

Gestió d'una Comunitat de Regants

Cristian González Roc

Grau d'Enginyeria Informàtica

Java EE

Consultor: Albert Grau Perisé

Professor responsable de l'assignatura: Santi Caballe Llobet

12/06/2019



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Gestió d'una Comunitat de Regants</i>
Nom de l'autor:	<i>Cristian González Roc</i>
Nom del consultor/a:	<i>Albert Grau Perisé</i>
Nom del PRA:	<i>Santi Caballe Llobet</i>
Data de lliurament (mm/aaaa):	<i>06/2019</i>
Titulació o programa:	<i>Grau Enginyeria Informàtica</i>
Àrea del Treball Final:	<i>Java EE</i>
Idioma del treball:	<i>Català</i>
Paraules clau	<i>JavaEE, Android, CRMB</i>
Resum del Treball (màxim 250 paraules): <i>Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball</i>	
<p>Aquest document és la memòria per al treball final de grau (TFG) Grau en Enginyeria Informàtica. A la memòria trobarem tota la informació de les diferents fases de planificació, anàlisi, disseny, implementació i arrancada del projecte.</p> <p>Per al desenvolupament del projecte s'ha utilitzat la tecnologia Java EE (Java Enterprise Edition) que es una plataforma Java per al desenvolupament d'aplicacions empresarials que consta de múltiples serveis, APIs i protocols que proporcionen les funcionalitats necessàries per desenvolupar aplicacions distribuïdes. S'ha utilitzat el patró Model Vista Controlador (MVC), per una banda és defineix els components per a la representació de la informació i per altra, la interacció de l'usuari.</p> <p>També s'ha creat una aplicació mòbil amb Android per a que els treballadors de la comunitat puguin fer les lectures dels comptadors de l'aigua i després poder pujar les dades recopilades al servidor web.</p> <p>El projecte és centra en una aplicació web per a dos Comunitats de Regants reals. L'objectiu a grans trets és la creació d'una aplicació web per a que els treballadors puguin gestionar la informació dels explotadors de les comunitats i l'aigua consumida per després poder facturar els consums i derrames dels seus terrenys. Els explotadors podran consultar a la web els seus consums d'aigua per mesos i consultar tot l'historial de consums enregistrats al sistema, d'altra banda, també podran consultar les derrames anuals dels terrenys que tenen en propietat i baixar-se-les en pdf.</p>	

Aquesta memòria intenta resumir el treball realitzat durant el desenvolupament del projecte, així com els coneixements i l'experiència adquirida durant la realització d'aquest.

Abstract (in English, 250 words or less):

This document is the memory for the TFG in Computer Engineering. In this memory you will find all the information about the different phases of planning, analysis, design, implementation and startup of the project.

To develop this project we used the technology JavaEE (Java Enterprise Edition) which is a platform for developing enterprise applications because it has multiple services, APIs and protocols that provides the needed functionalities to develop distributed applications.

The Model View Controller (MVC) model has been used in the project, by one side it defines the components for the representation of information and by the other side it defines the user interaction.

We also created an Android application so that community workers can read the water counters and then upload the data that they collect to the server.

The project focuses on a web application for two real irrigation communities. The main objective is the creation of a web application where the workers can manage the information of the exploiters of the communities and the water that they consume in order to be able to invoice the consumption and apportionments of the land that they own. The exploiters will be able to consult their monthly water consumption on the web and check the history of consumptions recorded in the system, also, they will be able to consult the annual apportionment of the land they own and download it with pdf.

This memory attempts to summarize the work carried out during the development of the project, as well as the knowledge and experience gained during its execution.

Índex

1	Introducció	1
1.1	Context i justificació del treball	1
1.2	Objectius del treball	1
1.3	Enfocament i mètode seguit	3
1.4	Planificació del treball	3
1.5	Breu sumari de productes obtinguts	6
2	Anàlisi	6
2.1	Casos d'ús i actors del servidor Java EE i Android	6
2.2	Fitxes de casos d'ús	7
2.2.1	Casos d'ús i JAVA EE	7
2.2.2	Casos d'ús Android	11
2.3	Model de pantalles (prototips)	13
2.3.1	Pantalles del servidor JAVA EE	13
2.3.2	Pantalles app Android	20
3	Disseny.....	24
3.1	Servidor Java EE	24
3.1.2	Disseny lògic de la base de dades	25
3.1.3	Normalització	27
3.2	Aplicació Android	28
3.2.1	Diagrama relaciona base de dades	28
3.2.2	Disseny lògic base de dades	28
3.2.3	Normalització	29
3.3	Base de dades històric	30
3.3.1	Diagrama relacional base de dades	30
3.3.2	Disseny lògic	30
3.3.3	Normalització	31
4	Implementació	31
4.1	Servidor Java EE.....	31
4.1.1	Diagrama components versió final	31
4.1.2	Clauscompostes JPAs	32
4.1.3	Figures API i frameworks	32

4.1.3.1 Generar PDFs derrama	32
4.1.3.2 Primecafes	32
4.1.3.3 JAX-RS	33
4.1.2 Mètode consums	34
4.2 Aplicació Android	41
4.2.1 APIs utilitzades	41
4.2.1.1 Room	41
4.2.1.2 Asyncchronous Http Client	42
4.2.1 Error conegut	42
4.3 Problemes principals desenvolupant l'aplicació	42
4.3.1 Problema Linux	42
5. Seguretat	43
5.1 JAAS	43
5.1.2 Login	44
5.1.3 Restricció de pàgines	44
5.1.4 Maneig de credencials d'usuari	44
5.2 SSL	44
5.3 Autenticació API rest	45
5.4 Còpies de seguretat	45
6 Futures millores	46
7 Conclusions	47
8 Glossari	48
9 Bibliografia	49

Figures

Figura 1 → Planificació	4
Figura 2 → Diagrama de Gantt	5
Figura 3 → Casos d'ús	6
Figura 4 → Pantalla regant	13
Figura 5 → Login regant	14
Figura 6 → Pantalla Última lectura	14
Figura 7 → Historial de consum	15
Figura 8 → Derrames	15
Figura 9 → Pantalles treballadors	16
Figura 10 → Pantalla terrenys	16
Figura 11 → Afegir terreny	17
Figura 12 → Afegir explotador/propietari	18
Figura 13 → Pantalla Tomes	19
Figura 14 → Afegir hidrant	19
Figura 15 → Tauler Consum aigua	19
Figura 16 → Login Android	20
Figura 17 → Elecció comunitat	20
Figura 18 → Tauler de Control Android	20
Figura 19 → Pantalla elecció hidrant toma	21
Figura 20 → Actualitzar base de dades servidor Java EE	21
Figura 21 → Pantalla Reparacions hidrant-toma	22
Figura 22 → Pantalla reparacions consum	22
Figura 23 → Introducció consum	23
Figura 24 → Diagrama relacional de la base de dades Servidor JavaEE	24
Figura 25 → Diagrama relacional de la base de dades Android	28
Figura 26 → Diagrama relacional de la base de dades històric	30
Figura 27 → Script copia	45
Figura 28 → Tasca programada per a la copia de seguretat	46

1. Introducció

1.1 Context i justificació del Treball

La Comunitat de Regants s'encarrega de captar aigua del riu Ebre i bombejar-la cap a unes bases a través d'unes tuberïes. La Comunitat reparteix l'aigua per als termes de dos poblacions Fabara i Nonasp, una vegada arriba l'aigua a aquestes bases, l'aigua es reparteix per pressió natural a totes les tuberïes amb ramificacions i més ramificacions de tuberïes més petites per abastir d'aigua a totes les parcel·les dels diferents explotadors. A les diferents parcel·les hi ha una caseta de formigó (hidrants) on hi ha una vàlvula per obrir o tancar el pas d'aigua a aquell ramal. Després cada explotador té la seva toma d'aigua on hi ha un comptador d'aigua amb una electro vàlvula que s'encarrega d'obrir o tancar el pas d'aigua cap a la seva parcel·la i comptar el pas d'aigua que està gastant cada regant.

Fins ara la forma que la comunitat tenia de recollir la lectura mensual dels comptadors d'aigua i gestionar-la es bastant rudimentària, s'ha d'anar una vegada mensualment per cada comptador de cada parcel·la i apuntar el consum a una llibreta, després s'exporta la informació de tots els comptadors a una taula d'excel, s'agafen els consums llegits a cada regant i se l'hi fa un sumatori de tots els comptadors que pugui tenir repartits per tota la comunitat i s'envia aquest excel a secretaria.

És un sistema bastant antiquat i és pot millorar bastant la forma d'enregistrar aquestes dades i gestionar-les per estalviar dies de feina als treballadors.

Amb aquest projecte és vol centralitzar totes les dades de les comunitats tant dels terrenys com dels consums d'aigua, que els treballadors i explotadors depenent el rol d'usuari que utilitzin puguin consultar o modificar les dades d'una forma intuïtiva i fàcil d'utilitzar.

1.2 Objectius del Treball

L'Objectiu d'aquest projecte és posar en pràctica els coneixements adquirits durant tot el grau i desenvolupar un projecte complet. Per dur a terme el projecte és treballarà amb els estàndards de l'enginyeria del programari, a banda, ampliar coneixements en l'ús de base de dades, Java, aplicacions web i Android. És vol aconseguir un producte que compleixi els requeriments de qualitat propis de l'enginyeria del programari.

Crear un servidor Java EE amb una base de dades PostgreSQL on es pugui gestionar la informació de cada explotador amb una interfície web per a consultar les seves dades, el seu consum d'aigua i els seus

terrenys. També incorporar una interfície d'administració on és pugui donar d'alta explotadors, propietaris, treballadors, ... i on també és pugui realitzar informes econòmics de facturació per a secretaria.

L'autenticació i l'apartat de seguretat del servidor s'implementarà mitjançant JAAS.

A banda, creació d'una aplicació mòbil Android per a que els treballadors puguin emmagatzemar a la base de dades la lectura dels comptadors de cada regant mensualment. L'aplicació haurà de funcionar sense connexió a Internet ja que a la majoria de llocs on es farà la lectura dels comptadors no hi ha cap tipus de connexió a Internet i que posteriorment és pugi bolcar les lectures dels comptadors a la base de dades una vegada és tingui connexió.

Requisits funcionals:

- Consultar, modificar o afegir consums d'aigua dels usuaris.
- Consultar, modificar o afegir hidrants a les comunitats.
- Consultar, modificar o afegir tomes d'aigua a les comunitats.
- Gestionar les possibles reparacions dels comptadors d'aigua.
- Permetre donar d'alta, modificar o eliminar usuaris al sistema.
- Afegir, consultar o modificar terrenys al sistema.
- Generar derrames al sistema.
- Consultar les derrames dels usuaris i baixar-les amb PDF.
- Generar informes dels consums d'aigua i de les derrames per crear els rebuts.

Requisits no funcionals:

- Desenvolupar l'aplicació per a tot tipus de dispositius: Smartphones, equips de taula, portàtils i tauletes.
- Crear una aplicació intuïtiva per als usuaris.
- Crear una aplicació eficient amb una resposta ràpida de les accions de l'usuari.
- Crear una aplicació segura per a l'usuari.

1.3 Enfocament i mètode seguit

Per a dur a terme el projecte s'ha elegit crear un producte totalment nou ja que fins ara l'empresa no utilitzava cap tipus d'aplicació d'aquest tipus.

Investigant per la xarxa he trobat productes per agafar idees però no he trobat cap projecte per a una comunitat de regants com el que vull realitzar, per tan, hem centrat en un projecte totalment nou.

Per a desenvolupar el projecte s'ha tingut en compte aspectes d'organització del temps disponible de treball i les diferents tasques a realitzar.

S'ha seguit la següent estratègia per dividir el projecte en diferents tasques:

- **Anàlisi del projecte** → Realitzar un petit estudi de l'empresa amb l'ajuda dels treballadors i secretaria per veure quina necessitat ha de cobrir l'aplicació. Després de definir els objectius s'han començat a plantejar les següents tasques.
- **Aprenentatge de les tecnologies a utilitzar** → Per poder planificar el projecte d'una forma realista, s'ha dissenyat una estratègia d'estudi de les tecnologies que no havia utilitzat mai, s'ha decidit dedicar unes hores a la setmana per aprendre Android ja que era una tecnologia nova per a mi. A banda, recerca de materials web de les APIs i frameworks a utilitzar.
- **Distribució dels lliuraments:**
 - **PAC 1** → Planificació del projecte – Lliurament 04-03
 - **PAC 2** → Anàlisi i disseny de l'arquitectura – Lliurament 08-04
 - **PAC 3** → Implementació – Lliurament 23-05
 - **PAC 4** → Memòria i presentació – Lliurament 12-06

Per la realització del projecte s'ha seguit la planificació de l'assignatura per als diferents lliuraments. Els terminis de les entregues son els adequats partint del temps que hi ha per al treball.

1.4 Planificació del Treball

S'ha organitzat el temps de treball en sessions diàries de tres hores mínim. En principi dedicaré al projecte de Dilluns a Dissabte per intentar seguir la planificació, si fos necessari també dedicaria temps els Diumenges.

He intentat ser el més realista possible amb la planificació del projecte, però és difícil ajustar-me a la realitat ja que es un projecte nou per a mi i dependrà una mica de les hores d'investigació i aprenentatge sobre les noves tecnologies que no he utilitzat mai.

Al següent esquema es pot veure la temporització resultant de les tasques i entregues del projecte:

Tarea

Nombre	Fecha de inicio	Fecha de fin
PAC1	20/02/19	4/03/19
Explicació dels requeriments de l'aplicació	20/02/19	24/02/19
Planificació del TFG	25/02/19	28/02/19
Explicació de les eines i frameworks	1/03/19	4/03/19
PAC2	5/03/19	8/04/19
Model de casos d'ús i actors	5/03/19	11/03/19
Fitxes de casos d'ús	12/03/19	18/03/19
Model de pantalles	19/03/19	21/03/19
Entrega parcial per al consultor	22/03/19	22/03/19
Disseny relacional de la base de dades	23/03/19	25/03/19
Diagrama de classes principals	26/03/19	28/03/19
Diagrama d'arquitectura	29/03/19	31/03/19
Investigació i aprenentatge de les tecnologies que s'utilitzaran	1/04/19	8/04/19
PAC3	9/04/19	23/05/19
Implementació del servidor	9/04/19	23/05/19
Cas d'ús 1 - Iteració 1	9/04/19	10/04/19
Cas d'ús 1 - Iteració 2	11/04/19	12/04/19
Cas d'ús 1 - Iteració 3	13/04/19	14/04/19
Cas d'ús 1 - Iteració 4	15/04/19	16/04/19
Implementació resta de casos d'ús	1/05/19	23/05/19
Implementació de l'aplicatiu Android	17/04/19	30/04/19
Cas d'ús 1 - Iteració 1	17/04/19	19/04/19
Cas d'ús 1 - Iteració 2	20/04/19	22/04/19
Cas d'ús 1 - Iteració 3	23/04/19	24/04/19
Cas d'ús 1 - Iteració 4	24/04/19	25/04/19
Implementació resta de casos d'ús	26/04/19	30/04/19
Entrega parcial per al consultor	1/05/19	1/05/19
Memòria i presentació	24/05/19	12/06/19
Redacció de la versió final de la memòria	24/05/19	8/06/19
Presentació	9/06/19	12/06/19

Figura 1 → Planificació

A banda de les entregues de cada PAC, he programat dos entregues parcials per fer un seguiment de les PACS amb el consultor.

A la següent figura es mostra el diagrama de Gantt resultant de la temporització de l'esquema anterior:

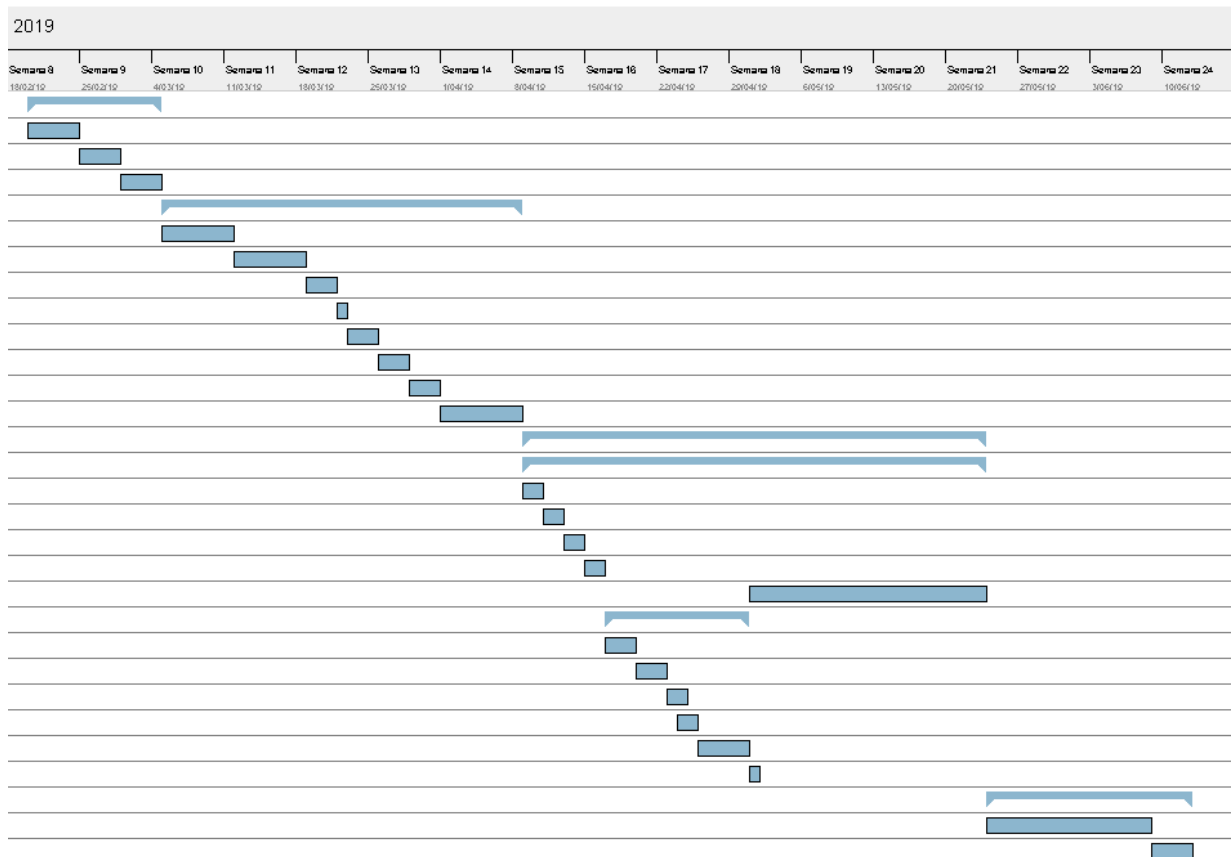


Figura 2 → Diagrama de Gantt

Eines i frameworks a utilitzar:

- **Eclipse**→S'utilitzarà com a editor per a desenvolupar l'aplicació Java EE.
- **Android Studio**→S'utilitzarà com a editor per a desenvolupar l'aplicació Android.
- **VirtualBox**→S'utilitzarà com a eina per la creació del servidor com a màquina virtual.
- **WildFly**→S'utilitzarà com a servidor Java EE per executar la part del servidor.
- **JAAS**→S'utilitzarà per la autenticació i restricció d'accés a l'aplicació web.
- **JSF**→S'utilitzarà per desenvolupar la interfície d'usuari del servidor.
- **PostgreSQL**→S'utilitzarà com a motor de base de dades.
- **Hibernate**→S'utilitzarà per a comunicar l'aplicació Java EE amb la base de dades de forma transparent.
- **JPA**→S'utilitzarà per a les tasques de persistència requerides en l'aplicació.
- **PrimeFaces** → Biblioteca de components per a JavaServer Faces (JSF) de codi obert que incorpora un conjunt de components enriquits que faciliten la creació d'aplicacions web.
- **OpenPDF** → Llibreria per a la creació i edició de PDFs.

