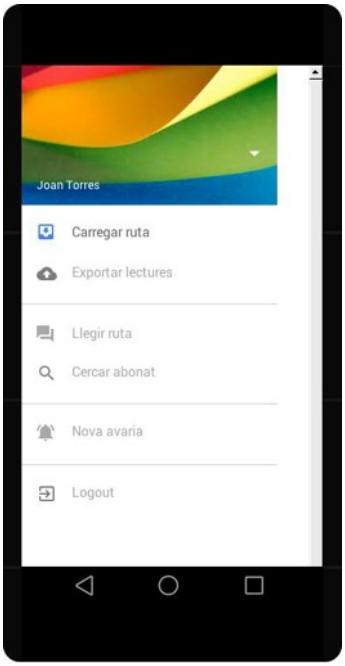


Figura 17: Menú

En aquest menú s'ha posat l'accés a les principals funcionalitats de l'aplicació que s'accediran mitjançant diverses pantalles.

Cal fer menció que s'han introduït una sèrie d'icones representatives de les opcions que realitza cada menú, així com que s'han ajuntat les funcionalitats per grups semblants.

Després es tenen les pantalles d'importació de rutes de lectura des d'un servidor a l'aplicació així com l'exportació de les lectures preses al servidor. Aquestes pantalles són semblants. El que s'ha considerat es posar un element animat que indica que l'aplicació està processant informació a més d'informar amb un text d'aquest procés també. Així l'usuari no s'impacientarà i pensarà que l'aplicació s'ha quedat blocada.

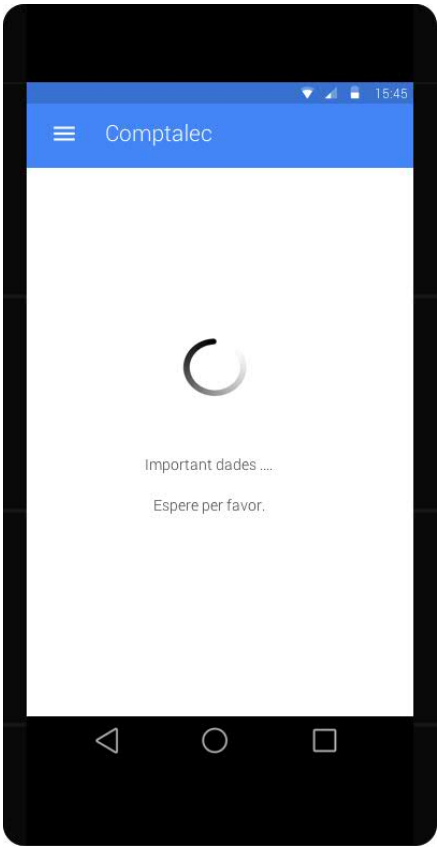


Figura 19: Importació de lectures des de servidor

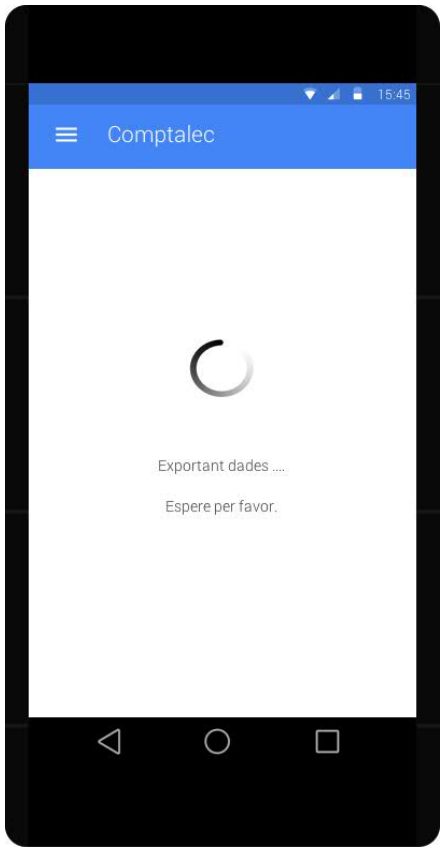


Figura 18: Exportació de lectures al servidor

A continuació es té la pantalla de presa de lectures d'un comptador determinat.

Figura 20: Llegir abonat

Per tal d'optimitzar les rutes de lectura de comptadors l'aplicació va guiant al lector per la mateixa. Per aquesta raó s'ha pres la decisió de tindre una botonera que permetrà anar al següent abonat, a l'anterior, al primer o l'últim de la ruta.

A més a més, també s'ha estructurat la informació de l'abonat en diverses pestanyes que permetran accedir de manera ràpida i directa a informació addicional que no cap en una única pantalla.

També s'ha ombrejat de color obscur tots aquells camps que només són de consulta perquè l'usuari puga diferenciar-los clarament d'aquells que pot introduir informació.

Com que les lectures es poden prendre de manera automàtica també a partir de què es puga capturar de la càmera del mòbil, s'ha dissenyat una pantalla per poder interactuar amb el lector de manera fàcil.

Aquesta pantalla permet capturar tots els textos que en aquests moments veu la càmera del mòbil i assenyalar-los mitjançant requadres blancs. Així el lector, per tal de facilitar-li la usabilitat, només ha de prémer amb el dit sobre el requadre blanc de la imatge que representa la lectura del comptador actual i l'aplicació automàticament li calcularà el consum d'aquest període. També cal remarcar que s'ha col·locat en aquesta pantalla un botó per tal de poder activar el flash de la càmera donat que el lector de vegades ha de prendre la lectura d'un comptador que està en un lloc poc il·luminat. La pantalla queda de la següent manera:

Figura 21: Lectura OCR del comptador

Una vegada situats en un abonat pot ser interessant determinar la ubicació exacta on està el comptador al plànol per poder accedir a ell. Aleshores l'aplicació té una pantalla que permet cobrir aquesta funcionalitat. Per al seu disseny s'ha optat per posar un plànol de GoogleMaps i marcar amb la icona clàssica la posició exacta del comptador.

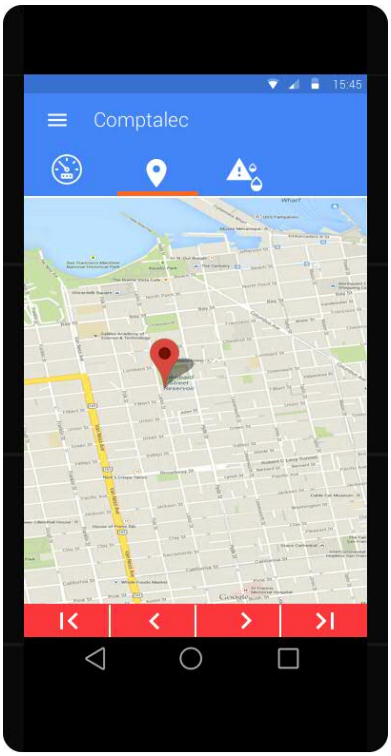


Figura 22: Posició comptador al mapa

També s'havia detectat la necessitat de poder introduir determinades incidències relacionades amb un abonat i poder comunicar de manera immediata amb aquest. Les pantalles que s'han creat es mostren a continuació.

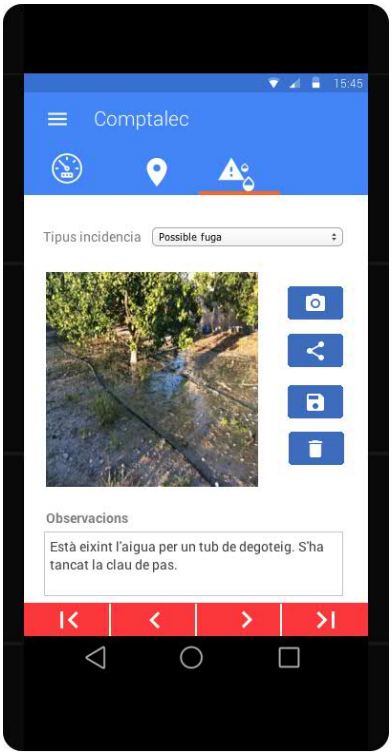


Figura 23: Incidència abonat

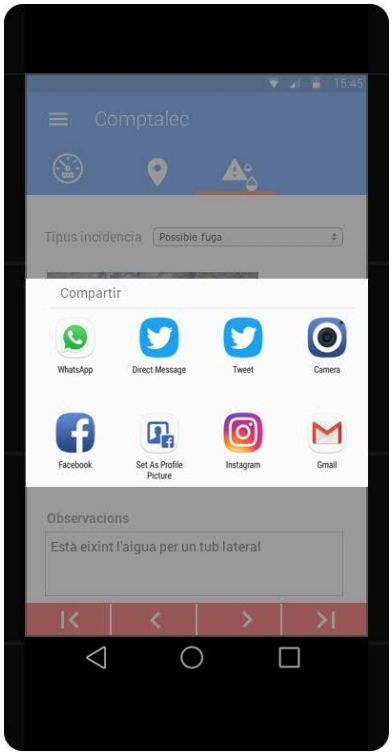


Figura 24: Compartir incidència

De la primera cal mencionar que es poden prendre fotografies de la incidència i que després es podran compartir amb l'abonat. Per a realitzar aquestes tasques s'han creat botons amb símbols representatius d'una càmera de fotos i el típic de compartició d'informació.

Una altra característica que havia de tindre l'aplicació era que havia de poder comunicar de manera ràpida qualsevol avaria que puga existir en la xarxa d'abastiment d'aigua. Per la qual cosa s'ha ideat una pantalla semblant a la d'una incidència d'un abonat perquè tenen moltes similituds (possibilitat d'afegir fotos, tipus avaria i un text lliure). A més, s'ha afegit un botó per poder consultar la posició actual i poder enregistrar també les coordenades GPS de la localització de l'avaria.

Per poder passar aquesta informació a la central de manera àgil s'ha considerat enviar-ho per correu electrònic adjuntant la foto i les coordenades GPS de la ubicació de l'avaria. Així tenim en el prototip un botó reservat per aquesta funció. Finalment el prototip d'aquesta finestra queda com segueix.

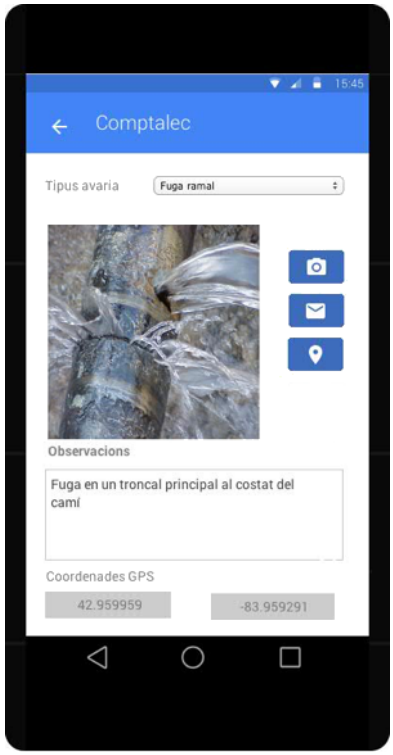


Figura 25: Introducció avaria

A més, s'ha detectat la necessitat de poder tindre un cercador d'abonats per poder de manera ràpida accedir a la informació del mateix. El prototip d'aquesta part s'ha dissenyat amb 3 pantalles: una que conté el cercador amb els seus criteris de cerca, una altra que té els resultats de la cerca i finalment la que conté la informació de l'abonat concret que s'accedeix després de prémer sobre un dels resultats de cerca.

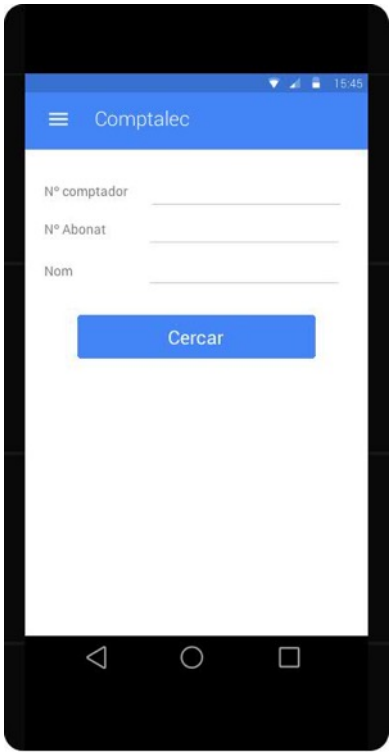


Figura 26: Cercador abonat

Figura 27: Resultat cerca

La finestra d'informació d'un abonat s'ha uniformitzat amb la de la pressa de lectura per tal de donar-li un mateix disseny, potenciar la usabilitat i facilitar-li la corba d'aprenentatge de l'aplicació a l'usuari final.

Figura 28: Informació d'un abonat

2.3.1 Avaluació del prototip

Un dels avantatges de la ferramenta emprada era que ens donava la possibilitat de pujar el prototip al web i poder navegar entre les diverses pantalles. La qual cosa ha permès realitzar una avaluació mitjançant l'observació directa. El prototip inicial es va publicar en la plana web i es va observar l'usuari com anava interactuant amb ell. A més també es va passar un qüestionari que està en l'annex 6.1 d'aquesta memòria.

Inicialment es va observar que l'usuari no sabia ben bé en la pantalla de Login quin era el camp per a posar l'usuari i quin era per a la contrasenya donat que no tenien un text indicatiu. Aleshores es va afegir. Després es va observar que l'usuari anava directament a llegir una ruta sense importar primer la informació del servidor. Aquest punt era propiciat perquè nosaltres havíem pensat que l'opció més important era la de lectura i l'havíem posada com a primera opció del menú. Després d'observar a l'usuari es van prendre com a mesures correctives el posar en les opcions de menú en primer lloc l'opció de carregar ruta. Així mateix es va prendre la decisió de què si l'usuari prem en l'opció de llegir ruta i aquesta està buida, que se li informe a l'usuari de què la ruta està buida i que cal primer carregar un ruta amb l'opció de menú corresponent. Una altra modificació obtinguda a partir de l'observació va ser posar en un color diferent la botonera de navegació dins d'una ruta de lectura per tal de què cridarà l'atenció de l'usuari i destaqués sobre la resta de la pantalla. Inicialment s'havia posat en color blau i estava massa integrada en la pantalla.

Per altra banda, i a partir del qüestionari emplenat per l'usuari, es va determinar que les icones de localització d'avaries i de compartició d'informació d'una incidència no eren inicialment les adequades perquè l'usuari no havia sabut trobar on estaven. Aleshores, consensuant-ho amb l'usuari, es van posar les icones actuals que eren més representatives segons l'usuari. En la part de les observacions del qüestionari l'usuari també va indicar que la distribució dels camps en la pantalla de lectura no era la més òptims per a ells. Aleshores es van redistribuir els camps segons les indicacions de l'usuari. Així mateix es va afegir un nou camp d'ubicació de comptador que no s'havia vist la necessitat en la presa de requeriments. Finalment l'usuari va suggerir també que la mesura dels botons de la botonera fora més gran.

Fruit d'aquesta avaluació s'ha modificat el prototip i ha quedat com s'ha detallat anteriorment.

2.4 Disseny tècnic

En aquest apartat es tracten les definicions dels casos d'ús. Aquests casos serviran per capturar els requisits funcionals de l'aplicació.

2.4.1 Actors

Un actor és qualsevol cosa que intercanvia informació amb el sistema i que és extern a ell.

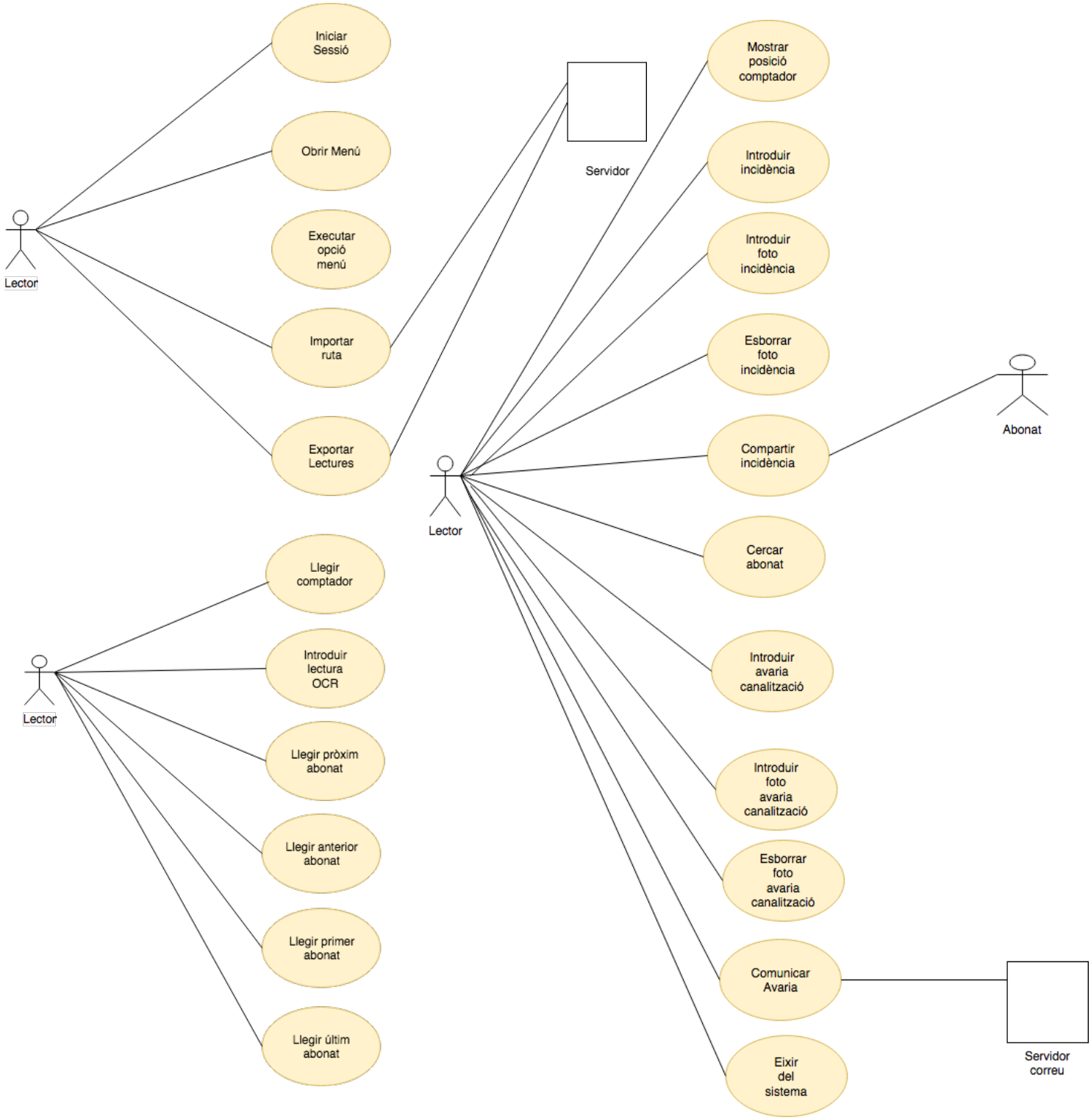
S'han identificat els següents actors:

- **Lector**: interacciona amb l'aplicació per explotar les seus funcionalitats. Treballa directament i freqüentment amb ella.
- **Abonat**: rep les notificacions dels sistema quan es comparteix una incidència.
- **Servidor**: element que permet la recepció de les lectures preses així com la descàrrega d'aquestes.
- **Servidor de correu**: permet l'enviament d'informació d'avaries.

2.4.2 Casos d'ús

Primerament es detalla un diagrama UML amb tots els actors i els casos d'ús més importants identificats:

Figura 29: Diagrama UML de casos d'ús



Acte seguit es detallen cadascun del casos d'ús.

Cas d'ús 01	Validar Usuari	
Descripció	Es permet que l'usuari valide la seua identitat en el sistema. Es mostra una pantalla amb dos camps de text on l'usuari introduirà el seu usuari i la seua contrasenya.	
Actor	Lector	
Precondició	L'usuari no està validat en el sistema.	
Flux	Pas	Acció
	1	L'usuari introdueix el seu usuari i contrasenya
	2	El sistema verifica que l'usuari i contrasenya són correctes
	3	El sistema dona accés a l'aplicació
Postcondició	L'usuari té accés a l'aplicació	
Excepcions	Pas	Acció
	2	Si l'usuari i la contrasenya no són correctes
	3	El sistema avisa amb un missatge d'error

Cas d'ús 02	Mostrar menú	
Descripció	Es mostra un menú amb totes les funcionalitat de l'aplicació disponibles.	
Actor	Lector	
Precondició	L'usuari està validat en el sistema.	
Flux	Pas	Acció
	1	L'usuari punxa en una icona
	2	El sistema mostra un menú lateral amb múltiples opcions
Postcondició	El sistema mostra un menú desplegat	
Excepcions		

Cas d'ús 03	Executar opció de menú	
Descripció	S'executa una opció de menú i s'executa una funcionalitat determinada.	
Actor	Lector	
Precondició	L'usuari està validat en el sistema i el menú està desplegat.	
Flux	Pas	Acció
	1	L'usuari tria una opció de menú
	2	El sistema executa la funcionalitat associada al menú
Postcondició	L'aplicació executa l'opció de menú seleccionada per l'usuari	
Excepcions		

Cas d'ús 04	Importar ruta	
Descripció	S'importa des d'un servidor la informació de les rutes de lectura de comptadors i es passa a una base de dades local.	
Actors	Lector, Servidor	
Precondició	L'usuari està validat en el sistema.	
Flux	Pas	Acció
	1	L'usuari tria l'opció de menú "importar rutes"
	2	El sistema connecta amb el servidor i es descarrega la informació
	3	El sistema emmagatzema la informació obtinguda del servidor en una base de dades local
	4	El sistema informa a l'usuari de què l'operació s'ha realitzat satisfactòriament
Postcondició	Les lectures estan emmagatzemades en una base de dades local	
Excepcions	Pas	Acció
	2.1	Si el sistema no pot connectar amb el servidor
	3.1	El sistema avisa amb un missatge d'error
	3.2	Si el sistema no pot emmagatzemar la informació en la base de dades local
	4.2	El sistema avisa amb un missatge d'error

Cas d'ús 05	Exportar lectures	
Descripció	S'exporten les lectures i les possibles fotos d'incidències existents.	
Actors	Lector, Servidor	
Precondició	L'usuari està validat en el sistema.	
Flux	Pas	Acció
	1	L'usuari tria l'opció de menú "exportar lectures"
	2	El sistema treu la informació de les lectures de la base de dades local i de les imatges de les incidències
	3	El sistema envia aquesta informació al servidor
	4	El sistema informa a l'usuari de què l'operació s'ha realitzat satisfactòriament
Postcondició	Les lectures estan emmagatzemades al servidor	
Excepcions	Pas	Acció
	2.1	Si el sistema no pot connectar amb la base de dades local
	3.1	El sistema avisa amb un missatge d'error
	3.2	Si el sistema no pot emmagatzemar la informació en el servidor
	4.2	El sistema avisa amb un missatge d'error

Cas d'ús 06	Llegir comptador	
Descripció	S'introdueix dins del sistema la lectura actual del comptador d'un abonat.	
Actor	Lector	
Precondició	L'usuari està validat en el sistema.	
Flux	Pas	Acció
	1	L'usuari es situa a un camp per introduir la lectura
	2	El sistema mostra un teclat numèric
	3	L'usuari introdueix números
	4	El sistema mostra dins del camp de lectura el número introduït
	5	L'usuari es mou amb la botonera situada a la part inferior
	6	El sistema emmagatzema la informació
Postcondició	El sistema emmagatzema la lectura actual	
Excepcions	Pas	Acció
	5.1	Si el sistema detecta que la lectura actual es menor que la lectura anterior
	6.1	El sistema avisa amb un missatge d'error

Cas d'ús 07	Introduir lectura OCR	
Descripció	S'introdueix dins del sistema la lectura actual del comptador a partir del reconeixement del que es veu en la càmera del mòbil.	
Actor	Lector	
Precondició	L'usuari està validat en el sistema i està dins de la informació d'un abonat	
Flux	Pas	Acció
	1	L'usuari prem sobre la icona d'obrir la càmera
	2	El sistema mostra una pantalla on es veu el que veu la càmera del mòbil
	3	El sistema reconeix automàticament tots els textos que hi ha en la imatge que està observant-se en la càmera
	4	El sistema emmarca cada grup de text amb una requadre blanc i a un costat es veu el text que reconeix a partir de la foto
	5	L'usuari prem en el requadre que conté la informació de la lectura del comptador
	6	Continuem en el pas 5 del cas d'ús 06
Postcondició	El sistema emmagatzema la lectura actual	
Excepcions	Pas	Acció
	5.1	Si el sistema detecta que la lectura actual es menor que la lectura anterior

