



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

APLICACIÓN ANDROID PARA LA GESTIÓN DE RESERVAS Y ASIGNACIÓN DE AGUA DE RIEGO A PARCELAS AGRARIAS

Trabajo Fin de Grado
Grado en Ingeniería Informática

Autor: Jose Ibáñez García
Tutor: Joaquim Francesc Arlandis Navarro
2016/2017

Resumen

En determinadas partidas del *País Valencià*, el reparto del agua de riego entre las parcelas agrarias en producción está sujeto a restricciones como la estructura de los canales del sistema de regadío, la disponibilidad de los recursos y la petición puntual de consumo expresada por los particulares que gestionan las parcelas a regar.

En este trabajo se ha diseñado y desarrollado una aplicación para dispositivos móviles con sistema operativo Android, para la automatización del protocolo actual de reserva y asignación de agua a las parcelas, destinada principalmente a las personas que las gestionan y que, además, permita al regante tener la información lo más actualizada posible, con el fin de gestionar mejor su tiempo y llevar un control claro y fiable de las reservas existentes.

Palabras clave: aplicación Android, gestión de turnos, riego, Java, Android, PHP, Apache, MySQL

Abstract

On certain plots of ground in *País Valencià*, the allocation for water for irrigation among agricultural parcels in production may be subject to restrictions such as the structure of the system of the channels for irrigating the land, the availability of resources and the request of consumption expressed by individuals who deal with the plots of land when it comes to watering.

This project consists of designing and developing an application suited for mobile devices with Operating System such as Android. Its main purpose is based on the automation of the existing protocol of the reservoirs and water allotment to the plots of ground, which is allocated by people who process them. What is more, it also allows the irrigator to have the most up-to-date information with the purpose of managing their free time and control clearly and trustworthily the existing reservoirs.

Keywords : Android application, turn management, irrigation, Java, Android, PHP, Apache, MySQL

Tabla de contenidos

1. Introducción	9
1.1. Contexto socio-cultural	9
1.2. Estructura del sistema de regadío	10
1.3. Motivación	12
1.4. Objetivos	12
1.5. Estructura de la memoria	13
2. Tecnologías	14
2.1. Arquitectura del sistema	14
2.2. Software utilizado	15
2.3. Hardware utilizado	18
3. Obtención de requisitos	20
3.1. Toma de requisitos	20
3.2. Definiciones	20
3.3. Actores del sistema	21
3.4. Características de la aplicación	22
3.5. Requisitos no funcionales	23
4. Especificación de requisitos	24
4.1. Diagrama de casos de uso de los responsables	24
4.2. Diagrama de casos de uso del regante	30
5. Diseño de la aplicación	36
5.1. Capa de persistencia	36
5.2. Capa lógica	37
5.3. Diagrama de flujo	39
6. Desarrollo de la aplicación	41
6.1. Estructura del proyecto Android	41
6.2. Web service	43
6.3. Librerías externas	46
7. Funcionamiento de la aplicación	47
8. Conclusiones y trabajo futuro	60
8.1. Conclusiones	60
8.2. Trabajo futuro	61



9. Bibliografía.....62



Índice de figuras

Figura 1 - Estructura de la partida de riegos	11
Figura 2 - Arquitectura del sistema	14
Figura 3 - Casos de uso del responsable	24
Figura 4 - Casos de uso del regante	30
Figura 5 - Modelo relacional.....	36
Figura 6 - Diagrama de clases	37
Figura 7 - Diagrama de flujo	40
Figura 8 - Paquetes de la aplicación	41
Figura 9 - Modelos	41
Figura 10 - Controladores	42
Figura 11 - Web service	42
Figura 12 - Conversores	42
Figura 13 - Adaptadores.....	43
Figura 14 - Uso de GET en el script PHP para obtener los anuncios	44
Figura 15 - Uso de POST en el script PHP de añadir reserva	44
Figura 16 - Implementación de una petición GET	45
Figura 17 - Implementación de una petición POST	46
Figura 18 - Inicio de sesión	47
Figura 19 - Selección de parcela	48
Figura 20 - Pantallas principales.....	49
Figura 21 - Realizar reserva.....	50
Figura 22 - Visualizar y cancelar reserva	51
Figura 23 - Menú lateral	52
Figura 24 - Menú principal	53
Figura 25 - Listado de reservas	54
Figura 26 - Reservar	55
Figura 27 - Horarios de los brazos	56
Figura 28 - Nuevo anuncio	57
Figura 29 - Información del responsable.....	58
Figura 30 - Información de la parcela	59

1. Introducción

En la zona valenciana, el riego en las diversas partidas está sujeto a múltiples restricciones. Por ello, se pretende agilizar el proceso de petición y asignación de agua para el riego de las distintas parcelas utilizando una tecnología actual y muy extendida, como son las aplicaciones para dispositivos Android.

1.1. Contexto socio-cultural

Desde finales del siglo XIX, en la zona valenciana es abundante la agricultura de regadío, siendo los naranjos el cultivo más abundante. Comúnmente, para el riego de estos cultivos se utiliza el llamado riego a manta, que consiste en cubrir con agua toda la superficie del campo, por lo que se requiere una gran cantidad de agua para su riego. Otro problema es la existencia de gran cantidad de usuarios, debido a la micro-parcelación, lo que provoca que haya mucha demanda para unos recursos que son limitados. Éste problema se acentúa durante los meses de Julio y Agosto, ya que es necesario el riego de cada parcela al menos una vez cada dos semanas.

Para la gestión del agua para su distribución entre parcelas, se requiere, entre otras tareas, el establecimiento de horarios para el reparto del caudal entre las acequias, así como algún protocolo de reserva de turnos de riego de las parcelas que, frecuentemente, requiere la interacción entre la persona encargada del riego de la parcela, conocido como responsable de la parcela, y el encargado público del reparto suspensión de los recursos hídricos, conocido como regante. Ésta actividad sigue haciéndose de forma muy poco ágil, ya que es necesario ponerse en contacto con el regante por teléfono o en persona para comunicar las intenciones de regar tu parcela.

En un mundo ampliamente interconectado mediante Internet, sería útil una herramienta que permita agilizar todo el proceso de reservas, lo cual traería un mayor control en el reparto de los recursos hídricos, aspecto importante en una zona donde habitualmente se producen periodos de sequía y no se puede derrochar el agua. Otro aspecto importante es que la gran mayoría de las personas poseen actualmente un smartphone y acceso a Internet, por lo que dicha herramienta les permitiría disponer de información en tiempo real para, por ejemplo, poder actuar en caso de que surja algún imprevisto que impida regar.

Aún con todo esto, hay que tener en cuenta que también es posible que exista algún responsable del riego de una parcela que no utilice móvil, por lo que, en algunos casos, seguirá siendo necesario contactar con el regante igualmente. Aún con éste handicap, las ventajas que aportaría un

sistema que permita hacer estas tareas de forma sencilla mediante un dispositivo son sustanciales, por lo cual sería un sistema altamente útil.

1.2. Estructura del sistema de regadío

Para el desarrollo del presente trabajo, se ha tomado como referencia la partida de riego y estructura de acequias y parcelas del *Mareny de Sant Llorenç* de Cullera.

En cuanto a su estructura, esta partida se compone de una acequia principal (acequia *Sant Salvador*), la cual se divide en cuatro acequias secundarias, las cuales se dividen a su vez en brazos.

Son las acequias principales las que nacen directamente en el río y acaban dividiéndose en varias acequias secundarias. No son las acequias quienes suministran el agua directamente a las parcelas, sino que se dividen en distintos brazos, los cuales, formando una estructura en árbol, son los que suministran el agua directamente a las parcela.

Habitualmente, solo circula agua por un brazo a la vez. Puede darse el caso que hayan varios brazos por donde circule agua al mismo tiempo, siempre y cuando los recursos hídricos sean suficientes, pero al no ser una situación habitual, este trabajo se implementará para cubrir la situación más habitual. Para ello será necesario establecer horarios en la semana durante el cual circulará agua por un brazo.

A continuación, en la Figura 1, nos encontramos dos imágenes. En primer lugar (arriba) con está la acequia de *Sant Salvador* (color rojo), la cual se divide en cuatro acequias secundarias (color azul). En la imagen inferior se muestra una acequia secundaria con sus distintos brazos.

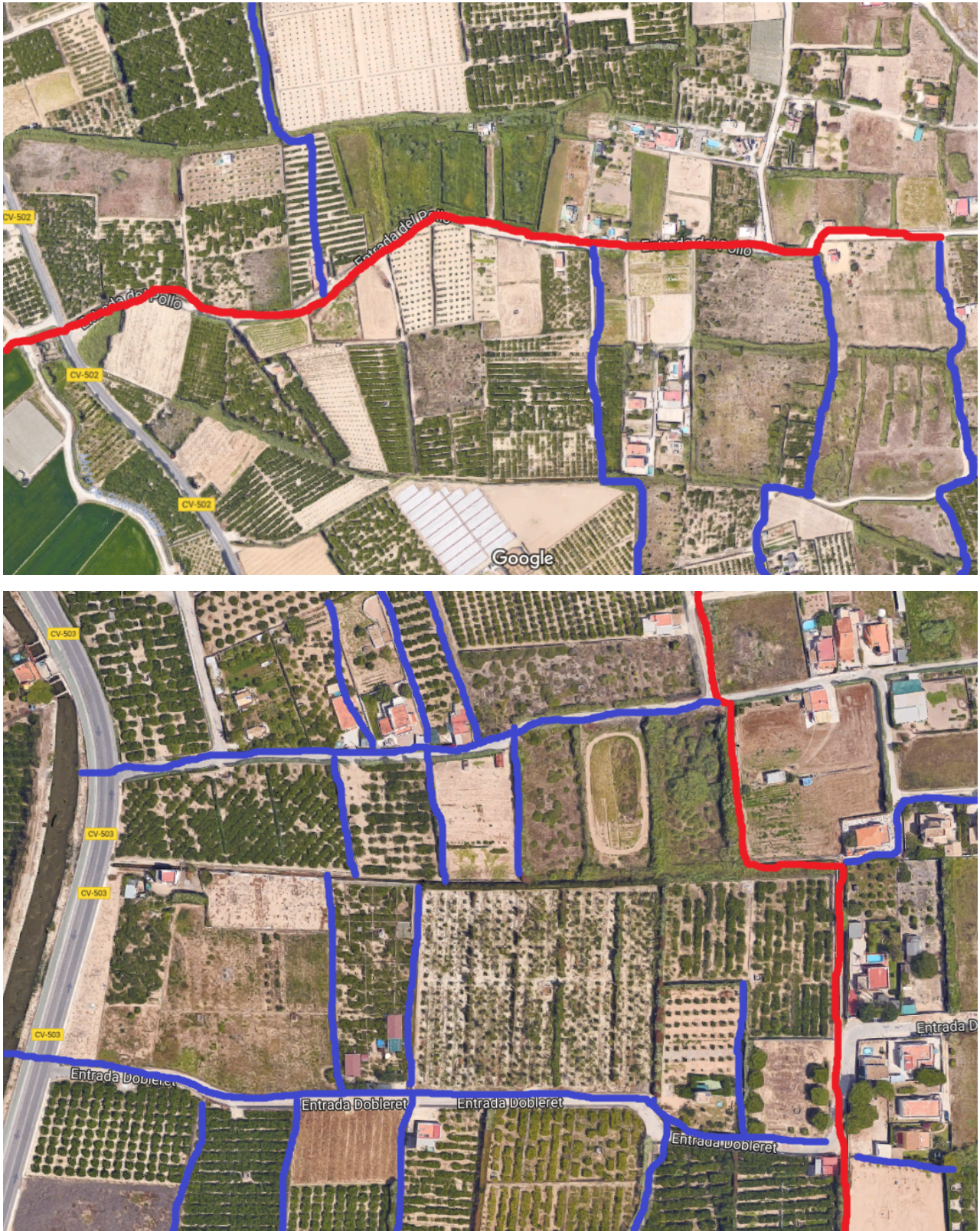


Figura 1 - Estructura de la partida de riegos

1.3. Motivación

Seleccioné este proyecto debido a diversos motivos. En primer lugar, me resulta interesante todo lo relacionado con la agricultura y el regadío, ya que, el pueblo donde he pasado toda mi infancia, Anna, es mayormente agrícola, y para mí es un campo conocido, ya que mi abuelo, ha sido durante muchos años el regante de una partida de riegos.

En segundo lugar, está el aspecto técnico. La programación de aplicaciones Android me parece, hoy en día, un campo bastante interesante y del cual se puede sacar mucho partido. Además, tenía un ligero conocimiento de ésta tecnología, por lo que vi una buena oportunidad para seguir ampliando conocimientos.

1.4. Objetivos

Desarrollar una aplicación para el sistema operativo Android para gestionar el proceso de reserva de recursos hídricos, que sea sencilla y haciendo dicho proceso más ágil para los usuarios del riego y mejorando el control que actualmente tiene el regante.

Se pretende que la aplicación permita a los usuarios responsables del riego de parcelas reservar un espacio de tiempo durante el cual tendrán disponibles los recursos para el regadío. Para dicho desarrollo tenemos dos partes bastante diferenciadas: la aplicación que se instalará en un dispositivo móvil Android y el servidor donde se almacenan los datos, que será accesible a través de Internet.

Se estudiarán las tecnologías más adecuadas para llevar a cabo dicho desarrollo, buscando el equilibrio entre las funciones que ofrece y la facilidad de uso.

