



Universidad
Carlos III de Madrid

Escuela Politécnica Superior

GRADO EN INGENIERÍA INFORMÁTICA

TRABAJO FIN DE GRADO:

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE
CONTROL REMOTO PARA EL REGADÍO DE
GRANADES PARCELAS

Autor: Sandra Juárez Puerta

Tutor: Javier Fernández Muñoz

*Aprende a enseñar,
enseñando aprenderás.
Hijo de Hombre – Phil Collins*

Agradecimientos

Mis más sinceros agradecimientos a:

Mi tutor Javier Fernández Muñoz por creer en mi idea y transformarla en un proyecto que seguiré desarrollando y mejorando en un futuro.

Profesores, maestros y todas aquellas personas que han sabido transmitirme todos sus conocimientos y pasión para que yo nunca deje de aprender e interesarme en las ciencias.

Todos mis compañeros, amigos y personas que se han interesado por mi proyecto y me han animado a hacerlo realidad.

A mi padre, mi familia y a mi novio por darme esta gran idea, ayudarme y creer en mi todo este tiempo.

Resumen

Este proyecto tiene como objetivo el desarrollo de un sistema de control para la automatización del riego en grandes parcelas.

Se implantará en una parcela un sistema de riego que pueda cambiar su estado de manera electrónica, con elementos tales como electroválvulas y bombas eléctricas que se conectarán a una placa con microcontrolador, en adelante llamada Arduino UNO, que modificará el estado de tales mecanismos para poner en funcionamiento o, en su defecto, detener el riego del cultivo de la parcela.

Para automatizar y poder realizar dichas tareas a distancia, el usuario tomará las decisiones pertinentes a través de una aplicación móvil para sistemas Android que se comunicará mediante SMS con la placa Arduino con conexión a un módulo GSM que interceptará dichos SMS.

Abstract

Currently, there are infinite possibilities to automate different tasks; however, the agriculture sector has lacked these improvements for a long time, which has motivated this project. By using an application, it will be possible to control an irrigation system for large fields remotely, thus not depending on the physical presence of the farmer and opening up a world of possibilities for both the farmer and the crops.

The general overview and objectives for this project are mentioned below, together with the factors which motivated its development.

General overview

In this project, we aim to develop an irrigation control system for large fields which will consist of two main parts: hardware with Arduino system and a GSM module directly connected to the actuating elements of the field – irrigation pump and solenoid valve – and the Android application which will enable the user to interact with the system.

Since this project is targeted to large fields, these will have to be divided into several plots of land, whose irrigation will remain independent for each of them; subsequently, fields will be provided with a hardware module to control every solenoid valve in each plot and an irrigation pump to allow passage of water.

Therefore, in order to activate the water flow through one or several plots, it will also be necessary to activate the irrigation pump belonging to those fields.

Regarding the hardware module, it will comprise an Arduino UNO with GSM connectivity through SIM card and an associated phone number. The Arduino will activate or deactivate the pump and the solenoid valves accordingly through

the application, which will connect hardware and user by preconfigured SMS messages.

Motivation

Technological advances in agriculture are not as noticeable as in other sectors. Most of the progress in communication and technology is oriented towards daily activities and urban life, and the internet has been the determining factor for the development of such big ideas.

Our lives have substantially improved thanks to these advances, together with the access to the internet and the current media. Nowadays, it is possible to automate and control certain tasks remotely, which has originated a revolutionary concept – the ‘Internet of Things’ (IoT).

Home automation is a clear example of this. Only some years ago, opening a closing a shutter remotely was unimaginable; however, systems automating household tasks have increased and diversified to such extent that they are sometimes a decisive factor when choosing a house.

There are infinite devices easily connectable between them by wireless networks, thus enabling us to automate and perform certain tasks remotely such as controlling the heating system, opening and closing the shutters and turning the light on and off.

As it has been stated above, agriculture has been deprived of these innovations which would facilitate work and benefit the sector, hence the idea of developing this innovation project to compensate its disadvantaged position.

Our aim is to improve the farmers’ quality of life by automating an eminently manual task, as the worker in charge had to commute daily to a particular place – several kilometres on average – to activate and/or deactivate this function according to the farmer’s criteria and the weather.

Due to the poor internet access in many rural areas, in addition to large distances, it is no surprise that the number of improvements in agriculture remains lower than in other sectors. For this reason, our project offers a remote solution for irrigation in large fields by means of text messages (SMS) and the global mobile communication system.

Objectives

The main aim of this final project is ‘to automate the traditional irrigation system in such a way that the worker can activate or deactivate it remotely, without the need for them to commute’.

Furthermore, some other objectives can be derived from the main one:

- To provide the user with a simple, useful and configurable application to manage irrigation in each field, as well as the plots into which they can be divided.
- To manage irrigation in fields which are divided into different plots, so that the user can operate either one or several valves.

Development of the proposal

We intend to create a mobile application which is easy and accessible at the same time, enabling us to communicate with the embedded system thanks to the Global System for Mobile Communications (GSM). There are plenty of mobile devices where this application can be run, but they are normally classified into three main groups depending on their operating system: Windows 10 Mobile, iOS and Android.

Out of these three, Android has the biggest market share, followed far behind by iOS and Windows 10 Mobile. The catalogue of devices running Android is vast not only regarding the price range but also in their features, allowing users to easily access its functions.

Moreover, it is worth mentioning the work of Android’s wide developer community and their focus on freeware, which boosts the development of applications for this operating system.

For all of the above and thanks to its well-proven experience, Android has consolidated as the greatest operating system for mobile devices and therefore, it has been the chosen OS for our project.



Ilustración 1: Android

On the other hand, it is essential to look at the main function to be performed by the platform prior to its installation in the field:

- ✓ To receive a SMS message with key information in order to run the connected hardware devices, thus allowing water in the field or, otherwise, blocking it.

Taking the above into account, the chosen device will have to be intended for interacting with other actuating elements. For these cases, a microcontroller is the best choice since it is oriented towards connectivity with other hardware components, whereas single-board computers focus on data processing.

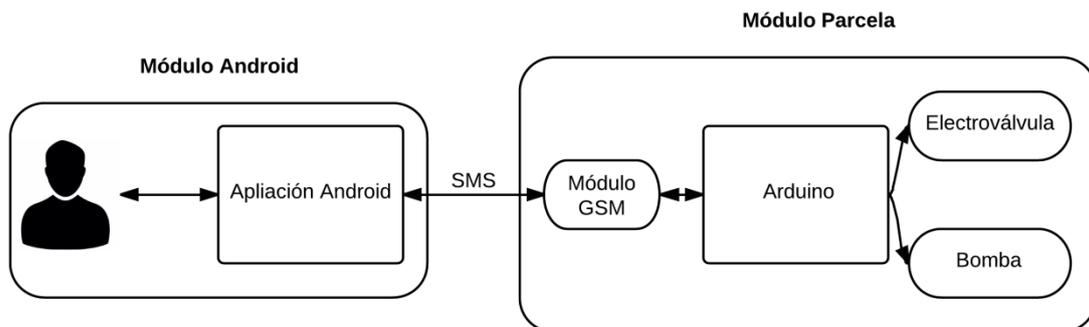
Arduino UNO seems the most suitable platform for our needs, as it aims to make electronics accessible for everyone in a simple, cheap way so that it can be a part of different multidisciplinary projects; besides, Arduino community provides users with help, support and countless projects.



Ilustración 2: Arduino UNO

This project revolves around an irrigation control system consisting of two distinct modules described below:

- The first module will be called Android Module and it will comprise the final user and a mobile application for him/her to control irrigation. The final user will interact with this module only.
- The second module will be called Hardware Module and it will be set up in the field, comprising an irrigation pump, as many solenoid valves as there are plots in the field, and finally there will be an Arduino UNO board with the corresponding GSM module and SIM card.



These two modules need a means of communication with each other, so that the user can interact with the irrigation system through the Android application. This communication will be established thanks to the mobile network (GSM) and to the sending and receiving of SMS messages.

SMS messages will contain key information and a predefined format, and allow the user to activate or deactivate irrigation, as well as to know the system status at any time.

In order to better explain the functioning of the system, two possible scenarios will be detailed below:

- The Arduino board starts up for the first time / reboots whether accidentally or on purpose.
- The Arduino board has just started up / rebooted.

The Arduino board starts up / reboots

This scenario occurs when the Arduino starts up for the first time or reboots, whether it has been accidentally or intended by the user.

In both cases, Arduino needs to check the status of the components to which it is connected. The communication process will work as follows:

- i. First of all, Arduino will send a message to the SIM card phone number, which belongs to the mobile phone running the Android application.
- ii. The application receives a configuration message from the field, gathers status from each plot and send them back to the field.
- iii. Arduino receives a message with the configuration to apply and sets up the status of every component accordingly.

Considering that a client-server model has a *client* which starts communication and a *server* providing a service, the Arduino will act as a client and the Android application will have the role of server.

The Arduino board has just started up / rebooted

This second scenario occurs when the Arduino has already carried out the configuration following the startup or reboot.

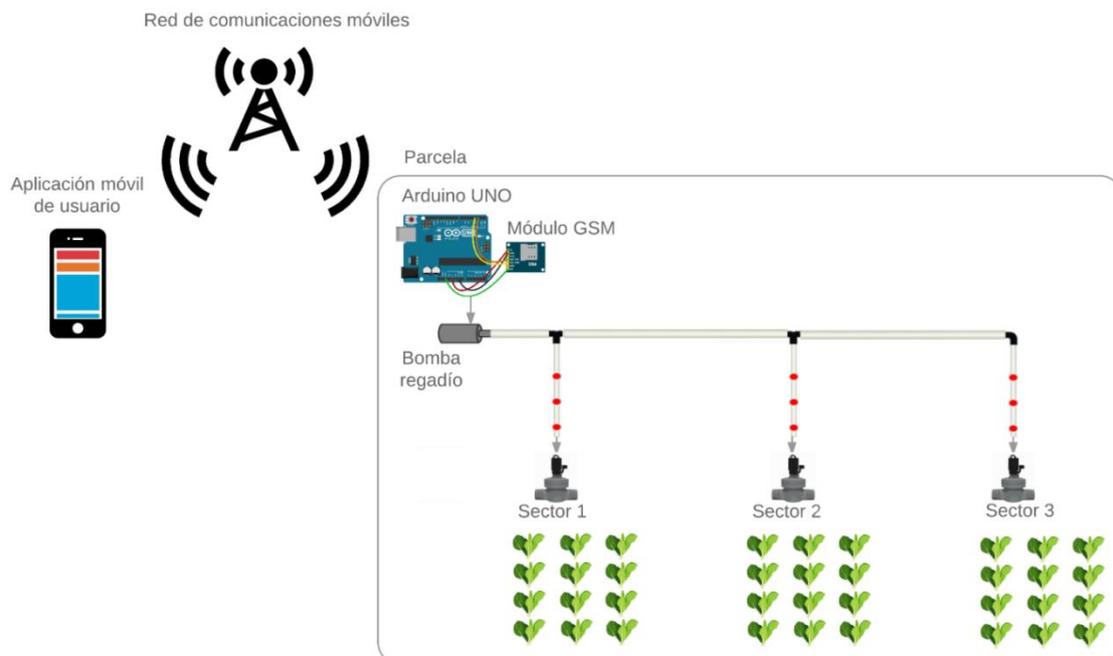
The system communicates as described below:

- i. The user starts communication with the irrigation system through the Android application.

- ii. The Android application communicates with the hardware module through a SMS message. This message will contain an action for the irrigation system to perform.
- iii. The message is received by the GSM module and sent again to the Arduino, which will interpret and perform the appropriate action.
- iv. Once the action has been successfully performed, Arduino will send a confirmation SMS message to the Android application.
- v. The Android application will receive the confirmation message and update the status of the system and its components on the application.

Since the Android application starts communication with the Arduino and waits for it to perform the requested action, the roles are switched this time: the Android application will be the client while Arduino will work as a server.

The following figure represents the irrigation system as a whole, with the Android Module to the left and the Hardware Module to the right. The two modules communicate with each other by SMS messages, which are sent by means of GSM and a SIM card attached to each module.



The Arduino board will be connected to the GSM module, the irrigation pump and the solenoid valves controlling the passage of water in each plot.

The GSM module is in charge of receiving the SMS message and sending it to the Arduino through the serial port so as to interpret and process it. Once the Arduino has processed the message, it will perform the necessary actions on the components to which it is connected – irrigation pump and solenoid valves.

By way of example, these are the steps to follow in case the user wants to activate irrigation in a field and deactivate it later:

- A field with three plots is registered on the mobile application.
- Irrigation in Plot 1 is activated (no plots were active before):
 - The irrigation pump is activated.
 - Solenoid valve in Plot 1 opens.
- Irrigation in Plot 2 is activated:
 - Solenoid valve in Plot 2 opens.
- Irrigation in Plot 1 is deactivated:
 - Solenoid valve in Plot 1 closes.
- Irrigation in Plot 2 is deactivated (plots are no longer active):
 - Solenoid valve in Plot 2 closes.
 - The irrigation pump is deactivated.

Conclusions of the project

From the outset, the underlying motivation behind this project has been to fulfil the objectives proposed in section 1.3, and these are founded on the requirements belonging to previous chapters, so that achieving the former has resulted in the latter.

Without a doubt, the objective summarising the motivation behind this final project has been successfully achieved:

- ✓ To automate the traditional irrigation system in such a way that the worker can activate or deactivate it remotely, without the need for them to commute.

Moreover, there were some objectives derived from the main one which were to improve the irrigation system usability and have also been accomplished:

- ✓ To provide the user with a simple, useful and configurable application to manage irrigation in each field, as well as the plots into which they can be divided.
- ✓ To manage irrigation in fields which are divided into different plots, so that the user can operate either one or several valves.

As resident of a village where agriculture is the main source of income, I have first-hand knowledge of this lack of automation in the plot and therefore, this final project intends to meet a very real need in my daily life.

With regard to the technology used, Arduino has proved itself as a very complete and accessible solution for all kinds of projects improving users' quality of life. Working with this microcontroller has opened a wide range of possibilities concerning everyday issues without an automatic alternative, for instance home automation.

These days, society is experiencing the strong influence of the 'Internet of Things', which benefits directly from a great internet access and products such as smartphones and Arduino-like platforms.

In this sense, Arduino is boosting innovation in order to satisfy the needs of the IoT, such is the case of home sensors, control circuits for surveillance cameras and Nest thermostats.

As a result, I have had the chance to enter this field and to acquire good knowledge of some great projects in home technology which will help us to connect humans to 'the things' surrounding us.

On the other hand, developing the Android application has allowed me a better understanding of this operating system, where brilliant ideas can be transformed into simple, intuitive apps made for everyone. Likewise, I have broadened my knowledge on Android and I have become more familiar with some of their many services.

Last but not least, this project only represents a little part of the possibilities and functionalities that this system may offer in the future. Starting from this basic proposal, our project is open to future lines of work as the ones described in the

previous section, with the purpose of creating a system which can be autonomously applied to the rural environment.

Future work

Due to the constraints of this project, what has been exposed is only a prototype of the fully functional model we envision for the future, which is to include the following features:

- In the final model, the Arduino board will be supplied by a solar panel. The components of this prototype, however, are connected to a local power supply.
- Depending on a previous study that should be carried out before installation, it will be decided which irrigation pump suits the field best. In our final project, a LED has simulated the performance of a pump, yet, it will be later substituted by the most suitable device.

Besides, new improvements and functions might eventually be added to the system, such as:

- Different sensors may provide useful information to the user, such as soil humidity or temperature, helping them make an efficient use of resources and, therefore, better decisions on each plot of the field.

In this case, a thorough study will have to be conducted in order to find the optimal spots to place those sensors.

- The previous sensors will enable to create a system where water resources can be managed in an autonomous way. On their part, the user may enter high and low ranges for activating and deactivating irrigation.
- Likewise, irrigation start time and end time for each plot may be included through time switch features.

Contenido

Agradecimientos	3
Resumen	4
Abstract.....	5
General overview.....	5
Motivation	6
Objectives	7
Development of the proposal.....	7
Conclusions of the project	12
Future work	14
Índice de ilustraciones.....	18
Índice de tablas	20
Capítulo 1	
Introducción	23
1.1 Visión general.....	23
1.2 Motivación.....	24
1.3 Objetivos	25
1.4 Estructura del documento	25
Capítulo 2	
Estado del arte	28
2.1 Sistemas de comunicación inalámbricos.....	28
2.2 Hardware.....	32
2.2.1 Computadores de placa reducida y microcontroladores	32
2.2.2 Otros componentes hardware.....	37
2.3 Sistemas operativos móviles	39
2.3.1 Windows Mobile	39

2.3.2	iOS.....	40
2.3.3	Android.....	40
2.3.4	Comparativa: sistemas operativos móviles	41
2.4	Proyectos similares	42
2.4.1	Sistemas con programador hardware.	43
2.4.2	Sistemas inteligentes con sensores.....	44
2.4.3	Sistemas basados en el Internet de las cosas (IoT)	45
2.4.4	Otros sistemas de riego	46
2.4.5	GreenIQ Smart Garden Hub.....	47
2.5	Conclusiones finales.....	48
 Capítulo 3		
Análisis del sistema		49
3.1	Especificación de requisitos	49
3.2	Requisitos de Usuario	50
3.2.1	Requisitos de Capacidad	50
3.2.2	Requisitos de Restricción	53
3.3	Casos de Uso	55
3.4	Requisitos de Sistema.....	59
3.4.1	Funcionales	59
3.4.2	No funcionales.....	62
3.5	Matriz de trazabilidad	64
3.5.1	Matriz de trazabilidad de requisitos de capacidad y requisitos de sistema funcionales	64
1.1.1	Matriz de trazabilidad de requisitos de restricción y requisitos de sistema no funcionales	65
1.1.2	Matriz de trazabilidad de requisitos de sistema funcionales y casos de uso	65
 Capítulo 4		
Diseño del sistema		67
2.1	Diseño e implementación del Hardware	72

2.2	Diseño e implementación del software	75
2.2.1	Diseño de la plataforma empotrada	75
2.2.2	Implementación de la plataforma empotrada.....	77
2.2.3	Diseño de la aplicación en Android.....	82
2.2.4	Implementación de la aplicación móvil en Android.....	90
Capítulo 5		
	Pruebas.....	96
5.1	Pruebas de validación.....	97
5.2	Matriz de trazabilidad de pruebas de funcionalidad y requisitos de sistema funcionales.....	100
Capítulo 6		
	Planificación del trabajo	101
Capítulo 7		
	Presupuesto	103
7.1	Resumen de tiempo dedicado	103
7.2	Coste de personal	104
7.3	Coste de Hardware	104
7.4	Coste de Software.....	105
7.5	Resumen de costes.....	105
Capítulo 8		
	Entorno socio-económico	106
Capítulo 9		
	Marco regulador	108
Capítulo 10		
	Conclusiones y trabajos futuros	109
10.1	Conclusiones del Proyecto	109
10.2	Conclusiones personales.....	110
10.3	Trabajos futuros.....	111
	Glosario	113
	Bibliografía	115

Índice de ilustraciones

Ilustración 1: Android.....	8
Ilustración 2: Arduino UNO	9
Ilustración 3: Redes inalámbricas (3).....	29
Ilustración 4: IoT en el Parque Nacional de Manu (Perú). Fuente: www.interempresas.net	31
Ilustración 5: versiones Android. Fuente: http://nksistemas.com	41
Ilustración 6: cuota mercado sistemas operativos móviles. Fuente: https://www.xatakamovil.com	42
Ilustración 7: sistemas con programador. Fuente: http://www.leroymerlin.es ...	43
Ilustración 8: sistemas inteligentes con sensores. Fuente: http://www.instructables.com	44
Ilustración 9: IoT. Fuente: http://ounae.com/gadgets-huerto-urbano/	45
Ilustración 10: Sistema IoT con Raspberry Pi. Fuente: http://chocolatesexyconsulting.es	46
Ilustración 11: sistema de control con temporizador y sensores para parcelas ...	46
Ilustración 12: GreenIQ Smart Garden Hub. Fuente: http://greeniq.co/product_tech.htm	48
Ilustración 13: casos de uso.....	55
Ilustración 14: módulos del sistema	67
Ilustración 15: diseño sistema	71
Ilustración 16: esquema conexión LED	72
Ilustración 17: esquema conexión electroválvula.....	73
Ilustración 18: esquema de conexión módulo GSM.....	74
Ilustración 19: funcionamiento software plataforma empotrada	76
Ilustración 20: funcionamiento del sistema en un inicio/reinicio.....	78
Ilustración 21: funcionamiento del sistema activación regadío.....	79
Ilustración 22: funcionamiento del sistema desactivación regadío.....	79
Ilustración 23: detección mensajes entrantes	81
Ilustración 24: esquema de navegación aplicación móvil.....	83

Ilustración 25: menú inicial.....	84
Ilustración 26: registrar parcela	85
Ilustración 27: insertar nombres sectores	86
Ilustración 28: lista parcelas	87
Ilustración 29: menú en lista parcelas.....	88
Ilustración 30: detalles de una parcela.....	89
Ilustración 31: detalles de un sector.....	90
Ilustración 32: modelo base de datos	91
Ilustración 33: registro del BroadcastReceiver en el archivo manifest.....	94
Ilustración 34: diagrama de Gantt.....	102