	6.1	El sistema avisa amb un missatge d'error

Cas d'ús 08	Llegir pròxim abonat	
Descripció	Es passa a la lectura del pròxim comptador de la ruta.	
Actor	Lector	
Precondició	L'usuari està validat en el sistema.	
Flux	Pas	Acció
	1	L'usuari prem una icona de la pantalla
	2	El sistema mostra mostra la informació del pròxim abonat de la ruta de lectura
Postcondició	El sistema visualitza la informació del pròxim abonat	
Excepcions	Pas	Acció
	2.1	Si és l'últim abonat de la ruta el sistema no avança
	2.2	Si el sistema no pot accedir a la informació
	3.2	El sistema avisa amb un missatge d'error

Cas d'ús 09	Llegir anterior abonat	
Descripció	Es passa a la lectura de l'anterior comptador de la ruta.	
Actor	Lector	
Precondició	L'usuari està validat en el sistema.	
Flux	Pas	Acció
	1	L'usuari prem una icona de la pantalla
	2	El sistema mostra mostra la informació de l'anterior abonat de la ruta de lectura
Postcondició	El sistema visualitza la informació de l'anterior abonat	
Excepcions	Pas	Acció
	2.1	Si és el primer abonat de la ruta el sistema no avança
	2.2	Si el sistema no pot accedir a la informació
	3.2	El sistema avisa amb un missatge d'error

Cas d'ús 10	Llegir últim abonat	
Descripció	Es passa a la lectura de l'últim comptador de la ruta.	
Actor	Lector	
Precondició	L'usuari està validat en el sistema.	
Flux	Pas	Acció
	1	L'usuari prem una icona de la pantalla

	2	El sistema mostra mostra la informació de l'últim abonat de la ruta de lectura
Postcondició	El sistema visualitza la informació de l'últim abonat	
Excepcions	Pas	Acció
	2.1	Si és l'últim abonat de la ruta el sistema no avança
	2.2	Si el sistema no pot accedir a la informació
	3.2	El sistema avisa amb un missatge d'error

Cas d'ús 11	Llegir primer abonat	
Descripció	Es passa a la lectura del primer comptador de la ruta.	
Actor	Lector	
Precondició	L'usuari està validat en el sistema.	
Flux	Pas	Acció
	1	L'usuari prem una icona de la pantalla
	2	El sistema mostra mostra la informació del primer abonat de la ruta de lectura
Postcondició	El sistema visualitza la informació de l'últim abonat	
Excepcions	Pas	Acció
	2.1	Si és el primer abonat de la ruta el sistema no avança
	2.2	Si el sistema no pot accedir a la informació
	3.2	El sistema avisa amb un missatge d'error

Cas d'ús 12	Mostrar posició comptador	
Descripció	Es mostra la posició del comptador en un mapa.	
Actor	Lector	
Precondició	L'usuari està validat en el sistema i està situat en la informació d'un abonat.	
Flux	Pas	Acció
	1	L'usuari prem una pestanya per mostrar la ubicació del comptador
	2	El sistema mostra una icona en un mapa indicant la posició del comptador
Postcondició	El sistema visualitza la posició del comptador en un mapa	
Excepcions	Pas	Acció
	2.1	Si el sistema no pot representar la posició en el mapa
	3.1	El sistema avisa amb un missatge d'error
	2.2	Si l'abonat no té posició del comptador introduïda
	3.2	El sistema no mostra el mapa sinó un missatge indicant-ho

Cas d'ús 13	Introduir incidència	
Descripció	S'introdueix en el sistema la informació d'una incidència relacionada amb un abonat concret.	
Actor	Lector	
Precondició	L'usuari està validat en el sistema i està situat en la informació d'un abonat.	
Flux	Pas	Acció
	1	L'usuari introdueix informació relacionada amb el tipus d'incidència i les observacions
	2	El sistema emmagatzema la informació
Postcondició	El sistema emmagatzema la informació d'una incidència d'un abonat	
Excepcions		

Cas d'ús 14	Introduir foto incidència	
Descripció	S'introdueix una foto en una incidència d'un abonat.	
Actor	Lector	
Precondició	L'usuari està validat en el sistema, està situat en la informació d'un abonat i té una incidència oberta.	
Flux	Pas	Acció
	1	L'usuari prem una icona
	2	El sistema mostra una càmera
	3	L'usuari prem un botó per fer la foto
	4	El sistema emmagatzema la foto
Postcondició	El sistema emmagatzema la foto d'una incidència d'un abonat	
Excepcions	Pas	Acció
	2.1	Si el sistema no pot obrir la càmera
	3.1	El sistema informa amb un missatge de l'error
	4.2	Si el sistema no pot emmagatzemar la foto
	5.2	El sistema informa amb un missatge de l'error

Cas d'ús 15	Esborrar foto incidència	
Descripció	S'esborra una foto d'una incidència d'un abonat.	
Actor	Lector	
Precondició	L'usuari està validat en el sistema, està situat en la informació d'un abonat, té una incidència oberta i la incidència té una foto annexada.	
Flux	Pas	Acció

	1	L'usuari prem una icona
	2	El sistema borra la foto
Postcondició	El sistema emmagatzema la foto d'una incidència d'un abonat	
Excepcions	Pas	Acció
	2.1	Si el sistema no pot esborrar la foto
	3.1	El sistema informa amb un missatge de l'error

Cas d'ús 16	Compartir incidència	
Descripció	Es comparteix la informació d'una incidència d'un abonat en una aplicació externa.	
Actor	Lector, Abonat	
Precondició	L'usuari està validat en el sistema, està situat en la informació d'un abonat i té una incidència oberta.	
Flux	Pas	Acció
	1	L'usuari prem una icona
	2	El sistema obri la finestra de les aplicacions disponibles per compartir la informació de la incidència
	3	L'usuari tria amb quina aplicació vol compartir la informació
	4	El sistema obri eixa aplicació emplenant una informació base (text i foto si existeix)
	5	L'usuari emplena el destinatari (i informació addicional si vol) i prem l'opció d'enviar
	4	El sistema envia/comparteix la informació de la incidència a l'abonat
Postcondició	El sistema envia la informació de la incidència	
Excepcions	Pas	Acció
	2.1	Si el sistema no pot obrir la finestra de compartició d'aplicacions
	3.1	El sistema informa amb un missatge de l'error

Cas d'ús 17	Cercar abonat	
Descripció	Es cerca informació per tal d'obtenir d'un abonat determinat.	
Actor	Lector	
Precondició	L'usuari està validat en el sistema, està situat en la pantalla de cerca.	
Flux	Pas	Acció
	1	L'usuari emplena els criteris de cerca
	2	El sistema filtra i troba la informació dels abonats que compleixen els criteris de cerca

	3	El sistema mostra una llista dels abonats a partir dels resultats obtinguts en el pas 2
	4	L'usuari prem en un element de la llista
	5	El sistema visualitza en pantalla la informació de l'abonat seleccionat
Postcondició	El sistema mostra la informació d'un abonat	
Excepcions	Pas	Acció
	2.1	Si el sistema no pot cercar la informació
	3.1	El sistema informa amb un missatge de l'error

Cas d'ús 18	Introduir avaria canalització	
Descripció	S'introdueix en el sistema la informació d'una avaria de la canalització de les xarxes de reg.	
Actor	Lector	
Precondició	L'usuari està validat en el sistema.	
Flux	Pas	Acció
	1	L'usuari tria l'opció de l'aplicació per introduir avaria
	2	El sistema mostra una pantalla d'introducció d'informació
	3	L'usuari introdueix la informació del tipus d'avaria, coordenades GPS automàtiques (mitjançant un botó) i unes observacions
	4	El sistema mostra la informació de l'avaria entrada
Postcondició	El sistema emmagatzema la informació de l'avaria	
Excepcions		

Cas d'ús 19	Introduir foto avaria canalització	
Descripció	S'introdueix una foto en una avaria de la canalització.	
Actor	Lector	
Precondició	L'usuari està validat en el sistema, està situat en la informació d'una avaria.	
Flux	Pas	Acció
	1	L'usuari prem una icona
	2	El sistema mostra una càmera
	3	L'usuari prem un botó per fer la foto
	4	El sistema emmagatzema la foto
Postcondició	El sistema emmagatzema la foto d'una avaria de la canalització	
Excepcions	Pas	Acció
	2.1	Si el sistema no pot obrir la càmera
	3.1	El sistema informa amb un missatge de l'error

	4.2	Si el sistema no pot emmagatzemar la foto
	5.2	El sistema informa amb un missatge de l'error

Cas d'ús 20	Esborrar foto avaria canalització	
Descripció	S'esborra una foto d'una avaria de la canalització.	
Actor	Lector	
Precondició	L'usuari està validat en el sistema, està situat en la informació d'una avaria de la canalització i l'avaria té una foto annexada.	
Flux	Pas	Acció
	1	L'usuari prem una icona
	2	El sistema esborra la foto
Postcondició	El sistema esborra la foto de l'avaria	
Excepcions	Pas	Acció
	2.1	Si el sistema no pot esborrar la foto
	3.1	El sistema informa amb un missatge de l'error

Cas d'ús 21	Comunicar avaria	
Descripció	S'envia a una adreça de correu electrònic la informació d'una avaria.	
Actor	Lector, Servidor de correu	
Precondició	L'usuari està validat en el sistema i està situat en la informació d'una avaria.	
Flux	Pas	Acció
	1	L'usuari prem una icona
	2	El sistema obri la finestra de les aplicacions de correu
	3	L'usuari tria l'aplicació
	4	El sistema emplena la informació automàticament i l'envia al destinatari que té configurat que la reba
Postcondició	El sistema envia la informació de l'avaria	
Excepcions	Pas	Acció
	2.1	Si el sistema no pot obrir la finestra de compartició d'aplicacions
	3.1	El sistema informa amb un missatge de l'error

Cas d'ús 22	Eixir del sistema
Descripció	Es desvincula la informació de l'usuari per eixir-se'n de la sessió.
Actor	Lector
Precondició	L'usuari està validat en el sistema.

Flux	Pas	Acció
	1	L'usuari selecciona una opció de menú
	2	El sistema esborra aquella informació que conté de la sessió actual de l'usuari
Postcondició	El sistema mostra la pantalla de validació i no hi ha cap vinculació de sessió	
Excepcions		

2.4.3 Diagrama de l'arquitectura del sistema

El sistema estarà format per una aplicació mòbil i per una part servidora que farà de nexe d'unió entre l'aplicació mòbil i el programari que tindrà la comunitat de regants per gestionar els seus abonats. A continuació es mostra un diagrama de com queda la mateixa:

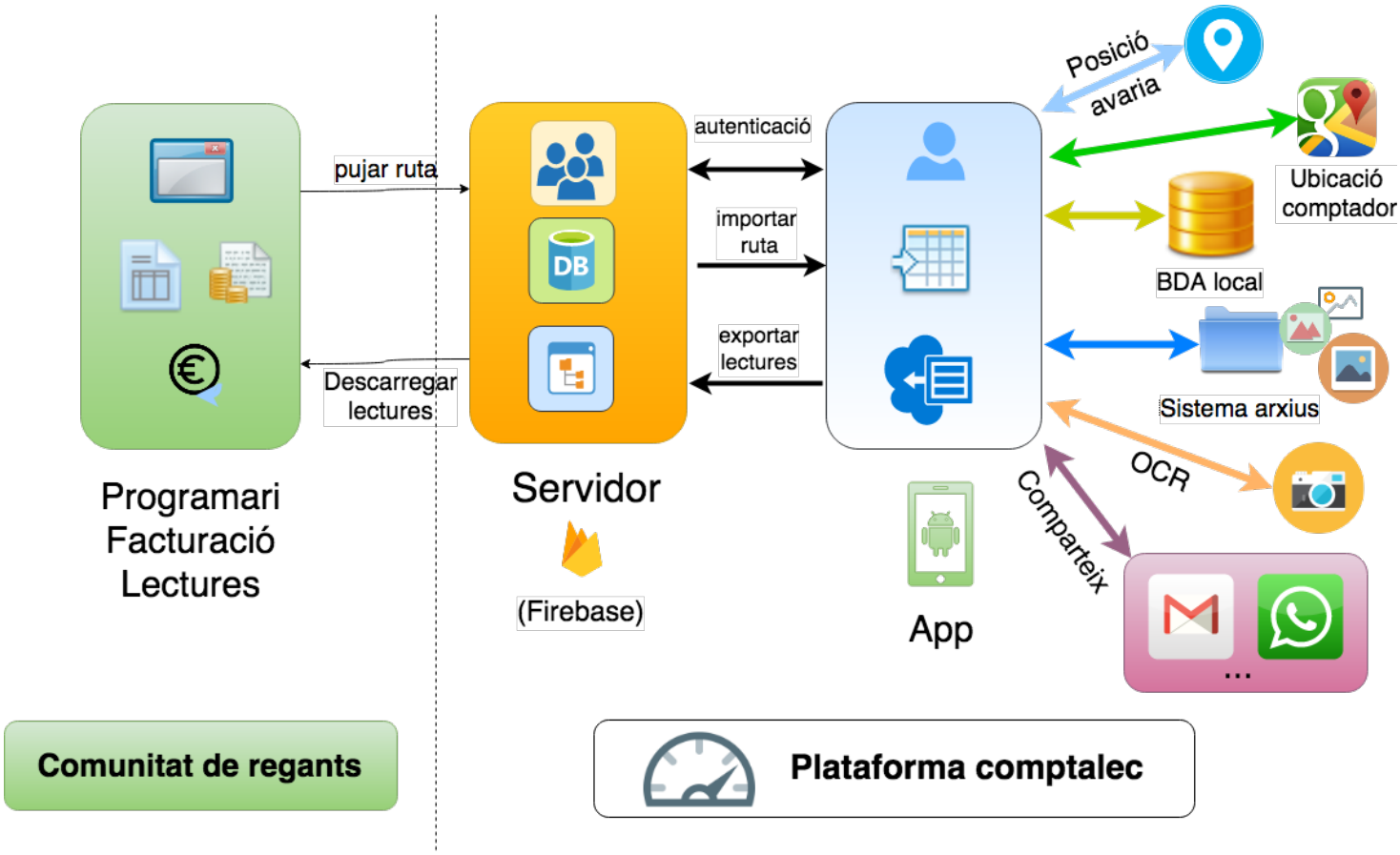


Figura 30: Arquitectura del sistema

En principi es podria considerar que la part servidora estigués integrada en la part de les comunitats de regants però no és possible donat a què una font d'ingressos de l'aplicació és el nombre de lectures que realitzen les comunitats de regants. Aleshores mitjançant el nostre servidor es podrà controlar aquest

aspecte per a després poder realitzar la nostra facturació interna a les comunitats de regants.

La façana de la part servidora que vincula amb les comunitats de regants s'usa per a què aquestes pugen les rutes de comptadors a llegir i després descarregar-se els fitxers de les lectures.

L'altra part de servidor que connecta amb l'aplicació mòbil serveix per a gestionar els usuaris i l'autenticació de l'aplicació mòbil, descarregar-se les rutes de lectura al terminal i rebre del terminal les lectures una vegada processades.

Pel que fa a l'aplicació mòbil, una vegada descarregada la ruta del servidor s'emmagatzema en una base de dades local. S'ha decidit fer així per temes de velocitat (és més ràpid accedir a la informació local que no al núvol) i perquè de vegades es pot perdre la cobertura en zones d'ombria.

Com es pot observar també a l'arquitectura, l'aplicació usa altres sistemes per poder dur a terme diverses tasques:

- **Sistema d'arxius**: es on s'emmagatzemaran les fotos preses en incidències d'abonats.
- **Base de dades local**: es on s'emmagatzemen les dades de les lectures.
- **GoogleMaps**: es posicionarà en aquest la ubicació de comptadors a llegir.
- **Càmera de fotos**: s'usa per prendre fotos d'incidències, avaries de la canalització i fer les lectures de comptadors amb OCR.
- **GPS**: s'utilitza per poder determinar les coordenades exactes d'una avaria.
- **Altres aplicacions (correu electrònic, whatsapp, etc.)**: A l'hora d'informar d'avaries o incidències l'aplicació pot compartir informació amb altres aplicacions instal·lades al telèfon.

Per allotjar la part servidora es podria fer de diverses maneres:

1) **Servidor físic allotjat en un CPD compartit** amb altres usuaris: es tracta d'un servidor que lloga una empresa d'allotjament. Es compartiria la màquina física amb altres usuaris.
2) **Servidor física allotjat en un CPD dedicat**: és un cas semblant a l'anterior però amb una màquina dedicada per a les nostres necessitats. Tindria un cost més elevat a la solució anterior.
3) **Servidor virtual dedicat (VPS)**: és una solució com l'anterior però el servidor en lloc de ser físic és un servidor virtualitzat. Les prestacions són semblants a l'anterior però el cost s'abarateix perquè el proveïdor en una màquina posa diversos servidors virtuals.
4) **Allotjar-ho en una plataforma al núvol**: en aquest cas el proveïdor ens dona una sèrie de ferramentes per poder dur a terme les nostres necessitats. Ací tenim solucions diverses com AWS (Amazon Web Services) o Firebase de Google entre altres.

Finalment s'ha decidit utilitzar Firebase per diverses raons:

a) la facilitat d'integració que es té amb aplicacions Android
b) diversos sistemes integrats en una única plataforma:
    b.1 Sistema d'autenticació d'usuaris
    b.2 Base de dades NoSQL (Realtime) per poder deixar les rutes de lectura. Vinculada l'autenticació d'usuaris amb la del punt b.1.

b.3 Sistema d'arxius (Cloud Storage) on poder deixar les lectures realitzades. Vinculada l'autenticació d'usuaris amb la del punt b.1.
b.4 Cost gratuït per a espais de disc, ample de banda i usuaris concurrents més que acceptable per a la nostra aplicació en la situació inicial (si després hi haguera més necessitats tècniques s'ofereix la possibilitat de créixer mitjançant un sistema de pagament sense canviar la plataforma)

Pel que fa a l'arquitectura de programari s'usarà el patró anomenat MVP (model-vista-presentador). Aquest patró és una derivació del patró MVC (model-vista-controlador).

El que pretenen ambdós patrons es separar les dades, la lògica de negoci i la representació de la informació. Per això tenen els tres components: model, vista i el controlador o presentador.

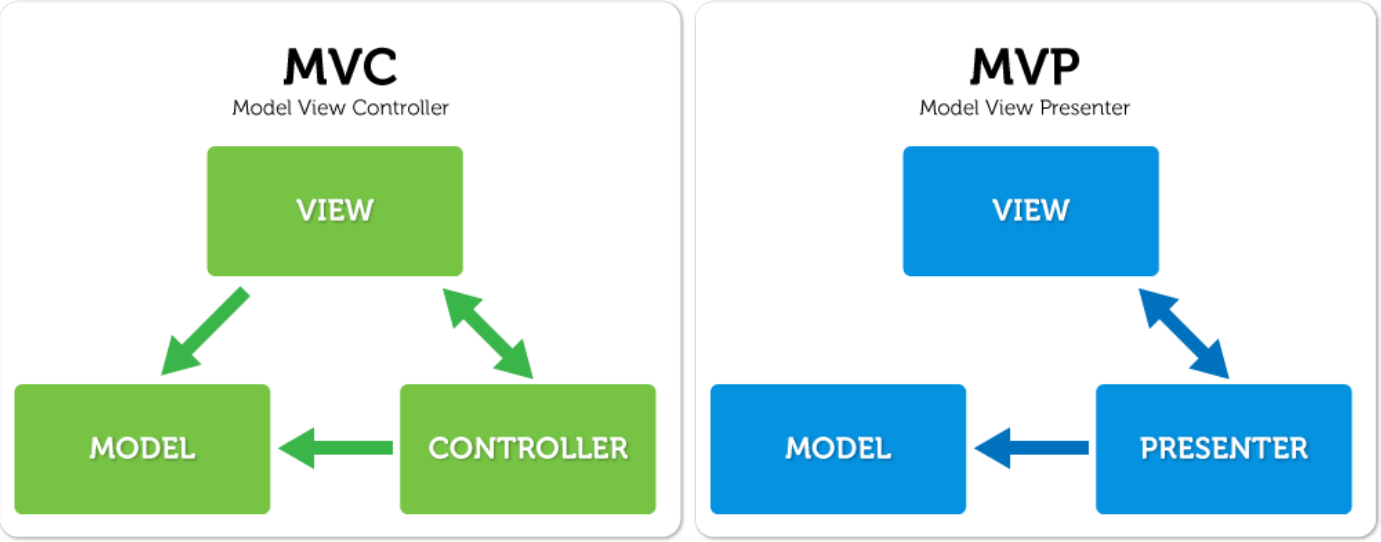


Figura 31: MVC vs MVP

La principal diferència entre els patrons resideix en què en el model MVC la vista pot fer ús del component model, mentre que en el model MVP no i només es pot comunicar amb el component presentador. Aleshores la vista és més lleugera i té menys codi. A més, també és una separació més clara de components la qual cosa ens durà a un codi més escalable, menys acoblat i més fàcil de mantenir.

Altres avantatges per els qual s'ha optat pel patró MVP (algunes també les conté el model MVC) són:

- Escriure codi més net i entenedor: a l'estar cada capa ben separada el codi és més fàcil de seguir, mantenir i provoca menys efectes laterals.
- Fer test unitaris més senzills: donat que la capa de presentació té la lògica, la capa de la vista serà molt simple i fàcil de fer les proves.
- Independència del model: si volem canviar l'accés per a què el servidor remot en compte de ser un determinat siga un altre o usar una altra base de dades local per emmagatzemarà la informació, només

implementant noves classes que respecten les interfícies de comunicació amb la capa de presentació ho tindrem resolt.

Pel que fa a l'elecció de la base de dades local a usar per emmagatzemar les rutes de lectura, tenim diverses possibilitats:

- SQLite: és una base de dades relacional lleugera que vé integrada en les últimes versions d'Android. És de codi obert. Implementa la major part de l'estàndar SQL92.
- Realm: és una base de dades d'objectes pensada per a aplicacions mòbils. És molt ràpida.
- CouchDB Lite: és una base de dades NoSQL que usa json.
- Berkey DB: és open source i té un alt rendiment. Pot treballar com una base de dades relacional o com una base de dades NoSQL.

Segons l'estudi [26] es pot observar en la gràfica de sota que la base de dades més usada és SQLite. Aleshores també s'emprarà aquesta en el desenvolupament del projecte per ser la més usada, lleugera i estar integrada en Android.

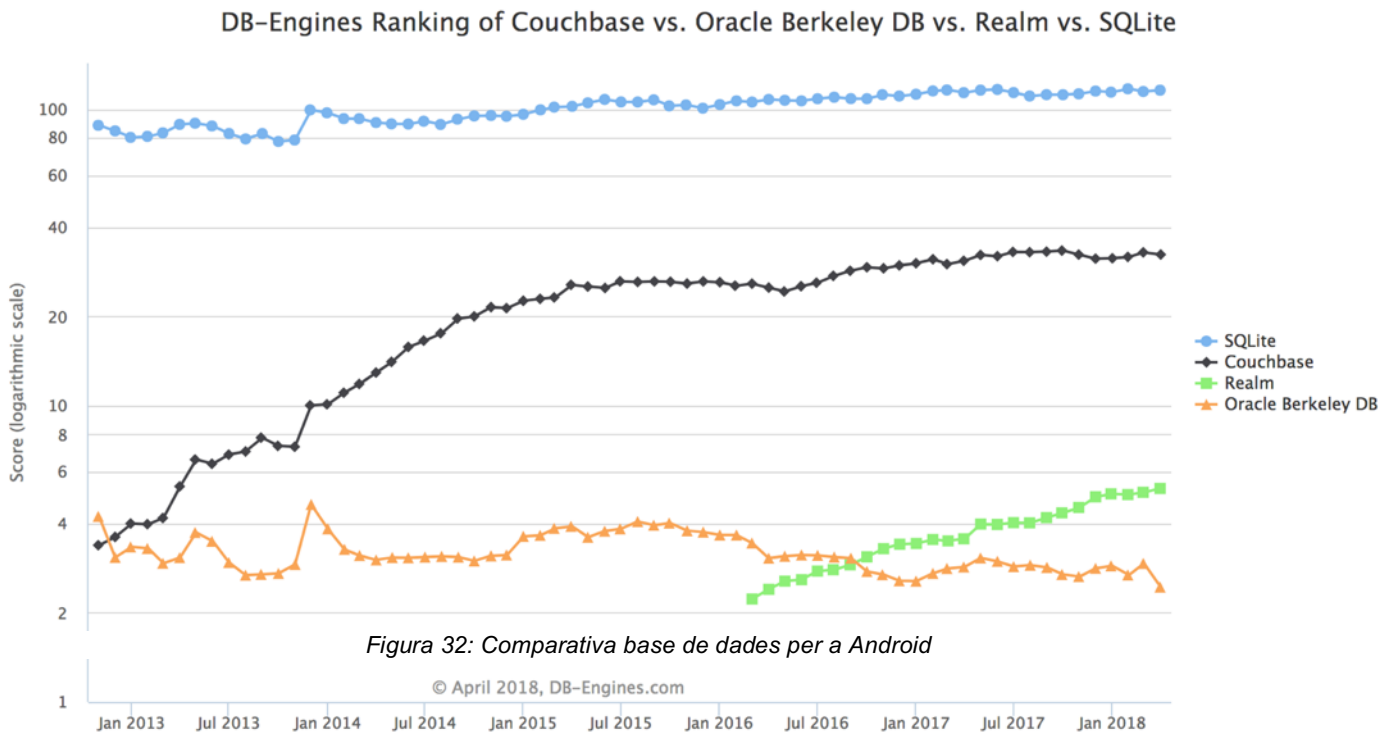


Figura 32: Comparativa base de dades per a Android

.