En cuanto a la aplicación en sí, se desarrollará una única aplicación que cubra tanto las necesidades de los responsables y del regante.

Será el regante quien tenga, entre otras acciones, que registrar las reservas de los usuarios que contacten con él de la forma tradicional. Esto permitirá mantener la información de las franjas de tiempo disponibles para poder realizar la reserva actualizada y fiable. El regante deberá tener también a su disposición el listado de reservas programadas, pudiendo visualizar sus detalles.

Además de éstas funciones principales, el regante debe ser quien se encargue de la gestión los perfiles de los responsables de las parcelas, contemplando todas las posibilidades que ésta conlleva: el cambio de usuario responsable del riego de una parcela a un usuario ya existente en el sistema, añadir un nuevo usuario al sistema en caso de que sea

necesario o la posibilidad de eliminar un usuario que no tenga parcela asociada.

La parte del servidor será totalmente ajena, tanto al responsable como al regante, ya que todas las interacciones con el sistema serán realizadas mediante la aplicación Android, quedando el servidor únicamente con la tarea de almacenar la base de datos y enviar los correos automáticos que requieran algunas de las tareas.

1.5. Estructura de la memoria

En el siguiente apartado se explicarán las distintas tecnologías, software y hardware, utilizadas en el desarrollo del proyecto.

A continuación, en los apartados 3 y 4, se explicará como se ha realizado la obtención de los requisitos. Después, a partir de esa primera información, se pasará a especificarlos mediante diagramas de casos de uso.

En el apartado 5, se detallará el diseño del sistema, que estará compuesto por una aplicación Android y un servidor con la base de datos.

Finalmente, se expondrán aspectos relevantes del proceso de desarrollo y se detallará el funcionamiento de la aplicación con capturas de pantalla de la aplicación desarrollada.

2. Tecnologías

En el siguiente apartado se describirán las tecnologías que se han utilizado en el desarrollo y funcionamiento de la aplicación. En primer lugar, se expondrá un gráfico con la arquitectura del sistema. A continuación, se citarán las tecnologías software utilizadas y, finalmente, los equipos hardware utilizados.

2.1. Arquitectura del sistema

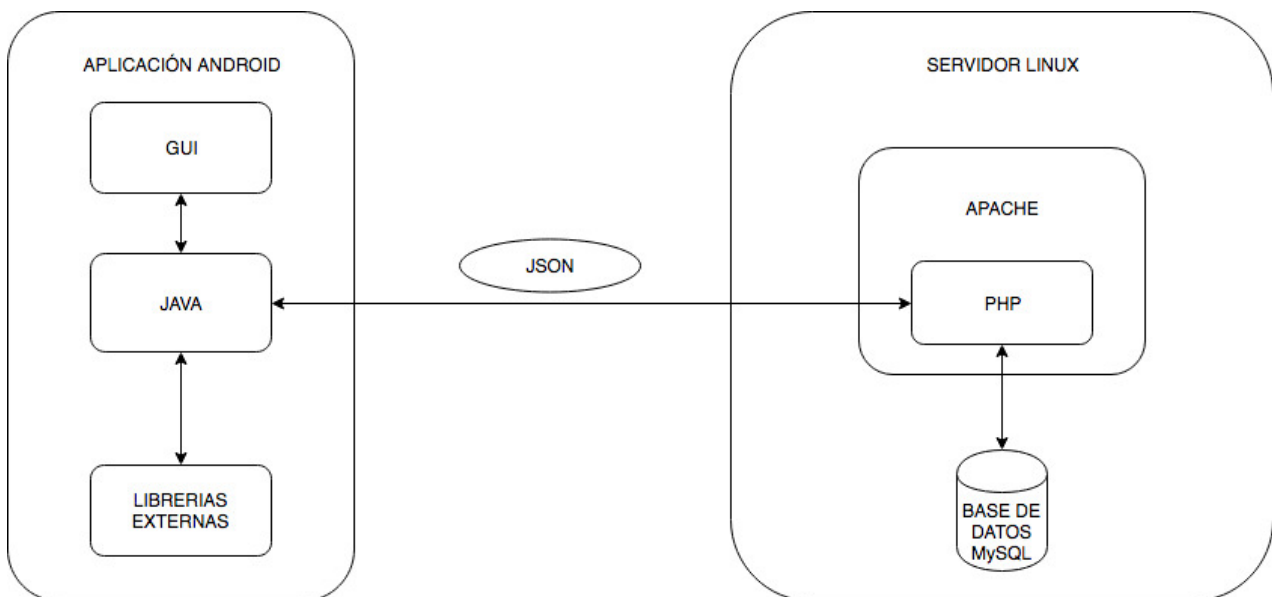


Figura 2 - Arquitectura del sistema

Como se especifica en la Figura 2, el sistema estará formado por una arquitectura cliente-servidor, que engloba, por una parte, a la aplicación Android (cliente) y, por otra, al servidor web (servidor).

Será la aplicación la encargada de mostrar todos los datos e implementar toda la lógica que hay detrás de la gestión de las reservas.

Por otro lado, la base de datos estará almacenada en servidor web, al cual se conectará la aplicación para recuperar todos los datos propios del sistema.

Dicha conexión se realizará mediante peticiones HTTP a *scripts* PHP, cuyo propósito es obtener los datos requeridos de una base de datos MySQL y codificarlos en formato JSON, para ser enviados en forma de respuesta HTTP a la aplicación.

2.2. Software utilizado

A continuación se detallarán las distintas tecnologías software utilizadas para el desarrollo del sistema.

Desarrollo de la aplicación Android

Para el desarrollo de aplicaciones Android existen diferentes alternativas.

En primer lugar tenemos el desarrollo de aplicaciones Android nativas, que es el método escogido en este trabajo. Consiste en el desarrollo de las aplicaciones mediante Java para el núcleo de la aplicación e interfaces XML para el diseño de las interfaces.

Éste método es el más recomendable, ya que utiliza el mismo lenguaje de programación que el propio sistema operativo, por lo que la aplicación resultante tiene un mejor rendimiento. Además, ofrece una mayor facilidad para interactuar con el hardware del sistema, como por ejemplo los sensores y la cámara.

Sin embargo, el tiempo que se necesita para el desarrollo de una aplicación nativa es mucho mayor que otras alternativas, por lo cual resulta más costoso su desarrollo.

Para el desarrollo de este tipo de aplicaciones se necesita disponer de un entorno de desarrollo (IDE), como son Eclipse o Android Studio. Éste último es la opción escogida en este trabajo, ya que, en comparación con Eclipse, posee aspectos clave en los que lo supera, como pueden ser la facilidad de uso, un código más estructurado y con buenas sugerencias para mejorarlo, además de un mejor soporte, porque está desarrollada directamente por Google, que es a su vez la empresa propietaria del sistema operativo Android. Además del IDE, es necesario tener instalados tanto el entorno de desarrollo de Java (JDK) como el de Android (SDK).

Otra alternativa para el desarrollo de una aplicación Android son las aplicaciones web, que, mediante *frameworks* de desarrollo, pueden ser exportadas a dispositivos Android e iOS. El desarrollo se presupone más sencillo que el de una aplicación nativa, aunque a costa de empeorar el rendimiento. Un ejemplo de éste tipo de frameworks es IONIC, basado en HTML5, CSS y Cordova para su funcionamiento base.

En este trabajo, se ha optado por el desarrollo de una aplicación nativa por diversas razones. La más importante es sus ventajas en cuanto a que ofrece un buen rendimiento, tanto en uso de recursos como en respuesta en tiempo de ejecución. Éste es un factor clave, ya que los usuarios pueden utilizar dispositivos antiguos o de bajas prestaciones.

Ubuntu

Ubuntu es un sistema operativo basado en GNU/LINUX que se distribuye bajo una licencia de Software libre y que tiene como base un sistema gráfico basado en Unity, el cual incluye programas básicos, como el navegador o una suite de ofimática, de software libre.

Aunque Ubuntu no necesita los mismos requisitos para instalarlo que un sistema Windows, hay que decir que su sistema operativo es actual y se actualiza como cualquier otro sistema, y que por tanto también requiere un hardware “potente” aunque sin llegar a exigir maquinas actuales como los sistemas operativos Windows o Mac.

En este trabajo, se ha utilizado la versión 16.04 LTS para el servidor, la cual fue lanzada en 2016, y trae por defecto software libre como el navegador Firefox o la suite de ofimática LibreOffice.

Apache

Apache es un poderoso servidor web, cuyo nombre proviene de la frase inglesa “a patchy server” y es completamente libre, ya que es un software de código libre y con licencia GPL. Una de las ventajas más grandes de Apache, es que es un servidor web multiplataforma, es decir, puede trabajar con diferentes sistemas operativos y mantener un excelente rendimiento

Entre las principales características de Apache, se encuentran las siguientes:

- Soporte de seguridad SSL y TLS.
- Puede realizar autenticación de datos utilizando SGDB.
- Puede dar soporte a diferentes lenguajes, como Perl, PHP, Python y TCL.

Apache es utilizado principalmente, para realizar servicio a páginas web, ya sean estáticas o dinámicas. Este servidor se integra a la perfección con otras aplicaciones, creando el famoso paquete XAMP con Perl, Python, MySQL y PHP, junto a cualquier sistema operativo, que por lo general es Linux, Windows o Mac OS.

MySQL

MySQL es un sistema de administración de bases de datos (*Database Management System, DBMS*) para bases de datos relacionales.

MySQL, como base de datos relacional, utiliza múltiples tablas para almacenar y organizar la información. MySQL fue escrito en C y C++ y

destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados, como son PHP, Perl y Java, y su integración en distintos sistemas operativos.

También es muy destacable la característica de ser código abierto de MySQL, que hace que su utilización sea gratuita e incluso se pueda modificar con total libertad, pudiendo descargar su código fuente. Esto ha favorecido muy positivamente su desarrollo y continuas actualizaciones, para hacer de MySQL una de las herramientas más utilizadas por los programadores orientados a Internet.

PHP

PHP es un lenguaje de programación para el desarrollo web de contenido dinámico en el lado del servidor. Está actualmente entre los proyectos de código abierto más populares. El código es interpretado por un servidor web con un módulo procesador de PHP que genera la página Web resultante.

Gracias en parte a la similitud de su sintaxis con el lenguaje C y Perl, PHP permite a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones. Resultado de ello son la publicación de blogs o la implementación de múltiples sitios web, como tiendas online. Por supuesto, estos sitios utilizan otras tecnologías en conjunción con PHP, por ejemplo un sistema de gestión de bases de datos como MySQL, que es un factor importante para este trabajo.

JSON

JSON (*JavaScript Object Notation* - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos entre entidades remotas. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript, *Standard ECMA-262 3rd Edition* - Diciembre 1999. JSON es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos. JSON está constituido por dos estructuras:

- Una colección de pares de nombre/valor. En varios lenguajes esto es conocido como un objeto, registro, estructura, diccionario, tabla *hash* o lista de claves.
- Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como *arrays*, vectores, listas o secuencias.

2.3. Hardware utilizado

A continuación se citarán y detallarán todos los equipos hardware utilizados durante el transcurso del trabajo.

Desarrollo de la aplicación Android

Para el desarrollo de la aplicación mediante Android Studio se ha utilizado un Macbook Pro Retina con las siguientes características:

- Procesador: Intel Core i5 a 2'4 GHZ
- Gráfica: Intel Iris
- RAM: 8GB DDR3 1600MHZ
- Disco duro: SSD 256 GB

Testeo de la aplicación Android

Existen dos opciones para el testeo de las aplicaciones: el uso de un dispositivo móvil real (*smartphone*) o una máquina virtual que lo emule en el propio PC usado para el desarrollo. Se ha optado por un dispositivo físico porque ofrece un mejor rendimiento en cuanto a tiempos de respuesta de la aplicación, lo que hace el desarrollo del proyecto más ágil. En cambio, el uso de una máquina virtual resulta mucho más lento a no ser que se posea un PC extremadamente potente.

El *smartphone* utilizado durante el desarrollo ha sido un Xiaomi Redmi Note 3 Pro, el cual tiene las siguientes características:

- Procesador: Qualcomm Snapdragon 650 a 1'6 GHZ HexaCore
- RAM: 3 GB
- Almacenamiento: 32 GB
- Pantalla: 5'5 pulgadas FullHD
- Versión de Android: 5.1 y 6.0

Para comprobar que la aplicación presenta un rendimiento aceptable en dispositivos menos potentes, también se han realizado pruebas en un ZTE Blade C2 con las siguientes características:

- Procesador: Mediatek MTK6572 a 1'3 GHZ DualCore
- RAM: 512 MB
- Almacenamiento: 4 GB
- Pantalla: 4 pulgadas
- Versión de Android: 4.2

Comparando el uso de la aplicación entre ambos dispositivos, se ha llegado a la conclusión de que no habrá problemas de rendimiento si se usa la aplicación en dispositivos poco potentes, ya que se comporta de manera muy similar en ambos dispositivos, siendo la velocidad de la conexión a Internet el factor más determinante, y no tanto la potencia del dispositivo.

