



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

# Monitorización y gestión de una red de sensores inteligentes: aplicación móvil

TRABAJO FINAL DE GRADO

Grado en Ingeniería Informática

*Autor:* Sergio Guijarro Carcelén

*Director:* Jose Luis Poza Luján

*Co-Director:* Juan Luis Posadas Yagüe

Curso 14/15

## Resumen

Este proyecto presenta una aplicación móvil basada en Android para la gestión de un sistema de sensores y actuadores. Un sistema de sensores y actuadores hace referencia a una serie de dispositivos que están conectados a un servidor, dichos dispositivos, disponen internamente tanto de sensores (para tomar distintos tipos de medidas del entorno) como de actuadores (para interactuar sobre el entorno).

La aplicación proporciona la capacidad para poder acceder a los sensores y actuadores, de manera que se ven las medidas almacenadas ó que está tomando un sensor en el instante, asimismo como enviar órdenes a un actuador (por ejemplo: encender una luz o emitir un sonido).

La aplicación evita la dependencia de accesorios que otras aplicaciones del sector tienen, ya que, está preparado para montar dispositivos propios de bajo coste. De esta forma, disponer de un solo dispositivo que cumpla con toda la recogida de información que se necesita, o bien, tantos como sean necesarios para tener todo el entorno monitorizado y controlado.

Todo ello a partir de unas interfaces sencillas y fáciles de manejar para, de esta forma, controlar el sistema eficientemente desde el dispositivo móvil.

*Palabras clave:* sensor, actuador, dispositivo inteligente, dispositivo móvil, monitorización, control, Internet de las cosas, Android

## Abstract

This project presents a mobile application that is based on Android for management of a smart sensors and actuators network. A sensor and actuator system refers to many devices that are connected to a server; such devices have internally both sensors (in order to measure different values from the environment) and actuators (in order to interact on the environment).

The app gives the capability to access the sensors and actuators in a way that the measurements stored or the ones a sensor is taking at the moment can be seen, as well as send orders to an actuator (such as switching on a light or producing a sound).

The app prevents the reliance of accessories that others apps have because it is prepared for designing its own low-cost devices. Thus, it is possible to have just one device for all the information required or as many as it is necessary to have the whole environment tracked and controlled.

All this is carried out through simple and easy-to-use interfaces so that the user can manage the whole system from the mobile device in an efficient way.

*Keywords:* sensor, actuator, smart device, mobile device, monitorization, control, Internet of Things, Internet, Android

# Índice general

<b>Lista de figuras</b>	<b>2</b>
<b>Lista de tablas</b>	<b>3</b>
<b>1. Introducción</b>	<b>4</b>
1.1. Contexto, entorno (motivación)	4
1.2. Objetivos	5
1.3. Descripción del documento	5
<b>2. Estudio del entorno</b>	<b>7</b>
2.1. Introducción	7
2.2. Sistemas similares	7
2.2.1. Sensor Kit	7
2.2.2. Sensor Network for LabVIEW	8
2.2.3. Wireless sensor networks	8
2.2.4. Wireless sensor location	9
2.2.5. Sensores multiherramienta	9
2.2.6. Sensor Ex	10
2.2.7. Sensors	10
2.2.8. EMX Sensors	11
2.2.9. Wally - Home Sensor Network	11
2.2.10. Buscar mi iPhone	12
2.2.11. WeMo	12
2.2.12. Nest	13
2.2.13. Home Network System	14
2.2.14. Enviromux-xD	14
2.2.15. Farm Logix1	15
2.2.16. My Weather Station II	15
2.2.17. Bluetooth Smart Scanner	16
2.2.18. BlueSense	16
2.2.19. Smart Barn	16

2.2.20. Snap Switch . . . . .	16
2.2.21. HomeWizard . . . . .	16
2.2.22. SmartThings Mobile . . . . .	17
2.2.23. Band Sensor Monitor . . . . .	17
2.2.24. INSTEON for Hub . . . . .	18
2.2.25. Smart Home . . . . .	18
2.2.26. IQONTROL . . . . .	19
2.2.27. Home Controller . . . . .	19
2.2.28. I Love Lamp . . . . .	20
2.3. Análisis . . . . .	20
2.3.1. Exposición de características . . . . .	20
2.3.2. Análisis detallado de las características de las aplica- ciones . . . . .	22
2.3.3. Análisis de funcionalidades . . . . .	25
2.4. Síntesis . . . . .	29
2.5. Conclusiones . . . . .	30
<b>3. Especificación de requisitos</b>	<b>31</b>
3.1. Introducción . . . . .	31
3.2. Definiciones . . . . .	31
3.3. Requisitos funcionales . . . . .	32
3.3.1. Listado Dispositivos . . . . .	32
3.3.2. Listado de Sensores de un Dispositivo . . . . .	32
3.3.3. Listado de Sensores . . . . .	32
3.3.4. Listado Actuadores de un Dispositivo . . . . .	33
3.3.5. Listado Actuadores . . . . .	33
3.3.6. Gestión Sensor . . . . .	33
3.3.7. Gestión Actuator . . . . .	33
3.3.8. Diseño . . . . .	34
3.3.9. Requisitos de usuario y tecnológicos . . . . .	34
<b>4. Diseño del sistema</b>	<b>35</b>
4.1. Introducción . . . . .	35
4.2. Especificación conceptual . . . . .	35
4.3. Especificación formal . . . . .	37
4.3.1. Capa persistencia . . . . .	37
4.3.2. Capa negocio . . . . .	39
4.3.3. Capa presentación . . . . .	40
4.4. Conclusiones . . . . .	46

---

<b>5. Implementación e implantación</b>	<b>47</b>
5.1. Introducción . . . . .	47
5.2. Implementación . . . . .	47
5.2.1. Capa persistencia . . . . .	47
5.2.2. Capa negocio . . . . .	49
5.2.3. Capa presentación . . . . .	51
5.3. Evaluación . . . . .	55
5.4. Implantación . . . . .	56
5.5. Conclusiones . . . . .	57
<b>6. Conclusiones</b>	<b>58</b>
6.1. Dificultades encontradas . . . . .	58
6.2. Aportaciones . . . . .	59
6.3. Ampliaciones . . . . .	59
<b>7. Referencias</b>	<b>60</b>

# Índice de figuras

2.1. Sensor Network For LabVIEW . . . . .	8
2.2. Wireless Sensor Location . . . . .	9
2.3. Sensor Ex . . . . .	10
2.4. Wally . . . . .	11
2.5. Buscar mi iPhone . . . . .	12
2.6. WeMo . . . . .	13
2.7. Nest . . . . .	14
2.8. My Weather Station II . . . . .	15
2.9. HomeWizard . . . . .	17
2.10. Band Sensor Monitor . . . . .	18
2.11. IQONTROL . . . . .	19
2.12. Precios apps . . . . .	23
2.13. Tiempo medio actualización por Sistema Operativo . . . . .	23
2.14. Tiempo medio hasta actualización . . . . .	24
2.15. Idiomas aplicaciones . . . . .	24
2.16. Aprendizaje manejo app . . . . .	27
2.17. Apariencia app . . . . .	27
2.18. Usabilidad app . . . . .	28
2.19. Variedad Sensores . . . . .	28
4.1. Diagrama conceptual 1 . . . . .	35
4.2. Diagrama conceptual 2 . . . . .	36
4.3. Diagrama conceptual 3 . . . . .	37
4.4. Casos de uso . . . . .	39
4.5. Mockup de Inicio de la aplicación . . . . .	40
4.6. Mockup de Listado Dispositivos . . . . .	41
4.7. Mockup de Listado Sensores . . . . .	42
4.8. Gestión Sensor . . . . .	43
4.9. Listado Actuadores . . . . .	44
4.10. Gestión Actuador . . . . .	45
4.11. Pantalla Información . . . . .	46

---

5.1. JSON lista de Dispositivos . . . . .	47
5.2. JSON lista de Sensores . . . . .	48
5.3. JSON lista de Actuadores . . . . .	48
5.4. JSON Dispositivo completo . . . . .	49
5.5. Código .java Sensores . . . . .	50
5.6. Captura pantalla de Inicio . . . . .	51
5.7. Captura pantalla de Sensores . . . . .	52
5.8. Captura pantalla de Actuadores . . . . .	53
5.9. Captura pantalla de Información . . . . .	54
5.10. Captura pantalla de Ajustes . . . . .	55
5.11. Ajustes orígenes desconocidos . . . . .	57

# Índice de tablas

2.1. Exposición características de las aplicaciones analizadas 1 . . .	21
2.2. Exposición características de las aplicaciones analizadas 2 . . .	22
2.3. Análisis de funcionalidades . . . . .	26

# Capítulo 1

## Introducción

### 1.1. Contexto, entorno (motivación)

Cada día que pasa hay muchas más cosas con Internet, objetos cotidianos que están continuamente conectados. Hace unos años era impensable el poder tener una pulsera, un televisor o incluso la nevera y el termostato conectados y sin embargo hoy en día, el tener una tele con conexión que nos permita navegar por Internet o un termostato al que le regulamos la temperatura sin estar en casa, es lo más normal.

Esto es lo que se le llama el Internet de las cosas (**IoT** = Internet of Things). Pero cuando se habla de IoT surge una pregunta, ¿a qué se refiere con cosa? En este contexto es un **smart device** o dispositivo inteligente. Entonces, si hay smart devices y los tenemos conectados, se puede:

- Saber qué ocurre a su alrededor y monitorizar los datos que recoge en tiempo real.
- Influir en su entorno actuando sobre ellos sin estar allí.

Además, teniendo smart devices en todas partes, y casi siempre, uno con nosotros conectado a internet como es nuestro dispositivo móvil, se puede utilizar una app (aplicación que se instala en el mismo) que permita realizar las funciones anteriormente detalladas.

Teniendo en cuenta la libertad de un S.O móvil como da Android, se puede aprovechar y desarrollar dicha app para gestionar los smart device que se quiera con solo sacar nuestro móvil del bolsillo.

## 1.2. Objetivos

**Objetivo general:** Especificar, diseñar e implementar una aplicación en Android para dispositivos móviles que permita monitorizar sobre los dispositivos sensores e influir sobre los dispositivos actuadores. Se resume en los siguientes objetivos parciales:

- **Objetivo 1:** Estudiar los sistemas similares al que se deberá desarrollar para analizarlos y conocer las características más relevantes y así poder incorporarlas al producto que se va a desarrollar.
- **Objetivo 2:** Especificar las características del producto a desarrollar para poder sentar las bases del diseño y de la implementación del mismo.
- **Objetivo 3:** Diseñar mediante un método formal la aplicación de manera que cumpla con las especificaciones de los requisitos desarrolladas en el objetivo anterior.
- **Objetivo 4:** Implementar y evaluar la aplicación diseñada para que se ajuste al objetivo general del proyecto.

## 1.3. Descripción del documento

En **este capítulo** se han presentado el contexto del que nace el proyecto y la necesidad que cubre, así como los objetivos del mismo. Además se presenta el documento en líneas generales, dando una visión detallada de que se va a encontrar en cada unos de los apartados del mismo.

El **capítulo 2**, comienza con el apartado de Introducción donde se dará una rápida descripción del estudio que se va a realizar sobre lo mostrado en el apartado de Sistemas Similares para, posteriormente profundizar en el apartado de Análisis explicando detalladamente las características y funcionalidades de los mismos. Después, se sintetizarán las características que deberá tener una aplicación de dicha temática y se sacarán las Conclusiones.

El **capítulo 3**, comienza con el apartado de Definiciones donde se darán unas breves explicaciones sobre unos conceptos necesarios, después se definirá los requisitos funcionales para cada una de las interfaces y la salida esperada para, seguidamente, proceder a especificar los requisitos de diseño y por último los requisitos de usuario y tecnológicos de la aplicación.