2.4.4 Diagrama UML d'entitats i classes

En aquest apartat es mostren les principals classes i entitats que contindrà el programari.

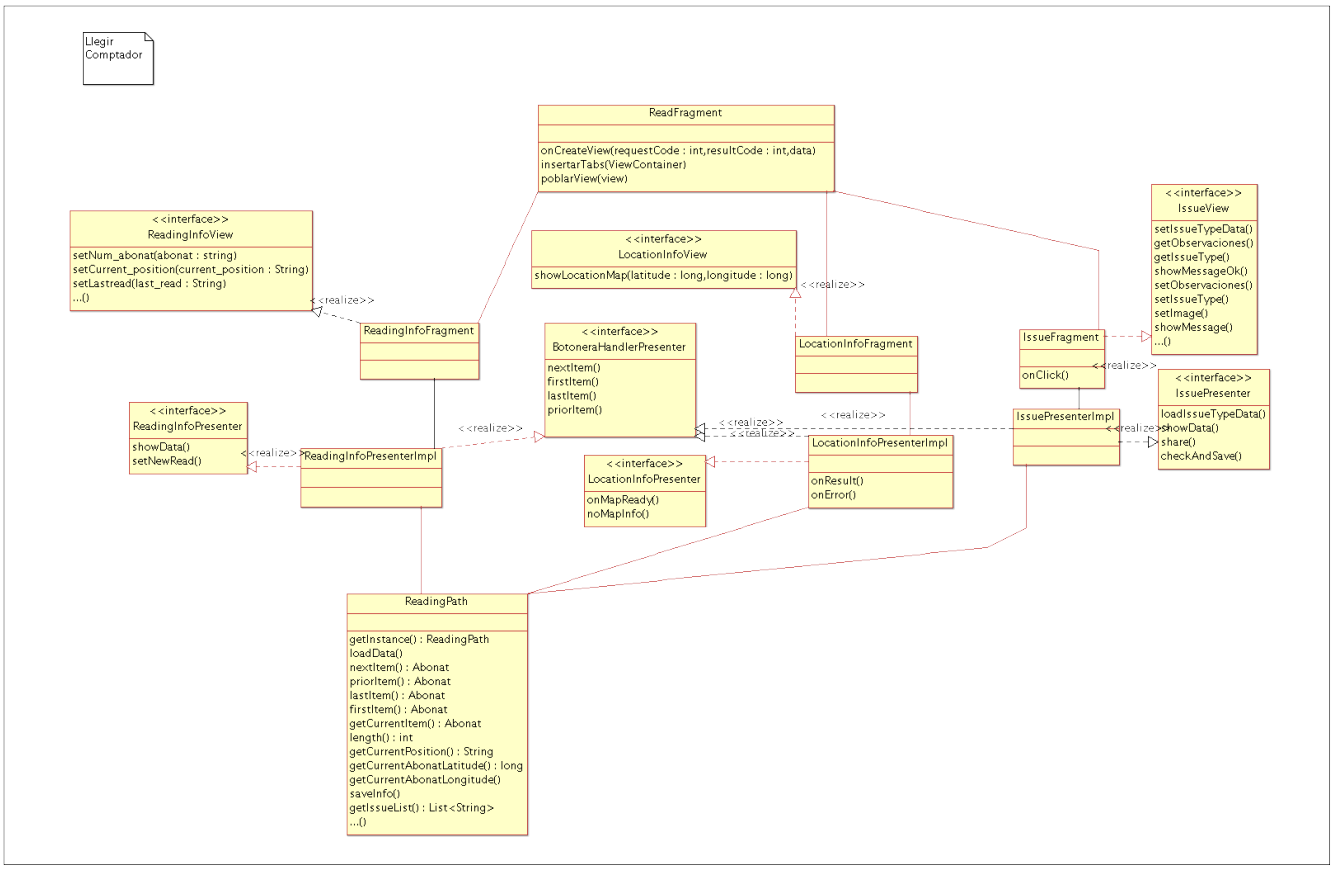


Figura 33: Diagrama classes UML llegir comptador

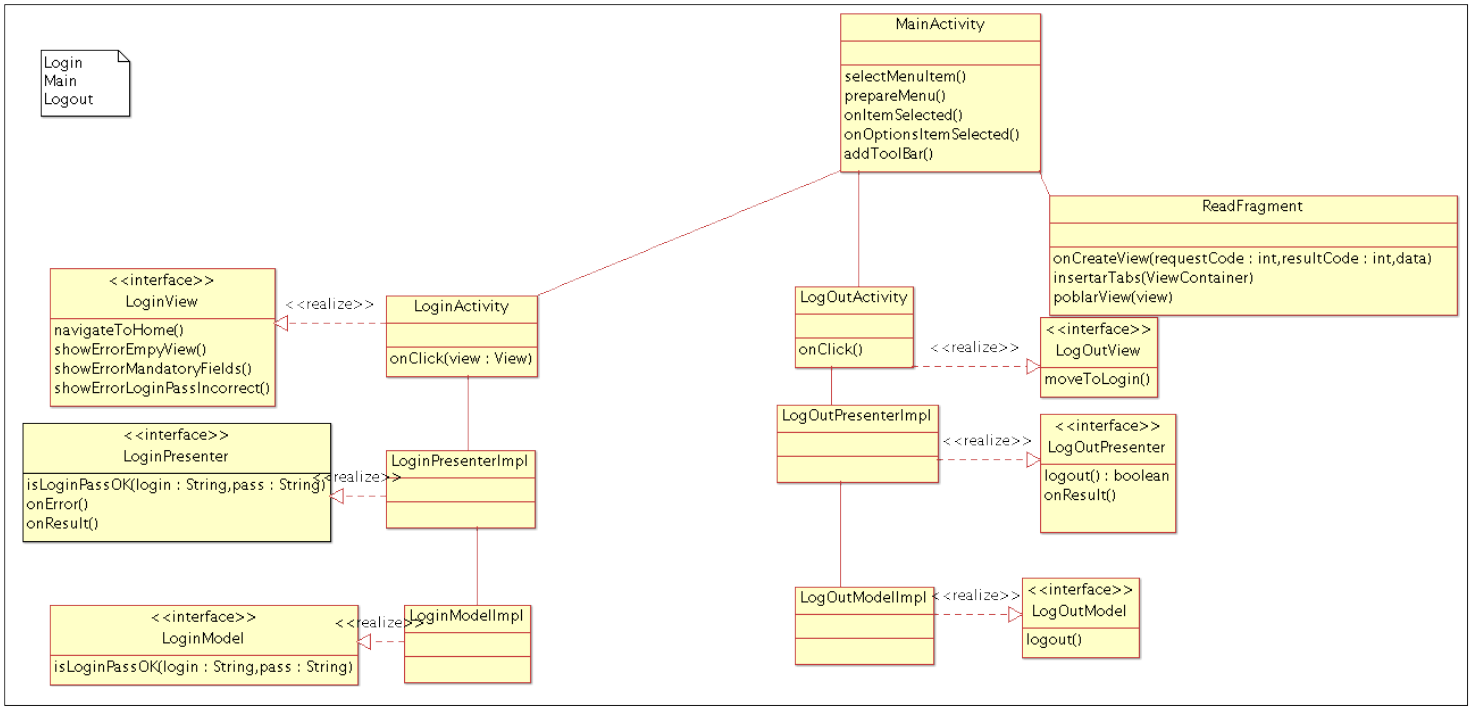


Figura 34: Diagrama classes UML login, logout, main

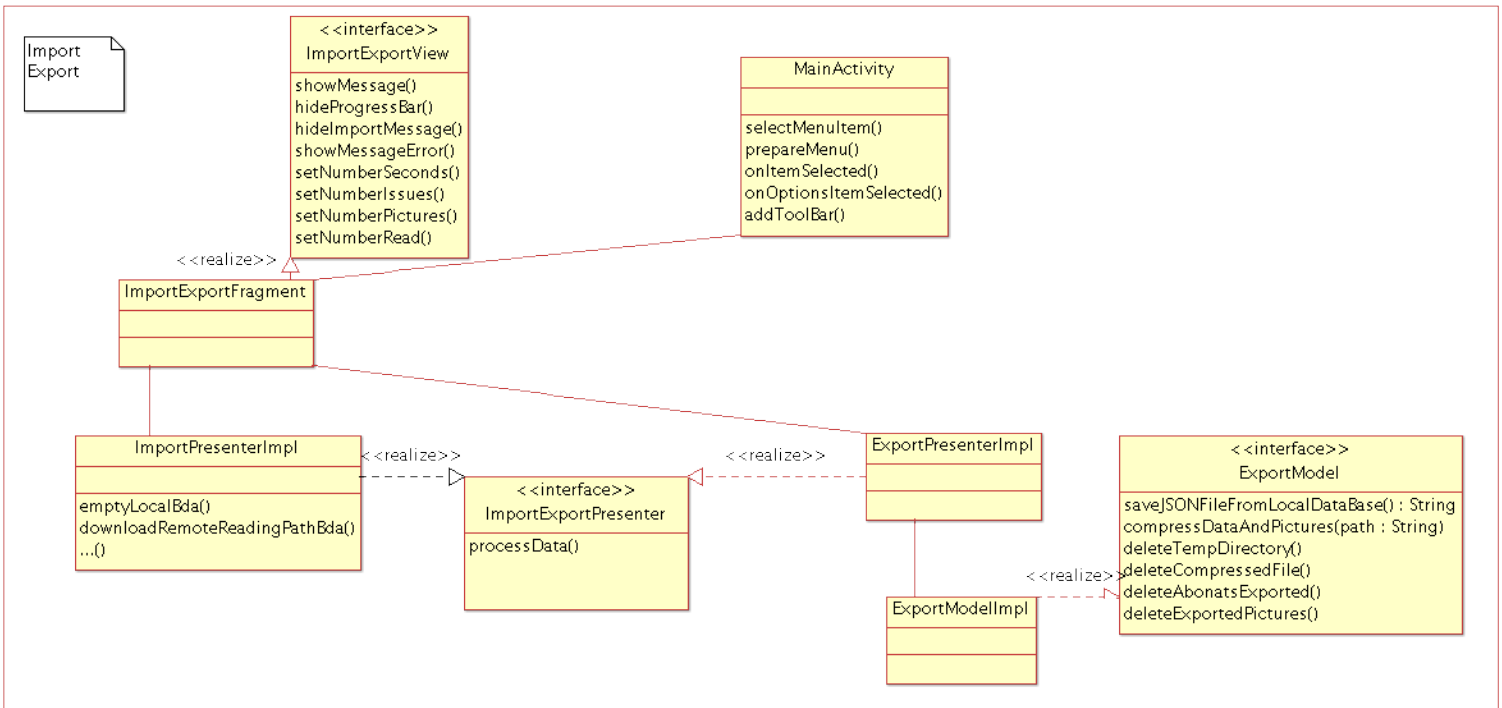


Figura 35: Model classes UML importar, exportar

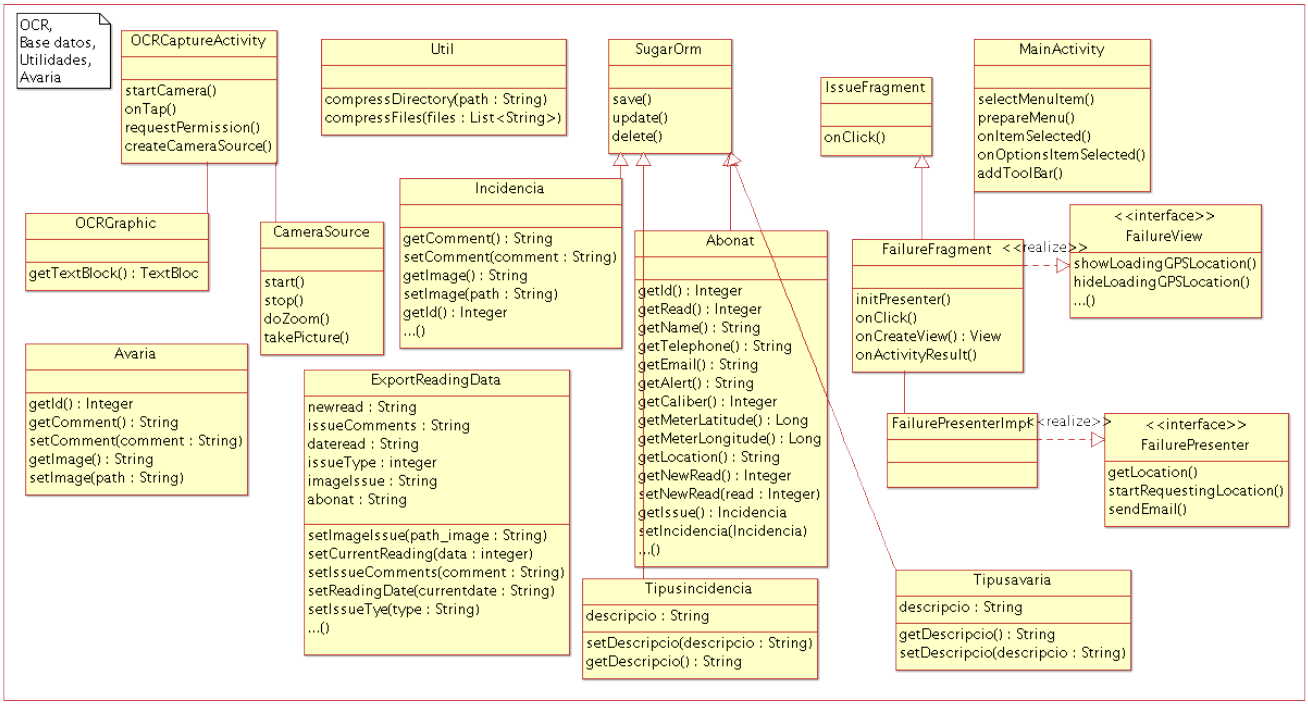


Figura 36: Diagrama classes UML OCR, base de dades, avaria i utilitats

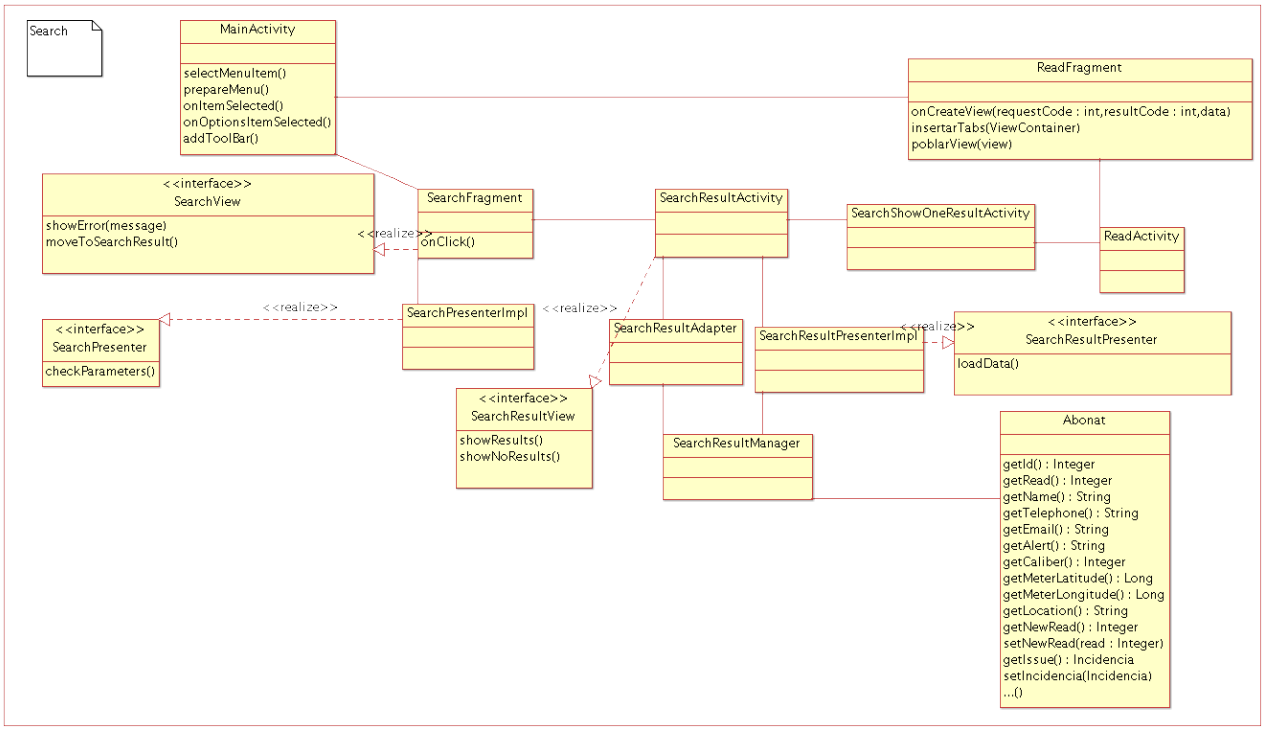


Figura 37: Diagrama classes UML cercar abonat

Dels diagrames anteriors cal fer menció que en les principals pantalles es creen interfícies per a cada vista, presentador i model i després hi ha una classe que els implementa per així queda ben clar el model MVP.

Així mateix també cal indicar que per a l'accés a la base de dades local es farà mitjançant un ORM (Object-relational mapping) perquè ens facilita la tasca de convertir elements de la base de dades en objectes i així poder interactuar amb aquesta de manera ràpida i senzilla.

Per a dur a terme aquesta última tasca existeixen diverses alternatives com ara ActiveAndroid, ORMLite, greenDAO i Sugar ORM.

De totes aquestes s'empra Sugar ORM per ser la més usada i tindre unes característiques que en faciten la interacció amb la base de dades:

- elimina escriure sentències SQL per accedir a la base de dades SQLite
- s'encarrega de crear les taules automàticament
- maneja la relació entre objectes
- ens dona un API net i simple per a les operacions amb la base de dades

2.4.5 Diagrama UML de la base de dades

En la part servidora s'emmagatzemen les rutes de lectura en una base de dades NoSQL per la seua senzillesa a l'hora d'exportar/importar fitxers en format json.

El que es pretén es tindre una estructura a la part servidor el més plana possible per a què siga fàcil d'importar per l'aplicació i aquesta funcione de manera àgil. Aleshores en el programari de les comunitat de regants existiran múltiples relacions per tal de vincular la informació i poder explotar-la de manera coherent, però a l'hora de pujar-la al nostre servidor serà una estructura sense relacions i amb la informació transformada en camps per a què es mostren directament en l'aplicació sense haver d'emprar vinculacions entre taules.

Aquesta possibilitat també es justifica perquè en la nostra aplicació seran quasi tots els camps de consulta, aleshores no cal emmagatzemar informació en taules addicionals. Així mateix, com tampoc s'introdueix informació que es vincularà en altres taules tampoc es fa un ús addicional d'espai que es una de les causes per les quals es creen noves taules i relacions entre elles.

Tot i que és una base de dades NoSQL ací està el model UML corresponent:

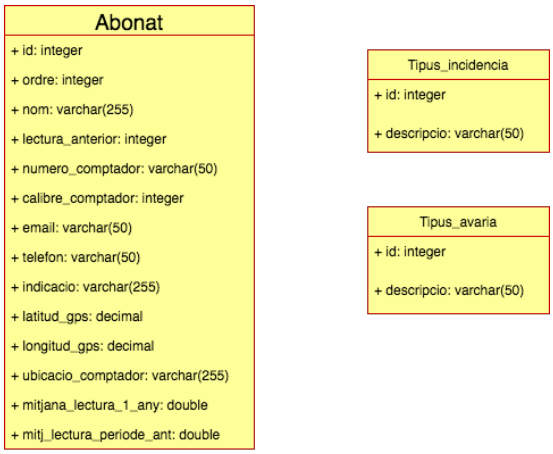


Figura 38: Model de dades rutes lectura servidor

Després d'importar la informació del servidor s'emmagatzema en una estructura semblant en una base de dades relacional local per a què l'aplicació s'execute més ràpida i no siga necessari tindre cobertura. En aquest cas sí que existeix una relació que fa referència a la vinculació entre incidència i abonat. Per la qual cosa si no s'introdueix cap incidència d'un abonat no reservarem espai en la taula abonat per als camps que contenen aquesta informació.

El model UML queda com segueix:

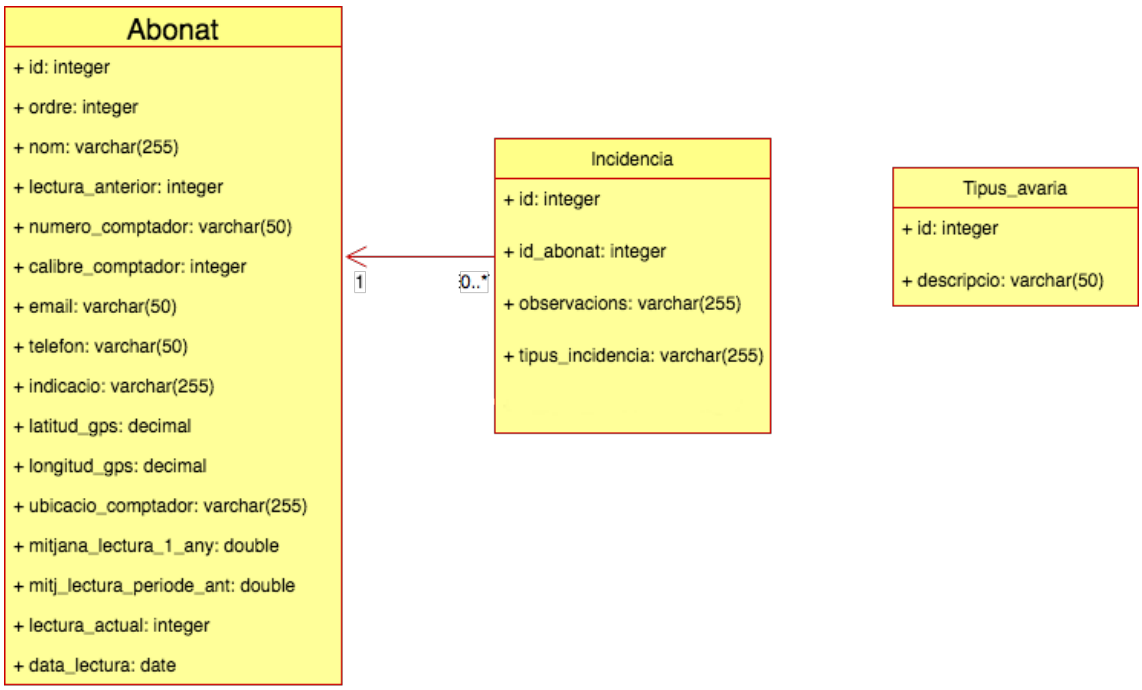


Figura 39: Model relacional de base de dades local

3. Implementació

En aquest apartat s'aborda el detall de com s'ha desenvolupat finalment l'aplicació. Entre altres, es detalla com ha quedat la interfície d'usuari, les proves realitzades i la descripció tècnica de les decisions preses per tal d'aconseguir l'aplicació final.