1.5 Breu sumari de productes obtinguts

- **Aplicació web** → Arxius de l'aplicació web amb els fitxers jsp, fitxers de configuració xml, .jar de les APIs utilitzades i fulles d'estils CSS.
- **Aplicació Android** → APK de l'aplicació Android per als treballadors.
- **Memòria** → Document que sintetitza les diferents fases de desenvolupament del treball.
- **Presentació** → Presentació en Power Point amb els principals conceptes explicats a la memòria.
- **Vídeo** → Vídeo-presentació del projecte amb l'explicació del funcionament de l'aplicació web i Android.

2. Anàlisi

2.1 Casos d'ús i actors del servidor Java EE i Android

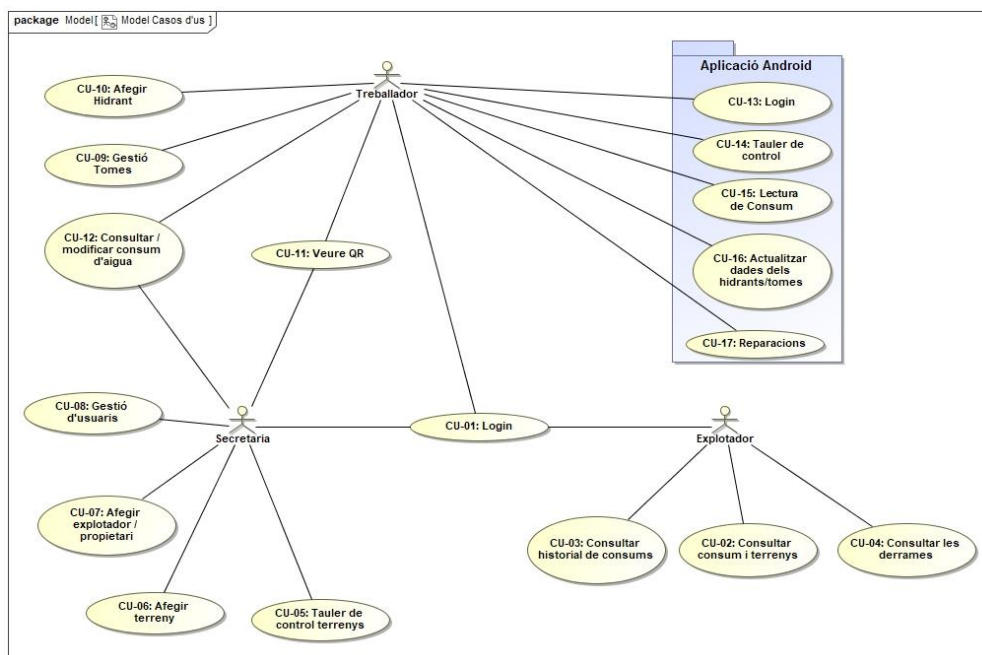


Figura 3 → Casos d'ús

Rols dels diferents usuaris:

Rol explotador → Seran els usuaris que només tindran accés al sistema per a consultar la informació de les seves finques i tomes d'aigua, no tindran permisos de modificació. Podran consultar els consums enregistrats al sistema i les derrames.

Rol secretaria → Secretaria tindrà accés de consulta i modificació a tot el sistema menys a l'aplicació Android. S'encarregarà de l'administració dels consums generats, derrames i gestió dels usuaris.

Rol treballador → Els treballadors tindran accés a l'aplicació Android per fer la lectura i reparació dels comptadors. També tindran accés al servidor Java EE per fer les consultes i modificacions necessàries dels consums d'aigua i publicar o despublicar la lectura dels comptadors per a que els exploradors ho puguin consultar.

El sistema només permetrà la creació d'usuaris explotadors al servidor web. Els usuaris treballadors seran inserits directament a la base de dades per l'administrador.

Qualsevol usuari que entri a la web de forma anònima no tindrà accés al sistema ja que només podran accedir els usuaris creats per l'administrador o secretaria.

2.2 Fitxes de casos d'ús

2.2.1 Casos d'ús Java EE

Identificador	CU-01
Nom	Login
Descripció	L'usuari vol identificar-se per entrar al sistema
Actors	Usuari explotador /Usuari treballador
Pre-Condicions	L'usuari ha d'existir
Escenari principal d'èxit	L'usuari introdueix el seu DNI i contrasenya. El sistema comprova si existeix l'usuari. En cas d'existir el sistema retorna l'usuari a la següent pantalla depenent del rol d'usuari que tingui. En cas de no existir el sistema mostrarà un missatge d'error.
Post-Condicions	Es guarden les credencials de l'usuari

Identificador	CU-02
Nom	Consultar el consum i terrenys
Descripció	Consultar les dades dels seus consums i terrenys
Actors	Usuari explotador
Pre-condicions	L'usuari explotador ha de tenir terrenys al seu nom
Escenari principal d'èxit	CU-01 Depenent la comunitat a la que pertany l'usuari, el sistema el redirigeix cap a la comunitat de Fabara o Nonasp. Si l'usuari pertany a les dues comunitats el sistema l'enviarà a la pantalla d'elecció de comunitat. El sistema mostra el consum de l'últim mes enregistrat al sistema desglossat per tomes i la informació dels seus terrenys.
Post-condicions	

Identificador	CU-03
Nom	Consultar l'historial de consums
Descripció	Consultar l'historial de consums desglossat per data de lectura
Actors	Usuari explotador
Pre-condicions	L'usuari explotador ha de tenir consums enregistrats
Escenari principal d'èxit	L'usuari escull l'opció consultar historial de consums. L'usuari escull una data a consultar. El sistema mostra el consum de la data seleccionada.
Post-condicions	

Identificador	CU-04
Nom	Consultar les derrames
Descripció	Consultar les derrames desglossades per data
Actors	Usuari explotador
Pre-condicions	L'usuari ha de tenir derrames enregistrades
Escenari principal d'èxit	El sistema mostra les derrames desglossades per data amb el cànon a pagar per hectàrea i l'import. L'usuari pot veure en detall les derrames dels terrenys desglossades per hidrant i toma.
Post-condicions	

Identificador	CU-05
Nom	Tauler de Control de Terrenys
Descripció	Consultar els terrenys dels explotadors
Actors	Usuari secretaria
Pre-condicions	Han d'existir terrenys al sistema
Escenari principal d'èxit	L'usuari escull al Tauler de Control l'opció de Terrenys. El sistema mostra la informació dels terrenys desglossats per: explotador, propietari, terme municipal, polígon, parcel·la, sub - parcel·la, superfície, data de concessió, hidrant i toma. L'usuari pot editar o esborrar un terreny.

Post-condicions	El terreny s'ha de modificar o esborrar correctament
-----------------	--

Identificador	CU-06
Nom	Afegir un terreny
Descripció	Afegir un terreny al sistema
Actors	Usuari secretaria
Pre-condicions	El terreny no ha d'estar introduït anteriorment al sistema
Escenari principal d'èxit	L'usuari escull l'opció "Terrenys/Afegir terreny". L'usuari introdueix les dades del terreny: Explotador, propietari, Terme municipal, polígon, parcel·la, sub – parcel·la, superfície, data concessió, hidrant, toma.
Post-condicions	El terreny s'ha d'afegir al sistema correctament.

Identificador	CU-07
Nom	Afegir explotador / propietari
Descripció	Afegir un nou explotador / propietari
Actors	Usuari secretaria
Pre-condicions	L'usuari no ha d'estar introduït anteriorment al sistema
Escenari principal d'èxit	L'usuari escull l'opció "Afegir explotador/propietari". L'usuari afegeix el tipus d'usuari que s'introduirà, depenent quin tipus d'usuari és seleccionerà el sistema mostrarà o ocultarà alguns camps. L'usuari introdueix les dades del nou explotador/propietari: Nom, DNI, primer cognom, segon cognom, contrasenya, direcció, població, CP, província, correu electrònic, telèfon, número de compte, entitat bancària, tipus adeudo, data i firma.
Post-condicions	S'ha d'afegir el nou usuari al sistema

Identificador	CU-08
Nom	Gestió d'usuaris
Descripció	Tauler de gestió d'usuaris
Actors	Usuari secretaria
Pre-condicions	Han d'existir usuaris al sistema
Escenari principal d'èxit	L'usuari elegeix al Tauler de Control l'opció de "Gestionar Usuaris". El sistema mostra la informació dels usuaris desglossada per: tipus usuaris, nom, primer cognom, segon cognom, DNI, direcció, població, CP, província, correu electrònic, email, telèfon, numero de compte, entitat bancària, tipus adeudo, data firma, contrasenya L'usuari pot modificar o esborrar un usuari.
Post-condicions	L'usuari s'ha de modificar o esborrar correctament

Identificador	CU-09
Nom	Gestió tomes
Descripció	Tauler de Control de la gestió de les tomes
Actors	Usuari treballador

Pre-condicions	Han d'existir terrenys introduïts al sistema
Escenari principal d'èxit	L'usuari elegeix al Tauler de Control l'opció "Tomes amb terreny". El sistema mostra la informació dels usuaris desglossada per: Explotador, terme municipal, polígon, parcel·la, sub parcel·la, superfície, data concessió, hidrant, toma. L'usuari pot modificar o esborrar una toma. L'usuari pot afegir un hidrant CU-10.
Post-condicions	L'usuari s'ha de modificar o esborrar correctament

Identificador	CU-10
Nom	Afegir hidrant
Descripció	Afegir un nou hidrant al sistema
Actors	Usuari treballador / Secretaria
Pre-condicions	L'hidrant nou no ha d'existir al sistema
Escenari principal d'èxit	L'usuari elegeix al Tauler de Hidrants "Afegir hidrant". L'usuari introdueix les dades de: Numero d'hidrant, coordenades.
Post-condicions	L'hidrant s'ha d'afegir correctament al sistema

Identificador	CU-12
Nom	Consultar/modificar consum d'aigua
Descripció	Consultar o modificar els consums d'aigua
Actors	Usuari treballador / Usuari secretaria
Pre-condicions	Han d'haver-hi consums introduïts al sistema
Escenari principal d'èxit	L'usuari elegeix al Tauler de Control l'opció "Consum aigua". El sistema mostra el consum d'aigua de l'última lectura desglossat per: data, hidrant, toma, explotador, consum, preu m ³ , merma, import L'usuari pot filtrar per data i/o explotador. L'usuari pot editar el consum d'un explotador o introduir una possible merma. L'usuari pot publicar o despublicar el consum d'un regant específicament. L'usuari pot publicar o despublicar el consum en una data de forma massiva.
Post-condicions	La informació s'ha de modificar correctament al sistema.

2.2.2 Casos d'ús Android

Identificador	CU-13
Nom	Login
Descripció	L'usuari vol identificar-se per entrar a l'aplicació
Actors	Usuari treballador
Pre-Condicions	L'usuari ha d'existir
Escenari principal d'èxit	L'usuari introdueix el seu DNI i contrasenya. El sistema comprova si existeix l'usuari. En cas d'existir el sistema retorna l'usuari a la següent pantalla depenent del rol d'usuari que tingui. En cas de no existir el sistema mostrarà un missatge d'error.
Post-Condicions	Es guarden les credencials de l'usuari

Identificador	CU-14
Nom	Tauler de control
Descripció	Tauler de control per elegir quina feina s'ha de fer
Actors	Usuari treballador
Pre-Condicions	
Escenari principal d'èxit	L'usuari escull quina feina ha de realitzar. L'usuari pot entrar al tauler de lectura de consums. L'usuari pot entrar al tauler de reparacions. L'usuari pot entrar al taulell d'actualitzar la base de dades.
Post-Condicions	

Identificador	CU-15
Nom	Lectura de consum
Descripció	Lectura dels comptadors en una data en concret
Actors	Usuari treballador
Pre-condicions	Les tomes a llegir han d'estar actualitzades a l'aplicació Android per evitar lectures de tomes que no existeixen o s'han modificat.
Escenari principal d'èxit	L'usuari selecciona una toma a llegir o escaneja el QR de la toma. L'usuari escriu amb el teclat numèric el consum d'aquella toma. El sistema actualitza a "llegit" el consum d'aquella toma. Una vegada llegides totes les tomes, l'usuari envia les dades de la base de dades de l'aplicació Android al servidor JavaEE.
Post-condicions	Les dades introduïdes a l'aplicació s'han d'enviar al servidor JavaEE correctament una vegada es tingui connexió a Internet.

Identificador	CU-16
Nom	Actualitzar dades dels hidrants/tomes
Descripció	Actualitzar les dades dels hidrants de la base de dades del servidor Java EE a l'aplicació Android per si hi ha hagut possibles modificacions dels hidrants
Actors	Usuari treballador
Pre-condicions	L'usuari ha de tenir connexió a Internet per realitzar l'actualització, és recomanable estar connectat per WIFI
Escenari principal d'èxit	L'usuari comprova l'última actualització de la base de dades, si hi han hagut modificacions posteriors, l'usuari fa clic al boto d'actualitzar la base de dades. L'aplicació Android demana el password d'actualització. L'aplicació Android descarrega la informació necessària dels hidrants i tomes del servidor Java EE. S'actualitza la base de dades de l'aplicació Android amb les noves modificacions dels hidrants.
Post-condicions	Actualitzar la base de dades de l'aplicació correctament

Identificador	CU-17
Nom	Reparacions
Descripció	Tauler de control per introduir una reparació
Actors	Usuari treballador
Pre-condicions	La base de dades de l'aplicació Android ha d'estar actualitzada, per possibles modificacions a les tomes.
Escenari principal d'èxit	L'usuari selecciona la toma que s'ha reparat. L'usuari introdueix el consum del comptador antic. L'usuari introdueix el consum del nou comptador. L'usuari una vegada tingui connexió, envia al servidor web la informació.
Post-condicions	

2.3 Model de pantalles (prototips)

2.3.1 Pantalles del servidor Java EE

Pantalles del regant

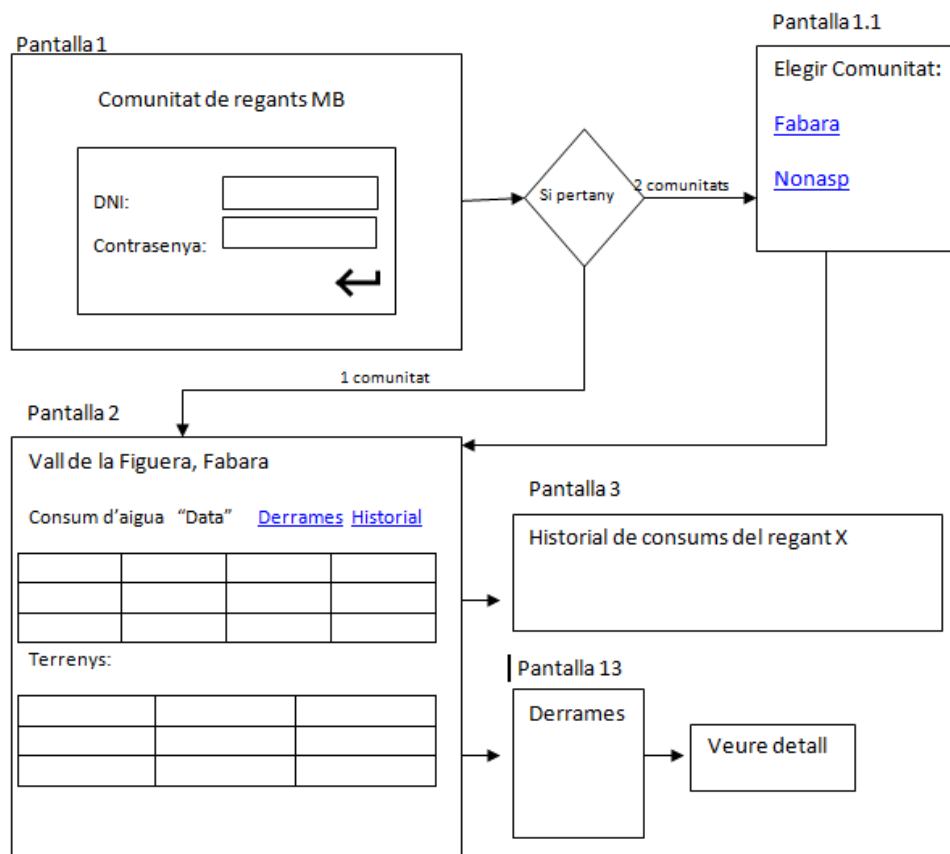


Figura 4 → Pantalla regant

Pantalla 1

Comunitat de regants MB

DNI:

Contrasenya:

Figura 5 → Login regant

Pantalla 2

Vall de la Figuera, Fabara

Usuari Jose Rios Lopez

[Derrames](#) [Historial de consums](#)