Servidor

Para la instalación y puesta en funcionamiento del servidor se ha utilizado un pequeño *Netbook*, PackardBell Dot S, porque ofrece una gran movilidad y no se requiere mucha potencia, ya que, en principio, la aplicación es solo con fines académicos y no se va a poner en producción. Sus características son las siguientes:

- Procesador: Intel Atom a 1'6 GHZ
- RAM: 2 GB DDR3
- Disco duro: HDD 250 GB

3. Obtención de requisitos

En este apartado se describirá, en primer lugar, cómo se han obtenido los requisitos que deberá cumplir la aplicación. Después, se definirán una serie de conceptos propios del sistema, así como los actores involucrados. Finalmente, se hará una lista tanto de las características generales del sistema como de los requisitos no funcionales, así como una breve explicación de cada uno de ellos.

3.1. Toma de requisitos

Con el fin de identificar las funcionalidades a desarrollar en el sistema, en la primera fase del proyecto se realizó una entrevista con el presidente del Sindicato de Regantes de Cullera.

En primer lugar, se explicó como es la estructura de regadío de la partida del *Mareny de Sant Llorenç*, y se distribuye el agua entre las distintas acequias y brazos.

Después, se explicó la forma en que, en la actualidad, se pedía turno para poder regar una parcela, la cual es, básicamente, contactar con el regante y elegir entre las horas que hay disponibles para el riego en el brazo al cual pertenece una determinada tu parcela.

De la entrevista, además, se obtuvieron datos relevantes para el desarrollo del sistema, como es la forma de calcular el tiempo de riego de una parcela, dependiendo de su superficie y otros factores.

3.2. Definiciones

A continuación se definirán una serie de términos propios de la aplicación:

- **Usuario del sistema:** Usuario que tiene acceso, mediante credenciales, a las utilidades de la aplicación. Puede ser de dos tipos, responsable y regante, teniendo cada uno una serie de funcionalidades propias.
- **Horario de un brazo:** Periodo de tiempo en que un brazo transporta agua para riego. En la aplicación el horario de un brazo se especifica desde el día y la hora que empieza hasta el día y la hora que acaba el riego dentro de una semana. Durante la ejecución de la aplicación, es ésta la encargada de mostrar el horario como uno distinto para cada día, ya que se considera que el horario diario para el regadío (horario de trabajo del regante) es desde las 7:00 a las 19:00.

- **Franja horaria:** Cada una de las fracciones de 30 minutos en que se divide el horario de un brazo. El sistema mostrará una lista de fracciones de tiempo de 30 minutos para la elección de la hora de comienzo de una reserva, la cual será dependiente del horario que se haya especificado para el brazo en el cual nos encontremos.
- **Reserva:** Periodo de tiempo en el que una parcela tiene disponibles los recursos hídricos para poder ser regada. En la aplicación, una reserva se especifica mediante una o varias franjas horarias consecutivas.
- **Número de parcela:** Es el número que se utiliza para identificar una parcela en la aplicación. Se compone del identificador de la acequia principal, el identificador de la acequia secundaria, el identificador del brazo y la posición (siguiendo recorrido en profundidad-primero preorden*) que ocupa la parcela en ese brazo.
- **Tiempo estimado de riego de una parcela:** Estimación del tiempo necesario para el riego de una parcela cuando el caudal del brazo es el máximo.
Este valor depende, además del caudal, de otros dos factores: la distancia de la parcela a la toma de la acequia secundaria a la que pertenece y la superficie de la parcela. En nuestro sistema, hemos asignado un tiempo del riego de 30 minutos por hanegada (unidad de medida que en el territorio del antiguo *Regne de València* corresponde a 831 m²).
- **Información estática de una parcela:** Se considera información estática de una parcela su superficie, el brazo al que pertenece y el número de parcela asociado.

3.3. Actores del sistema

Distinguiremos dos tipos de usuarios del sistema:

- **Responsable de parcela:** Es la persona encargada del riego de una o más parcelas. Tendrá privilegios para la gestión de las reservas de sus parcelas asociadas, así como para modificar su propia información de usuario.
- **Regante:** Es la persona que gestiona el sistema de riego de la partida. En la aplicación, el regante puede verse como el administrador del sistema y tendrá acceso a todas las reservas y la gestión de los horarios disponibles para el riego.

**Recorrido en profundidad-primero preorden:* Se elige el nodo raíz del árbol y, recursivamente, se sigue con el sub-árbol izquierdo y después el sub-árbol derecho

3.4. Características de la aplicación

A continuación se especificarán las características generales de la aplicación, dando una breve descripción de ellas e indicando qué actor del sistema está involucrado en cada una:

Gestión de reservas	
Descripción	El usuario será capaz de crear nuevas reservas, así como visualizar y eliminar las reservas realizadas.
Actor involucrado	Responsable de parcela y regante.

Tablón de anuncios	
Descripción	Existirá un tablón de anuncios en el cual el regante colgará los anuncios que crea que son relevantes para los responsables.
Actor involucrado	Responsable de parcela y regante.

Edición de perfil del usuario	
Descripción	El usuario será capaz tanto de editar la información correspondiente a su perfil.
Actor involucrado	Responsable de parcela.

Gestión de horarios	
Descripción	Creación, consulta y supresión de horarios de los brazos.
Actor involucrado	Regante.

Gestión de responsables	
Descripción	El usuario debe poder crear, eliminar, visualizar y editar la información de los perfiles de los responsables existentes en la aplicación.
Actor involucrado	Regante.

Gestión de la información de las parcelas	
Descripción	El usuario deberá poder editar la información de las parcelas no considerada estática.
Actor involucrado	Regante.

3.5. Requisitos no funcionales

La aplicación debe cumplir los siguientes requisitos no funcionales:

- No se puede cancelar una reserva cuando falten dos horas o menos para que de inicio dicha reserva.
- No pueden haber más de una reserva para una parcela el mismo día.
- Los usuarios podrán reservar franjas horarias para la semana actual y la siguiente.
- Una franja horaria solo puede estar ocupada por una reserva.
- Cada parcela tendrá un tiempo estimado de riego definido.
- Si el regante cancela una reserva, debe notificarse el motivo de la cancelación mediante un correo electrónico. Además, se creará un anuncio dirigido a dicho usuario.
- La aplicación debe funcionar en cualquier dispositivo Android con versión del sistema operativo 4.1 o superior.
- Para su funcionamiento, el dispositivo debe tener conexión a Internet.

4. Especificación de requisitos

En este apartado se muestran los diagramas de casos de uso, tanto de los responsables de parcela como del regante, junto con la descripción detallada de cada uno de los casos de uso.

4.1. Diagrama de casos de uso de los responsables

A continuación, en la Figura 3 se muestra el diagrama de clases correspondiente a los responsables de las parcelas.

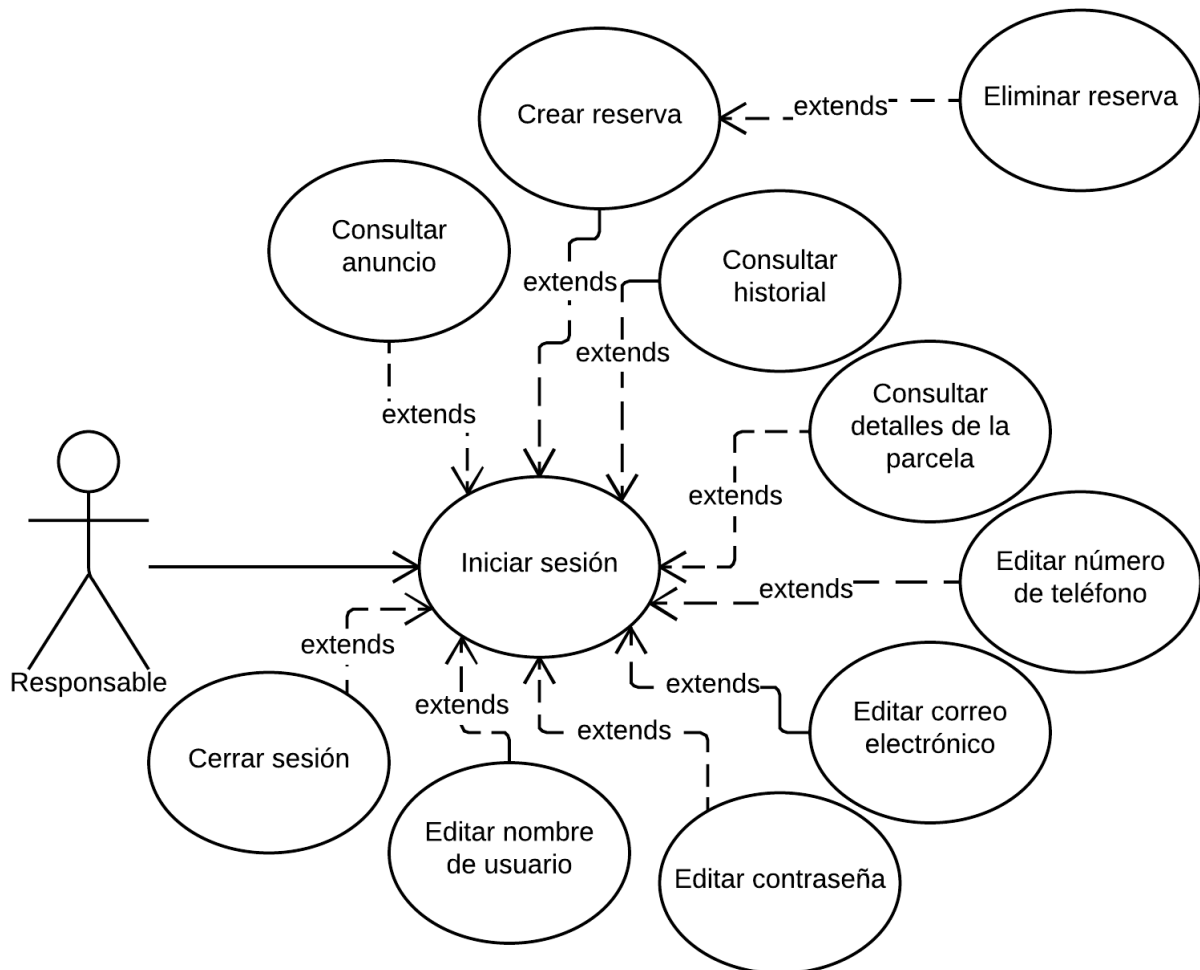


Figura 3 - Casos de uso del responsable

Iniciar sesión	
Descripción	Acceder a la aplicación con un perfil de usuario.
Precondición	Iniciar la aplicación.
Pasos	1. Abrir en la aplicación.
Postcondición	Accedes al perfil.
Excepciones	- Usuario y contraseña no coinciden.

Crear reserva	
Descripción	Reservar franja de tiempo para uso del agua.
Precondición	Tener cuenta en la aplicación.
Pasos	<ol style="list-style-type: none"> 1. Iniciar sesión en la aplicación. 2. Seleccionar parcela, en caso de que haya más de una asociada al perfil. 3. Acceder a la pestaña "Reservar". 4. Seleccionar franja horaria en la cual se desea que comience la reserva. 5. Especificar el tiempo de riego si no se desea que sea el que trae la parcela por defecto.
Postcondición	La reserva queda registrada en la aplicación.
Excepciones	<ul style="list-style-type: none"> - El tiempo de riego no debe superar el doble del tiempo que trae la parcela por defecto. - La reserva no debe solaparse con otra ya existente.

Eliminar reserva	
Descripción	Eliminar reserva de la aplicación.
Precondición	Tener reservas registradas en la aplicación.
Pasos	<ol style="list-style-type: none"> 1. Iniciar sesión en la aplicación. 2. Seleccionar parcela, en caso de que haya más de una asociada al perfil. 3. Acceder a la pestaña “Historial”. 4. Seleccionar la reserva que se desea eliminar.
Postcondición	La reserva queda eliminada de la aplicación.
Excepciones	- La reserva no se eliminará a partir de dos horas antes de su comienzo.

Consultar historial	
Descripción	Consulta de todas las reservas realizadas para una parcela.
Precondición	Tener reservas registradas en la aplicación
Pasos	<ol style="list-style-type: none"> 1. Iniciar sesión en la aplicación. 2. Seleccionar parcela, en caso de que haya más de una asociada al perfil. 3. Acceder a la pestaña “Historial”.
Postcondición	Se muestra la lista de con todas las reservas asociadas a una parcela.
Excepciones	—

Consultar detalles de la parcela	
Descripción	Consulta de la información de la parcela.
Precondición	Seleccionar una parcela.
Pasos	<ol style="list-style-type: none"> 1. Iniciar sesión en la aplicación. 2. Seleccionar parcela, en caso de que haya más de una asociada al perfil. 3. Acceder a la pestaña “Detalles”.
Postcondición	Se visualizan los detalles de la parcela seleccionada.
Excepciones	—

Editar correo electrónico	
Descripción	Editar la dirección de correo electrónico asociada al usuario.
Precondición	Iniciar sesión en la aplicación como responsable.
Pasos	<ol style="list-style-type: none"> 1. Iniciar sesión en la aplicación. 2. Seleccionar parcela, en caso de que haya más de una asociada al perfil. 3. Acceder al menú lateral. 4. En el apartado “Editar perfil”, seleccionar la opción “Correo electrónico”.
Postcondición	La nueva dirección de correo queda registrada en la aplicación.
Excepciones	—

Editar número de teléfono	
Descripción	Editar el teléfono de contacto del responsable.
Precondición	Iniciar sesión en la aplicación como responsable.
Pasos	<ol style="list-style-type: none"> 1. Iniciar sesión en la aplicación. 2. Seleccionar parcela, en caso de que haya más de una asociada al perfil. 3. Acceder al menú lateral. 4. En el apartado “Editar perfil”, seleccionar la opción “Número de teléfono”.
Postcondición	El nuevo número de teléfono queda registrado en la aplicación.
Excepciones	- El número de teléfono debe tener un formato válido.