3.1 Estructura de paquets i funcions de cadascun

A l'hora d'estructurar els fitxers i les dades dins del projecte s'ha seguit una jerarquia determinada per tal de tindre un ordre i filosofia de treball. L'estructura ha quedat de la següent manera:

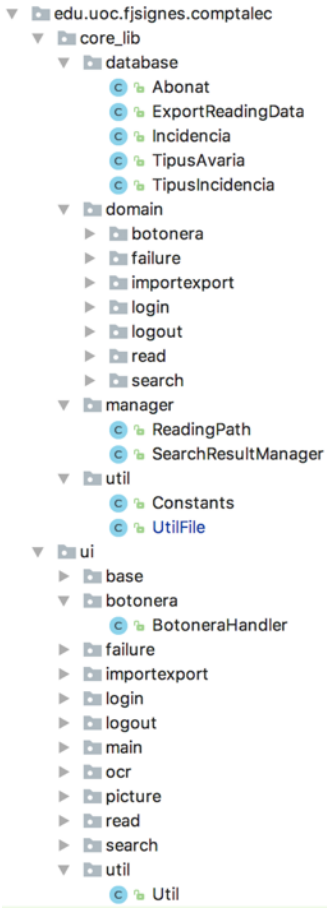


Figura 40: Estructura de paquets del projecte

Es pot observar que en un primer nivell existeixen dos grans paquets:

- **core_lib**: dins d'aquest paquet van totes les classes que en el model vista presentador fan referència a la capa del model i la del presentador. És a dir, totes aquelles classes que teòricament es podrien empaquetar en un jar i fer ús en una altra aplicació implementant només la capa de vista per a la nova aplicació.
- **ui**: ací s'inclouen totes les classes que es relacionen amb la interfície d'usuari (Activities i Fragments).

Acte seguit s'explica en un major nivell de detall cadascun dels subdirectoris (subpaquets) que existeixen en la jerarquia dels dos paquets principals.

1) **core_lib/database**: conté les classes de persistència per a emmagatzemar les dades en una base de dades local. Les principals classes són:
    - Abonat: conté la informació d'un abonat (lectura anterior, número abonat, nom, telèfon, lectura actual, incidències, etc.)
    - Incidència: fa referència a les incidències que es poden enregistrar d'un abonat mentre es fa la lectura del comptador
    - Tipus incidència i Tipus avaria: contenen els diversos tipus que es poden triar per a categoritzar una avaria o una incidència.
    - ExportReadingData: aquesta classe s'usa per exportar les dades de cadascuna de les lectures realitzades als abonats. Com que conté menys informació que la classe Abonat, s'ha creat aquesta nova classe per tal d'optimitzar la pujada d'informació al servidor.

2) **core_lib/domain**: conté les interfícies i les classes del domini. És a dir, les interfícies de les vistes, el model i el presentador. A més a més conté la implementació de les interfícies model i presentador en classes.
S'ha seguit una nomenclatura normalitzada per fer referència a aquestes: sempre es posa el prefix de la pantalla a què fa referència seguit de "View" si es tracta de la interfície de la vista d'aquesta pantalla, "Presenter" si es tracta de la interfície de la capa de presentació d'aquesta pantalla, "PresenterImpl" si es tracta de la implementació de la interfície de presentació, "Model" si es tracta de la interfície de la capa de model i "ModelImpl" si es tracta de la classe d'implementació del model.
Exemple:
core_lib/domain/LoginView: es tracta de la interfície de la vista de la pantalla de Login
core_lib/domain/LoginPresenter: es tracta de la interfície de la capa de presentació de la pantalla de Login
core_lib/domain/LoginPresenterImpl: es tracta de la classe que implementa la interfície LoginPresenter
core_lib/domain/LoginModel: es tracta de la interfície de la capa de model de la pantalla de Login
core_lib/domain/LoginModelImpl: es tracta de la classe que implementa la interfície LoginModel

Els subpaquets que conté aquest paquet són:

a) **core_lib/domain/botonera**: conté la interfície de les funcions que ha de desenvolupar una botonera en l'aplicació. La botonera és un element que ens permetrà desplaçar-nos per tots els abonats que conté una ruta i poder anotar les lectures dels mateixos. Es tracta d'un conjunt de quatre botons que permetrà avançar o anar cap enrere en la ruta.
b) **core_lib/domain/failure:** en aquest lloc estan totes les interfícies i classes que tracten la pantalla d'introduir una nova avaria.
c) **core_lib/domain/importexport:** aquest subpaquet conté totes les interfícies i classes que fan referència a les pantalles d'importació i exportació de lectures.

d) **core_lib/domain/login:** aquest directori conté les interfícies i classes que s'usen en la pantalla d'entrada al sistema.
e) **core_lib/domain/logout:** aquest subpaquet allotja les interfícies i classes que s'utilitzen per a poder sortir de l'aplicació.
f) **core_lib/domain/read:** allotja les interfícies i classes que s'utilitzen per a poder realitzar les lectures, saber la ubicació del comptador en un mapa i poder enregistrar una incidència relacionada amb l'abonat que s'està llegint actualment.
g) **core_lib/domain/search:** conté les interfícies i classes que s'empren per a cercar informació dels abonats que compleixen uns criteris.

3) **core_lib/manager:** aquest paquet conté dos classes:
   a) **ReadingPath:** Es tracta d'una classe Singleton que carrega els elements d'una ruta de lectura en memòria. Així a l'hora de poder avançar o retrocedir al pròxim o anterior abonat de la ruta, en tot moment ho tenim en aquesta espècie de memòria cau que accelerà l'experiència de l'usuari i farà que l'aplicació siga més ràpida.
   En aquest punt cal matisar que l'estructura interna és un array d'abonats i després hi ha una sèrie de mètodes que permeten interactuar amb aquesta estructura. Com que les rutes de lectura tenen més o menys uns 50 abonats (que es el que pot diàriament com a màxim llegir un lector) no és una quantitat massa elevada i no consumeix molta memòria del dispositiu mòbil. En cas de què aquesta ruta contenira més abonats o hi haguera problemes de memòria en els dispositius mòbils, el que s'hauria de fer com a punt de millora seria emmagatzemar només un array dels IDs del abonats de la base de dades (en compte d'un array amb la informació de classes Abonat) i en treure la informació d'un abonat concret accedir a la base de dades i obtindre la informació d'aquest abonat. Aleshores un array de números ocupa menys que un array de classes Abonat.
   Com sempre en programació es pren una decisió de compromís entre la velocitat i l'espai que s'ocupa en memòria. En aquest cas, inicialment, s'ha pres la determinació de donar-li prioritat a la velocitat sacrificant espai en memòria.
   b) **SearchResultManager:** Aquesta classe contindrà tots els abonats resultants d'una cerca i permetrà poder accedir a cadascun d'ells.

4) **core_lib/util:** Aquest paquet conté les classes d'utilitats que poden ser usades al llarg de tota l'aplicació. En concret conté les classes:
   a) **Constants:** conté totes les constants que té l'aplicació. S'ha centralitzar en un lloc concret per tal de poder de manera senzilla i ràpida canviar paràmetres del tipus: màxima resolució de les imatges grans capturades de la càmera per ser emmagatzemades en l'aplicació, URL a tindre present quan s'envien les coordenades GPS d'una avaria per email, etc.
   A més també conté elements com els noms dels arguments que es passen com a paràmetres entre les activitats i els fragments, o els identificadors de peticions d'esdeveniments quan es llances activitats i esperem els seus resultats.
   b) **UtilFile**: Aquesta classe conté utilitats relacionades amb l'ús de fitxers (còpia de fitxers, comprimir directoris o fitxers, etc.)

5) **ui/base**: conté una interfície per a implementar alguns dels missatges que pot mostrar una vista.
6) **ui/botonera**: conté la implementació de la botonera per a poder navegar per una ruta.
7) **ui/failure**: allotja la implementació de la vista de la pantalla d'avaries.
8) **ui/importexport**: ací estan les implementacions de les vistes de les pantalles d'importació i exportació. Cal matisar que en els paquets ui la nomenclatura seguida es el nom de la pantalla i després Activity si es tracta d'una activity o Fragment si es tracta d'un Fragment.
9) **ui/login i ui/logout**: contenen les pantalles que implementen les vistes d'inici i fi de sessió en l'aplicació.
10) **ui/main:** conté la pantalla principal de l'aplicació. Com que es depenent de la plataforma ací també s'ha ubicat el Presenter i View d'aquesta classe.
11) **ui/ocr:** aquest subpaquet conté totes les classes i interfícies relacionades amb la detecció automàtica de text a partir d'una imatge.
12) **ui/picture:** conté una pantalla que fa gran una imatge d'una avaria o una incidència.
13) **ui/read:** allotja les activitats i fragments que implementen les vistes per a prendre lectures, mostrar ubicació de comptadors en un mapa i l'anotació d'una avaria d'abonat.
14) **ui/search:** conté les pantalles que implementen la cerca d'abonats.
15) **ui/util:** allotja la classe Util que conté una sèrie d'utilitats que depenen de la plataforma mòbil. Entre altres permet saber si hi ha connexió de dades en el dispositiu, conté també una funció per amagar el teclat virtual, aconseguir una URI a partir d'un Bitmap, etc.

Un altre punt important que s'ha tingut present en l'aplicació és tindre totes les cadenes de text dins del fitxer strings.xml així podem afavorir la traducció de l'aplicació. Dins d'aquest fitxer s'ha seguit una nomenclatura indicant la pantalla on està la cadena per tal de poder ubicar de manera ràpida des d'on es fa ús d'aquesta cadena.

3.2 Interfície d'usuari

A continuació detallem com ha quedat finalment la implementació dels prototips definits en la fase anterior.

3.2.1 Login

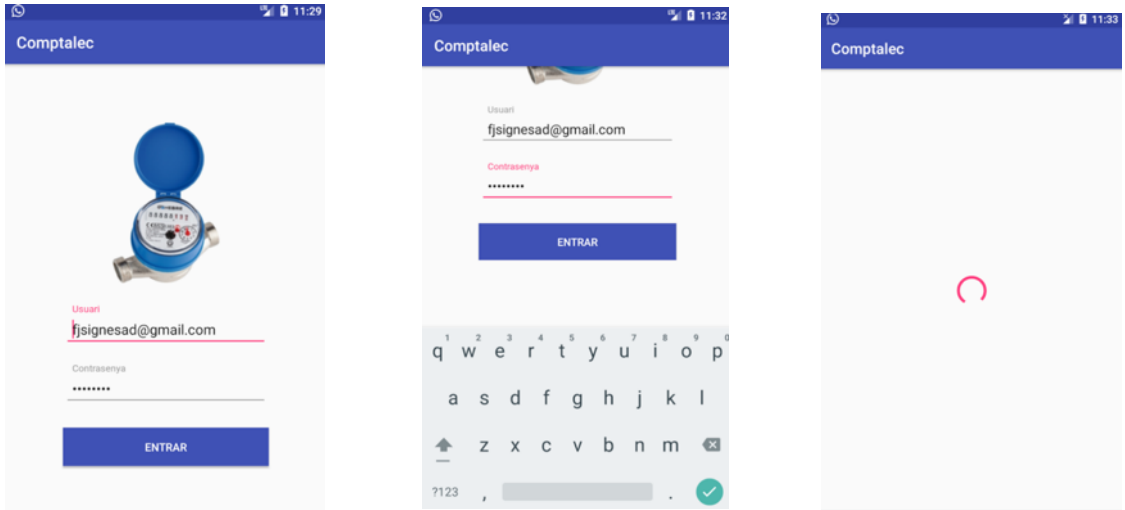
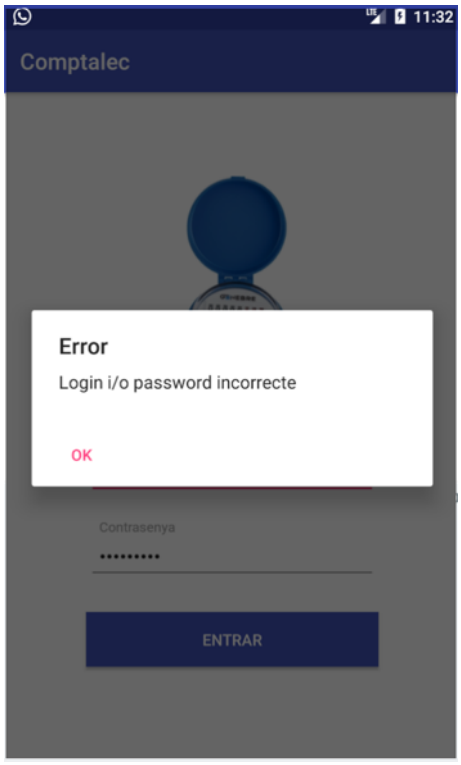
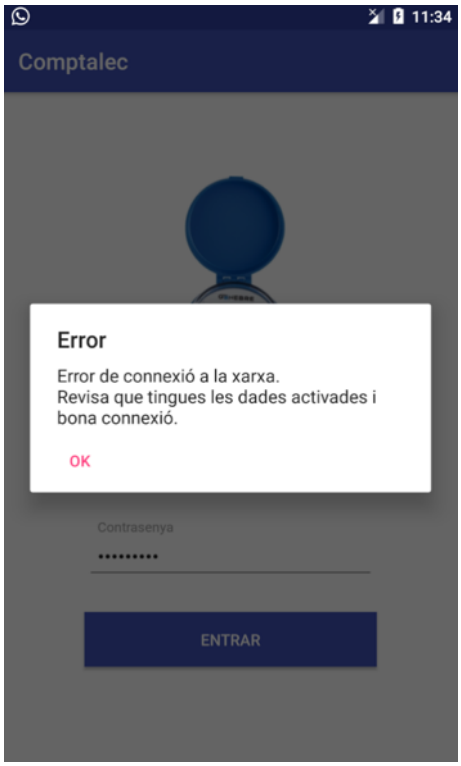
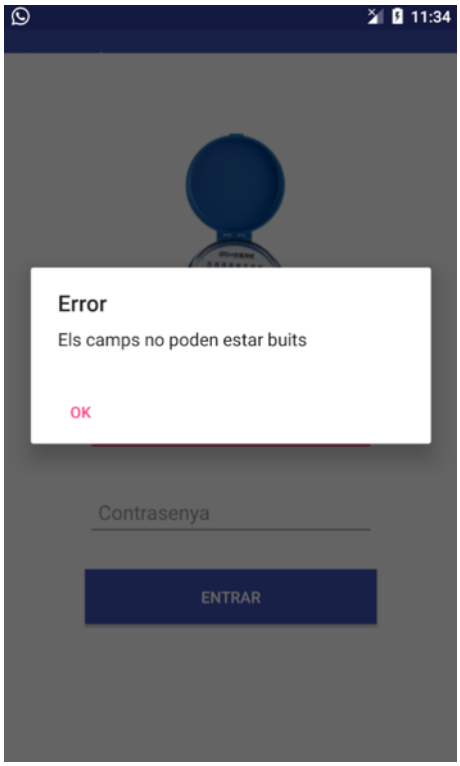


Figura 41: Implementació login







3.2.1.1 Detalls d'implementació

S'ha usat una Activity amb els camps d'usuari, contrasenya i el botó per a poder entrar. L'aplicació emmagatzema la informació de l'últim usuari que s'ha logat al sistema per a què aquest tinga una experiència més satisfactòria. Així només ha de tornar a introduir la contrasenya si se n'ha eixit del sistema. Aquesta part s'ha abordat mitjançant l'ús de SharedPreferences que permeten emmagatzemar informació en un fitxer privat xml situat en el directori /data/data/edu.uoc.fjsignes.comptalec/shared_prefs.

Fer menció que la validació es realitza utilitzant l'API d'autenticació de Firebase. A més també indicar que quan s'escriu dins del camp de la contrasenya s'ha tingut la cura de pujar la pantalla cap amunt per tal de què el teclat virtual no amague aquests camps. Així mateix s'ha fet que quan es prema el botó entrar s'amague automàticament el teclat virtual. També cal indicar que l'aplicació mostra una barra de progrés mentre es realitza la consulta al servidor remot i que informa mitjançant missatges si ha hagut algun tipus d'error o l'usuari/contrasenya no són vàlids.

3.2.1.2 Proves

Definició de la prova	Comportament esperat	Resultat
Introduir usuari i contrasenya correctes	Mostrar barra de progrés mentre es consulta al servidor i entrar en la pantalla principal de l'aplicació	Correcte
Introduir usuari i /o contrasenya incorrectes	Mostrar barra de progrés mentre es consulta al servidor i després mostrar missatge d'error indicant: "Usuari i/o contrasenya incorrecte"	Correcte
Deixar camp usuari i /o contrasenya buit	Mostrar missatge d'error indicant: "Els camps no poden estar buits"	Correcte

Sense connexió amb el servidor	Mostrar missatge d'error indicant: "Error de connexió a la xarxa. Revisa que tingues les dades activades i bona connexió"	Correcte

3.2.2 Menú i pantalla principal

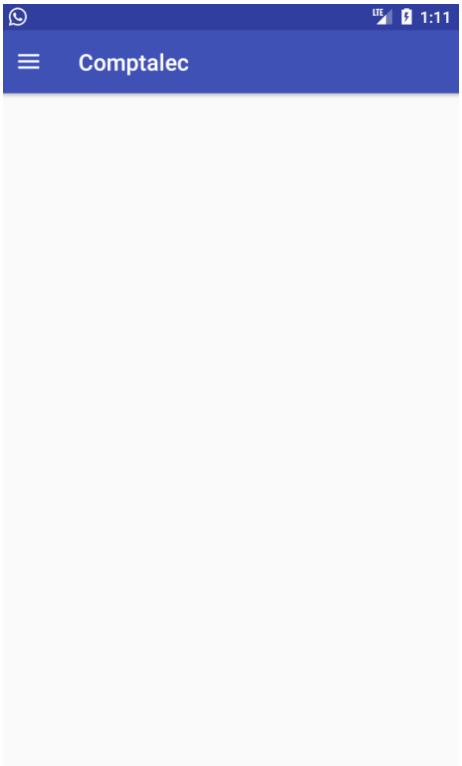


Figura 43: Implementació pantalla principal

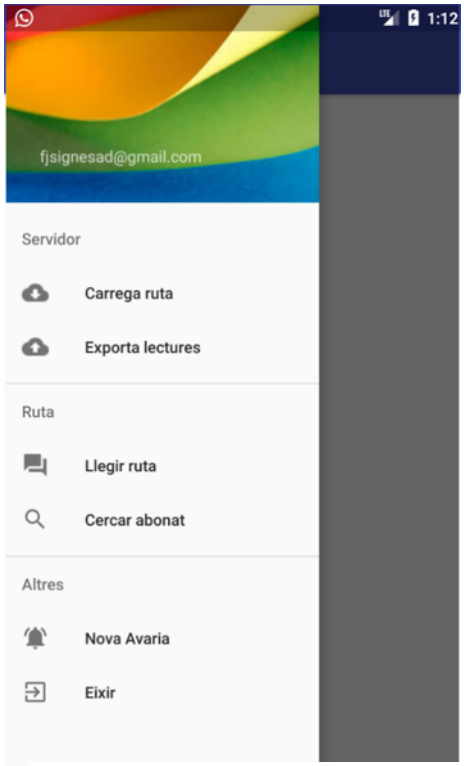


Figura 42: Implementació menú

Figura 44: Implementació pantalla principal quan hi ha una ruta de lectura carregada

3.2.2.1 Detalls d'implementació

La pantalla principal s'ha implementat mitjançant una Activity amb un menú tipus "hamburguesa" situat en la part superior esquerra. La primera vegada que s'entra en l'aplicació, si no tenim cap de ruta de lectura carregada en la base de dades local, ens mostra un fons blanc i sense res. Si tenim una ruta carregada ens mostra la informació de l'últim abonat que s'havia quedat el lector seguint la seua ruta de lectura per tal de què puga continuar des d'aquest abonat.

El menú s'ha implementat mitjançant un DrawerLayout. Al desplegar-se s'observen les diverses opcions que té l'aplicació i prement sobre elles podem anar a les funcionalitats implementades. S'ha tingut la cura de què si hi ha un teclat virtual obert i despleguem el menú principal, tancar el teclat virtual per tal de poder veure totes les opcions del menú.

Una altra consideració que s'ha tingut present es que si l'usuari no li dona a eixir de l'aplicació en tornar a reobrir-la carrega aquesta pantalla principal sense demanar usuari i contrasenya. Si l'usuari ix de l'aplicació en tornar a entrar es mostrarà la pantalla de Login.

3.2.2.2 Proves

Definició de la prova	Comportament esperat	Resultat
Entrar en l'aplicació sense cap ruta de lectura carregada estant validat	Mostrar el fons blanc i el menú a la part esquerra	Correcte

Entrar en l'aplicació amb un ruta carregada estant validat	Mostrar la pantalla de lectura per l'últim abonat que s'havia quedat el lector abans de tancar l'aplicació	Correcte
Entrar en l'aplicació sense estar validat	Mostrar la pantalla de login	Correcte
Obrir el menú	Mostrar totes les opcions del menú	Correcte
Obrir el menú estant el teclat virtual obert	Es tanca el teclat virtual i es mostra el menú	Correcte
Prémer les opcions de menú	Mostrar la pantalla pertinent a la determinada opció	Correcte

3.2.3 Carregar ruta de lectura

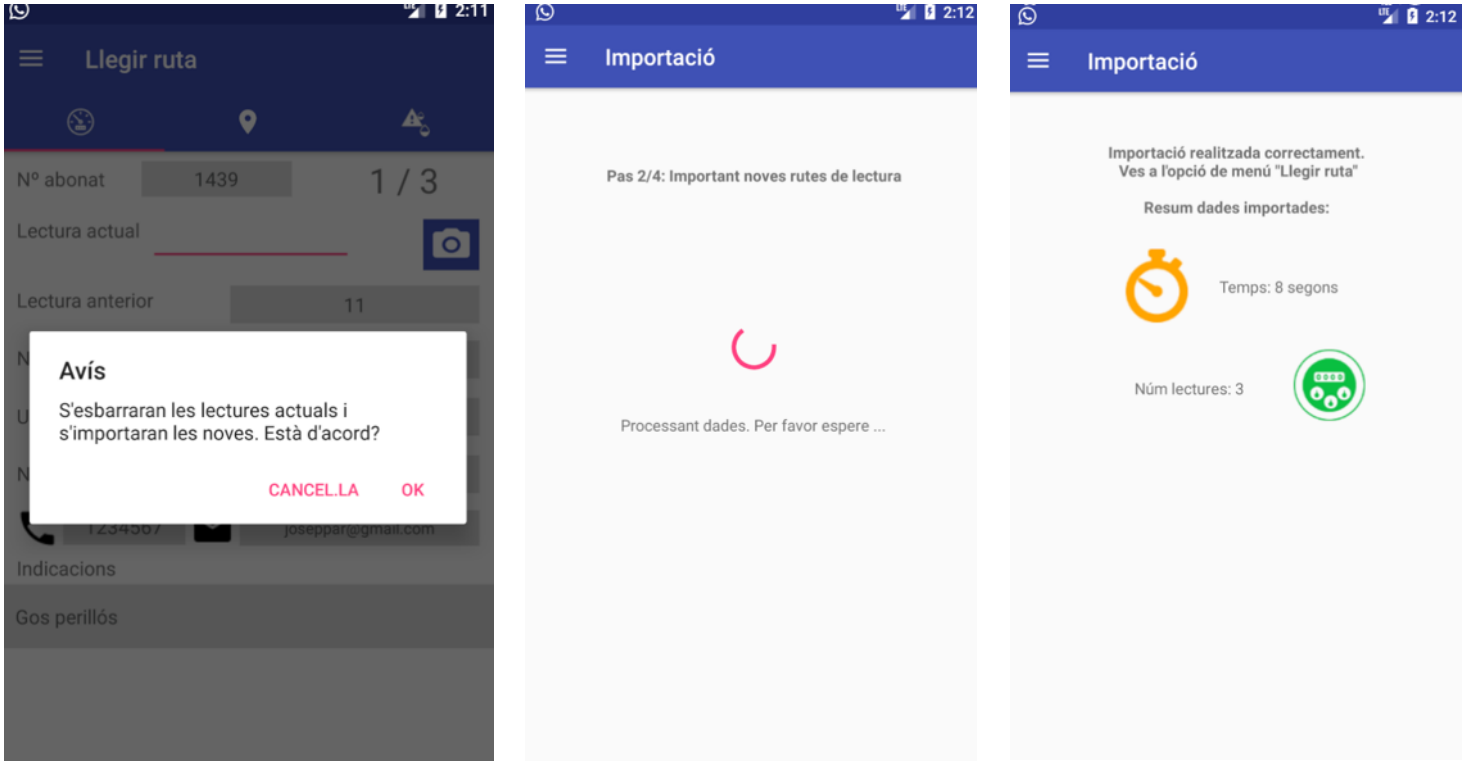


Figura 45: Implementació pantalla carregar ruta de lectura

3.2.3.1 Detalls d'implementació

La importació de rutes de lectura del servidor consisteix a baixar un fitxer en format JSON i emmagatzemar la informació d'eixe fitxer en una base de dades SQLite en local.