Consum d'aigua última lectura "02/01/2019 - 04/02-2019"

Hidrant	Toma	Consum m ³	Preu m ³	Preu €
12	0	78	0.1	7.8
111	2	114	0.1	11.4
168	1	1206	0.1	120.6


Terrenys amb regadiu




Explotador	Propietari	Terme municipal	Polígon	Parcel·la	Sub parcel·la	Superfície (Ha)	Hidrant
Jose Rios Lopez	Manolo Castillo Albiac	Fabara	22	155-c		0.6138	12
Jose Rios Lopez	Roberto Casado Llop	Fabara	20	12		1.9065	111
Jose Rios Lopez	Alberto Roc Raffles	Fabara	4	145-g	G	0.8275	168

Figura 6 → Pantalla Última lectura

Pantalla 3

Historial de consum

Lectura a mostrar: 


Usuari Jose Rios Lopez




Data lectura	Hidrant	Toma	Consum m ³	Preu m ³	Preu a pagar €
01/01/2019 – 02/02/2019	12	0	78	0.1 €	7.8
...
...
TOTAL			Σ		Σ

Figura 7 → Historial de consum

Pantalla 13

Derrames

Any a mostrar: 

Usuari Jose Rios Lopez

Exercici	Data	Canon €/ha	Import Total €	Veure detalls
2018	01/01/2018	50	2200	Link
2018	01/04/2018	50	1100	Link
...	Link

Figura 8 → Derrames

Pantalles treballador

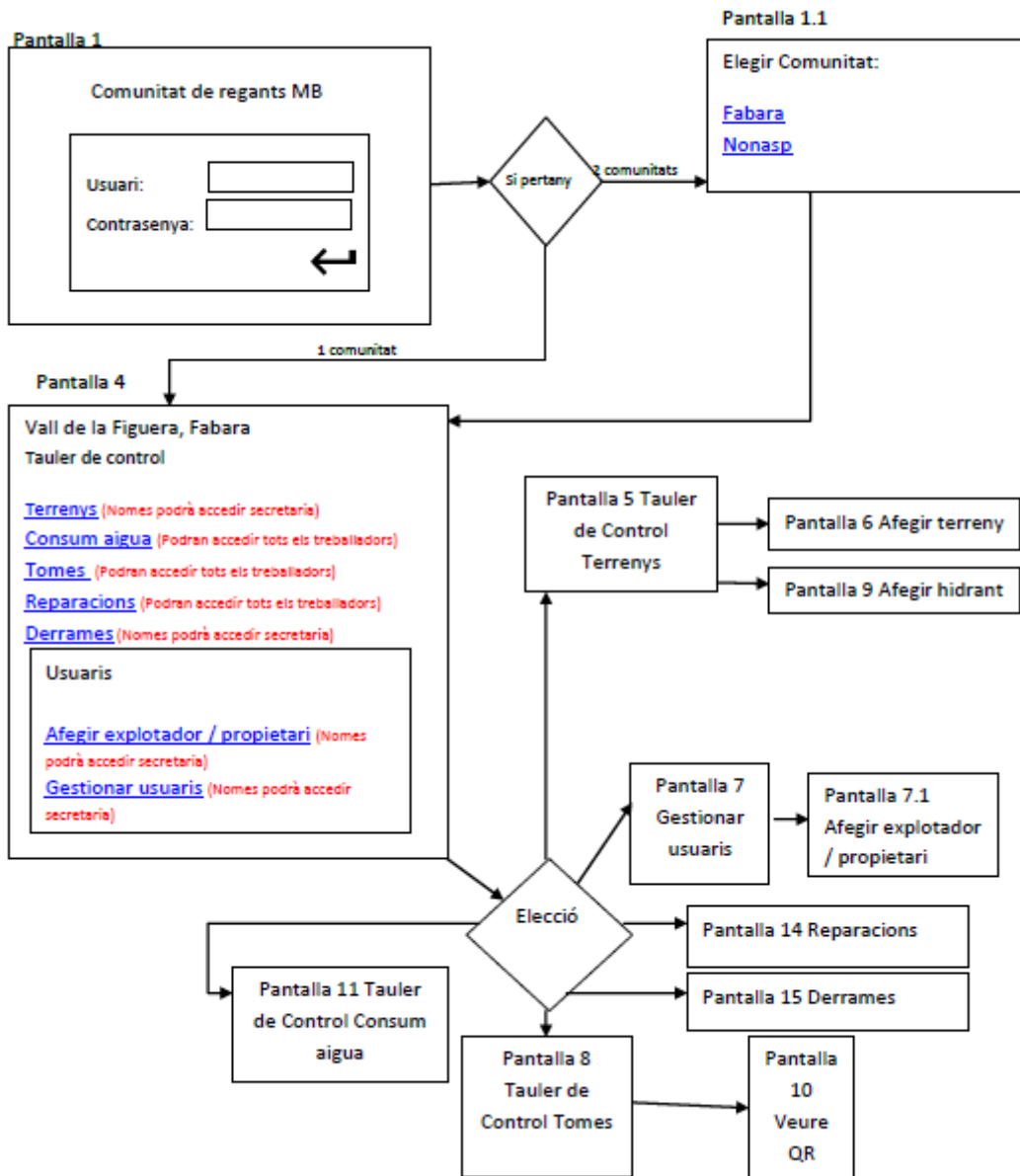


Figura 9 → Pantalles treballadors

Pantalla 5

Vall de la Figuera, Fabara


Tauler de Control Terrenys



Treballador Alberto Gutierrez



+ Afegir terreny + Afegir hidrant



Explotador	Propietari	Terme Municipal	Polígon	Parcel·la	Sub - parcel·la	Superfície (Ha)	Data concessió	Hidrant	Toma	Veure QR	Editar	Esborrar
...		
...		
...		

Figura 10 → Pantalla terrenys

Afegir terreny 

Explotador:   "Afegir explotador sino existeix"

Propietari:   "Afegir propietari sino existeix"

Terme municipal:   "Afegir terme municipal sino existeix"



Poligon:

Parcel·la:

Sub – parcel·la: "Potser null"

Superfície:

Data concessió:

Hidrant:   "Afegir hidrant sino existeix"

Toma




Figura 11 → Afegir terreny

Pantalla 7.1

Afegir explotador / propietari
Tipus usuari:

Explotador
 Propietari
 Tots dos

Nom:

DNI:

Prim Cognom:

Seg cognom:

Contrasenya:

Direcció:

Població:

CP:

Província:

Email:

Telèfon:

Num de compte:

Entitat bancària:

Tipus adeudo: " Per defecte RCUR"

Data firma:




Figura 12 → Afegir explotador/propietari

Pantalla 8

Tauler de Control Tomes

+ Afegir hidrant Usuari treballador

Explotador	Terme municipal	Polígon	Parcel·la	Sub – parcel·la	Superfície (Ha)	Data concessió	Hidrant	Toma	Veure QR	Editar
...	
...	
...	

Figura 13 → Pantalla Tomes

Pantalla 9

Afegir hidrant

Numero hidrant:

Coordena des:

Figura 14 → Afegir hidrant

Pantalla 11

Tauler de Control Consum aigua

Lectura a mostrar:

Ordenar per:

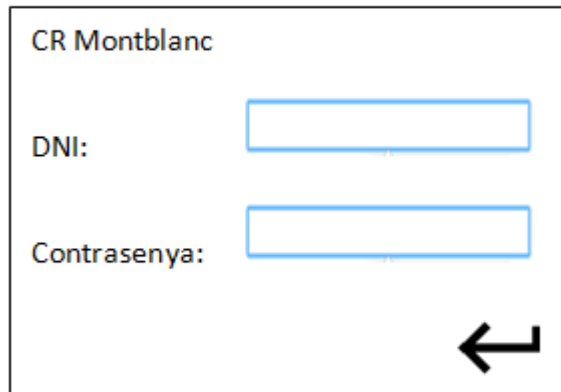
[Publicar / Despublicar](#) "Massivament"

Lectura "Data"	Hidrant	Toma	Explotador	Consum(m³)	Preu m³	Merma	Import	
...	Pub/Des
...	Pub/Des

Figura 15 → Tauler Consum aigua

2.3.2 Pantalles de l'aplicació Android

Pantalla 1



CR Montblanc

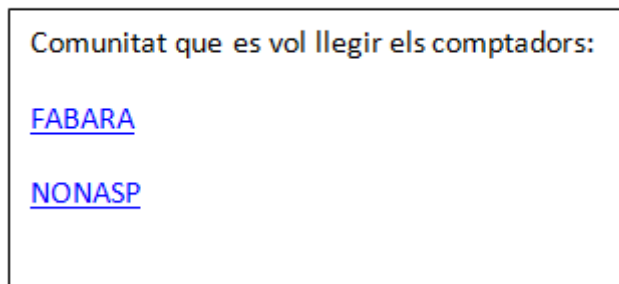
DNI:

Contrasenya:

←

Figura 16 → Login Android

Pantalla 2



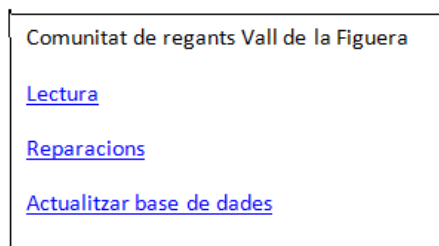
Comunitat que es vol llegir els comptadors:

[FABARA](#)

[NONASP](#)

Figura 17 → Elecció comunitat

Pantalla 2.1



Comunitat de regants Vall de la Figuera

[Lectura](#)

[Reparacions](#)

[Actualitzar base de dades](#)

Figura 18 → Tauler de Control Android

Pantalla 3

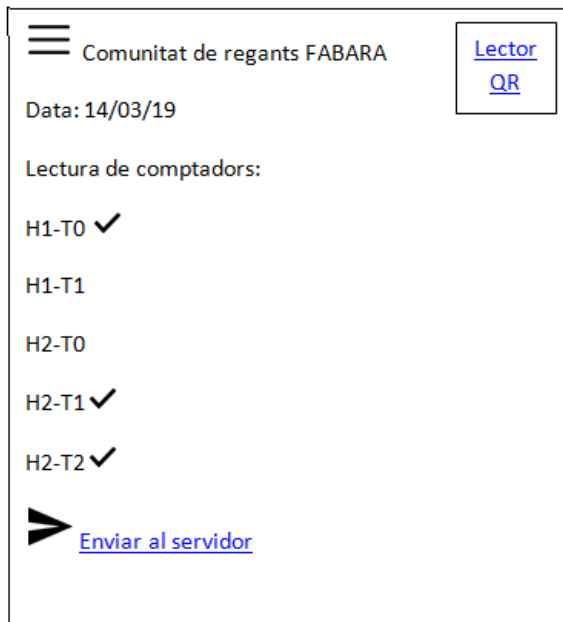


Figura 19 → Pantalla elecció hidrant toma

Pantalla 5

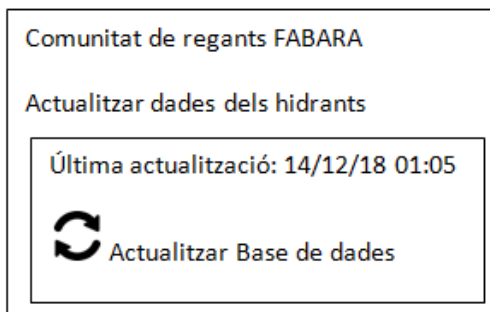


Figura 20 → Actualitzar base de dades servidor Java EE

Pantalla 6

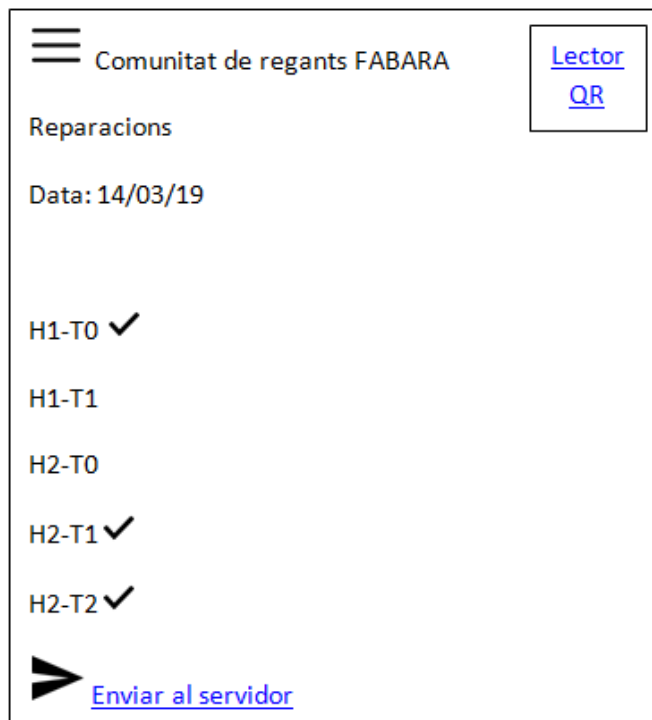


Figura 21 → Pantalla Reparacions hidrant-toma

Pantalla 6

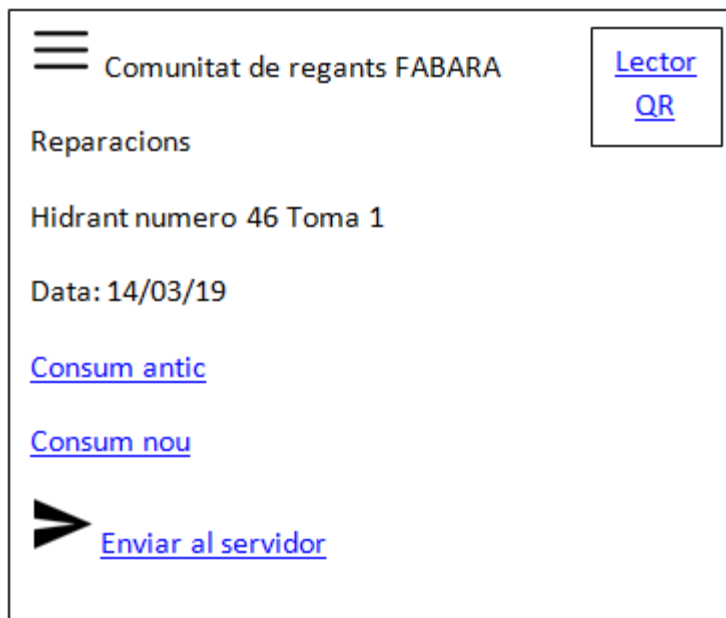


Figura 22 → Pantalla reparacions consum

Introdueix consum:

1	2	3	← x
4	5	6	Sig.
7	8	9	
	0		⚙️

Figura 23 → Introducció consum

3. Disseny

3.1 Servidor Java EE

3.1.1 Diagrama relacional de la base de dades

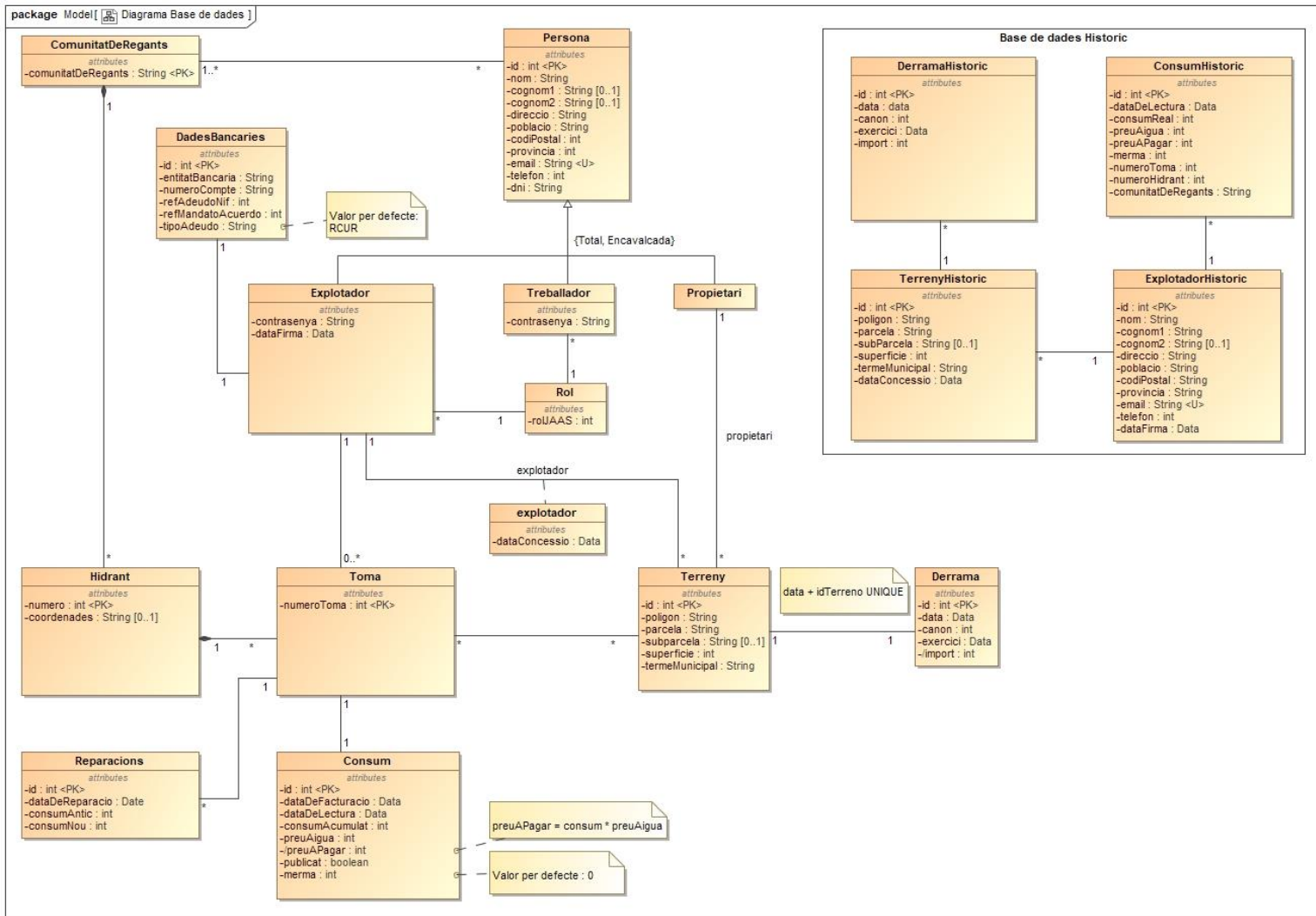


Figura 24 → Diagrama relacional de la base de dades Servidor JavaEE

3.1.2 Disseny lògic de la base de dades

Negreta i subratllat: Clau primària.