Índice de tablas

Tabla 1: descripción especificación de requisitos.....	49
Tabla 2 : Requisito de capacidad RC – 01.....	50
Tabla 3: requisito de capacidad RC – 02.....	50
Tabla 4: requisito de capacidad RC – 03.....	51
Tabla 5: requisito de capacidad RC – 04.....	51
Tabla 6: requisito de capacidad RC – 05.....	51
Tabla 7: requisito de capacidad RC – 06.....	51
Tabla 8: requisito de capacidad RC – 07.....	52
Tabla 9: requisito de capacidad RC – 08.....	52
Tabla 10: requisito de capacidad RC – 09.....	52
Tabla 11: requisito de capacidad RC – 10.....	52
Tabla 12: requisito de restricción RC – 01.....	53
Tabla 13: requisito de restricción RC – 02.....	53
Tabla 14: requisito de restricción RC – 03.....	53
Tabla 15: requisito de restricción RC – 04.....	53
Tabla 16: requisito de restricción RC – 05.....	54
Tabla 17: requisito de restricción RC – 06.....	54
Tabla 18: requisito de restricción RC – 07.....	54
Tabla 19: requisito de restricción RC – 08.....	54
Tabla 20: descripción especificación casos de uso.....	55
Tabla 21: caso de uso CU – 01.....	56
Tabla 22: caso de uso CU – 02.....	56
Tabla 23: caso de uso CU – 03.....	57
Tabla 24: caso de uso CU – 04.....	57
Tabla 25: caso de uso CU – 05.....	57
Tabla 26: caso de uso CU – 06.....	58
Tabla 27: caso de uso CU – 07.....	58
Tabla 28: caso de uso CU – 08.....	59
Tabla 29: requisito de sistema funcional RSF – 01.....	59

Tabla 30: requisito de sistema funcional RSF – 02	60
Tabla 31: requisito de sistema funcional RSF – 03	60
Tabla 32: requisito de sistema funcional RSF – 04	60
Tabla 33: requisito de sistema funcional RSF – 05	60
Tabla 34: requisito de sistema funcional RSF – 06	61
Tabla 35: requisito de sistema funcional RSF – 07	61
Tabla 36: requisito de sistema funcional RSF – 08	61
Tabla 37: requisito de sistema funcional RSF – 09	61
Tabla 38: requisito de sistema funcional RSF – 10	62
Tabla 39: requisito de sistema funcional RSF – 11	62
Tabla 40: requisito de sistema no funcional RSNF – 01	62
Tabla 41: requisito de sistema no funcional RSNF – 02	62
Tabla 42: requisito de sistema no funcional RSNF – 03	63
Tabla 43: requisito de sistema no funcional RSNF – 04	63
Tabla 44: requisito de sistema no funcional RSNF – 05	63
Tabla 45: requisito de sistema no funcional RSNF – 06	63
Tabla 46: requisito de sistema no funcional RSNF – 07	64
Tabla 47: requisito de sistema no funcional RSNF – 08	64
Tabla 48: matriz de trazabilidad de requisitos de capacidad y de sistema funcionales.....	65
Tabla 49: matriz de trazabilidad de requisitos de restricción y de sistema no funcionales.....	65
Tabla 50: descripción matriz de trazabilidad de requisitos de sistema funcionales y casos de uso.....	66
Tabla 51: protocolo sms de configuración	77
Tabla 52: protocolo sms de activación/desactivación	78
Tabla 53: mapeo número de sector y pin.....	80
Tabla 54: descripción tablas de pruebas	96
Tabla 55: tabla prueba P – 01	97
Tabla 56: tabla prueba P – 02	97
Tabla 57: tabla prueba P – 03	98
Tabla 58: tabla prueba P – 04	98
Tabla 59: tabla prueba P – 05	98
Tabla 60: tabla prueba P - 06	98
Tabla 61: tabla prueba P – 07	99
Tabla 62: tabla prueba P – 08	99
Tabla 63: tabla prueba P – 10	99
Tabla 64: tabla prueba P – 11	99

Tabla 65: matriz trazabilidad de pruebas funcionales y requisitos de sistema
funcionales..... 100
Tabla 66: planificación del trabajo 101

Capítulo 1

Introducción

Actualmente contamos con infinitas posibilidades para la automatización de distintas tareas, sin embargo, el sector agrícola se ha visto desprovisto de estas nuevas ideas o mejoras, lo que ha promovido la realización de este proyecto. Por medio de una aplicación, se controlará a distancia un sistema de regadío para grandes parcelas, sin tener que atar al agricultor a presentarse físicamente en un sitio concreto. Esto abre un nuevo mundo de posibilidades de las que se beneficiarán el agricultor y sus cultivos.

Por otro lado, en este capítulo, primera sección de este trabajo, se exponen los objetivos y una visión general de este proyecto, así como la motivación para su desarrollo. También se explicará, más adelante, la estructura que seguirá este documento, seguida de un glosario de términos al final, para una correcta comprensión del trabajo.

1.1 Visión general

Con este proyecto se quiere desarrollar un sistema del control de regadío para parcelas extensas, formado por dos partes principales: la parte hardware que engloba el sistema formado por Arduino y un módulo GSM conectado directamente a los elementos actuadores de la Parcela (bomba de regadío y electroválvulas) y la aplicación móvil con la que interactuará el usuario en Android.

Debido a que este proyecto está orientado a parcelas con extensas superficies, será necesario dividir éstas en distintos sectores, pudiendo activar o desactivar el riego en ellos de manera independiente. Cada parcela contará con un módulo hardware que controlará todas las electroválvulas asociadas a cada sector y la bomba de regadío que será necesario permitir el paso de agua por el circuito.

Para que el paso de agua por los sectores de la parcela sea efectivo, se necesita poner en marcha una bomba de regadío que se asociará a cada parcela.

El módulo hardware que se situará en la parcela constará de una plataforma empotrada con conectividad GSM con una tarjeta SIM y un número de teléfono asociado a ésta. Con esto y la llegada de distintos SMS con un formato acordado con la aplicación móvil del usuario, la plataforma empotrada activará/desactivará la bomba y las electroválvulas pertinentes a la que esté conectado.

1.2 Motivación

El sector agrícola es un campo en el que los avances tecnológicos no han sido tan visibles como en otros. Nuestro día a día en el hogar y las grandes ciudades han acaparado los mejores progresos en tecnología y comunicación, siendo el acceso a internet el factor más determinante para el desarrollo de estas grandes ideas.

Nuestra vida se ha visto ampliamente mejorada gracias a estos avances junto con el acceso a internet y los medios de comunicación existentes a día de hoy. La automatización o control a distancia de ciertas tareas ahora es posible, naciendo así un concepto revolucionario: el Internet de las Cosas.

El Internet de las Cosas surge como un cambio en cómo las personas se relacionan con los objetos y cómo éstos, a su vez, se relacionan con otros objetos. (1)

Podemos ver un ejemplo claro en la domótica. Hace apenas unos años, subir o bajar una persiana de manera remota era impensable, sin embargo ahora, los sistemas capaces de automatizar las tareas del hogar han aumentado hasta convertirse incluso en un punto determinante a la hora de comprar o empezar a construir una nueva vivienda.

Contamos con innumerables dispositivos que pueden conectarse entre sí fácilmente con redes inalámbricas, permitiéndonos en todo momento automatizar y realizar a distancia ciertas tareas como pueden ser controlar la calefacción, subir y bajar persianas o encender las luces en una vivienda.

Surge entonces la idea de realizar un proyecto de innovación en un sector desfavorecido en cuanto a número de avances tecnológicos. Ya que como bien se ha explicado, el sector agrícola se ve mermado de estas nuevas ideas que mejoren y hagan más fácil el negocio del cultivo.

El objetivo es mejorar la calidad de vida de los agricultores automatizando una tarea hasta ahora manual: Se necesitaba que el propietario de una parcela se desplazara hasta donde ésta estuviera (varios kilómetros de recorrido) diariamente para que pudiera activar/desactivar el riego en ella en función de determinados criterios seguidos por el agricultor o dependiendo del tiempo atmosférico.

La imposibilidad del acceso a internet en muchos medios rurales además de las grandes distancias a las que se encuentran de las poblaciones, fomentan la reducción del desarrollo de nuevas ideas que mejoren el sector agrícola. Es aquí donde nace la motivación a desarrollar este proyecto: se quiere proponer una solución para el control a distancia del regadío de grandes parcelas por medio del envío de mensajes de texto (SMS), gracias al sistema global de comunicaciones móviles.

1.3 Objetivos

El objetivo principal de la realización de este Trabajo Fin de Grado, en adelante TFG, es el de automatizar el sistema de regadío actual, de tal manera que la persona encargada de su activación/desactivación pueda hacerlo desde cualquier lugar de manera remota, sin tener que realizar un desplazamiento.

A su vez, existen también otros objetivos derivados del anterior que se explican a continuación:

- Ofrecer al usuario una aplicación simple, útil y configurable para gestionar el riego de cada parcela, así como los diferentes sectores en los que puede dividirse ésta.
- Gestionar el riego de parcelas donde éstas quedarán divididas en diferentes sectores, de acuerdo a que, cada vez, el usuario pueda poner en funcionamiento sólo uno de estos o, en su defecto, varios a la vez.

1.4 Estructura del documento

A continuación se propone explicar cada uno de los diferentes apartados en los que se divide el presente documento, con la finalidad de agilizar y facilitar su lectura y comprensión.

1. Introducción

En este punto se manifiesta una introducción del proyecto desarrollado, explicando la motivación dada para su realización, los objetivos que se pretenden abordar, así como una explicación de los apartados en los que se divide este documento.

2. Estado del arte

Descripción del entorno donde se posiciona nuestro proyecto, así como las tecnologías utilizadas para su desarrollo. Se proporcionará un contexto al lector para que pueda comprender fácilmente el resto del documento.

3. Análisis del sistema

Listado de requisitos que describen todas las funcionalidades que debe realizar el sistema y los casos de uso asociados. Finalmente se añaden las matrices de trazabilidad entre requisitos.

4. Diseño e implementación del sistema

Descripción profunda del funcionamiento del sistema, explicando cómo se han diseñado e implementado todas las funcionalidades que éste debe realizar.

5. Pruebas

Apartado donde se recogen todas las pruebas realizadas para asegurar el correcto funcionamiento de todos los módulos implicados en este trabajo.

6. Planificación del trabajo

Aquí se expondrán las diferentes tareas realizadas durante la ejecución de este proyecto, así como el tiempo empleado en la elaboración de cada una de ellas, acompañado de un diagrama de Gantt.

7. Presupuesto

Desglose de los costes de personal implicado, software y hardware usados en la realización del proyecto, así como el impacto del proyecto.

8. Entorno socio-económico

Impacto social y económico del proyecto.

9. Marco regulador

Descripción de todos los aspectos legales asociados al desarrollo de este proyecto.

10. Conclusiones y trabajos futuros

En este apartado se proporciona un análisis del trabajo realizado, así como los diferentes problemas encontrados, junto con las posibles mejoras que se pueden incluir y realizar para perfeccionar este proyecto.

11. Glosario

Agrupación de los distintos términos usados en la redacción de esta memoria.

12. Bibliografía

Apartado que contiene todas las referencias bibliográficas usadas en el presente documento.

Capítulo 2

Estado del arte

El objetivo del siguiente apartado es exponer y justificar las distintas tecnologías utilizadas a lo largo del desarrollo de este proyecto. También se ofrecerá como una puesta en contexto del ámbito en el que se impulsará este sistema de riego y sus tecnologías, donde se enumerarán, también, varias soluciones ya implementadas que ofrecen un servicio semejante. De este modo, este apartado constituye una investigación sobre el sector donde se enmarca este proyecto, que será finalmente la base para implementar el sistema ideado y convertirlo en un proyecto de éxito.

2.1 Sistemas de comunicación inalámbricos

Para empezar, en este proyecto se ha determinado que deben existir dos extremos que necesitan interactuar a gran distancia: usuario y sistema de riego. Ambos deben poder comunicarse con independencia del lugar y momento donde se encuentren. Se crea así un problema que será resuelto en este apartado.

Existen varias formas de comunicación y tecnologías que podrían implementarse en este proyecto. Sin embargo, sólo una ha sido elegida para ser desarrollada en este sistema automático de riego, siendo expuesta a continuación.

Los sistemas de comunicación inalámbricos son aquellos que permiten conectar dos extremos (llamados receptor y emisor) sin necesidad de estar unidos por un medio de propagación físico, es decir, sin cables, estableciendo entre ellos una comunicación exitosa utilizando el espectro electromagnético. (2)

Las ventajas de los sistemas de comunicación inalámbricos son claras: se puede establecer una comunicación flexible y con movilidad, contando también con gran capacidad para un número elevado de suscriptores. Sin embargo, esta tecnología

posee una velocidad de transmisión mucho menor a diferencia de las comunicaciones tradicionales (cableadas).

Es por esta gran movilidad que actualmente no podemos contemplar nuestro día a día sin esta tecnología. Estamos inmersos en la llamada revolución tecnológica de las comunicaciones inalámbricas.

Los usuarios de telefonía móvil superaron a los de la fija en sólo cinco años de la existencia de la primera. Esto se debe a que a que el usuario no tenía que tener una ubicación fija, además de que el medio de transmisión (el aire) ya estaba preparado para realizar una comunicación entre dos extremos. (2)

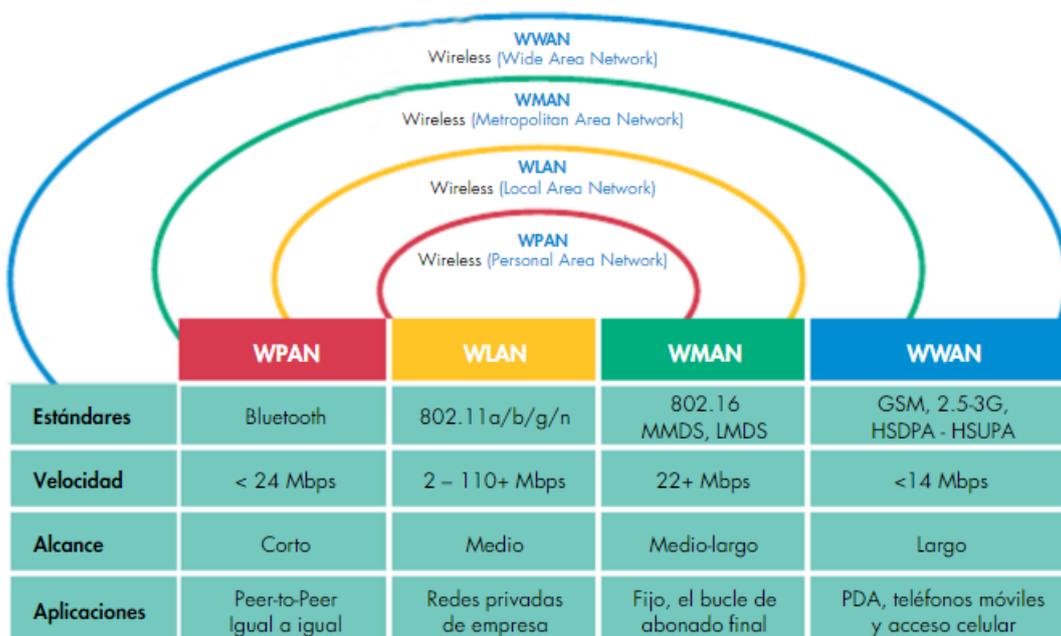


Ilustración 3: Redes inalámbricas (3)

Como se puede ver en la imagen superior, existen distintos tipos de tecnologías de comunicación inalámbrica, ordenadas de menor a mayor cobertura. Las redes de área personal (WPAN en rojo) serán en las que se necesite una mayor cercanía por parte de los participantes en la comunicación, mientras que en las redes WWAN, dichos usuarios podrán estar separados por una gran extensión de terreno.

A continuación se detalla brevemente cada una de las características de los distintos tipos de comunicación inalámbrica:

Wireless Personal Area Network (WPAN)

Permiten la conexión inalámbrica de dispositivos móviles de uso personal sin necesidad de contar con una infraestructura intermedia, es decir, de igual a igual, donde cada dispositivo tendrá que estar muy cerca del otro. Un ejemplo es el Bluetooth o el NFC.

Estas redes tienen un principal inconveniente que imposibilita su uso en este proyecto: los extremos de comunicación tienen que estar poco separados, desde unos centímetros al espacio límite de una habitación de moderadas dimensiones, dependiendo de la tecnología usada. (3)

Wireless Local Area Network (WLAN)

Nacen para conectar dispositivos que se encuentren a un máximo de 10 metros aproximadamente. Para estas redes es necesario disponer de un punto de acceso (PA), al cual se conectarán otros dispositivos permitiendo así la intercomunicación entre ellos gracias al acceso al PA.

Estas redes inalámbricas tampoco pueden usarse en este proyecto para la comunicación de la aplicación del usuario con el sistema de control de riego. Esto se debe a que se quiere que el usuario pueda controlar el sistema de riego desde cualquier lugar donde esté, teniendo estas redes un máximo de 10 metros de alcance.

Redes inalámbricas de área metropolitana (WMAN)

Las redes de área metropolitana tienen un mayor alcance, permitiendo conectar dispositivos en varias ubicaciones de áreas metropolitanas, como por ejemplo algunas zonas de ciudades, campus universitarios etc. (3)

Un ejemplo muy interesante de estas redes es el llamado ZigBee, caracterizado por su bajo consumo. Es frecuentemente usado para conectar distintos módulos a una central que controlará el funcionamiento de éstos.

En distintos parques de Perú han añadido esta tecnología para controlar el riego de sus parques. Se cuenta con distintos módulos que controlan el riego en distintas zonas de un parque y extraen información del lugar (temperatura, humedad, etc.). Éstos están conectados a una central de la que reciben órdenes y a su vez,

le envían la información obtenida de sus sensores. Por otra parte, la central también estará conectada a un servidor en la nube.

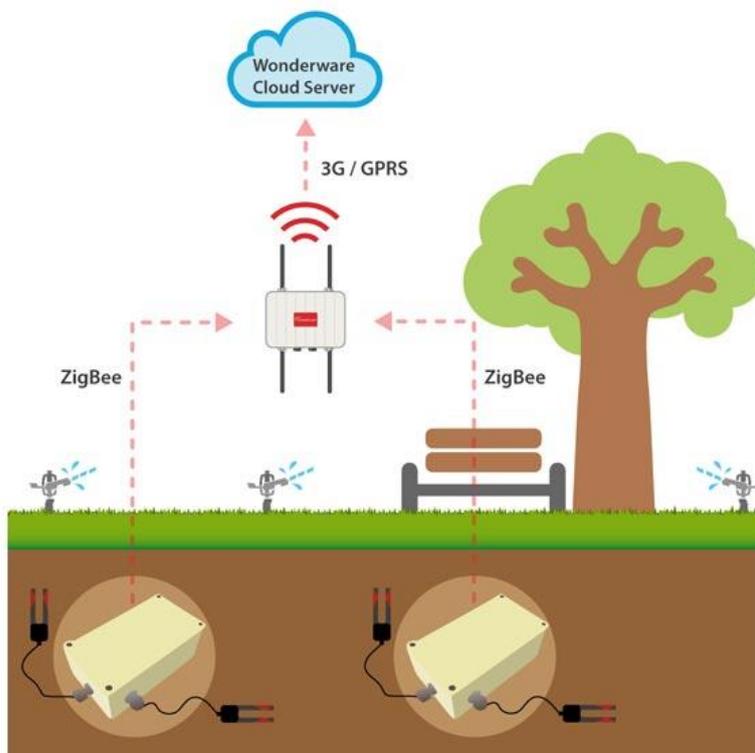


Ilustración 4: IoT en el Parque Nacional de Manu (Perú). Fuente: www.interempresas.net

Redes inalámbricas de gran alcance (WWAN)

Dan servicio a una extensa zona geográfica, teniendo el alcance más amplio que cualquiera de las mencionadas en apartados anteriores. Se usa tecnología de redes celulares de telecomunicaciones móviles como LTE, GPRS, EDGE, GSM, HSPA o 3G para la transmisión de datos. (3)

Son estas las redes elegidas para desarrollar este proyecto, donde el usuario será portador de una tarjeta SIM de la compañía de telecomunicaciones que estime oportuna, con la cual podrá comunicarse con el sistema de riego que también contará con otra SIM con diferente número de teléfono, permitiendo así la comunicación entre los dos extremos.

Debido al gran avance en las telecomunicaciones y la necesidad actual de estar conectado con el resto del mundo en cualquier lugar y en cualquier momento, son pocos los lugares donde el usuario no podría establecer conexión con el sistema de

regadío a través de esta tecnología. Gracias a esta virtud, las redes WWAN son elegidas para ser usadas como medio de comunicación en este proyecto.

2.2 Hardware

En el siguiente apartado se detallarán todos los elementos hardware usados en la implementación del proyecto, así como otros que por ciertas características no han sido elegidos.

2.2.1 Computadores de placa reducida y microcontroladores

Contamos con dos tipos distintos de dispositivos hardware posibles para desarrollar este proyecto, los cuales harán funcionar los distintos actuadores existentes. Éstos son los computadores de placa reducida y los microcontroladores.

Computadores de placa reducida

Los computadores de placa reducida o placas computadoras (SBC) son ordenadores completos donde todos sus componentes se encuentran en una sola placa de tamaño reducido. Estos componentes se basan en un microprocesador, memoria RAM, memoria principal, entradas y salidas, GPUS y diferentes puertos para ampliar su funcionamiento.

Tienen bajas prestaciones pero cumplen a la perfección las tareas para las que han sido creados.

Microcontroladores

Los controladores son circuitos integrados programables compuestos por una memoria, una CPU necesaria para el procesamiento y varias entradas y salidas. A día de hoy los componentes de un controlador han reducido su tamaño a un mínimo increíble, con lo cual se les ha dado el nombre de microcontroladores.

Como se ha mencionado anteriormente, se les llama programables debido a que es el usuario el que almacena en la memoria los datos que se convertirán después en las instrucciones a ejecutar por la CPU. Los microcontroladores por sí mismos no tienen una función útil, se les deberán asociar sensores o dispositivos

actuadores para procesar datos, realizar acciones o esperar eventos entre otras muchas más funcionalidades.

2.2.1.1 Raspberry Pi

Es uno de los más famosos computadores de placa simple, un pequeño ordenador creado en un principio para promover las ciencias de la computación en las escuelas, donde como principal ventaja presenta ser un componente muy accesible por su coste y tamaño. (4)

El componente está formado por un procesador Broadcom con memoria RAM y GPU, para la memoria de almacenamiento se cuenta con una ranura para una conectar una tarjeta MicroSD a juicio del usuario final. Adicionalmente, cuenta con puertos USB, HDMI, conector Ethernet y un conector para cámaras. También está dotado de 40 pines GPIO para conectar otros elementos hardware.

