

# PORTADA





Universidad de Jaén  
Escuela Politécnica Superior de Jaén  
Departamento de Informática

Dña. Macarena Espinilla Estévez, tutora del Trabajo Fin de Máster titulado: MONITORIZACIÓN DE UN AMBIENTE INTELIGENTE DE BAJO COSTE, que presenta Daniel Esteban Martínez Martínez, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, 2 de marzo de 2017

El alumno:

Los tutores:

Daniel Esteban Martínez Martínez

Macarena Espinilla Estévez



### *Agradecimientos*

*Me gustaría agradecer a todas las personas que me han ayudado, interesado y apoyado antes de desarrollar este proyecto y durante el transcurso del mismo:*

*A mí tutora, Macarena Espinilla Estévez, por la ayuda que me ha prestado cuando lo he necesitado resolviéndome las dudas, y/o aportándome otras posibles soluciones.*

*A toda mi familia por apoyarme, por cuidarme y animarme en los momentos en los que me atrancaba y por estar pendiente de la evolución del proyecto.*

*Y a mis compañeros y amigos que han estado apoyándome durante la realización de este proyecto y con los que he podido contar en todo momento.*



# ÍNDICE

1	Introducción .....	13
1.1	Motivación .....	14
1.1.1	Ambientes inteligentes y envejecimiento de la población .....	15
1.1.2	Sistema de reconocimiento de actividades .....	16
1.2	Propuesta .....	19
1.3	Objetivos .....	20
1.4	Estructura de la memoria .....	20
1.5	Planificación .....	21
1.5.1	Especificación de tareas .....	22
2	Sistema de sensores de bajo coste .....	25
2.1	Introducción .....	26
2.2	Estudio de mercado .....	28
2.3	Configuración de la red de sensores para el apartamento de inteligencia ambiental de la Universidad de Jaén .....	40
2.3.1	Raspberry Pi .....	41
2.3.2	Red de sensores y localización .....	43
2.4	Presupuesto .....	45
3	Reconocimiento de actividades basado en sensores .....	48
3.1	Introducción .....	49
3.1.1	Algoritmos de generación de prototipos .....	51
3.1.2	Enfoque vecino más cercano (NN) .....	53
3.2	Experimentación para el modelado de reconocimiento de actividades .....	54
3.2.1	Experimentación con distintos algoritmos de generación de prototipos .....	58
3.2.2	Elección del algoritmo de GP para el ambiente inteligente .....	75

4	Aplicación móvil de monitorización de ambientes inteligentes .....	78
4.1	Contexto de la aplicación móvil .....	79
4.2	Especificación de requisitos .....	80
4.2.1	Requisitos funcionales .....	81
4.2.2	Requisitos no funcionales .....	82
4.3	Análisis del sistema .....	84
4.3.1	Modelos de caso de uso.....	85
4.3.2	Escenarios .....	92
4.4	Diseño del sistema .....	98
4.4.1	Diseño de los datos .....	98
4.4.2	Diseño de la interfaz .....	105
4.5	Implementación del sistema.....	111
4.5.1	Arquitectura de la aplicación .....	111
4.5.2	Lenguajes de programación .....	112
4.5.3	Entornos de programación .....	114
4.5.4	Servidor web .....	115
4.5.5	Aplicación móvil desarrollada en Android .....	119
4.5.6	Desarrollo de la implementación del sistema.....	121
5	Conclusiones .....	124
5.1	Conclusiones personales.....	126
5.2	Trabajos futuros.....	127
6	Manual de instalación.....	130
A.1	Raspberry pi .....	131
A.1.1	Puesta en marcha.....	131
A.1.2	Instalación del módulo RaZberry .....	132



A.1.3 Vinculación de los sensores .....	133
A.2 Instalación del servidor web .....	136
7 Manual de usuario .....	142
8 Bibliografía y referencias .....	152

## ÍNDICE DE ILUSTRACIONES

Ilustración 1 - Diagrama de Gantt .....	23
Ilustración 2 - Sensor de contacto de Fibaro .....	31
Ilustración 3 - Sensor de contacto de Everspring .....	31
Ilustración 4 - Sensor de contacto de Aeon Labs .....	32
Ilustración 5 - Sensor de contacto de Zipato .....	33
Ilustración 6 - Sensor de movimiento de Fibaro .....	34
Ilustración 7 - Sensor de movimiento de Everspring .....	35
Ilustración 8 - Sensor de presión de Arun .....	35
Ilustración 9 - Controlador Home Center de Fibaro .....	36
Ilustración 10 - Controlador de Vera .....	37
Ilustración 11 - Controlador de Eedomus .....	38
Ilustración 12 - Módulo controlador RaZberry para Raspberry Pi .....	39
Ilustración 13 - Módulo acoplado a la Raspberry Pi .....	39
Ilustración 14 - Raspberry Pi .....	43
Ilustración 15 - Esquema del sistema .....	80
Ilustración 16 - Diagrama frontera .....	86
Ilustración 17 - Esquema conceptual .....	101
Ilustración 18 - Esquema resultante .....	102
Ilustración 19 - Guía de estilos del menú lateral .....	107
Ilustración 20 - Guía de estilos de las listas .....	108
Ilustración 21 - Guía de estilo de los botones .....	108
Ilustración 22 - Guía de estilos de la barra de herramientas .....	108

Ilustración 23 - IDE Notepad++...	110
Ilustración 24 - IDE Notepad++	114
Ilustración 25 - IDE Spyder	115
Ilustración 26 - Android Studio	115
Ilustración 27 - Esquema de FCM	121