El **capítulo 4**, comienza con el apartado de Introducción donde se dará una rápida descripción de las especificaciones sobre el diseño del sistema a realizar para, después darnos en el siguiente apartado una Especificación

conceptual y en el siguiente una Especificación formal donde se hablará, de las capas de Persistencia, Negocio y Presentación. Después por último se sacarán las conclusiones.

El **capítulo 5**, comienza con el apartado de Introducción donde se hará una rápida descripción acerca de la implementación e implantación para, en el apartado siguiente Implementación mostrar las capas de Persistencia, Negocio y Presentación. Luego se dará la información de la Evaluación de la aplicación, de su Implantación y unas Conclusiones.

En el **capítulo 6**, se comienza con las conclusiones donde se habla de las Dificultades Encontradas, seguir hablando de las Aportaciones del proyecto y finalmente hablar de las posibles ampliaciones.

Por último, el **capítulo 7** que se enumerarán las distintas Referencias de interés.

# Capítulo 2

## Estudio del entorno

### 2.1. Introducción

En este apartado se van a describir los distintos sistemas similares al nuestro encontrado durante el análisis del entorno. Para la realización del mismo, se han tenido en cuenta tanto las valoraciones de los usuarios a nivel de comentario como las características técnicas del producto. Todo ello se ha realizado mediante las distintas tiendas de aplicaciones de los distintos sistemas móviles actuales (Android, IOS y Windows Phone).

### 2.2. Sistemas similares

En este apartado se va a hablar sobre cada uno de los sistemas similares. Se han analizado un total de 31 aplicaciones de los 3 sistemas operativos móviles. Como se puede observar en las dos siguientes imágenes (Figuras 2.1 y 2.2) se describen las características técnicas de cada sistema.

#### 2.2.1. Sensor Kit

El primer sistema analizado está desarrollado por Ace Sensor Inc y necesita de un sistema operativo Android, como mínimo en la versión 4.3. La aplicación es de instalación gratuita y se encuentra en la versión 1.2.0 la cual data de la fecha 3 de Diciembre de 2014. Sólo está disponible en inglés y permite la utilización de sensores externos mediante Bluetooth de bajo consumo.

### 2.2.2. Sensor Network for LabVIEW

Este sistema está desarrollado por SCCT Development team y necesita un sistema operativo Android, como mínimo en la versión 2.1. Además, la aplicación es de instalación gratuita y se encuentra en la versión 1.1.2 la cual data de la fecha 20 de Octubre de 2013. Sólo está disponible en inglés. Esta aplicación permite la comunicación entre el dispositivo móvil y LabVIEW de modo que éste último pueda recoger datos de los sensores del primero.

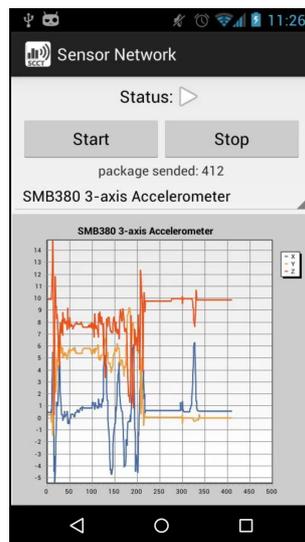


Figura 2.1: Sensor Network For LabVIEW

La **ventaja** de la aplicación es que permite la utilización de un dispositivo móvil y usar sus datos en LabVIEW.

El **inconveniente** es que su usabilidad es muy concreta y sólo permite gastar los datos del dispositivo móvil, no de otros sensores externos.

En **conclusión** es una gran aplicación que, necesitaría algún añadido más, como el poder gastar información de sensores externos que no fueran de un dispositivo móvil.

### 2.2.3. Wireless sensor networks

Este sistema está desarrollado por O'Reilly Media y necesita un sistema operativo Android, como mínimo en la versión 1.5. Además, la aplicación necesita ser comprada al precio de 3,64€ para poder instalarla y se encuentra en la versión 1.2.10 la cual data de la fecha 20 de Febrero de 2011. La aplicación sólo está disponible en inglés.

### 2.2.4. Wireless sensor location

Este sistema está desarrollado por IT - Universidade de Aveiro y necesita un sistema operativo Android, como mínimo en la versión 2.1. Además, la aplicación es de instalación gratuita y se encuentra en la versión 1 la cual data de la fecha 11 de Mayo de 2011. Sólo está disponible en inglés y no permite la utilización de sensores externos. Permite la localización usando el GPS y Sensor de redes inalámbricas.



Figura 2.2: Wireless Sensor Location

La **ventaja** de la aplicación es que permite recoger datos de los sensores del móvil para su posterior envío al ordenador.

El **inconveniente** es que la interfaz está muy mal distribuida y queda visualmente poco atractiva.

En **conclusión** es una gran aplicación que necesitaría algún añadido más, dado que el propio móvil hoy en día te puede mostrar dicha información con la aplicación de mapas, instalada por defecto.

### 2.2.5. Sensores multiherramienta

Este sistema está desarrollado por Wered Software y necesita de un sistema operativo Android, como mínimo en la versión 4.0. Además, la aplicación es de instalación gratuita y se encuentra en la versión 1.0.3 la cual data de la fecha 2 de Febrero de 2015. Sólo está disponible en inglés y no permite la utilización de sensores externos. Sirve para recibir información de todos los sensores que posee tu dispositivo móvil, con gráficas en tiempo real.

### 2.2.6. Sensor Ex

Este sistema está desarrollado por TarCo y necesita de un sistema operativo Android, como mínimo en la versión 2.2. Además la aplicación es de instalación gratuita y se encuentra en la versión 1.6.4 la cual data de la fecha 6 de Febrero de 2013. Sólo está disponible en inglés y no permite la utilización de sensores externos. La aplicación sirve para la recogida de datos de los sensores del dispositivo móvil y los envía mediante WiFi al ordenador.

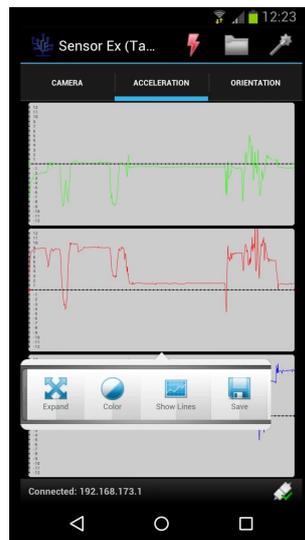


Figura 2.3: Sensor Ex

La **ventaja** de la aplicación es que permite localizar un dispositivo y ubicarlo en un mapa. Además, se puede saber la latitud y la longitud de nuestra posición.

El **inconveniente** es que su apariencia está muy poco cuidada.

En **conclusión** es una gran aplicación que, necesitaría un cambio en cuanto a la apariencia ya que, no llama la atención y puede parecer muy descuidada.

### 2.2.7. Sensors

Este sistema está desarrollado por Tom Rice y necesita de un sistema operativo Android, como mínimo en la versión 2.3. Además la aplicación es de instalación tras compra por 0,74€ y se encuentra en la versión 1.2 la cual data de la fecha 14 de Abril de 2013. Sólo está disponible en inglés y no permite la utilización de sensores externos. La aplicación sirve para la recogida de datos de los sensores del dispositivo móvil.

### 2.2.8. EMX Sensors

Este sistema está desarrollado por EMX Industries y necesita de un sistema operativo Android, como mínimo en la versión 4.0. Además la aplicación es de instalación gratuita y se encuentra en la versión 12.0 la cual data de la fecha 11 de Julio de 2014. Sólo se encuentra disponible en inglés y no permite la utilización de sensores externos.

### 2.2.9. Wally - Home Sensor Network

Este sistema está desarrollado por SNUPI Technologies y dispone de dos versiones (las cuales sólo están disponibles en inglés):

- Para **ANDROID** necesita como mínimo una versión 4.0 del sistema y se encuentra en la versión 2.9 la cual data del 29 de Enero de 2015.
- Para **IOS** necesita como mínimo una versión 7.0 del sistema y se encuentra en la versión 1.14 la cual data del 2 de Febrero de 2015.

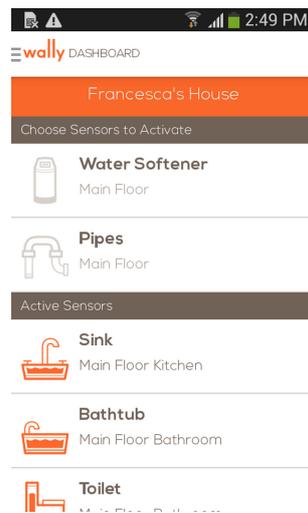


Figura 2.4: Wally

La **ventaja** de la aplicación es que tiene una interfaz sencilla, con una estética muy cuidada y es de fácil manejo.

El **inconveniente** es que necesitas disponer de los sensores de la misma, lo cual hace que no puedas utilizarlo con sensores caseros.

En **conclusión** es una gran aplicación pero que se aleja de la nuestra ya que se quiere que se pueda gastar con cualquier sensor que se conecta al servidor.

### 2.2.10. Buscar mi iPhone

Este sistema está desarrollado por iTunes S.a.r.l y necesita de un sistema operativo IOS, como mínimo en la versión 8.0. Además la aplicación es de instalación gratuita y se encuentra en la versión 4.0 la cual data de la fecha 17 de Septiembre de 2014. Se encuentra disponible en 32 idiomas y si permite la utilización de sensores externos.

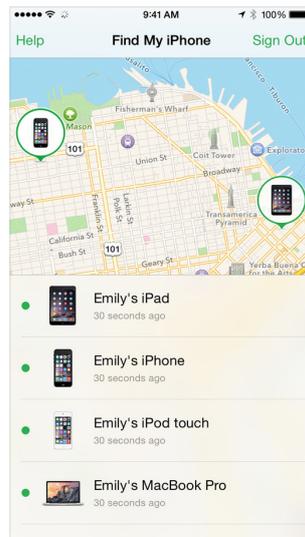


Figura 2.5: Buscar mi iPhone

La **ventaja** de la aplicación es que tiene una interfaz sencilla con una estética muy cuidada y, es de fácil manejo. Además, permite localizar dispositivos y, luego, bloquearlos o borrar su contenido u otras utilidades varias (para el caso de que se hayan perdido).

El **inconveniente** es que necesitas que el dispositivo esté conectado para poder comunicarte con él.

En **conclusión** es una gran aplicación pero que se aleja de la nuestra ya que esta sólo sirve para casos muy concretos, como localizar un dispositivo perdido y poder bloquearlo.

### 2.2.11. WeMo

Este sistema está desarrollado por Belkin y dispone de dos versiones (las cuales están disponibles en 9 idiomas):

- Para **ANDROID** necesita como mínimo una versión 4.0 del sistema y se encuentra en la versión 1.10.1 la cual data del 1 de Diciembre de 2014.

- Para **IOS** necesita como mínimo una versión 6.0 del sistema y se encuentra en la versión 1.10.2 la cual data del 5 de Enero de 2015.

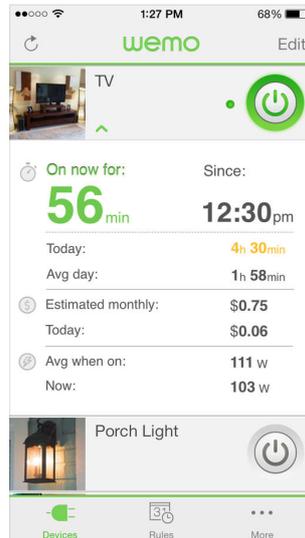


Figura 2.6: WeMo

La **ventaja** de la aplicación es que tiene fácil manejo. Además permite ver el consumo de los dispositivos conectados, saber desde cuando lo están, recibir notificaciones o encender dispositivos cuando detectan algo.

El **inconveniente** es que necesitas que el dispositivo con el que conectarse sea de la propia compañía.

En **conclusión** es una gran aplicación pero que se aleja de la nuestra ya que esta sólo sirve para los sensores de la propia compañía, se quiere que funcionen con cualquier sensor conectado al servidor.