Necesita de un Sistema Operativo para funcionar, actualmente se desarrolla un SO con software open source llamado Raspbian adaptado de Debian. En cualquier caso, este dispositivo no está ligado solamente a este sistema operativo siendo posible instalar otros. Actualmente hay una gran comunidad de apoyo y soporte a este proyecto. (5)

Raspberry Pi 3 Modelo B

A continuación se añaden las especificaciones del último modelo de este computador de placa simple.

Especificaciones técnicas	
Modelo	Raspberry Pi 3 Modelo B
Memoria flash	Ranura para tarjeta MicroSD
RAM	1GB LPDDR2
CPU	1,2 GHz de cuatro núcleos ARM Cortex-A53
GPU	Dual Core VideoCore IV (®) Multimedia Co-procesador
Conectividad	Ethernet, 802.11 b / g / n LAN inalámbrica, Bluetooth, HDMI, jack 3,5mm, USB x4, conector GPIO y conector de cámara.



2.2.1.2 Arduino

Arduino es el nombre de una compañía y comunidad que diseña, produce y provee hardware libre, así como también software para interactuar con él.

Todo su hardware se basan en placas de circuito impreso a las que se añade un microcontrolador y puertos digitales y analógicos que sirven como entradas y salidas. (6) Además cuentan con un entorno de desarrollo IDE gratuito para programar sus placas.

A continuación se expone de manera breve las peculiaridades, funcionalidades y capacidades de la placa Arduino UNO.

Arduino UNO

Arduino UNO es una placa con microcontrolador basado en el microcontrolador ATmega328.

El lenguaje usado en los programas en Arduino UNO es C++, aunque no C++ puro, sino una adaptación. (7)

Puede ser programado a través del software Arduino IDE, un entorno fácil y gratuito, descargable desde la página oficial de Arduino.

Especificaciones técnicas

Modelo	Arduino UNO
Voltaje operativo	5V
Tensión de entrada recomendada	7-12V
Memoria flash	32KB
SRAM	2KB
Frecuencia reloj	16 MHz
EEPROM	1KB
Conectividad	14 pines digitales E/S, de los cuales 6 pueden usarse como salidas PWM y otros 6 analógicos que solo serán entradas.



Aún contando con estas recortadas características que nos parecerán inviables a día de hoy debido a la cada vez mayor potencia de todos los dispositivos que nos rodean, Arduino UNO es capaz de realizar las tareas para las que está pensado de forma rápida y eficiente.

Es también un dispositivo barato, accesible, fácil de usar y con una comunidad enorme de desarrollo y proyectos detrás, lo que lo posiciona en siempre en un buen lugar frente a sus competidores.

2.2.1.3 *Onion Omega*

Onion Omega es una placa computadora de un tamaño muy reducido que ejecutará una distribución basada en Linux llamada OpenWRT con arquitectura MIPS32.

Al igual que la Raspberry Pi son SBC pero Omega2 cuenta con unas prestaciones más recortadas como por ejemplo en el apartado de gráficos, ya que no incluye GPU. Sin embargo el punto fuerte de esta placa son sus pines para hardware que lo asemejan también a un microcontrolador, podría decirse que es un híbrido entre Raspberry Pi y Arduino. (8)

Onion Omega2

A continuación se añaden las especificaciones del último modelo de este computador de placa simple.

Especificaciones técnicas	
Modelo	Onion Omega2
Memoria flash	16 MB
RAM	64 MB
Frecuencia reloj	580 MHz
EEPROM	1KB
Conectividad	USB 2.0, b/g/n Wi-Fi, 15 GPIO, 2 PWM, 2 UART



2.2.1.4 Comparativa: computadores de placa reducida y microcontroladores

Para decidirnos entre una de las anteriores opciones hardware que controlarán el regadío de las parcelas, es pertinente elegir entre usar un computador de placa simple o un microcontrolador. Para ello se analizará la principal función que deberá realizar el componente hardware:

- ✓ Esperar a recibir un SMS con información clave para realizar una determinada acción sobre los dispositivos hardware a los que está conectado con el fin de permitir o no la entrada de agua en la parcela.

Como podemos ver, necesitamos que el dispositivo elegido esté bien adaptado para tratar con otros elementos actuadores. Para estos casos, la mejor elección es usar un microcontrolador, más orientado a la conectividad con otros componentes hardware que un computador de placa simple, el cual está más enfocado al procesamiento de datos.

Por otro lado, tenemos la placa Onion Omega2 que ofrece también un fácil soporte para estos elementos hardware y además cuenta con la ventaja de ejecutar un sistema operativo basado en Linux. Sin embargo, ésta es mucho menos conocida y por tanto, el soporte y la comunidad es mucho menor.

Finalmente, la placa Raspberry Pi tiene gran soporte y comunidad detrás pero su interacción con elementos hardware es menor que los dos dispositivos analizados en este apartado.

Se elegirá así Arduino para el desarrollo de este proyecto. Nace para poner a nuestro alcance la electrónica de manera fácil y barata y hacer que ésta pueda ser incluida en distintos proyectos multidisciplinarios sin límites de manera sencilla, teniendo una comunidad detrás de ayuda, soporte y proyectos enorme.

Finalmente a modo de resumen, los puntos por los que la plataforma Arduino es elegida son los siguientes:

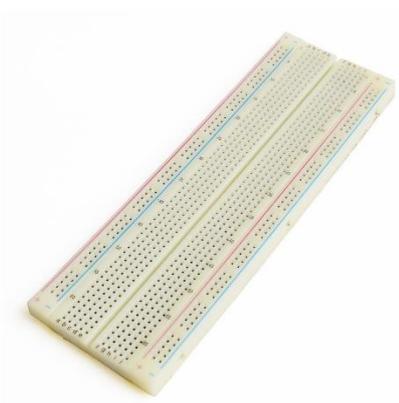
- ✓ Fácil conexión e interacción con elementos hardware.
- ✓ Poco consumo.
- ✓ Gran comunidad de soporte y ayuda.
- ✓ Facilidad de programación.

2.2.2 Otros componentes hardware

A continuación se realiza un análisis de las principales características técnicas del resto de componentes hardware usados en este proyecto.

Protoboard

Especificaciones técnicas	
Material	Plástico ABS blanco de alta calidad.
Contactos	Niquel-plata
Cables	Calibre 20AWG ~ 29 AWG
Dimensiones	16.5 x 5.5 x 1 cm
Puntos conexión	830



Relé

Especificaciones técnicas	
Marca	Songle
Modelo	srd-05vdc-sl-c
Capacidad de conmutación	10A
Bobina	5V
10A 250VAC	10A 125VAC
10A 30VDC	10A 28VDC



Módulo GSM

Este módulo se basa en el Sistema global de comunicaciones móviles (GSM), estándar en Europa, donde gracias al GSM podremos transmitir voz y datos (SMS, Internet). (9)

Especificaciones técnicas	
Modelo	WINGONEER SIMCOM SIM900 GPRS GSM
Alimentación externa	4.8 ~ 5VDC
Bandas	850/900/1800/1900 MHz
Comandos	Comandos AT
Funcionalidades	Llamadas, SMS, MMS y GPRS



Electroválvula

Especificaciones técnicas	
Modelo	Electroválvula JTV
Caudal	230 - 6800 l/h
Regulable	No
Medida	1"
Voltaje	12V



LED

Especificaciones técnicas	
Color	Rojo
Intensidad luminosa	30000 mcd
Intensidad corriente	20 mA
Voltaje	2.6 ~ 4.0



Especificaciones técnicas	
Modelo	Xiaomi Mi5 32GB
CPU	Qualcomm Snapdragon 820
Memorial RAM	3 GB
Memoria interna	32 GB
Sistema Operativo	Android 7.0 MIUI 8.2
GPU	Adreno 530

A white Xiaomi Mi5 smartphone is shown from a front-facing perspective. The screen displays the time 04:22 and the date Wednesday, February 24. Below the time, there are several app icons: Camera, Gallery, Music, Themes, Settings, App Store, Tools, and Security. At the bottom of the screen, there are four more icons: Phone, Contacts, Browser, and Messaging. The phone has a silver-colored metal frame and a camera lens with a flash on the back.

2.3 Sistemas operativos móviles

De cara a la realización de la parte que gestionará la comunicación del usuario con el sistema, se ha determinado realizar una aplicación para una plataforma móvil. La elección de realizar una aplicación para un dispositivo móvil viene motivada por el hecho de dotar de movilidad al sistema.

En primer lugar se elige un dispositivo móvil con soporte para tarjeta SIM, el cual deberá dar servicio GSM a través de una tarjeta SIM pudiendo recibir y/o enviar SMS al resto del sistema de control de riego situado en la parcela.

Actualmente, en el mercado hay varios sistemas operativos para dispositivos móviles, a continuación se exponen las virtudes de las que cada uno de ellos presume.

2.3.1 Windows Mobile

Es el sistema operativo móvil que ha desarrollado e implantado Microsoft en su familia de dispositivos móviles inteligentes.

Se trata de Software propietario y nace con el nombre de Pocket PC, dando servicio a equipos móviles con capacidades limitadas donde, posteriormente, es renovado y llamado Windows Mobile para una nueva línea de dispositivos.

Después de esto, se renovará completamente la interfaz y pasará a llamarse Windows Phone, pero con la salida del nuevo Windows 10 y, tras la decisión de

querer desfragmentar su amplio mercado de dispositivos, sacan Windows 10 Mobile, el actual sistema operativo móvil de Microsoft. (10)

2.3.2 iOS

Sistema operativo propiedad de Apple, creado exclusivamente para los dispositivos de este fabricante. Una de sus características más notables es que cuenta con una interfaz muy fluida y optimizada pues es un Sistema Operativo creado exclusivamente para unos pocos dispositivos. Esta es una de las virtudes de este sistema frente a otro como Android por ejemplo, el cual está presente en millones de dispositivos de compañías distintas. (11)

2.3.3 Android

Android es el sistema operativo más usado en la actualidad, ocupando la mayor cuota de mercado y estando presente en la mayoría de los smartphones existentes.

En un principio, fue desarrollado para dispositivos móviles táctiles, pero a día de hoy su uso se ha expandido a otros dispositivos diferentes tales como televisores, ordenadores portátiles, coches, relojes de pulsera o receptores decodificadores integrados. Como vemos, es fácil encontrar a día de hoy un dispositivo que implemente este sistema operativo.

Nace de la mano de un ex trabajador de Apple y Microsoft, en 2003, siendo propiedad de Android Inc., compañía que fue comprada en 2005 por Google.

Este sistema operativo está basado en el núcleo de Linux, dependiendo de éste para los servicios base del sistema, y se desarrolla de forma abierta, pudiendo acceder a su código fuente. (12)

En la actualidad Android cuenta con distintas versiones, cada una de ellas identificada por el nombre de un dulce, ordenadas alfabéticamente. En este presente momento alcanzan la letra N, Android 7.0 Nougat.



Ilustración 5: versiones Android. Fuente: <http://nksistemas.com>

A continuación se exponen las características de la versión más reciente de Android, lanzada el 22 de agosto de 2016:

Android 7.0 supone un gran avance en todos los aspectos posibles con sus predecesoras. El punto fuerte de esta aplicación es el de mejorar la interacción del usuario con las notificaciones de las distintas aplicaciones que puedan instalarse, creando nuevas formas de responder a éstas: ahora se muestra información completa de varios correos o distintas conversaciones dentro de una misma aplicación, pudiendo interactuar con ellas sin necesidad de acceder a la aplicación. También se agrega una nueva funcionalidad muy deseada para los usuarios de Android: poder usar dos aplicaciones a la vez gracias a la pantalla dividida. (13)

2.3.4 Comparativa: sistemas operativos móviles

Se pretende realizar una aplicación móvil sencilla al uso y accesible, que permita comunicarse con el sistema empujado gracias a las tecnologías inalámbricas dispuestas anteriormente. Contamos con multitud de dispositivos móviles donde podría ejecutarse esta aplicación, los cuales podríamos clasificarlos en tres grandes grupos, teniendo en cuenta el sistema operativo que implementan: Windows 10 Mobile, iOS o Android, plataformas que han sido descritas anteriormente.

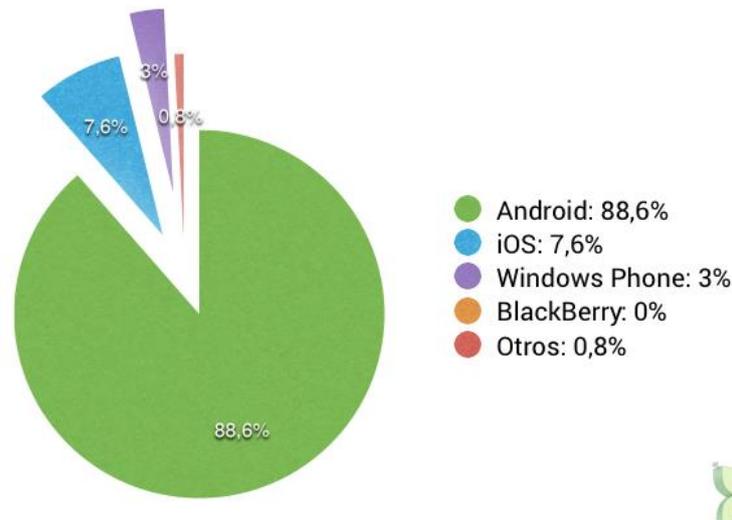


Ilustración 6: cuota mercado sistemas operativos móviles. Fuente: <https://www.xatakamovil.com>

De los siguientes, Android tiene mayor cuota de mercado seguido de lejos por iOS y Windows 10 Mobile en este orden. El catálogo de dispositivos con Android es inmenso tanto en precios, forma o prestaciones, teniendo el usuario gran facilidad para acceder a un dispositivo con este SO.

Además de lo anterior, se quiere destacar la amplia comunidad de desarrollo detrás de Android y su filosofía por el software libre, lo que ayuda sensiblemente al desarrollo de aplicaciones en este SO.

Es por esto y gracias al largo recorrido de Android que lo ha consolidado como el mayor sistema operativo usado en dispositivos móviles, será el SO elegido en el que desarrollar la aplicación móvil con la que interactuará el usuario final.

2.4 Proyectos similares

Actualmente se cuenta con multitud de sistemas de riego que varían en forma y componentes, pero con un sencillo objetivo común: dotar de agua una zona destinada al cultivo. En este apartado (y documento) nos centraremos en las diferentes formas de activar y desactivar el paso de agua por el sistema, aunque sin preocuparnos de los diferentes mecanismos de salida del agua como pueden ser aspersores o difusores, que quedarán a juicio del usuario final.

2.4.1 Sistemas con programador hardware.

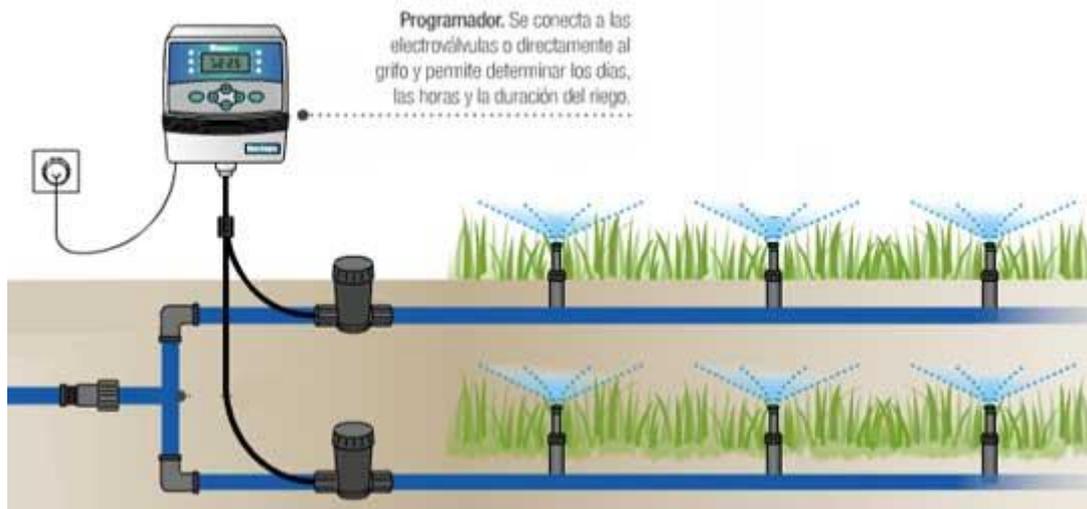


Ilustración 7: sistemas con programador. Fuente: <http://www.leroymerlin.es>

En primera instancia tenemos aquellos sistemas que cuentan con un temporizador programable por el usuario. Será éste quien especifique al sistema cuando tendrá que activar y desactivar el paso de agua en las horas/días/meses y durante el tiempo definido en tal dispositivo.

Es una de las opciones más fáciles de instalar debido a su largo periodo de implantación en el mercado y la facilidad de acceso al elemento controlador. En cualquier tienda de jardinería podríamos encontrar estos dispositivos.

Los temporizadores funcionan bien para pequeños cultivos o para zonas en las que se quiera regar de manera periódica, debido a que estos elementos no permiten una manipulación dinámica del riego por parte del usuario.

Sin embargo, en grandes zonas destinadas al cultivo, expuestas al aire libre, la realidad es otra: el tiempo atmosférico, la humedad del suelo o los diferentes criterios tomados por el agricultor pueden cambiar dinámicamente las necesidades de agua en el terreno, dificultando así la gestión del riego cuando se usan temporizadores.

2.4.2 Sistemas inteligentes con sensores

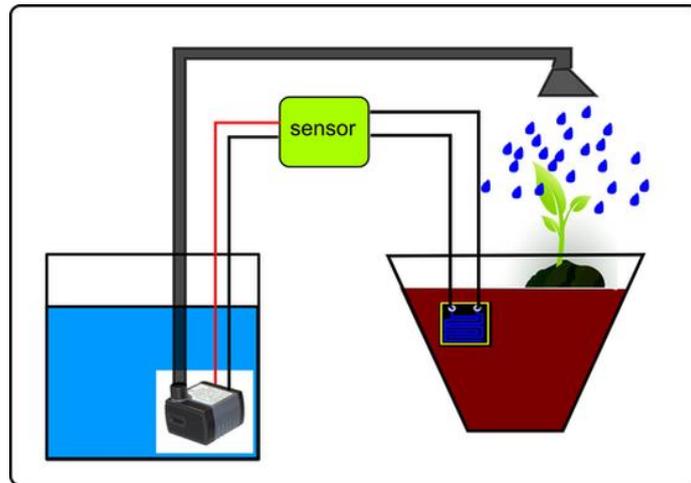


Ilustración 8: sistemas inteligentes con sensores. Fuente: <http://www.instructables.com>

Adicionalmente, se pueden encontrar otros prototipos de riego inteligentes basados en información que se proporciona al sistema de control mediante diferentes sensores situados en lugares clave.

Los sensores que habitualmente se usan son los de humedad, temperatura, lluvia, viento etc. Cuando el sistema de procesamiento de estos sensores considere oportuno, pondrá en funcionamiento el riego o, en su defecto, lo parará.

Están muy enfocados al ahorro de recursos debido a que sólo activarán el paso de agua en el caso que sea necesario. Por ejemplo, si la tierra no tiene la humedad suficiente y el ambiente es demasiado seco entonces se abrirá el paso de agua.

El criterio para activar o desactivar el regadío en sistemas que se basan en información extraída de distintos sensores viene determinado por el programador, el cual establecerá unos márgenes de actuación. Así mismo, otros sistemas más completos permiten cambiar a gusto del usuario final dichos márgenes.

A pesar del significativo ahorro en recursos del que estos sistemas se benefician, es difícil implantarlos en terrenos extensos debido a que los sensores tendrían que estar presentes en zonas demasiado separadas entre sí. A su vez, habría que colocar bastantes de estos dispositivos para poder obtener información real sobre el estado de todo el terreno a regar. Es así que el uso de este tipo de sistemas quedó descartado a la hora de desarrollar este proyecto.

Por otra parte, no se descartaría en un futuro ampliar la funcionalidad del sistema e incluir algún sensor de los mencionados anteriormente en este proyecto. Así el usuario podrá estar informado en todo momento del estado del terreno.

En esta primera implantación quedará descartada esta práctica debido a que la plataforma empotrada, la cual controla el paso de agua a través de la electroválvula, normalmente está alejada del cultivo en sí, por lo que en principio la suma de estos sensores al proyecto no parece realmente útil, aunque sí será reconsiderada en un futuro, donde podría conectarse un módulo inalámbrico que enviara la información de los sensores al módulo principal.

2.4.3 Sistemas basados en el Internet de las cosas (IoT)

Gracias a la alta conectividad con la que contamos a día de hoy, se pretenden solucionar y dotar de automatismo a las todas cosas que nos rodean y a todas tareas cotidianas, es aquí de donde nace el Internet de las cosas (IoT) que tiene una alta participación en proyectos similares al desarrollado en este documento.

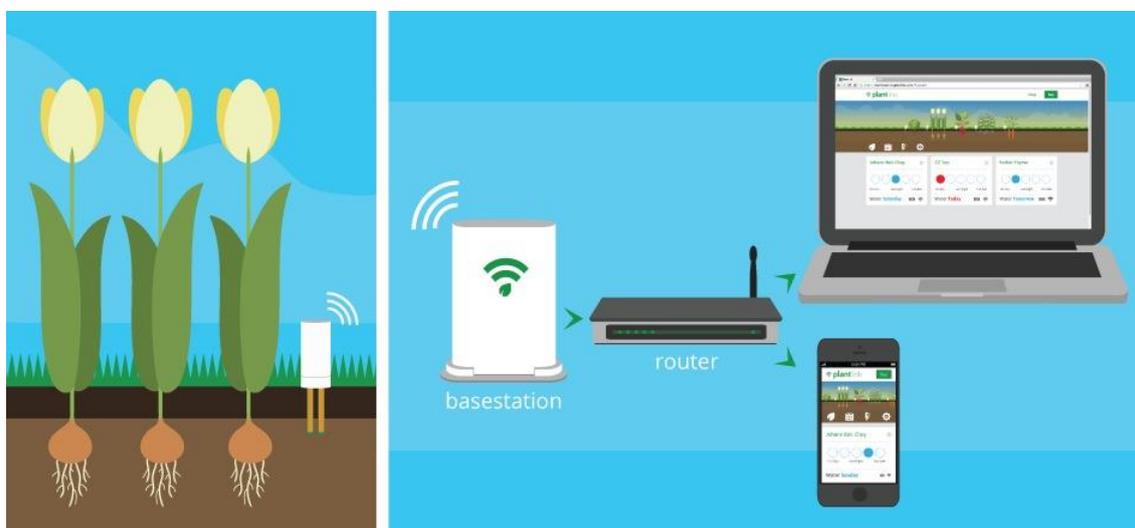


Ilustración 9: IoT. Fuente: <http://ounae.com/gadgets-huerto-urbano/>

Todos los proyectos inmersos en el llamado IoT cuentan con acceso a internet, permitiendo al usuario obtener información del sistema en todo momento y siéndole posible tomar decisiones sobre el control del riego esté donde esté, facilitando su uso y dando comodidad de actuación al propietario.