## ÍNDICE DE ILUSTRACIONES DE ANEXOS

Ilustración A. 1 – Win32 Disk Imager	131
Ilustración A. 2 - Conexiones GPIO	132
Ilustración A. 3 - Instalación del módulo en los GPIO	132
Ilustración A. 4 - Login del software	133
Ilustración A. 5 - Página principal	134
Ilustración A. 6 - Sensores vinculados	134
Ilustración A. 7 - Ventana de inclusión de un sensor	135
Ilustración A. 8 - Vista principal de instalación de XAMPP	136
Ilustración A. 9 - Selección de los componentes	137
Ilustración A. 10 - Selección de la carpeta de instalación	137
Ilustración A. 11 - Ventana de información	138
Ilustración A. 12 - Ventana previa a la instalación	138
Ilustración A. 13 - Proceso de instalación	139
Ilustración A. 14 - Finalización de la instalación	139
Ilustración A. 15 - Panel de control de XAMPP	140
Ilustración A. 16 - Pantalla principal	143
Ilustración A. 17 - Menú de la aplicación	145
Ilustración A. 18 - Pantalla de Histórico	146
Ilustración A. 19 - Listado de opciones de ordenación	147
Ilustración A. 20 - Listado de opciones de filtrado	148
Ilustración A. 21 - Ventana de selección de fecha	148
Ilustración A. 22 - Pantalla de actividades	149

Ilustración A. 23 - Pantalla de notificaciones..... 150  
Ilustración A. 24 - Pantalla de Acerca de ..... 151



# CAPÍTULO 1

## INTRODUCCIÓN

---

## 1.1 MOTIVACIÓN

---

Hoy en día estamos en pleno auge de tecnologías que hasta hace unos años considerábamos impensables como la realidad virtual, impresión 3D, pequeños ordenadores portátiles en nuestro bolsillo, etc. Todo esto es posible gracias al increíble avance que se está consiguiendo en la miniaturización de los componentes electrónicos que componen dichos productos. Todo este incremento tecnológico está haciendo que cambiemos nuestra forma de vivir y de cómo interactuamos con el mundo que nos rodea. Y en gran parte, es posible, gracias a Internet, que ha permitido que prácticamente todo lo que nos rodea, esté conectado a Internet y de esta manera lograr una comunicación rápida y efectiva. Bajo dichas premisas, nace el concepto de internet de las cosas, (IoT, Internet of Things), siendo una de las tecnologías que están originando un fuerte impacto en el presente y que, sin duda, irán en aumento en el futuro.

Las aplicaciones de Internet de las Cosas son inagotables, como el sol y el viento. El amplio abanico de aplicaciones engloba desde electrodomésticos que nos avisan cuando han terminado de realizar sus tareas, o si se han averiado; alertarnos si hemos dejado alguna luz encendida y apagarla; mantener la temperatura adecuada de nuestro hogar para que cuando lleguemos éste nos resulte confortable; controlar las ventanas de nuestro hogar y programarlas para que se suban o bajen a horas determinadas y en función del clima exterior, o si nos vamos de vacaciones, simulen que hay gente en la vivienda; podemos conocer la temperatura en todo momento de nuestro hogar; o detectar si alguna persona ha entrado sin nuestro permiso, y un largo etcétera.

Como se ha mencionado, todo esto es posible gracias al concepto de Internet de las Cosas en conjunción con dispositivos inteligentes que se encargan de monitorizar y actuar en nuestras viviendas. Estos dispositivos inteligentes, dependiendo de la acción que llevan en el ambiente se denominan sensores y actuadores respectivamente, y combinados entre sí, junto a algoritmos de inferencia y razonamiento, forman lo que se denomina “Ambiente Inteligente”.

### 1.1.1 Ambientes inteligentes y envejecimiento de la población

Los ambientes inteligentes son espacios que usan la tecnología de sistemas empotrados, así como otras tecnologías, para crear ambientes interactivos que acerquen la computación al mundo físico y a los problemas cotidianos.

Según Alan Steventon y Steve Wright:

*“Los ambientes inteligentes son sistemas en los que la computación es usada para introducir mejoras imperceptibles o superficiales en las actividades comunes” [1]*

Uno de los motivos del interés en los ambientes, no solamente es lograr que los sistemas sean verdaderamente amigables con los usuarios, sino lograr que esencialmente sean invisibles o casi inexistentes para ellos.

Los ambientes inteligentes describen y manejan entornos físicos en los cuales las tecnologías de la información y la comunicación, así como los sistemas de sensores, pasan mayoritariamente desapercibidos para los usuarios, puesto que se hallan discretamente integrados a objetos físicos, a infraestructuras, y al entorno cotidiano en el cual vivimos, viajamos, y trabajamos. El objetivo de estos sistemas es el de permitir que ordenadores y sensores participen en actividades en las que nunca antes habían estado involucrados, posibilitando a la gente (a los usuarios) interactuar con los distintos dispositivos vía gestos, voz, movimientos, o simple información de contexto.

Una de las finalidades de estos ambientes inteligentes puede ser hacer frente al envejecimiento general de la población, ya que, según las previsiones del Departamento de Economía de las Naciones Unidas, a fecha de 2013, la población mundial mayor de 65 años aumentará en los próximos años de un 11,7% del total en 2013 a 21,1% en 2015 [2]. Los motivos de este incremento están directamente relacionados con el decremento de la fertilidad y con el incremento de la esperanza de vida. [3] En el caso de Europa, se espera un incremento de las personas mayores de un 16% en el año 2000 al 24% en 2030. [4]

Dentro de la población actual de personas mayores un 40% de la población mayor de 60 años vive sola y de forma independiente [2]. Esta tasa se encuentra en incremento y varía en función de la zona de estudio. Mientras que, en el norte de Europa, un 90% de la población vive de forma independiente (cifra similar a Estados Unidos en el sur (Grecia, Italia, España y Portugal) solamente un 40% vive en su hogar [5]. Estas diferencias están condicionadas principalmente por aspectos culturales y sociales.

A medida que la población envejece, la necesidad de invertir en atención sanitaria aumenta y se pone en relieve la necesidad de crear soluciones tecnológicas que permitan mejorar la atención reduciendo costes. El uso de sistemas de monitorización basados en el uso de las TIC y de tecnologías wearables permitirán cubrir esa necesidad [6].