### 2.2.12. Nest

Este sistema está desarrollado por Nest Labs, Inc. y dispone de dos versiones (las cuales están disponibles en 5 idiomas):

- Para **ANDROID** necesita como mínimo una versión 2.3.3 del sistema y se encuentra en la versión 4.5.2.3 la cual data de la fecha 24 de Febrero de 2015.
- Para **IOS** necesita como mínimo una versión 6.0 del sistema y se encuentra en la versión 4.5.2 la cual data de la fecha 24 de Febrero de 2015.



Figura 2.7: Nest

La **ventaja** de la aplicación es que tiene una muy cuidada interfaz y a su vez sencilla.

El **inconveniente** es que necesitas que el dispositivo con el que conectarse sea de la propia compañía.

En **conclusión** es una gran aplicación pero que se aleja de la nuestra ya que esta sólo sirve para los sensores de la propia compañía, la idea es buena pero le falta poder conectar cualquier sensor.

### 2.2.13. Home Network System

Este sistema está desarrollado por Panasonic System Networks Co. Ltd y necesita de un sistema operativo IOS, como mínimo en la versión 5.0. Además la aplicación es de instalación gratuita y se encuentra en la versión 1.4 la cual data de la fecha 25 de Febrero de 2015. Se encuentra disponible en 44 idiomas y no permite la utilización de sensores externos.

### 2.2.14. Enviromux-xD

Este sistema está desarrollado por Network Technologies Inc y necesita de un sistema operativo IOS, como mínimo en la versión 6.1. Además la aplicación es de instalación tras compra por 49,99€ y se encuentra en la versión 1.0 la cual data de la fecha 30 de Enero de 2014. Se encuentra disponible solamente en inglés, permite la utilización de sensores externos. La aplicación sólo se utiliza como monitor de sensores externos de la misma marca.

### 2.2.15. Farm Logix1

Este sistema está desarrollado por CPF IT CENTER COMPANY LIMITED y necesita de un sistema operativo IOS, como mínimo en la versión 4.3. Además la aplicación es de instalación gratuita y se encuentra en la versión 1.0, la cual data de la fecha 23 de Octubre de 2011. Además la aplicación se encuentra disponible en inglés y permite la utilización de sensores externos. La aplicación sirve como monitor de sensores instalados.

### 2.2.16. My Weather Station II

Este sistema está desarrollado por JrYao Chang y necesita de un sistema operativo IOS, como mínimo en la versión 7.1. Además la aplicación es de instalación gratuita y se encuentra en la versión 1.1.1, la cual data de la fecha 13 de Diciembre de 2014. Además la aplicación se encuentra disponible en 3 idiomas y permite la utilización de sensores externos.

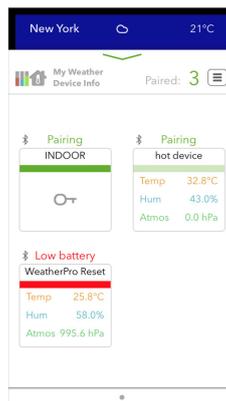


Figura 2.8: My Weather Station II

- La **ventaja** de esta aplicación está en la fácil utilización de la misma, con una interfaz sencilla. Además se sincroniza fácilmente con los accesorios "WeatherCube" de la marca.
- El **inconveniente** se encuentra en la necesidad de usar dichos "WeatherCube".
- En **conclusión** es una aplicación buena, que cumple con las necesidades pero que peca de la necesidad de tener que comprar sus accesorios.

### **2.2.17. Bluetooth Smart Scanner**

Este sistema está desarrollado por Ace Sensor Inc. y necesita de un sistema operativo IOS, como mínimo en la versión 6.0. Además la aplicación es de instalación gratuita y se encuentra en la versión 1.0.5, la cual data de la fecha 11 de Julio de 2014. Además la aplicación se encuentra disponible en inglés y permite la utilización de sensores externos.

### **2.2.18. BlueSense**

Este sistema está desarrollado por BlueRadios, Inc. y necesita de un sistema operativo IOS, como mínimo en la versión 7.0. Además la aplicación es de instalación gratuita y se encuentra en la versión 1.4.1, la cual data de la fecha 23 de Febrero de 2015. Además la aplicación se encuentra disponible en inglés y permite la utilización de sensores externos.

### **2.2.19. Smart Barn**

Este sistema está desarrollado por Andrew Klein y necesita de un sistema operativo IOS, como mínimo en la versión 6.0. Además la aplicación es de instalación gratuita y se encuentra en la versión 1.1.2, la cual data de la fecha 31 de Octubre de 2014. Además la aplicación se encuentra disponible en inglés y permite la utilización de sensores externos.

### **2.2.20. Snap Switch**

Este sistema está desarrollado por Snap Software y necesita de un sistema operativo Windows Phone, como mínimo en la versión 7.5. Además la aplicación es de instalación tras compra por 7,99€y se encuentra en la versión 6.8.223.2317, la cual data de la fecha 24 de Febrero de 2014. Además la aplicación se encuentra disponible en inglés y permite la utilización de sensores externos.

### **2.2.21. HomeWizard**

Este sistema está desarrollado por Rajen y necesita de un sistema operativo Windows Phone, como mínimo en la versión 8.0. Además la aplicación es de instalación gratuita y se encuentra en la versión 1.2.3.0, la cual data de la fecha 21 de Noviembre de 2014. Además la aplicación se encuentra disponible en 2 idiomas y permite la utilización de sensores externos.



Figura 2.9: HomeWizard

- La **ventaja** de la aplicación es una gran funcionalidad para controlar luces, ver vídeo, etc. Además está disponible a través de internet si configuras el router como indica su asistente.
- El **inconveniente** es que aún teniendo bastantes funciones necesitas de sus accesorios (soporta bastantes marcas lo que supone gran variedad).
- En **conclusión** muy buena aplicación pero como las demás, necesita de los accesorios soportados para funcionar, no es tan genérica como lo que se pretende por nosotros.

### 2.2.22. SmartThings Mobile

Este sistema está desarrollado por SmartThings, Inc. y necesita de un sistema operativo Windows Phone, como mínimo en la versión 8.1. Además la aplicación es de instalación gratuita y se encuentra en la versión 201.521.217.242.817, la cual data de la fecha 18 de Febrero de 2014. Además la aplicación se encuentra disponible en inglés y permite la utilización de sensores externos.

### 2.2.23. Band Sensor Monitor

Este sistema está desarrollado por Venetasoft y necesita de un sistema operativo Windows Phone, como mínimo en la versión 8.1. Además la aplicación es de instalación tras compra por 1,99€ y se encuentra en la versión 1.2.5.2, la cual data de la fecha 10 de Febrero de 2014. Además la aplicación se encuentra disponible en inglés y permite la utilización de sensores externos.



Figura 2.10: Band Sensor Monitor

- La **ventaja** de la aplicación es que puedes ver en la pantalla todas las medidas producidas por el accesorio.
- El **inconveniente** es que solamente sirve para dicho accesorio.
- En **conclusión** es una buena aplicación pero, al ser solamente capaz de recibir datos de un dispositivo, se queda muy limitada en comparación a lo que se busca con nuestra aplicación.

#### 2.2.24. INSTEON for Hub

Este sistema está desarrollado por SmartLabs y necesita de un sistema operativo Windows Phone, como mínimo en la versión 8.0. Además la aplicación es de instalación gratuita y se encuentra en la versión 1.0.0.2, la cual data de la fecha 19 de Septiembre de 2014. Además la aplicación se encuentra disponible en inglés y permite la utilización de sensores externos.

#### 2.2.25. Smart Home

Este sistema está desarrollado por Rogers Communications Inc. y necesita de un sistema operativo Windows Phone, como mínimo en la versión 8.0. Además la aplicación es de instalación gratuita y se encuentra en la versión 6.1.31.0, la cual data de la fecha 10 de Diciembre de 2014. Además la aplicación se encuentra disponible en 2 idiomas y permite la utilización de sensores externos.

### 2.2.26. IQONTROL

Este sistema está desarrollado por mediola - connected living AG y necesita de un sistema operativo Windows Phone, como mínimo en la versión 8.0. Además la aplicación es de instalación gratuita y se encuentra en la versión 1.0.6.0, la cual data de la fecha 10 de Marzo de 2014. Además la aplicación se encuentra disponible en 2 idiomas y permite la utilización de sensores externos.



Figura 2.11: IQONTROL

- La **ventaja** de la aplicación es que tiene una gran cantidad de controles variados, hasta dispositivos por infrarrojos.
- El **inconveniente** es que para configurar el dispositivo necesitas una puerta Mediola.
- En **conclusión** es una buena aplicación pero, al necesitar de un accesorio concreto para su configuración, se queda muy limitada en comparación a lo que se busca con nuestra aplicación.

### 2.2.27. Home Controller

Este sistema está desarrollado por Robby Ceyssens y necesita de un sistema operativo Windows Phone, como mínimo en la versión 8.0. Además la aplicación es de instalación gratuita y se encuentra en la versión 2.0.0.0, la cual data de la fecha 20 de Abril de 2014. Además la aplicación se encuentra disponible en 3 idiomas y permite la utilización de sensores externos.

### 2.2.28. I Love Lamp

Este sistema está desarrollado por Sebastian Johansson y necesita de un sistema operativo Windows Phone, como mínimo en la versión 8.0. Además la aplicación es de instalación tras compra por 3,49€ y se encuentra en la versión 1.4.1.0, la cual data de la fecha 24 de Noviembre de 2014. Además la aplicación se encuentra disponible en 4 idiomas y permite la utilización de sensores externos.

## 2.3. Análisis

### 2.3.1. Exposición de características

En la tabla siguiente En este apartado se va a analizar los datos de las aplicaciones similares, en los datos recogidos se puede observar las distintas columnas que hacen referencia a:

- **Nombre Aplicación:** es el nombre oficial de la aplicación la cual sirve para ser encontrada en la tienda de aplicaciones correspondiente a su S.O.
- **Precio:** es la cantidad de dinero que se tiene que pagar para poder instalar la aplicación en nuestro dispositivo móvil.
- **S.O Requerido:** sistema operativo que lleva el dispositivo móvil.
- **S.O min. requerido:** versión minima de dicho sistema que necesita para poder instalar la aplicación.
- **Versión actual app:** versión de la aplicación actual que está disponible para instalar desde la tienda correspondiente a su S.O.
- **Última Actualización:** fecha de la última actualización realizada por el equipo de desarrollo de dicha app.
- **Idioma/s:** idioma en el que se encuentra (en caso de que solamente esté desarrollada en 1) y si no número de idiomas disponibles en el que se encuentra la app.
- **Sensores Externos:** esta columna dice si la aplicación soporta la utilización de sensores ajenos al dispositivo móvil. Puede variar entre 1 ó 0 (Si y No respectivamente).

- **Nº Instalaciones:** número de veces que ha sido descargada la aplicación por los usuarios del S.O correspondiente. Puede no venir especificado por todas las tiendas de aplicaciones.
- **Desarrollador:** dice el nombre de la persona o el equipo que ha desarrollado la app correspondiente.