Editar contraseña	
Descripción	Editar la contraseña asociada al usuario.
Precondición	Iniciar sesión en la aplicación como responsable.
Pasos	<ol style="list-style-type: none"> 1. Iniciar sesión en la aplicación. 2. Seleccionar parcela, en caso de que haya más de una asociada al perfil. 3. Acceder al menú lateral. 4. En el apartado “Editar perfil”, seleccionar la opción “Contraseña”.
Postcondición	La nueva contraseña queda registrada en la aplicación.
Excepciones	- La contraseña debe tener más de 8 caracteres.

Editar nombre de usuario	
Descripción	Editar el nombre de usuario.
Precondición	Iniciar sesión en la aplicación como responsable.
Pasos	<ol style="list-style-type: none"> 1. Iniciar sesión en la aplicación. 2. Seleccionar parcela, en caso de que haya más de una asociada al perfil. 3. Acceder al menú lateral. 4. En el apartado “Editar perfil”, seleccionar la opción “Nombre de usuario”.
Postcondición	El nombre de usuario queda registrado en el perfil.
Excepciones	- No pueden haber dos nombres de usuario iguales.

Consultar anuncios	
Descripción	Consulta de los anuncios generales y asociados al perfil.
Precondición	Iniciar sesión en la aplicación como responsable.
Pasos	<ol style="list-style-type: none"> 1. Iniciar sesión en la aplicación. 2. Seleccionar parcela, en caso de que haya más de una asociada al perfil. 3. Acceder a la pestaña “Anuncios”.
Postcondición	Se muestra la lista de con todas los anuncios generales y asociados al perfil.
Excepciones	—

Cerrar sesión	
Descripción	Acceder a la aplicación con un perfil de usuario.
Precondición	Iniciar sesión en la aplicación como responsable.
Pasos	<ol style="list-style-type: none"> 1. Iniciar sesión en la aplicación. 2. Seleccionar parcela, en caso de que haya más de una asociada al perfil. 3. Acceder al menú lateral. 4. En el apartado “Opciones de sesión”, seleccionar la opción “Cerrar sesión”.
Postcondición	Se muestra la pantalla de inicio de sesión de la aplicación.
Excepciones	—

4.2. Diagrama de casos de uso del regante

A continuación, en la Figura 4 se muestra el diagrama de clases correspondiente al regante.

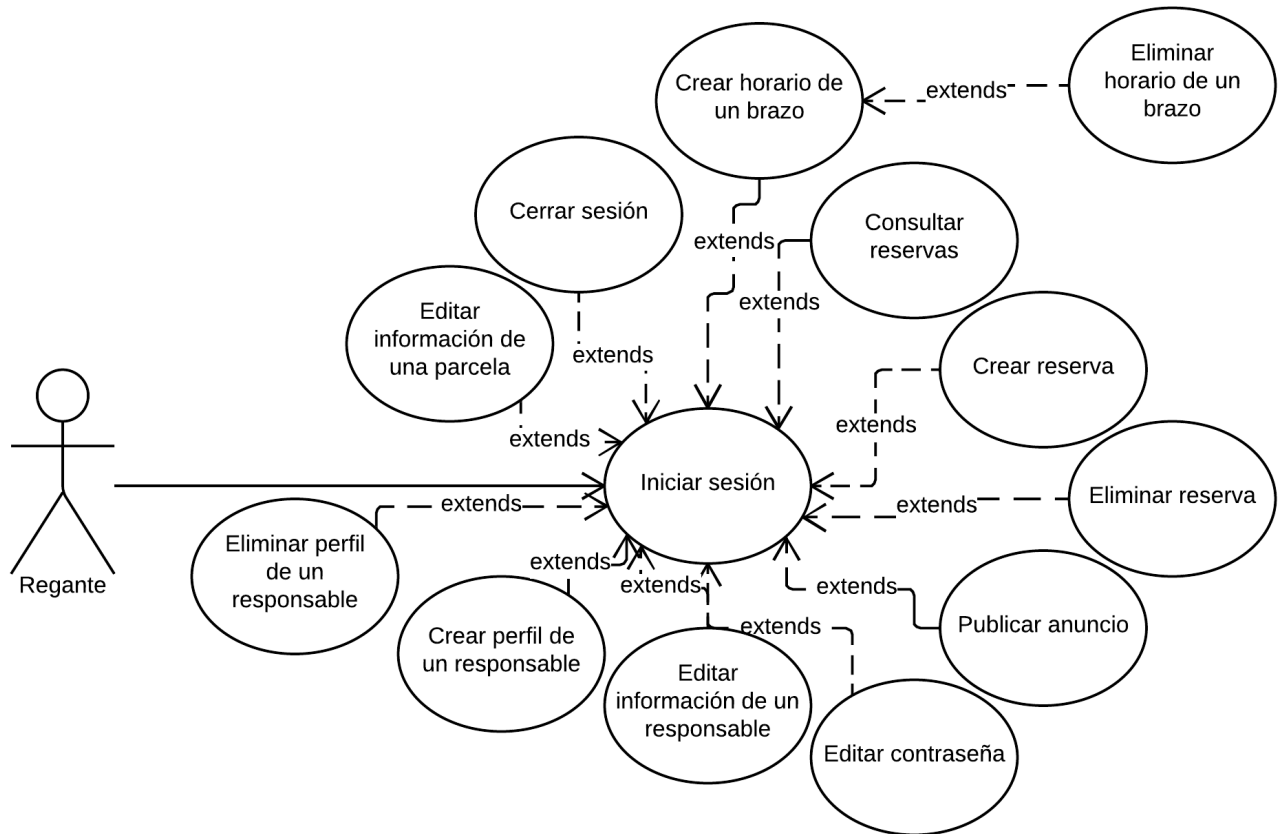


Figura 4 - Casos de uso del regante

Iniciar sesión	
Descripción	Acceder a la aplicación con un perfil de usuario.
Precondición	Iniciar la aplicación.
Pasos	1. Abrir en la aplicación.
Postcondición	Accedes al perfil.
Excepciones	- Usuario y contraseña no coinciden.

Consultar reservas	
Descripción	Consulta de todas las reservas para la semana actual y la próxima.
Precondición	Tener reservas registradas en la aplicación
Pasos	<ol style="list-style-type: none"> 1. Iniciar sesión en la aplicación con la cuenta del regante. 2. Seleccionar la opción “Ver reservas”
Postcondición	Se muestra la lista de con todas las reservas para la semana actual y la próxima.
Excepciones	—

Crear reserva	
Descripción	Reservar espacio de tiempo para obtener recursos hídricos.
Precondición	Seleccionar el usuario responsable de la parcela para la cual se desea realizar la reserva.
Pasos	<ol style="list-style-type: none"> 1. Iniciar sesión en la aplicación con la cuenta del regante. 2. Seleccionar la opción “Reservar”. 3. Seleccionar el usuario responsable de la parcela y la parcela para la cual se desea reservar. 4. Seleccionar franja horaria en la cual se desea que comience la reserva. 5. Especificar el tiempo de riego si no se desea que sea el que trae la parcela por defecto.
Postcondición	La reserva queda registrada en la aplicación.
Excepciones	<ul style="list-style-type: none"> - El tiempo de riego no debe superar el doble del tiempo que trae la parcela por defecto. - La reserva no debe solaparse con otra.

Eliminar reserva	
Descripción	Eliminar una reserva de la aplicación.
Precondición	Existir reservas registradas en la aplicación.
Pasos	<ol style="list-style-type: none"> 1. Iniciar sesión en la aplicación con la cuenta del regante. 2. Acceder a la opción “Ver reservas”. 3. Seleccionar la reserva. 4. Especificar el motivo de la cancelación.
Postcondición	La reserva queda eliminada de la aplicación y el responsable de la parcela afecta recibe un correo electrónico y un anuncio para notificarlo.
Excepciones	- La reserva no se eliminará a partir de dos horas antes de su comienzo.

Crear horario de un brazo	
Descripción	Crear horario para la asignación de agua a un brazo.
Precondición	Iniciar sesión como regante.
Pasos	<ol style="list-style-type: none"> 1. Iniciar sesión en la aplicación con la cuenta del regante. 2. Seleccionar la opción “Horarios”. 3. Seleccionar la opción “Nuevo horario”. 4. Especificar fecha y hora, tanto de inicio como de fin, y seleccionar el brazo.
Postcondición	El horario queda registrado en la aplicación.
Excepciones	- El horario no puede solaparse con otro ya existente en la aplicación.

Eliminar horario de un brazo	
Descripción	Eliminar un horario de la aplicación.
Precondición	Iniciar sesión como regante.
Pasos	<ol style="list-style-type: none"> 1. Iniciar sesión en la aplicación con la cuenta del regante. 2. Seleccionar la opción “Horarios”. 3. Seleccionar de la lista el horario que se desea eliminar.
Postcondición	La horario queda eliminado de la aplicación.
Excepciones	—

Publicar anuncio	
Descripción	Publicar anuncio de interés para los responsables de parcela.
Precondición	Iniciar sesión como regante.
Pasos	<ol style="list-style-type: none"> 1. Iniciar sesión en la aplicación con la cuenta del regante. 2. Seleccionar la opción “Anuncios”. 3. Especificar el título y el anuncio.
Postcondición	El anuncio queda registrado en la aplicación.
Excepciones	—

Editar información de un responsable	
Descripción	Editar la información de perfil de un responsable de parcela.
Precondición	Iniciar sesión como regante.
Pasos	<ol style="list-style-type: none"> 1. Iniciar sesión en la aplicación con la cuenta del regante. 2. Seleccionar la opción “Responsables”. 3. Seleccionar de la lista el responsable.
Postcondición	Los cambios quedan registrados en la aplicación.
Excepciones	<ul style="list-style-type: none"> - Si cambia la contraseña, ésta debe tener 8 caracteres como mínimo. - Si modifica el número de teléfono, éste debe tener un formato válido.

Crear perfil de un responsable	
Descripción	Crear un nuevo perfil para un responsable de parcela.
Precondición	Iniciar sesión como regante.
Pasos	<ol style="list-style-type: none"> 1. Iniciar sesión en la aplicación con la cuenta del regante. 2. Seleccionar la opción “Responsables”. 3. Seleccionar la opción “Crear perfil”. 4. Especificar todos los datos del responsable.
Postcondición	El nuevo perfil queda registrado en la aplicación
Excepciones	<ul style="list-style-type: none"> - La contraseña debe tener 8 caracteres como mínimo. - El número de teléfono debe tener un formato válido.

Eliminar perfil de un responsable	
Descripción	Eliminar el perfil para un responsable de parcela.
Precondición	Iniciar sesión como regante.
Pasos	<ol style="list-style-type: none"> 1. Iniciar sesión en la aplicación con la cuenta del regante. 2. Seleccionar la opción “Responsables”. 3. Seleccionar la opción “Eliminar perfil”. 4. Seleccionar el responsable a eliminar de la aplicación.
Postcondición	El perfil queda eliminado de la aplicación
Excepciones	- El responsable que se desea eliminar no puede tener parcelas asociadas en la aplicación.

Editar información de una parcela	
Descripción	Editar la información no estática de una parcela.
Precondición	Iniciar sesión como regante.
Pasos	<ol style="list-style-type: none"> 1. Iniciar sesión en la aplicación con la cuenta del regante. 2. Seleccionar la opción “Parcelas”. 3. Seleccionar de la lista la parcela.
Postcondición	Los cambios quedan registrados en la aplicación.
Excepciones	—

Cerrar sesión	
Descripción	Cerrar sesión en la aplicación.
Precondición	Iniciar sesión en la aplicación como responsable.
Pasos	<ol style="list-style-type: none"> 1. Iniciar sesión en la aplicación con la cuenta del regante. 2. Pulsar el icono de cerrar sesión situado a la derecha en la barra superior.
Postcondición	Se muestra la pantalla de inicio de sesión de la aplicación.
Excepciones	—

5. Diseño de la aplicación

En este apartado entraremos en los detalles del diseño de la aplicación. En primer lugar se expondrá el modelo relacional de la base de datos. A continuación, se presentará el diagrama de clases correspondiente a la aplicación. Finalmente, se mostrará una representación gráfica del algoritmo seguido por la aplicación mediante su diagrama de flujo.