Y a consecuencia de esto, dentro de este ámbito, un campo altamente estudiado es el reconocimiento de una actividad que nos permita monitorizar qué está realizando la persona, con el objetivo de estudiar sus rutinas e identificar posibles enfermedades como la demencia, TOCs, etc.

Es por ello, que en este TFM nos vamos a centrar en el proceso de reconocimiento de actividades.

### 1.1.2 Sistema de reconocimiento de actividades

En la última década, el avance de las tecnologías que incorporan los sensores, han conseguido que estos tengan: un bajo consumo, por lo que pueden ser alimentados con pilas o baterías durante meses, e incluso años; sean baratos, lo que da como resultado productos atractivos para la gente; tengan una gran precisión a la hora de tomar lecturas del entorno, y además éstos sean pequeños, por lo que no molestan en la vida diaria de las personas.

A su vez, las redes de comunicación por cable e inalámbricas y las técnicas de procesamiento de datos han mejorado sustancialmente, lo que permite mayor alcance y rapidez en la transmisión de los datos.

El reconocimiento de actividades es un proceso complejo que puede clasificarse en cuatro tareas básicas. Estas tareas son:

- 1) Elegir y desplegar sensores apropiados a objetos y entornos para monitorizar y capturar el comportamiento de un usuario.
- 2) Recopilar, almacenar y procesar la información obtenida empleando técnicas de análisis de datos.
- 3) Crear modelos de actividad computacional de una manera que permita a los sistemas llegar a conclusiones razonables.
- 4) Seleccionar o desarrollar algoritmos de razonamiento para inferir actividades.



Generalmente, el proceso de reconocimiento de actividades se agrupa en dos categorías, el reconocimiento basado en visión y en sensores.

#### 1.1.2.1 Reconocimiento basado en visión versus basado en sensores

El reconocimiento basado en visión se basa en el uso de instalaciones de detección visual, tales como cámaras de vídeo, para monitorizar el comportamiento de un actor y los cambios en el ambiente. Los datos generados son secuencias de vídeo o datos visuales digitalizados. Los enfoques de esta categoría emplean técnicas de visión computarizada, incluyendo la extracción de características, modelado estructural, segmentación de movimiento, extracción de acciones, y seguimiento de movimiento para analizar y reconocer patrones [6].

Por otro lado, el reconocimiento basado en sensores, hace uso de nuevas tecnologías basadas en redes de sensores, para realizar un seguimiento de las actividades realizadas en el ambiente inteligente. Los datos que generan los sensores son, principalmente, **eventos** en los que se deja constancia de cambios de estado. En estos enfoques, los sensores pueden acoplarse al objeto que se pretende observar, por ejemplo, “wearables” o teléfonos inteligentes, u objetos que forman parte del ambiente inteligente. Los “wearables”, a menudo, miden la inercia del cuerpo humano y son capaces de leer etiquetas de identificación por radiofrecuencia (RFID) para recopilar el comportamiento de una persona. Este enfoque es efectivo para reconocer movimientos físicos tales como el ejercicio físico [7] [8].

Sin embargo, a lo largo del tiempo, se ha ido viendo como el reconocimiento basado en sensores es más empleado que el basado en visión, debido a cuestiones de privacidad, ya que en el segundo son menores. Además, también a nivel computacional, el basado en visión requiere de altos niveles de procesamiento, aspecto que, es opuesto a la propuesta de este trabajo.

**Es por esto, que el presente trabajo se centra en el reconocimiento de actividades basado en sensores empleando dispositivos de bajo coste y minimizando el nivel computacional del reconocimiento.**

Aunque el registro de actividades puede ser obtenido mediante el simple monitoreo de los sensores, los modelos de actividades son necesarios para interpretar los eventos registrados por los sensores y, de esta forma, poder deducir las actividades. En particular, los mecanismos de reconocimiento de actividades están estrechamente relacionados con la naturaleza y la representación de los modelos de actividad.

En general, la construcción de los modelos de actividad puede estar dirigida por datos o por conocimiento:

#### 1.1.2.1.1 *Modelos de actividad basados en datos versus basados en el conocimiento*

El reconocimiento de actividades basados en datos consiste en aprender modelos de actividad a partir de conjuntos de datos preexistentes a gran escala con técnicas de minería de datos y aprendizaje automático. Este método implica la creación de modelos probabilísticos o de actividad estadística, seguidos de procesos de aprendizaje. El reconocimiento de la actividad resultante se basa en la clasificación probabilística o estadística, y a menudo, se la denomina enfoque basado en datos.

Las ventajas de estos enfoques radican en que son capaces de manejar la incertidumbre y la información en base al tiempo. Sin embargo, este método requiere conjuntos de datos grandes para el aprendizaje, y tiene como inconveniente la escasez de datos o el "arranque en frío". También es difícil que un modelo de actividad de una persona sea útil para otra.

Otro método para construir modelos de actividad es aprovechar un conocimiento previo en el área donde queremos reconocer actividades para construir modelos de actividad directamente usando ingeniería del conocimiento y tecnologías de gestión. Los modelos de actividad generados por este método se usan normalmente para el reconocimiento o predicción de la actividad a través del razonamiento lógico formal, por ejemplo, deducción, inducción o abducción. Como tal, este método se conoce como enfoque basado en el conocimiento.

Los enfoques basados en el conocimiento tienen las ventajas de ser semánticamente claros, lógicamente elegantes y fáciles de empezar. Sin embargo, son débiles a la hora de manejar la incertidumbre e información temporal y los modelos pueden ser vistos como estáticos e incompletos.