S'ha realitzat mitjançant un Fragment.

En primer lloc se li adverteix a l'usuari que la importació d'una nova ruta esborrarà tota la informació que conté la base de dades local. Si l'usuari accepta, s'esborra tot el contingut de les taules de la base de dades locals, es reinicien les seqüències de les claus primàries autoincrementals d'aquestes taules i es baixa de Firebase en format JSON la informació dels abonats d'una ruta que se'ls ha de prendre lectures. Aquesta informació rebuda s'introdueix de nou en la base de dades SQLite local i es carrega en la classe anomenada ReadingPath. Com s'ha comentat abans és una classe Singleton que actua com un tipus de memòria cau per interactuar amb la base de dades. Així mateix,

s'importen els tipus d'avaries i els tipus d'incidències possibles. D'aquesta forma l'aplicació de gestió de lectures de la comunitat de regants és la que marca les possibles opcions a triar pel que fa a tipus d'avaries o tipus d'incidències. I si se n'afegeix un nova, es mostrarà automàticament en l'aplicació a l'importar una nova ruta.

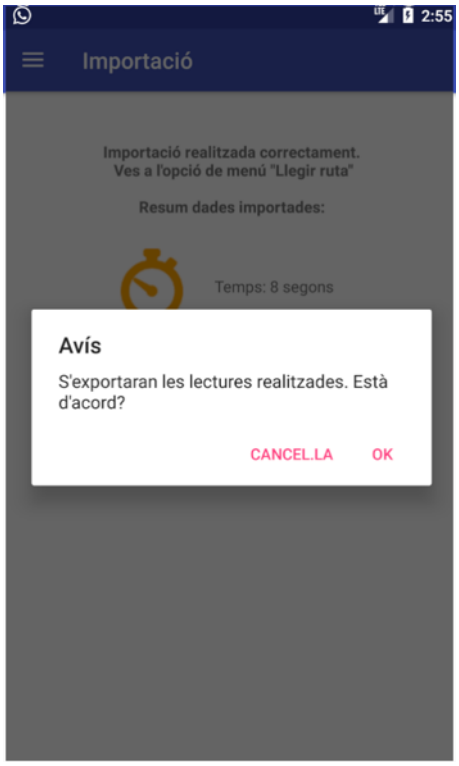
La pantalla d'importació ens mostra en tot moment el pas en el que es troba del total que s'han de fer. A més també se li indica a l'usuari que s'està processant informació i que s'espere. En acabar, l'aplicació informa del temps que ha tardat, així com el nombre d'abonats que conté aquesta ruta de lectura.

Cal fer menció de què si no hi ha connexió de dades activa (no es té cobertura ni xarxa wifi o estan desactivades les dades) el sistema informa a l'usuari amb un error.

3.2.3.2 Proves

Definició de la prova	Comportament esperat	Resultat
Entrar en l'opció de menú "Carregar ruta" i cancel·lar l'opció de la finestra emergent	El sistema mostra una finestra d'advertència indicant que s'esborra la informació que tenim en local. El sistema tanca la finestra després de què l'usuari cancel·le	Correcte
Entrar en l'opció de menú "Carregar ruta" i acceptar l'opció de la finestra emergent	El sistema mostra una finestra d'advertència indicant que s'esborra la informació que tenim en local. El sistema comença a importar mostrant una barra de progrés circular donant informació del que fa. En acabar mostra resum del que s'ha fet	Correcte
Entrar en l'opció de menú "Carregar ruta" sense tindre connexió de dades	El sistema mostra un missatge d'error indicant a l'usuari error de connexió a la xarxa	Correcte

3.2.4 Exportació lectures



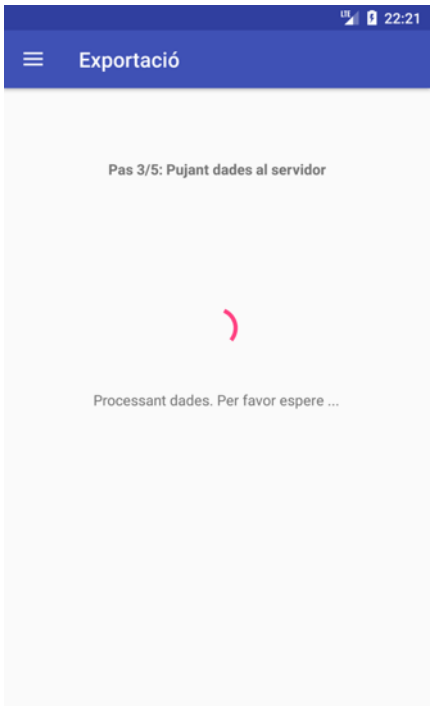


Figura 46: Pantalles implementació exportació lectures

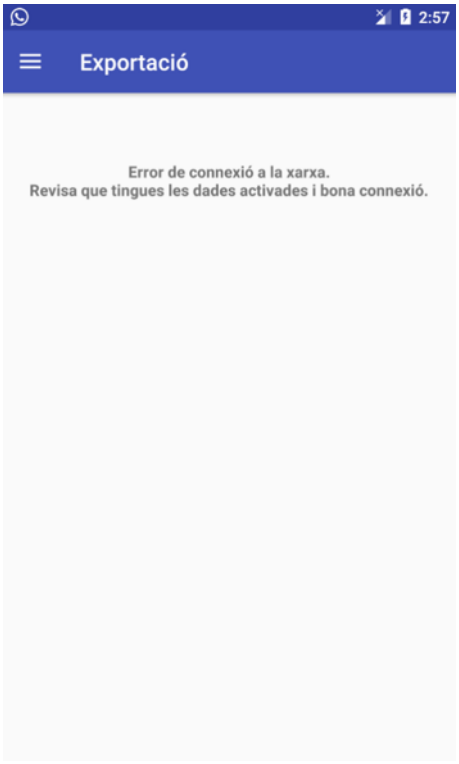


Figura 47: Missatges pantalla exportació lectures

3.2.4.1 Detalls d'implementació

L'exportació de lectures consisteix en enviar un fitxer comprimit en format JSON amb les lectures i les fotos de les incidències a Firebase Storage.

Aquesta pantalla comparteix la vista amb la pantalla d'importació. L'estructura és la mateixa i l'únic que canvien són els missatges.

Un avantatge que s'aconsegueix amb el MVP és que es pot fer ús de la mateixa vista. Només es canvia la implementació dels Presenters en la importació de rutes de lectura i en l'exportació de les lectures.

El primer que es fa quan s'executa l'opció de menú, a l'igual que en la pantalla d'importació, és indicar amb un missatge a l'usuari que la informació de les lectures realitzades s'esborrarà de la base de dades local i s'enviarà al servidor. Si l'usuari autoritza l'operació el que es fa és exportar la informació de les lectures a una fitxer en format JSON. S'exporta la informació dels abonats que tenen lectures preses en aquesta ruta i les incidències relacionades amb aquestes. Alhora també es copien les fotos de les incidències a un directori temporal. Seguidament es procedeix a comprimir el fitxer JSON i el directori temporal de fotos. Després es puja el fitxer comprimit al servidor. Si s'ha pujat de manera satisfactòria la informació es passa a esborrar el fitxer temporal JSON, el directori temporal d'imatges, a esborrar els abonats exportats de la base de dades i les incidències relacionades així com les imatges de les incidències aquestes. D'aquesta manera no es deixa brutícia per cap lloc. Es realitza aquest pas d'esborrament com a últim perquè si hi ha algun tall de la xarxa o error i no es transmet la informació al servidor, sempre es podrà tornar-ho a reintentar i es tindrà la informació encara en local.

En acabar es mostra una estadística de la informació que s'ha exportat així com el temps que s'ha tardat.

Si no hi ha connexió de dades quan comença l'exportació de dades, l'aplicació avisa amb un missatge a l'usuari. També cal indicar que l'aplicació mentre estan fent-se tots els passos per a exportar la informació al servidor, li dona a l'usuari una retroalimentació del pas que està fent i quin número té aquest del total dels passos que s'han de fer per exportar la informació. Aleshores l'usuari en tot moment veu que l'aplicació està fent coses i no es pensa que pot estar penjada.

3.2.4.2 Proves

Definició de la prova	Comportament esperat	Resultat
Entrar en l'opció de menú "Exportar lectures" i cancel·lar l'opció de la finestra emergent	El sistema mostra una finestra d'advertència indicant que s'exportarà la informació de les lectures realitzades. El sistema tanca la finestra després de què l'usuari cancel·le.	Correcte
Entrar en l'opció de menú "Exportar lectures" i acceptar l'opció de la finestra emergent	El sistema mostra una finestra d'advertència indicant que s'exportarà la informació de les lectures realitzades. El sistema comença a exportar mostrant una barra de progrés circular donant informació del que fa. En acabar mostra resum del que s'ha exportat.	Correcte
Entrar en l'opció de menú "Exportar lectures" sense tindre connexió de dades	El sistema mostra un missatge d'error indicant a l'usuari error de connexió a la xarxa.	Correcte
Entrar en l'opció de menú "Exportar lectures" sense tindre lectures introduïdes	El sistema mostra un missatge indicant que no hi ha cap lectura per a exportar.	Correcte

3.2.5 Llegir ruta

Aquesta pantalla està formada per un Fragment amb un TabLayout amb diverses pestanyes. Entre aquestes pestanyes tenim la d'introduir informació de lectura, la de veure la posició del comptador en un plànol i la d'introduir una incidència d'un abonat. Cadascuna d'aquestes pestanyes són alhora també un Fragment. Com a dificultat tècnica trobada indicar que en principi només es carregava una única pestanya al costat de la pestanya activa. Aleshores com que tenim 3 pestanyes si s'estava en la pestanya de lectura i es passava a la pestanya d'incidència no es carregava la informació de l'abonat en el que s'estava en la ruta. Això es devia a què per defecte un ViewPager només carrega una pestanya al costat de la finestra activa. La solució ha passat per augmentar aquest valor a 2 mitjançant el mètode setOffscreenPageLimit.

3.2.5.1 Botonera
3.2.5.1.1 Detalls d'implementació

Un altre element a destacar de les pantalles per a llegir la ruta és la botonera.

Com s'observa en les imatges del costat s'amaguen o es mostren botons en funció de si estem al primer element de la ruta de lectura, en l´últim o en un element intermedi.

Figura 48: Implementacions de la botonera

Per a aquesta botonera s'ha inclòs el seu layout dins d'un fitxer i després s'inclou des de les pestanyes de lectura, posició en mapa i incidència per tal de reaprofitar el codi. A més a més també s'ha creat una classe denominada BotoneraHandler per controlar la lògica de navegació per la ruta de lectura quan estiguem en la pantalla de lectura, del mapa d'ubicació de comptador o incidència. Així es refresca la informació de cada pestanya sempre que premem la botonera en cadascuna de les pantalles incloses en el Tab. Es tracta aleshores d'una espècie de component.

Quan es premen els botons de moviment, si s'ha modificat alguna informació de lectura i incidència relacionada amb l'abonat actual de la ruta, aquesta s'emmagatzema de manera persistent.

3.2.5.1.2 Proves

Definició de la prova	Comportament esperat	Resultat
Estem en el primer element de la ruta de lectura	S'amaguen els botons de primer i anterior element en la botonera	Correcte
Estem en l'últim element de la ruta de lectura	S'amaguen els botons de següent i últim element en la botonera	Correcte
Estem en un element central de la ruta de lectura	Es mostren tots els botons de la botonera	Correcte
Estem en un element de la ruta i premem el botó de següent, anterior, primer o últim	El sistema es mou a la posició de la ruta de lectura corresponent segons s'haja pres un botó o un altre i si s'ha modificat alguna informació aquesta s'ha emmagatzemat correctament	Correcte

3.2.5.2 Presa de lectura

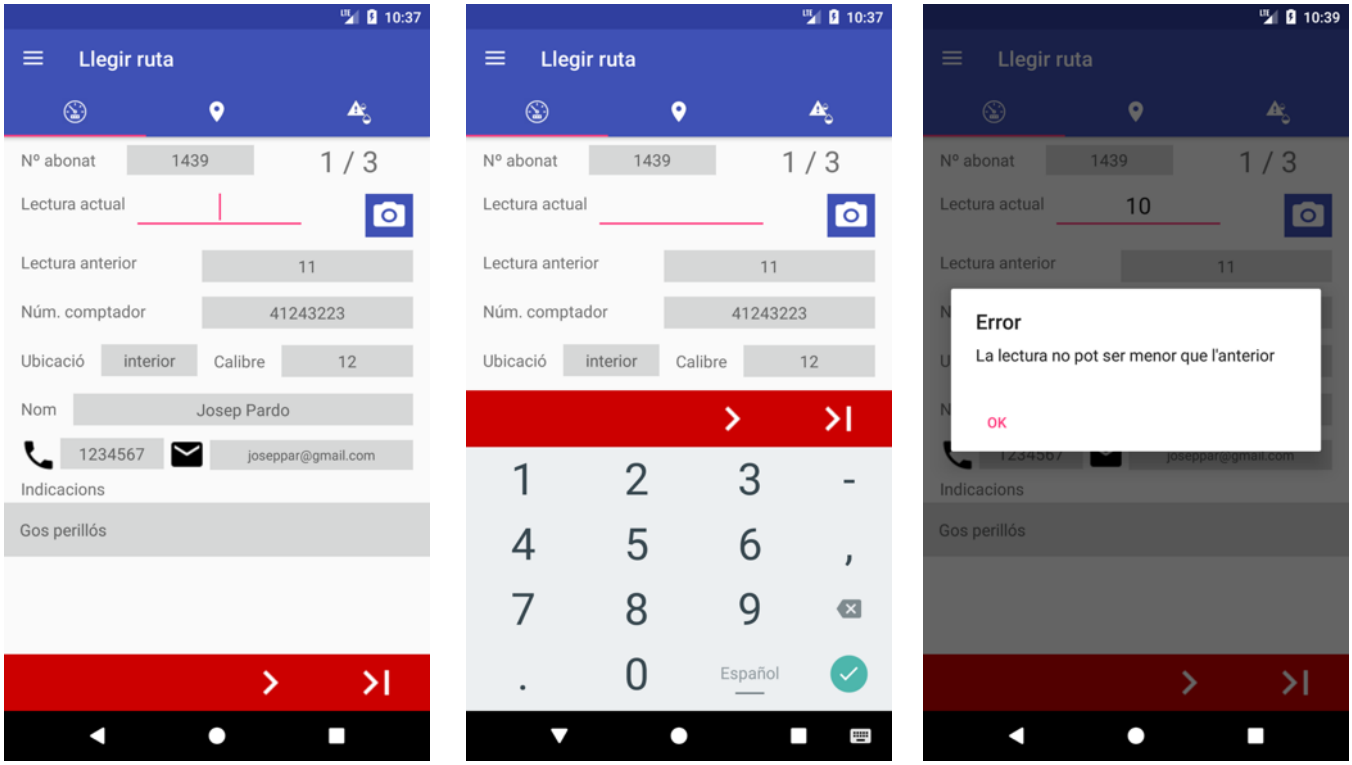


Figura 49: Implementació pantalla de lectura de comptador

3.2.5.2.1 Detalls d'implementació

En aquesta pantalla tenim la informació de l'abonat actual de la ruta de lectura. Aquesta pantalla va canviant la informació segons anem prement en la botonera. Només hi ha un camp d'escriptura i la resta són de consulta. El camp d'escriptura només és de tipus numèric. Aleshores quan es mostra el teclat virtual només mostra xifres. En introduir la lectura i prémer algun botó de la botonera la informació s'emmagatzema en la base de dades. Si la lectura és menor que la lectura anterior el sistema informa amb un error i no podem canviar d'element de la ruta de lectura.

Si estem amb el teclat virtual desplegat el lector pot anar movent-se per la ruta de lectura amb la botonera i anotant les lectures donat que té la informació més importat dalt del teclat virtual. Així no cal que posicione cada vegada el cursor en el camp de lectura actual per a què es desplegue el teclat virtual.

Cal indicar també que si premem el botó de la càmera ens mostra la càmera i al damunt una capa amb els possibles textos que hi ha a la imatge.

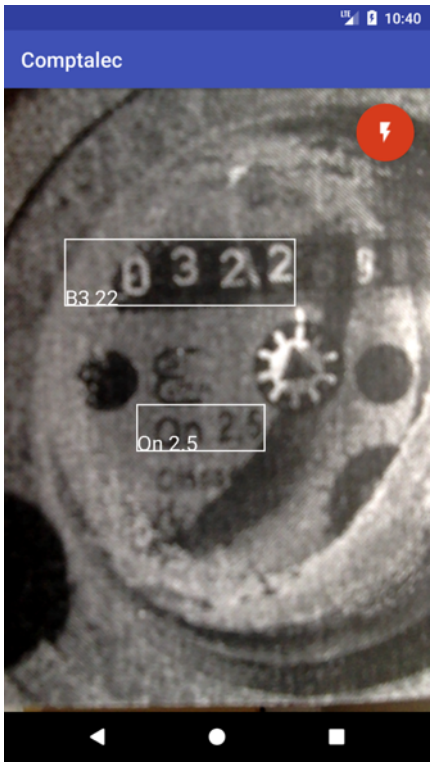


Figura 50: Captura OCR de lectura

En aquesta pantalla el que s'ha fet és usar l'API OCR de Google Play Services Vision. El que fa es agafar imatges de la càmera i processar-les detectant textos en ella. Després mostra en una nova capa al damunt de la imatge el text que ha reconegut. Quan l'usuari prem al damunt d'un text reconegut es tanca la càmera i es posa aquest text dins del camp de lectura actual del comptador.

En aquesta pantalla s'ha inclòs el botó d'activar el flash donat que de vegades els comptadors estan en llocs on hi ha poca visió.

Com que l'OCR de vegades detecta lletres on eren números se li ha aplicat un algorisme de neteja de caràcters estranys i passar lletres detectades a números (per exemple la lletra "o" es converteix en un zero o la "l" es converteix en un ú). A més també es procedeix al filtratge de la part decimal del metres cúbics consumits donat que aquesta part no es factura. Només es facturen metres cúbics sencers no fraccions. Aquesta part s'aconsegueix anant llevant caràcters de la dreta fins que el número detectat siga menor que el triple de la lectura anterior.

3.2.5.2.2 Proves

Definició de la prova	Comportament esperat	Resultat
Introduir lectura menor que l'anterior i intentem canviar d'abonat	El sistema indica un error que no es pot emmagatzemar un lectura que siga menor que l'anterior	Correcte

Introduir lectura major que l'anterior i intentem canviar d'abonat	El sistema emmagatzema la lectura	Correcte
Prémer el botó de càmera i no tindre permisos d'ús de la càmera	El sistema mostra una finestra demanant els permisos	Correcte
Prémer el botó de càmera i no tindre suficient espai per poder processar informació OCR	El sistema mostra un missatge d'error indicant que no podrà reconèixer text	Correcte
Prémer el botó de càmera i no tindre descarregades les llibreries OCR de Google	El sistema mostra un missatge d'error indicant que no podrà reconèixer text	Correcte
Prémer el botó de càmera i no tindre instal·lat Google Play	El sistema mostra un missatge indicant que s'ha d'instal·lar Google Play	Correcte

3.2.5.3 Ubicació de comptador en el mapa

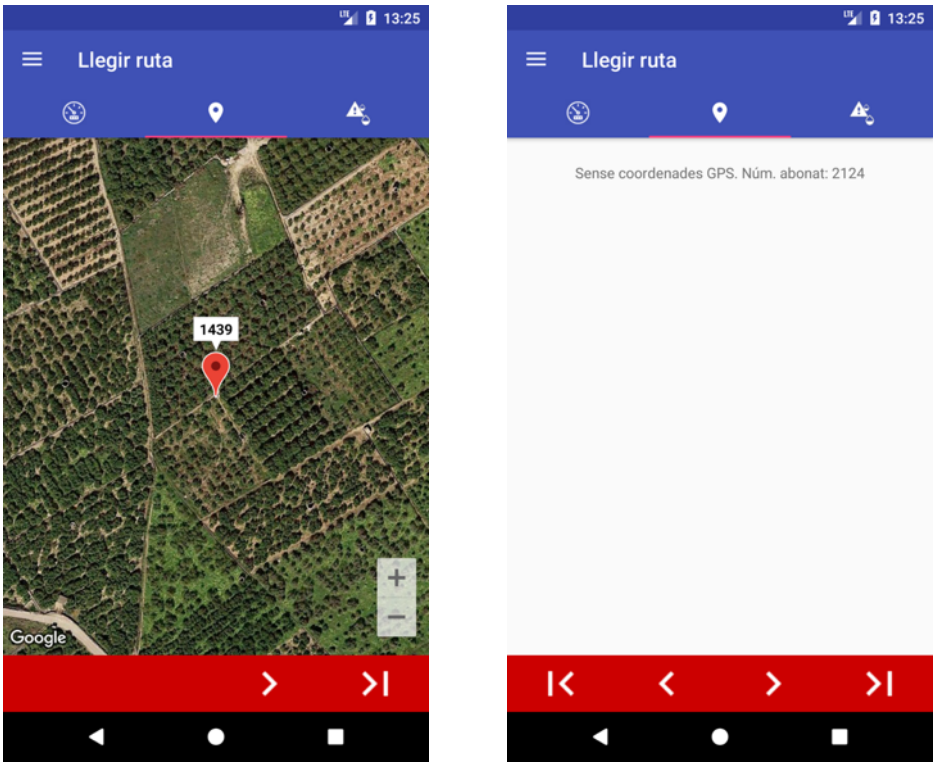


Figura 51: Implementació posició comptador en el mapa

3.2.5.3.1 Detalls d'implementació

Per implementar aquesta pantalla s'ha fet ús dels mapes de Google. Per a això s'ha donat d'alta una clau per poder fer ús d'aquest API. I aquesta clau s'ha posat dins del fitxer d'AndroidManifest.xml.

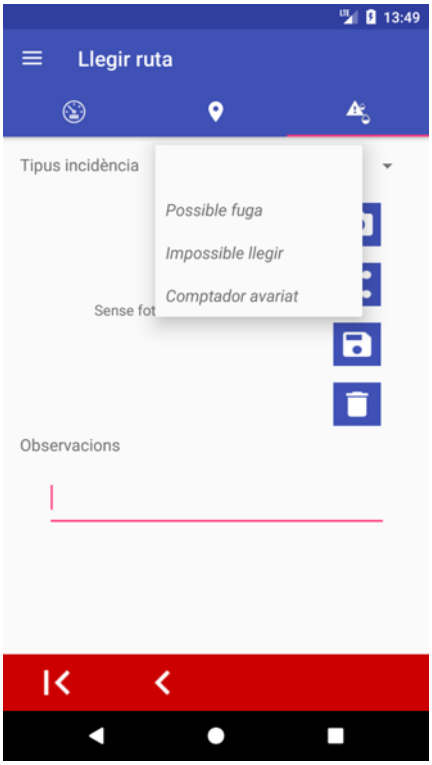
S'ha inclòs la botonera al final per poder moure'ns per la ruta de lectura i poder veure en el mapa on estan situats els comptadors dels abonats. Si en la informació de l'abonat venen les coordenades GPS del comptador ens mostra el mapa amb un símbol on està situat el comptador i un text indicant el número

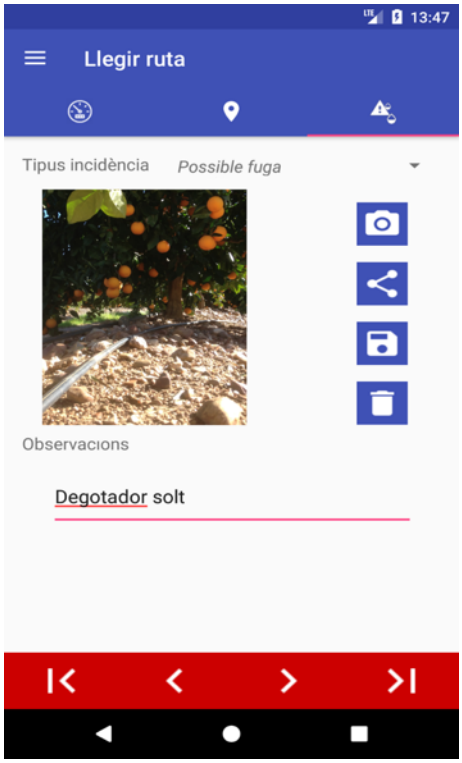
d'abonat. En cas que no tinguem les coordenades avisa amb un missatge de què no tenim la informació per a mostrar-ho en el mapa.