Nombre Aplicación	Precio	S.O Requerido	S.O min requerido	Versión actual app	Última Actualización
Sensor Kit	Gratis	Android	4.3	1.2.0	3/12/14
Sensor Network for LabVIEW	Gratis	Android	2.1	1.1.2	20/10/13
Wireless Sensor Networks	3,64 €	Android	1.5	1.2.10	20/2/11
Wireless Sensor Location	Gratis	Android	2.1	1	11/5/11
Sensores Multiherramienta	Gratis	Android	4.0	1.0.3	2/2/15
Sensor Ex	Gratis	Android	2.2	1.6.4	6/2/13
Sensors	0,74 €	Android	2.3	1.2	14/4/13
EMX Sensors	Gratis	Android	4.0	12.0	11/7/14
Wally - Home Sensor Network	Gratis	IOS	7.0	1.14	2/2/15
Buscar mi iPhone	Gratis	Android	4.0	2.9	29/1/15
Buscar mi iPhone	Gratis	IOS	8.0	4.0	17/9/14
Home Network System	Gratis	IOS	5.0	1.4	25/2/15
Enviromux-xD	49,99 €	IOS	6.1	1.0	30/1/14
Farm Logix1	Gratis	IOS	4.3	1.0	23/10/11
WeMo	Gratis	IOS	6.0	1.10.2	5/1/15
WeMo	Gratis	Android	4.0	1.10.1	1/12/14
Nest	Gratis	IOS	6.0	4.5.2	24/2/15
Nest	Gratis	Android	2.3.3	4.5.2.3	24/2/15
Bluetooth Smart Scanner	Gratis	IOS	6.0	1.0.5	11/7/14
My Weather Station II	Gratis	IOS	7.1	1.1.1	13/12/14
BlueSense	Gratis	IOS	7.0	1.4.1	23/2/15
Smart Barn	Gratis	IOS	6.0	1.1.2	31/10/14
HomeWizard	Gratis	Windows Phone	8.0	1.2.3.0	21/11/14
Snap Switch	7,99 €	Windows Phone	7.5	6.8.223.2317	24/2/15
Band Sensor Monitor	1,99 €	Windows Phone	8.1	1.2.5.2	10/2/15
SmartThings Mobile	Gratis	Windows Phone	8.1	201.521.217.242.817	18/2/15
INSTEON for Hub	Gratis	Windows Phone	8.0	1.0.0.2	19/9/14
Smart Home	Gratis	Windows Phone	8.0	6.1.31.0	10/12/14
IQONTROL	Gratis	Windows Phone	8.0	1.0.6.0	10/3/14
Home Controller	Gratis	Windows Phone	8.0	2.0.0.0	20/4/14
I Love Lamp	3,49 €	Windows Phone	8.0	1.4.1.0	24/11/14

Tabla 2.1: Exposición características de las aplicaciones analizadas 1

Nombre Aplicación	Idioma/s	Sensores Externos	Nº Instalaciones	Desarrollador
Sensor Kit	Inglés	1	50 - 100	Ace Sensor Inc.
Sensor Network for LabVIEW	Inglés	0	5000 - 10000	SCCT Development team
Wireless Sensor Networks	Inglés	0	50 - 100	O'Reilly Media
Wireless Sensor Location	Inglés	0	1000 - 5000	IT - Universidade de Aveiro
Sensores Multiherramienta	Inglés	0	500 - 1000	Wered Software
Sensor Ex	Inglés	0	5000 - 10000	TarCo
Sensors	Inglés	0	100 - 500	Tom Rice
EMX Sensors	Inglés	0	1000 - 5000	EMX Industries
Wally - Home Sensor Network	Inglés	1		SNUPI Technologies
	Inglés	1	500 - 1000	
Buscar mi iPhone	32 Idiomas	1		iTunes S.a.r.l.
Home Network System	44 Idiomas	0		Panasonic System Networks Co. Ltd.
Enviromux-xD	Inglés	1		Network Technologies Inc
Farm Logix1	Inglés	1		CPF IT CENTER COMPANY LIMITED
WeMo	9 Idiomas	1		Belkin
		1	100000 - 500000	
Nest	5 Idiomas	1		Nest Labs, Inc.
		1	100000 - 500000	
Bluetooth Smart Scanner	Inglés	1		Ace Sensor Inc.
My Weather Station II	3 Idiomas	1		JrYao Chang
BlueSense	Inglés	1		BlueRadios, Inc.
Smart Barn	Inglés	1		Andrew Klein
HomeWizard	2 Idiomas	1		Rajen
Snap Switch	Inglés	1		Snap Software
Band Sensor Monitor	Inglés	1		Venetasoft
SmartThings Mobile	Inglés	1		SmartThings, Inc.
INSTEON for Hub	Inglés	1		SmartLabs
Smart Home	2 Idiomas	1		Rogers Communications Inc.
IQONTROL	2 Idiomas	1		mediola - connected living AG
Home Controller	3 Idiomas	1		Robby Ceyskens
I Love Lamp	4 Idiomas	1		Sebastian Johansson

Tabla 2.2: Exposición características de las aplicaciones analizadas 2

Toda esta información se va a analizar más detalladamente para concretar cuales son las características que debe tener una aplicación de este tipo.

### 2.3.2. Análisis detallado de las características de las aplicaciones

Como se puede observar en las tablas del análisis detallado, la mayoría de las aplicaciones analizadas (más o menos el 80 %) son gratuitas. Además, en cuanto al Sistema Operativo de las aplicaciones hay variedad, hay aplicaciones tanto para Android, como para IOS y Windows Phone y también, en cuanto a la versión mínima del sistema que necesitan.

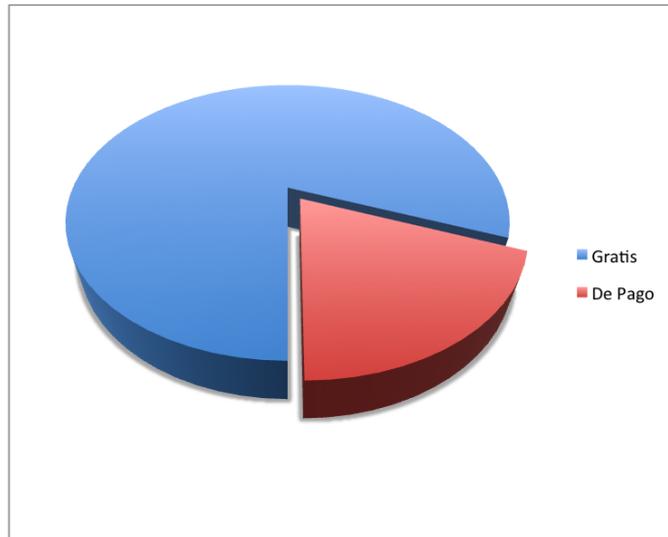


Figura 2.12: Precios apps

En cuanto a las fechas de **última actualización** hay muchas aplicaciones que llevan mucho tiempo sin actualizarse (tanto que no se han actualizado desde el año pasado o hace dos). Además, se puede observar que el tiempo medio que tarda una aplicación en actualizarse varía según el sistema operativo de la misma. De esta forma vemos el tiempo medio de actualizaciones según sistema.

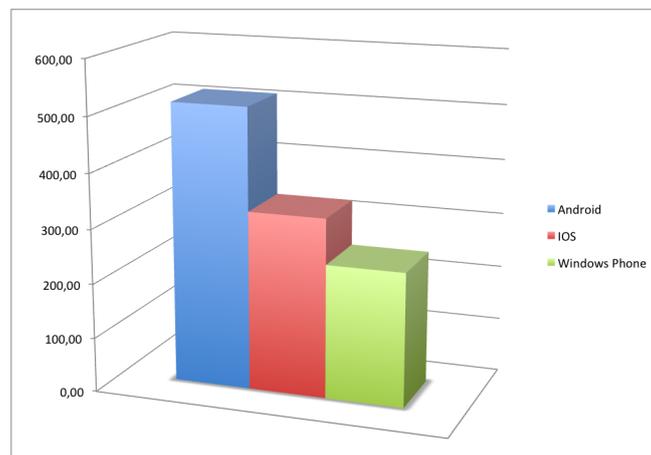


Figura 2.13: Tiempo medio actualización por Sistema Operativo

Además, si únicamente se distingue entre precios de aplicación (gratis o de pago) se observa que el tiempo medio de actualización por las aplicaciones de pago es inferior.

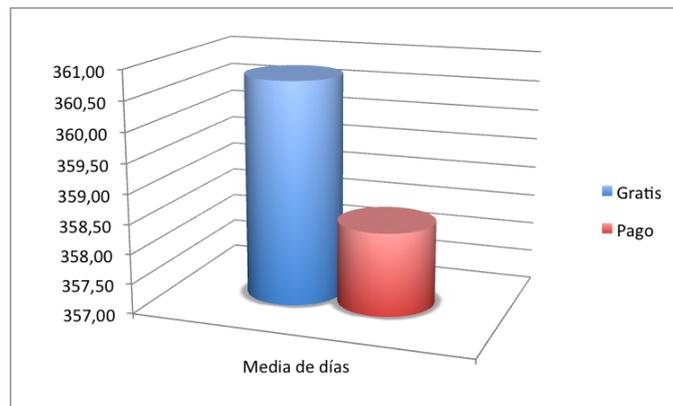


Figura 2.14: Tiempo medio hasta actualización

En cuanto al **idioma** de las aplicaciones observamos que la mayoría de las analizadas (casi el 62 %) se encuentran solamente en inglés, mientras que el 38 % restante tienen varios idiomas o más.

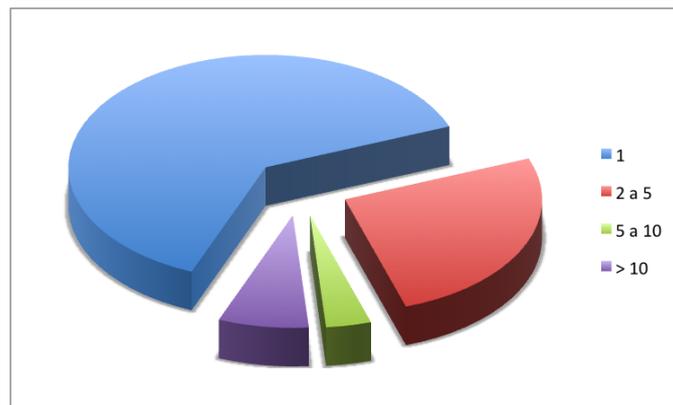


Figura 2.15: Idiomas aplicaciones

La columna de **sensores externos** nos indica si la aplicación permite la utilización de sensores ajenos al dispositivo móvil que la utiliza. Se observa que un 23 % de las aplicaciones no los permiten, solamente se limitan a recoger los datos del dispositivo móvil.

Luego en **Nº Instalaciones** se puede ver las veces que ha sido instalada (en el caso de que se especificara en la tienda). Y Por último vemos el **desarrollador** de la aplicación en cuestión.

### 2.3.3. Análisis de funcionalidades

En la tabla siguiente se puede ver las distintas columnas que hacen referencia a:

- **Nombre Aplicación:** el nombre oficial de la aplicación la cual sirve para ser encontrada en la tienda de aplicaciones correspondiente al S.O.
- **Apariencia:** define como es la aplicación visualmente. Dichos valores variarán entre Atractiva, Sencilla y Pobre.
- **Usabilidad:** hace referencia a como es el funcionamiento de la aplicación cuando la utilizas. Dichos valores variarán entre Eficiente, Sencilla, Agradable y Caótica.
- **Aprendizaje:** da información acerca de el tiempo que tarda un usuario en aprender a manejar dicha aplicación. Los valores pueden variar entre Sencilla y Difícil.
- **Variedad Sensores:** nos dice si la aplicación permite que haya de distintas variedades (puertas, luces, temperatura...). Los valores pueden variar entre 1 y 0 (Sí y No respectivamente).