En el caso de este trabajo, se va a centrar en el reconocimiento de actividades basado en datos, puesto que se dispone de un repositorio de datasets, CASAS, el cual es más que suficiente para poder generar un conjunto de datos preexistente. Para realizar el reconocimiento de actividades, se usará el algoritmo KNN, puesto que es un algoritmo muy sencillo de implementar, de buena exactitud en comparación con otros y de coste computacional no muy elevado, siempre que, el modelo de actividad no sea grande. En caso contrario, el coste puede llegar a ser muy elevado, ya que emplea todo el modelo para crear las reglas.

Por ello, y dado que el sistema a implementar es de bajo coste, se busca que la carga computacional también sea baja. **Para ello, se propone la generación de prototipos, que reduce el número de instancias utilizadas para realizar la clasificación posterior en el KNN.**

## 1.2 PROPUESTA

---

La propuesta del presente trabajo es la puesta en marcha de un sistema de monitorización, basado en sensores de bajo coste, de un ambiente inteligente con un sistema de reconocimiento de actividades con baja carga computacional.

Para ello, el sistema contará con una red de sensores desplegada en un ambiente inteligente, en nuestro caso, el laboratorio de inteligencia ambiental de la Universidad de Jaén. Dicha red de sensores recogerá los cambios en el ambiente, denominados eventos, y que nos permiten conocer las iteraciones que se realizan en el ambiente como, por ejemplo, abrir puertas, tirar de la cisterna, sentarse en el sofá, etc. Para el despliegue de este sistema se contarán con sensores bajo coste inalámbricos que enviarán la información a un dispositivo llamado Raspberry Pi.

Además, este TFM no solamente propone un sistema de monitorización de ambientes, si no que va más allá, tratando el problema de reconocer una actividad a partir de una secuencia de eventos, es decir, de un conjunto de cambios de estado en los sensores. Dado que este TFM se centra en el uso de dispositivos bajo coste se buscará un enfoque de reconocimiento de actividades con baja carga computacional sin mermar la precisión de los resultados.

Finalmente, este TFM presenta una aplicación móvil desarrollada en Android, donde el usuario podrá monitorizar el ambiente inteligente y las actividades que se realizan en él en cualquier lugar y en cualquier momento a través de su dispositivo inteligente.

Por tanto, las propuestas que se realizan en este TFM están enfocadas a proporcionar un sistema de monitorización que pueda ser desplegado en los hogares de personas mayores a un precio muy competitivo con el objetivo de monitorizar su interacciones y actividades para permanecer el mayor tiempo posible en sus hogares, a la vez, que sirve de ayuda a los cuidadores informales de estos.

## 1.3 OBJETIVOS

---

Para llevar a cabo la propuesta anteriormente explicada, es necesario abordar una serie de objetivos más específicos:

- Estudio y despliegue de una red de sensores inalámbricos de bajo coste que monitoricen un ambiente inteligente.
- Estudio y desarrollo de un dispositivo que reciba la información de los sensores de manera inalámbrica y la almacene en una base de datos.
- Estudio de reducción de la carga computacional de reconocimiento de actividades basados en KNN en una configuración de sensores a través de la generación de prototipos.
- Implementación del algoritmo de reconocimiento de actividades KNN con los prototipos generados para el apartamento de inteligencia ambiental de la Universidad de Jaén.
- Realización de una aplicación móvil completa que permita monitorizar el ambiente inteligente y las actividades que se realizan en él.

## 1.4 ESTRUCTURA DE LA MEMORIA

---

A continuación, explicaré brevemente el contenido de éste documento, especificando cada uno de sus apartados y qué contiene cada uno de ellos.

En el capítulo 1 se realiza una introducción a los conceptos generales en los que se basan este proyecto como ambiente inteligente y sistema de reconocimiento de actividades, se plantean los objetivos que este proyecto ha de cumplir y la planificación que se ha de llevar para conseguirlos.

En el capítulo 2 se efectúa el estudio y despliegue tanto de los sensores de bajo coste como del dispositivo encargado de recibir la información. En dicho estudio se realizará un estudio del mercado actual.

En el capítulo 3 se realiza el estudio de los algoritmos de generación de prototipos, junto con el algoritmo KNN, teniendo como objetivo la reducción de carga computacional del algoritmo.

En el capítulo 4 se explica de manera detallada el análisis, diseño e implementación de la aplicación móvil estipulada en el quinto objetivo.

En el capítulo 5 se enumeran las conclusiones que he sacado de la realización de este proyecto, así como las futuras mejoras que podrían incorporar al proyecto.

Finalmente se incluyen dos anexos: manual de instalación, explicando cómo realizar la instalación de cada uno de los apartados que componen el sistema; y manual de usuario, donde se explica detalladamente, como se usa la aplicación móvil.

## 1.5 PLANIFICACIÓN

---

El objetivo de la planificación del proyecto de software es proporcionar un marco de trabajo que permita al gestor de planificación hacer estimaciones razonables de recursos, costos y planificación temporal

Estas estimaciones se hacen dentro de un marco de tiempo limitado al comienzo de un proyecto de software, y deberían actualizarse regularmente a medida que progresa el proyecto.

Las estimaciones deberían definir el escenario del mejor caso, y peor caso de modo que los resultados del proyecto pueden limitarse.

A la hora de planificar un proyecto se deben tener en cuenta una serie de cuestiones:

- Determinar las condiciones exactas para que el proyecto sea finalizado o completado. Antes de que estén claros cuales son los objetivos del proyecto, no tiene sentido comenzar a estimar cuanto tiempo llevará y/o cuánto costará.
- Hacer un inventario de todo el trabajo que se requiere sea hecho.
- Decidir si este plan tiene sentido, es decir, si los costos justifican los beneficios.
- Definir dependencias entre tareas. Algunas tareas necesitan ser terminadas antes que otras tareas puedan comenzar.
- Crear un cronograma de proyecto, usando un diagrama de Gantt.

En este caso, los pasos para concretar la planificación del proyecto son la especificación de las tareas y el cronograma Gantt que las muestra en función del tiempo.