Subratllat: Clau forana.

Cursiva: restricció UNIQUE.

A l'hora de realitzar el disseny lògic de la relació d'herència de Persona he considerat les dos opcions següents:

OPCIÓ 1:

Persona: id, *dni*, nom, cognom1, cognom2, direcció, població, codiPostal, província, *email*, telèfon.

dni es un camp UNIQUE.

email es un camp UNIQUE.

Explotador: idExplotador, contrasenya, dataFirma, rolJAAS.

idExplotador es clau forana de Persona.

rolJAAS es clau forana de Rol.

Treballador: idTreballador, rolJAAS, contrasenya.

idTreballador es clau forana de Persona.

rolJAAS es clau forana de Rol.

Propietari: idPropietari.

idPropietari es clau forana de Persona.

Es clau forana de Persona.

OPCIÓ 2:

Persona: id, *dni*, nom, cognom1, cognom2, direcció, població, codiPostal, província, *email*, telèfon, contrasenya, dataFirma, rolJAAS, tipoUsuari.

email i dni son camps UNIQUE.

contrasenya, dataFirma i rolJAAS poden ser null.

He decidit utilitzar l'opció 2 ja que considero que al ser una herència total i encavalcada el nombre d'atributs nuls que hi haurà a cada fila es acceptable, a més considero que l'opció numero 1 complica excessivament el model de la base de dades ja que hi haurien taules, com Propietari, amb simplement una clau forana a Persona sense mes atributs. A banda tant Explotador com Treballador comparteixen els camps contrasenya i rolJAAS per tant el numero de camps nul és suficientment petit com per decantar-me per l'opció 2.

DadesBancaries: id, entitatBancaria, numeroCompte, refAdeudoNIF, refMandatoAcuerdo, tipoAdeudo, idExplotador.

idExplotador es clau forana de Persona.

ComunitatDeRegants: comunitatDeRegants.

CDRPersona: idPersona, comunitatDeRegants.

comunitatDeRegants es clau forana de ComunitatDeRegants.
idPersona es clau forana de Persona.

Hidrant: numero, comunitatDeRegants, coordenades.

comunitatDeRegants es clau forana de ComunitatDeRegants i forma clau primària composta amb numero.

Toma: numeroToma, numeroHidrant, comunitatDeRegants, idExplotador.

idExplotador es clau forana de Persona.
numeroHidrant i comunitatDeRegants es clau forana de Hidrant i forma clau primària composta amb numeroToma.

Reparacions: id, dataDeReparacio, consumAntic, consumNou, numeroToma, numeroHidrant, comunitatDeRegants.

numeroToma, numeroHidrant i comunitatDeRegants es clau forana de Toma.

Consum: id, dataDeLectura, consumAcumulat, preuAigua, preuAPagar, publicat, merma, numeroToma, numeroHidrant, comunitatDeRegants.

numeroToma, numeroHidrant, comunitatDeRegants es clau forana de Toma.

preuAPagar es un atribut derivat.

dataDeLectura, numeroToma, numeroHidrant i comunitatDeRegants formen una restricció UNIQUE.

TomesDelsTerrenys: idTerreny, numeroToma, numeroHidrant, comunitatDeRegants.

idTerreny es clau forana de Terreny i forma clau primària composta amb numeroToma, numeroHidrant i comunitatDeRegants.

numeroToma, numeroHidrant i comunitatDeRegants es clau forana de Toma.

Terreny: id, polígon, parcel·la, subparcela, superfície, termeMunicipal, dataConcessio, propietari, explotador.

propietari es clau forana de Persona.

explotador es clau forana de Persona.

Derrama: id, data, canon, exercici, import, idTerreny.

Import es un camp derivat.
idTerreny es clau forana de Terreny.
data i idTerreny formen una restricció UNIQUE.

3.1.3 Normalització

Primera forma normal:

Totes les taules estan automàticament en primera forma normal ja que a cada fila de totes les taules únicament hi haurà un valor.

Segona forma normal:

- Persona, DadesBancaries, ComunitatDeRegants, Consum, Terreny, Derrama i Reparacions: la clau primària es un únic camp per tant automàticament esta en 2FN.
- CDRPersona i TomesDelsTerrenys: la clau primària es composta per diversos camps, però com no hi ha mes camps a les taules a banda de les claus primàries considero que ja esta en 2FN.
- Toma: La clau primària esta formada per el numeroToma, numeroHidrant i comunitatDeRegants. En aquest cas el camp que no forma part de la clau primària (idExplotador) es clau forana i a més depèn completament dels camps que si son clau primària per tant es compleix la 2FN.
- Hidrant: la clau primària esta formada per numero del hidrant i la comunitatDeRegants. En aquest cas el camp que no forma part de la clau primària (coordenades) depèn completament dels camps que si son clau primària per tant es compleix la 2FN.

Tercera forma normal:

És comprova que la base de dades compleix amb la tercera forma normal verificant que no hi ha cap camp que depengui d'un altre camp que no sigui clau primària en cap de les taules.

Excepcions:

- El camp preuAPagar de la taula Consum, però això es degut a que es un camp derivat dels camps preuAigua i consum.
- El camp dataConcessio de la taula Terreny depèn del id del terreny i del idExplotador, però considero que no es rellevant ja que es un atribut propi de la relació entre Terreny i Explotador.

Forma normal de Boyce-Codd:

Considero que totes les taules estan en FNBC, la única taula que pot presentar dubtes es Consum, ja que es la única taula que te dos conjunts de claus candidates compostes. No obstant com les claus no presenten dependències entre les parts de les claus considero que la taula compleix la FNBC.

3.2 Aplicació Android

3.2.1 Diagrama relacional de la base de dades

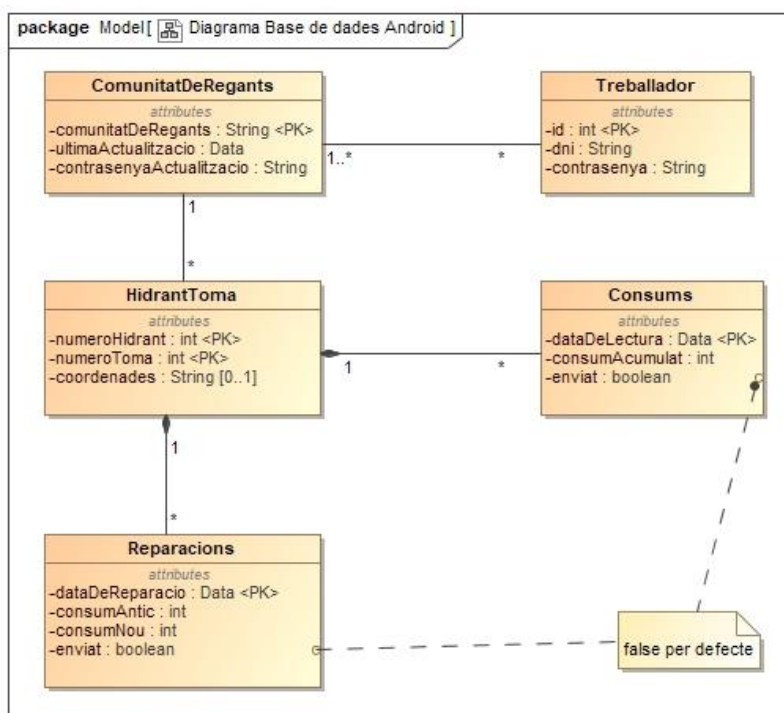


Figura 25 → Diagrama relacional de la base de dades Android

3.2.2 Disseny lògic de la base de dades

- **Negreta i subratllat:** Clau primària.
- Subratllat: Clau forana.
- *Cursiva:* restricció UNIQUE.

Treballador: id, dni, contrasenya.

dni te una restricció UNIQUE.

CDRTreballador: idTreballador, comunitatDeRegants.

idTreballador es clau forana de Treballador.
comunitatDeRegants es clau forana de ComunitatDeRegants.

ComunitatDeRegants: comunitatDeRegants, ultimaActualitzacio,
contrasenyaActualitzacio.

HidrantToma: numeroHidrant, numeroToma, comunitatDeRegants,
coordenades, escala.

comunitatDeRegants es clau forana de ComunitatDeRegants i forma
clau primària composta amb numeroHidrant i numeroToma.

Consums: dataDeLectura, numeroHidrant, numeroToma,
comunitatDeRegants consumAcumulat, enviat.

numeroHidrant, numeroToma, comunitatDeRegants es clau forana de
HidrantToma i forma clau primària composta amb dataDeLectura.

Reparacions: dataDeReparacio, numeroHidrant, numeroToma,
comunitatDeRegants, consumAntic, consumNou, enviat.

numeroHidrant, numeroToma, comunitatDeRegants es clau forana de
HidrantToma i forma clau primària composta amb dataDeReparacio.

3.2.3 Normalització

Primera forma normal:

Totes les taules estan automàticament en primera forma normal ja que a cada
fila de totes les taules únicament hi haurà un valor.

Segona forma normal:

- Treballador, ComunitatDeRegants: la clau primària es un únic camp per tant automàticament esta en 2FN.
- CDRTreballador : la clau primària és composta per diversos camps, però com no hi ha mes camps a les taules a banda de les claus primàries considero que ja esta en 2FN.
- HidrantToma: La clau primària esta formada per el numeroToma, numeroHidrant i comunitatDeRegants. En aquest cas els camps que no formen part de la clau primària (coordenades, escala) depenen completament dels camps que si son clau primària per tant es compleix la 2FN.
- Consums, Reparacions: la clau primària esta formada per la data de reparació / consum el numero del hidrant, numero de toma i la comunitatDeRegants. En aquest cas els camp que no formen part de la clau primària depenen completament dels camps que si son clau primària per tant es compleix la 2FN.

Tercera forma normal:

S'ha comprovat que la base de dades compleix amb la tercera forma normal verificant que no hi ha cap camp que depengui d'un altre camp que no sigui clau primària en cap de les taules.

Forma normal de Boyce-Codd:

Considero que totes les taules estan en FNBC, ja que no hi ha cap taula que tingui dos conjunts de claus candidates compostes.

3.3 Base de dades Històric

3.3.1 Diagrama relacional de la base de dades

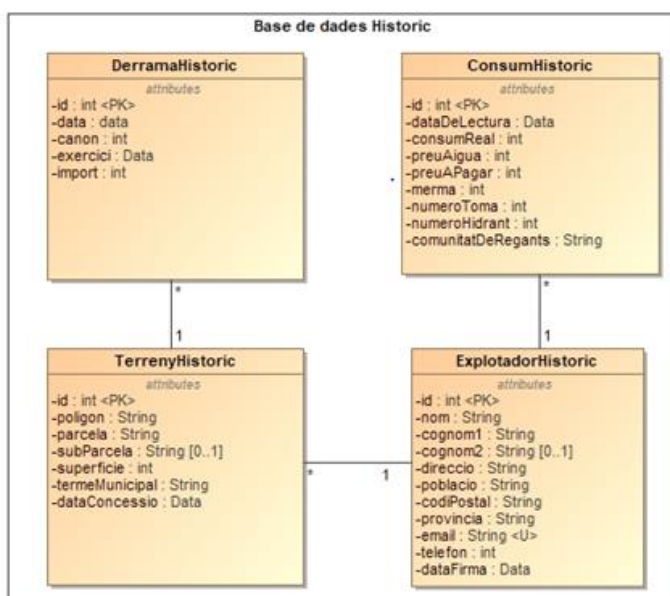


Figura 26 → Diagrama relacional de la base de dades històric

3.3.2 Disseny lògic

La base de dades historial no serà accessible des de l'aplicació, únicament es guardaran les dades per a que es puguin consultar en cas de que algun regant donat de baixa tingui alguna reclamació en el futur sobre els consums o les derrames.

- **Negreta i subratllat:** Clau primària.
- Subratllat: Clau forana.
- *Cursiva:* restricció UNIQUE.

DerramaHistoric: id, data, canon, exercici, import, idTerreny.

idTerreny es clau forana de TerrenyHistoric.

TerrenyHistoric: id, polígon, parcel·la, subParcela, superfície, termeMunicipal, dataConsessio, idExplotador.

idExplotador es clau forana de ExplotadorHistoric

ExplotadorHistoric: id, nom, cognom1, cognom2, direcció, població, codiPostal, provincia, email, telèfon, dataFirma.

ConsumHistoric: id, dataDeLectura, consumReal, preu Aigua, preuAPagar, merma, numeroToma, numeroHidrant, comunitatDeRegants, idExplotador.

idExplotador es clau forana de ExplotadorHistoric.

3.3.3 Normalització

Primera forma normal:

Totes les taules estan automàticament en primera forma normal ja que a cada fila de totes les taules únicament hi haurà un valor.

Segona forma normal:

DerramaHistoric, TerrenyHistoric, ExplotadorHistoric, ConsumHistoric: la clau primària es un únic camp per tant automàticament esta en 2FN.

Tercera forma normal:

S'ha comprovat que la base de dades compleix amb la tercera forma normal verificant que no hi ha cap camp que depengui d'un altre camp que no sigui clau primària en cap de les taules.

Forma normal de Boyce-Codd:

Considero que totes les taules estan en FNBC, ja que no hi ha cap taula que tingui dos conjunts de claus candidates compostes.

4. Implementació

En aquest apartat argumentaré els aspectes mes destacats sobre la implementació del projecte.

4.1 Servidor Java EE

4.1.1 Diagrama de components versió final

El diagrama de components l'adjunto en una imatge a l'entrega final ja que és una imatge molt gran.

4.1.2 Claus compostes JPAs

Tal i com es veu al diagrama anterior hi ha diverses entitats JPA que tenen claus compostes formades per objectes d'altres entitats JPA, per exemple la clau primària de TomaJPA és el numero de la toma (int) + un objecte de tipus hidrant i per la seva banda el hidrant te com a clau primaria un string identificatiu + un objecte comunitatDeRegants

```
@ManyToOne()
@JoinColumns({
    @JoinColumn(name="`numeroHidrant`", referencedColumnName="`numeroHidrant`", nullable=false),
    @JoinColumn(name="`comunitatDeRegants`", referencedColumnName="`comunitatDeRegants`", nullable=false)
})
private HidrantJPA hidrant;

@Column(name = "`numeroToma`", nullable = false)
private int numeroToma;
```

Figura → TomaPK

```
|
@Column(name = "`numeroHidrant`", nullable = false, length = 6)
private String numeroHidrant;

@ManyToOne()
@JoinColumn(name="`comunitatDeRegants`", nullable=false)
private ComunitatDeRegantsJPA comunitatDeRegants;
```

Figura → HidrantPK

Com s'està utilitzant JPA 2.1 és va haver de generar unes classes especials que actuessin com a clau primària per a embeirles (Embedded) a les classes JPA de les quals havien de ser claus primàries.

4.1.3 APIs i frameworks

4.1.3.1 Generar PDFs Derrama

L'aplicació incorpora una funcionalitat per a generar pdfs de derrames, per a generar aquests pdfs s'ha utilitzat la llibreria "openpdf-1.2.17" ja que és una llibreria gratuïta i opensource.

4.1.3.2 Primefaces

S'ha utilitzat el framework Primefaces per a implementar les pàgines JSF ja que és un framework molt potent i disposa de moltes funcionalitats interessants. Per exemple, s'ha utilitzat per a crear totes les taules de l'aplicació les quals tenen les següents funcionalitats:

- Camps de filtratge

- Paginador
- Celes editables
- Amplada de files escalable
- Carga de dades de la taula lazy
- Implementació ajax molt còmoda
- Generador automàtic d'excels/pdfs (llibreries poi-4.1.0 i openpdf-1.2.17)

Altres funcionalitats:

- Missatges flotants
- Gràfiques
- Botons
- Barra de navegació
- Inputs en menció especial al input de calendari
- Validadors de inputs i missatges d'error
- CommandButton amb Ajax

4.1.3.3 JAX-RS

Al haver utilitzat WildFly com a servidor ja és tenia disponible aquesta API per a implementar un servei REST el qual s'utilitza per a comunicar l'aplicació Android amb el servidor.

Les rutes per a poder accedir a aquest servei son les següents:

En la ruta següent obtenim un llistat de tots els hidrants i les tomes del servidor per a que el Android pugui actualitzar la seva base de dades interna de hidrants i tomes.

https://ip_del_server:8443/CRMB/rest/hidrantsTomes/getAllHidrantsTomes

En la ruta següent s'obte tots els consums que estan guardats al servidor web per a que quan s'envii la informació dels hidrants i les tomes de la ruta anterior, també es pugi consultar el consum de totes les tomes.

http://ip_del_server:8443/CRMB/rest/consums/getAllConsums

Les dos rutes anteriors s'utilitzen en el moment en el que desde l'aplicació Android és selecciona l'opció d'actualitzar la base de dades desde el servidor quan es fa aquesta operació s'accedeix a aquestes dos rutes perquè el mes provable es que es doni el cas que l'aplicació estigui instal·lada a diversos telèfons de varis treballadors, aleshores és la forma en la que cada cop que un treballador s'actualitza en front al servidor si Android detecta que el servidor te consums que ell no te aleshores assumeix que algun altre treballador a introduït consums i els introdueix com a propis a la base de dades Android.

Per tan, seguint aquest sistema tan la base de dades del servidor com les dels diversos dispositius Android estan sincronitzats, a més, per evitar problemes de duplicacions de claus primàries (per exemple si utilitzéssim un integer auto

generat per als consums) s'ha indicat que la clau primària dels consums la formen la data de lectura, la toma, l'hidrant i la comunitat de regants per tan, al utilitzar aquesta clau composta ens assegurem que no hi haurà dos treballadors diferents que intentin llegir el mateix consum i en cas de que ho facin simplement s'interpretarà que el consum més actual és el vàlid i s'actualitzarà la informació.

La ruta següent s'utilitza per a que els treballadors quan fan la ronda de lectura de comptadors un cop han llegit tots els comptadors puguin enviar les dades recollides al servidor i que aquest les introdueixi a la base de dades, en cas de trobar que una lectura ja ha estat introduïda l'actualitza.

http://ip_del_server:8443/CRMB/rest/consums/sincronitzarConsums

4.1.2 Mètode consums

He decidit documentar el funcionament d'aquest mètode ja que és el més complex de l'aplicació i és important documentar-ho per entendre com funciona.