Usualmente, estos proyectos con acceso a Internet se dan en huertos pequeños en hogares donde mayormente se usará la tecnología Wi-Fi. Es así que este tipo de proyecto no se suele implantar en terrenos más extensos y alejados, donde el acceso a internet sólo se obtiene mediante GSM. Es por ello que este Trabajo Fin de Grado propone ofrecer una solución para tal problema.

Es ahora, con la gran popularidad que están tomando las pequeñas placas empotradas como Arduino o Raspberry Pi y con la facilidad de éstas para poder

conectarse a una red inalámbrica, que proyectos de IoT están emergiendo y entrando en la mayoría de hogares.

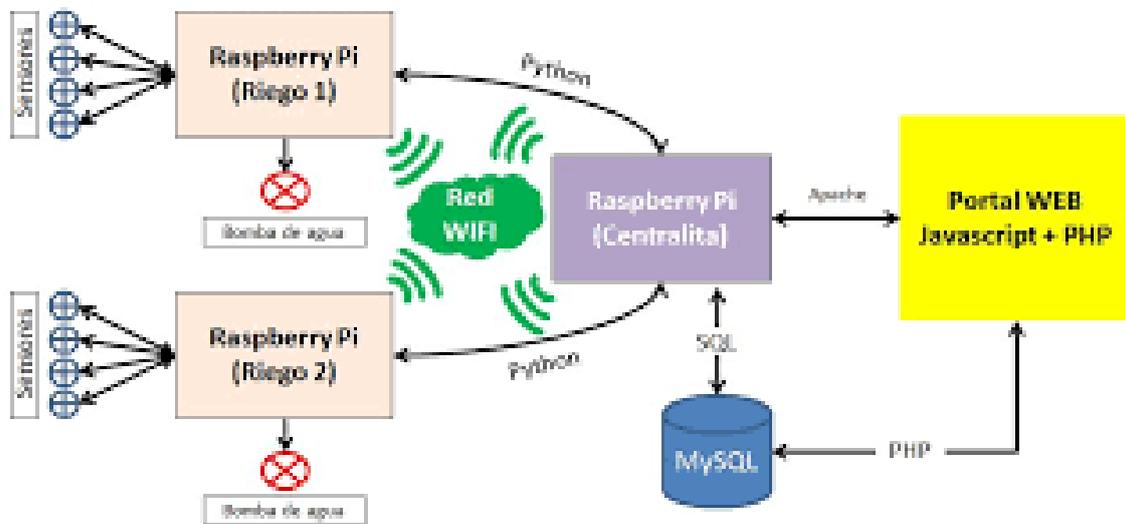


Ilustración 10: Sistema IoT con Raspberry Pi. Fuente: <http://chocolatesexyconsulting.es>

2.4.4 Otros sistemas de regadío

Por otro lado, en grandes terrenos, se están empezando a implantar los siguientes sistemas de regadío.

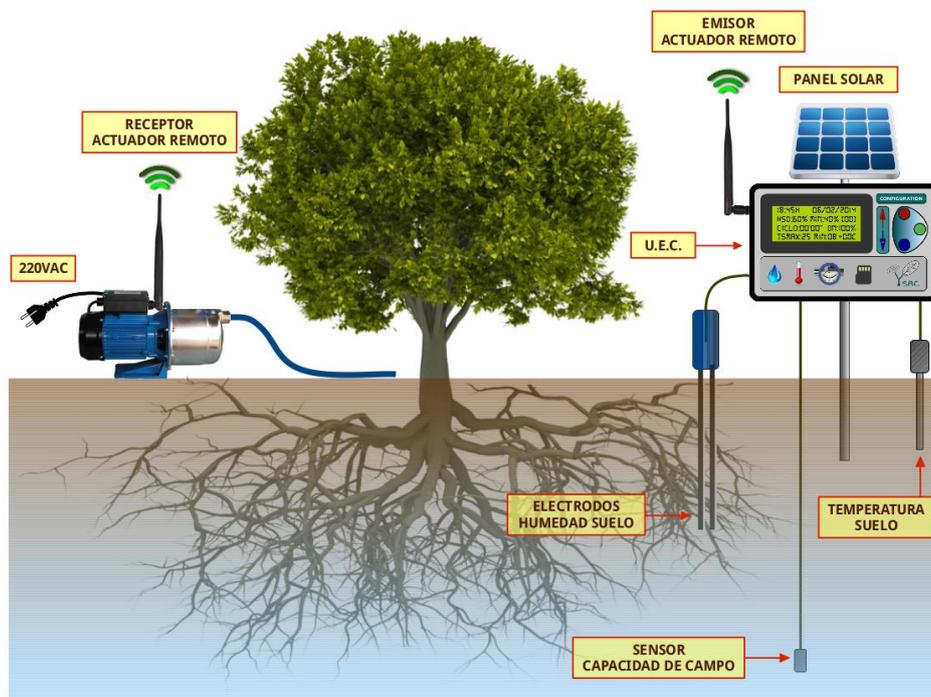


Ilustración 11: sistema de control con temporizador y sensores para parcelas

Una placa solar alimenta al controlador programable del sistema de riego, el cual se conectará con una bomba de riego y con las diferentes electroválvulas que permitirán el paso de agua por el circuito elegido.

Estos sistemas tienen enfocada su instalación en grandes terrenos, expuestos al aire libre, donde la única forma de ofrecer electricidad al sistema es mediante una placa solar.

El usuario puede programar estos sistemas de cara a que activen o desactiven el riego de forma periódica, especificando días, horas y duración. Además, gracias a la información que proporcionan diferentes sensores implantados en lugares clave, se podrá optimizar el aprovechamiento de recursos.

Como vemos cuentan con algunas de las características de los proyectos explicados en puntos anteriores.

Estos sistemas son los que se están implementando actualmente en los grandes cultivos pero cuentan, aún así, con una gran desventaja: no pueden ser programados a distancia. Debido a esto el usuario debe acudir junto al sistema de control para poder realizar los cambios requeridos. Siendo esta desventaja la principal motivación de realización de este Trabajo Fin de Grado.

2.4.5 GreenIQ Smart Garden Hub

Dentro del Internet de las Cosas encontramos GreenIQ Smart Garden Hub, un sistema orientado al riego automático de jardines. Se ha empezado a comercializar hace poco, alcanzando un gran éxito en el mercado debido a que presenta una completa, fácil y dinámica solución para el riego e iluminación de nuestros jardines.

Como se puede ver en la imagen de abajo, el usuario tendrá el control del GreenIQ Smart Garden Hub a través de una aplicación móvil o web, la cual permitirá la programación de este sistema, así como el visualizado de la información proveniente de los distintos sensores a los que puede ser conectado. Para permitir esta comunicación, usuario - sistema GreenIQ Smart Garden Hub, éste último deberá contar con acceso a internet vía Wifi o, en modelos especiales y para un mayor despliegue, podrá ser conectado a redes 3G. (14)

El GreenIQ tendrá el control de las distintas llaves de paso de agua con las que cuenta nuestro jardín y, así mismo, de la iluminación de éste. Se activarán y desactivarán estos dispositivos en función de las órdenes del usuario a través de la aplicación y también gracias a la información reunida por los sensores. (15)

Es un sistema muy completo al que pueden conectarse una multitud de dispositivos más para incluir nuevas funcionalidades. También está optimizado para gestionar eficientemente los recursos de nuestro jardín, permitiendo que, por ejemplo, en días lluviosos, nuestro sistema de riego no se dispare.

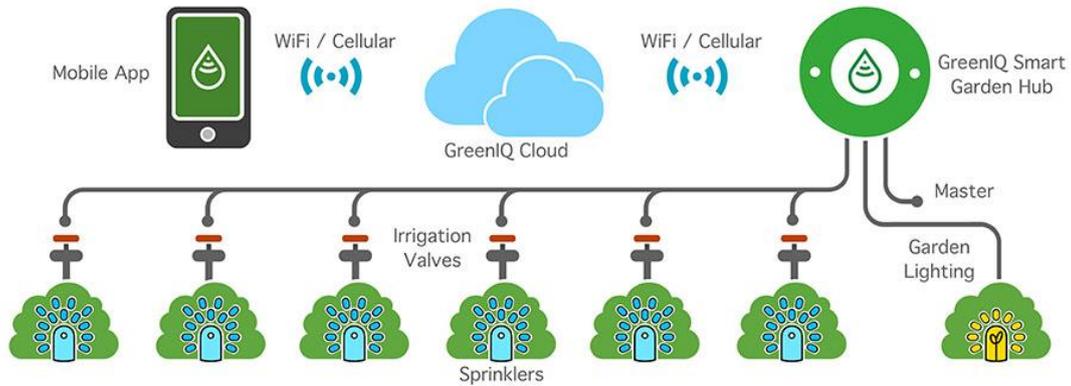


Ilustración 12: GreenIQ Smart Garden Hub. Fuente: http://greeniq.co/product_tech.htm

2.5 Conclusiones finales

Una vez analizado el mercado de actuación de este proyecto y fijadas las necesidades del usuario, se ha propuesto desarrollar un proyecto que intercomunicará, a través de redes 2G, una aplicación móvil en Android con el controlador del regadío en la parcela (placa empotrada Arduino UNO). Éste último activará y desactivará el riego siguiendo el criterio del usuario a través de la aplicación Android.

Capítulo 3

Análisis del sistema

En este capítulo se pone por escrito la naturaleza concreta del sistema de riego, describiendo su funcionalidad y las características específicas del proyecto con cada uno de los requisitos, tanto funcionales como no funcionales, los cuales se exponen a continuación.

3.1 Especificación de requisitos

A continuación, se exponen los requisitos de este proyecto siguiendo las siguientes normas de estilo y redacción: sin ambigüedad, completitud, verificación, coherencia, trazabilidad y usabilidad. El objetivo es crear unos requisitos con las características anteriores y construir un proyecto con una funcionalidad sólida.

Cada requisito, tanto de usuario como de sistema, vendrá claramente detallado en una tabla que tendrá el siguiente formato:

Identificador	R[X] – [YY]		
Prioridad	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción			

Tabla 1: descripción especificación de requisitos

Cada campo de la tabla anterior indica:

- Identificador: Identificará unívocamente a cada requisito, R[X] – [YY], donde:
 - X. Tomará el valor C para requisitos de capacidad, R para requisitos de restricción y, por último, SF y SNF para requisitos de sistema funcionales y no funcionales respectivamente.

- YY. Especifica el número de requisito para una correcta identificación.
- Prioridad: Determinará la prioridad para el cumplimiento del requisito dado, donde se tendrán tres niveles: alta, media o baja.
- Necesidad: Indicará la necesidad con la que ese requisito debe cumplirse, contando con una necesidad esencial, deseable u opcional.
- Estabilidad: Identifica la probabilidad de cambio de un requisito durante el desarrollo del proyecto, se cuenta con tres niveles de estabilidad: alta, media o baja.
- Descripción: Contendrá una breve descripción del requisito.

3.2 Requisitos de Usuario

3.2.1 Requisitos de Capacidad

A continuación se detallarán todos aquellos requisitos funcionales o de capacidad que indicarán las funcionalidades y necesidades del sistema a desarrollar.

Identificador		RC – 01		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional	
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Descripción	El sistema contará con una aplicación móvil con la que el usuario gestionará el riego de sus parcelas desde cualquier lugar con cobertura 2G.			

Tabla 2 : Requisito de capacidad RC – 01

Identificador		RC – 02		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional	
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Descripción	En la aplicación móvil, el usuario podrá agregar una parcela con un determinado nombre, bomba y número de sectores de regadío.			

Tabla 3: requisito de capacidad RC – 02

Identificador RC – 03			
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	En la aplicación móvil, el usuario podrá asociar un nombre y un número de móvil único a cada sector de una parcela.		

Tabla 4: requisito de capacidad RC – 03

Identificador RC – 04			
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	En la aplicación móvil, el usuario podrá visualizar todas las parcelas que haya creado.		

Tabla 5: requisito de capacidad RC – 04

Identificador RC – 05			
Prioridad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial	<input checked="" type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	En la aplicación móvil, el usuario podrá borrar una determinada parcela junto con sus sectores asociados.		

Tabla 6: requisito de capacidad RC – 05

Identificador RC – 06			
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	En la aplicación móvil, el usuario podrá visualizar todos los sectores de regadío asociados a una parcela.		

Tabla 7: requisito de capacidad RC – 06

Identificador		RC – 07		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional	
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Descripción	En la aplicación móvil, el usuario podrá visualizar el nombre y el estado (activado/desactivado) de un determinado sector de regadío.			

Tabla 8: requisito de capacidad RC – 07

Identificador		RC – 08		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional	
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Descripción	Desde la aplicación móvil, el usuario podrá activar el riego de un determinado sector desde cualquier lugar con cobertura 2G.			

Tabla 9: requisito de capacidad RC – 08

Identificador		RC –09		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional	
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Descripción	Desde la aplicación móvil, el usuario podrá desactivar el riego de un determinado sector desde cualquier lugar con cobertura 2G.			

Tabla 10: requisito de capacidad RC – 09

Identificador		RC – 10		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional	
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Descripción	La disponibilidad del sistema será 24/7.			

Tabla 11: requisito de capacidad RC – 10

3.2.2 Requisitos de Restricción

En las tablas siguientes se especifican los requisitos no funcionales, siendo aquellos que imponen las restricciones del sistema de riego y especifican su funcionamiento y no su funcionalidad.

Identificador RR - 01			
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	La plataforma empotrada sólo procesará SMS provenientes del número asociado en la instalación.		

Tabla 12: requisito de restricción RC - 01

Identificador RR - 02			
Prioridad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial	<input checked="" type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	Los nombres de parcela deberán ser únicos.		

Tabla 13: requisito de restricción RC - 02

Identificador RR - 03			
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	Los números asociados a cada parcela deberán ser únicos.		

Tabla 14: requisito de restricción RC - 03

Identificador RR - 04			
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El dispositivo móvil deberá tener conexión 2G para conectarse con la plataforma empotrada.		

Tabla 15: requisito de restricción RC - 04

Identificador RR - 05			
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	La aplicación tiene que tener permisos para el envío y lectura de SMS a la aplicación móvil.		

Tabla 16: requisito de restricción RC - 05

Identificador RR - 06			
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema empotrado deberá tener conexión 2G para poder comunicarse con la aplicación móvil.		

Tabla 17: requisito de restricción RC - 06

Identificador RR - 07			
Prioridad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial	<input checked="" type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	La bomba de regadío será simulada en esta implementación.		

Tabla 18: requisito de restricción RC - 07

Identificador RR - 08			
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	No se podrán crear más de 10 sectores.		

Tabla 19: requisito de restricción RC - 08

3.3 Casos de Uso

Los casos de uso que se describirán a continuación son los representados en la siguiente imagen:

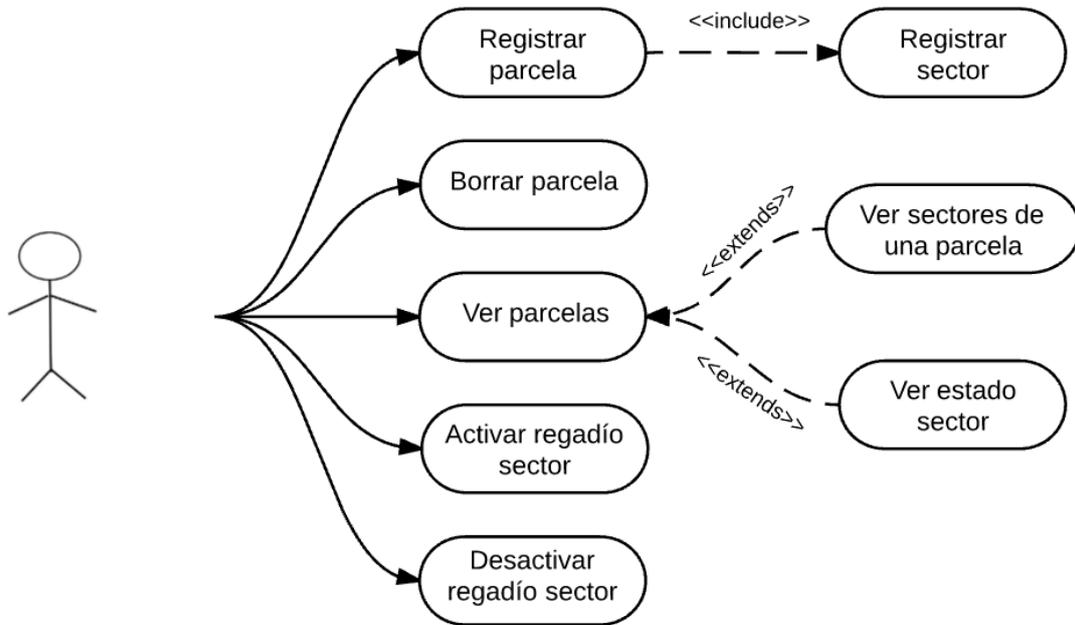


Ilustración 13: casos de uso

Las tablas usadas para describir los casos de uso siguen la siguiente plantilla:

Identificador	
Actor	
Objetivo	
Pre-condición	
Escenario	
Post.-condiciones	

Tabla 20: descripción especificación casos de uso

Cada campo de la tabla anterior indica:

- **Identificador:** Identificará unívocamente a cada caso de uso.
- **Actor:** Agente externo encargado de interactuar con el sistema.
- **Objetivo:** Acción a llevar a cabo después de la interacción del agente con el sistema.

- Pre-condiciones: Condiciones iniciales que deben darse para realizar la acción.
- Escenario: Pasos a realizar por el actor para cumplir con el objetivo.
- Post-condiciones: Condiciones finales que serán cumplidas si la acción del objetivo se realizar correctamente

Identificador		CU - 01
Actor	Usuario	
Objetivo	Registrar una parcela junto a su bomba y sectores asociados.	
Pre-condición		
Escenario	<ul style="list-style-type: none"> ▪ Acceder a la aplicación ▪ Pulsar la opción “crear nueva parcela” ▪ Introducir nombre parcela y número de sectores asociados. ▪ Introducir número único para bomba de regadío. ▪ Introducir nombres para cada sector y número únicos para cada uno de ellos. 	
Post.-condiciones	La parcela quedará guardada en la base de datos del dispositivo del usuario.	

Tabla 21: caso de uso CU – 01

Identificador		CU - 02
Actor	Usuario	
Objetivo	Borrar una parcela junto a su bomba y sectores asociados.	
Pre-condición	La parcela a borrar debe haber sido creada previamente.	
Escenario	<ul style="list-style-type: none"> ▪ Acceder a la aplicación. ▪ Pulsar la opción “gestionar parcelas”. ▪ Hacer una pulsación larga sobre la parcela a borrar. ▪ Elegir la opción borrar del menú emergente. 	
Post.-condiciones	La parcela se eliminará y no aparecerá listada.	

Tabla 22: caso de uso CU – 02

Identificador	CU - 03
Actor	Usuario
Objetivo	Ver listado de todas las parcelas registradas.
Pre-condición	Creación de una o varias parcelas anteriormente.
Escenario	<ul style="list-style-type: none"> ▪ Acceder a la aplicación. ▪ Pulsar la opción “gestionar parcelas”.
Post.-condiciones	

Tabla 23: caso de uso CU – 03

Identificador	CU - 04
Actor	Usuario
Objetivo	Visualizar información de una parcela.
Pre-condición	La parcela a visualizar debe haber sido creada previamente.
Escenario	<ul style="list-style-type: none"> ▪ Acceder a la aplicación. ▪ Pulsar la opción “gestionar parcelas”. ▪ Pulsar la parcela que se desea visualizar.
Post.-condiciones	

Tabla 24: caso de uso CU – 04

Identificador	CU - 05
Actor	Usuario
Objetivo	Ver listado sectores de una parcela.
Pre-condición	La parcela asociada a los sectores a visualizar debe haber sido creada previamente.
Escenario	<ul style="list-style-type: none"> ▪ Acceder a la aplicación. ▪ Pulsar la opción “gestionar parcelas”. ▪ Pulsar la parcela de la cual quieren visualizarse sus sectores.
Post.-condiciones	

Tabla 25: caso de uso CU – 05

Identificador	CU - 06
Actor	Usuario
Objetivo	Visualizar detalles de un sector de una parcela.
Pre-condición	La parcela asociada al sector a visualizar debe haber sido creada previamente.
Escenario	<ul style="list-style-type: none"> ▪ Acceder a la aplicación. ▪ Pulsar la opción “gestionar parcelas”. ▪ Pulsar la parcela de la cual quieren visualizarse sus sectores. ▪ Pulsar el sector que quiere visualizarse en más detalle.
Post.-condiciones	

Tabla 26: caso de uso CU – 06

Identificador	CU - 07
Actor	Usuario
Objetivo	Activar regadío en un sector.
Pre-condición	<ul style="list-style-type: none"> ▪ Parcela con el sector deseado a activar debe haber sido creada previamente. ▪ Tener cobertura 2G.
Escenario	<ul style="list-style-type: none"> ▪ Acceder a la aplicación. ▪ Pulsar la opción “gestionar parcelas”. ▪ Pulsar la parcela de la cual quieren visualizarse sus sectores. ▪ Pulsar el sector que quiere visualizarse en más detalle. ▪ Pulsar el botón “activar regadío”.
Post.-condiciones	El regadío del sector seleccionado debe activarse.