5.1. Capa de persistencia

En la Figura 5 se muestra el modelo relacional de la base de datos:

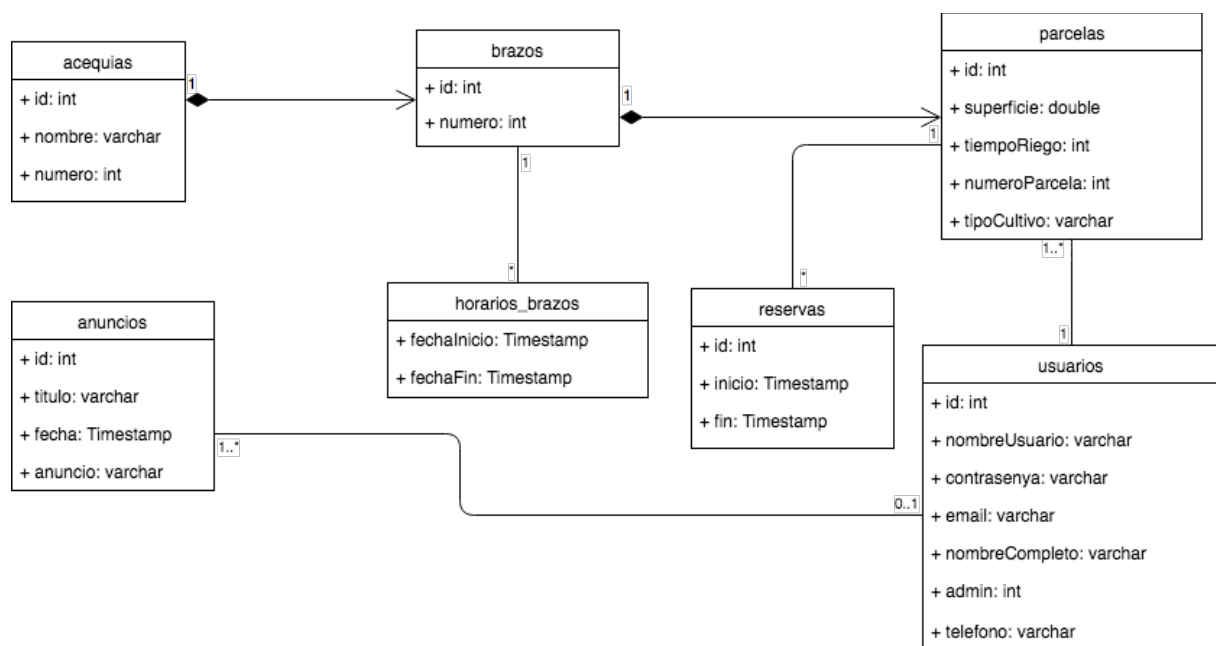


Figura 5 - Modelo relacional

En primer lugar tenemos las tablas relacionadas con la estructura de la partida de riego: acequias, brazos y parcelas. El objeto acequia pertenece únicamente a las acequias secundarias. Las acequias primarias no están directamente representadas en el sistema, sino que se guarda el nombre de ésta en el campo nombre de la tabla acequias.

Los horarios para la disponibilidad de agua depende del brazo en el que esté ubicada la parcela, por lo cual la tabla de horarios (horarios_brazos) debe estar relacionada con la tabla brazos, con una relación de uno a varios, es decir, cada horario pertenece a un solo brazo y cada brazo puede tener muchos horarios.

En cuanto la tabla de usuarios, ésta está relacionada con la tabla parcelas en una relación uno a muchos, lo que significa que una parcela pertenece de un usuario y un usuario puede poseer muchas parcelas.

La tabla de reservas está enlazada a la tabla parcelas, con una relación de uno a muchos, es decir, una parcela puede tener varias reservas (por ejemplo, si se riega una vez por semana) pero una reserva pertenece únicamente a una parcela. Con esto se hace que las reservas sean independientes del usuario, por lo que el usuario podrá realizar todas las reservas que quiera en un mismo día, siempre y cuando cada reserva corresponda a una parcela distinta. Esto último es uno de los requisitos no funcionales que debe cumplir la aplicación.

Finalmente, nos encontramos con la tabla anuncios, los cuales pueden estar asociados o no a un usuario. Cuando el anuncio no esté asociado a ningún usuario será mostrado a todos los usuarios.

5.2. Capa lógica

En este apartado se presentará el diagrama de clases correspondiente a la aplicación Android. El diagrama es el siguiente:

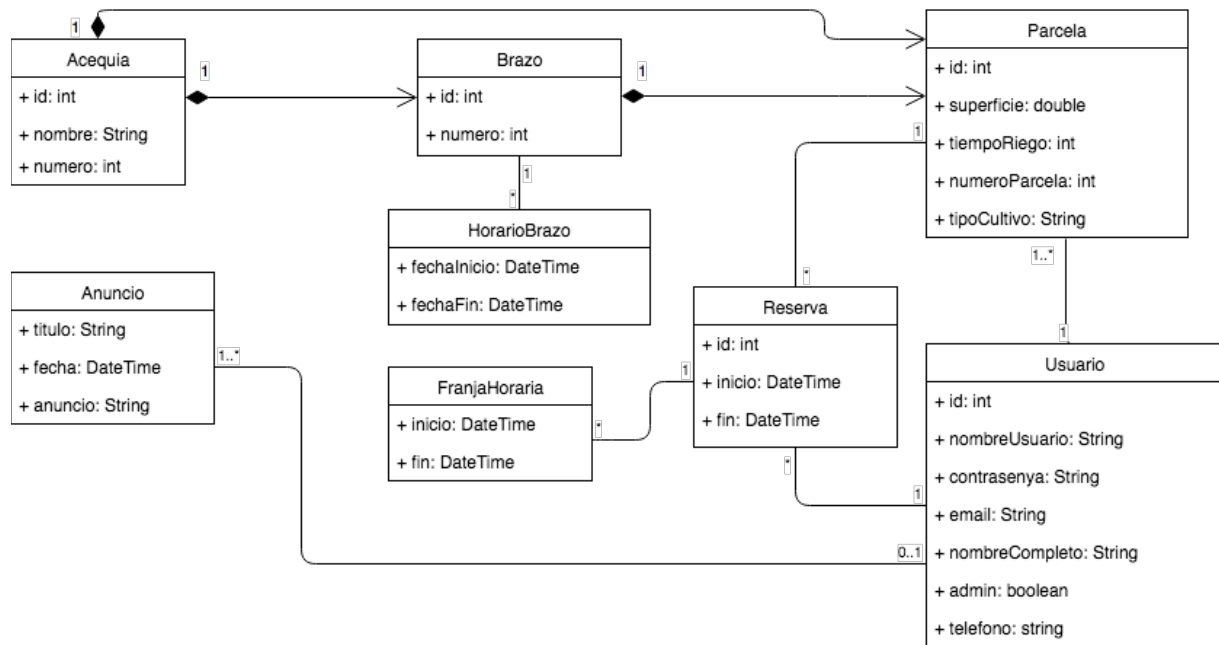


Figura 6 - Diagrama de clases

Las primeras entidades que nos encontramos son las correspondientes a la estructura de la partida de riegos: Acequia, Brazo y Parcela. Éstas clases son prácticamente iguales a las que encontraremos a las tablas de persistencia, exceptuando que en la Parcela se guarda también la referencia de la Acequia, ya que, se usa constantemente esta referencia y

no sería lógico tener que acceder cada vez al Brazo al que pertenece la parcela y extraer de ahí la información de la Parcela.

En relación con los horarios, éstos se guardan en el objeto HorarioBrazo, el cual guarda la fecha de inicio y fin del horario. Es la aplicación la que, en tiempo de ejecución, dividirá el horario del brazo en las distintas franjas horarias de 30 minutos.

A continuación se encuentra el objeto Usuario, el cual, posee los mismos campos que encontramos en la tabla usuarios en la base de datos.

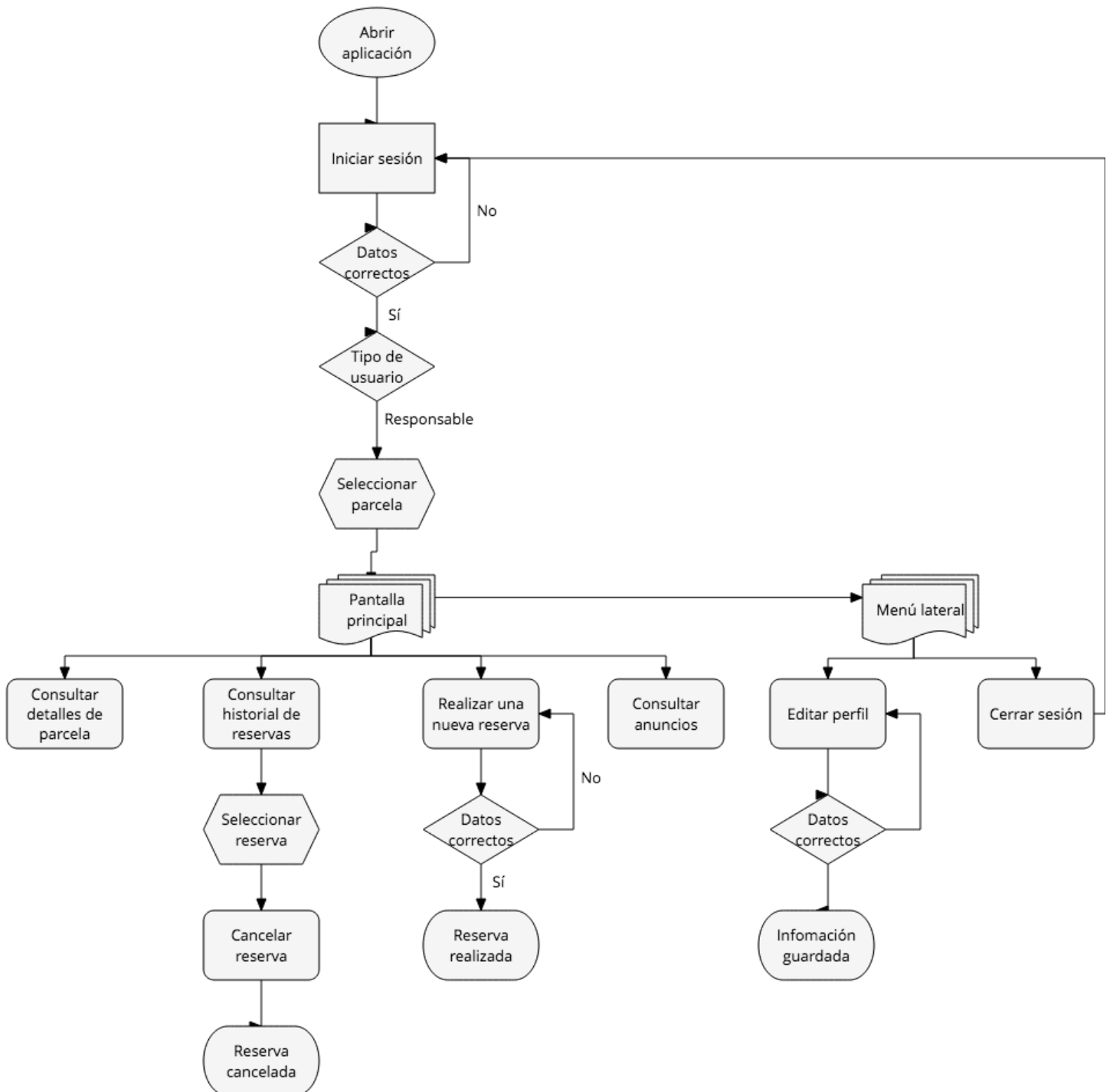
En lo que se refiere a la Reserva, en ésta se guarda también el nombre del Usuario propietario de la Parcela que está asociada a dicha reserva, para mostrar más información a todos los usuarios, tanto propietarios como regante, de la Reserva.

En cuanto al objeto Anuncio, éste puede estar o no relacionado con un responsable de parcela (Usuario). En caso de que lo esté, solo será visible para ese usuario. Si no tiene ningún usuario relacionado, será un anuncio global y será visible para todos los responsables.

Finalmente, la principal diferencia con la base de datos está en el objeto FranjaHoraria, el cual no está reflejado en la base de datos. Esto es así para evitar tráfico de datos innecesario, ya que no supone mucha penalización de rendimiento calcularlas en tiempo de ejecución y además, es muy posible que la aplicación tarde más en mostrar las franjas si las obtuviera de la base de datos. La aplicación las mostrará en periodos de media hora, dentro del horario del Brazo y del horario de trabajo del regante. Para calcular la disponibilidad de cada franja, se calculará, con la lista de reservas disponibles, que franjas están disponibles y cuáles no, agrupando las franjas que correspondan a una misma reserva en una sola. Ésta información no persiste, sino que se crea en tiempo de ejecución.

5.3. Diagrama de flujo

Para visualizar de forma global el funcionamiento de la aplicación y las funciones que se pueden realizar, a continuación se presentará un diagrama de flujo donde se muestra el flujo de acciones y procesos que se seguirán para llevar a cabo las funciones descritas anteriormente en ambos diagramas de Casos de Uso. Dicho diagrama se ha dividido en la perspectiva del responsable (página actual) y del regante (página siguiente) para facilitar su visualización.