### 1.5.1 Especificación de tareas

Las tareas básicas que vamos a encontrar en el desarrollo de este proyecto son:

- **Realización de experimentaciones:** Comprende todo el proceso de experimentación llevado a cabo con distintos datasets y algoritmos para concluir cual es el algoritmo más óptimo para la generación del modelo de actividad.
- **Estudio del mercado actual:** Se realiza todo el proceso de estudio de sensores existentes en el mercado y cuales se ajustan a nuestros requisitos, así como el controlador que recogerá los eventos de estos sensores.
- **Despliegue de la red de sensores:** Una vez adquiridos todos los elementos necesarios, se realiza el despliegue en el ambiente inteligente. Tras éste, se procede a su configuración y emparejamiento con el controlador.
- **Implementación de algoritmos:** En esta tarea se desarrolla todo lo relacionado con el reconocimiento de actividades, tanto el algoritmo que reconoce las actividades, como el KNN, que las clasifica.
- **Implementación del servidor web:** En esta tarea, la principal misión será llevar a cabo el despliegue del servidor web con su correspondiente servicio web, que dará soporte a la API, que permitirá la comunicación con la aplicación móvil.
- **Implementación de la aplicación móvil:** Esta tarea comprende todo el desarrollo de la aplicación móvil para Android. Esta será la encargada de mostrar toda la información al usuario. Una vez desarrollada se realizará un conjunto de pruebas para comprobar su correcto funcionamiento.

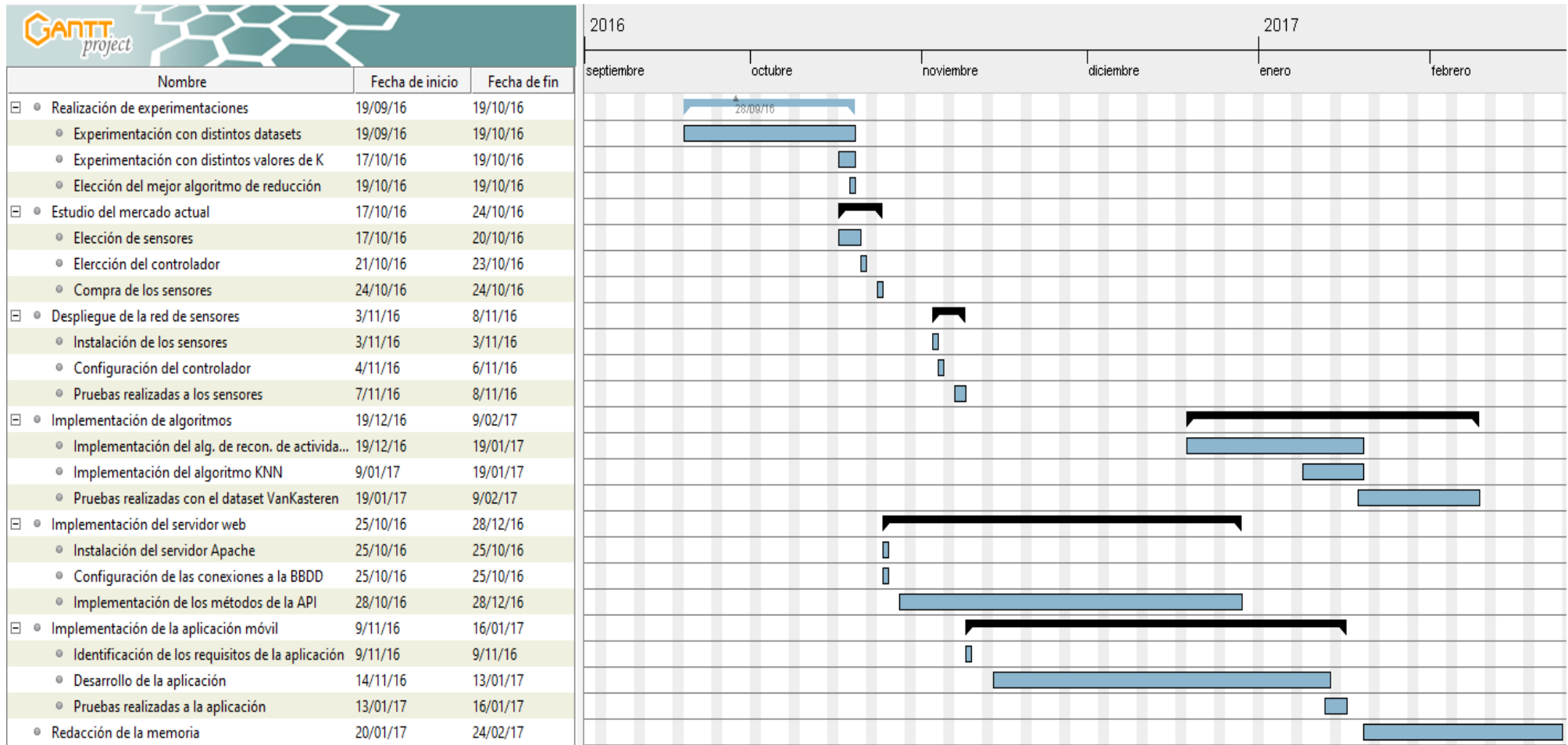


Ilustración 1 - Diagrama de Gantt





# CAPÍTULO 2

## SISTEMA DE SENSORES DE BAJO COSTE

---

## 2.1 INTRODUCCIÓN

---

Para poder realizar el reconocimiento de actividades, primero, es necesario disponer de un sistema implantado que sea el encargado de recoger todos los eventos que vayan surgiendo en el ambiente inteligente.

Ya que una de las características del sistema es que sea de bajo coste, se ha tenido en cuenta el precio de cada uno de los componentes escogidos, sin que esto afecte a la precisión y/o calidad de los mismos.