Tenint en compte que l'aplicació funcionarà en un cas real, he dedicat bastant temps amb la implementació d'aquest mètode ja que és molt important que és calculi correctament el consum d'aigua consum per cada regant.

Aquest mètode ha de contemplar els següents casos:

- Fer una lectura normal entre dos dates sense casos especials.
- Fer una lectura amb una reparació pel mig.
- Fer una lectura amb dos o més reparacions pel mig.
- Fer una lectura amb volta de comptador (els comptadors de l'aigua podent donar la volta, al arribar a 99.999 m³ el comptador torna a començar en 0 m³ i d'altres, al arribar a 999.999 m³ tornen a començar en 0 m³) per tant el mètode a de contemplar aquests casos especials per evitar errors de lectura.

Codi del mètode:

```
//Parametres: El consum actual del qual es vol calcular el preu a pagar
//Retorna a la posicio 0 preuAPagar (float)
//Retorna a la posicio 1 consumReal (float)
public float[] calcularPreuAPagar(ConsumJPA consum) throws ExceptionBean {

    String numhidrant = consum.getIdHidrant();
    String comunitat = consum.getComunitat();
    int idToma = consum.getIdToma();
    LocalDate dataConsumActual = consum.getId().getDataDeLectura();
    float consumActual = consum.getConsumAcumulat();

    /*S'obte el l·listat de consums Anteriors al consum actual.
    S'obte una l·lista de consums i no simplement el consum anterior directe ja que es possible estar en el cas
seguent:
    Gener: Toma X consum: 200 m3;
    Febrer: Toma X consum: 0 m3;
    Març: Toma X consum: 500 m3;
    //Aixo pot passar si un treballador no passa per la toma X el mes de febrer pero la posa a 0 per defecte per
    indicar que aquest mes no s'ha passat per fer la lectura de la toma X.
    */
    List<ConsumJPA> llistaConsumsAnterior = getLlistaConsumsAnteriors(numhidrant, comunitat, idToma,
    dataConsumActual);
    //Si no hi ha consums anteriors, vol dir que estem davant del primer consum de la toma, per tant tant el preu com
    el consumReal es posen a 0
        if(llibraConsumsAnterior.isEmpty()) {
            float[] r = {0,0};
            return r;
        }

    //Ara es busca quin es l'ultim consum que no te el valor de consumAcumulat a 0,
```

```

//seguint amb l'exemple anterior de la toma X si estem calculant el consum de març hauriem d'ignorar el consum de febrer i seleccionar el de gener
    int consumNoZeroSeleccionat=0;
    for(int i=0;i<llistaConsumsAnterior.size();i++) {
        if(llistaConsumsAnterior.get(i).getConsumAcumulat()!=0) {
            consumNoZeroSeleccionat=i;
            break;
        }
    }

//Guardem quin es el consum real (diferent de 0) del actual i guardem quin es el valor del seu consum acumulat.
    ConsumJPA consumAnterior = llistaConsumsAnterior.get(consumNoZeroSeleccionat);
    float valorConsumAnterior = consumAnterior.getConsumAcumulat();

//A continuació es busca si la toma de la qual s'està calculant el consum té reparacions sense comptabilitzar
    List<ReparacionsJPA> llistaReparacions = getAllReparacionsTomaSenseComptabilitzar(numhidrant, comunitat, idToma);

    /*
    Aquesta variable guardara el consumReal que ha tingut la toma entre la lectura anterior i la lectura actual.
    Es necessita conèixer aquest valor ja que es el que finalment s'ha de multiplicar per el preu de l'aigua
    */
    float consumReal=0;

    if (!llistaReparacions.isEmpty()) {//Si la toma té reparacions que s'han de tenir en compte ->

//Es calcula el consumReal tenint en compte que la primera reparació es un cas especial
//En primer lloc es mira si el consumAntic de la reparació < que el valor del consumAnterior,
//En cas afirmatiu vol dir que el comptador de la toma ha arribat al màxim i ha donat la volta
//De comptadors hi ha de 2 tipus els que donen la volta a 100.000 m3 i els que donen la volta a 1.000.000 m3
        if(llistaReparacions.get(0).getConsumAntic() < valorConsumAnterior) {//Si el comptador de la toma a donat la volta ->

            if(valorConsumAnterior < 100000) {//Si el comptador dona la volta a 100.000 ->

```

```

consumReal = consumReal + 100000 - valorConsumAnterior + llistaReparacions.get(0).getConsumAntic();
}
    else {//Sino es que dona la volta a 1.000.000 ->

        consumReal = consumReal + 1000000 - valorConsumAnterior +
        llistaReparacions.get(0).getConsumAntic();
        }
}
    else {//El comptador no ha donat la volta -->

        consumReal = consumReal + llistaReparacions.get(0).getConsumAntic() - valorConsumAnterior;
        }
//Es marca la reparacio com ja comptabilitzada per a no tenirla en compte en futurs calculs de consums
        llistaReparacions.get(0).setComptabilitzat(true);

//Si hi ha hagut mes d'una reparacio, es calcula en aquest bucle com afecta al consumReal totes les
reparacions posteriors a la primera i diferents de la ultima
        for(int i=1;i<llistaReparacions.size();i++) {

//Al igual que en el cas anterior s'ha de mirar si el comptador reparat de la toma ha donat la volta
            if(llistaReparacions.get(i).getConsumAntic() < llistaReparacions.get(i-1).getConsumNou()) {//En
cas que el comptador hagi donat la volta ->

                if(llistaReparacions.get(i-1).getConsumNou() < 100000) {//Si el comptador dona la volta a
100.000 ->

                    consumReal = consumReal + 100000 - llistaReparacions.get(i-1).getConsumNou() +
                    llistaReparacions.get(i).getConsumAntic();

                }
            }
            else {//Sino es que dona la volta a 1.000.000 ->

                consumReal = consumReal + 1000000 - llistaReparacions.get(i-1).getConsumNou() +
                llistaReparacions.get(i).getConsumAntic();
            }
        }
    }
}

```

```

    }
    else { //El comptador reparat no ha donat la volta -->

    consumReal = consumReal + llistaReparacions.get(i).getConsumAntic() -
    llistaReparacions.get(i-1).getConsumNou();
    }
    //Es marca la reparacio com ja comptabilitzada per a no tenirla en compte en futurs
calculs de consums
    llistaReparacions.get(i).setComptabilitzat(true);
    //Logger.getLogger(this.getClass()).info("Cas intermig: "+i+" ConsumReal: "+consumReal);
    }

//Un cop calculat com afecta tot el conjunt de reparacions de la toma sense comptabilitzar al
consumReal es calcula el consumReal final correcte
//Es mira si el comptador actual del consum ha donat la volta

    if(consumActual < llistaReparacions.get(llistaReparacions.size()-1).getConsumNou()) { //En cas
    que el comptador hagi donat la volta ->

        if(llistaReparacions.get(llistaReparacions.size()-1).getConsumNou() < 100000) { //Si el
        comptador dona la volta a 100.000 ->

            consumReal = consumReal + 100000 -
            llistaReparacions.get((llistaReparacions.size()-1)).getConsumNou() + consumActual;

        }
        else { //Sino es que dona la volta a 1.000.000 ->

            consumReal = consumReal + 1000000 -
            llistaReparacions.get((llistaReparacions.size()-1)).getConsumNou() + consumActual;

        }
    }
    else { //El comptador del consum actual no ha donat la volta -->

    consumReal = consumReal + consumActual -
    llistaReparacions.get((llistaReparacions.size()-1)).getConsumNou();

```

```

    }

    //Es marca la ultima reparacio com ja comptabilitzada per a no tenirla en compte en futurs
    //calculs de consums
    llistaReparacions.get((llistaReparacions.size()-1)).setComptabilitzat(true);
}
//Si la toma no te reparacions que s'han de tenir en compte ->
else {

    //Es comprova que el comptador la toma no hagi donat la volta
    if(consumActual < valorConsumAnterior ) { //En cas que el comptador hagi donat la volta ->

        if(valorConsumAnterior < 100000) { //Si el comptador dona la volta a 100.000 ->

            consumReal = consumReal + 100000 - valorConsumAnterior + consumActual;

        }
        else { //Sino es que dona la volta a 1.000.000 ->

            consumReal = consumReal + 1000000 - valorConsumAnterior + consumActual;

        }
    }
}
else { //El comptador del consum actual no ha donat la volta -->
    //Aquest es el cas mes habitual: la toma no ha tingut reparacions i el comptador de la
    //toma no ha donat la volta.

    consumReal = consumReal + consumActual - valorConsumAnterior;
}
}

//Retorna a la posicio 0 preuAPagar
//Retorna a la posicio 1 consumReal
float[] ret = {consumReal*consum.getPreuAigua(),consumReal};
return ret;

```

M'hagués agradat fer proves unitàries d'aquest mètode amb JUnit però no he tingut prou temps per fer-ho.

He dissenyat diferents casos d'ús manualment per comprovar que retorna els valors esperats depenent el tipus de cas.

Cas 1 → Lectura normal del comptador d'aigua

El dia 02/05 es va realitzar la lectura del comptador de l'hidrant 101 Toma 1, la lectura va ser de 84236 m³, el dia 02/06 es va tornar a fer la lectura del comptador, la lectura estava a 90000 m³.

Cas 2 → Lectura normal amb una reparació

El dia 30/03 es va realitzar la lectura del comptador de l'hidrant 202 Toma 1, la lectura va ser de 90000 m³, el dia 12/04 es va realitzar el canvi del comptador i es va generar una reparació. La lectura del comptador que es va treure era de 90500 m³ i es va substituir per un altre comptador inicialitzat a 0 m³. Finalment el dia 30/04 es va realitzar la lectura del comptador que estava a 3000 m³.

Cas 3 → Lectura normal amb dues reparacions el mateix mes

El dia 30/03 es va realitzar la lectura del comptador de l'hidrant 202 Toma 1, la lectura va ser de 14500 m³, el dia 12/04 es va realitzar el canvi del comptador i es va generar una reparació. La lectura del comptador que es va treure era de 20000 m³ i es va substituir per un altre comptador inicialitzat a 300 m³. El dia 20/04 va haver-hi una nova averia al comptador i es va tenir que canviar per un altre comptador. La lectura del comptador que es va treure era de 1000 m³ i es va substituir per un altre comptador inicialitzat a 200 m³. Finalment el dia 30/04 es va realitzar la lectura del comptador que estava a 3000 m³.

Cas 4 → Lectura amb volta del comptador a 100000 m³

El dia 30/04 és va realitzar la lectura del comptador de l'hidrant 202 Toma 1, la lectura va ser de 90000 m³, el dia 30/05 és va realitzar una nova lectura del mateix comptador i la lectura va ser 300 m³.

4.2 Aplicació Android

4.2.1 APIs utilitzades

4.2.1.1 Room

S'ha utilitzat aquesta API per a implementar la base de dades de l'aplicació Android, és molt similar a JPA ja que permet abstraure'ns del maneig de la base de dades SQLite mapejant objectes directament a la base de dades.

En aquest cas és defineixen per una banda les entitats les quals son les classes que es mapejen a la base de dades i per altra banda és defineixen els

DAO els quals són classes abstractes que especifiquen mitjançant anotacions les operacions sql que es poden executar sobre les classes de tipus entitat.

4.2.1.2 Asynchronous Http Client

Aquesta és la API que s'utilitza per implementar les crides http a fora del thread de la interfície gràfica cap al servidor JavaEE.

4.2.1 Error conegut

L'aplicació Android presenta un error conegut el qual en un entorn real no hauria de passar però per equivocació dels treballadors potser que passi.

Concretament si hi ha consums no sincronitzats (és a dir que encara no han sigut enviats al servidor) en dos o més mesos diferents, aleshores en el moment en que s'envien al servidor com s'envien tots a la vegada i no en diverses crides al servidor agrupades per mesos el càlcul de consums reals i del preu a pagar no és realitza correctament per a tots els consums posteriors al primer consum més antic que s'està enviant de cop.

4.3 Problemes principals desenvolupant l'aplicació

4.3.1 Problema Linux

Inicialment al ser aquest un projecte pensat per a ser portat a un entorn real es va voler instal·lar el servidor JAVA EE en un Debian9.8 x64 ja que considero que el aquest sistema operatiu es molt millor que un entorn Windows per a un servidor.

El problema va sorgir en el moment que és volia executar qualsevol consulta JPQL, per exemple:

```
List<PersonaJPA> persona = (List<PersonaJPA>) em.createQuery("from PersonaJPA t where t.dni = :dni", PersonaJPA.class).setParameter("dni", dni).getResultList();
```

Executant aquesta consulta saltava una exception de tipus ClassCastException indicant que jpa.PersonaJPA cannot be cast to jpa.PersonaJPA.

Es va arribar a la conclusió que per algun motiu el classloader del objecte que retorna el entityManager es diferent al classLoader de la classe PersonaJPA. És va assegurar que el problema era aquest mitjançant el següent codi.

```
Query query = em.createQuery("SELECT t.nom from PersonaJPA t where t.dni = :dni");
query.setParameter("dni", dni);
Object result = null;
try {
    result = query.getSingleResult();
    PersonaJPA p = (PersonaJPA) result;
    return p.getNom();
}
```

```

} catch (javax.persistence.NoResultException ex) {
    return null;
} catch (ClassCastException ex) {
    if(result.getClass().getClassLoader() !=
        PersonaJPA.class.getClassLoader()) {
        return ("Object classloader: " +
            result.getClass().getClassLoader() + "--"+ "Target class
            classloader: " + PersonaJPA.class.getClassLoader());
    }
}
}

```

Amb aquesta modificació el return del mètode era el següent:

```

Object classloader: null--Target class classloader: ModuleClassLoader
for Module "deployment.CRMB.ear.CRMB.war" from Service Module Loader

```

Per tan, va resultar evident que el problema era que s'estaven utilitzant dos classloader diferents.

Es van intentar múltiples solucions:

- Crear el projecte de nou
- Instal·lar versions diferents del WildFly
- Instal·lar MariaDB en comptes de PostgreSQL
- Utilitzar JDBC's diferents
- Utilitzar versions diferents de JPA
- Modificar els fitxers de configuració
- Compilar el projecte amb Maven en comptes de amb Ant
- Modificar permisos de carpetes
- Etc

Tot i així no és va aconseguir solucionar el problema, per tan, finalment és va decidir instal·lar el servidor en un Windows 10 i utilitzant exactament les mateixes versions que s'havien utilitzat amb el Linux va funcionar sense cap problema.

5. Seguretat

Com el projecte s'implementarà en un cas real s'ha decidit intentar aportar-li un mínim de seguretat. A continuació es detallarà els diferents mecanismes que intervenen en aquests aspectes.

5.1 JAAS

S'ha utilitzat JAAS principalment per a complir dos objectius, per una banda implementar un login i per altra banda definir diversos rols d'usuari per a determinar depenent el rol en que s'estigui logejat a quines pàgines o funcionalitats és pot accedir.

5.1.2 Login

En primer lloc al fitxer standalone.xml del WildFly s'ha definit un nou security-domain el qual s'encarrega de realitzar la verificació d'usuari, contrasenya i rol mitjançant les consultes SQL definides a les opcions rolesQuery i principalsQuery.

A banda, en el fitxer web.xml s'han definit quatre security-role els quals son:

- Secretaria
- Explotador
- Propexpl
- Treballador

Depenent de amb quin rol s'autentifiqui l'usuari guanyarà accés a unes pàgines o unes altres.

A més, en aquest fitxer s'indica que el mètode d'identificació serà del tipus FORM.

5.1.3 Restricció de pàgines

S'utilitza JAAS per delimitar a quines pàgines pot accedir cada usuari i també per a que en el moment en el que entra un usuari per primer cop a l'aplicació i un cop estigui autenticat se'l redireccioni automàticament al home que l'hi pertoca.

5.1.4 Maneig de credencials d'usuari

S'utilitza un ManagedBean anomenat JaasMBean per a interactuar amb el mecanisme de persistència de JAAS ja que ofereix diferents mètodes per a obtenir informació sobre el usuari que està logejat i com és guarda la informació a través de JAAS i no de ManagedBean aquest ManagedBean pot ser RequestScoped.

5.2 SSL

S'ha configurat el servidor WildFly per a utilitzar un certificat auto signat generat amb l'eina keyTool per a proporcionar una connexió SSL a tots els seus clients tant de l'aplicació web com del servidor REST.

S'ha definit al fitxer web.xml que totes les pàgines de l'aplicació estan en mode CONFIDENTIAL el qual vol dir que es força a utilitzar SSL per accedir al servidor.

A més, també s'ha modificat el security-real anomenat ApplicationRealm per a que utilitzi la keyStore generada mitjançant l'eina KeyTool.

Per part de l'aplicació Android com s'està utilitzant un certificat auto signat s'ha tingut que forçar que el client http confii amb aquest tipus de certificats mitjançant els paràmetres següents:

```
SyncHttpClient client = new SyncHttpClient(true, 8080, 8443)
```

5.3 Autenticació API Rest

Per a proporcionar un mínim de seguretat a la API REST s'ha decidit enviar a través del canal SSL, i incrustat-lo en el header de les peticions, un usuari i contrasenya de rol treballador que ja ve introduït amb la pròpia aplicació per a que si qualsevol persona intenta accedir a la API mitjançant un navegador o qualsevol altra forma que no sigui mitjançant l'aplicació Android es trobi que la API retorna un objecte JSON buit en comptes del resultat esperat. Hi ha aproximacions molt més segures que aquesta, però de moment he considerat que era suficient.