Nombre Aplicación	Apariencia	Usabilidad	Aprendizaje	Variedad Sensores
Sensor Kit	Atractiva	Eficiente	Sencillo	0
Sensor Network for LabVIEW	Pobre	Sencilla	Sencillo	0
Wireless Sensor Location	Pobre	Sencilla	Sencillo	0
Sensores Multiherramienta	Atractiva	Eficiente	Sencillo	0
SensorEx	Pobre	Caótica	Difícil	0
Sensors	Pobre	Sencilla	Sencillo	0
Wally - Home Sensor Network	Atractiva	Eficiente	Sencillo	0
Buscar mi iPhone	Atractiva	Eficiente	Sencillo	0
Home Network System	Atractiva	Eficiente	Sencillo	1
Enviromux-xD	Pobre	Caótica	Difícil	0
Farm Logix1	Pobre	Sencilla	Sencillo	0
WeMo	Atractiva	Eficiente	Sencillo	1
	Atractiva	Eficiente	Sencillo	1
Nest	Atractiva	Eficiente	Sencillo	1
	Atractiva	Eficiente	Sencillo	1
Bluetooth Smart Scanner	Sencilla	Sencilla	Sencillo	0
My Weather Station II	Atractiva	Agradable	Sencillo	1
BlueSense	Sencilla	Sencilla	Sencillo	1
Smart Barn	Sencilla	Sencilla	Sencillo	1
HomeWizard	Sencilla	Sencilla	Sencillo	1
Snap Switch	Sencilla	Agradable	Sencillo	1
Band Sensor Monitor	Atractiva	Eficiente	Sencillo	1
SmartThings Mobile	Atractiva	Agradable	Sencillo	1
INSTEON for Hub	Atractiva	Agradable	Sencillo	1
Smart Home	Sencilla	Eficiente	Sencillo	1
IQONTROL	Atractiva	Agradable	Sencillo	1
Home Controller	Pobre	Caótica	Difícil	1
I Love Lamp	Atractiva	Sencilla	Sencillo	1

Tabla 2.3: Análisis de funcionalidades

En cuanto al de funcionalidades se comprueba las valoraciones realizadas sobre las aplicaciones anteriores, en las cuales, se ha intentado darle un valor a ciertos aspectos de las mismas. Dichos valores se centran en términos de aprendizaje, de usabilidad, de apariencia y de la variedad de sensores externos que permiten.

En cuanto al aprendizaje en primera instancia, comparar las aplicaciones que nos han resultado sencillas y cuáles difíciles. Para ello nos se ha basado tanto en nuestra opinión como la de los usuarios que han dejado reseñas en las distintas tiendas de aplicaciones.

De esta manera se puede decir que casi el 90 % de aplicaciones (concretamente el 89,66) cuentan con un sencillo. Por otra parte, se ve en oposición que el 10,34 % tienen un difícil aprendizaje.

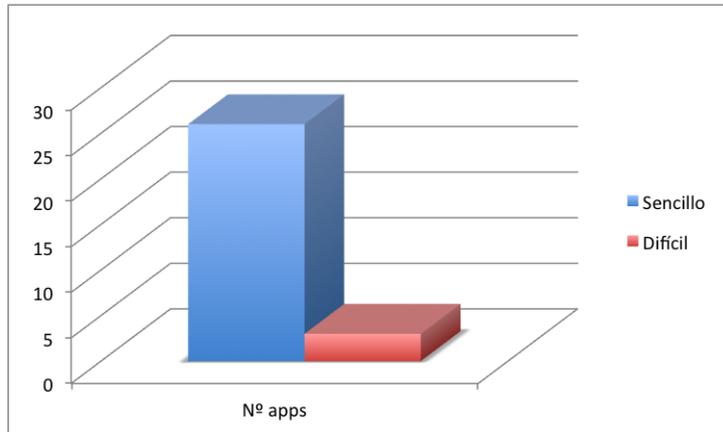


Figura 2.16: Aprendizaje manejo app

Además, en el apartado de la apariencia se observa que:

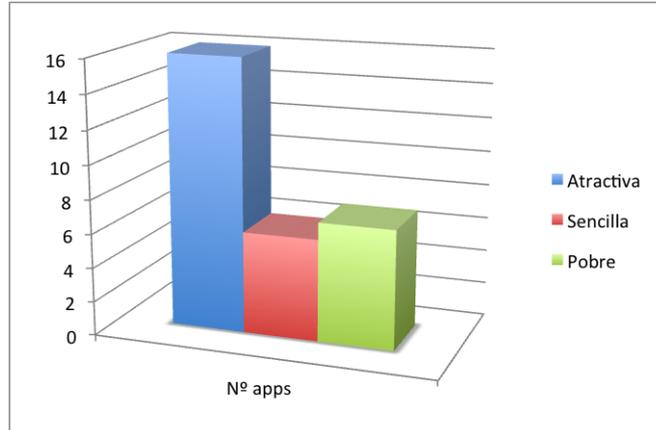


Figura 2.17: Apariencia app

- El 55,17 % son de aspecto atractivo.
- El 20,69 % de aspecto sencillo.
- El 24,14 % restante de aspecto pobre.

Para continuar, se va a valorar la usabilidad de la aplicación (uno de los elementos más importantes). Para ello he dividido dicha valoración en 4 posibles tipos. Dichos tipos son Eficiente, Sencilla, Agradable y Caótica.

Se observa que lo que la gente prioriza a la hora de desarrollar las aplicaciones son la eficiencia y la sencillez. De esta manera se consiguen aplicaciones muy buenas. Según los datos analizados previamente un 41.38 % son Eficientes, un 31.03 % son Sencillas, un 17.24 % son Agradables y un 10.34 % son Caóticas.

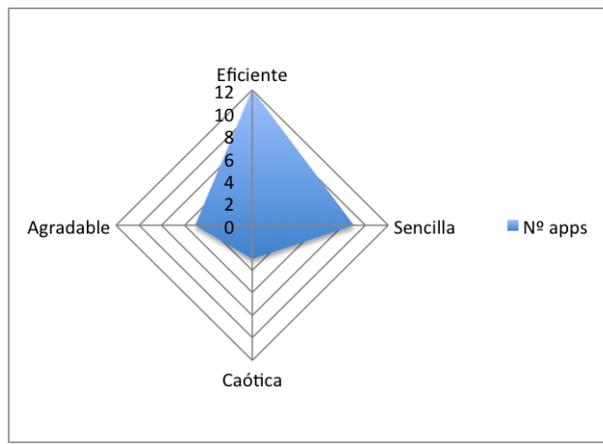


Figura 2.18: Usabilidad app

Y finalmente he pasado a valorar la aplicación desde el punto de vista de la variedad de sensores externos que permiten. Lo que podemos observar es que el 58,62 % permite variedad (puertas, temperatura, luces...), mientras que el 41,38 % no lo permite.

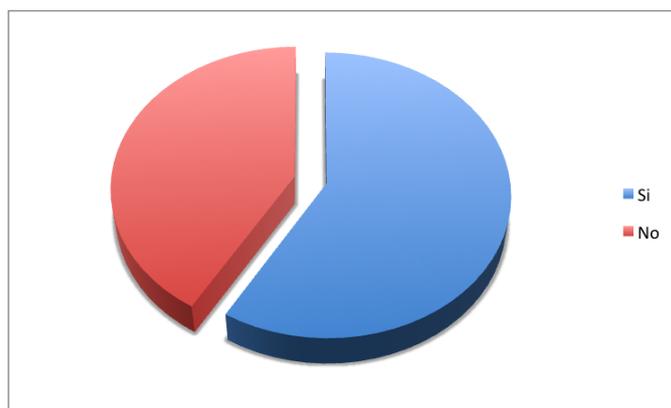


Figura 2.19: Variedad Sensores

## 2.4. Síntesis

Basándonos en el análisis del entorno realizado con anterioridad se puede decir que las características deseadas por cualquier aplicación de similar índole deberían de ser:

- **Idioma:** como se ha podido ver en el análisis anterior la gran mayoría de aplicaciones cuentan solamente con 1 idioma. Por lo que sería conveniente tener, por lo menos, 2 ó 3 idiomas para llegar a una mayor variedad gente.
- **Sistema operativo:** dado que cada sistema operativo tiene un tipo de consumidor, sería conveniente conseguir llevar la aplicación a los 3 sistemas operativos, de este modo se asegura cubrir la mayoría de los tipos de consumidores.
- **Actualizaciones:** el tiempo de actualizaciones debería ser mejor que la media, lo que implicaría, un par de actualizaciones anuales de la aplicación.
- **Sensores externos:** según se ha estudiado, hay muchas aplicaciones que no permiten utilizar sensores de fuera del dispositivo móvil, por lo cual sería algo muy deseable por los usuarios.
- **Variedad sensores externos:** según el análisis anterior, hay muchas aplicaciones que solamente funcionan con los accesorios que te vende la misma marca, permitir variedad hará que muchos clientes puedan converger en esta aplicación y no depender de unos pocos.
- **Manejo:** según se ha estudiado sería conveniente que, como la mayoría de las aplicaciones, se tuviera una curva de aprendizaje sencilla de modo que no lleve casi tiempo poder manejarla.
- **Apariencia:** muchas de las aplicaciones estudiadas anteriormente tenían una apariencia atractiva pero, sin embargo, hay también un gran grupo de interfaces sencillas. De esto se puede decir que la apariencia idea sería unas interfaces sencillas pero atractivas a la vez, creando un equilibrio entre ambas.
- **Usabilidad:** las aplicaciones estudiadas nos han dejado constancia de que son eficientes en su usabilidad, dicha aplicación debería mantener dicha característica.

## 2.5. Conclusiones

Como se ha visto mediante el estudio de los sistemas similares y de su posterior análisis, tanto detalladamente de sus características como de sus funcionalidades, se han dejado clara las necesidades de una aplicación de dicha índole.

A continuación se van a especificar todos los requisitos necesarios (funcionales, de diseño, de usuario y tecnológicos) para que el sistema a desarrollar los cumpla una vez implementado.

# Capítulo 3

## Especificación de requisitos

### 3.1. Introducción

En este apartado se va a especificar los requisitos funcionales, de diseño, de usuario y tecnológicos de la aplicación a desarrollar, para ello, se ha empleado el estándar IEEE 830.

### 3.2. Definiciones

- **Dispositivo:** es un aparato electrónico de tamaño reducido que contiene una serie de sensores y/o actuadores conectados. En el caso de nuestro sistema va a ser una Raspberry Pi.
- **Sensor:** es un pequeño aparato electrónico el cual es capaz de medir una magnitud en una unidad de medida. En nuestro sistema serán de temperatura, de humedad o de posición.
- **Actuador:** es un pequeño aparato electrónico el cual es capaz de variar una magnitud. En nuestro sistema será un emisor de sonido o un led que se encienda.
- **Escritura:** es una estructura generada cuando se quiere comunicar una orden a un actuador a través del servidor.
- **Lectura:** es un valor que es obtenido por un sensor al medir una magnitud en una unidad.
- **Mockup:** boceto de la apariencia de la aplicación con el que se puede hacer una idea de como quedará la misma.

- **Acrónimos:**
  - **BBDD:** Base de Datos.
  - **S.O:** Sistema Operativo.

### 3.3. Requisitos funcionales

Se han establecido una serie de requisitos para la aplicación:

#### 3.3.1. Listado Dispositivos

El sistema permite visualizar todos los dispositivos que hay dados de alta.

- **Entrada:** Pulsación del usuario sobre "Dispositivos".
- **Proceso:** El sistema busca todos los dispositivos que hay y los muestra en pantalla.
- **Salida:** {Id + Id\_dispositivo + Nombre}

#### 3.3.2. Listado de Sensores de un Dispositivo

El sistema permite visualizar todos los sensores que pertenecen a un dispositivo concreto.

- **Entrada:** Pulsación del usuario sobre un "Dispositivo". [Id\_dispositivo]
- **Proceso:** El sistema busca todos los sensores que pertenecen al dispositivo elegido.
- **Salida:** {Id\_sensor + Nombre + Tipo + Id\_dispositivo}

#### 3.3.3. Listado de Sensores

El sistema permite visualizar todos los sensores que pertenecen a un dispositivo concreto.

- **Entrada:** Pulsación del usuario sobre "Sensores".
- **Proceso:** El sistema busca todos los sensores que pertenecen al dispositivo elegido.
- **Salida:** {Id\_sensor + Nombre + Tipo + Id\_dispositivo}

### 3.3.4. Listado Actuadores de un Dispositivo

El sistema permite visualizar todos los actuadores que pertenecen a un dispositivo concreto.

- **Entrada:** Pulsación del usuario sobre ".Actuadores". [Id\_dispositivo]
- **Proceso:** El sistema busca los actuadores que pertenecen a un dispositivo elegido.
- **Salida:** {Id\_actuador+Nombre\_actuador+Id\_dispositivo+Tipo\_actuador}

### 3.3.5. Listado Actuadores

El sistema permite visualizar todos los actuadores.