Se considera que los sensores inalámbricos son más adecuados respecto a los cableados, puesto que poseen una serie de ventajas:

- **Seguridad:** Pueden ser utilizados en lugares de difícil acceso debido a las condiciones extremas, tales como alta temperatura, pH, presión, etc. Al utilizar los sensores inalámbricos, resulta útil para la obtención de datos en lugares de difícil acceso, y donde el sensor está en movimiento y/o rotación.
- **Conveniencia:** Los sensores inalámbricos se pueden utilizar para formar una red que permita al ingeniero controlar un número de lugares diferentes a partir de una estación.
- **Reducción de costos:** El control del proceso inalámbrico puede reducir el costo de la supervisión y despliegue, eliminando la necesidad de cables de extensión, conductos y otros accesorios costosos.
- **Batería amplia:** Estos sensores incorporan una batería capaz de durar hasta 2 años, lo cual, no lo convierte en una desventaja en comparación con sus opuestos cableados.

También es conveniente a la hora de elegir los sensores tener en cuenta los siguientes aspectos:

- **Tipo de medida:** Es importante entender lo que se está midiendo. Los transmisores inalámbricos (que incorporan la medición y control de proceso inalámbrico) suelen tener una función única. Los sensores están diseñados específicamente para temperatura, presión, caudal, etc., y se deben seleccionar en consecuencia. En nuestro caso, requerimos de al menos 3 tipos de medida:
  - **Presión:** Es un sensor con forma de manta que se coloca debajo de donde se desea detectar la presión, como una cama, sofá o escaleras. Al hacer presión sobre él, por

ejemplo, cuando alguien lo pisa o se echa sobre él, sus dos capas interiores entran en contacto, por lo que se cierra el circuito. Una vez desaparece la presión, vuelve a su estado original. Estos cambios son enviados al receptor.

- **Contacto:** Incorpora un sensor magnético de proximidad, para detectar la apertura de una puerta o ventana. El dispositivo consta de una parte con un imán (la parte móvil) fijada en la puerta o la ventana, y la unidad principal situado en la parte fija de la puerta o ventana con tornillos o un adhesivo. Cuando ambas partes no están enfrentadas, cambia de estado y envía el evento al receptor.
- **Movimiento:** Estos sensores incorporan un componente PIR (receptor de infrarrojos pasivo) que detecta la presencia de una persona en una habitación. El funcionamiento de estos componentes se basa la captura de la radiación infrarroja que emiten todos los cuerpos. Al existir un cambio de temperatura brusco en el campo de visión del sensor, como el paso de una persona, este es detectado por el sensor que pasa a enviar el evento al receptor. El sensor se encuentra siempre en estado de hibernación a menos que se detecte un movimiento. De esta forma se consigue un gran ahorro de batería.
- **Precisión y tiempo de respuesta:** La mayoría de los sensores inalámbricos son tan precisos como sus contrapartes cableadas; sin embargo, las lecturas se transmiten típicamente en intervalos de unos pocos segundos para conservar la energía de la batería. Si la medición instantánea es necesaria, se deberá tener en cuenta a la hora de seleccionar el transmisor inalámbrico, porque algunos modelos no pueden ofrecer el tiempo de respuesta deseada.
- **Rango:** La gama de sensores inalámbricos es muy variada. Algunos están diseñados para corto alcance, aplicaciones interiores hasta 100 metros, mientras que otros sensores pueden transmitir datos a un receptor situado a varios kilómetros de distancia. Independientemente de la capacidad de los sensores, el rango de una señal inalámbrica está siempre limitada por los obstáculos. La transmisión a través de paredes y/o aparatos eléctricos, degrada la fuerza de la señal y reduce la capacidad de alcance. Como resultado, el rango de un transmisor localizado en el interior es típicamente significativamente menor que el mismo transmisor que transmite afuera en un campo abierto.
- **Frecuencia:** La radio frecuencia de transmisión también es importante considerarla. Las leyes varían según el país y/o región en relación a las partes del espectro inalámbrico que están disponibles para uso sin licencias específicas. En Europa, los productos inalámbricos

normalmente operan en 868MHz o 2,4GHz (Wifi), y no es necesario una licencia de radio para operar en estas frecuencias. Debido a los requisitos regulatorios, los productos pueden estar disponibles sólo en ciertas regiones.

## 2.2 ESTUDIO DE MERCADO

---

Hoy en día hay productos más o menos completos dependiendo de la marca y de la tecnología que usen para transmitir la información. Esto, obviamente, influye en el precio final.

Aunque primero hay que mencionar las tecnologías disponibles a la hora de transmitir la información inalámbricamente:

### **Z-Wave**

Es un protocolo de comunicación inalámbrico utilizado principalmente para la domótica. Está orientado al control y automatización de hogares y pretende proporcionar un método sencillo y fiable para controlar de forma inalámbrica la iluminación, la climatización, los sistemas de seguridad, el cine en casa, los controles de piscinas y balnearios y los controles de acceso de garaje y hogar. Al igual que otros protocolos y sistemas dirigidos al mercado doméstico y de automatización de oficinas, un sistema basado en Z-Wave puede ser controlado a través de Internet, con un dispositivo de control central que sirve como controlador.

Z-Wave fue desarrollado originalmente por la empresa dinamarquesa Zen-Sys y posteriormente adquirida por Sigma Designs en 2008. Hay más de 1.500 productos Z-Wave interoperables comercializados bajo diferentes marcas y más de 35 millones han sido vendidos desde 2005. [9]

### **ZigBee**

ZigBee es el nombre de la especificación del conjunto de protocolos de alto nivel de comunicación inalámbrica basada en el estándar IEEE 802.15.4 de redes inalámbricas de área personal (wireless personal area network, WPAN). Su objetivo son las aplicaciones que requieren comunicaciones seguras con baja tasa de envío de datos y maximización de la vida útil de sus baterías.