Tabla 27: caso de uso CU – 07

Identificador		CU - 08
Actor	Usuario	
Objetivo	Desactivar regadío en un sector.	
Pre-condición	<ul style="list-style-type: none"> ▪ Parcela con el sector deseado a desactivar debe haber sido creada previamente. ▪ Tener cobertura 2G. 	
Escenario	<ul style="list-style-type: none"> ▪ Acceder a la aplicación. ▪ Pulsar la opción “gestionar parcelas”. ▪ Pulsar la parcela de la cual quieren visualizarse sus sectores. ▪ Pulsar el sector que quiere visualizarse en más detalle. ▪ Pulsar el botón “desactivar regadío”. 	
Post.-condiciones	El regadío del sector seleccionado debe desactivarse.	

Tabla 28: caso de uso CU – 08

3.4 Requisitos de Sistema

3.4.1 Funcionales

Identificador		RSF – 01		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional	
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Descripción	<p>La aplicación contará con un menú principal con las siguientes opciones:</p> <ul style="list-style-type: none"> ▪ Agregar una nueva parcela. ▪ Administrar parcelas guardadas. 			

Tabla 29: requisito de sistema funcional RSF – 01

Identificador		RSF – 02		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional	
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Descripción	Para crear una nueva parcela se le pedirá al usuario que ingrese la siguiente información: <ul style="list-style-type: none"> ▪ Nombre descriptivo de la parcela. ▪ Número de teléfono asociado. ▪ Número de sectores en los que se divide la parcela. ▪ Nombres descriptivos para todos los sectores. 			

Tabla 30: requisito de sistema funcional RSF – 02

Identificador		RF – 03		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional	
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Descripción	Para guardar una parcela es necesario que el usuario haya introducido todos los datos.			

Tabla 31: requisito de sistema funcional RSF – 03

Identificador		RSF – 04		
Prioridad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja	
Necesidad	<input type="checkbox"/> Esencial	<input checked="" type="checkbox"/> Deseable	<input type="checkbox"/> Opcional	
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Descripción	Cuando la parcela se guarde correctamente se le notificará al usuario.			

Tabla 32: requisito de sistema funcional RSF – 04

Identificador		RSF – 05		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional	
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Descripción	Cuando el usuario haya ingresado algún valor repetido al agregar una nueva parcela, la aplicación le notificará que debe cambiarlo.			

Tabla 33: requisito de sistema funcional RSF – 05

Identificador		RSF – 06		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional	
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Descripción	<p>En el listado de todas las parcelas guardadas hasta el momento el usuario podrá realizar dos acciones:</p> <ul style="list-style-type: none"> ▪ Visualizar una parcela. ▪ Eliminar una parcela. 			

Tabla 34: requisito de sistema funcional RSF – 06

Identificador		RSF – 07		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional	
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Descripción	<p>Todos los sectores asociados a una parcela se mostrarán en una lista.</p>			

Tabla 35: requisito de sistema funcional RSF – 07

Identificador		RSF – 08		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional	
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Descripción	<p>El usuario podrá visualizar en detalle un sector de una parcela.</p>			

Tabla 36: requisito de sistema funcional RSF – 08

Identificador		RSF – 09		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional	
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Descripción	<p>Para activar un sector, la aplicación móvil enviará un SMS de activación cumpliendo un estándar definido.</p>			

Tabla 37: requisito de sistema funcional RSF – 09

Identificador		RSF – 10		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional	
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Descripción	Para desactivar un sector, la aplicación móvil enviará un SMS de desactivación cumpliendo un estándar definido.			

Tabla 38: requisito de sistema funcional RSF – 10

Identificador		RSF – 11		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional	
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Descripción	El sistema empotrado deberá reconectarse si pierde la cobertura.			

Tabla 39: requisito de sistema funcional RSF – 11

3.4.2 No funcionales

Identificador		RSNF – 01		
Prioridad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja	
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional	
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Descripción	En el sistema empotrado se restringirán las llamadas.			

Tabla 40: requisito de sistema no funcional RSNF – 01

Identificador		RSNF – 02		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional	
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Descripción	El usuario identificará las parcelas guardadas mediante su nombre.			

Tabla 41: requisito de sistema no funcional RSNF – 02

Identificador	RSNF – 03		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema identificará las parcelas guardadas mediante su número de teléfono asociado.		

Tabla 42: requisito de sistema no funcional RSNF – 03

Identificador	RSNF – 04		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	La aplicación móvil notificará al usuario si no dispone de conexión 2G para mandar SMS.		

Tabla 43: requisito de sistema no funcional RSNF – 04

Identificador	RSNF – 05		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	La aplicación móvil pedirá al usuario que permita el envío y lectura de SMS.		

Tabla 44: requisito de sistema no funcional RSNF – 05

Identificador	RSNF – 06		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema empujado se reiniciará si no tiene cobertura 2G.		

Tabla 45: requisito de sistema no funcional RSNF – 06

Identificador		RSNF – 07		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional	
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Descripción	Para simular la bomba de regadío se contará con un LED que se encenderá/apagará para simular la activación/desactivación de la bomba respectivamente.			

Tabla 46: requisito de sistema no funcional RSNF – 07

Identificador		RSNF – 08		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional	
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja	
Descripción	La aplicación móvil notificará al sistema cuando el usuario intente registrar más de 10 sectores.			

Tabla 47: requisito de sistema no funcional RSNF – 08

3.5 Matriz de trazabilidad

3.5.1 Matriz de trazabilidad de requisitos de capacidad y requisitos de sistema funcionales

Habiendo recogido y puesto en escrito todos los requisitos del proyecto, es necesario ahora comprobar que cada requisito de usuario está contemplado en uno o varios de sistema.

Para realizar dicha comprobación se crea una matriz de trazabilidad, donde cada requisito de usuario está identificado en una fila de la matriz, así como cada requisito de sistema en una columna.

Se pretende relacionar los requisitos de usuario con los de sistema y observar, así, si el proyecto se llevará o no a buen término. Para ello cada fila de la matriz deberá tener una o varias X, lo que significa que el requisito de usuario identificado en esa fila está contemplado en los requisitos de sistema.

	RC-01	RC-02	RC-03	RC-04	RC-05	RC-06	RC-07	RC-08	RC-09	RC-10
RSF-01										
RSF-02										
RSF-03										
RSF-04										
RSF-05										
RSF-06										
RSF-07										
RSF-08										
RSF-09										
RSF-10										
RSF-11										

Tabla 48: matriz de trazabilidad de requisitos de capacidad y de sistema funcionales

Como podemos observar, todos los requisitos de usuario están contemplados en los de sistema.

1.1.1 Matriz de trazabilidad de requisitos de restricción y requisitos de sistema no funcionales

Se dispone de la matriz que relaciona los requisitos de restricción con los de sistema no funcionales.

	RR-01	RR-02	RR-03	RR-04	RR-05	RR-06	RR-07	RR-08
RSNF-01								
RSNF-02								
RSNF-03								
RSNF-04								
RSNF-05								
RSNF-06								
RSNF-07								
RSNF-08								

Tabla 49: matriz de trazabilidad de requisitos de restricción y de sistema no funcionales

1.1.2 Matriz de trazabilidad de requisitos de sistema funcionales y casos de uso

A continuación se expone la matriz de trazabilidad de los requisitos de sistema frente a los casos de uso.

	RSF-01	RSF-02	RSF-03	RSF-04	RSF-05	RSF-06	RSF-07	RSF-08	RSF-09	RSF-10	RSF-11
CU-01											
CU-02											
CU-03											
CU-04											
CU-05											
CU-06											
CU-07											
CU-08											

Tabla 50: descripción matriz de trazabilidad de requisitos de sistema funcionales y casos de uso

Capítulo 4

Diseño del sistema

En este apartado se estudiarán el diseño y arquitectura de todos los elementos que conforman el sistema de control de riego en el que se basa este proyecto. Se explicará cómo se relacionan todas y cada una de estas partes entre sí.

Este proyecto es un sistema de control de riego con dos partes bien diferenciadas, se detallan a continuación:

- Una de ellas es aquel al que llamaremos módulo Android en el que estarán presentes el usuario final y una aplicación móvil. El usuario sólo interactuará con la aplicación móvil, la cual se le facilitará para el control del riego de sus parcelas.
- La segunda es el módulo de la parcela o módulo hardware. Esta parte se implantará en aquellos lugares de cultivo donde se quiera tener el control de su sistema de riego. Lo conforman una placa empotrada, un módulo GSM, la bomba de riego y diferentes electroválvulas que controlarán el paso de agua.

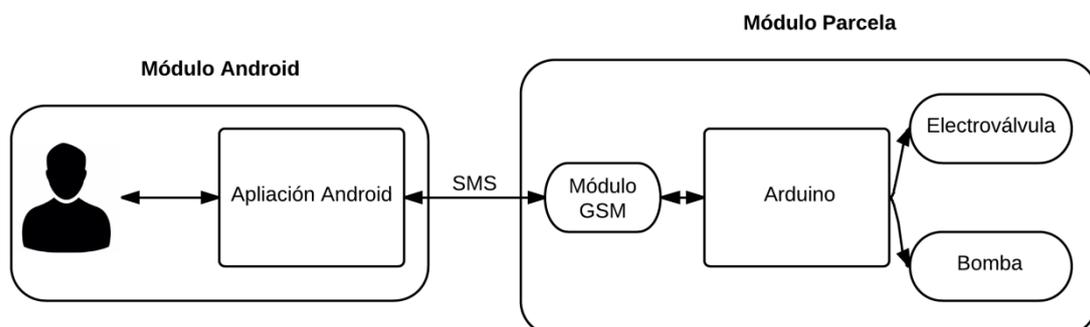


Ilustración 14: módulos del sistema

Estas dos partes deberán estar conectadas de algún modo para que el usuario pueda establecer una comunicación a través de la aplicación móvil con la parcela que cuente con el sistema de control remoto de riego. Esta conexión se establecerá gracias a la red de comunicaciones móviles y al envío/recibo de SMS.

Dichos SMS contendrán información clave, siguiendo un convenio (el cual se explicará más adelante). Su envío permitirá activar/desactivar el riego y mantener informado al usuario del estado del sistema en todo momento.

El dispositivo móvil del usuario donde está instalada la aplicación del control del riego dispondrá de una tarjeta SIM operativa y a su vez, también tendrán uno los módulos GSM conectados a las placas empotradas.

El sistema ha sido diseñado de acuerdo a identificar las distintas parcelas mediante los números de teléfono asociados a las tarjetas SIM de cada módulo GSM. De esta manera, cada parcela tendrá un número de teléfono único e identificativo.

La arquitectura que siguen las dos partes principales del sistema sigue el modelo cliente-servidor.

En este modelo hay dos partes diferenciadas. Por un lado se encuentran los clientes que son programas que usan servicios proporcionados por otros programas. Por otra parte, aquellos programas que proporcionan estos servicios son los llamados servidores. Normalmente el usuario interactúa con los programas cliente, estos programas realizan entonces una petición de un servicio y, finalmente, es el servidor el que lleva a cabo este servicio. Los servidores responden únicamente a peticiones de clientes: gestión de recursos, peticiones de datos, realización de cálculos, informaciones de estado... (16)

Antes de decidir quien posee el papel de cliente y quien el de servidor será necesario diferenciar antes dos escenarios posibles:

- La plataforma empotrada se inicia por primera vez o se reinicia.
- Después de que la plataforma empotrada realice la configuración de inicio/reinicio.

La plataforma empotrada se inicia/reinicia.

El siguiente escenario se daría una vez que el Arduino se inicie por primera vez o tenga un reinicio fortuito o intencionado por parte del usuario/profesional de instalación.

Ya sea la primera vez que se pone en funcionamiento o si se ha producido un reinicio, la plataforma empotrada necesita saber los estados de los componentes a los que está conectado. Estos componentes son la bomba de riego y las distintas electroválvulas. Se seguirá el siguiente proceso de comunicación:

- i. Primero el Arduino enviará un mensaje al número de teléfono determinado en la instalación. Será el número de la tarjeta SIM del dispositivo desde el que se vaya a usar la aplicación Android.
- ii. La aplicación recibe un mensaje de configuración e identifica la parcela que se lo ha enviado, reúne todos los estados de los sectores de ésta y los envía de vuelta a la parcela.
- iii. La plataforma empotrada de la parcela recibe el mensaje de vuelta con la configuración que debe realizar y, de acuerdo a ésta, configura los estados de los elementos a los que está conectado.

En este caso, y sabiendo que el cliente es aquel que inicial la comunicación y el servidor quien proporciona un servicio, el Arduino hará el papel de cliente y la aplicación móvil el de servidor.

Después de realizar la configuración de inicio/reinicio.

Este escenario entraría en vigor después de que la plataforma hardware realice las configuraciones necesarias después del primer inicio o de un reinicio.

A continuación se explica de forma sencilla las comunicaciones que se dan en este escenario:

- i. El usuario empieza la comunicación con el sistema por medio de la aplicación móvil.
- ii. Posteriormente ésta, y de acuerdo a las acciones que quiere realizar el usuario, se comunicará con el módulo hardware en la parcela por medio del envío de distintos SMS.
- iii. Los SMS son recibidos por la plataforma hardware, donde serán procesados para determinar la acción a realizar.

- iv. Una vez realizada la acción, el Arduino envía un SMS de confirmación a la aplicación móvil.
- v. La aplicación móvil recibe el SMS de confirmación y actualiza el estado correspondiente en la aplicación.

Expuesto lo anterior, la aplicación móvil será la encargada de iniciar la comunicación con el sistema empotrado y esperará a que éste realice la operación solicitada y devuelva una respuesta. Es por esto que, en este caso, el papel de cliente lo toma la aplicación en Android y el de servidor, la plataforma Arduino.

Finalmente, todo el sistema queda representado claramente en la ilustración 12. En un lado quedará la aplicación móvil (izquierda) que se comunicará mediante SMS con la parcela (derecha).

La plataforma empotrada quedará conectada al módulo GSM, la bomba de regadío y las electroválvulas que controlan el paso de agua de cada sector.

El encargado de recibir el SMS en la parcela será el módulo GSM y éste se lo enviará al Arduino por el puerto serie para que pueda interpretarlo y procesarlo. Una vez el Arduino interprete el SMS realizará las acciones pertinentes sobre los elementos a los que está conectado (bomba y electroválvulas).

Red de comunicaciones móviles

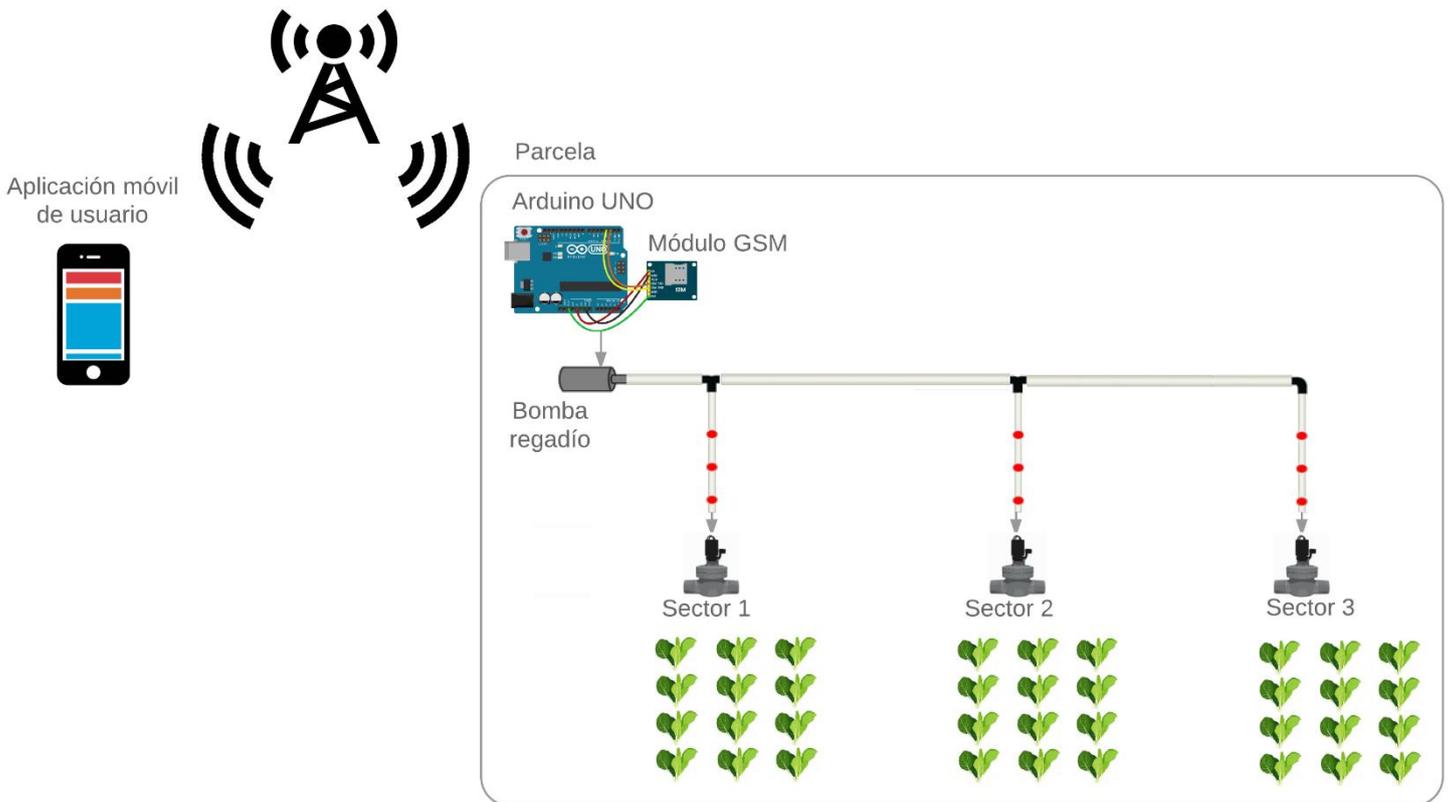


Ilustración 15: diseño sistema

A continuación se proporciona un simple ejemplo de funcionamiento donde se activará y desactivará el riego en una parcela.

- Se registra una parcela con tres sectores en la aplicación móvil.
- Se activa el regadío en el sector 1 (actualmente no hay ningún sector activo):
 - Se activa el funcionamiento de la bomba de regadío.
 - Se abre la electroválvula del sector 1.
- Se activa el regadío en el sector 2:
 - Se abre la electroválvula del sector 2.
- Se desactiva el regadío del sector 1:
 - Se cierra la electroválvula del sector 1.
- Se desactiva el regadío en el sector 2 (se desactiva el último sector activo):
 - Se cierra la electroválvula del sector 2.
 - Se desactiva el funcionamiento de la bomba de regadío.

El circuito de agua de cada sector en una parcela estará controlado por una electroválvula que permitirá o no el paso de agua por el. Fue elegida una electroválvula debido a que es un componente que puede modificar su estado de forma electrónica.

Una electroválvula puede cambiar de estado gracias a una bobina solenoide que posee en su interior. Sus posiciones son: abierto y cerrado. (17)

Para la modificación de su estado será necesario contar con un relé y con una fuente de corriente alternativa que alimentará a la electroválvula con 9V. Para mantener el estado “abierto” en la electroválvula será necesario dotarla de alimentación continua.

Su esquema de conexión es el siguiente:

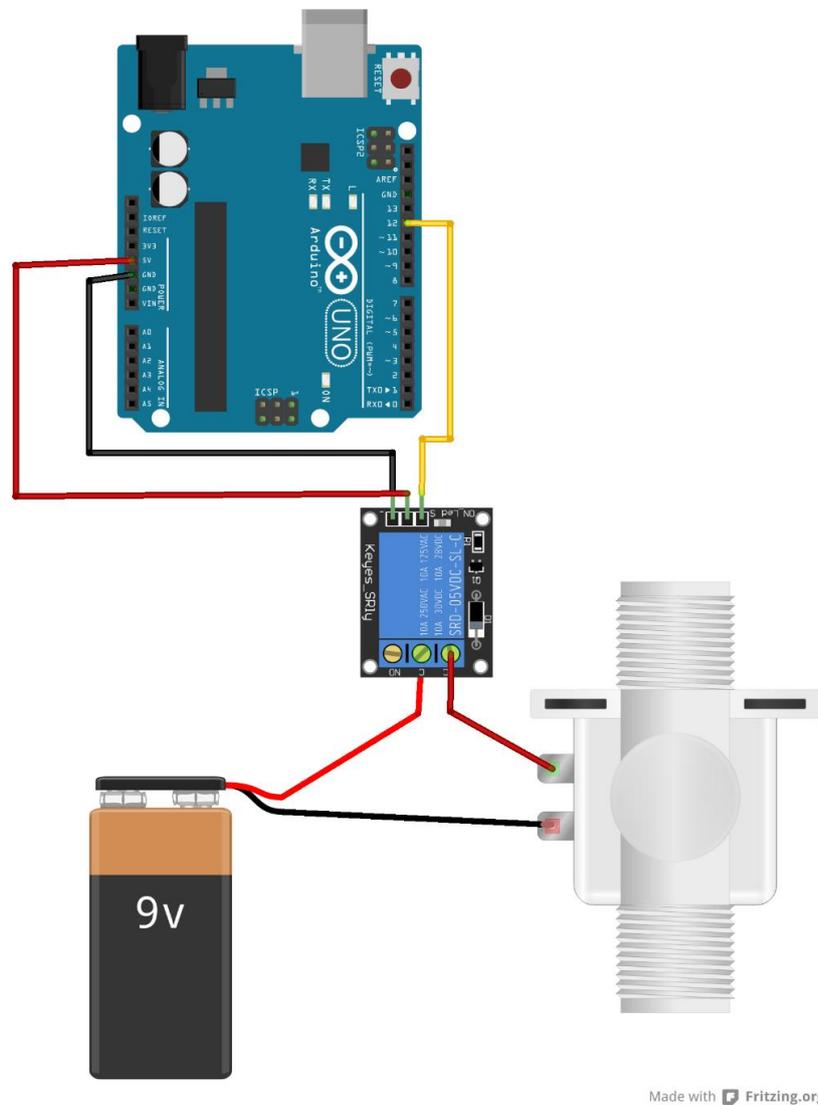


Ilustración 17: esquema conexión electroválvula

Finalmente, para que la aplicación móvil pueda comunicarse con la plataforma empotrada será necesario conectar a ésta un módulo GSM encargado del envío/recibo de SMS.