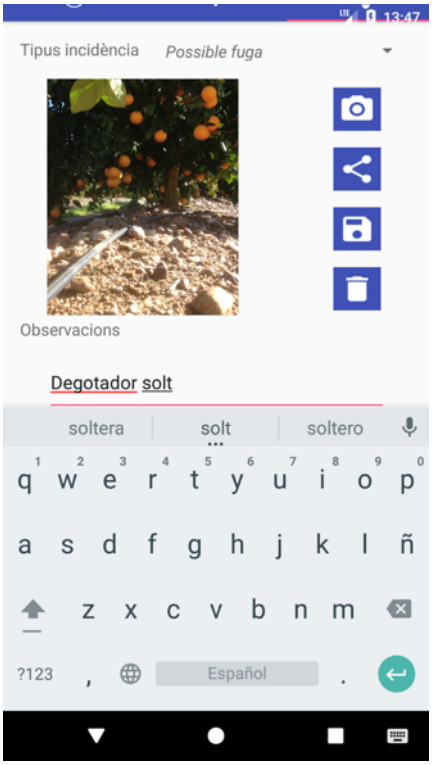
3.2.5.3.2 Proves

Definició de la prova	Comportament esperat	Resultat
Prémer en la icona de localització en la pestanya superior tenint informació de coordenades GPS de l'abonat	El sistema mostra un mapa amb una icona indicant la posició i un cartell mostrant el número d'abonat	Correcte
Prémer en la icona de localització en la pestanya superior sense tindre informació de coordenades GPS de l'abonat	El sistema mostra missatge indicant que no hi ha coordenades GPS per a aquest abonat.	Correcte

3.2.5.4 Incidència abonat







3.2.5.4.1 Detalls d'implementació

En aquesta pantalla es pretén enregistrar una incidència relacionada amb un abonat. A més també es permetrà la possibilitat de comunicar-la a l'abonat.

En primer lloc els valors del desplegable es carreguen a partir de valors d'una taula de la base de dades que s'importa a l'hora d'importar informació d'una ruta. Aleshores quan es té una incidència enregistrada s'ha de tornar a emplenar amb el valor del tipus d'incidència que s'havia emmagatzemat. Els valors del desplegable es tenen en la classe ReadingPath que actua com a memòria cau per accelerar l'emplenament dels desplegables i el moviment per la ruta de lectura. Pel que fa a l'emplenament del camp observacions s'ha tingut la cura de què quan es pose el cursor en aquest camp, el teclat virtual no tape el

que s'està escrivint. Referent a la introducció d'una fotografia de la incidència s'ha tingut present que si no es tenen els permisos d'accés a la càmera es demanen per a poder prendre la fotografia. Aquesta fotografia s'emmagatzema en l'estructura d'arxius de l'aplicació en un directori concret de cau que hem creat per a aquesta finalitat. En la base de dades local només s'emmagatzema el nom del fitxer per així alleugerir allò que té aquesta. Cada fitxer segueix una nomenclatura del tipus: id_incidencia_time_stamp_data_presa_fotografia.jpg. La ruta inicial del directori cau de fotos s'ha indicat amb un fileprovider dins del fitxer xml/fileprovider.xml. Després amb la constant DIRECTORY_ISSUE_PHOTO del fitxer util/Constants.java s'ha indicat el subdirectori dins del directori de cau. La ruta física exacta dins del dispositiu pot quedar així segons els dispositiu: /data/data/edu.uoc.fjsignes.comptalec/app_issuepictures

S'ha agafat la foto en mesura real i no la foto en miniatura per així tindre major definició exacta de la incidència. Per a això s'ha deixat la fotografia presa en un fitxer temporal i després s'ha copiat a la ruta definitiva. La mesura d'aquesta imatge gran es defineix en el fitxer de constants amb els paràmetres PHOTO_MAX_WIDTH i PHOTO_MAX_HEIGHT.

A l'hora de compartir aquesta incidència es pot prémer el botó de compartir . Si es comparteix la informació per Whatsapp s'envia també la imatge si existia en el missatge. Per facilitar el trobar el contacte en Whatsapp s'ha copiat al portapapers l'adreça email de l'abonat. Així en obrir el Whatsapp, cercar i apegar l'adreça d'email (si es té en el nom del contacte l'email també) localitza el destinatari de forma ràpida. A més si s'envia per correu electrònic s'ha posat l'adreça d'email de l'abonat com a destinatari annexant la fotografia també si existeix. Per poder compartir amb altres aplicacions la fotografia s'ha hagut de crear un fitxer temporal i passar-li la URI d'aquesta com a paràmetre en la invocació d'altres aplicacions perquè sinó no es podia compartir la fotografia.

A continuació tenim dos captures de com ha quedat la compartició d'informació per Whatsapp i per correu electrònic. A més també es té la fotografia ampliada en una finestra independent que s'obri al punxar sobre la fotografia en miniatura.

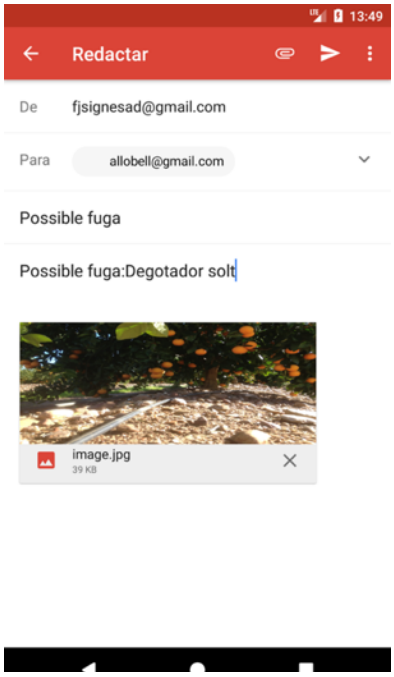


Figura 52: Exemple compartir incidència per email

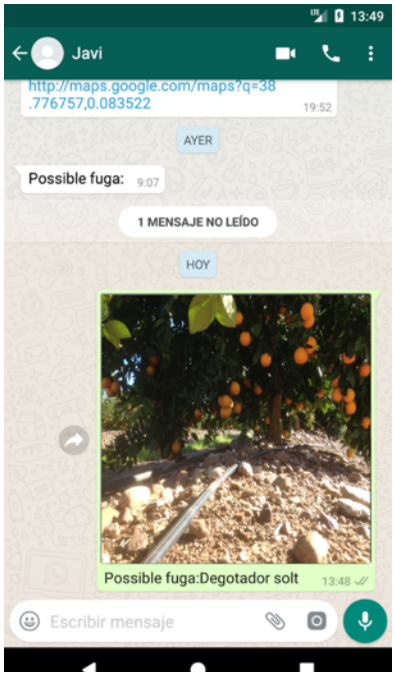


Figura 53: Exemple compartir incidència per Whatsapp

Indicar que només es passa la informació introduïda en la incidència. És a dir, si hi ha text es passa el text i si hi ha imatge també. Si no hi ha cap informació per enviar i es prem el botó de compartir l'aplicació avisa que no hi ha res per compartir.

Com a dificultat tècnica trobada en aquest Fragment ha sigut que al llançar activitats i esperar la resposta inicialment només les rep l'Activity pare on està inclòs el Fragment. En aquest cas MainActivity. Aleshores el que s'ha fet és que les classes heretaren de NestedActivityResultFragment. Es tracta d'una classe continguda en la llibreria com.inthecheesefactory.thecheeselibrary:stated-fragment-support-v4:0.10.0. A més se li ha indicat a la classe MainActivity que reenviara els esdeveniments a tots els seus Fragments fills posant dins de onActivityResult el següent codi:

```
ActivityResultBus.getInstance().postQueue(new
ActivityResultEvent(requestCode, resultCode, data));
```

3.2.5.4.2 Proves

Definició de la prova	Comportament esperat	Resultat
Emplenar informació de tipus d'incidència, observacions, fotografia i prémer el botó de desar	S'ha emmagatzemat la informació en la base de dades i al tornar al mateix abonat està la informació introduïda (inclosa la fotografia si existeix)	Correcte
Esborrar fotografia de la incidència	Lleva la fotografia de la base de dades i del directori físic on estava emmagatzemada	Correcte
Compartir incidència sense tindre informació de la incidència	El sistema mostra un missatge indicant que no hi ha cap informació per compartir	Correcte
Compartir incidència amb text i fotos per Whatsapp	S'apega el valor del portapapers en la cerca de Whatsapp i es té l'email de l'abonat. Després s'envia la foto i el text a sota de la fotografia	Correcte
Compartir text d'incidència sense foto per Whatspp	Es pega el valor del portapapers en la cerca de Whatsapp i es té l'email de l'abonat. Després s'envia el text	Correcte
Compartir incidència amb text i fotos per correu electrònic	S'obri l'aplicació de correu electrònic amb un nou missatge dirigit a l'email de l'abonat que té la incidència, el tema del correu és el tipus d'avaria. En el cos del missatge es tenen les observacions i la fotografia annexada	Correcte
Compartir text d'incidència sense foto per correu electrònic	El mateix que l'anterior punt però no conté la fotografia	Correcte
Fer fotografia sense tindre permisos per obrir la càmera	El sistema demana permisos per poder obrir la càmera	Correcte
Fer fotografia sense tindre'n cap abans	El sistema pren la fotografia, l'emmagatzema al directori corresponent amb el nom adequat, insereix el nom de la mateixa a la base de dades i posa la foto en miniatura l'espai reservat per a la foto en la pantalla	Correcte

Fer fotografia tenint-ne una abans	El sistema esborra del directori de fotos la foto anterior, posa la nova, actualitza el contingut de la base de dades i mostra la miniatura.	Correcte
Prémer el botó d'esborrar	El sistema esborra la fotografia del directori (si existeix) i s'esborra la informació de la base de dades	Correcte
Prémer damunt d'una foto en miniatura en la incidència	El sistema mostra la fotografia en gran en una nova finestra	Correcte
Prémer damunt d'una foto en miniatura en la incidència sense tindre cap fotografia associada	El sistema no fa res	Correcte

3.2.6 Cercar abonat

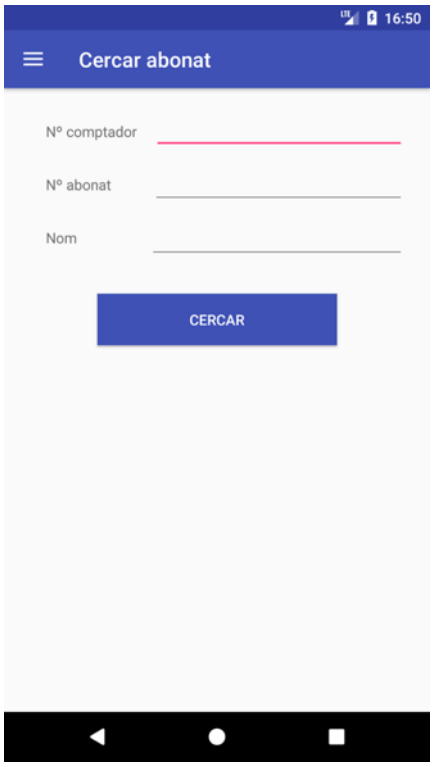


Figura 54: Implementació cercar abonat

Figura 55: Resultat cercar abonat

Figura 56: Detall d'un abonat cercat

3.2.6.1 Detalls implementació

La cerca d'abonat consisteix en tres pantalles. Una primera que mostra els criteris de cerca. Una segona que treu els resultats de la cerca. I una tercera que al prémer sobre un resultat de cerca ens mostra el detall d'un abonat determinat.

La pantalla de cerca d'abonat conté diversos paràmetres i al cercar anem al resultat de la cerca. El que fa és LIKE del contingut de cada criteri de cerca i un AND de tots els camps que tenen algun criteri introduït.

El resultat de la cerca s'emmagatzema en la classe SearchResultManager que permet tindre tots els elements resultants de la cerca.

En la pantalla del resultat de cerca mostrem en un RecyclerView tots els elements que conté la classe SearchResultManager. Al prémer sobre un element de la llista s'obri la informació d'un abonat concret. La pantalla amb la informació de l'abonat concret és la mateixa que la de llegir abonats però en

aquest cas llevant-li la botonera i afegint un botó “desa” a la pantalla de lectura. S’ha fet així per donar-li uniformitat a la interfície d’usuari i també s’ha reaprofitat el codi ja elaborat.

Com ha dificultat tècnica trobada ací ha sorgit que la pantalla de lectura era un fragment que estava dins de la pantalla principal. Al ser ací una Activity independent per a poder afavorir el tornar al resultat de cerca sense haver de tornar a cercar novament, s’ha hagut de crear una nova Activity que contenia al seu interior el Fragment d’un abonat concret.

3.2.6.2 Proves

Definició de la prova	Comportament esperat	Resultat
Intentar cercar sense posar cap criteri de cerca	El sistema indica un missatge alertant de què s’han de posar criteris de cerca	Correcte
Posar un criteri de cerca que obté resultats	El sistema mostra la pantalla de resultats que compleixen el criteri de cerca	Correcte
Posar criteris de cerca que no obtenen resultats	El sistema mostra en la finestra de resultat que no hi ha resultat de cerca per els criteris introduïts	Correcte
Combinar diversos criteris de cerca que sabem que treuen resultats	Comprovar que els resultats són els adients	Correcte
Prémer en un resultat de cerca	El sistema mostra la informació d’aquest abonat	Correcte
Prémer el botó “desa” de la pantalla de resultat d’un abonat	El sistema emmagatzema en la base de dades la lectura si s’havia introduït	Correcte
En el resultat d’un abonat introduir nova incidència	El sistema emmagatzema la informació introduïda de la incidència i avisa amb un missatge	Correcte
Introduir lectura amb OCR funcione	El sistema obri la càmera i al tornar emmagatzema la lectura al camp adient	Correcte

3.2.7 Nova avaria

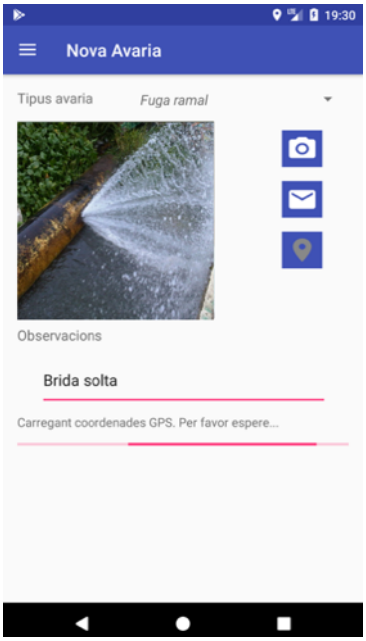


Figura 57: Pantalla avaria

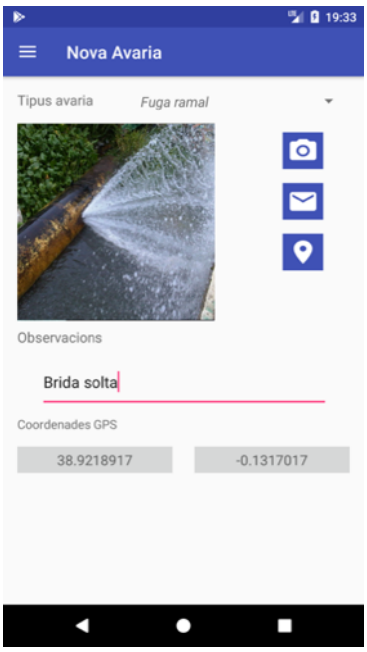


Figura 58: Pantalla enviament avaria

3.2.7.1.1 Detalls d'implementació

La part visual d'aquesta pantalla és semblant a la d'una nova incidència d'abonat. L'únic que canvia és que en aquesta pantalla podem prendre les coordenades de la nostra ubicació actual i només enviar les dades de l'avaria per un dels clients de correu electrònic que té el dispositiu mòbil instal·lat. Per la qual, i aprofitant l'avantatge d'haver usat el model vista presentador, s'ha usat la mateixa vista. I la part del Presenter de la nova avaria hereta del Presenter de nova incidència. Així el Presenter de la nova avaria sobreescriu aquells mètodes del Presenter de nova incidència que tenen un comportament diferent. A més també afegeix nous mètodes per al tractament específic de funcionalitats que té la nova avaria que no té la introducció d'una nova incidència.

El tema de la presa d'una foto segueix els mètodes usats en la classe pare, a diferència que ací només es queda la foto en el directori temporal perquè no s'ha d'emmagatzemar en la base de dades informació de l'avaria. Només s'envia per correu. Quan es pren una nova foto aquesta foto temporal es modificada.

Pel que fa a com agafar les coordenades de la ubicació actual hi havien dues alternatives: usar l'API de Google Play Location Service i l'API d'Android Location Services. Seguint les últimes recomanacions que ha indicat Google, i aprofitant que ja es tenia instal·lat Google Play per a l'ús d'altres APIS com el de Maps, s'ha optat per utilitzar l'API de Google Play Location Service. Aquest API permet treure la ubicació de diverses formes segons la precisió de les coordenades, el consum de bateria que emprem i el tipus de xarxa a partir del qual s'obté la posició. S'ha optat per agafar les coordenades amb la màxima precisió i per a això s'ha de fer ús del GPS instal·lat en el dispositiu mòbil. Fer menció també que encara que a l'hora d'aconseguir la ubicació tarda més temps que altres mètodes, aquest mètode és més precís i no té dependència de xarxes sense fil o de la xarxa de telefonia mòbil que moltes vegades no es tenen a l'abast del lloc concret on treballen els lectors amb aquesta aplicació. També indicar que es necessari conèixer la posició més exacta possible de l'avaria donat que moltes vegades la localització on està ubicada és al mig d'un lloc ple d'herba i les brigades d'avaries poden no trobar-la si la precisió no és el més exacta possible. Amb el GPS aquesta precisió té un error d'aproximadament 5-10 metres.

S'ha tingut la cura de què com el GPS és un sensor que consumeix molta bateria, si el Fragment entra en segon pla o es canvia, desactivarà el GPS. Quan es torne a entrar en l'aplicació si s'havia minimitzat es retorna a usar el GPS si estava engegat. També indicar que només entra en funcionament el GPS quan es prem el botó de localitzar la ubicació d'aquesta pantalla.

Per altra banda comentar també que a l'hora de determinar la posició exacta el sistema itera fins determinar que la posició obtinguda té una errada menor de 10m de les mesures que ens dona el GPS o si s'han produït 25 consultes de posició i no s'ha trobat alguna posició amb aquesta precisió. Aquesta última part es fa per a no entrar en un bucle infinit.

Finalment destacar que quan s'envia el correu electrònic amb la informació de l'avaria presa pel lector conté l'enllaç a Google Maps amb les coordenades per a què així es vega directament on està localitzada l'avaria.

3.2.7.1.2 Proves

Definició de la prova	Comportament esperat	Resultat
Prendre fotografia sense tindre permisos de la càmera	El sistema indica un missatge alertant de què no es tenen permisos i si en volem donar ara	Correcte
Prémer botó d'enviament de correu d'avaria sense tindre dades	El sistema mostra un missatge alertant de què no hi ha dades	Correcte
Prémer el botó de localització sense tindre engegat el GPS	El sistema mostra un missatge indicant si volem engegar el GPS	Correcte
Emplenar dades d'avaria i prémer botó enviament	El sistema mostra una finestra emergent amb només els clients de correu que té instal·lats el dispositiu. Al triar-ne un posa la informació de l'avaria, un enllaç al mapa de Google amb les coordenades (si en té de coordenades l'avaria) i una foto adjunta (si és el cas)	Correcte
No es tenen permisos per obtenir la ubicació i es prem el botó d'aconseguir la posició actual	El sistema mostra un missatge indicant que no es tenen permisos i si s'autoritzen o no els permisos	Correcte

3.2.8 Eixir

3.2.8.1.1 Detalls d'implementació

Aquesta funcionalitat l'únic que realitza es desvincular la informació d'usuari logat en el sistema i ens envia a la pantalla de validació.

3.2.8.1.2 Proves

Definició de la prova	Comportament esperat	Resultat
Prémer l'opció de logout	L'aplicació mostra la finestra de Login	Correcte
Tornar enrere després de fer logout i mostrar pantalla de validació	El sistema mostra encara la pantalla de validació	Correcte

3.3 Proves realitzades

Les diverses proves que s'han realitzat s'han indicat en els apartats anteriors després de cada pantalla. Cal remarcar que es tracta de diverses proves funcionals, proves unitàries automàtiques o codificades i instrumentades. Les proves unitàries s'han realitzat d'aquelles classes i/o mètodes que s'han considerat més importants prenent com a premissa que l'esforç que s'ha de dedicar a fer el 100% de les proves automàtiques front a la recompensa obtinguda no és proporcional. A més no té sentit arribar al 100% de cobertura en un test codificat perquè no tot el codi es pot provar des d'un punt de vista unitari o es prova en un altre tipus de test.

Fer menció també que s'han realitzat proves amb el servei de Firebase Test Lab per tal de veure el funcionament de l'aplicació amb diverses versions d'APIS i possibles errades en diversos dispositius físics i virtuals. A més també s'han emprat dos dispositius físics per part del desenvolupador per veure el comportament de l'aplicació i si la interfície era la desitjada. Indicar que en tots els casos han sigut satisfactòries les proves.

A continuació es mostra el resultat de les proves automàtiques no instrumentades:

<default package>	8s 526ms
LocationInfoPresenterImplTest	1s 73ms
shouldShowDataOnMapWhenAbonatHasGPSCoordenates	1s 72ms
shouldShowNoDataMessageWhenAbonatHasNOGPSCoordenate	1ms
ExportPresenterImplTest	1s 73ms
shouldShowErrorMessageWhenNetworkDataIsNotAvailable	1s 57ms
shouldShowMessageWhenNoDataToExport	4ms
shouldShowMessageWhenDataExportedProperly	12ms
ImportPresenterImplTest	1s 17ms
shouldShowErrorMessageWhenNetworkDataIsNotAvailable	1s 11ms
shouldShowMessageWhenImportingData	6ms
SearchResultPresenterImplTest	521ms
shouldShowAlertMessageWhenNoResultsInASearch	515ms
shouldShowResultsMessageWhenDataInASearch	6ms
LoginPresenterImplTest	986ms
shouldShowErrorMessageWhenLoginIsEmpty	940ms
shouldShowErrorMessageWhenPasswordIsEmpty	1ms
shouldCheckValidationWhenLoginAndPasswordAreNotEmpty	5ms
shouldShowLastLoginWhenStoreLoginIsNotEmpty	27ms
shouldSetLastLoginWhenIsNotEmpty	13ms
SearchPresenterImplTest	828ms
shouldShowErrorMessageWhenSearchCriteriaAreEmpty	826ms
shouldShowResultScreenWhenSearchCriteriaAreNotEmpty	2ms
FailurePresenterImplTest	918ms
shouldShowAlertMessageWhenGPSisDisabled	907ms
shouldShowAlertMessageWhenLocationPermissionisDisabled	4ms
shouldShowProgressBarAndRequestLocationsWhenGettingLoca	3ms
shouldShowAlertWhenSendingNewFailureEmptyData	2ms
shouldShowBigPictureWhenClickOnThumbnail	2ms
IssuePresenterImplTest	1s 294ms
shouldShowSavedIssueDataWhenItWasStoredPreviously	1s 284ms
shouldShowEmptyIssueDataWhenItIsEmpty	6ms
shouldShowBigPictureWhenDataStoreInDatabaseAndClickOnThu	4ms
ReadingInfoPresenterImplTest	814ms
shouldStoreNewReadWhenUserMovesWithBotonera	802ms
shouldShowErrordWhenNewReadisLowerThanBefore	2ms
shouldStoreReadAndMoveNextWhenNextButtonBotonerClicked	3ms
shouldStoreReadAndMovePriorWhenNextButtonBotonerClicked	3ms
shouldStoreReadAndMoveLastWhenNextButtonBotonerClicked	2ms
shouldStoreReadAndMoveFirstWhenNextButtonBotonerClicked	2ms
UtilTest	2ms
shouldReturnValidReadWhenSetOCRString	2ms

Figura 59: Resultat de les proves automàtiques no instrumentades

3.4 Criteris d'optimització presos durant la implementació

Durant la implementació s'han pres una sèrie de mesures per tal d'optimitzar l'aplicació en diversos àmbits. Aquestes optimitzacions són:

- Tindre una base de dades local per mantenir informació de rutes de lectura d'abonats. Amb aquesta opció el que s'aconsegueix es tindre menys connexions al servidor estalviant ample de banda i temps de resposta de l'aplicació. A més, si un lector estigués en una zona sense cobertura es podria treballar igualment amb l'aplicació prenent lectures.
- A l'hora de pujar lectures, incidències i fotos de les incidències al servidor, s'ha pres la decisió de comprimir tota aquesta informació en un únic fitxer i pujar-ho d'un cop. El que s'aconsegueix mitjançant l'aplicació d'aquesta mesura és estalviar ample de banda i temps en la pujada de la informació.
- Pel que fa a la racionalització de l'espai, en finalitzar l'exportació de lectures s'han esborrat les fotografies de les avaries que s'han exportat.
- Per accedir de manera més ràpida a la informació emmagatzemada en la base de dades local s'ha creat una estructura auxiliar que actua com una memòria cau entre la base de dades i l'aplicació. Amb aquest punt s'augmenta el rendiment de l'aplicació.
- S'ha tingut la cura de què quan s'activa el GPS per posicionar una nova avaria només estiga actiu el temps necessari per tal d'estalviar bateria. A més també s'ha pausat quan l'activitat on es realitzen aquestes tasques entra en segon pla. Quan sortim d'aquesta pantalla s'ha aturat el GPS per a què no estiga actiu.
- Les fotografies de les incidències s'emmagatzemen en un directori temporal en compte d'inserir-les en la base de dades. En la base de dades només està el nom de la fotografia. D'aquesta manera la base de dades és més ràpida i ocupa menys espai físic.