ZigBee pretende ser más sencillo y menos costoso que otras redes inalámbricas de área personal (WPAN), como Bluetooth o Wi-Fi. Su bajo consumo de energía limita las distancias de transmisión a

10 - 100 metros de alcance, dependiendo de la potencia de emisión y las características ambientales. No obstante, los dispositivos ZigBee pueden llegar a transmitir datos a largas distancias haciendo uso de la red mallada formada por múltiples dispositivos que hacen de puente para llegar a otros más distantes.

ZigBee se utiliza típicamente en aplicaciones que no requieren una alta transmisión de datos, garantizando un bajo consumo de batería y creando redes seguras (las redes ZigBee están protegidas con claves de cifrado simétricas de 128 bits). Por defecto, la velocidad de transmisión es de 250 kbit/s, ideal para transmisiones de datos intermitentes desde un sensor o dispositivo de entrada.

### **Wi-Fi**

Wi-Fi es una tecnología inalámbrica de área local con dispositivos basados en los estándares IEEE 802.11. Existen multitud de dispositivos que utilizan esta tecnología Wi-Fi como: PCs, videoconsolas, smartphones, cámaras digitales, tabletas, reproductores de audio digital e impresoras modernas. Los dispositivos compatibles con Wi-Fi se conectan a Internet a través de una red WLAN y un punto de acceso inalámbrico. Este punto de acceso (o hotspot) tiene un alcance de unos 50 metros en interiores y un rango mayor al aire libre.

Esta tecnología utiliza las frecuencias 2,4 GHz y 5 GHz. Para su protección, posee varias tecnologías de cifrado. Uno de los primeros cifrados en surgir fue Wired Equivalent Privacy (WEP) pero fue fácil de romper, por lo que, agregaron protocolos de mayor calidad, como Wi-Fi Protected Access (WPA, WPA2).

### **Bluetooth**

Bluetooth es una especificación para Redes Inalámbricas de Área Personal (WPAN) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia a 2.4 GHz de frecuencia. La versión más actual de esta tecnología es la 4.2.

La especificación de Bluetooth define un canal de comunicación a un máximo 720 kbit/s (1 Mbit/s de capacidad bruta) con rango óptimo de 10 m (opcionalmente 100 m con repetidores).

De todas estas tecnologías, se podría decir que las más enfocadas al internet de las cosas son Z-Wave y ZigBee, por lo que serán estudiadas más a fondo.

<i>Características</i>	<i>Z-Wave</i>	<i>Zigbee</i>
<i>Consumo</i>	Muy Eficiente	Eficiente
<i>Alcance</i>	80 - 100 m	60 – 80 m
<i>Precio</i>	Medio / Bajo	Alto / Medio
<i>Dispositivos existentes</i>	Amplia cantidad de dispositivos usando esta tecnología	Número reducido de dispositivos que usan esta tecnología
<i>Interoperabilidad</i>	Muy buena	Buena
<i>Número de dispositivos por HUB</i>	232 dispositivos	64 mil dispositivos
<i>Máximo ancho de banda</i>	100 Kb/s	250 kb/s

*Tabla 1 - Tabla comparativa entre distintas tecnologías*

Por tanto, como se puede observar en la Tabla 1, en prácticamente todos los aspectos, Z-Wave sale vencedor con respecto a ZigBee.

Pero sin duda, Z-Wave ha sido el ganador por disponer de una gran cantidad de fabricantes y dispositivos en el mercado que apuestan por esta tecnología en comparación con Zigbee. Además, algo a tener en cuenta es que a pesar de existir múltiples fabricantes que empleen Z-Wave en sus dispositivos, todos son compatibles entre sí, por lo que, el fabricante no supone un inconveniente a la hora de elegir un dispositivo u otro, sino más bien sus características y precio.

Por tanto, una vez tenemos claro la tecnología a usar, es hora de buscar qué fabricantes existen en el mercado, para decantarnos por aquel que mejor satisfaga nuestras necesidades. Por mencionar a algunos de ellos: Fibaro, Everspring, Netatmo, Qubino, Nest, Aeon, Zipato, etc.

Como se mencionó anteriormente, los sensores requeridos para este sistema son de tres tipos: sensores de contacto, de presión y de movimiento.

A continuación, vamos a mostrar los sensores anteriormente citados de algunas de las marcas existentes en el mercado:

### **Sensores de contacto**

- **Fibaro:** Además, del sensor habitual de proximidad por imán, incorpora una entrada analógica que permite conectar un sensor de temperatura, junto con otra conexión que permite conectar un dispositivo de cierre binario. Esto permitirá conectar los sensores de presión, que serán explicados con más detalle. Su precio es de 41 €.



*Ilustración 2 - Sensor de contacto de Fibaro*

Sus características principales son las siguientes:

- Alimentación: Batería de ER14250 (1/2AA) 3,6 V
  - Dimensiones: 76 x 17 x 19 mm
  - Frecuencia: 868.42 MHz
  - Distancia de transmisión: Campo 50m libre, 30 m en edificios
  - Temperatura de funcionamiento: 0 - 40 ° C
- **Everspring:** Este no incorpora entrada para añadirle un sensor de temperatura, aunque si las



*Ilustración 3 - Sensor de contacto de Everspring*

conexiones para conectar un sensor binario externo. Su precio es de 40 €.

Sus características principales son las siguientes:

- Alimentación: 3 pilas AAA \* Micro
  - Dimensiones: 100 x 28.5 x 18.9 mm
  - Frecuencia: 868.42 MHz
  - Distancia: hasta 30 m en edificios
  - Temperatura de funcionamiento: 0 - 40 ° C
- 
- **Aeon Labs:** Como ventajas al resto, incorpora dos sensores de proximidad por lo que es posible instalar el lado opuesto en la parte superior o lateral. Por el contrario, no incorpora



*Ilustración 4 - Sensor de contacto de Aeon Labs*

conexión para sensores binarios externos, ni medidor de temperatura. Su precio ronda los 35 €.