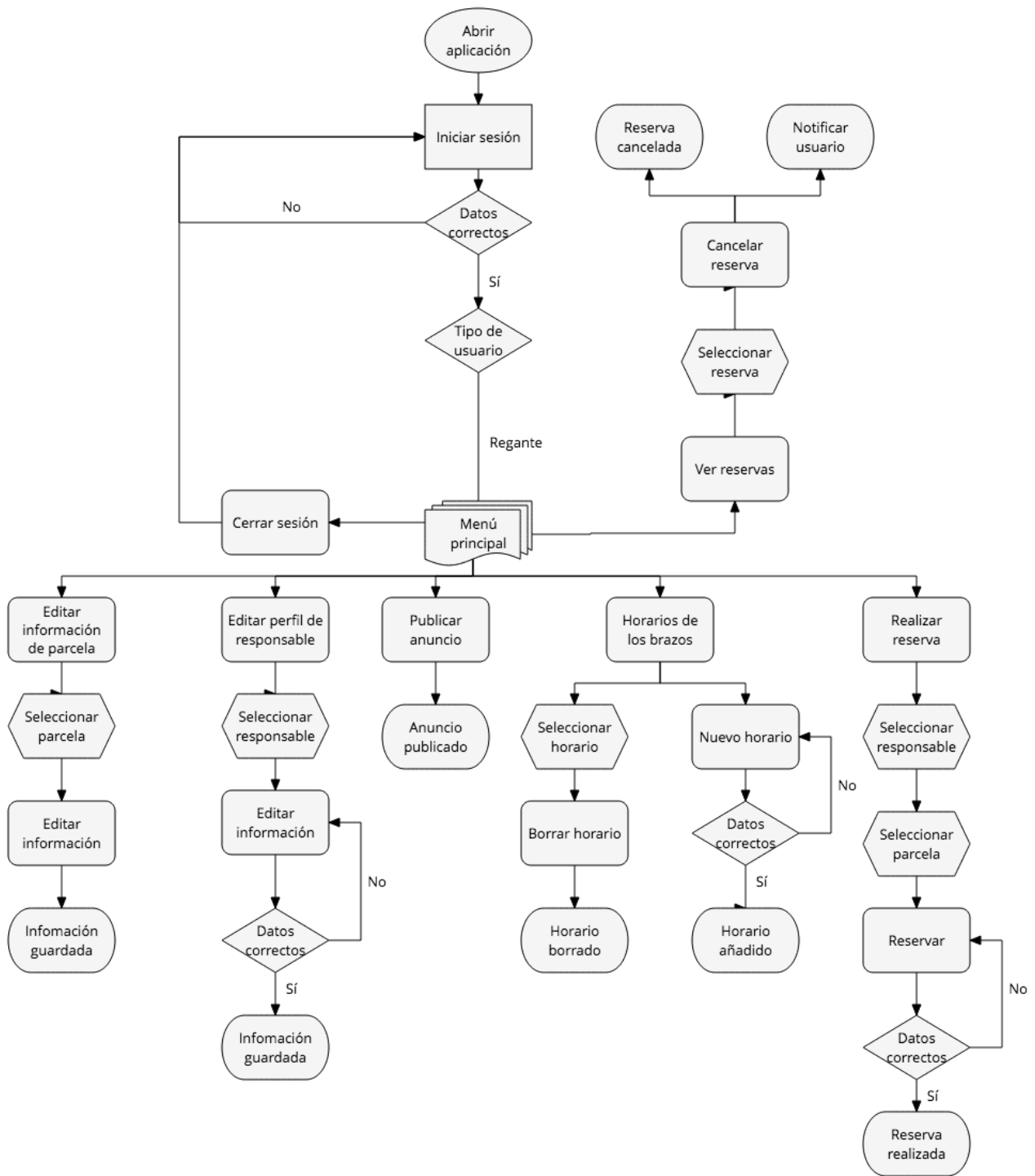


Figura 7 - Diagrama de flujo

6. Desarrollo de la aplicación

En el apartado actual se hará una descripción de aquellos aspectos de la implementación de la aplicación que son relevantes. En primer lugar se explicará la estructura del proyecto de software Android. A continuación, se detallará la tecnología utilizada para el *web service* y, finalmente, se citarán las librerías externas utilizadas en el desarrollo de la aplicación.

6.1. Estructura del proyecto Android

Para todo proyecto Android existe una estructura ya predefinida de la cual se parte. Dicha estructura es, por un lado, la lógica de la aplicación, y por otro, las vistas. Este patrón es común para toda aplicación Android, pero en este trabajo se ha ampliado la estructura del proyecto.

El apartado de las vistas es simplemente una carpeta donde se encuentran todos los archivos .xml correspondientes a las diferentes vistas de la aplicación. Éste no se ha modificado en el presente trabajo.

Por otro lado, tenemos el apartado de la lógica de la aplicación. En él se han creado distintos paquetes (Figura 8) para cada rol diferenciado de las clases Java. Estos paquetes son los siguientes: Modelo, Controladores, WebService, Conversores y Adaptadores.

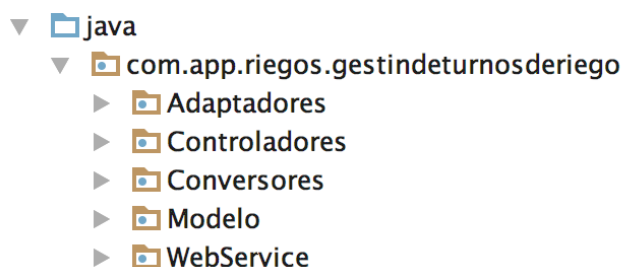


Figura 8 - Paquetes de la aplicación

En el paquete Modelo (Figura 9) se encuentran las distintas clases de la aplicación que corresponden a los distintos objetos a representar.

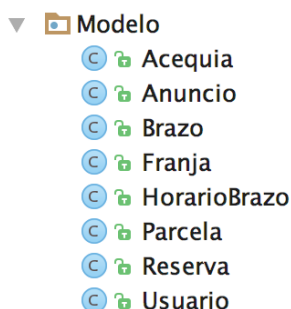


Figura 9 - Modelos

En cuanto al paquete Controladores, es donde se encuentran los controladores de todas las pantallas de la aplicación (en la Figura 10 se muestran unos pocos de ellos).

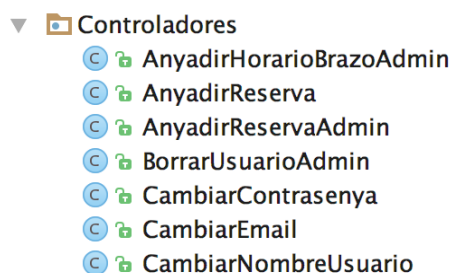


Figura 10 - Controladores

En el el paquete WebService (Figura 11) se encuentra la clase que se utiliza para realizar todas las llamadas al servidor.

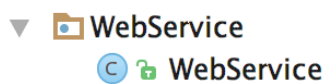


Figura 11 - Web service

En el paquete Conversores (Figura 12) se encuentra, por un lado, la clase ConversorJSON, en la cual está implementado el código que procesa las respuestas del servidor (en formato JSON) y crea los correspondientes objetos del modelo. Por otro lado, tenemos un conversor de fechas, utilizado para visualizar y formatear fechas para ser almacenadas en la base de datos.

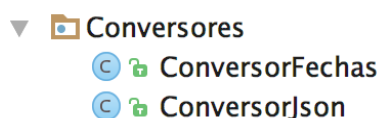


Figura 12 - Conversores

Finalmente se encuentra el paquete Adaptadores (Figura 13). En él se tienen todos los adaptadores de los ítems que se visualizan en las listas. Esto es, cada lista muestra un tipo de datos diferente, con lo cual habrá veces que se quieran mostrar más datos o menos, o incluso imágenes. Para esto necesitamos una vista para cada tipo de datos que queramos visualizar en una lista y un adaptador que, dada una lista de elementos, los visualice en la vista correspondiente.

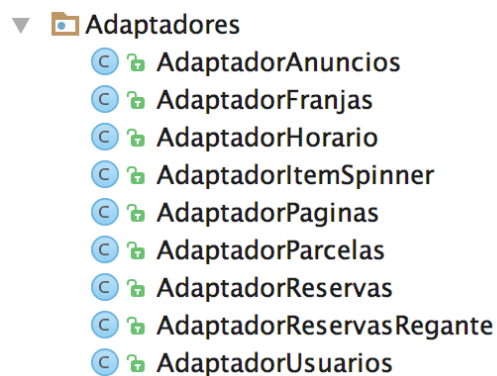


Figura 13 - Adaptadores

6.2. Web service

Para la conexión entre la aplicación y el servidor donde está almacenada la base de datos se ha implementado una interfaz *REST*.

REST es cualquier interfaz entre sistemas que use HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como XML y JSON. Es una alternativa en auge a otros protocolos estándar de intercambio de datos como SOAP, que disponen de una gran capacidad pero también mucha complejidad. A veces es preferible una solución más sencilla de manipulación de datos como *REST*.

Aunque *REST* posee cuatro operaciones relacionadas con los datos (*GET*, *POST*, *PUT* y *DELETE*), en este trabajo solo se ha hecho uso de *GET* y *POST*, ya que, todas las operaciones que se realizan en la aplicación se pueden llevar a cabo con éstas.

En el lado del servidor, las peticiones HTTP las recogen los distintos archivos PHP. Cada operación tiene asociado un archivo PHP distinto, para hacer el código lo más estructurado posible.

Son estos archivos PHP los encargados de realizar las consultas a la base de datos y codificar las respuestas en formato JSON. A continuación se mostrarán dos sencillos scripts, cada uno usando una operación distinta.

En la Figura 14 tenemos el script que, mediante la operación *GET*, y especificando el ID del usuario en cuestión, obtendrá los anuncios globales y los dirigidos a ese usuario.

```

<?php

require 'Database.php';

if ($_SERVER['REQUEST_METHOD'] == 'GET') {

    if (isset($_GET['destinatario'])) {
        $consulta1 = "SELECT * FROM anuncios WHERE destinatario = 0 OR
            destinatario = ?"." ORDER BY fecha DESC";
        $destinatario = $_GET['destinatario'];
        try {
            // Preparar sentencia
            $comando1 = Database::getInstance()->getDb()->prepare($consulta1);
            // Ejecutar sentencia preparada
            $comando1->execute(array($destinatario));
            // Capturar primera fila del resultado
            $anuncios = $comando1->fetchAll(PDO::FETCH_ASSOC);

            print json_encode(array($anuncios));

        } catch (PDOException $e) {
            print json_encode(array('error' => "Error"));
        }
    }
}

```

Figura 14 - Uso de *GET* en el *script* PHP para obtener los anuncios

Se ha utilizado la conexión de tipo *GET* para todas las peticiones que no requerían el envío de muchas variables para realizarse, como puedan ser la obtención de reservas, la obtención de la estructura de la partida, el inicio de sesión, así como todos los listados de datos de la perspectiva del regante, como los de usuarios, parcelas y reservas.

En la Figura 15 se muestra el *script* que, utilizando la operación *POST*, transmite los valores necesarios para crear una nueva reserva en el sistema.

```

<?php
/**
 * Añadir nueva reserva
 */

require 'Database.php';

if($_SERVER['REQUEST_METHOD'] == 'POST') {

    if (isset($_POST['inicio']) && isset($_POST['fin']) && isset($_POST['parcela'])) {

        $inicio = $_POST['inicio'];
        $fin = $_POST['fin'];
        $parcela = $_POST['parcela'];

        //Se crea la consulta para insertar la reserva
        $consulta = "INSERT INTO reservas (inicio,fin,parcela) VALUES (?,?,?,?)";
        // Se prepara la consulta
        $comando = Database::getInstance()->getDb()->prepare($consulta);
        // Se ejecuta con los valores proporcionados
        $comando->execute(array($inicio,$fin,$parcela));

        print json_encode(array('resultado' => "correcto"));
    }
    else{
        print json_encode(array('resultado' => "falta valor"));
    }
}

```

Figura 15 - Uso de *POST* en el *script* PHP de añadir reserva

Con la operación *POST* se han implementado las peticiones de edición y adición de datos, las cuales requieren muchos datos. Varios ejemplos de estas peticiones son la adición de reservas, usuarios y la edición de campos pertenecientes a registros de usuarios o parcelas.

Ambos *scripts* (Figuras 14 y 15) poseen la dependencia de otro *script*: Database.php. La conexión con la base de datos se realiza siempre a través de éste, por lo que es requerido por todos los scripts del servidor.

En el lado de la aplicación, se han implementado dos métodos para el envío de datos al servidor, uno basado en peticiones de tipo *GET* y en *POST*.

Para las peticiones *GET* simplemente hay que concatenar los campos que se desee en la propia URL. La Figura 16 muestra, por un lado, como se construye la petición *GET* y, a continuación, su uso para obtener los anuncios, siguiendo con el ejemplo anterior (el título SERVER_ID corresponde al nombre de dominio (DNS) del servidor).

```
private static String conexionBDGET(String direccion){
    HttpURLConnection urlConnection = null;
    StringBuilder result = new StringBuilder();
    try {
        URL url = new URL(SERVER_ID+direccion);
        urlConnection = (HttpURLConnection) url.openConnection();
        InputStream in = new BufferedInputStream(urlConnection.getInputStream());
        BufferedReader reader = new BufferedReader(new InputStreamReader(in));
        String line;
        while ((line = reader.readLine()) != null) {
            result.append(line);
        }
    }catch( Exception e) {
        return "sin conexion";
    }
    finally {
        urlConnection.disconnect();
    }
    return result.toString();
}

public static String getAnuncios(int destinatario){
    return conexionBDGET("anuncios.php?destinatario="+destinatario);
}
```

Figura 16 - Implementación de una petición GET

Por otro lado tenemos la petición *POST*, las cuales se realizan a una URL y se le codifican los valores a enviar, por lo que no se tiene la limitación de caracteres de las URL. La Figura 17 muestra el método implementado para las peticiones *POST*, la llamada que se realiza para añadir una reserva y la codificación de los valores.

```

private static String conexionBDPOST(String direccion,String valores){
    HttpURLConnection urlConnection = null;
    StringBuilder result = new StringBuilder();
    try {
        URL url = new URL(SERVER_ID+direccion);
        urlConnection = (HttpURLConnection) url.openConnection();
        urlConnection.setRequestMethod("POST");
        urlConnection.setDoOutput(true);
        OutputStreamWriter wr = new OutputStreamWriter(urlConnection.getOutputStream());
        wr.write(valores);
        wr.flush();
        BufferedReader rd = new BufferedReader(new InputStreamReader(urlConnection.getInputStream()));
        String line;
        while ((line = rd.readLine()) != null) {
            result.append(line);
        }
        wr.close();
        rd.close();
    }catch(Exception e){
        return "sin conexion";
    }
    finally {
        urlConnection.disconnect();
    }
    return result.toString();
}

public static String anyadirReserva(String valores){
    return conexionBDPOST("anyadirReserva.php",valores);
}

valores = URLEncoder.encode("inicio", "UTF-8") + "=" + URLEncoder.encode(ConversorFechas.fechaBD(inicioReserva), "UTF-8");
valores += "&" + URLEncoder.encode("fin", "UTF-8") + "=" + URLEncoder.encode(ConversorFechas.fechaBD(finReserva), "UTF-8");
valores += "&" + URLEncoder.encode("parcela", "UTF-8") + "=" + URLEncoder.encode(""+parcela.getIdParcela(), "UTF-8");

```

Figura 17 - Implementación de una petición *POST*

6.3. Librerías externas

Para el desarrollo de este sistema, a priori, no se consideraba necesaria la utilización de librerías externas al JDK de Java, ya que, se utilizan tipos de datos bastante comunes: cadenas de texto (*String*), números enteros (*Integer*) y fechas (*Date* o *Calendar*).