- **Entrada:** Pulsación del usuario sobre ".Actuadores".
- **Proceso:** El sistema busca todos los actuadores que hay y los muestra en pantalla.
- **Salida:** {Id\_actuador+Nombre\_actuador+Id\_dispositivo+Tipo\_actuador}

### 3.3.6. Gestión Sensor

Permite mostrar la información del sensor y además las lecturas que proporciona.

- **Entrada:** Pulsación del usuario sobre un sensor concreto. [Id\_sensor]
- **Proceso:** El sistema carga la información del sensor y busca las lecturas producidas por dicho sensor.
- **Salida:** {Fecha\_lectura+Hora\_lectura+Medida}

### 3.3.7. Gestión Actuador

Permite interactuar con el actuador seleccionado.

- **Entrada:** Pulsación del usuario sobre un actuador concreto. [Id\_actuador]
- **Proceso:** El sistema deja elegir un valor (según el tipo de actuador) para enviárselo.
- **Salida:** {Id\_actuador+Nombre+Id\_dispositivo+Tipo+Valor\_enviado}

### 3.3.8. Diseño

Se quería hacer una aplicación que fuera:

- Visualmente sencilla, para ello se han elegido una combinación de colores muy suave que no sobrecargue al usuario (en este caso un par de tonos verdes y blanco).
- Fácil de manejar, para ello se han dejado interfaces muy fáciles de conocer y habituarse a ellas.
- Animaciones ligeras para no sobrecargar el funcionamiento.

### 3.3.9. Requisitos de usuario y tecnológicos

- **Requisitos de usuario:** El usuario tipo es aquel que quiera montarse una red de sensores. Las interfaces deben de ser sencillas para en poco tiempo hacerse con el manejo del sistema.
- **Requisitos tecnológicos:** Los requisitos necesarios que debe tener el dispositivo móvil del usuario que va a interactuar con nuestra aplicación deberían ser:
  - **Requisitos mínimos:**
    - Dispositivo móvil con sistema operativo Android.
    - Versión del sistema operativo 4.0.
    - Pantalla del dispositivo de 4".
  - **Requisitos recomendados:**
    - Dispositivo móvil con sistema operativo Android.
    - Versión del sistema operativo 5.0.
    - Pantalla del dispositivo de 5".

# Capítulo 4

## Diseño del sistema

### 4.1. Introducción

A lo largo de este apartado se va a dejar claro como va a ser el Sistema que se va a desarrollar, de modo que se irá perfilando tanto durante los distintos sub-apartados.

### 4.2. Especificación conceptual

Para comenzar, se observa el primer diagrama del Sistema. Dicho diagrama muestra a grandes rasgos la visión global de la aplicación, de modo que, a partir de él se tiene una idea de su funcionamiento.

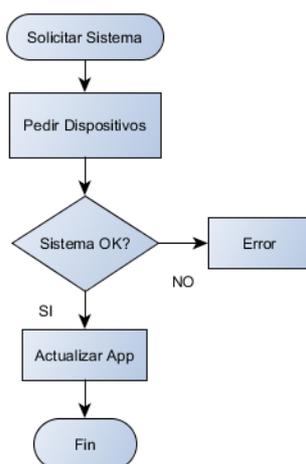


Figura 4.1: Diagrama conceptual 1

Se puede ver que en dicho flujo, cuando inicia la aplicación se pide al sistema los dispositivos, si no hubiera ninguno o diera error, se informará al usuario actualizando la pantalla de la aplicación para que viera que algo ha salido como no se esperaba. Ahora el segundo diagrama:

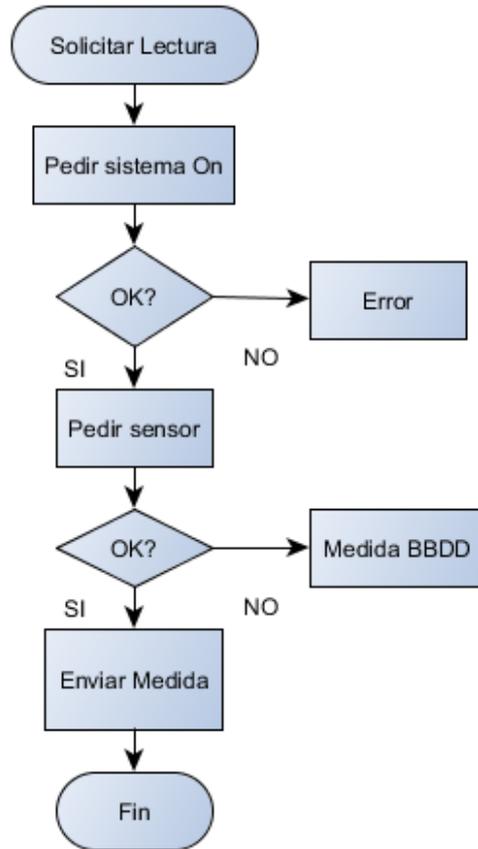


Figura 4.2: Diagrama conceptual 2

En este diagrama se puede ver el proceso que ocurre para solicitar las lecturas de un sensor, lo primero que se hace es comprobar que el sistema funciona para, a continuación, pedir el sensor y al mismo, solicitarle las lecturas, si no hubiesen lecturas que sean proporcionadas por el sensor directamente, se consultaría la última que se tiene en la BBDD y esta sería la enviada.

Y el tercer diagrama de flujo:

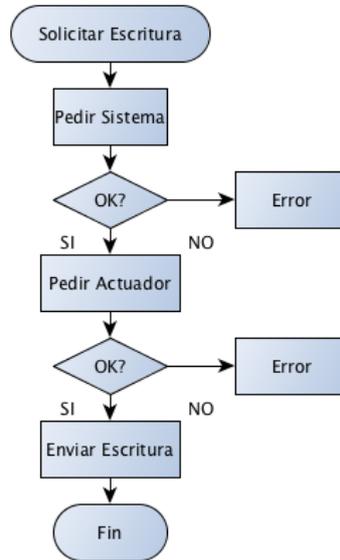


Figura 4.3: Diagrama conceptual 3

Este tercer diagrama habla de la opción contraria a la anterior, antes se pedía un dato al sensor y ahora se envía una orden al actuador. Tras comprobar previamente de que se tienen los dispositivos, se pedirá el actuador para, si no está disponible, avisar al usuario y actualizar la app. Si el actuador está disponible se le envía una orden y se actualiza la app.

## 4.3. Especificación formal

### 4.3.1. Capa persistencia

Para entender que tipo de dato se va a utilizar se necesita saber:

- **INT:** Dato numérico que puede representar un subconjunto finito de los números enteros.
- **STRING:** Cadena de caracteres, palabras, ristras de caracteres o frase.
- **DATETIME:** Formato de datos de tipo fecha.
- **LIST:** Lista que contiene otros objetos (en el caso de la aplicación Sensores o Actuadores).

- **NN:** Hace referencia a que el valor especificado junto al que va, no podrá ser nulo.

Se puede ver ahora que las estructuras para los objetos de la aplicación son las siguientes:

- **Dispositivo**

- Id INT (NN)
- Id-Dispositivo INT (NN)
- Nombre-Sensor STRING
- Lista-Sensores LIST
- Lista-Actuadores LIST

- **Sensor**

- Id-Sensor INT (NN)
- Id-Dispositivo INT (NN)
- Nombre-Sensor STRING
- Tipo-Sensor INT (NN)
- Id-Magnitud INT (NN)
- Id-Unidad INT (NN)

- **Actuador**

- Id-Actuador INT (NN)
- Id-Dispositivo INT (NN)
- Nombre-Actuador STRING
- Tipo-Actuador INT (NN)

- **Lectura**

- Id INT (NN)
- Id-Dispositivo INT (NN)
- Id-Sensor INT (NN)
- Fecha-Lectura DATETIME
- Valor-Lectura INT

#### ■ Escritura

- Id-Dispositivo INT (NN)
- Id-Actuador INT (NN)
- Valor-Escribir INT
- Fecha-Solicitud DATETIME

### 4.3.2. Capa negocio

En este apartado se procede a describir los posibles casos que puedan ocurrir al utilizar la aplicación por un usuario.

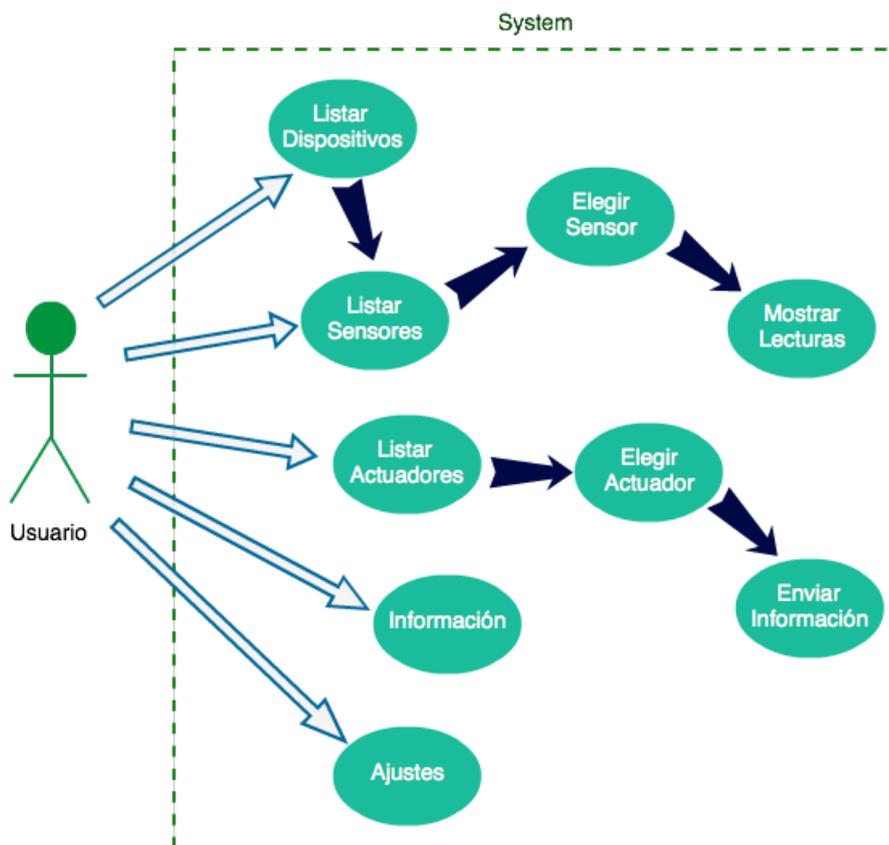


Figura 4.4: Casos de uso

### 4.3.3. Capa presentación

En este apartado se va a explicar los distintos mockups que he creado para hacer una idea de como quedará la aplicación una vez implementada más adelante.

Para empezar se muestra la pantalla de inicio:



Figura 4.5: Mockup de Inicio de la aplicación

Desde esta primera pantalla se puede acceder a los distintos sub-apartados de la misma. Por lo tanto se puede:

- En la barra superior hay acceso tanto a la información de la aplicación (autor, imagen y título de la aplicación) como a la pantalla de ajustes.
- En el centro acceso directo a los listados de Dispositivos, Sensores y Actuadores.
- En la parte inferior la barra de navegación del S.O.

El siguiente mockup corresponde al listado de dispositivos:

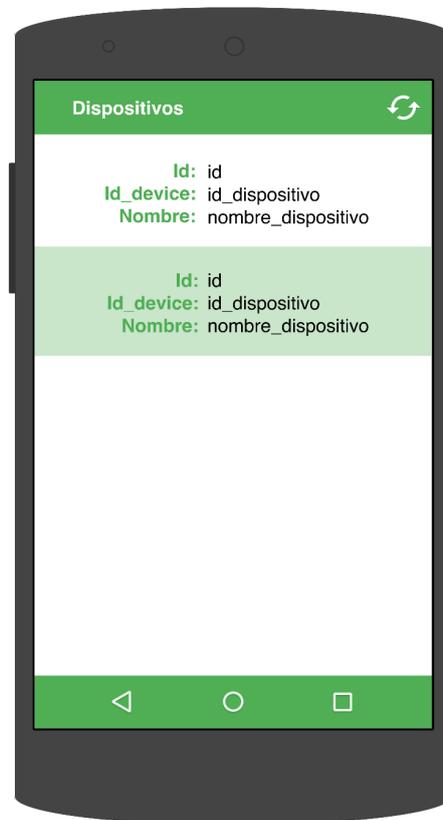


Figura 4.6: Mockup de Listado Dispositivos

- En la barra superior hay acceso a la acción de volver a la página de inicio.
- En el centro está el listado de los distintos dispositivos que hayan conectados.
- En la parte inferior la barra de navegación del S.O.