5.4 Còpies de seguretat

S'ha decidit crear una còpia de la base de dades diàriament. Per dur a terme la còpia s'ha utilitzat l'eina pg_dump.exe que porta incorporat el PostgreSQL. S'ha creat el següent script per generar la còpia.

```
@echo off

for /f "skip=8 tokens=2,3,4,5,6,7,8 delims=: " %%D in ('robocopy /1 * \ \ /ns /nc /ndl /nfl /np /njh /XF * /XD *') do (
    set "dow=%%D"
    set "month=%%E"
    set "day=%%F"
    set "HH=%%G"
    set "MM=%%H"
    set "SS=%%I"
    set "year=%%J"
)

set datestr=%day%_%month%_%year%_%HH%_%MM%_%SS%
echo datestr is %datestr%

set BACKUP_FILE=C:\BACKUPS_DB_POSTGRESQL\BACKUPS\Backup_CRMB_%datestr%.backup
echo backup file name is %BACKUP_FILE%
SET PGPASSWORD=
echo off
bin\pg_dump -h localhost -p 5432 -U USER -F c -b -v -f %BACKUP_FILE% postgresCRMB
```

Figura 27 → Script copia seguretat

Posteriorment s'ha programat l'execució de l'script amb l'eina "Tareas Programadas" que porta incorporada Windows per a que es realitzi una còpia de la base de dades cada dia a les 03:00 de la matinada.

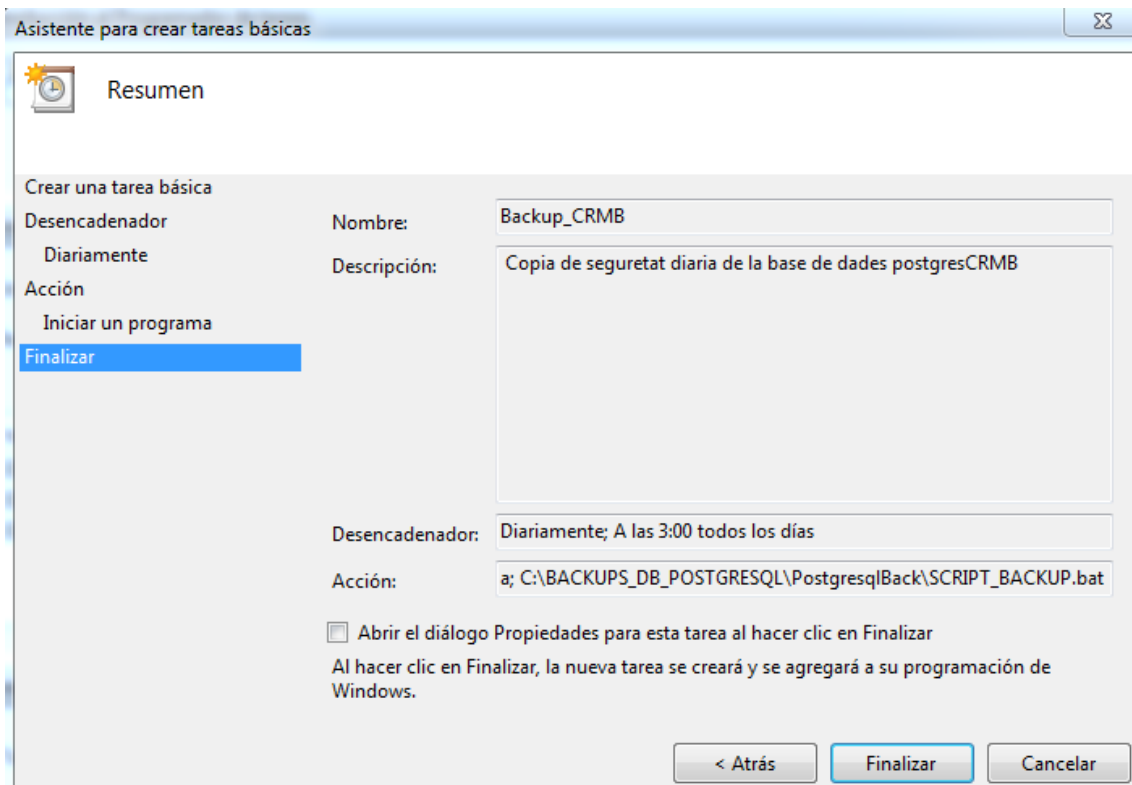



Figura 28 → Tasca programada per a la copia de seguretat

Finalment l'script generà un fitxer .backup amb el format:

 Backup_CRMB_09_Jun_2019_06_04_18.backup

6. Futures millores

S'ha pensat amb les següent millores per implementar-les a l'aplicació però no obstant queden fora de l'abast del projecte actual.

- Millorar la seguretat de la API REST
- Incorporar un certificat que no sigui auto signat per a SSL
- Incorporar un menu lateral a l'aplicació Android per als botons d'interacció amb el servidor
- Incorporar un mecanisme de sincronització automàtica de la base de dades Android al servidor JavaEE i viceversa
- Instal·lar l'aplicació en un servidor Linux tot i incorporant a aquest servidor un firewall iptables complet
- Afegir més estadístiques a les pàgines informatives
- Incorporar la base de dades històric per emmagatzemar l'historial de terrenys, tomes, usuaris, consums etc...

7. Conclusions

Aquest treball ha significat un gran repte per a mi ja que fins ara no havia realitzat mai un projecte d'aquesta grandària. A estat molt satisfactori la creació d'un producte a partir d'una idea principal i desenvolupar-la passant per les diferents fases de vida. He obtingut un punt de vista més realista del que tenia fins ara per a la realització d'un producte final, ja que fins ara havia realitzat petits treballs però mai un projecte complet començant per la fase de planificació fins a la posada en funcionament.

Personalment crec que he assolit casi tots els objectius que tenia plantejats ja que he pogut aplicar els coneixements que he anat adquirint durant el grau, aprendre Android, frameworks que eren nous per a mi i centrar aquests coneixements per realitzar un producte final bastant complet. L'únic objectiu que no he pogut assolir de cara a l'entrega del projecte és la creació d'un servidor funcionant amb Linux, ja que vaig tenir problemes amb el funcionament del servidor i finalment vaig tenir que muntar el servidor en una màquina Windows per poder acabar el projecte a temps. No obstant, tinc pensat fer-ho funcionar amb un servidor Linux Debian per la posada en marxa a la feina i poder aplicar regles de seguretat.

Crec que la metodologia que he seguit al llarg del projecte ha estat l'adequada ja que més o menys he seguit la planificació que tenia plantejada al principi, en alguns moments pensava que no arribava a la entrega ja que com he comentat anteriorment vaig perdre uns dies intentant fer funcionar el servidor en una màquina Linux però vaig poder rectificar i seguir endavant amb la implementació del servidor en Windows. He intentat ser el més previsor possible amb la fase d'implementació ja que com hem va avisar el consultor era una fase amb molt consum de temps i he tingut que dedicar més hores de les que esperava, sobretot perquè Android era bastant nou per a mi i he tingut que dedicar moltes hores d'aprenentatge. D'altra banda, si tot va be el projecte s'implantarà a la realitat a la feina que estic actualment, he intentat realitzar un producte que sigui funcional i el més intuïtiu possible per als treballadors i usuaris de l'aplicació.

En definitiva, estic molt satisfet amb el producte obtingut i espero poder aplicar els coneixements adquirits a futurs treballs.

8. Glossari

Explotador → Agricultor que pertany a la comunitat que te terrenys en regadiu.

JAAS → Java authentication and authorization

Android → Sistema operatiu per a telèfons intel·ligents

Smartphone → telèfon intel·ligent

APK → Instal·lador d'una aplicació Android

Login → Inici de sessió a la pàgina web

API → Llibreria de programació

PostgreSQL → Motor de Base de dades

API REST → Interfície entre sistemes a través de HTTP

SSL → Secure Sockets Layer

Iptables → Firewall incorporat en moltes distribucions Linux

Framework → Entorn de desenvolupament d'una aplicació

Eclipse → Editor i compilador de codi

9. Bibliografia

- [7/04/2019] <https://www.logicbig.com/tutorials/java-ee-tutorial/jsf/converter-and-validator.html>
- [8/04/2019] <https://stackoverflow.com/questions/6250685/how-to-set-a-custom-message-for-a-standard-validator-in-jsf>
- [8/04/2019] <https://www.primefaces.org/showcase/index.xhtml>
- [10/04/2019] <https://stackoverflow.com/questions/19548838/updating-entire-pdatatable-on-complete-of-pajax-event-celledit/19927578#19927578>
- [10/04/2019] <https://webdev.jhuep.com/~jcs/ejava-javaee/coursedocs/content/html/ejbt-container.html#ejbt-container-scope-required>
- [10/04/2019] <https://github.com/LibrePDF/OpenPDF>
- [14/04/2019] <https://www.javacodegeeks.com/2013/09/jaas-secured-jax-rs-end-point.html>
- [16/04/2019] <https://stackoverflow.com/questions/30936865/how-to-make-sync-or-async-http-post-get>
- [16/04/2019] <http://www.mastertheboss.com/jboss-server/jboss-security/complete-tutorial-for-configuring-ssl-https-on-wildfly>
- [23/04/2019] <https://www.mkyong.com/webservices/jax-rs/get-http-header-in-jax-rs/>
- [23/04/2019] <https://www.vogella.com/tutorials/REST/article.html>
- [07/05/2019] <https://developer.android.com/training/data-storage/room/referencing-data>
- [07/05/2019] <https://android.jelise.eu/getting-started-with-room-persistence-library-8932276b4d8c>
- [12/05/2019] <https://stackoverflow.com/questions/30594572/send-json-as-a-post-request-to-server-by-asynchttpclient>
- [15/05/2019] <https://stackoverflow.com/questions/5554217/google-gson-deserialize-listclass-object-generic-type>
- [23/05/2019] <https://stackoverflow.com/questions/12850143/android-basics-running-code-in-the-ui-thread>
- [24/05/2019] <https://stackoverflow.com/questions/5245840/how-to-convert-jsonstring-to-jsonobject-in-java>
- [28/05/2019] https://github.com/codepath/android_guides/wiki/Using-Android-Async-Http-Client
- [05/06/2019] <https://android.jelise.eu/android-architecture-components-room-relationships-bf473510c14a>

10. Anexos

10.1 Manual per a treballadors de l'aplicació web

Inicio de sesión

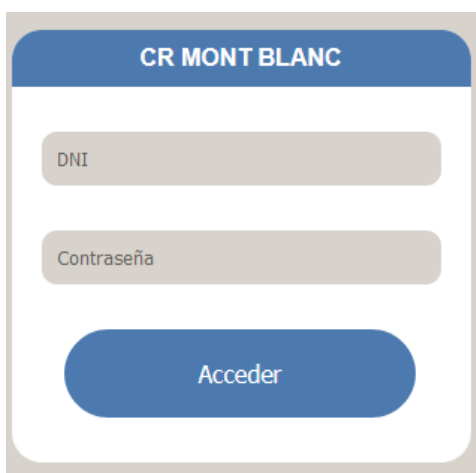


Figura 1 → Inicio de sesión

Introducir el usuario y la contraseña que corresponde al trabajador

Inicio de sesión incorrecto



Figura 2 → Inicio de sesión incorrecto

Si el usuario introduce un usuario o contraseña incorrectos se le muestra el mensaje de error de la figura 2, para volver al inicio de sesión pulsar en el botón “volver al login”.

Página de elección de comunidad



Figura 3 → Elección de la comunidad

La página de la figura 3 solo aparecerá a los trabajadores que trabajen para las dos comunidades, para los trabajadores que sean miembros únicamente de una de las dos comunidades la web les direccionará directamente al panel de control de la comunidad que les corresponda.

Panel de Control

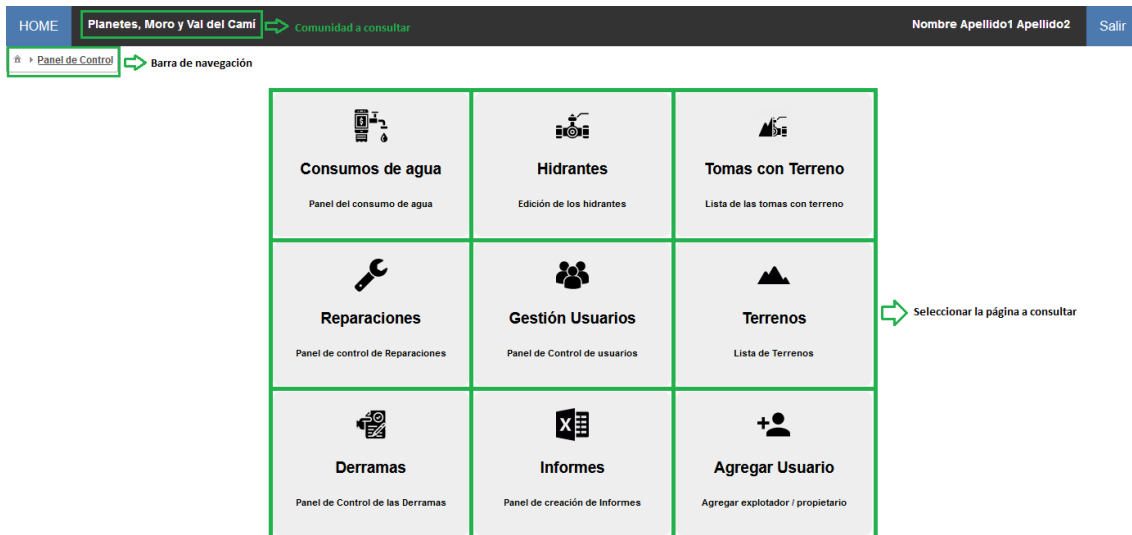


Figura 4 → Panel de Control de la comunidad

En la página de la figura 4 podemos seleccionar la página que queremos consultar.

Consumos de agua

En la pantalla de consumos de agua el trabajador puede consultar los consumos de agua que existen en el sistema. Tiene la opción de filtrar por columnas, introducir una merma y publicar un consumo una vez revisado.

HOME Planetes, Moro y Val del Cami Minerva Rafal Ferre Salir

Panel de Control Consumos de agua

Listado de Consumos (1 of 4)

Lectura de	Fecha Lectura Real	Hidrante	Toma	Explotador	Consumo Acumul	Consumo Real (m ³)	Precio (m ³)	Merma (m ³)	Precio a pagar	Publicado
ene-2018	2018-02-17	NH46	2	Antonio Garcia Lec	5785.0	0.0	0.1€	0.0	0.0€	<input checked="" type="checkbox"/>
mar-2018	2019-02-17 2018-0	NH46	2	Antonio Garcia Lec	5985.0	200.0	0.1€	0.0	20.0€	<input checked="" type="checkbox"/>
abr-2018	2018-04-06 2018-0	NH46	2	Antonio Garcia Lec	6085.0	100.0	0.1€	0.0	10.0€	<input checked="" type="checkbox"/>
may-2018	2018-05-06 2018-0	NH46	2	Antonio Garcia Lec	6385.0	300.0	0.1€	0.0	30.0€	<input checked="" type="checkbox"/>
jun-2018	2018-06-06 2018-0	NH46	2	Antonio Garcia Lec	6485.0	100.0	0.1€	0.0	10.0€	<input checked="" type="checkbox"/>
jul-2018	2018-07-06 2018-0	NH46	2	Antonio Garcia Lec	6785.0	300.0	0.1€	0.0	30.0€	<input checked="" type="checkbox"/>
ago-2018	2018-08-06 2018-0	NH46	2	Antonio Garcia Lec	6885.0	100.0	0.1€	0.0	10.0€	<input checked="" type="checkbox"/>
sep-2018	2018-09-06 2018-1	NH46	2	Antonio Garcia Lec	6985.0	100.0	0.1€	0.0	10.0€	<input checked="" type="checkbox"/>
oct-2018	2018-10-06 2018-1	NH46	2	Antonio Garcia Lec	7085.0	100.0	0.1€	0.0	10.0€	<input checked="" type="checkbox"/>
nov-2018	2018-11-06 2018-1	NH46	2	Antonio Garcia Lec	7185.0	100.0	0.1€	0.0	10.0€	<input checked="" type="checkbox"/>

Figura 5 → Consumos de agua

El usuario tiene la opción de exportar la tabla seleccionada.

Hidrantes

HOME Planetes, Moro y Val del Cami Nombre Apellido1 Apellido2 Salir

Panel de Control Hidrantes

Agregar Hidrante Ir a la página de agregar hidrante

Exportar tabla seleccionada en excel o PDF

Listado de hidrantes (1 of 2)

Hidrante	Coordenadas	Eliminar Hidrante
HN78	7.287534;4.476456	<input checked="" type="checkbox"/> Eliminar hidrante de la fila seleccionada
H46 <small>Filtrar por hidrante</small>	2.287534;9.476456	<input type="checkbox"/>
H87	2.287534;9.476456	<input type="checkbox"/>
HN56	2.287534;9.476456	<input type="checkbox"/>
H202	2.287534;9.476456	<input type="checkbox"/>
H98	2.287534;9.476456	<input type="checkbox"/>
H64	2.287534;9.476456	<input type="checkbox"/>
H77	2.287534;9.476456	<input type="checkbox"/>
H208	2.287534;9.476456	<input type="checkbox"/>
H55	2.287534;9.476456	<input type="checkbox"/>

Figura 6 → Panel Hidrantes

En el panel de hidrantes podemos consultar los hidrantes que tiene la comunidad, filtrar por las diferentes columnas o eliminar un hidrante.

Si el hidrante tiene tomas asociadas en propiedad no nos dejara **eliminar el hidrante** (saldrá un mensaje de advertencia). Para poder eliminarlo primero tendremos que desvincular las tomas asociadas.

Tenemos la opción de **paginar la tabla** para mostrar más o menos elementos o mostrar los siguientes elementos.

Tenemos la opción de **exportar la tabla** en Excel o PDF.

Si queremos **agregar un nuevo hidrante** hacer click al botón "Agregar Hidrante".

Agregar hidrante

Nuevo Hidrante

Número de hidrante

Coordenadas

Agregar Hidrante

Figura 7 → Formulario para agregar un nuevo hidrante

En la página de agregar hidrante tenemos que rellenar los campos del nuevo hidrante.

Campos:

- **Número hidrante** → El formato de los hidrantes tiene que ser H001 o HN001.
- **Coordenadas** → El formato de las coordenadas tiene que ser 17.534;14.47646.