Después se deberá insertar una tarjeta SIM de tamaño normal y operativa, en la parte de atrás del módulo GSM para que éste pueda conectarse a un operador de red.

Para conectar el módulo GSM al Arduino se deberá disponer de una fuente de alimentación externa de 5V encargada de suministrar energía a la tarjeta SIM. Por otra parte, se deberán unir los pines 7 y 8 del módulo GSM con los pines del mismo número del Arduino. También, para alimentar a la tarjeta se deberán conectar los pines 9 de ambos elementos. (9)

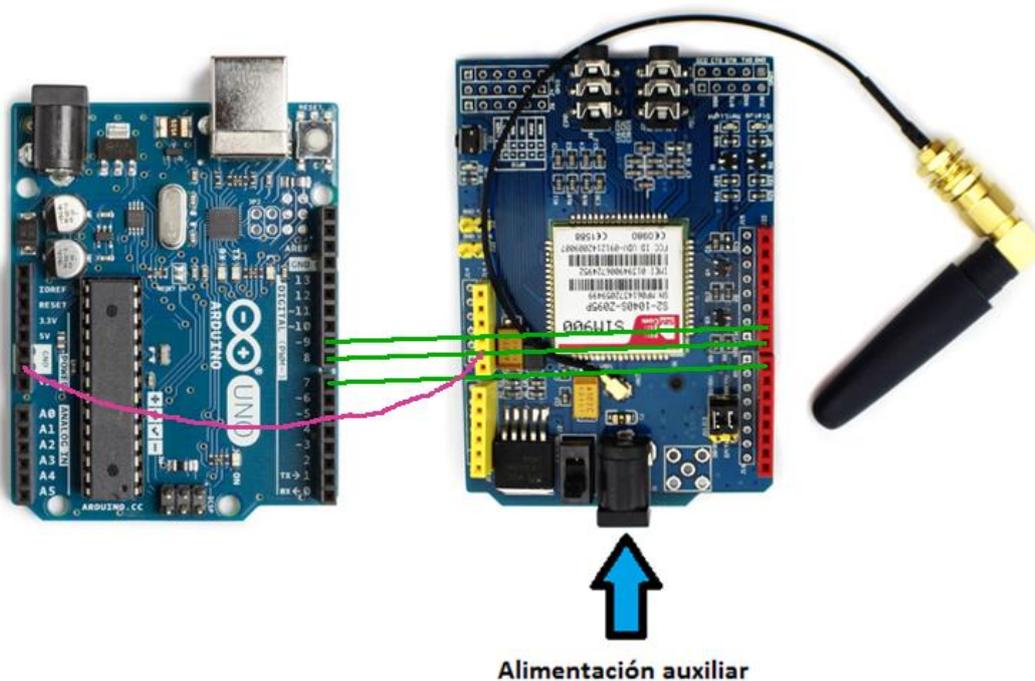


Ilustración 18: esquema de conexión módulo GSM

Finalmente, es necesario soldar el pad “R13” situado al lado de los pines rojos “J16” para poder activar la alimentación de la tarjeta SIM por programa, a través del pin 9.

2.2 Diseño e implementación del software

A continuación se explica en profundidad el diseño e implementación del software tanto el de la plataforma empotrada Arduino, como el correspondiente a la aplicación móvil realizada en Android.

2.2.1 Diseño de la plataforma empotrada

Para explicar con detenimiento este apartado hay que saber, en primer lugar, que la plataforma hardware Arduino no guarda sus estados cuando sufre un reinicio. Es por tanto necesario contar con un procedimiento que tras un reinicio, vuelva a dejar en el mismo estado los elementos conectados a la plataforma hardware (bomba de regadío y electroválvulas).

La única parte del sistema que cuenta con almacenamiento persistente es la aplicación móvil. Es por esto que, cuando el Arduino sufra un reinicio, preguntará a la aplicación cuál era su estado antes de este. La aplicación móvil le devolverá una respuesta con el estado y el Arduino así se recuperaría del paro sufrido.

Cabe destacar que no se puede diferenciar entre un reinicio y el primer inicio de la plataforma, así que, para ambos casos, se aplicará el procedimiento descrito anteriormente.

Después de la puesta en marcha, la plataforma hardware se quedará esperando a que lleguen nuevas órdenes a través de SMS, estas son:

- *Activar el regadío de un sector:* se deberá abrir el paso de agua a través de la electroválvula del sector indicado. Si además es el primer sector que se va a abrir, es necesario iniciar también la bomba de regadío.
- *Desactivar el regadío de un sector:* se deberá cerrar el paso de agua a través de la electroválvula del sector indicado. Si además es el último sector que se va a cerrar, es necesario también parar la bomba de regadío.

A continuación puede verse en un esquema todo lo descrito anteriormente:

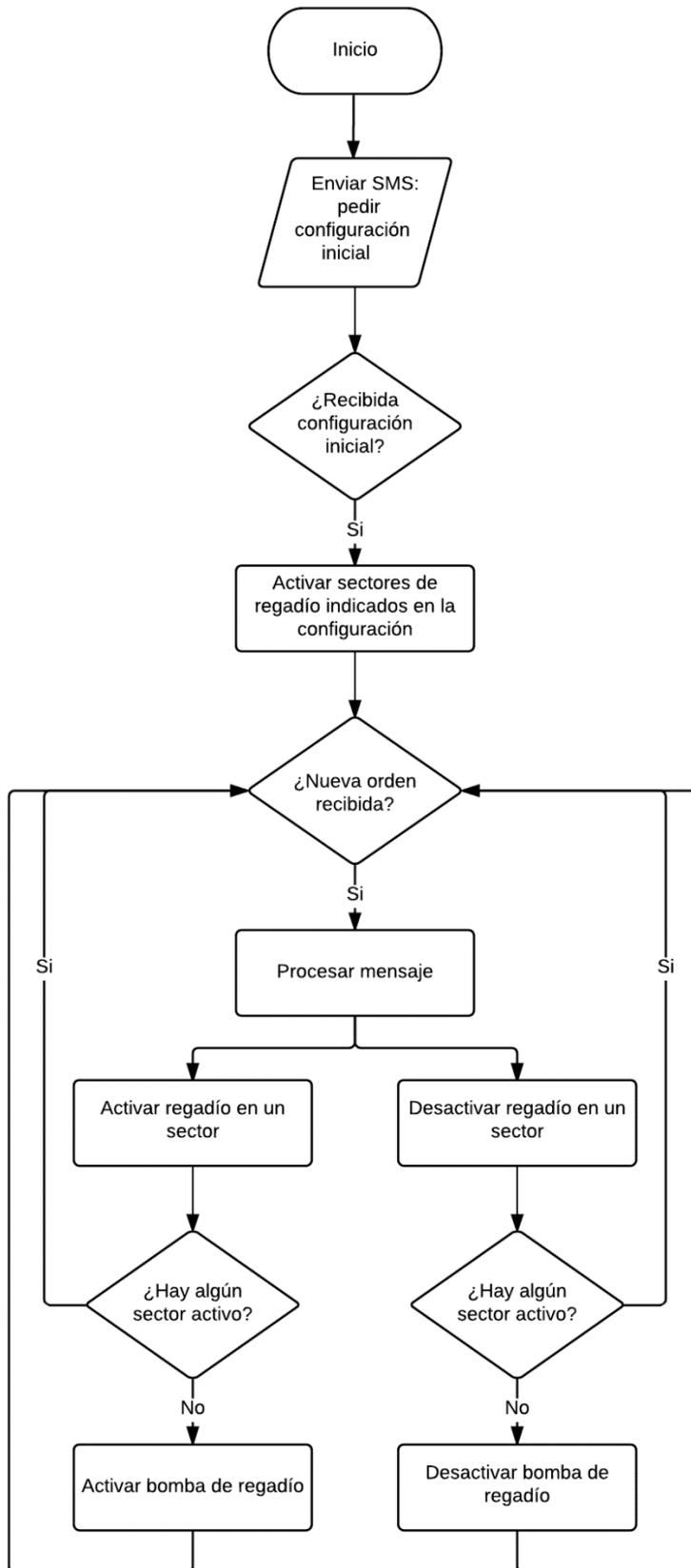


Ilustración 19: funcionamiento software plataforma empotrada

2.2.2 Implementación de la plataforma empotrada

Es acertado describir, en primer lugar, el contenido de los distintos SMS que se enviarán los diferentes componentes del sistema. Para ello se distinguen los dos escenarios posibles en los que puede encontrarse el sistema en un momento dado:

El Arduino se inicia/reinicia

En este escenario el Arduino envía un mensaje a la aplicación para saber en qué estados dejar la bomba de regadío y las electroválvulas correspondientes a cada sector.

Una vez que el SMS de configuración es recibido por la aplicación, ésta recopila los estados de todos los sectores y se los envía de vuelta al Arduino.

Finalmente, después de recibir la respuesta de configuración, el Arduino procesa el mensaje y establece el estado correspondiente a cada elemento.

Contenido SMS	Descripción
CONFIG	Arduino envía mensaje de configuración.
;CONFIG;estado_sector_1;...;estado_sector_n	Aplicación envía respuesta al mensaje de configuración.
OKCONFIG	Arduino responde cuando haya terminado de configurar los estados de los elementos a los que está conectado.

Tabla 51: protocolo sms de configuración

Donde *estado_sector_x* tomará el valor de *0* si en el sector *x* el riego está desactivado o *1* si está activado.

Veamos un ejemplo:

El Arduino de la parcela con número de teléfono 678 se reinicia. Cuenta con cuatro sectores, dos de ellos, el segundo y tercero, estaban activos justo cuando el Arduino se ha reiniciado.

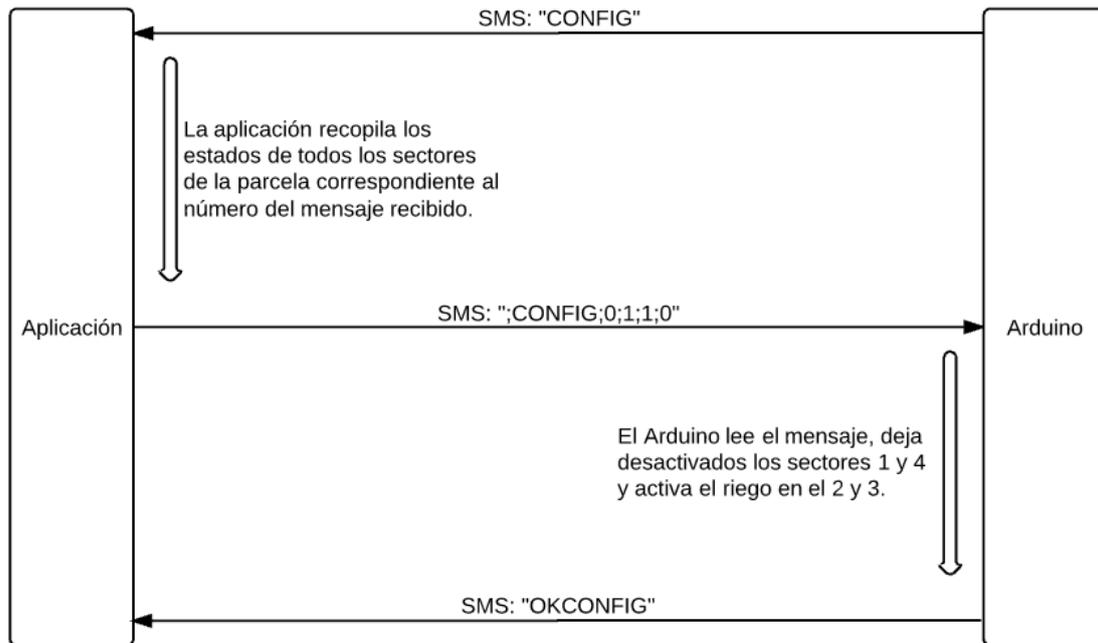


Ilustración 20: funcionamiento del sistema en un inicio/reinicio

Después de realizar la configuración de inicio/reinicio

En este escenario, el Arduino estará esperando órdenes de la aplicación móvil. Las órdenes posibles son la activación o desactivación de algún sector. El formato de los mensajes que contienen estas órdenes se exponen a continuación:

Contenido SMS	Descripción	Respuesta
;ONRIEGO;sector_x;...	Activar riego en sector x	OKONRIEGO
;OFFRIEGO;sector_x;...	Desactivar riego en sector x	OKOFFRIEGO

Tabla 52: protocolo sms de activación/desactivación

Donde *sector_x* tomará el número de sector donde se quiera activar/desactivar el riego.

A continuación se muestra un ejemplo:

Se pretende activar el riego de los sectores 2 y 3 de la parcela 678.

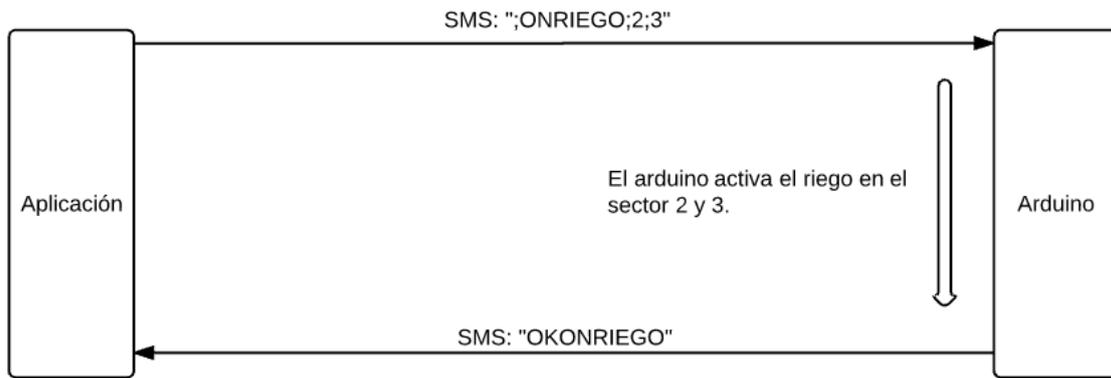


Ilustración 21: funcionamiento del sistema activación regadío

Se pretende desactivar el riego de los sectores 2 y 3 de la parcela 678.



Ilustración 22: funcionamiento del sistema desactivación regadío

A continuación se describirá la implementación final de todo lo descrito anteriormente.

Se ha querido crear un sistema independiente del número de sectores que puedan contener las parcelas en la que vaya a instalarse. Para llevar esto a cabo se ha realizado lo siguiente:

- Los pines digitales disponibles en la plataforma empotrada son los siguientes: 0, 1, 2, 3, 4, 5, 6, 10, 11, 12. Los demás quedan ocupados por la comunicación por el puerto serie y la activación de la alimentación de la tarjeta SIM.
- La bomba quedará conectada al pin 13 del Arduino.
- En la instalación de la plataforma hardware, se deberá acordar con el usuario cuál será el sector que se conecte a cada pin del Arduino.
- En la aplicación móvil, en el momento de asignación de nombres identificativos para el usuario a los sectores de una parcela, se le mostrará

por pantalla qué número de sector tiene cada uno y a qué pin del Arduino está conectado. El sector 1 estará conectado al pin 0 del Arduino, el sector 2 al pin 1 y así sucesivamente. De esta forma el usuario tendrá identificado de manera correcta cada sector.

NÚMERO SECTOR	1	2	3	4	5	6	7	8	9	10
PIN ARDUINO	0	1	2	3	4	5	6	10	11	12

Tabla 53: mapeo número de sector y pin

Con esta configuración, cuando llegue un SMS a la plataforma empotrada donde se especifique apagar el riego en el sector 1, el Arduino debe mapear este sector para saber cuál es su pin correspondiente y así ejecutar la orden de manera correcta.

A continuación se muestra la implementación del sketch grabado en el Arduino.

En primer lugar se ha creado un método llamado *enviarAT()* encargado de enviar un comando AT y esperar por su respuesta durante un tiempo determinado. Tanto el comando AT, como la respuesta y el tiempo de espera se especifican como parámetros.

Con esto nos aseguramos que el Arduino envía al módulo GSM de manera correcta el comando deseado y espera por la respuesta correcta durante el tiempo impuesto. De esta manera evitamos enviar varios comandos a la vez antes de recibir sus respectivas respuestas. Este método se usará en toda la implementación cada vez que el Arduino quiera enviar órdenes al módulo GSM.

Cuando el Arduino se inicia/reinicia ejecuta una vez toda la configuración contenida en el método *setup()*. Dentro de esta sección se configuran los pines que se usarán como salida, se establece la velocidad del puerto serie (19200 baudios) para comunicar el módulo GSM y el Arduino y, finalmente, se inicia dicho módulo.

Para la iniciación del módulo GSM es necesario enviar un pulso eléctrico desde el pin 9 del Arduino para activar la alimentación de la tarjeta SIM. Una vez realizado esto, se envía el comando “AT” cada dos segundos y, cuando se reciba por fin la respuesta correcta, el módulo GSM habrá sido iniciado correctamente.

Posteriormente, se enviará el comando “AT+CREG?” periódicamente para comprobar cuándo el módulo ha conseguido conectarse a la red móvil, es decir tener cobertura.

Como configuración final del módulo GSM, se deberá establecer el modo texto para poder enviar y recibir mensajes con el siguiente comando: “AT+CMGF=1\r”. Y con “AT+CNMI=2,2,0,0,0\r” el módulo nos mostrará el texto de los mensajes.

Finalmente, se deberá enviar a la aplicación un mensaje de configuración, esperar por su respuesta y configurar los estados de la bomba de regadío y electroválvulas. La forma en cómo se espera por el mensaje y se procesa es la misma que la que se explicará en párrafos posteriores.

Después de la configuración inicial, el Arduino ejecutará en bucle todo lo contenido en el método *loop()*. Dentro de éste se comprobará repetidamente si el módulo GSM transmite alguna información por el puerto serie. Si se detecta un SMS entrante, éste se empieza a guardar en una variable global llamada *mensaje* y posteriormente se llamará a la función *identifica_SMS()* la cual procesará dicho mensaje y realizará las acciones pertinentes.

```
void loop() {
  if (SIM900.available() > 0)
  {
    incoming_char = SIM900.read(); //Guardamos el carácter del GPRS
    Serial.print(incoming_char); //Mostramos el carácter en el monitor serie
    mensaje = mensaje + incoming_char ; // Añadimos el carácter leído al mensaje
  }
  identifica_SMS();
}
```

Ilustración 23: detección mensajes entrantes

En *identifica_SMS()* se comprobará si el SMS recibido sigue el estándar de comunicación acordado en el sistema, si lo sigue, éste se dividirá en varias partes: la acción a realizar y el sector sobre el cuál realizar esta acción.

Las acciones posibles son las siguientes:

- *Activación regadío*: Se comprueba si hay más sectores activos, si no hay ninguno, se procederá a activar primero la bomba de regadío y después el sector solicitado.

- *Desactivación regadío:* Se comprueba si el sector que se va a desactivar es el último, si es el último, se deberá apagar la bomba de regadío en primer lugar y después cerrar el paso de agua en el sector solicitado.

2.2.3 Diseño de la aplicación en Android

La aplicación en Android es creada para que el usuario pueda interactuar con el sistema de regadío de forma fácil e intuitiva. Esta aplicación se conectará con la plataforma Arduino por medio del envío y recibo de SMS con un determinado formato que ha sido explicado en los apartados anteriores.

Con esta aplicación el usuario será capaz de registrar todas las parcelas que cuenten con la instalación del módulo hardware, así como sus sectores asociados.

A su vez también se podrán revisar todos los datos anteriores incluyendo la visualización y modificación de los estados del riego en cada sector.

Con el esquema que se incluye a continuación se comprenderá en profundidad el funcionamiento de la aplicación:

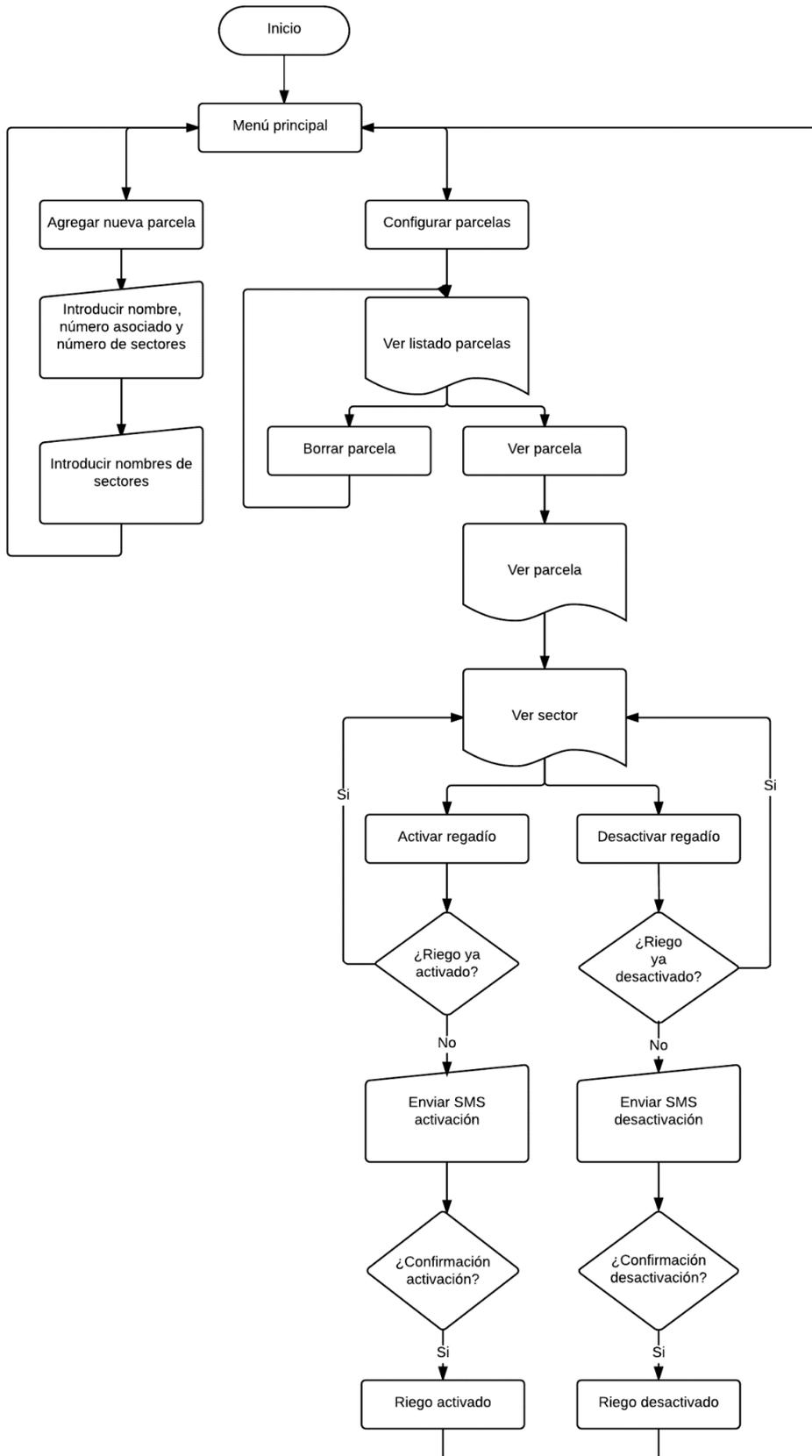


Ilustración 24: esquema de navegación aplicación móvil

A continuación se incluyen imágenes de cada una de las pantallas con las que cuenta la aplicación y se detallará las acciones que pueden realizarse en cada una de ellas, así como su funcionalidad.