El siguiente mockup corresponde al listado de Sensores:

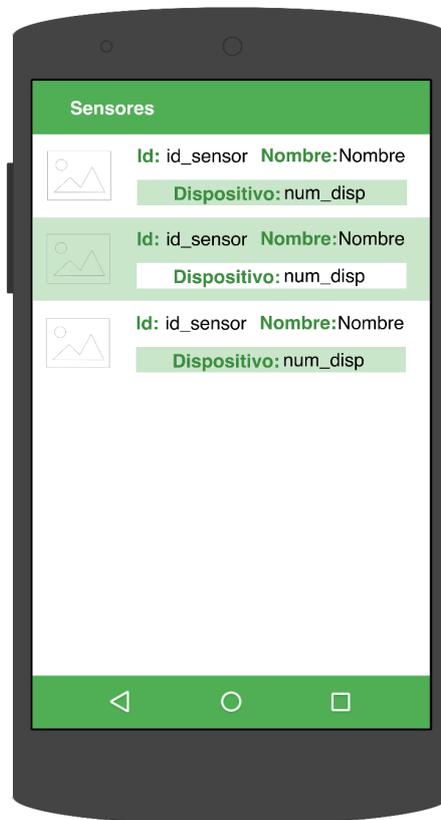


Figura 4.7: Mockup de Listado Sensores

- En la barra superior hay acceso a la acción de volver a la página de inicio.
- En el centro está el listado de los distintos dispositivos que hayan conectados, dicho listado contendrá el id del sensor, su nombre, el dispositivo al que pertenece y la imagen del tipo de sensor para qué, con solo un vistazo se pueda saber de que tipo es (temperatura, humedad, posición,etc.).
- En la parte inferior la barra de navegación del S.O.

Además, a éste listado de Sensores se puede llegar desde el listado de Dispositivos, para ello solamente se tiene que pulsar encima de uno de los dispositivos y se abrirá el listado de Sensores pero sólo se cargarán los que están en dicho dispositivo.

El siguiente mockup corresponde a la interfaz de un Sensor:



Figura 4.8: Gestión Sensor

- En la barra superior se tiene acceso a la acción de volver a la página del listado de Sensores.
- En el centro hay:
  - En la parte superior, como en el listado de Sensores, está la información del Sensor elegido pero ampliando la anterior información: con la magnitud y la unidad de dicho sensor.
  - En la parte inferior las medidas realizadas por dicho sensor, las cuales se mostrarán por un listado donde pondrá: fecha, hora y medida.
- En la parte inferior la barra de navegación del S.O.

El siguiente mockup corresponde a la interfaz del listado de Actuadores:



Figura 4.9: Listado Actuadores

- En la barra superior se tendrá un acceso a la acción de volver a la página del Inicio de la aplicación.
- En el centro habrá un listado de los distintos Actuadores, se mostrarán: su id, su nombre, el número del dispositivo al que pertenece y además una imagen del tipo de actuador.
- En la parte inferior la barra de navegación del S.O.

Se puede observar que el listado es igual al de Sensores, pero cambia en cuanto a la información con la que trabajan.

El siguiente mockup corresponde a la interfaz de un Actuador:

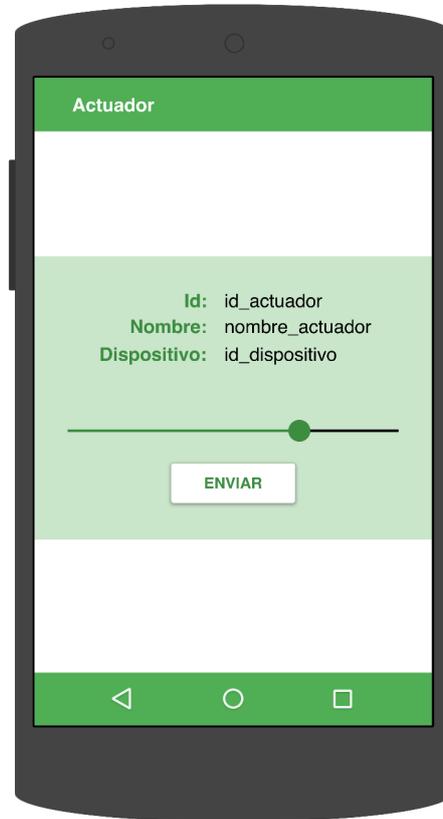


Figura 4.10: Gestión Actuador

- En la barra superior hay acceso a la acción de volver a la página del listado de Actuadores.
- En el centro se tiene la información relativa a el actuador seleccionado en el listado anterior la cual es: el id del actuador, el nombre, el dispositivo al que pertenece, una barra deslizante y un botón de enviar. Dicha barra será la que se utilizará para enviar al actuador el dato ya sea digital (se enviará 0 ó 1) o analógico (0 a 100).
- En la parte inferior la barra de navegación del S.O.

El siguiente mockup corresponde a la interfaz de Información:



Figura 4.11: Pantalla Información

- En la barra superior se tiene acceso a la acción de volver a la página del listado de Actuadores.
- En el centro está la información relativa a la app, es decir, el título de la misma con la foto y el nombre del Autor.
- En la parte inferior la barra de navegación del S.O.

## 4.4. Conclusiones

Como conclusión, se tiene una clara idea de lo que hay que implementar e implantar en la siguiente fase del desarrollo, ya que se ha dejado bien marcado su diseño previo. En el siguiente apartado se muestra la aplicación mostrando la capa de persistencia convertida ya en ficheros y además, capturas de pantalla y trozos de código ya programado.

# Capítulo 5

## Implementación e implantación

### 5.1. Introducción

En este apartado se va a mostrar claro como la aplicación ha tomado forma gracias al capítulo anterior. Para ello se muestra como ha evolucionado la capa de persistencia en los ficheros necesarios, la capa de negocio en código de programación y los mockups en interfaces de nuestra aplicación.

### 5.2. Implementación

#### 5.2.1. Capa persistencia

Para mostrar la capa de persistencia, se observan las estructuras de los objetos que se utilizarán cuando se reciben los datos por parte del servidor al cual se le solicitan, o bien, cuando se le envían datos al mismo a la hora de dar órdenes a los actuadores.

Un ejemplo de la estructura de los dispositivos es ésta:

```
[
  {
    "id":"0001",
    "id_device":"0110",
    "device_name":"Dispositivo 1",
    "sensores":{
    },
    "actuadores":{
    }
  }
]
```

Figura 5.1: JSON lista de Dispositivos

En el ejemplo de la imagen anterior se puede ver que contiene los datos del propio dispositivo y además, dos listas que contienen a los sensores y actuadores que pertenecen al dispositivo. Así que se va mostrar cada una de esas listas, las cuales son:

```
[
  {
    "id_sensor": "0001",
    "id_device": "0110",
    "sensor_name": "Peter",
    "id_type": "01",
    "magnitudo": "1",
    "unit": "C°"
  },
  {
    "id_sensor": "0002",
    "id_device": "0110",
    "sensor_name": "Pan",
    "id_type": "02",
    "magnitudo": "3",
    "unit": "Coords"
  }
]
```

Figura 5.2: JSON lista de Sensores

En esta lista estarán los distintos identificadores necesarios para el Sensor (id propio del sensor y del dispositivo al que pertenece), así como el nombre de dicho sensor. Además, habrá un identificador referente al tipo de datos, dicho id viene con una serie de tipos establecidos fijos, según dicho tipo se le mostrará un icono u otro. Será necesario también que tenga una magnitud y una unidad en la que se estará haciendo la toma de medidas por dicho sensor.

En la imagen correspondiente a la lista siguiente se verá un ejemplo de como se estructurarán los Actuadores en el formato JSON.

```
[
  {
    "id_act": "0001",
    "id_device": "0110",
    "act_name": "Bocina",
    "id_type": "01"
  },
  {
    "id_act": "0002",
    "id_device": "0110",
    "act_name": "Luz LED",
    "id_type": "02"
  }
]
```

Figura 5.3: JSON lista de Actuadores

Ahora se va a mostrar un ejemplo de como sería los datos en formato JSON referentes a un dispositivo con dos sensores y dos actuadores.

```
[
  {
    "id": "0001",
    "id_device": "0110",
    "device_name": "Dispositivo 1",
    "sensores": [
      {
        "id_sensor": "0001",
        "id_device": "0110",
        "act_name": "Peter",
        "id_type": "01",
        "magnitudo": "1",
        "unit": "C°"
      },
      {
        "id_sensor": "0001",
        "id_device": "0110",
        "act_name": "Pan",
        "id_type": "02",
        "magnitudo": "3",
        "unit": "Coords"
      }
    ],
    "actuadores": [
      {
        "id_act": "0001",
        "id_device": "0110",
        "act_name": "Bocina",
        "id_type": "01"
      },
      {
        "id_act": "0002",
        "id_device": "0110",
        "act_name": "Luz LED",
        "id_type": "02"
      }
    ]
  }
]
```

Figura 5.4: JSON Dispositivo completo

### 5.2.2. Capa negocio

En este apartado se muestra distintos trozos de código explicándolos brevemente.

```
private void cargaInterfaz() {
    SensorsAdapter adapter = new SensorsAdapter(this, lista);

    Dashboard.setSensors(lista);
    final ListView list = (ListView) findViewById(R.id.listView);
    list.setAdapter(adapter);
    list.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> arg0, View arg1, int position, long arg3) {
            Sensor sen = (Sensor) lista.get(position);
            int c = sen.getId_sensor();
            String id = "Id_sensor: " + c ;

            Toast.makeText(getApplicationContext(), id, Toast.LENGTH_SHORT).show();
            Intent i = new Intent(list.getContext(), tfg.guijarro.sergio.agss.Sensor.class);

            i.putExtra("id_sensor", c);
            startActivity(i);
            overridePendingTransition(R.anim.left_in, R.anim.left_out);
        } //Fin onItemClick()
    });
} //Fin cargaInterfaz()
```

Figura 5.5: Código .java Sensores

En este primer código mostrado se realiza la función de rellenar el listado donde se mostrarán todos los sensores cargados. Para ello se ha creado un adaptador previamente (dicho adaptador recoge los datos de cada elemento de la lista y los establece en el adaptador).

Además, se crea en cada uno de los elementos una llamada a la actividad del Sensor donde se gestiona individualmente. Para dicha llamada se le pasa como extra un Id con el del Sensor pulsado y se muestra un Toast (mensaje por pantalla) indicándolo.

Como añadido también se modifica la transición por defecto para producir un efecto de deslizamiento.

### 5.2.3. Capa presentación

En este apartado se presentan las distintas capturas de pantalla que se han sacado de la aplicación funcionando.

Esta primera captura corresponde a la interfaz de Inicio:



Figura 5.6: Captura pantalla de Inicio

Desde esta primera pantalla se accede a los distintos sub-apartados de la misma. Se podría por lo tanto:

- En la barra superior hay acceso tanto a la información de la aplicación (autor, imagen y título de la aplicación) como a la pantalla de ajustes.
- En el centro están los accesos directos a los listados de Dispositivos, Sensores y Actuadores.
- En la parte inferior la barra de navegación del S.O.