Tomas con terrenos

id	Explotador	Propietario	Poligono	Sub-parcela	Superficie (Ha)	Termino municij	Fecha de conce	Hidrante	Toma	Desvincular toma
15	Rosalía Argumo	Angel Torres Pa	12	a	0.14	Nonaspe	2019-01-11	H87	-1	
7	Antonio Garcia I		1	0	0.9632	Nonaspe	2019-03-03	H202	0	
12	Rosalía Argumo		1	0	0.0025	Nonaspe	2019-03-03	H61	0	
3	Antonio Garcia I		1	0	124.0	Nonaspe	2019-03-03	H121	1	
8	Jose Salinas Mo		1	0	2.2445	Nonaspe	2019-03-03	H87	1	
13	Jose Salinas Mo		1	0	2.3465	Nonaspe	2019-03-03	H61	1	
14	Pedro Lopez Ca		1	0	3.1243	Nonaspe	2019-03-03	H61	2	
4	Antonio Garcia I		4	1	8.4878	Nonaspe	2019-03-03	H46	2	
11	Explotador Dos		1	0	2.5555	Nonaspe	2019-03-03	H202	2	
5	Pedro Lopez Ca		2	0	0.1587	Nonaspe	2019-03-03	H87	3	

Figura 8 → Panel de edición tomas con terrenos

En el Panel de tomas con terreno podemos consultar las tomas que pertenecen a cada terreno/explotador. Los usuarios tienen permisos para filtrar por columnas.

Las tomas marcadas de color rojo **IMPORTANTE** no pertenecen a ninguna toma y salen en número negativo ya que no están vinculadas a ningún terreno. Los trabajadores tienen que asignar el número de toma al que corresponden ya que si no esos terrenos no tendrán consumos de agua ya que no corresponderán a ningún contador vinculado.

La última columna es para **desvincular la toma del terreno**.

Tenemos la opción de **paginar la tabla** para mostrar más o menos elementos o mostrar los siguientes elementos.

Tenemos la opción de **exportar la tabla** en Excel o PDF.

Reparaciones

Contabilizado	Reparacion	Fecha Reparacion	Consumo Antiguo	Consumo Nuevo	Hidrante	Toma	Explotador de la Toma
No	1	2019-06-10	120000.0	0.0	NH121	1	Minerva Rafal Ferre

Figura → Pantalla de reparaciones

En la pantalla de reparaciones el usuario tiene la opción de consultar las reparaciones realizadas y filtrarlas por columnas.

El usuario tiene la opción de exportar el listado de reparaciones en Excel.

Para generar una reparación nueva el usuario debe hacer click en “Generar Reparación”.

Generar Reparación

Fecha Reparacion:

Hidrante:

Toma:

Consumo contador antiguo

Consumo contador nuevo

Figura → Generar Reparación

En el formulario de generar derrama el usuario tiene que completar los siguientes campos:

Fecha reparación → En este campo el trabajador tiene que introducir la fecha en la que se está cambiando el contador averiado.

Hidrante → Hidrante que se va a reparar

Toma → Toma a la que se va a cambiar el contador

Consumo contador antiguo → Lectura del contador antiguo para que al introducir un consumo nuevo el sistema pueda recalculer el consumo real.

Consumo contador nuevo → Cifra en la que el contador nuevo empieza a contar. Si el contador es nuevo introducimos el valor 0, si se está utilizando un contador reutilizado introducir el valor que marca el contador reutilizado.

Agregar Usuario

Figura 10 → Formulario para agregar un usuario

En la página de la figura 10 tenemos que rellenar el formulario para agregar un nuevo usuario.

Campos:

- **Tipo de usuario** → Elegir el tipo de usuario que queremos agregar: explotador, propietario, ambos.
- **Amortización** → Marcar este campo si el usuario tiene el préstamo amortizado (para aplicar las derramas con amortización).
- **Nombre** → Introducir nombre del explotador.
- **Apellidos** → Introducir apellidos del explotador.
- **DNI** → Introducir DNI con letra. Formato:12345678A
- **Contraseña** → El sistema genera una contraseña automáticamente (guardar la contraseña).
- **Dirección** → Introducir la dirección del explotador.
- **Población** → Introducir la población del explotador.
- **Código Postal** → Introducir el código postal de la población del explotador.
- **Provincia** → Introducir la provincia del explotador.
- **Email** → Introducir el correo electrónico del explotador.
- **Teléfono** → Introducir un teléfono de contacto del explotador.
- **Número de cuenta** → Introducir el número de cuenta del explotador para facturar consumos.
- **Entidad bancaria** → Nombre de la entidad bancaria del explotador.

- **Tipo adeudo** → Tipo adeudo del explotador
- **Fecha firma** → Fecha de alta del explotador a la comunidad.

Gestión de Usuarios

HOME Planetes, Moro y Val del Camí Nombre Apellido1 Apellido2 Salir

Panel de Control > Gestión de Usuarios

Listado de Usuarios (1 of 1)

id	Tipo de Usuario	Nombre	DNI	Direccion	Poblacion	Codigo Pk	Provincia	Email	Telefono	Número de	Entidad ban	Tipo adeudo	Fecha firma	Amortizació	Eliminar usu
6	explotador	Antonio Garcia	12345678Z	Carrer Sant Mig	nonaspe	43740	Zaragoza	test4@gma	638874582	ES6524713	Bantierra	rcur	2019-02-02	<input checked="" type="checkbox"/>	
7	explotador	Pedro Lopez C	12785678Z	Carrer Sant Fer	nonaspe	43740	Zaragoza	test5@gma	638874582	ES1845613	OpenBank	rcur	2019-02-02	<input type="checkbox"/>	
9	explotador	Rosalia Argum	79100347Y	Carrer Sant Jos	nonaspe	43740	Zaragoza	test7@gma	638874582	ES1005613	Bantierra	rcur	2019-02-02	<input type="checkbox"/>	
10	explotador	Jose Salinas II	85236978R	Carrer Sant Pac	nonaspe	43740	Zaragoza	test8@gma	638874582	ES1995613	OpenBank	rcur	2019-02-02	<input type="checkbox"/>	
11	propietari	Angel Torres P	78496787Z	Carrer Sant Jor	fabara	43740	Zaragoza	test10@gm	638874582	ES1997213	Bantierra	rcur	2019-02-02	<input type="checkbox"/>	
13	propietari	Hugo Julio Cor	88457888R	Carrer Santa M	nonaspe	43740	Zaragoza	test12@gm	638874582				2019-02-02	<input type="checkbox"/>	
14	propietari	Joaquin Cortes	36556677C	Carrer Santa M	nonaspe	43740	Zaragoza	test13@gm	638874582				2019-02-02	<input type="checkbox"/>	
15	explotador	Explotador Dos	8546972R	Carrer Sant Juli	Reus	43740	Tarragona	test814@gr	638874582				2019-02-02	<input type="checkbox"/>	

Figura 11 → Panel de gestión de usuarios

En la página de la figura 11 podemos consultar todos los usuarios explotadores y propietarios que tenemos en la comunidad y podemos editar su información o eliminar el usuario. No se podrán eliminar usuarios con terrenos vinculados.

Terrenos

HOME Planetes, Moro y Val del Camí Nombre Apellido1 Apellido2 Salir

Panel de Control > Terrenos

Agregar un nuevo terreno

Listado de Terrenos (1 of 2)

id	Explotador	Propietario	Poligono	Sub-parcela	Superficie (Ha)	Termino municipal	Fecha de concesion	Eliminar terreno
3	Antonio Garcia Leor		1	0	124.0	Nonaspe	2019-03-03	
4	Antonio Garcia Leor		4	1	8.4878	Nonaspe	2019-03-03	
5	Pedro Lopez Cabrer		2	0	0.1587	Nonaspe	2019-03-03	
6	Rosalia Argumosa L		1	0	0.9987	Nonaspe	2019-03-03	
7	Antonio Garcia Leor		1	0	0.9632	Nonaspe	2019-03-03	
8	Jose Salinas Morenc		1	0	2.2445	Nonaspe	2019-03-03	
9	Angel Torres Pascua		1	0	4.5203	Nonaspe	2019-03-03	
10	Angel Torres Pascua		1	0	2.7895	Nonaspe	2019-03-03	
11	Explotador Dos Com		1	0	2.5555	Nonaspe	2019-03-03	
12	Rosalia Argumosa L		1	0	0.0025	Nonaspe	2019-03-03	

Figura 12 → Panel de Terrenos

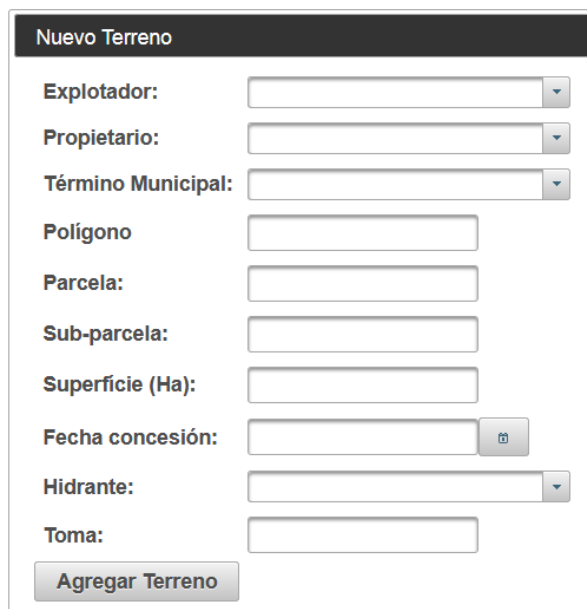
En la página de la figura 12 podemos ver el listado de terrenos de la comunidad, tenemos la opción de filtrar por columnas o eliminar un terreno. Solo se podrán eliminar los terrenos que no estén vinculados a ninguna toma. Tenemos la opción de modificar la información de los terrenos.

Tenemos la opción de **paginar la tabla** para mostrar más o menos elementos o mostrar los siguientes elementos.

Tenemos la opción de **exportar la tabla** en Excel o PDF.

Para agregar un terreno tenemos que hacer click al link de “Agregar Terreno”.

Agregar Terreno



Formulario "Nuevo Terreno" con los siguientes campos:

- Explotador:
- Propietario:
- Término Municipal:
- Polígono:
- Parcela:
- Sub-parcela:
- Superficie (Ha):
- Fecha concesión:
- Hidrante:
- Toma:

Figura 13 → Formulario para agregar un terreno

En la página de la figura 13 tenemos que rellenar el formulario para agregar un terreno.

Campos:

Explotador → Elegir un explotador de la lista desplegable (el explotador tiene que estar añadido).

Propietario → Elegir un propietario si es necesario guardar el propietario del terreno (el propietario tiene que estar añadido).

Término Municipal → Elegir un término municipal para el terreno.

Polígono → Introducir el polígono que identifica al terreno.

Parcela → Introducir la parcela que identifica al terreno.

Sub-parcela → Introducir la sub-parcela (si es necesario) que identifica al terreno.

Superficie → Introducir la superficie en hectáreas.

Fecha concesión → Introducir la fecha concesión del terreno.

Hidrante → Introducir el hidrante al que pertenece el terreno.

Toma → Introducir la toma a la que pertenece el terreno (no es un campo obligatorio ya que se puede completar en la página de tomas con terrenos).

Derramas

HOME Planetes, Moro y Val del Cami Nombre Apellido1 Apellido2 Salir

Panel de Control Derramas

Generar Derrama Ir a la página de generar derrama

Listado de Derramas (1 of 1)

Derrama	Explotador	Fecha cobro	Fecha creacion	Canon	Ejercicio	Importe	Ver Derrama
7;8;	Angel Torres Pascual	2019-05-09	2019-05-18	150.0	2019	1096.46	
5;1;2;	Antonio Garcia Leon	2019-05-09	2019-05-18	75.0	2019	10008.82	
9;	Explotador Dos Comun	2019-05-09	2019-05-18	150.0	2019	383.32	
6;11;	Jose Salinas Moreno	2019-05-09	2019-05-18	150.0	2019	688.64	
3;12;	Pedro Lopez Cabrera	2019-05-09	2019-05-18	150.0	2019	492.44	
4;10;	Rosalia Argumosa Lop	2019-05-09	2019-05-18	150.0	2019	150.17	

(1 of 1)

Figura 14 → Panel de derramas

A la página de la figura 14 podemos ver el listado de derramas que se han generado, tenemos la opción de filtrar por columnas. En la columna de la derecha tenemos la opción de abrir o descargar la derrama seleccionada. Tenemos la opción de **paginar la tabla** para mostrar más o menos elementos o mostrar los siguientes elementos.

Tenemos la opción de **exportar la tabla** en Excel.

COMUNIDAD DE REGANTES PLANETES MORO Y VAL DEL CAMI <hr/> Avda. Orbe Cano, s/n. 50794 NONASPE (Zaragoza) NIF V50889336	Angel Torres Pascual Carrer Sant Jorge 43740 Fabara Zaragoza
--	--

Nonaspe, sábado, 18 de mayo de 2019

Estimado regante:

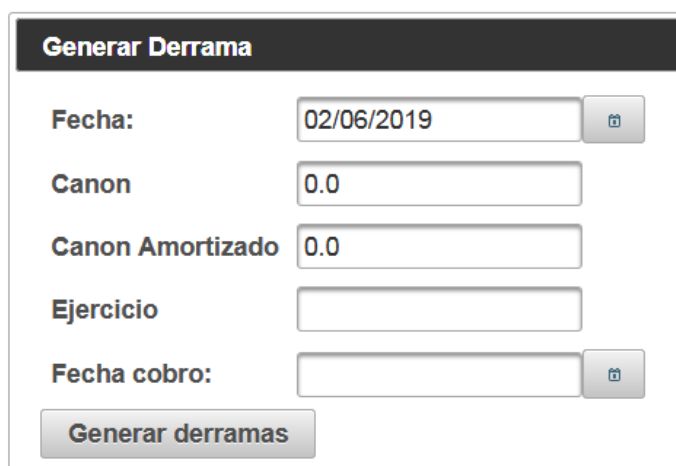
El motivo de la presente es informarle que el próximo 09/05/2019 se pasará al cobro el importe abajo indicado a la derrama por superficie del presente ejercicio.

EJERCICIO: 2019 CANON/HA: 150.0 €/ha

Hidrante	Toma	Poligono	Parcela	Sub-parcela	Superficie	Término	Fecha concesión	Importe
H98	3	1	967	0	4.5203	Nonaspe	2019-03-03	678.04 €
H64	4	1	987	0	2.7895	Nonaspe	2019-03-03	418.42 €
TOTAL: 1096.46 €								

Figura 15 → Ejemplo de derrama en PDF

Generar Derrama



Formulario para generar una derrama. El formulario tiene un encabezado "Generar Derrama" en un recuadro negro. Debajo hay cinco campos de entrada: "Fecha:" con el valor "02/06/2019" y un icono de calendario; "Canon" con el valor "0.0"; "Canon Amortizado" con el valor "0.0"; "Ejercicio" que está vacío; y "Fecha cobro:" que está vacío y tiene un icono de calendario. En la parte inferior hay un botón "Generar derramas".

Figura 16 → Formulario generar derrama

En la página de la figura 16 el usuario tiene que rellenar el formulario para generar una derrama.

Campos:

Fecha → Fecha actual.

Canon → Canon para explotadores sin amortización en euros.

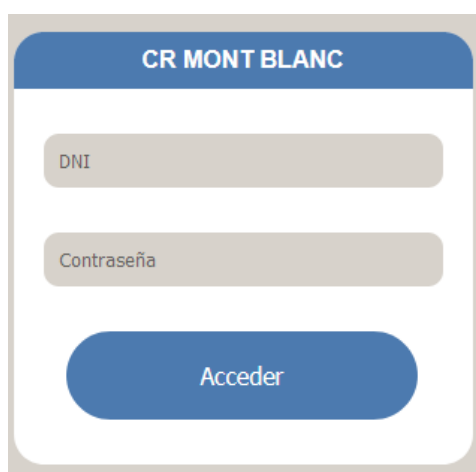
Canon amortizado → Canon para explotadores con préstamo amortizado en euros.

Ejercicio → Ejercicio al que hace referencia la derrama.

Fecha cobro → Introducir fecha en la que se pasara a cobrar la derrama.

10.2 Manual per als usuaris explotadors de l'aplicació web

Inicio de sesión



Formulario de inicio de sesión. El formulario tiene un encabezado "CR MONT BLANC" en un recuadro azul. Debajo hay dos campos de entrada: "DNI" y "Contraseña", ambos con un fondo gris. En la parte inferior hay un botón "Acceder" con un fondo azul.

Figura 1 → Inicio de sesión

Introducir el usuario y la contraseña que corresponde al trabajador

Inicio de sesión incorrecto



Figura 2 → Inicio de sesión incorrecto

Si el usuario introduce un usuario o contraseña incorrectos se le muestra el mensaje de error de la figura 2, para volver al inicio de sesión pulsar en el botón “volver al login”.

Página de elección de comunidad



Figura 3 → Elección de la comunidad

La página de la figura 3 solo aparecerá a los trabajadores que trabajen para las dos comunidades, para los trabajadores que sean miembros únicamente de una de las dos comunidades la web les direccionará directamente al panel de control de la comunidad que les corresponda.

Panel de control del usuario



Figura 4 → Panel de control

Panel de control donde el usuario puede elegir la pantalla que desea consultar.

Consumos de agua

En la pantalla de consumos de agua el usuario puede consultar el listado de lecturas de los diferentes contadores leídos el último mes.

Fecha Lectura Real	Hidrante	Toma	Consumo Real (m3)	Precio (m3)	Merma (m3)	Precio a pagar
2019-05-02 2019-06-06	NH61	2	400.0	0.1€	0.0	40.0€
2019-03-06 2019-04-06	NH46	2	100.0	0.1€	0.0	10.0€

Figura 5 → Listado de consumos del último mes

El usuario puede filtrar los registros por columnas.
El usuario tiene la opción de exportar el listado mostrado en Excel.

Historial de consumos

En la pantalla de historial de consumos, el usuario puede consultar el historial de todos los consumos guardados al sistema.