3.5 Revisió planificació en l'entrega de la implementació

En aquesta fase s'ha complit la planificació que s'havia proposat inicialment arribant a cobrir el projecte amb les expectatives de qualitat i funcionalitat esperades.

4. Conclusions

En aquest treball s'han aprés diversos temes. En primer lloc s'ha vist la importància de fer una bona planificació. D'ella depèn en gran mesura l'èxit o el fracàs del projecte. Una mala previsió pot dur a què el projecte tarde més temps del previst en ser desenvolupat i ens durà a un augment del cost econòmic i temporal. La qual cosa comportarà una insatisfacció del client.

Després s'ha vist l'avantatge d'usar un patró arquitectònic per construir interfícies d'usuari com el model vista presentador. Mitjançant aquest s'ha pogut usar una mateixa vista amb diversos presentadors. A més es podrà aprofitar part del codi per a fer la integració amb diverses aplicacions de lectura de comptadors de les comunitats de regants. També ha facilitat la confecció de proves automàtiques.

Un altre assumpte aprés ha estat la transcendència del prototip. Usar una bona ferramenta de prototipat dalt nivell portarà a què el desenvolupament del producte siga molt fidel a l'esperat. A més de servir de plataforma inicial d'avaluació per veure si el que ha transmet l'usuari final és el que se li ha presentat.

Com a punt final d'aprenentatge del projecte indicar que els coixins de temps per a cobrir imprevistos són molt necessaris perquè en tot projecte van a sorgir contratemps. Aleshores és una bona pràctica tindre'ls presents a l'hora de fer una planificació precisa. Com menys coneixements es tinguen de les tecnologies a emprar majors hauran de ser aquests coixins.

Pel que fa a l'assoliment dels objectius inicials comentar que s'han aconseguit tots. S'ha aconseguit implementar l'aplicació amb totes les funcionalitats esperades i superar de manera exitosa tots els riscos contemplats inicialment. A més també s'han realitzat les proves pertinents per donar-li qualitat i robustesa a l'aplicació creada. Part d'aquest triomf ha sigut fruit de l'encertada planificació inicial. En totes les fases s'han complit correctament les fites de temps i de material lliurable.

Referent a la metodologia emprada indicar que ens ha permès tindre una retroalimentació de l'usuari a partir del prototip i modificar aquelles coses que no s'havien interpretat correctament abans d'implementar l'aplicació.

Com a punts de millora per a l'aplicació citarem els següents:

- Creació automàtica de contactes dels abonats en el dispositiu mòbil. D'aquesta manera al compartir amb ells incidències no s'hauran de crear perquè ja existiran.
- Al prémer sobre el número de telèfon de l'abonat que l'aplicació truque automàticament al seu número.
- Calcular automàticament la ruta en un mapa des de la posició actual del lector fins al pròxim comptador.
- Millora de l'algorisme d'intel·ligència artificial per al reconeixement de textos en funció del tipus de comptador.
- Avisos de possible fuga o avaries en pantalla configurables segons paràmetres.
- Pantalla d'administració de paràmetres de l'aplicació.

5. Glossari

API: (Application Programming interface) Interfície de programació. Són un conjunt de subrutines, funcions i procediments, o mètodes si és programació orientada a objectes, que ofereixen una biblioteca per ser usat per un altre programari.
IDE: (Integrated development environment) Aplicació informàtica que proporciona al programador uns serveis integrals per desenvolupar programes.
Freelance: Persona que es dedica a realitzar treballs de manera autònoma o independent.
Lector: Persona que es dedica a anar llegint els comptadors de les comunitats de regants.
Abonat: Persona que té un camp amb un comptador de reg que pertany a una comunitat de regants.
OCR: (Optical Character Recognition) Es tracta d'un procés que permet identificar automàticament en una imatge els textos que existeixen i obtenir-los en forma de dades.
UML: (Unified Modeling Language) Llenguatge de propòsit general per al modelat orientat a objectes.
JSON: format de text lleuger per l'intercanvi de dades.
CPD: (Centre de processament de dades) Espai on es concentren el recursos necessaris per al processament d'informació. Normalment són espais amb bona connexió a internet, temperatura constant i adequada per a les màquines i sistemes de tolerància a fallades (elèctric, d'internet, etc.)
VPS: (Servidor Virtual Privat) És un mètode de partir un servidor físic en diversos servidors de tal manera que tot funciona com si estigués executant-se en una única màquina.
ORM: (Object-relational mapping) Tècnica que permet interactuar amb una base de dades relacional convertint els seus elements en objectes i així poder emprar-los en llenguatges orientats a objectes.
Bitmap: format d'imatge mitjançant el qual es representa aquesta amb un mapa de bits. Cada píxel és representat per un nombre determinat de bits.
Firebase Test Lab: infraestructura basada en el núvol per provar aplicacions d'Android.

6. Bibliografia

[1] Wikipedia. Visitat l'1 de març de 2018.
https://es.wikipedia.org/wiki/Sector_primario.
[2] Article revista "Tecnologia del agua". Autor: Carmen Trillo San José. Abril 2006. Pàgina 271 de la revista. Pàgina 5 de l'article. Visitat l'1 de març de 2018.
http://www.ugr.es/~ctrillo/Revista Tecnologia Agua.pdf
[3] Legado andalusí: Las aportaciones árabes a nuestra cultura. Visitat el 1 de març de 2018.
http://www.balansiya.com/herencia_regadio.html
[4] Viquipèdia. Regadiu. Visitat l'1 de març de 2018.
https://ca.wikipedia.org/wiki/Regadiu
[5] L'economia de la Comunitat Valenciana: Diagnòstic estratègic. CaixaBank Research. Pàgines 147-148. Visitat l'1 de març de 2018.
http://www.caixabankresearch.com/sites/default/files/documents/valencia valencianoweb.pdf
[6] Diari La Veu. El cultiu de regadiu supera el de secà per segon any consecutiu. Data de publicació 8 de gener de 2018. Visitat el 2 de març de 2018.
https://www.diarilaveu.com/noticia/78948/regadiu-supera-al-de-seca
[7] El regadiu a la Comunitat Valenciana. Servei de Documentació, Publicacions i Estadística Departamental. Desembre 2017. Pàgina 1 i pàgina 6. Visitat el 2 de març de 2018.
http://www.agroambient.gva.es/documents/162218839/164381878/El+Regadiu+a+la+Comunitat+Valenciana/69556fe6-2795-4c8f-87ba-62e756d21d6a
[8] GIA. Lectura de contadores. Visitat el 4 de març de 2018.
https://play.google.com/store/apps/details?id=com.clmtec.lectoraguasgia&hl=es
http://www.clmtec.es/lector-de-contadores/
[9] Galileo Ingenieria y Servicios. Visitat el 4 de març de 2018.
http://www.galileoiys.es/
https://apkpure.com/lectura-de-contadores/com.phonegap.LecturaContadores
[10] P. Letalier. Proceso de desarrollo de software.
Departamento de sistemas informáticos y computación.
Universitat Politècnica de València, 2003.
[11] INTECO. Ingeniería del software: metodologías y ciclos de vida. Laboratorio Nacional de Calidad del Software de INTECO, 2009.
[12] Native, web or hybrid Apps? What's the difference?. Visitat el 5 de març de 2018.
https://www.mobiloud.com/blog/native-web-or-hybrid-apps/
[13] A guide to mobile App development: web vs native vs hybrid. Visitat el 5 de març de 2018.
https://www.mobiloud.com/blog/native-web-or-hybrid-apps/
[14] ¿App nativa, web o híbrida? Visitat el 5 de març de 2018.
http://www.raona.com/aplicacion-nativa-web-hibrida/
[15] Native vs Hybrid apps. What to choose in 2017? Visitat el 5 de març de 2018.

https://blog.techmagic.co/native-vs-hybrid-apps/
[16] Garner says worlwide sales of smartphones grew 7 percent in the fourth quarter of 2016. Visitat el 6 de març de 2018.
https://www.gartner.com/newsroom/id/3609817
[17] Los mejores móviles en relación calidad-precio. Visitat el 9 de març de 2018.
https://elpais.com/elpais/2017/08/18/escaparate/1503032496_023139.html
[18] Mejores móviles en relación calidad-precio: 19 smartphones muy buenos desde 149 euros. Visitat el 9 de març de 2018.
https://www.xataka.com/moviles/mejores-moviles-en-relacion-calidad-precio-17-smartphones-muy-buenos-desde-149-euros
[19] Los mejores móviles relación calidad precio de 2017. Visitat el 7 de març de 2018.
https://computerhoy.com/listas/moviles/mejores-moviles-relacion-calidad-precio-2017-71513?page=3
[20] Comercial TPV. PDA Robusta protecció aigua i pols IP67 amb Android. Visitat el 8 de març de 2018.
https://www.comercialtpv.com/comandera-hosteleria-mustek-mk6000s.html?gclid=EAIaIQobChMIw4b0tfXf2QIV7LDtCh2NZw3TEAQYAiABEgL6sfD_BwE
[21] ¿Cuánto cuesta desarrollar una app? Visitat l'11 de març de 2018.
http://www.expansion.com/economia-digital/innovacion/2017/05/09/59104685ca4741700b8b459e.html
[22] Encuesta sobre superfícies y rendimientos de cultivo (esyrce). Comunidad Valenciana 2017. Pàgina 13.
http://www.agroambient.gva.es/documents/162218839/164381878/Encuesta+de+Superficies+y+Rendimientos+de+Cultivos+de+la+Comunitat+Valenciana+2017/88251ab5-76e7-4326-9fb8-7b4c88ca45e0
[23] "Secciones de cultivo en la Comunidad Valenciana" 10/03/2008. Diapositiva 4. Visitat el 12 de març de 2018.
http://www.agro-alimentarias.coop/ficheros/doc/02163.pdf
[24] Federació de Comunitats de regants de la Comunitat Valenciana. Visitat l'11 de març de 2018.
http://www.fecoreva.es/entidades/
[25] Las comunidades de regantes en La Safor. Gestión y uso del agua de regadío.
https://www.chj.es/es-es/ciudadano/libros/Captulos/Los riegos de la Safor y la Valldigna/4.CAPÍTULO IV.pdf
[26] DB-Engines Ranking - Trend of Couchbase vs. Oracle Berkeley DB vs. Realm vs. SQLite Popularity.
https://db-engines.com/en/ranking_trend/system/Couchbase%3BOracle+Berkeley+DB%3BRealm%3BSQLite

7. Annexos

7.1 Qüestionari d'avaluació del prototip

Després de navegar pel prototip emplena la següent informació:

1) Has sabut entrar en l'aplicació? SI NO

   Punts de millora _____

2) Has sabut carregar una ruta? SI NO

   Punts de millora _____

3) Has sabut introduir una lectura? SI NO

   Punts de millora _____

4) Has pogut navegar pels abonats dins d'una ruta de lectura? SI NO

   Punts de millora _____

5) Has sabut exportar les lectures? SI NO

   Punts de millora _____

6) Has sabut introduir una incidència d'un abonat? SI NO

   Punts de millora _____

7) Has sabut on associar una foto d'una incidència? SI NO

   Punts de millora _____

8) Has sabut com compartir la informació d'una incidència? SI NO

   Punts de millora _____

9) Has sabut com introduir la informació d'una avaria de la xarxa d'abastiment? SI NO

   Punts de millora _____

10) Has sabut com enviar una avaria? SI NO

    Punts de millora _____

11) Has sabut com treure la localització d'una avaria? SI NO

    Punts de millora _____

12) Has sabut com determinar la ubicació d'un comptador en el mapa?
SI NO

Punts de millora _____

13) Has sabut trobar la informació d'un abonat concret? SI NO

Punts de millora _____

14) Has trobat la informació que necessites d'un abonat en la fitxa de lectura? SI NO

Punts de millora _____

15) Altres comentaris:

7.2 Manual d'instal·lació

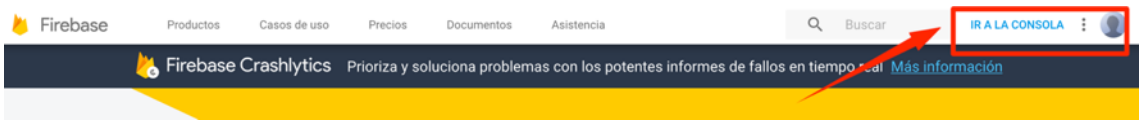
L'aplicació per instal·lar-la només cal executar l'apk que ve amb el lliurament.

Per a compilar l'aplicació cal descomprimir el fitxer de codi font i després des d'Android Studio obrir un projecte triant aquest directori.

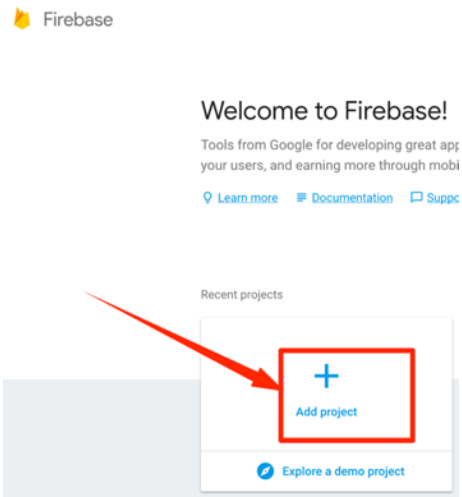
Per a la preparació de la part servidora cal donar d'alta en Firebase la base de dades de rutes, els usuaris per a entrar en l'aplicació i una part per pujar les lectures (i fotografies d'incidències) una vegada s'han enregistrat.

7.2.1 Alta del projecte en Firebase

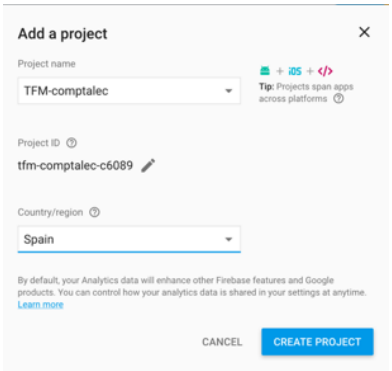
En aquest pas s'haurà de tindre un compte donat d'alta dins de Firebase. Una vegada validats en Firebase s'haurà d'anar a la part superior dreta i clicar en "Ir a la consola".



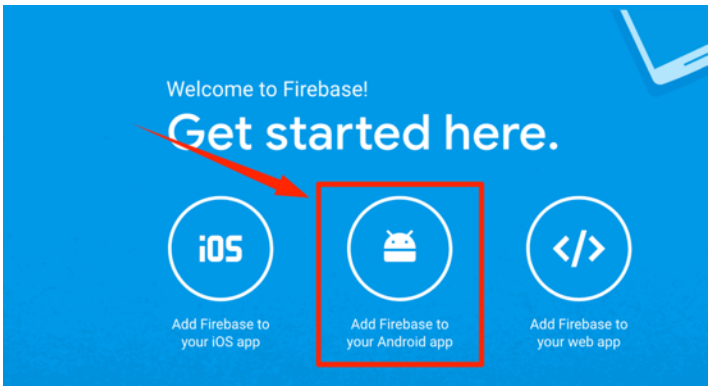
Ara en la consola s'haurà de triar "Afegir projecte"



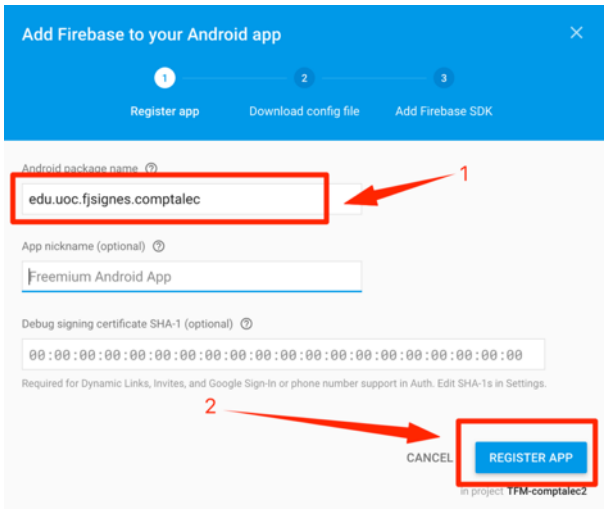
Acte seguit s'emplenarà la informació del projecte: nom i país.



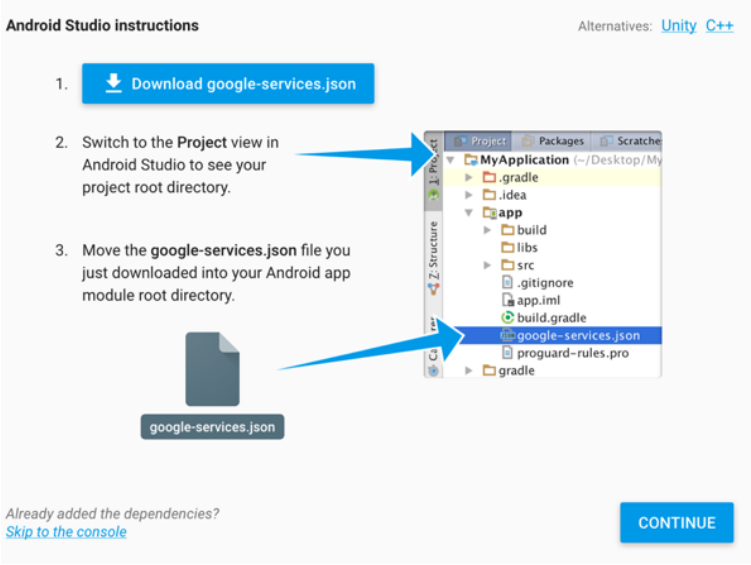
Després caldrà configurar l'aplicació per a què puga accedir a aquest projecte. Per això cal triar l'opció "Add Firebase to your Android app".



Seguidament cal emplenar el nom del paquet de l'aplicació i prémer el botó "Register APP":

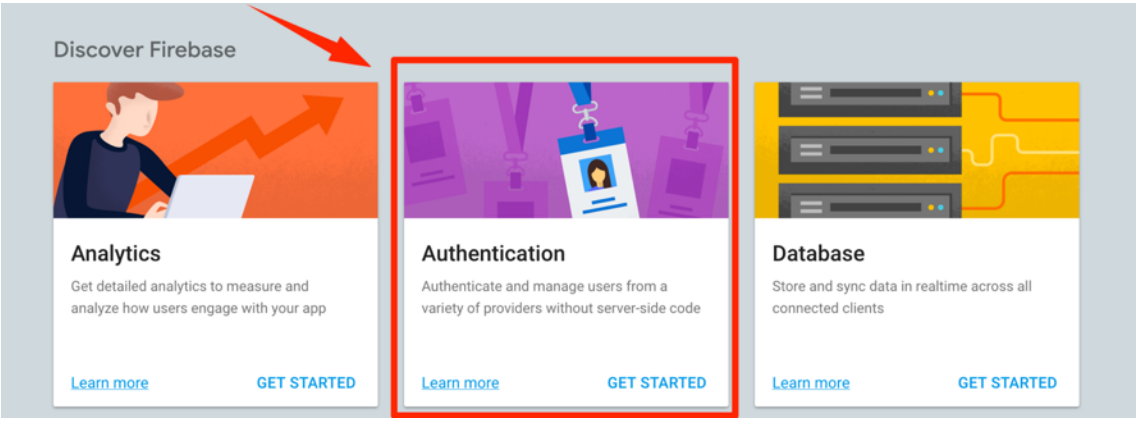


A continuació s'ha de descarregar un fitxer anomenat google-services.json i posar-lo a nivell de les fonts del projecte:

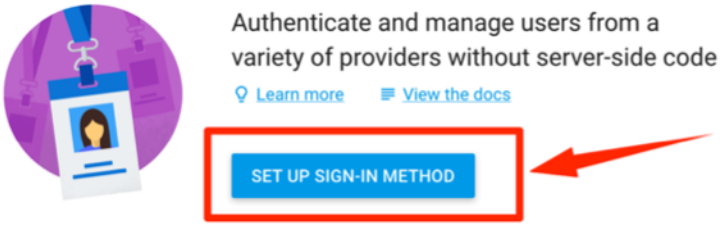


7.2.2 Alta dels usuaris per a validar-se en l'aplicació

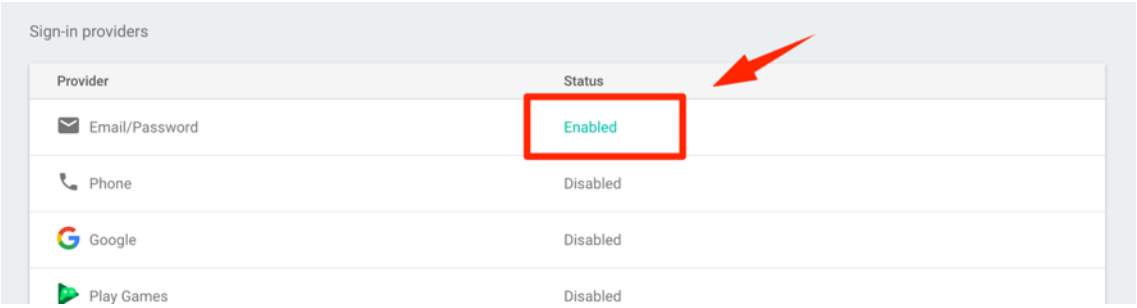
Per donar d'alta usuaris per a què puguen entrar a la nostra aplicació cal en primer lloc anar al projecte que s'acaba de donar d'alta en el pas anterior i prémer el botó anomenat "Authentication".