No obstante, cuando se comenzó a trabajar con fechas, se vio que la clase habitual para trabajar con fechas en Java (*Date*) no era suficiente, ya que, no provee apenas operaciones para trabajar con fechas, necesarias para la implementación de la aplicación, como por ejemplo, obtener el número de semana del año de una fecha dada. La clase *Calendar*, aunque más completa que *Date*, tampoco cubría las expectativas. Finalmente se decidió usar la librería Joda-Time, la cual ofrece una gran variedad de funciones ya implementadas para el manejo de fechas.

Se probaron varias funciones de la aplicación que incluían la utilización de fechas en los dos dispositivos móviles Android mencionados anteriormente con un resultado satisfactorio, por lo que se continuó el desarrollo con la utilización de dicha librería.

7. Funcionamiento de la aplicación

En este apartado se mostrarán las pantallas de la aplicación desarrollada y se explicarán algunos detalles de su funcionamiento.

Inicio de sesión

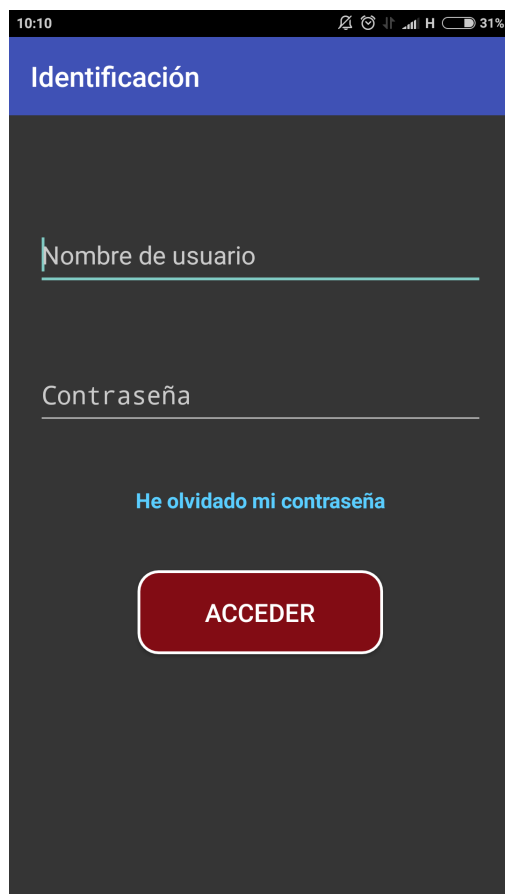


Figura 18 - Inicio de sesión

La primera pantalla que nos encontramos es la de inicio de sesión (Figura 18). Este apartado es común para el responsable y el regante ya que, la cuenta de usuario con la que se accede será la que determine qué perspectiva de la aplicación se utilizará. En ella, además, encontramos el botón que re-dirige a una pantalla donde poder recuperar tu contraseña mediante el correo electrónico.

Perspectiva del responsable

Selección de parcela

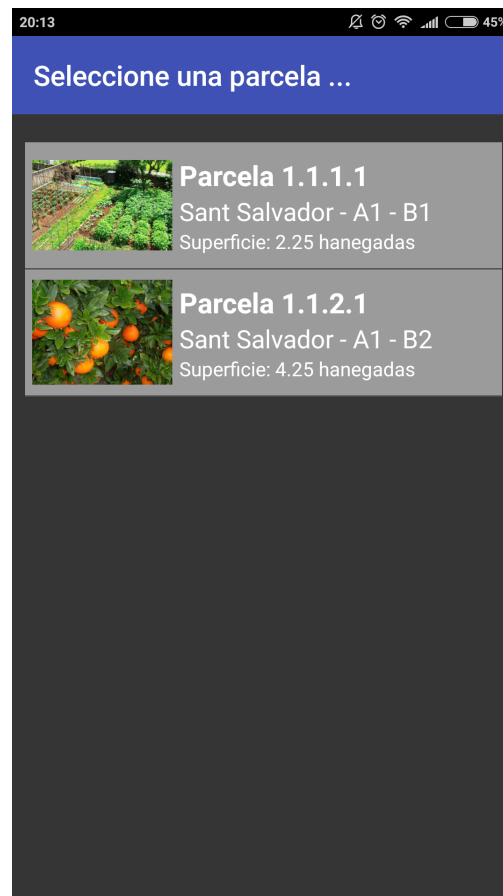


Figura 19 - Selección de parcela

Al iniciar sesión con un usuario de tipo responsable, si éste posee más de una parcela asociada, se le mostrará una lista con ellas para que elija con cual quiere trabajar (Figura 19). Por otra parte, si no tiene ninguna parcela asociada, se le seleccionara su parcela automáticamente y se mostrará la siguiente pantalla.

Pantalla principal

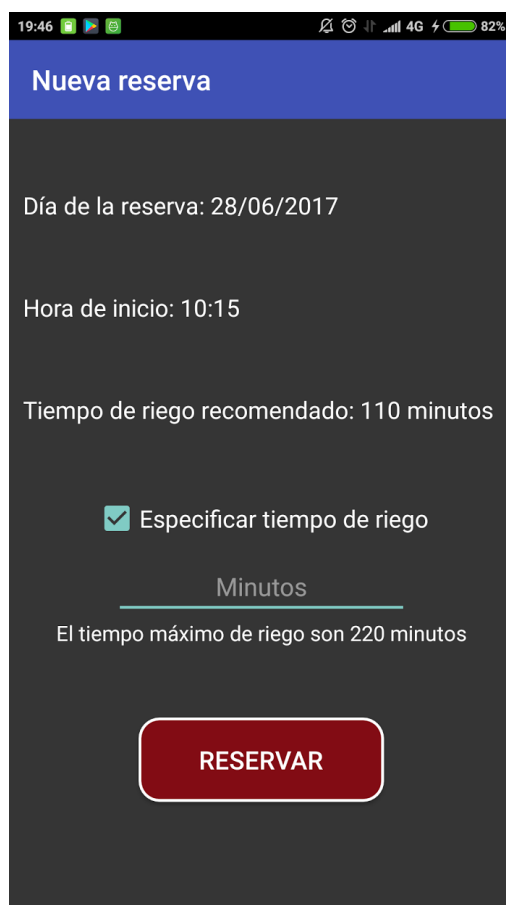


Figura 20 - Pantallas principales

Una vez seleccionada una parcela, la aplicación continúa con las pantallas principales (Figura 20). La primera que nos encontramos es la

sección de anuncios (arriba - izquierda), donde el usuario puede consultar tanto los anuncios globales como los destinados a él. Seguidamente, se encuentra el listado con las franjas horarias disponibles para reservar (arriba - derecha). Éstas dependerán del brazo al cual pertenezca la parcela, pudiendo elegir entre la semana actual y la siguiente. A continuación se encuentra el historial de reservas de la parcela (abajo - izquierda) y, finalmente, la sección de detalles de la parcela (abajo - derecha).

Realizar reserva



19:46 4G 82%

Nueva reserva

Día de la reserva: 28/06/2017

Hora de inicio: 10:15

Tiempo de riego recomendado: 110 minutos

Especificar tiempo de riego

Minutos

El tiempo máximo de riego son 220 minutos

RESERVAR

Figura 21 - Realizar reserva

Al seleccionar una franja horaria, se mostrará una pantalla (Figura 21) con la información de inicio de la reserva, el tiempo de riego por defecto de la parcela y un apartado para, en caso de querer regar más o menos tiempo, especificar el tiempo de riego.

Visualizar y cancelar reserva



Figura 22 - Visualizar y cancelar reserva

Al seleccionar una franja horaria no disponible o pendiente, se mostrará una pantalla (Figura 22) con la información de la reserva, es decir, fecha de inicio, fecha de finalización, número de parcela asociado y el nombre del responsable de la parcela (izquierda). En caso de que la reserva sea del usuario y no falten menos de dos horas para el inicio de la reserva, se mostrará un botón para cancelar la reserva (derecha).

Menú lateral

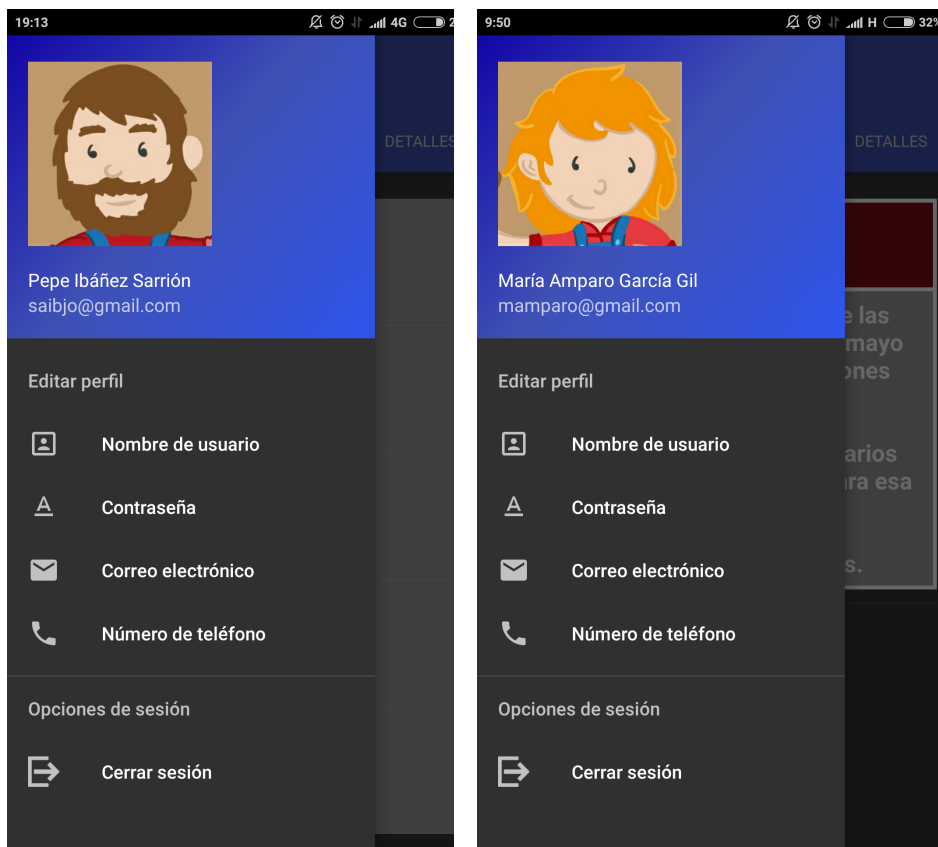


Figura 23 - Menú lateral

Mientras el usuario se encuentre en la pantalla principal de la aplicación, al deslizar desde el borde izquierdo o pulsando en el icono ≡, tendrá disponible un menú lateral (Figura 23) en el cual se encuentran las opciones de edición de perfil.

Perspectiva del regante

Menú principal

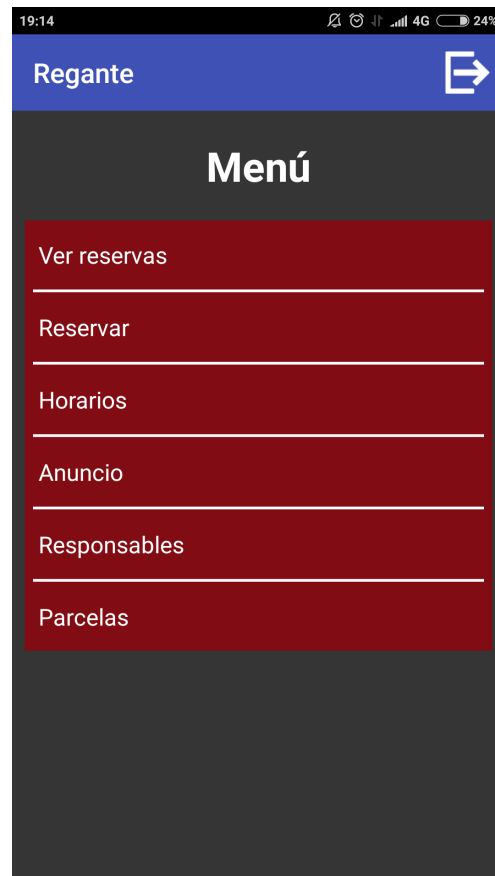


Figura 24 - Menú principal

Cuando se inicia sesión con el usuario regante, la primera pantalla que aparece es un menú (Figura 24) con las distintas funcionalidades implementadas para el regante.