Una vez el usuario abra la aplicación verá el menú principal de ésta (ilustración 21), consta de dos opciones:

- **Agregar nueva parcela:** el usuario deberá elegir esta opción para registrar en la aplicación una nueva parcela junto a sus sectores para que, después, éste pueda activar o desactivar el regadío en ella.
- **Administrar parcelas:** esta opción deberá ser elegida cuando el usuario quiera visualizar todas las parcelas registradas, así como sus sectores y donde podrá activar/desactivar el regadío en ellas.



Ilustración 25: menú inicial

Si el usuario selecciona la opción para agregar una nueva parcela, la pantalla que se mostrará será la que se muestra en la ilustración 22. Se deberán rellenar cada uno de los inputs: nombre de la parcela, número asociado y número de sectores en los que se dividirá. Ningún input puede quedar vacío para registrar una parcela correctamente.

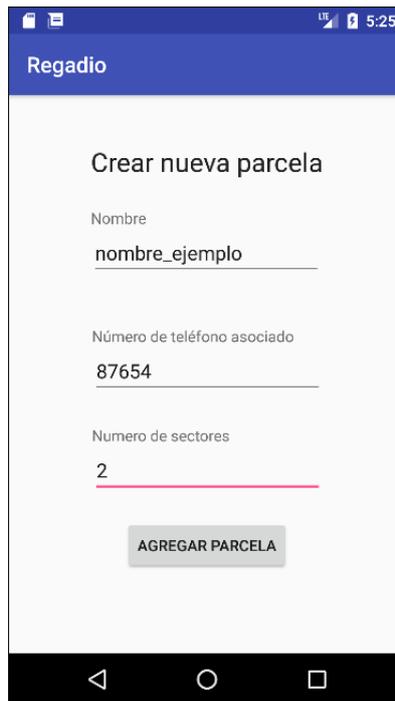


Ilustración 26: registrar parcela

Después de ingresar los datos primarios de una parcela, es necesario concretar los nombres que obtendrán los sectores en los que se divide ésta para para que el usuario pueda identificarlos de manera correcta.

Estos nombres no tienen ningún fin de identificación interno para la aplicación, simplemente son una herramienta útil que se le facilita al usuario para que pueda identificar cada sector con alguna palabra o frase con significado para él.

A su vez, se le dará información de a qué pin del Arduino está conectado cada una de las electroválvulas de los sectores.

Para ello el usuario tendrá que rellenar los inputs de cada sector (todos obligatorios) en una lista en la que podrá hacer scroll y, posteriormente, darle al botón de “guardar”.

Con esto, el registro de una parcela queda cubierto correctamente y al usuario se le devolverá al menú principal. Además se le mostrará un mensaje de éxito en la operación.

Si por el contrario, el usuario no rellena alguno de los inputs mostrados en esta pantalla y en la anterior, la aplicación le comunicará al usuario que es necesario completar todo y no le dejará guardar el proceso.



Ilustración 27: insertar nombres sectores

Si en el menú principal, el usuario elige la acción “gestionar parcelas”, accederá a la pantalla de la ilustración 24, la cual contendrá una lista de todas las parcelas registradas hasta ese momento.

En la lista aparecerá el nombre dado a la parcela, su número asignado, el número de sectores en los que se divide y una frase en gris “riego desactivado” si todos los sectores tienen el riego desactivado o “riego activado”, en color verde, si hay uno o varios sectores con el riego activado.

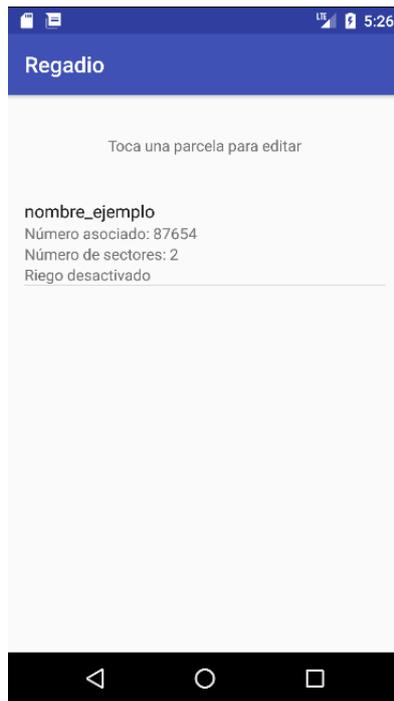


Ilustración 28: lista parcelas

Aún sin salirnos de la pantalla anterior, el usuario puede realizar una larga pulsación en la parcela que desee y se abrirá un desplegable con dos opciones (ilustración 25):

- *View*: si el usuario elige esta opción, se pasará a otra pantalla donde se visualizará en detalle completo la parcela seleccionada.
- *Delete*: con esta acción se borrará la parcela con todos sus sectores de la aplicación.

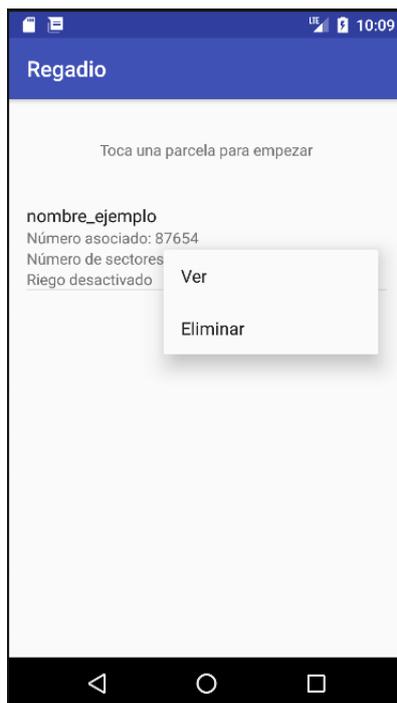


Ilustración 29: menú en lista parcelas

Ya sea pulsando la opción “ver” de la pantalla de la ilustración 25 o dando un toque corto a una parcela de la lista, se accederá a una nueva interfaz donde se podrá visualizar toda la información de la parcela seleccionada: nombre y número asociado y todos sus sectores junto con una etiqueta para visualizar el estado del riego:

- “*Riego desactivado*”: en color gris si el regadío esta desactivado en ese sector.
- “*Riego activado*”: en color verde si el regadío está activado en ese sector.



Ilustración 30: detalles de una parcela

Cualquiera de los sectores mostrados por pantalla en la interfaz de la ilustración 26 son seleccionables mediante un toque corto. De esta manera, una vez pulsado uno de esos sectores, el usuario pasará a otra pantalla donde podrá ver el estado de ese sector y dispondrá de dos botones para cambiar el estado de éste (ilustración 27):

- **ACTIVAR:** Si el usuario pulsa este botón y el regadío estaba previamente desactivado, la aplicación enviará un SMS al módulo GSM correspondiente para activar el regadío de ese sector. Si el regadío ya estaba activado, la pulsación de este botón no realizará ninguna acción.
- **DESACTIVAR:** tiene exactamente la misma lógica del anterior, excepto que este botón desactiva el regadío.

Tras pulsar estos botones, si la operación se realiza con éxito, el estado del sector se actualizará.



Ilustración 31: detalles de un sector

2.2.4 Implementación de la aplicación móvil en Android

Anteriormente se ha explicado de manera sencilla la lógica que seguirá el sistema, pero es necesario, ahora, profundizar más en dichas acciones e instrucciones para entender con total claridad cómo ha sido implementada la aplicación móvil.

Como bien se ha visto en el apartado anterior, la aplicación tiene una parte para el registro de parcelas y otra destinada para su visualización y gestión. En ésta última se podrá gestionar el riego de cada uno de los sectores de una parcela. Para ello se ha seguido la siguiente lógica:

- Un sector podrá tener tres estados: activado, desactivado y esperando confirmación. Será posible así que el usuario conozca en todo momento como se encuentra cada sector.
- Cuando el usuario desee activar (o desactivar) el regadío de un sector, se enviará un SMS al número asociado a la parcela de ese sector. En dicho SMS vendrá contenido el sector y la acción que se quiere realizar sobre él (activar o desactivar regadío).

Inmediatamente después del envío del SMS, la aplicación pondrá el sector en el estado “esperando confirmación”.

- La plataforma empotrada de la parcela elegida recibirá un SMS, el cual procesará para determinar si sigue el formato definido en el sistema. Si sigue este formato, se identificará la acción a realizar y el sector sobre el cual realizar ésta y se hará efectiva dicha orden. Finalmente, la plataforma hardware enviará un SMS indicando éxito en la operación.
- De nuevo entra en escena la aplicación móvil. Ésta quedará esperando de manera asíncrona la respuesta de éxito del Arduino, una vez que la reciba, cambiará el estado del sector (activado o desactivado).

Gracias al estado “esperando confirmación” se evitará que el usuario vuelva a enviar más SMS para realizar una misma acción. Además así, se consigue optimizar más los recursos.

A continuación se describirá la forma en la que la aplicación móvil guarda y recupera información sobre las parcelas, sectores y estados.

Base de datos

En la creación de la aplicación ha sido necesario disponer de una base de datos en SQLite para manejar y almacenar la información. Así, cuando el usuario cierre la aplicación, los datos y sus estados quedarán almacenados de forma persistente en su dispositivo móvil.

La base de datos tiene la siguiente estructura:

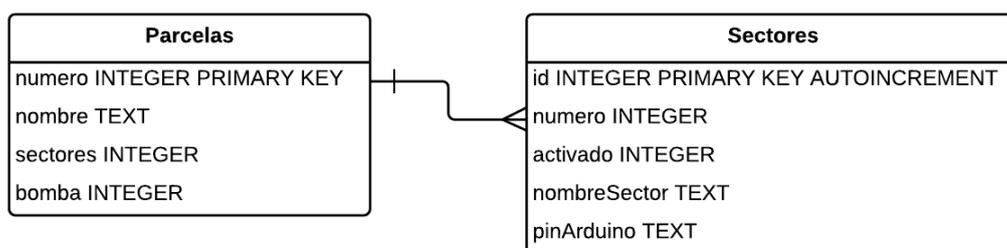


Ilustración 32: modelo base de datos

La base de datos consta de dos tablas con varios campos cada una. La primera tabla *Parcelas* guardará la información de todas las parcelas registradas por el usuario y, la segunda tabla *Sectores*, contendrá todos los sectores de todas las parcelas de la tabla anterior.

La relación entre ambas es de 1 a muchos, donde una parcela podrá tener muchos sectores asociados.

Cada tabla contendrá los siguientes campos:

Parcelas

- *numero*: número de teléfono móvil asociado a esa parcela. Será aquél a cual mande la aplicación mensajes cuando quiera activar/desactivar el regadío de uno de sus sectores.
Será la clave primaria de esta tabla, con la cual se podrá identificar cada parcela puesto que cada una tendrá un número de teléfono único.
- *nombre*: nombre dado por el usuario para identificar de manera fácil cada parcela.
- *sectores*: número de sectores que componen esa parcela.
- *bomba*: tomará el valor 0 si la bomba de regadío está desactivada y 1 si está activada.

Sectores

- *id*: número identificativo usado por la aplicación para identificar a cada sector. Es, por tanto, la clave primaria de esta tabla y será un valor que imponga la base de datos de manera automática tras insertar una fila.
- *numero*: número de teléfono de la parcela de ese sector, con este campo se podrá identificar a cuál parcela está asociado ese sector.
- *activado*: campo que representará el estado de ese sector. Tomará el valor 0 si el regadío no está activado, 1 si está activado y 2 si se está esperando la confirmación de una acción.
- *nombreSector*: nombre dado por el usuario para identificar a ese sector de manera fácil.
- *pinArduino*: pin de la plataforma Arduino a la cual está asociada la electroválvula de ese sector.

Cada vez que el usuario solicite la activación de un sector de regadío y se obtenga una respuesta válida por parte de la placa empotrada, en la base de datos, el campo *activado* del sector seleccionado cambiará su valor a 1. En el caso de desactivar un sector, dicho campo tomará el valor de 0. Sin embargo, en el tiempo desde que el usuario pide la acción hasta que se confirma, el estado será el de *esperando confirmación* identificado por el valor 2 en la base de datos.

De la misma forma se actualizará el campo *bomba* de la tabla *Parcelas*: si hay algún sector activado (por tanto la bomba debe de estar activa) este campo tendrá el valor de 1 y 0 en caso contrario.

Para la creación de esta base de datos y llevar a cabo el acceso a ella, así como su modificación se ha creado una clase llamada *BDmanager* que hereda de *SQLiteOpenHelper*. A continuación se muestran sus principales métodos, contienen nombres descriptivos que indican la operación que realizan:

- *insertParcela()*.
- *updateParcela()*.
- *insertSectores()*.
- *updateSectores()*.
- *deleteParcela()*.
- *deleteSectores()*.
- *findParcela()*.
- *getAllParcelas()*.
- *getAllSectores()*.
- *findSector()*.
- *stateSector()*.

Objetos

La aplicación manejará dos objetos principales, parcelas y sectores, donde una parcela podrá tener uno o varios sectores asociados. Un objeto del tipo parcela se relaciona directamente con una tupla de la tabla *parcelas*, así como un objeto del tipo sector con una de *sectores*.

La creación de estos objetos ha sido motivada para poder gestionar de forma coherente toda la información de una parcela y sus sectores. Así se podrán manejar los datos de forma sencilla para su posterior interpretación, modificación y/o visualización.

SMS recibidos

Cuando un evento sucede, el sistema operativo Android envía un mensaje “broadcast” a las aplicaciones que previamente han registrado este evento. Un evento puede ser el recibo de un SMS, una llamada, batería baja, etc.

Para que una aplicación pueda “suscribirse” a los mensajes generados por un evento en concreto, debe crear y registrar un *BroadcastReceiver*, ya sea

contemplándolo previamente en el archivo manifest o registrándolo dinámicamente mientras se ejecuta la aplicación. (18)

Este componente es una gran herramienta que nos proporciona Android para controlar el suceso de un evento, por tanto ha sido el componente elegido para gestionar la llegada de nuevos SMS.

Como se ha descrito anteriormente, dicho BroadcastReceiver debe quedar registrado previamente para que el sistema operativo pueda enviar un mensaje a la aplicación móvil ante un evento concreto. Se ha optado por registrarlo en el archivo manifest con la siguiente entrada:

```
<receiver
  android:name=".SMSListener"
  android:enabled="true"
  android:exported="true">
  <intent-filter>
    <action android:name="android.provider.Telephony.SMS_RECEIVED" />
  </intent-filter>
</receiver>
```

Ilustración 33: registro del BroadcastReceiver en el archivo manifest

Así cuando el sistema detecte que ha llegado un SMS (evento identificado por "android.provider.Telephony.SMS_RECEIVED"), pondrá en ejecución el código contenido en el método *onReceive()* de la clase *SMSListener* de nuestra aplicación.

La aplicación móvil solo tratará mensajes entrantes que confirmen cambios de estado en el regadío de sectores registrados previamente. Es así que el código del citado método anterior contendrá las instrucciones necesarias para desempaquetar el SMS recibido, identificar la fuente de procedencia, procesar su contenido para ver si éste sigue el formato definido en la aplicación y, finalmente, si este formato se confirma, actualizar la base de datos con el nuevo cambio de estado en el regadío.

El ciclo de vida de los BroadcastReceiver es muy corto, cuando un evento registrado sucede, el sistema llama al método *onReceive()* del BroadcastReceiver que espera por tal suceso. Una vez termina de ejecutarse este método, el ciclo de vida del Broadcast termina.

SMS enviados

Para el envío de SMS desde la aplicación móvil se ha creado una clase llamada GSM, a continuación se muestran sus atributos y métodos.

Atributos:

- *ACTIVAR*: variable final que contendrá el texto que se deberá enviar cuando el usuario quiera activar el regadío de un sector.
- *DESACTIVAR*: variable final que contiene el texto que se debe enviar cuando el usuario quiere desactivar el regadío.

Métodos:

- *activar()*: método que será llamado cuando el usuario pulse el botón correspondiente en la interfaz para activar el regadío de un sector.
Este método creará un mensaje concatenando varios elementos:
 - ✓ *Acción*: acción de activación la cual está contenida en el atributo *ACTIVAR*.
 - ✓ *Sector en el cual realizar la acción*: número de sector. Este elemento puede tomar el valor de 1 a n, donde n es el número máximo de sectores asociados a la misma parcela.
- *desactivar()*: sigue el mismo procedimiento que el anterior, a excepción que cuando se le llame enviará un SMS con la acción de desactivación del riego.

Capítulo 5

Pruebas

Para asegurar el correcto funcionamiento conjunto de este proyecto se ha procedido a realizar distintas pruebas con el fin de identificar y resolver posibles fallos y errores.

A continuación se describirán todas las pruebas a las que se ha sometido este proyecto asegurando el cumplimiento de todos los requisitos descritos en apartados anteriores.

Con la realización de estas pruebas se han encontrado distintos fallos en el proyecto que han sido solucionados inmediatamente tras su identificación.

La tabla que contendrá las descripciones y resultados de cada prueba viene determinada por el siguiente modelo:

Identificador	RF –
Descripción	
Acciones	
Resultado	

Tabla 54: descripción tablas de pruebas

Cada campo de la tabla anterior indica:

- **Identificador:** Identificará unívocamente cada prueba realizada.
- **Descripción:** Contendrá una breve descripción de la prueba.
- **Acciones:** Acciones a realizar para la ejecución de la prueba descrita.
- **Resultado:** Indica el resultado final que se espera tras la ejecución de la prueba.

5.1 Pruebas de validación

Identificador	P - 01
Descripción	Agregar parcela con varios sectores de regadío.
Acciones	<ol style="list-style-type: none">1. Abrir la aplicación Android.2. Pulsar el botón “Agregar parcela”.3. Introducir el nombre, número de teléfono y número de sectores asociados.4. Completar los posteriores nombres para los sectores indicados.5. Pulsar el botón de “Guardar”.
Resultado	Válido. La parcela queda guardada correctamente y es posible visualizarla en la aplicación.

Tabla 55: tabla prueba P - 01

Identificador	P - 02
Descripción	Agregar parcela con campos en blanco.
Acciones	<ol style="list-style-type: none">1. Abrir la aplicación Android.2. Pulsar el botón “Agregar parcela”.3. Introducir el nombre, número de teléfono y dejar número de sectores asociados en blanco.4. Pulsar el botón de “Guardar”.
Resultado	Válido. La aplicación avisa al usuario que es necesario rellenar todos los campos. La parcela no queda guardada y, por tanto, no puede visualizarse en la aplicación.

Tabla 56: tabla prueba P - 02

Identificador	P - 03
Descripción	Agregar parcela con valores repetidos.
Acciones	<ol style="list-style-type: none">1. Abrir la aplicación Android.2. Pulsar el botón “Agregar parcela”.3. Introducir un nombre, un número de teléfono ya asociado a otra parcela y el número de sectores asociados.4. Pulsar el botón de “Guardar”.

Resultado	Válido. La aplicación avisa al usuario que no se pueden introducir números de teléfono repetidos. La parcela no queda guardada y, por tanto, no puede visualizarse en la aplicación.
-----------	--

Tabla 57: tabla prueba P - 03

Identificador	P - 04
Descripción	Visualizar todas las parcelas guardadas hasta el momento en la aplicación.
Acciones	<ol style="list-style-type: none"> 1. Abrir la aplicación Android. 2. Pulsar el botón “Gestionar parcelas”.
Resultado	Válido. Es posible visualizar una lista con todas las parcelas guardadas.

Tabla 58: tabla prueba P - 04

Identificador	P - 05
Descripción	Visualizar una parcela en concreto.
Acciones	<ol style="list-style-type: none"> 1. Abrir la aplicación Android. 2. Pulsar el botón “Gestionar parcelas”. 3. En la lista, buscar la parcela deseada y tocar sobre ella.
Resultado	Válido, se puede visualizar toda la información de la parcela seleccionada.

Tabla 59: tabla prueba P - 05

Identificador	P - 06
Descripción	Visualizar un sector de una parcela en concreto.
Acciones	<ol style="list-style-type: none"> 1. Abrir la aplicación Android. 2. Pulsar el botón “Gestionar parcelas”. 3. En la lista, buscar la parcela deseada y tocar sobre ella. 4. Pulsar sobre el sector deseado.
Resultado	Válido, se puede visualizar toda la información del sector seleccionado.