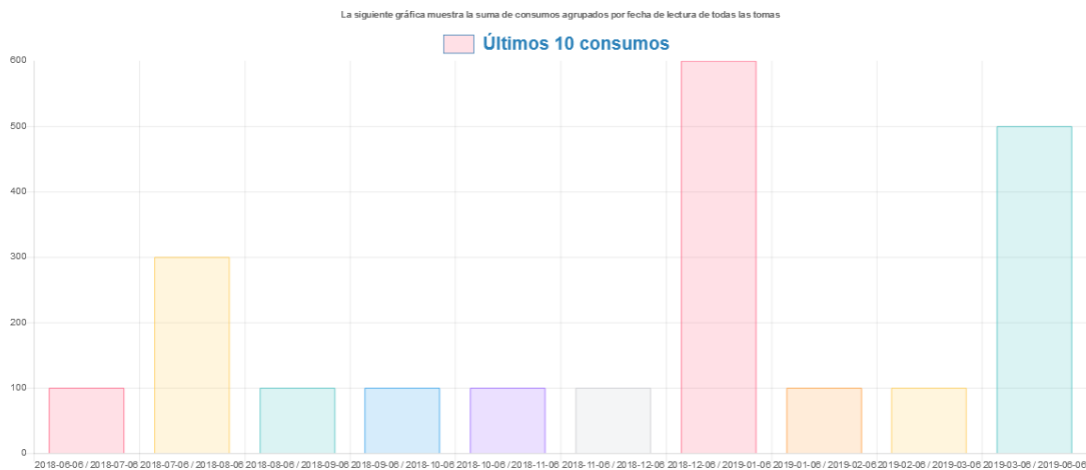


Figura 6 → Últimos 10 consumos

En esta pantalla el usuario puede consultar la gráfica de la figura 6 que muestra la suma de los últimos 10 consumos agrupados por fecha de lectura de todas las tomas.



Listado de Consumos						
(1 of 2) 1 2 10						
Fecha Lectura Real	Hidrante	Toma	Consumo Real (m3)	Precio (m3)	Merma (m3)	Precio a pagar
2019-05-02 2019-06-06	NH61	2	400.0	0.1€	0.0	40.0€
2019-03-06 2019-04-06	NH46	2	100.0	0.1€	0.0	10.0€
2019-02-06 2019-03-06	NH46	2	100.0	0.1€	0.0	10.0€
2019-01-06 2019-02-06	NH46	2	100.0	0.1€	0.0	10.0€
2018-12-06 2019-01-06	NH46	2	600.0	0.1€	0.0	60.0€
2018-11-06 2018-12-06	NH46	2	100.0	0.1€	0.0	10.0€
2018-10-06 2018-11-06	NH46	2	100.0	0.1€	0.0	10.0€
2018-09-06 2018-10-06	NH46	2	100.0	0.1€	0.0	10.0€
2018-08-06 2018-09-06	NH46	2	100.0	0.1€	0.0	10.0€
2018-07-06 2018-08-06	NH46	2	300.0	0.1€	0.0	30.0€

(1 of 2) 1 2 10

Figura 7 → Historial de consumos

En la tabla de historial de consumos el usuario puede consultar el historial de todos los consumos que tiene y filtrarlos por columnas.

El usuario tiene la opción de exportar el listado mostrado en Excel.

Derramas



Listado de Derramas					
(1 of 1) 1 10					
Fecha cobro	Fecha creacion	Canon (€/ha)	Ejercicio	Importe	Ver Derrama
2019-05-09	2019-05-18	75.0€	2019	10008.82€	

(1 of 1) 1 10

Figura 8 → Listado de derramas

El usuario puede consultar todas las derramas generadas de sus terrenos.

El usuario puede ver o descargar en pdf las derramas detalladas.

En la figura 9 podemos ver un ejemplo de derrama en pdf.

**COMUNIDAD DE REGANTES PLANETES
MORO Y VAL DEL CAMI**

Avda. Orbe Cano, s/n. 50794
NONASPE (Zaragoza) NIF V50889336

Antonio Garcia Leon

Carrer Sant Miguel
43740 Nonaspe
Zaragoza

Nonaspe, sábado, 18 de mayo de 2019

Estimado regante:

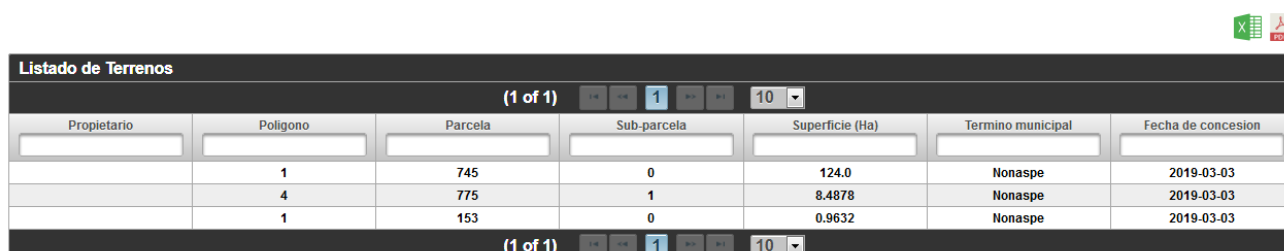
El motivo de la presente es informarle que el próximo 09/05/2019 se pasará al cobro el importe abajo indicado a la derrama por superficie del presente ejercicio.

EJERCICIO: 2019 CANON/HA: 75.0 €/ha

Hidrante	Toma	Poligono	Parcela	Sub-parcela	Superficie	Término	Fecha concesión	Importe
NH121	1	1	745	0	124.0	Nonaspe	2019-03-03	9300.0 €
NH46	2	4	775	1	8.4878	Nonaspe	2019-03-03	636.58 €
NH202	0	1	153	0	0.9632	Nonaspe	2019-03-03	72.24 €
TOTAL: 10008.82 €								

Figura 9 → Derrama en pdf detallada por hidrantes y tomas

Terrenos



Propietario	Poligono	Parcela	Sub-parcela	Superficie (Ha)	Termino municipal	Fecha de concesion
	1	745	0	124.0	Nonaspe	2019-03-03
	4	775	1	8.4878	Nonaspe	2019-03-03
	1	153	0	0.9632	Nonaspe	2019-03-03

Figura 10 → Listado de terrenos del usuario

En la pantalla de terrenos el usuario puede consultar los terrenos que tiene en concesión y filtrar los datos por columnas.

El usuario tiene la opción de exportar la tabla seleccionada en Excel o PDF.

10.3 Manual per als usuaris de l'aplicació Android

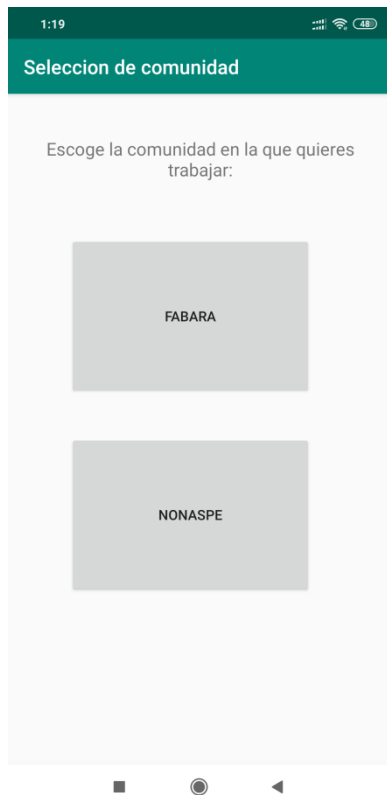


Figura 1 → Al abrir la aplicación nos aparece la pantalla de la figura 1 en la que podemos seleccionar la comunidad en la que queremos trabajar.



Figura 2 → En la pantalla de la figura 2 nos aparece el panel de control de la comunidad seleccionada, en ella podemos entrar en las pantallas de consumos, enviar datos al servidor o actualizar la base de datos desde el servidor.

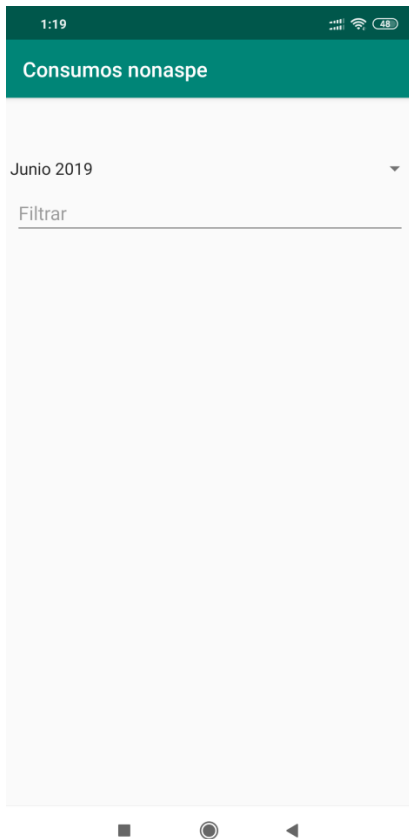


Figura 3 → Una vez seleccionada la opción de consumos, nos aparecen los diferentes hidrantes de la base de datos del servidor. Si no nos aparecen consumos como en el ejemplo de la figura 3 es porque todavía no se a actualizado la base de datos desde el servidor.



Figura 4 → Para actualizar la base de datos tenemos que pulsar el botón “actualizar DB desde el servidor” marcado en el cuadro naranja.

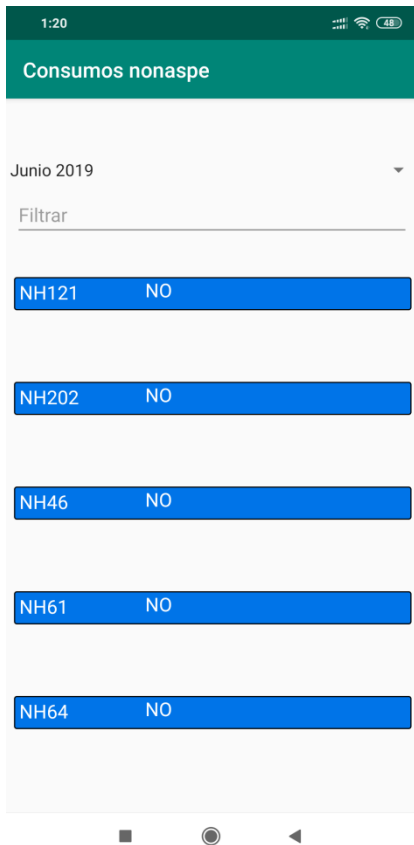


Figura 5 → A continuación al entrar en la pantalla de consumos podemos ver como se han actualizado los hidrantes desde la base de datos del servidor. Ahora ya podemos hacer la lectura de consumos con los hidrantes y tomas actualizados del servidor.



Figura 6 → Si seleccionamos la opción que está marcada en naranja nos dará acceso a un menú desplegable donde podremos elegir el mes que deseamos hacer la lectura o consultar lecturas anteriores.

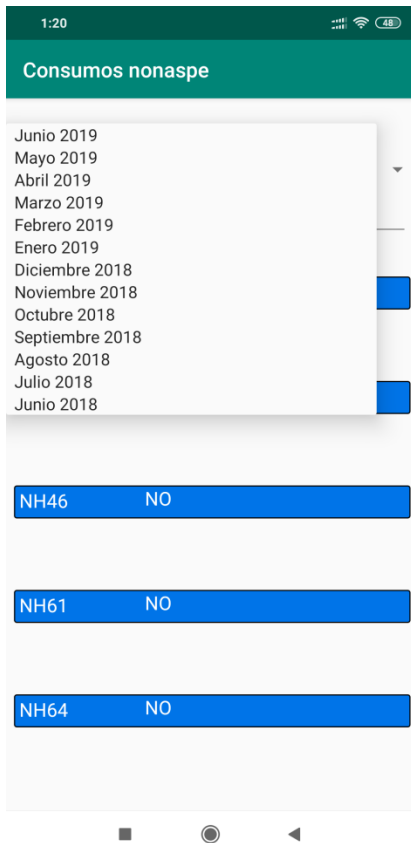


Figura 7 → Seleccionamos en el menú desplegable el mes que deseamos leer y/o consultar.

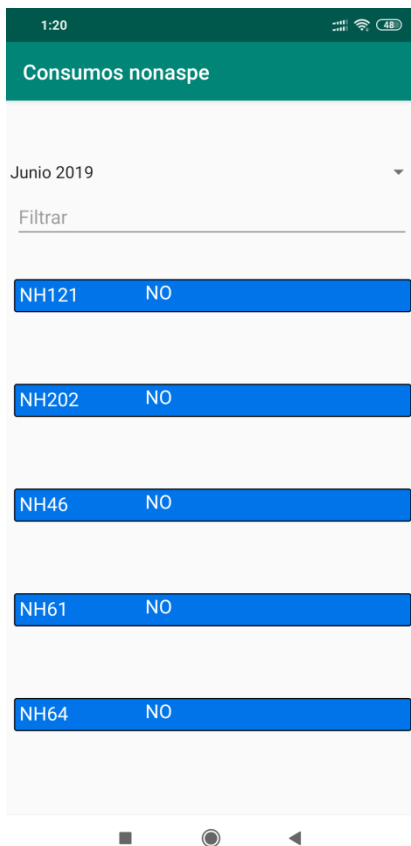


Figura 8 → En la pantalla de la figura 8 seleccionamos el hidrante que queremos leer i/o consultar.

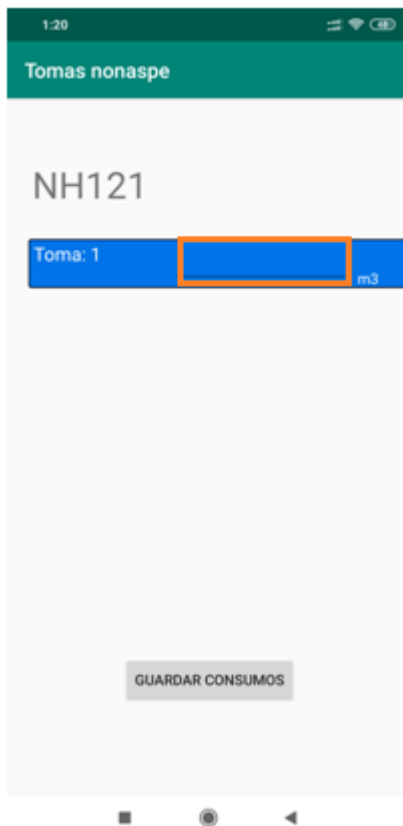


Figura 9 → Pantalla de las tomas del hidrante seleccionado. Para introducir un consumo nuevo, damos un toque con el dedo en la zona marcada en naranja de la toma que queremos leer.

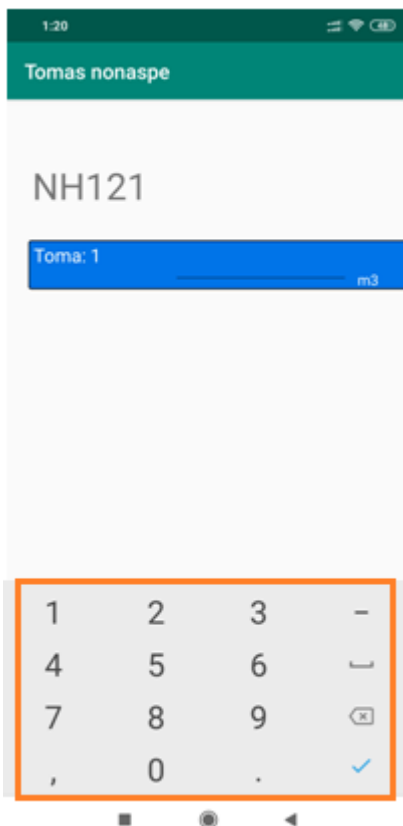


Figura 10 → A continuación nos aparecerá un teclado numérico donde tenemos que marcar el consumo registrado en el contador. Una vez introducido el consumo, tenemos que hacer clic al visto del teclado.

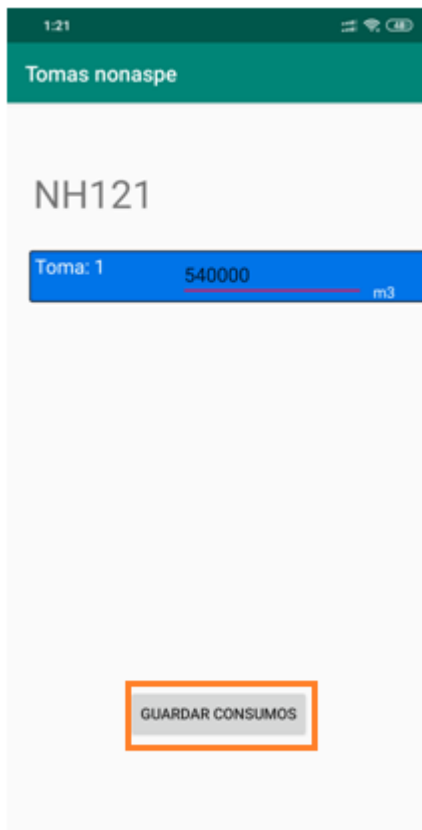


Figura 11 → Para que se guarde el consumo en la aplicación Android tenemos que dar un toque al botón marcado en naranja “Guardar consumos”.



Figura 12 → Una vez guardado el/los consumos en la aplicación, la pantalla de la figura 12 nos marcara con un “SI” los hidrantes con los consumos introducidos tal y como se marca en el rectángulo naranja.



Figura 13 → Una vez guardados todos los consumos de los hidrantes leídos tenemos que enviar los datos al servidor para que se transfieran los datos de la aplicación Android a la base de datos de nuestro servidor web.

Para enviar los datos al servidor web, damos un toque al botón “Enviar datos al servidor” marcado en naranja.

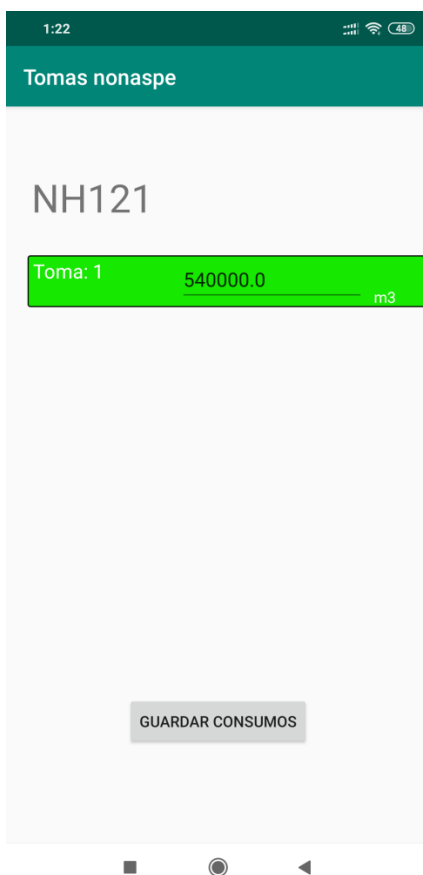


Figura 14 → Si se han sincronizado los datos de la aplicación Android al servidor web correctamente, las tomas leídas nos aparecerán marcadas en color verde.