Listado de reservas



Figura 25 - Listado de reservas

Al elegir la opción “Ver reservas”, se muestra con un listado de las reservas programadas en la aplicación (Figura 25 - izquierda), pudiendo elegir para que se muestren las de la semana actual y las de la siguiente.

Al seleccionar una reserva del listado, se muestra una pantalla (Figura 25 - derecha) muy similar a la que poseen los responsables, pero con un añadido. Al igual que los responsables, el regante puede cancelar una reserva, pero además debe añadir el motivo de la cancelación de la reserva, para que el responsable sea notificado con un anuncio y un correo electrónico.

Reservar



Figura 26 - Reservar

Al seleccionar la opción del menú “Reservar”, y una vez seleccionados el responsable (Figura 26 - izquierda) y parcela (Figura 26 - centro), al igual que en la perspectiva de los responsables, se muestran una lista de franjas para reservar (Figura 26 - derecha), pudiendo elegir entre la semana actual y la siguiente.

Horarios de los brazos



Figura 27 - Horarios de los brazos

Al seleccionar la opción "Horarios", la aplicación nos mostrará el listado de horarios registrados (Figura 27 - izquierda). Al seleccionar uno, se podrá eliminar, ya sea porque es incorrecto, o porque ya ha pasado y son datos desechables de la aplicación.

Además del listado de horarios almacenados en la aplicación, pulsando el botón "Nuevo", se una nueva pantalla para poder añadir horarios nuevos (Figura 27 - derecha).

Nuevo anuncio

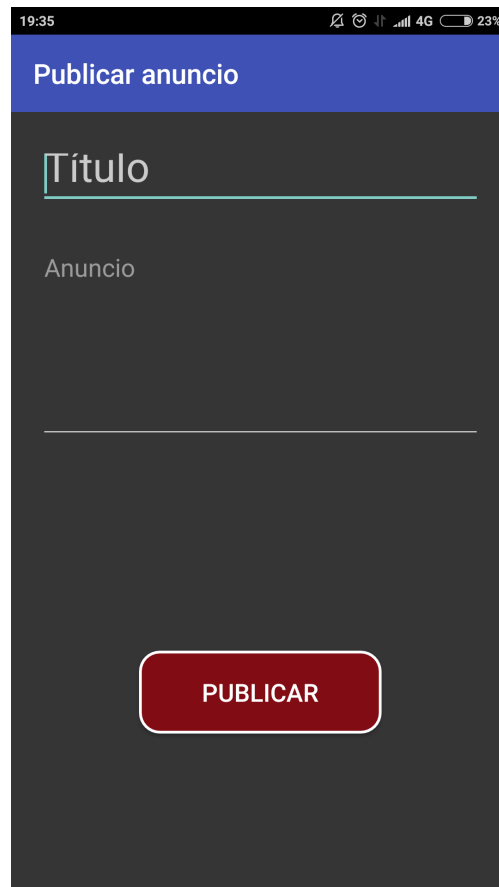


Figura 28 - Nuevo anuncio

Al seleccionar la opción “Anuncio” del menú, se mostrará una nueva pantalla para publicar un nuevo anuncio (Figura 28), donde será necesario especificar el título del anuncio y el anuncio en sí.

Editar información del responsable

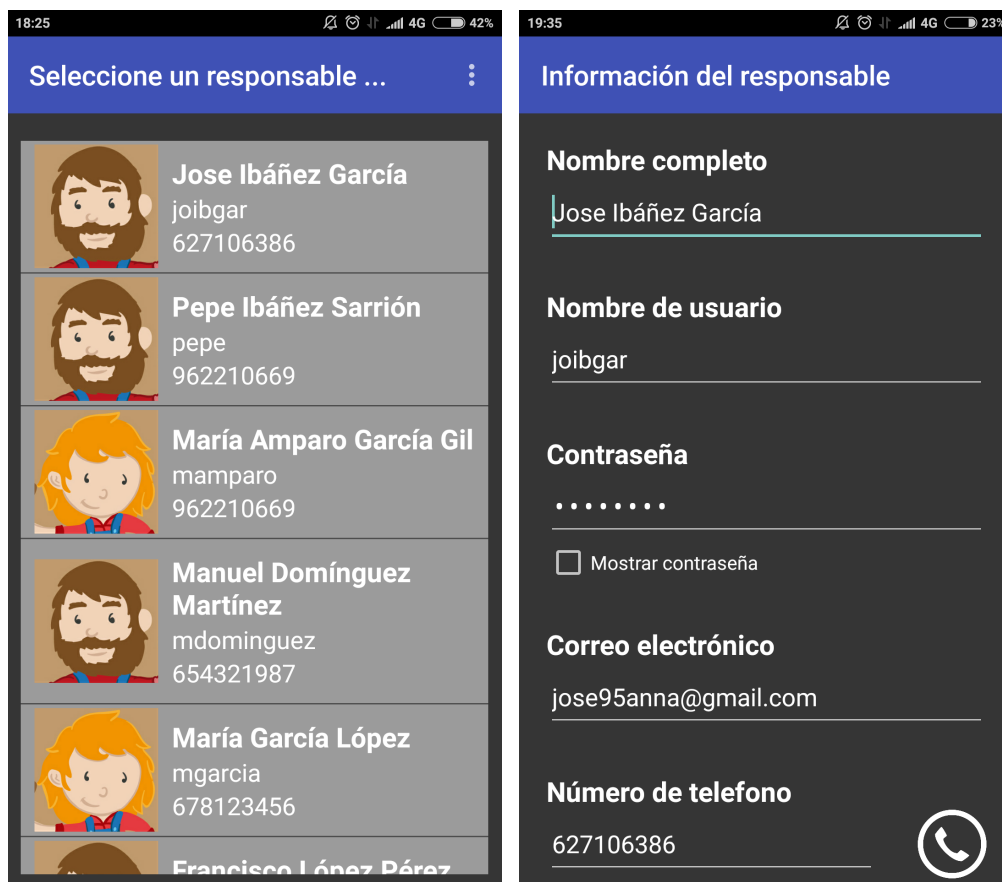


Figura 29 - Información del responsable

Al seleccionar la opción del menú principal “Responsables” y seleccionar un responsable de la lista (Figura 29 - izquierda), se pueden ver y editar todos los campos del perfil (Figura 29 - derecha). Ésta misma pantalla se utiliza también para añadir nuevos perfiles al sistema.

Editar información de la parcela

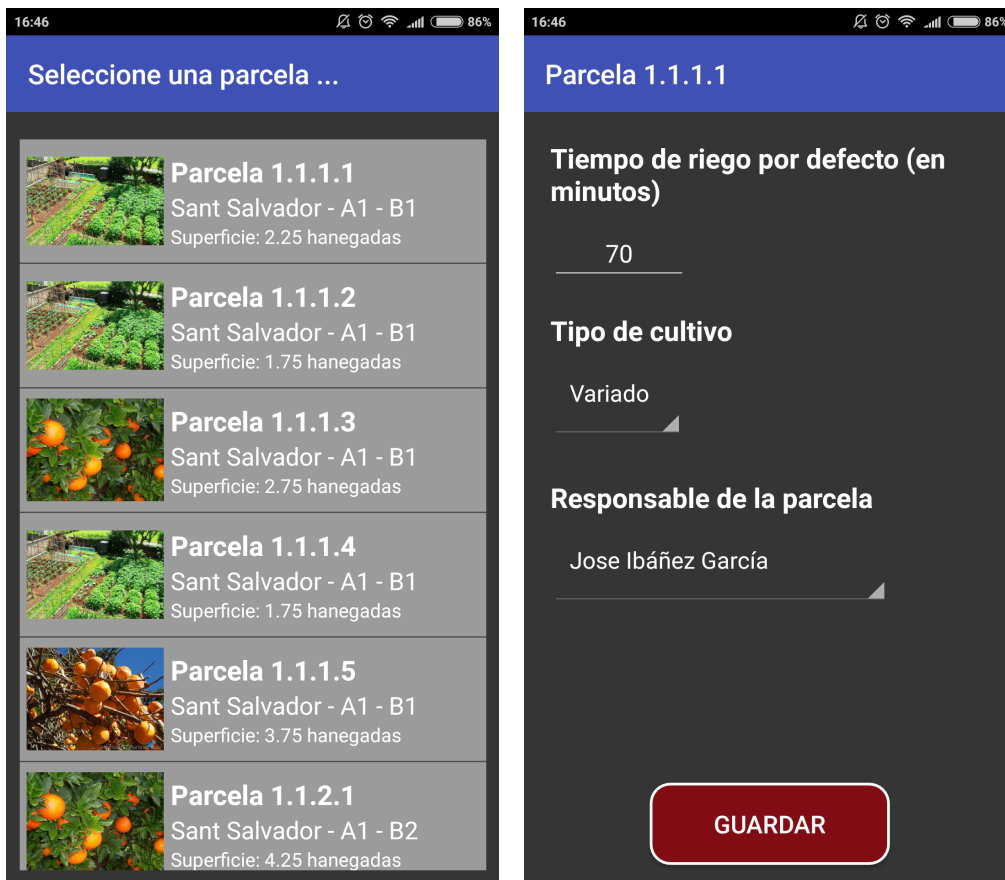


Figura 30 - Información de la parcela

Seleccionando la opción “Parcelas” del menú del regante, se mostrará un listado con todas las parcelas registradas (Figura 30 - izquierda). Al seleccionar una, se muestra la información no estática de la parcela para poder editarla (Figura 30 - derecha).

8. Conclusiones y trabajo futuro

Para finalizar, se expondrán las conclusiones del trabajo, tanto técnicas como personales y se citarán algunas ideas que corresponden al trabajo que todavía se podría hacer para ampliar y mejorar la aplicación.

8.1. Conclusiones

Una vez concluido el trabajo, solo puedo estar satisfecho por el trabajo realizado y por los conocimientos adquiridos. Cuando comencé, tenía unos pocos conocimientos sobre la programación Android, los cuales han aumentado considerablemente.

Otro apartado, desconocido hasta que comencé el proyecto, es la configuración de servidores, ya que no tenía conocimientos sobre Apache ni MySQL. Además, he aprendido los conceptos básicos del lenguaje PHP, el cual conocía, pero no había trabajado nunca con él.

Además de los apartados técnicos del desarrollo de la aplicación, también me ha sido muy interesante conocer el funcionamiento de las partidas de riego de otros lugares, ya que, además, la zona de Cullera y alrededores está, mayormente, ocupada por cultivos como el arrozal y el naranjo, los cuales son los cultivos más característicos de la zona valenciana.

En cuanto al aspecto técnico, se han cumplido los objetivos, ya que se ha desarrollado una aplicación de manejo sencillo, ya que está dirigida a usuarios de bajo perfil tecnológico.

Además, es una herramienta útil que cumple con su principal propósito: agilizar el proceso de reserva de agua para el riego, ya que no es necesario el contacto directo con el regante para estar al tanto de cuando se puede (o no) disponer de los recursos para el riego, sino que desde la aplicación se puede tener toda la información.

Finalmente, en términos de rendimiento, ésta ha sido probada en dos dispositivos móviles muy diferentes, uno bastante actual y potente, y otro más antiguo y sencillo. Los resultados obtenidos de la comparación nos hacen pensar que la aplicación se comportará de manera óptima en cualquier dispositivo.

8.2. Trabajo futuro

En cuanto a las posibles mejoras y ampliaciones de la aplicación, a continuación se proponen una serie de ideas para mejorar la aplicación para hacerla más funcional. A continuación se muestran dichas ideas:

- **Riego simultáneo de más de una parcela:** Existirán periodos de tiempo en los cuales habrán suficientes recursos hídricos como para que se puedan abrir varias acequias o brazos a la vez, por lo que la aplicación debería permitir la existencia de reservas simultáneas.
- **Gestión de incidencias:** Sería de gran utilidad poder contar con un apartado donde los responsables de las parcelas puedan reportar cualquier problema al regante, ya sean averías en la estructura de riego o problemas con otro responsable que no respeta los turnos.
- **Mejoras en la integridad y confidencialidad de los datos:** Cambiar el tipo de conexión de HTTP a HTTPS, ya que ésta última permite la encriptación de los datos transmitidos entre la aplicación y el servidor, útil por ejemplo para los datos personales sensibles.
- **Geolocalización de las parcelas:** Incluir la localización de las parcelas para poder visualizarlas en el mapa, para ayudar al regante, por ejemplo, a identificar las parcelas que tienen reservas o la que se está regando en un momento determinado.

9. Bibliografía

Documentos web

Características técnicas de los smartphones

- www.smart-gsm.com

Documentación técnica de Android

- androidstudiofaqs.com
- developer.android.com

Documentación oficial Java

- docs.oracle.com

Documentación oficial PHP

- php.net

Documentación oficial Apache

- wiki.apache.org

Documentación oficial MySQL

- dev.mysql.com

Librería Joda-Time

- www.joda.org

Libros

- *El Gran Libro de Android*, Jesús Tomás Gironés, editorial Marcombo, año 2016.
- *The Apache Modules Book: Application Development with Apache*, Nick Kew, editorial Prentice Hall, año 2007.
- *Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5 (Learning Php, Mysql, Javascript, Css & Html5)*, Robin Nixon, editorial O'Reilly Media, año 2014.