Tabla 60: tabla prueba P - 06

Identificador	P - 07
Descripción	Borrar una parcela guardada previamente.
Acciones	<ol style="list-style-type: none"> 1. Abrir la aplicación Android. 2. Pulsar el botón “Gestionar parcelas”.

	<ol style="list-style-type: none"> 3. En la lista, buscar la parcela deseada y realizar una pulsación larga. 4. En el menú desplegable elegir “borrar”.
Resultado	Válido, la parcela desaparece de la lista.

Tabla 61: tabla prueba P – 07

Identificador	P - 08
Descripción	Activar el riego de un sector con estado desactivado.
Acciones	<ol style="list-style-type: none"> 1. Abrir la aplicación Android. 2. Pulsar el botón “Gestionar parcelas”. 3. En la lista, buscar la parcela deseada y tocar sobre ella. 4. Pulsar sobre el sector deseado. 5. Pulsar sobre el botón “Activar”.
Resultado	Válido, el sector se activa y cambia su estado en la aplicación.

Tabla 62: tabla prueba P – 08

Identificador	P - 9
Descripción	Desactivar el riego de un sector con estado activado.
Acciones	<ol style="list-style-type: none"> 1. Abrir la aplicación Android. 2. Pulsar el botón “Gestionar parcelas”. 3. En la lista, buscar la parcela deseada y tocar sobre ella. 4. Pulsar sobre el sector deseado. 5. Pulsar sobre el botón “Desactivar”.
Resultado	Válido, el sector se desactiva y cambia su estado en la aplicación.

Tabla 63: tabla prueba P – 10

Identificador	P - 10
Descripción	Realizar una pérdida intencionada de cobertura en el módulo GSM.
Acciones	<ol style="list-style-type: none"> 1. Meter una orden al Arduino para que se desconecte de la red en un momento dado. 2. Mandar una orden de activación del riego en un sector. 3. Volver a obtener cobertura.
Resultado	Válido. El mensaje llega cuando el módulo vuelve a tener cobertura y se activa el sector especificado.

Tabla 64: tabla prueba P – 11

5.2 Matriz de trazabilidad de pruebas de funcionalidad y requisitos de sistema funcionales

	P-01	P-02	P-03	P-04	P-05	P-06	P-07	P-08	P-09	P-10
RSF-01										
RSF-02										
RSF-03										
RSF-04										
RSF-05										
RSF-06										
RSF-07										
RSF-08										
RSF-09										
RSF-10										
RSF-11										

Tabla 65: matriz trazabilidad de pruebas funcionales y requisitos de sistema funcionales

Capítulo 6

Planificación del trabajo

Con el objetivo de tener una clara visión de todas las tareas y trabajos realizados para realizar este Trabajo Fin de Grado, a continuación se enumerarán y explicarán en una tabla todas las fases de este proyecto, acompañadas de una breve descripción y del tiempo total empleado en la realización de cada una. Para finalizar, se expondrán estas tareas y su duración en un diagrama de Gantt (Ilustración 31) donde se podrá observar con facilidad las fechas de inicio y fin de cada una de ellas.

Tarea	Fecha comienzo	Fecha final	Duración (días)
Propuesta y concretización idea	01/11/2016	14/11/2016	10
Estudio de viabilidad	15/11/2016	21/11/2016	5
Documentación inicial	22/11/2016	30/11/2016	7
Análisis	01/12/2016	12/12/2016	10
Diseño del sistema	13/12/2016	23/12/2016	9
Instalación y configuración del entorno de trabajo	09/01/2017	16/01/2017	6
Implementación	17/01/2017	30/04/2017	74
Pruebas finales	01/05/2017	31/05/2017	23
Documentación	01/06/2017	10/07/2017	28
		Tiempo total (días)	172

Tabla 66: planificación del trabajo



Ilustración 34: diagrama de Gantt

Capítulo 7

Presupuesto

En esta sección se desglosará el presupuesto empleado en el desarrollo de este proyecto. Se detallarán los costes en función de la naturaleza de los mismos.

7.1 Resumen de tiempo dedicado

A continuación se resumen las horas totales dedicadas en el proyecto dadas a partir del apartado anterior sobre la planificación del trabajo.

Para el cálculo de horas se ha realizado lo siguiente:

$$\text{Horas} = \text{días laborales} * 8 \text{ horas}$$

Tareas	Duración horas
Propuesta y concretización idea	80
Estudio de viabilidad	40
Documentación inicial	56
Análisis	64
Diseño del sistema	72
Instalación y configuración del entorno de trabajo	48
Implementación	592
Pruebas finales	184
Documentación	224

Total:	1360 horas
--------	------------

7.2 Coste de personal

En la tabla siguiente se desglosa el coste del personal implicado en la realización de este proyecto en función de la dedicación indicada en el diagrama de Gantt.

Categoría	Dedicación (hombre/hora)	Coste hombre/hora (€)	Coste (€)
Analista	240	18,75	4500
Diseñador	120	13,125	1575
Programador	1000	12,5	12500

Donde el coste ha sido calculado mediante la siguiente fórmula:

$$Coste\ personal = \frac{Total\ días * horas/días}{Dedicación\ hombre/mes} * Coste\ hombre/mes$$

7.3 Coste de Hardware

A continuación se muestra el coste de los distintos equipos y herramientas hardware que han sido usadas en la realización del Trabajo Fin de Grado:

Equipo/herramienta	Coste (€)	Tiempo de vida (meses)	Coste mensual (€)	Meses utilizado	Coste amortizado (€)
Dell XPS13	1160	48	24,16	9	217,44
Xiaomi Mi5	320	24	13,33	5	66,7
Placa XDRduino	20				20
Módulo GSM SIM900	40				40
Componentes Arduino	15				15
Coste total hardware:					359,14

Para todos los equipos y herramientas mencionados anteriormente sólo se deben imputar los costes amortizados durante el desarrollo del proyecto debido a que su uso se ampliará en un futuro, no solamente siendo exclusivo del Trabajo Fin de Grado.

7.4 Coste de Software

Para la realización de este Trabajo Fin de Grado se han usado numerosos programas pero la gran parte de ellos son de código libre y gratis y por tanto no han aumentado el coste del proyecto. Se desglosan a continuación para los que sí ha sido preciso comprar una licencia.

Descripción	Coste (€)
Microsoft Office Hogar y Estudiantes 2016	149
Windows 10 Pro	279
Total:	428 €

7.5 Resumen de costes

Finalmente, en este apartado queda concretado el coste total del proyecto, donde habrá que incluir los costes de anteriores apartados, más costes indirectos, impuestos y un margen de riesgo y beneficio.

Se pueden observar en la siguiente tabla:

Descripción	Coste total
Personal	18575
Hardware	359,14
Software	428
Costes indirectos (20%)	3872,428
Margen de riesgo (15%)	2904,321
Beneficios (15%)	2904,321
Total sin IVA:	29043,21
Total con IVA (21%):	35142,2841

Capítulo 8

Entorno socio-económico

En el sector del cultivo pueden diferenciarse dos escenarios claros: Aquellos cultivos poco extensos y los que ocupan vastas extensiones de terreno.

Los cultivos poco extensos suelen darse en zonas cercanas a una vivienda o a zonas pobladas. La cercanía a estas zonas les ofrece una serie de ventajas a la hora de gestionar remotamente el abastecimiento de recursos a estos cultivos.

Estas ventajas tienen que ver con que cuentan con fácil acceso a internet. Hay millones de proyectos con grandes soluciones para dotar de automatismo a estos cultivos. Sin embargo, ocurre todo lo contrario con la otra cara de la moneda, aquellos cultivos de grandes dimensiones.

Los cultivos extensos suele situarse fuera de zonas pobladas, donde se restringe el acceso a internet. Esto ha creado un estancamiento en la creación de proyectos que aporten buenas soluciones para la gestión de éstos.

Actualmente hay gran variedad de productos basados en redes 3G/2G que pueden gestionar remotamente el cultivo, sin embargo la gran mayoría de ellos no están adaptados a estas vastas extensiones de terreno.

Por otra parte, también se cuenta con cantidad de proyectos que automatizan, hasta cierto punto, el control de estos cultivos. Sin embargo el agricultor deberá personarse en el terreno a cultivar para cambiar la programación del sistema de control.

De esta forma, se ha creado un sistema que permite el control total del agricultor sobre el regadío de su parcela de forma completamente remota. Este sistema está pensado exclusivamente para este segundo escenario: los cultivos extensos y alejados de las poblaciones.

Este proyecto repercutirá de manera clara en la calidad de vida del agricultor. Personarse dos veces diarias en cada parcela que tenga para activar y desactivar el riego contribuye a un gasto de tiempo y recursos verdaderamente valiosos para el agricultor. Además hay que tener en cuenta que estas parcelas suelen encontrarse alejadas unas de otras y de la vivienda de la persona encargada de ellas.

Con este sistema de control del riego el agricultor podrá mejorar la gestión de sus cultivos de manera significativa, proporcionándole un beneficio económico y personal importante.

Capítulo 9

Marco regulador

A continuación se detallarán las licencias de las tecnologías usadas en este proyecto:

- Android Studio: Ya que se usará este IDE para el desarrollo de la aplicación móvil, se deberán sus aceptar los términos y condiciones.
- En todo caso la creación de la aplicación quedará regulada por la licencia GNU GPL3 del sistema operativo Android la cual garantiza la libertad de usar, estudiar, copiar y modificar el software por parte del usuario final.

Por otro lado en este proyecto se usan datos potencialmente sensibles como lo son los números de teléfono móviles. Sin embargo, éstos no están adscritos a ninguna información del titular del mismo por lo que no se conciben como un dato de carácter personal en ningún momento. Con lo cual no hay que aplicar ninguna serie de medidas extra para evitar filtraciones de los mismos.

Finalmente, cabe destacar que la aplicación no será subida a Google Play con lo cual no deberá ser necesario aceptar las condiciones de ésta.

Capítulo 10

Conclusiones y trabajos futuros

10.1 Conclusiones del Proyecto

La realización de este proyecto ha estado guiada en todo momento con el propósito de satisfacer los objetivos propuestos en el apartado 1.3. Dichos propósitos provienen directamente de los requisitos planteados en capítulos anteriores, de tal manera que, el cumplimiento de unos, compromete el cumplimiento de los otros.

Sin duda, el objetivo que resume el principal motivo de realización de este Trabajo Fin de Grado ha sido cumplido satisfactoriamente:

- ✓ Automatizar el sistema de regadío actual, de tal manera que la persona encargada de su activación/desactivación pueda hacerlo desde cualquier lugar de manera remota, sin tener que realizar un desplazamiento.

Sin embargo, también se contaba con otros puntos derivados del objetivo principal, que permitían dotar al sistema de regadío de mayor usabilidad, los cuales también han sido cumplidos:

- ✓ Ofrecer al usuario una aplicación simple, útil y configurable para gestionar el riego de cada parcela, así como los diferentes sectores en los que puede dividirse ésta.
- ✓ Gestionar el riego de parcelas donde éstas quedarán divididas en diferentes sectores, de acuerdo a que, cada vez, el usuario pueda poner en funcionamiento sólo uno de estos o, en su defecto, varios a la vez.

10.2 Conclusiones personales

Como habitante de un pequeño pueblo donde la agricultura es el principal sustento, la realización de este proyecto viene motivada por el hecho de conocer de primera mano la falta de automatización en el sector agrícola. Así, el desarrollo de este Trabajo Fin de Grado tiene la intención de solucionar una necesidad real muy presente en mi vida cotidiana.

Respecto a las tecnologías usadas, Arduino me parece una solución muy completa y accesible para realizar todo tipo de proyectos que mejoren la calidad de vida de los usuarios. Trabajar con este microcontrolador ha abierto un amplio abanico de posibilidades para resolver problemas reales y muy cotidianos a los que aún no hemos dotado de automatismo, como por ejemplo en domótica.

Actualmente, estamos inversos en el llamado Internet de las cosas, mundo que se beneficia directamente de la gran accesibilidad a Internet y a elementos como teléfonos móviles inteligentes o plataformas como Arduino.

En este sentido gracias a Arduino se están creando múltiples iniciativas para satisfacer las necesidades del IoT, tales como sensores para el hogar, circuitos con control y gestión de cámaras de videovigilancia o incluso termostatos de tipo Nest.

Gracias a plataformas empotradas como Arduino a las que se puede acceder fácilmente, se crean millones de iniciativas para satisfacer las necesidades del IoT. Éstas incluyen sensores para el hogar, circuitos de control, gestión de videocámaras... (19)

Es así que, gracias al desarrollo de este sistema de regadío, he podido introducirme en este nuevo mundo y ver de primera mano los grandes proyectos que se realizan a diario con tecnologías caseras y que ayudan a conectar a humanos con “las cosas” que nos rodean.

Por otro lado, la implementación de la aplicación en Android me ha hecho conocer el gran sistema operativo móvil que es, donde se pueden desarrollar grandes ideas en forma de aplicaciones, de las que se podrán beneficiar la gran cantidad de usuarios que tiene este sistema detrás. También, gracias al desarrollo de esta parte, he mejorado mis conocimientos acerca de éste ámbito y me he familiarizado con muchos más de los servicios que Android ofrece.

Por último, este proyecto es una pequeña parte de todo el abanico de posibilidades y funcionalidades que se podrán añadir a este sistema en un futuro. Este Trabajo

Fin de Grado soluciona la idea más básica propuesta, de tal forma que, más adelante, este proyecto pueda ser completado con todas las líneas descritas en el apartado siguiente, creando finalmente un sistema capaz de ser implementado de manera autónoma en el entorno para el que ha sido ideado.

10.3 Trabajos futuros

Debido a la extensión de este proyecto, este no es más que un prototipo del modelo plenamente funcional que se quiere realizar en un futuro. Para ello, posteriormente se deberán incluir las siguientes funcionalidades:

- En el modelo final, la placa Arduino situada en la parcela deberá ser alimentada por la energía proporcionada por una placa solar. En este prototipo, sin embargo, los componentes se han conectado a una fuente de alimentación local.
- Para instalar una bomba de regadío en una parcela concreta es necesario realizar un estudio de la extensión y situación de la parcela, número de sectores, etc. Es por esto que para el desarrollo de este Trabajo Fin de Grado, se ha usado un LED como simulación del funcionamiento de la bomba, pero en un futuro deberá ser sustituido por el dispositivo adecuado.

Por otro lado, se podrán añadir distintas mejoras y complementos que añadan nuevas funcionalidades al sistema, tales como:

- Se podrán añadir diferentes sensores en las parcelas que proporcionen información útil al usuario, tal como la humedad de la tierra o la temperatura, ayudándole a usar eficientemente los recursos y, por tanto, tomar mejores decisiones sobre cada uno de los sectores con los que cuente dicha parcela.
En este caso, se deberá realizar un estudio de la parcela para encontrar los lugares óptimos donde colocar dichos dispositivos.
- Con los sensores mencionados en el punto anterior, se podrá crear un sistema capaz de gestionar de manera autónoma los recursos de agua. El usuario podrá introducir en la aplicación márgenes superiores e inferiores de activación y desactivación automáticos del regadío.

- También se podrán incluir funcionalidades propias de los temporizadores. Donde se podría programar el inicio y fin de los periodos del riego en cada sector.

Glosario

ANDROID	<i>Sistema operativo móvil para teléfonos inteligentes táctiles basado en el núcleo de Linux.</i>
ARDUINO UNO	<i>Placa empotrada con microcontrolador.</i>
DOMÓTICA	<i>Conjunto de técnicas de control para dotar de automatismo e inteligencia a una vivienda.</i>
LED	<i>Diodo emisor de luz.</i>
GSM	<i>Sistema global para las comunicaciones móviles.</i>
SIM	<i>Módulo de identificación de suscripción.</i>
SMS	<i>Servicio de mensajes cortos.</i>
TFG	<i>Trabajo fin de grado.</i>
IoT	<i>Internet de las Cosas.</i>
WPAN	<i>Redes inalámbricas de área personal.</i>
WLAN	<i>Redes inalámbricas de área local.</i>
WMAN	<i>Redes inalámbricas de área metropolitana.</i>
WWAM	<i>Redes inalámbricas de gran alcance</i>
LTE	<i>Cuarta generación de tecnologías de telefonía móvil.</i>
GPRS	<i>Servicio general de paquetes vía radio.</i>
HSPA	<i>Acceso de paquetes a alta velocidad.</i>

3G	<i>Tercera generación de transmisión de voz y datos en telefonía móvil.</i>
2G	<i>Segunda generación de telefonía móvil.</i>
SBC	<i>Ordenador de placa reducida.</i>
RAM	<i>Memoria de acceso aleatorio.</i>
SRAM	<i>Memoria estática de acceso aleatorio.</i>
EPROM	<i>Memoria de solo lectura programmable y borrable.</i>
CPU	<i>Unidad central de procesamiento.</i>
GPU	<i>Unidad de procesamiento gráfico.</i>
HDMI	<i>Interfaz multimedia de alta definición.</i>
USB	<i>Bus serial universal.</i>
IDE	<i>Entorno de desarrollo integrado.</i>
PWM	<i>Modulación por ancho de pulsos.</i>
OpenWRT	<i>Firmware basado en Linux para dispositivos empotrados.</i>
MIPS32	<i>Arquitectura para computadores con conjuntos de instrucciones reducidas.</i>
SO	<i>Sistema operativo.</i>
Wi-Fi	<i>Mecanismo para conectar dispositivos de manera inalámbrica.</i>

Bibliografía

1. Sanz, Elena. Muyinteresante.es. *¿Qué es el "Internet de las cosas"?* [En línea] [Citado el: 10 de Julio de 2017.] <https://www.muyinteresante.es/curiosidades/preguntas-respuestas/ique-es-el-qinternet-de-las-cosasq>.
2. Blázquez, Josep Prieto. exabyteinformatica.com. *Introducción a los sistemas de comunicación inalámbricos*. [En línea] [Citado el: 10 de Junio de 2017.] Introducción a los sistemas de comunicación inalámbricos. [https://www.exabyteinformatica.com/uoc/Informatica/Tecnologia_y_desarrollo_en_dispositivos_moviles/Tecnologia_y_desarrollo_en_dispositivos_moviles_\(Modulo_1\).pdf](https://www.exabyteinformatica.com/uoc/Informatica/Tecnologia_y_desarrollo_en_dispositivos_moviles/Tecnologia_y_desarrollo_en_dispositivos_moviles_(Modulo_1).pdf).
3. Raúl V. Lerma Blasco, Elvira Mifsud Talón. *Servicios en red*. s.l. : McGraw-Hill Interamericana de España S.L, 2013.
4. raspberrypi.org. www.raspberrypi.org. *About us*. [En línea] [Citado el: 23 de Junio de 2017.] <https://www.raspberrypi.org/about/>.
5. www.raspberrypi.org. *Downloads*. [En línea] [Citado el: 29 de Junio de 2017.] <https://www.raspberrypi.org/downloads/>.
6. es.wikipedia.org. *Arduino*. [En línea] [Citado el: 10 de Junio de 2017.] <https://es.wikipedia.org/wiki/Arduino>.
7. Aprendiendoarduino.wordpress.com. *Lenguaje de programación de Arduino, estructura de un programa*. [En línea] [Citado el: 10 de Junio de 2017.] <https://aprendiendoarduino.wordpress.com/2015/03/26/lenguaje-de-programacion-de-arduino-estructura-de-un-programa/>.
8. Corporation, Onion. onion.io. *omega2*. [En línea] [Citado el: 08 de Junio de 2017.] <https://onion.io/omega2/>.

9. Prometec.net. *SMS, MÓDULO GSM/GPRS: LLAMAR Y ENVIAR*. [En línea] [Citado el: 1 de Septiembre de 2017.] <http://www.prometec.net/gprs-llamar-enviar-sms/>.
10. EC, Redacción. elcomercio.pe. *Adiós, Windows Phone: Microsoft solo hablará de Windows 10*. [En línea] [Citado el: 19 de Julio de 2017.] <http://elcomercio.pe/paginas/adios-windows-phone-microsoft-hablara-windows-10-326573>.
11. es.wikipedia.org. *iOS*. [En línea] [Citado el: 10 de Junio de 2017.] <https://es.wikipedia.org/wiki/IOS>.
12. Gonzalez, Alejandro Nieto. www.xatakandroid.com. *¿Qué es Android?* [En línea] [Citado el: 2017 de Junio de 22.] <https://www.xatakandroid.com/sistema-operativo/que-es-android>.
13. www.android.com. *History*. [En línea] [Citado el: 26 de Junio de 2017.] https://www.android.com/intl/es-419_mx/history.
14. greeniq.co. *GreenIQ Smart Garden Hub*. [En línea] [Citado el: 12 de Junio de 2017.] http://greeniq.co/product_tech.htm.
15. greeniq.co. *Technical Specifications*. [En línea] [Citado el: 13 de Junio de 2017.] <http://greeniq.co/specifications.htm>.
16. Center, IBM® Knowledge. *Modelo de cliente/servidor*. [En línea] [Citado el: 6 de Junio de 2017.] https://www.ibm.com/support/knowledgecenter/es/SSAL2T_8.2.0/com.ibm.cics.tx.doc/concepts/c_clnt_sevr_model.html.
17. es.wikipedia.org. *Electroválvula*. [En línea] [Citado el: 16 de Junio de 2017.] <https://es.wikipedia.org/wiki/Electrov%C3%A1lvula>.
18. developer.android.com. *Broadcast*. [En línea] [Citado el: 23 de Julio de 2017.] <https://developer.android.com/guide/components/broadcasts.html>.
19. Xataka.com. *Las 3 tecnologías clave para el Internet de las cosas*. [En línea] [Citado el: 11 de Agosto de 2017.] <https://www.xataka.com/internet-of-things/las-3-tecnologias-clave-para-el-internet-de-las-cosas>.
20. Aprendiendoarduino.wordpress.com. *Actuadores y periféricos de salida*. [En línea] [Citado el: 1 de Septiembre de 2017.] <https://aprendiendoarduino.wordpress.com/tag/rele/>.