Esta pantalla corresponde al Listado de Sensores:

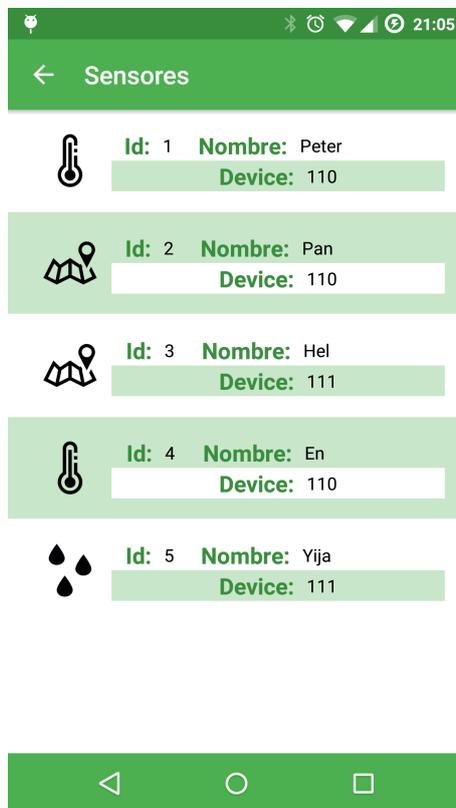


Figura 5.7: Captura pantalla de Sensores

- En la barra superior hay acceso a la acción de volver a la página de inicio.
- En el centro está el listado de los distintos dispositivos que hayan conec-tados, dicho listado contiene el id del sensor, su nombre, el dispositivo al que pertenece y la imagen del tipo de sensor para qué, con solo un vistazo se sabe de que tipo es (temperatura, humedad, posición,etc.).
- En la parte inferior la barra de navegación del S.O.

Además, a este listado de Sensores se puede llegar desde el listado de Dispositivos, para ello solamente se tiene que pulsar encima de uno de los dispositivos y se abrirá el listado de Sensores pero sólo se cargarán los que están en dicho dispositivo.

Como se puede ver a diferencia del mockup aquí se tienen datos reales.

En esta captura se visualiza la pantalla del Actuador:



Figura 5.8: Captura pantalla de Actuadores

- En la barra superior hay acceso a la acción de volver a la página del listado de Actuadores.
- En el centro está la información relativa a el actuador seleccionado en el listado anterior la cual es: el id del actuador, el nombre, el dispositivo al que pertenece, una barra deslizante y un botón de enviar. Dicha barra es la que se utiliza para enviar al actuador el dato, ya sea digital (se enviará 0 ó 1) ó analógico (0 a 100).
- En la parte inferior la barra de navegación del S.O.

Al igual que en la captura anterior, se ve que aquí a diferencia del mockup utiliza datos reales.

La siguiente captura de pantalla corresponde a la interfaz de Información:



Figura 5.9: Captura pantalla de Información

- En la barra superior hay acceso a la acción de volver a la página de Inicio.
- En el centro está la información relativa a la app, se puede ver la foto del autor, el nombre y además, su título.
- En la parte inferior la barra de navegación del S.O.

En la siguiente captura se muestra la interfaz de Ajustes:



Figura 5.10: Captura pantalla de Ajustes

- En la barra superior hay acceso a la acción de volver a la página de Inicio.
- En el centro está el campo de texto donde configurar la IP pública de nuestro servidor utilizado. Esta información es almacenada y se utiliza en cualquier otra clase donde hace falta comunicarse con el servidor.
- En la parte inferior la barra de navegación del S.O.

### 5.3. Evaluación

En este apartado se van a evaluar los casos de uso ya en la aplicación funcionando y, se determina si la salida producida era la esperada.

- **Listar Dispositivos:** se han pasado los parámetros adecuados y se ha obtenido el resultado esperado, lo cual lleva a que carga los dispositivos con todos sus sensores y actuadores, y los lista correctamente.
- **Listar Sensores:** se han pasado los parámetros correctos y la salida producida correspondía con lo que tenía que salir por pantalla, además, si se llega del listado de dispositivos se cargan solamente los de dicho dispositivo.
- **Elegir Sensor:** se han pasado los parámetros correctos y la salida producida era correcta, esto quiere decir que, cuando se elige un sensor de la lista abre dicho sensor y carga su información correctamente.
- **Mostrar Lecturas:** se han pasado los datos correctos y el resultado es el correcto, lo cual quiere decir que, pasando un id del sensor se obtienen correctamente sus medidas producidas.
- **Listar Actuadores:** se han pasado los datos correctos y la salida producida era la correcta, lo que quiere decir que, se cargan todos los actuadores y se muestran correctamente.
- **Elegir Actuator:** se han pasado los datos correctos y el resultado producido era el esperado, lo que quiere decir que, al pinchar sobre un actuator de la lista se abría correctamente y cargaba sus datos.
- **Enviar Información:** se han pasado los datos correctos y el resultado producido era el esperado.
- **Información:** para dicha ejecución no se necesitaban datos, solamente la carga de la interfaz y el resultado producido era el esperado.
- **Ajustes:** se han pasado los datos correctos y el resultado era el esperado.

## 5.4. Implantación

En este apartado se describe el proceso por el cual se implanta la aplicación (en nuestro caso, se instala en el dispositivo móvil).

Debido a que la app no está en la tienda oficial, no se la puede buscar directamente en ella para instalarla así que, se debe tener el “.apk” . Una vez con el fichero “.apk” descargado en nuestro dispositivo móvil, se debe de permitir instalación desde los orígenes desconocidos.

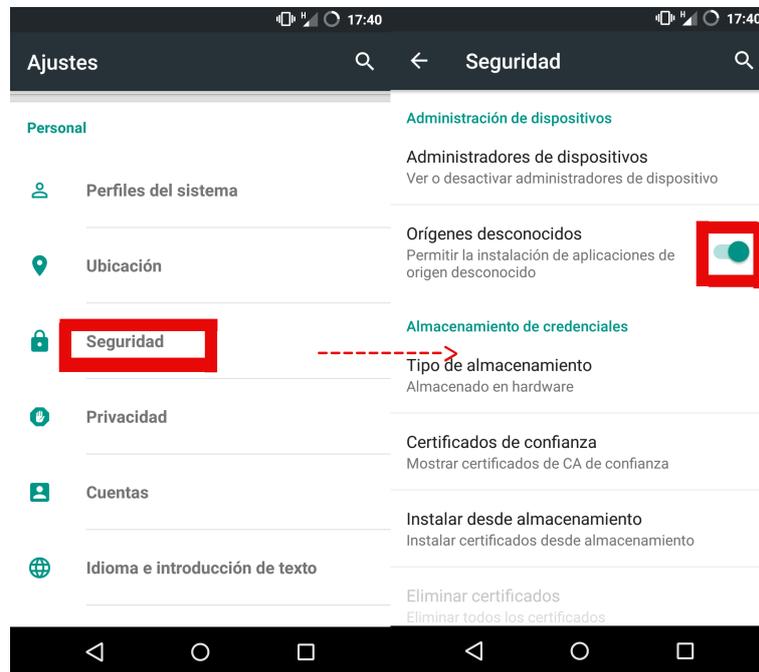


Figura 5.11: Ajustes orígenes desconocidos

Cuando se tiene activada esta opción, abrir el archivo “.apk” para su instalación. Después de esto, la aplicación ya estará implantada completamente.

## 5.5. Conclusiones

Se puede concluir teniendo una clara representación de la funcionalidad de la aplicación desarrollada, gracias a los diagramas anteriores junto a las capturas de pantalla y los trozos de código.

En el siguiente apartado se sacan las conclusiones finales.

# Capítulo 6

## Conclusiones

### 6.1. Dificultades encontradas

Para hablar de las dificultades que he encontrado durante la realización del proyecto hay que poner en antecedentes, que el mismo, tiene mucho en común entre otros proyectos coordinados por el mismo profesor. Es por eso, que se han realizado un dos reuniones semanales para, entre todos, sentar unas bases comunes de trabajo. Gracias a dichas bases, se ha quedado en una estructura común para la comunicación entre el servidor, el cliente y los sensores.

Por otra parte, están las dificultades encontradas en el desarrollo puramente dicho en el cual producía errores por los cuales, muy de vez en cuando, tocaba desinstalar la aplicación y volver a ejecutarla para que la mostrara sin error. Además, al principio tuve dificultades con los estilos de la aplicación. Tras mucho modificarlos y que no hubiera cambio alguno en la aplicación, resultó que a partir de la API utilizada tenía 2 archivos de estilos y utilizaba el de la versión que no correspondía a la versión del S.O de mi dispositivo móvil.

Después de dicho error, tuve problemas con la visualización de las listas de manera correcta debido a que se muestran una fila de cada color de fondo, al final tuve que crear para poder hacerlo de esta manera una interfaz distinta con el color de fondo nuevo y aplicarlo una interfaz o la otra según si la fila era par o impar.

Por último, la dificultad relativa a adaptar la pantalla lo mejor posible a varios tamaños, testeando con varios dispositivos y versiones del S.O. También se ha intentado adaptar la interfaz a la orientación del dispositivo, para que al rotar la pantalla no quede mal la visualización.

## 6.2. Aportaciones

Las aportaciones del proyecto son las siguientes:

1. En el capítulo 2 se proporciona un informe de análisis de aplicaciones similares en el mercado actual para, poder desarrollar proyectos en el ámbito del proyecto presentado.
2. En los capítulos 2 y 3 se proporciona un ejemplo de especificación de requisitos y de esquemas formales para proyectos similares.
3. Del capítulo 4 se proporciona la principal aportación, la aplicación desarrollada que, cumple con los requisitos establecidos y es completamente funcional.

## 6.3. Ampliaciones

Para futuras ampliaciones se podría incorporar:

- Más estilos para la aplicación de modo que el usuario pueda elegir como la quiere visualizar.
- Posibilidad de enlazar sensores y actuadores para que, usando las medidas del primero, condicionar a que actúe uno de los segundos.

# Capítulo 7

## Referencias

- **“Desarrollo De Aplicaciones Para Android”**. Autor: Joan Ribas Lequerica. Editor: Anaya Multimedia. Año: 2015. ISBN: 978-8441536807.
- **“EL GRAN LIBRO DE ANDROID 4ªEd”**. Autor: Jesús Tomás Gironés. Editor: Marcombo. Año: 2013. ISBN: 978-8426719768.
- **“Aprende a programar con Java”**. Autores: Alfonso Jiménez Marín y Francisco Manuel Pérez Montes. Editor: Ediciones Paraninfo, S.A. Año: 2012. ISBN: 978-8497321303.
- **“The Internet of Things (MIT Press Essential Knowledge)”**. Autor: Samuel Greengard. Editor: The Mit Press. Año: 2015. ISBN: 978-0262527736.
- **“Beginning JSON”**. Autor: Ben Smith. Editor: Apress. Año: 2015. ISBN: 978-1484202036.
- **“Android”** URL: “<https://www.android.com>”. Palabra clave: Android. Nombre empresa: Google. Fecha visualización: 28/05/2015.
- **“Internet de las cosas. La tecnología revolucionaria que todo lo conecta”**. Autores: Adrian McEwen y Hakim Cassimally. Editor: Anaya Multimedia. Año: 2014. ISBN: 978-8441536111.
- **“Domótica para ingenieros”**. Autor: J.M Maestre. Editor: Paraninfo. Año: 2015. ISBN: 978-8497329767.
- **“Getting Started - Android”** URL: <https://developer.android.com/training/>. Nombre empresa: Google. Fecha visualización: Junio 2015.

- **“ECMA-404 The JSON Data Interchange Standard”**. URL: <http://json.org>. **Palabra clave:** JSON. **Fecha visualización:** Junio 2015.
- **“Diseñando apps para móviles”**. **Autores:** Javier Cuello y José Vittone. **Editor:** Createspace. **Año:** 2014. **ISBN:** 978-1495454448.
- **“Apps para dispositivos móviles: casos de estudio”**. **Autor:** Rob Ford. **Editor:** Taschen. **Año:** 2011. **ISBN:** 978-3836528832.