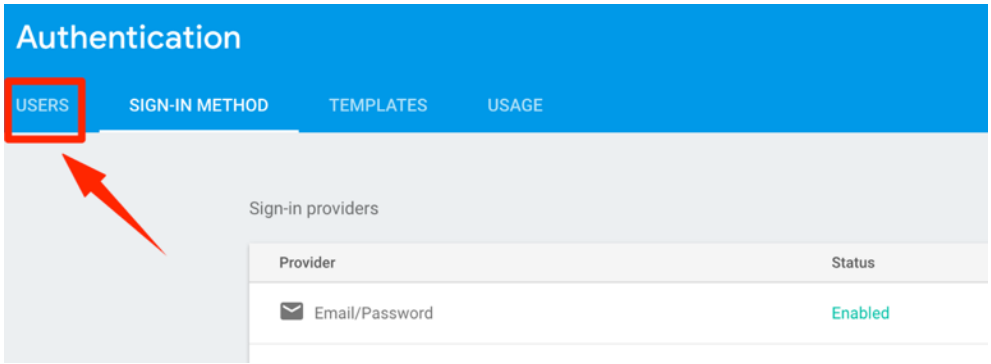
Després cal configurar el mètode de validació mitjançant el botó "Set up sign-in method"



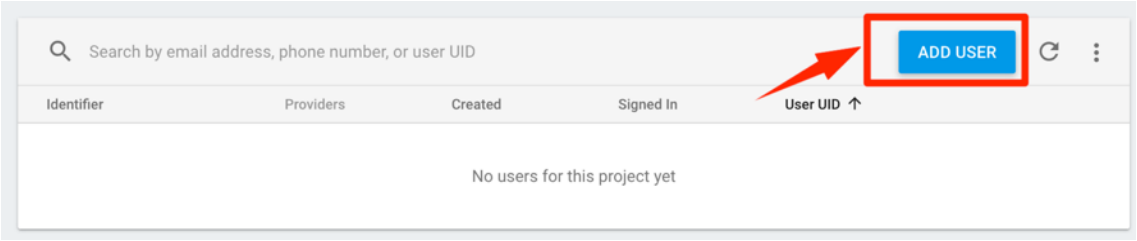
En aquest cas serà per correu electrònic. S'activarà, de totes les opcions, la de correu electrònic.



Ara cal donar d'alta els usuaris amb l'opció "Users"

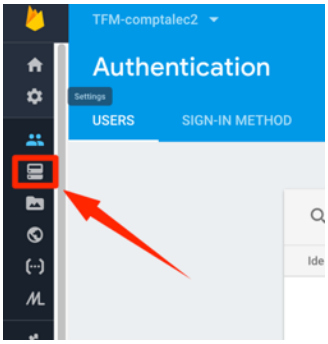


I afegir-los amb el botó “Add users”

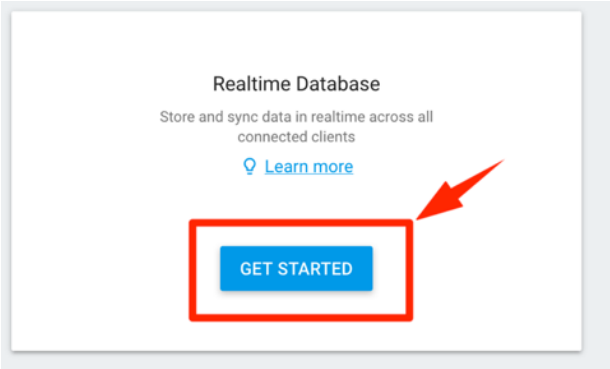


7.2.3 Alta de fitxer de rutes

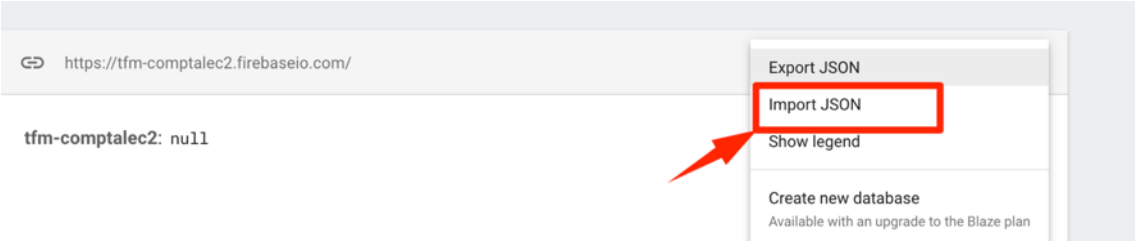
Per a pujar un fitxer de rutes de lectura cal anar a la consola i s’ha de triar dins del projecte el botó de base de dades.



Ara caldrà crear una base de dades nova de tipus “Realtime Database”.

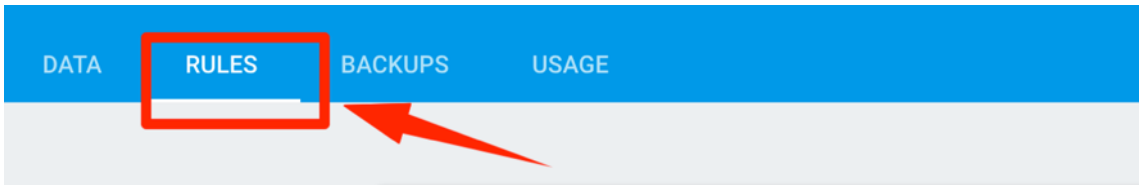


Una vegada creada s’haurà de pujar el fitxer ruta_lectura.js que està amb el codi del projecte.



A més també caldrà importar els fitxers tipus_incidencia.json i tipus_avaria.json que contenen els tipus d'incidències possibles i els tipus d'avaries.

Dins de l'apartat "rules" de la base de dades s'hauran de donar permisos de lectura per al usuaris autoritzats en el sistema.

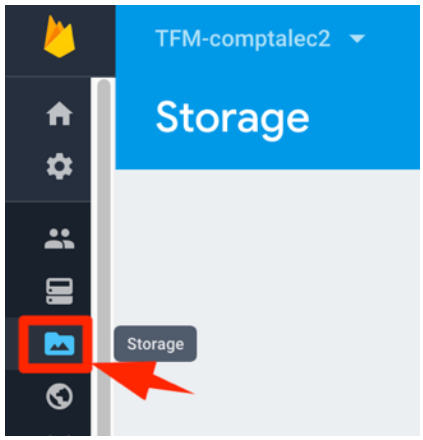


I es posarà:

```
//These rules require authentication
{
"rules": {
".read": "auth != null",
".write": "auth != null"
}
}
```

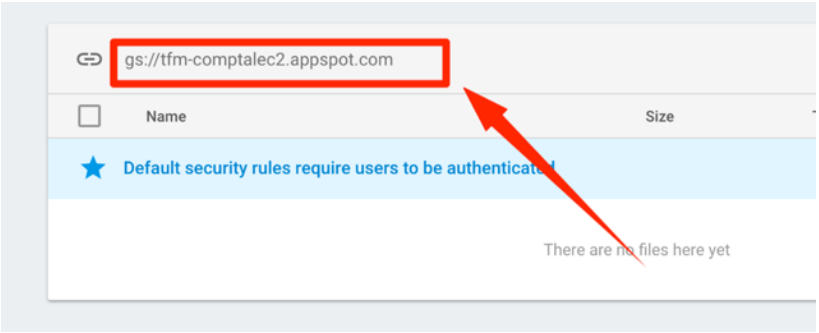
7.2.4 Alta espai d'emmagatzematge per pujar les lectures de la ruta i les fotos de les incidències.

Per abordar aquest apartat cal anar al projecte de Firebase i prémer la icona lateral que fa referència a "Storage" de Firebase.



I s'afegirà un nou espai d'emmagatzematge.

Cal fixar-se amb el nom que ens donarà el sistema perquè aquest nom es posarà dins del fitxer del projecte util/Constants.java en el valor anomenat URI_EXPORT_READS.



En aquest cas s'hauria de posar gs://tfm-comptalec2.appspot.com

7.2.5 Alta clau per a poder accedir a Google Maps

Com que l'aplicació accedeix a Google Maps per tal de mostrar la ubicació d'un comptador en el mapa, cal donar d'alta una clau per poder accedir a aquest servei de Google. L'aplicació ja té donada inicialment una clau, però si calgués donar-ne una altra per alguna raó, s'hauria de seguir el següent tutorial:

https://developers.google.com/maps/documentation/android-api/map-with-marker?hl=es

7.3 Manual d'usuari

En aquest manual s'indicarà com dur a terme les principals funcionalitats desenvolupades en l'aplicació.

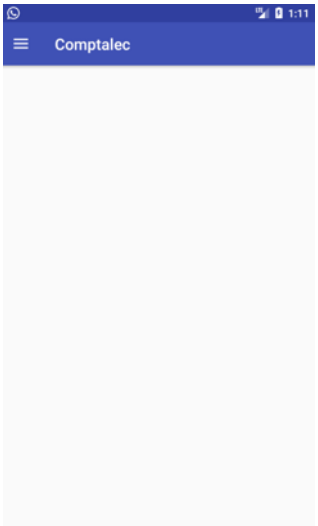
a) Validació

Per poder accedir a l'aplicació és necessari accedir amb un usuari i una contrasenya. Aquests els hauran de donar d'alta les comunitats de regants en la part servidora.

Pantalla de validació

b) Pantalla principal

En validar-se correctament es té la pantalla principal. La primera vegada aquesta pantalla només conté un menú superior perquè no es té cap ruta de lectura.



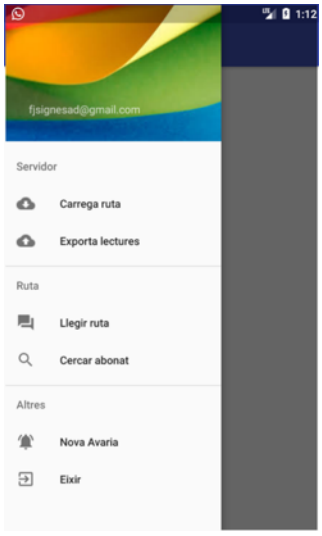
Pantalla inicial sense cap ruta carregada

Pantalla inicial amb una ruta carregada

Per altra banda també pot obrir-se amb l'últim abonat de la ruta on s'havia quedat el lector abans de tancar l'aplicació. Aquesta situació es produirà quan hi haja una ruta carregada prèviament.

c) Menú principal

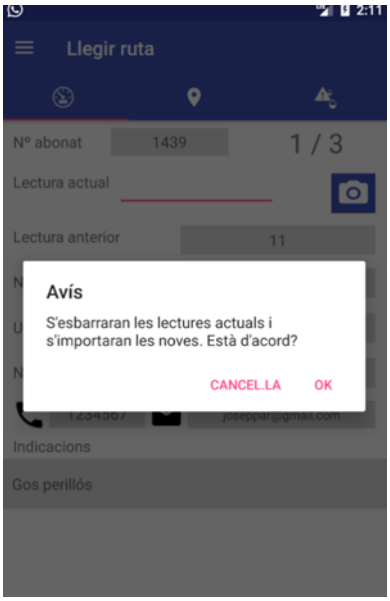
L'aplicació conté en tot moment en la part superior esquerra un menú que s'obri en prémer sobre el botó . Mitjançant aquest menú es podran executar totes les funcionalitats que ens ofereix el programa.



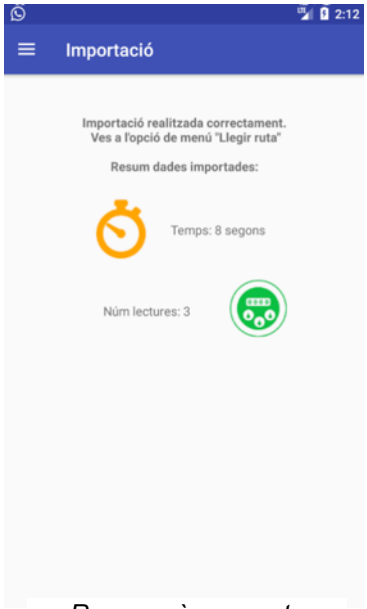
Menú de l'aplicació

d) Carrega ruta

Aquesta funcionalitat permet descarregar des del servidor la ruta de lectura que haja pujat la comunitat de regants. Cal matisar que en descarregar-se la informació del servidor s'esborrarà qualsevol informació de rutes anteriors que tinga l'aplicació emmagatzemada en local.



Avís d'importació

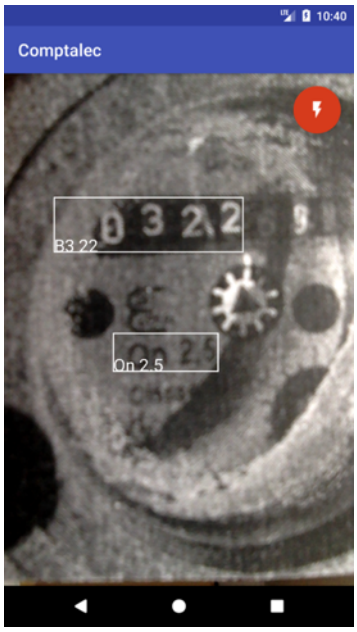


Resum càrrega ruta

e) Llegir lectures

Aquesta és la principal funcionalitat de l'aplicació. Permet introduir lectures dels abonats de la ruta així com incidències que el lector puga detectar en qualsevol abonat. La pantalla principal té un camp denominat "lectura actual" que permet la introducció de lectures.

Pantalla d'introducció de lectura



Pantalla d'introducció de lectura amb OCR

També es permet la lectura automàtica amb reconeixement d'imatges. Per a dur a terme aquesta funció s'ha de prémer el botó que té el símbol d'una càmera de fotos . El sistema obrirà la càmera del mòbil i en la imatge del comptador que es mostra, detecta tots els camps de text posant-los amb un rectangle blanc al voltant i el text que reconeix en aquesta part de la imatge. En prémer sobre qualsevol requadre l'aplicació porta aquest valor al camp de lectura actual de la pantalla d'introducció de lectures. Cal indicar que l'aplicació usa un algorisme per tal de filtrar contingut que no siga numèric o per solucionar qualsevol confusió de caràcters.

Per desar la informació de la lectura s'ha de prémer al damunt de qualsevol botó situat en la botonera vermella de la part inferior.

Botonera de l'aplicació amb diversos botons actius

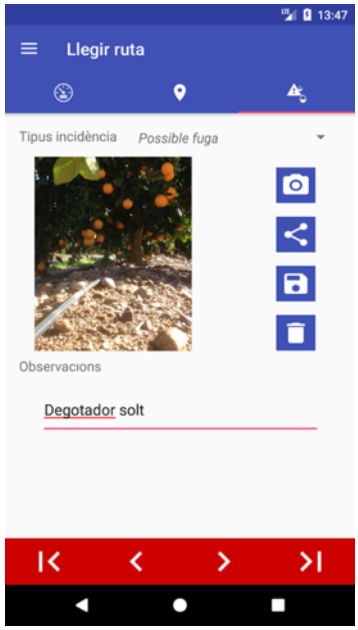
Aquesta botonera permetrà també el desplaçament pels abonats d'una determinada ruta seguint l'ordre de lectura més òptim. Es mostren els

botons davant, enrere, primer i últim segons l'abonat de la ruta on ens trobem.
En aquesta pantalla de lectura també es pot consultar la ubicació del comptador en un mapa quan es prem sobre la pestanya que té el símbol d'ubicació .

Ubicació d'un comptador en el plànol

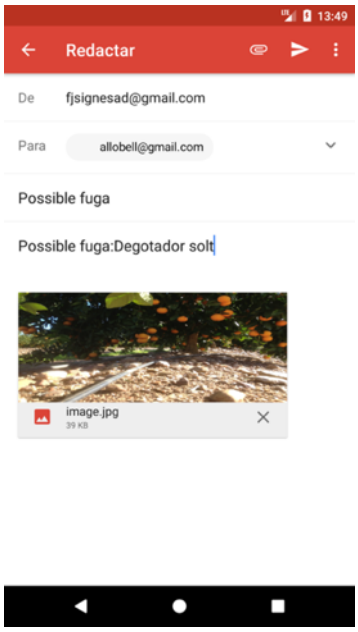
Finalment en la pantalla de lectura també es pot introduir una incidència que es detecte en aquest abonat. Per exemple, si el lector detecta una possible fuga la podrà enregistrar i compartir-la amb l'abonat.

Per enregistrar la incidència s'ha de triar la pestanya que té la icona .

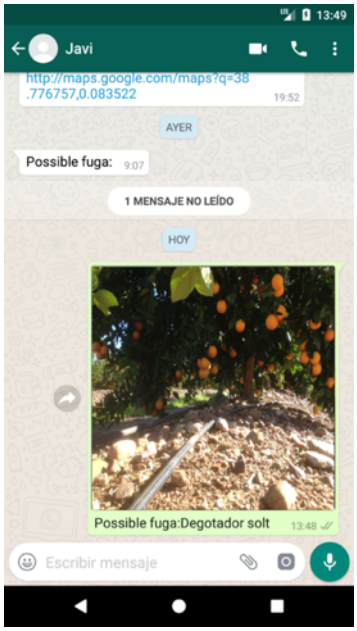


Introducció d'incidències

La informació a introduir serà el tipus d'incidència (triant-se d'un desplegable), un text d'observacions i una fotografia. Després aquesta informació es podrà compartir amb l'abonat mitjançant diverses aplicacions com ara Whatsapp o correu electrònic. S'aconsegueix prement el botó de compartir .



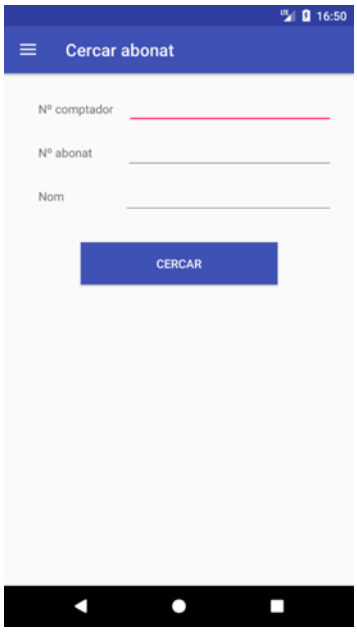
Compartir incidència amb un abonat a través del correu electrònic



Compartir incidència amb un abonat a través de Whatsapp

f) Cercar abonats

De vegades pot ser interessant anar a un determinat abonat de la ruta que no se sap en quin ordre està situat dins de la ruta. Per resoldre aquesta necessitat es té l'opció de cercar abonats segons uns criteris.

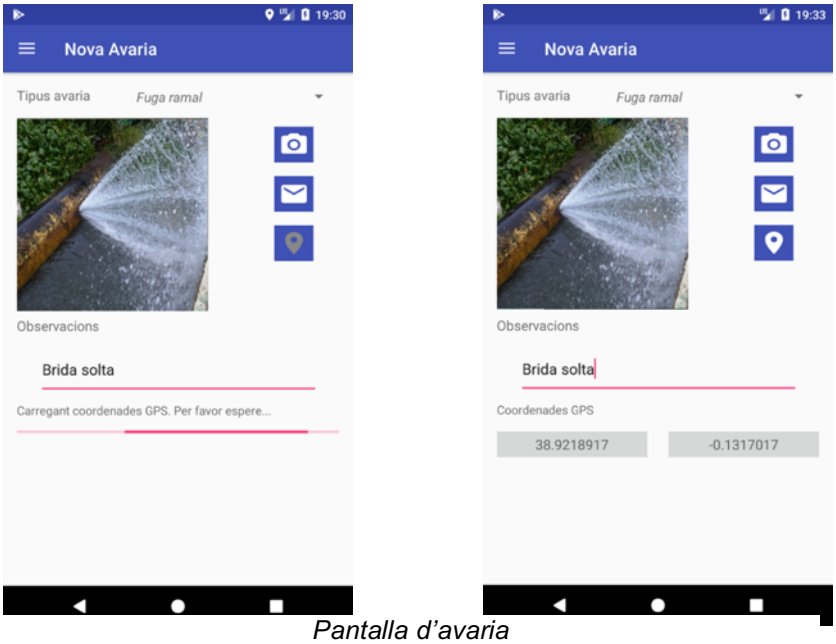


Pantalles de cerca, resultat de cerca i detall d'un abonat de la cerca

A l'hora de treure resultats, segons els criteris de cerca, l'aplicació el que fa és mostrar tots els abonats que continguen parcialment el que s'ha introduït en els camps de cerca. Si hi ha diversos camps de cerca amb valors, el que farà es que el resultat continga aquests valors parcialment en tots el camps de la base de dades que coincideisquen amb els criteris de cerca. És a dir es farà un "Y" lògic.
En treure el conjunt de resultats es podrà seleccionar un i anar al detall per veure informació, introduir lectures o incidències de l'abonat concret.

g) Avaries

Aquesta funcionalitat és semblant a la d'incidència però en aquest cas li permetrà al lector informar a la comunitat de regants de manera immediata d'una avaria de la xarxa d'abastiment d'aigua.



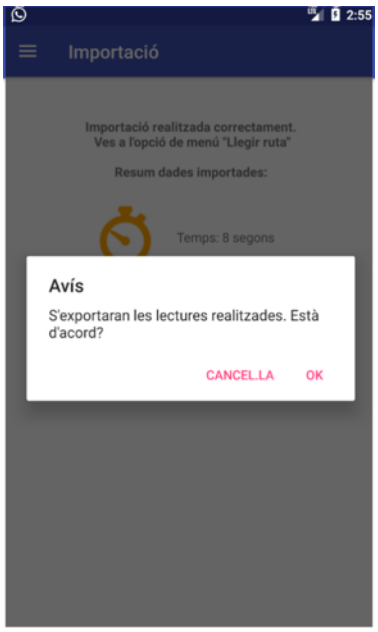
Pantalla d'avaria

Es pot triar el tipus d'avaria d'un conjunt, prendre una foto d'on està l'avaria i introduir una sèrie d'observacions. A més a més també es pot indicar les coordenades GPS exactes d'on es troba l'avaria per tal de facilitat a les brigades de manteniment que troben fàcilment la localització d'aquesta una vegada vagen a reparar-la. Per obtenir les coordenades GPS cal prémer el botó . Pot trigar una estona donat que l'aplicació treu la posició amb molta precisió.
Finalment es pot enviar aquesta informació per correu electrònic a l'oficina de la comunitat de regants. En el correu es posa en un enllaç la ubicació exacta al mapa.

Contingut del correu d'enviament d'una avaria

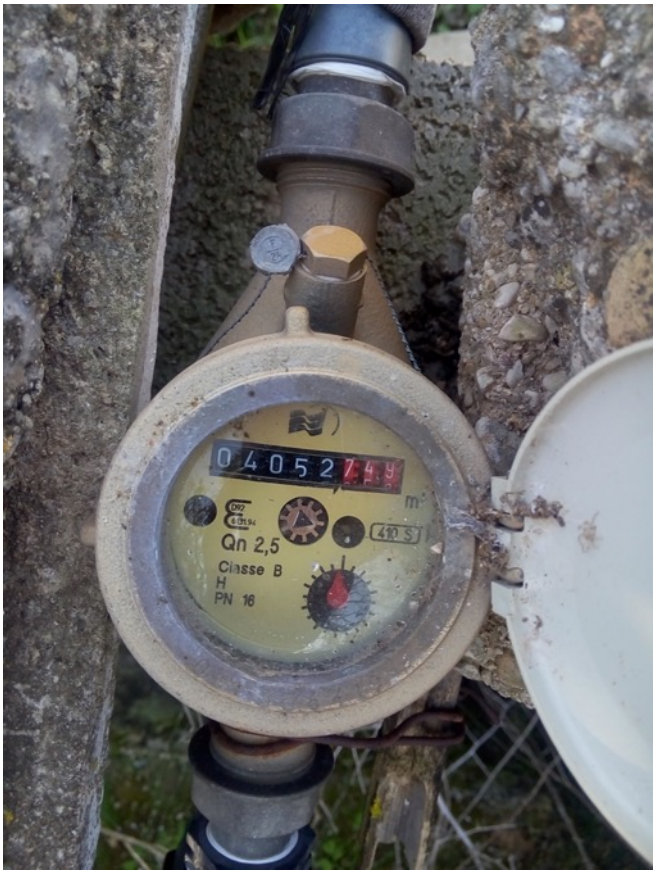
h) Exportar informació

Una vegada preses totes o algunes lectures i/o incidències (amb les fotos associades), el lector pot exportar-les a un servidor per tal de què l'oficina de la comunitat de regants puga descarregar-se-les a la seva aplicació. Cal indicar que els abonats que no tenen les lectures preses no s'exportaran al servidor. Així, es podrà continuar llegint i anotant les seves lectures després de l'exportació. En acabar l'exportació l'aplicació mostra un resum del que ha realitzat.



Pantalles mostrades durant l'exportació

i) Eixir

Si es vol que siga necessari introduir usuari i contrasenya per a tornar a entrar en l'aplicació cal donar-li a l'opció de menú anomenada "Eixir". D'aquesta maner en tornar a executar l'aplicació mostrarà la pantalla de validació. Si no s'eix des d'aquesta opció de menú sinó que es tanca directament l'aplicació, en tornar a executar-la el sistema entrarà sense demanar validació.

7.4 Fotos de comptadors per a fer proves d'OCR