Sus características son las siguientes:

- Alimentación: Batería recargable con 500 mAh de capacidad.
- Dimensiones: 9 x 72 mm.
- Frecuencia: 868.42 MHz
- Se puede pintar, de manera que no se destaca con el resto del mobiliario.
- Fabricado con plástico ABS.



- Hasta 150 m de alcance punto a punto.
- **Zipato:** Este sensor incorpora 3 funciones diferentes: sensor de contacto, sensor de temperatura y de luz. Esto supone una ventaja a la hora de adquirir información del



*Ilustración 5 - Sensor de contacto de Zipato*

ambiente. Tiene un precio de 42 €.

Sus principales características son:

- Alimentación: 3V CR123A
- Dimensiones:
  - Sensor: 28 x 96 x 23 mm
  - Imán: 10 x 50 x 12 mm
- Frecuencia: 868.42 MHz
- Distancia de transmisión: 30m en interiores
- Sensor de temperatura: -10 a 70 ° C
- Sensor de luz: 0-500 lux
- Peso: 52 g
- Temperatura de funcionamiento: -10 a 40 ° C

### **Sensores de movimiento**

- **Fibaro:** Además del sensor de movimiento lleva incorporado sensor de temperatura y luminosidad. Esto supone un plus extra a la hora de monitorizar el ambiente inteligente. Su precio es de 48 €.



*Ilustración 6 - Sensor de movimiento de Fibaro*

Sus principales especificaciones son:

- Potencia: CR123A 3,6 VDC
- Rango de temperatura de medición: -20 ° C a 100 ° C
- Precisión de la medida: 0,5 ° C
- Rango de medición de la luminosidad: 0-32000 LUX
- Frecuencia: 868.42 MHz
- Distancia de transmisión: de campo libre de 50 metros, 30 metros en interiores
- Medidas: 4,4 cm de diámetro
- Temperatura de funcionamiento: 0 - 40 ° C

- **Everspring:** No incorpora sensores aparte del de movimiento. Tiene protección eléctrica IP44



*Ilustración 7 - Sensor de movimiento de Everspring*

por lo que puede ser colocado en exterior. Su precio es de 45 €.

Sus principales características son:

- Distancia: hasta 12 m
- Ángulo de visión: 110 grados de forma predeterminada
- Fuente de alimentación: 3 \* pilas AA
- Peso: 150 gr.
- Dimensiones: 100x96x93 mm

### ***Sensores de presión***

- **Arun:** Es la única marca que he encontrado a un precio asequible en el mercado. Dispone de



*Ilustración 8 - Sensor de presión de Arun*

varios tamaños, dependiendo del lugar donde se vayan a colocar. Su precio ronda los 14 €.

Medidas disponibles:

- 720 x 560 x 3mm
- 595 x 170 x 3mm
- 680 x 170 x 3mm

Una vez vistos algunos de los sensores existentes, es importante elegir el controlador donde se conectarán los sensores y al que enviarán la información. Normalmente cada fabricante suele incluir el suyo propio, aunque esto no significa que no sea compatible con el resto de dispositivos. Además, existen fabricantes que únicamente comercializan éstos productos. Todos ellos disponen de conexión a internet, por lo que permiten que se pueda supervisar y controlar los dispositivos conectados a ellos mediante Z-Wave. A continuación, se muestra una lista de algunos de ellos:



*Ilustración 9 - Controlador Home Center de Fibaro*

### **Fibaro**

Sus principales características:

- Arquitectura de hardware extremadamente eficiente.
- Consumo de energía ultra bajo
- Acceso remoto a través de la página web o un teléfono móvil/Smartphone/tablet
- Interfaz simple y fácil de usar
- Configuración rápida y sencilla
- Envía notificaciones vía SMS

- Escenas, dependiendo del clima y otras variables, definidas por el usuario
- Copia de seguridad del sistema siempre. se guarda en un pen drive conectado, oculta en la caja
- Registro histórico

La principal desventaja de este dispositivo es su elevado precio, alrededor de 500 €, lo que lo hace inviable para el propósito que estamos tratando de construir un sistema de bajo coste.



*Ilustración 10 - Controlador de Vera*

### **Vera**

Sus principales características:

- Soporte con los nuevos modelos Z-Wave Plus
- Interfaz simplificada con respecto a sus antecesores
- Compatibilidad con dispositivos ajenos a Z-Wave
- Gran comunidad que lo respalda

Una vez más, el principal inconveniente que tiene es el precio, ya que ronda los 150 €.

### ***Eedomus Plus***



*Ilustración 11 - Controlador de Eedomus*

Sus principales características son:

- Versión actualizada del eedomus
- Compatible con Z-Wave Plus
- Uso sencillo, compatible con muchos objetos conectados y termostatos Wifi
- Creación de escenas sencilla y rápida
- Compatible con HomeKit y Siri de Apple (vía Homebridge)

Como el resto de productos anteriormente vistos, el precio vuelve a ser un inconveniente, ya que roza los 300 €



*Ilustración 12 – Módulo controlador RaZberry para Raspberry Pi*

### **RaZberry**

Es una pequeña placa de extensión que permite transformar una Raspberry Pi en un controlador Z-Wave Plus. Se conecta muy fácilmente a la Raspberry Pi, directamente a los puertos GPIO de ésta. Por tanto, no necesita alimentación externa ya que se alimenta de la propia Raspberry Pi. Dispone además de una antena integrada y dos indicadores LED de color rojo y verde para dar información al usuario sobre su estado en cada momento.



*Ilustración 13 - Módulo acoplado a la Raspberry Pi*

Una vez instalado el software encargado de controlar este dispositivo, es posible desarrollar un propio software de control o bien emplear soluciones ya consolidadas como Z-Wave Smart Home, JEEDOM o Domoticz, entre otros.

Sus principales características son:

- Posibilidad de crear su propio Controlador
- Basado en Z-Wave SDK 4.54 y 6.51
- Servicio de copia de seguridad y recuperación
- Optimización de la cola para acelerar el proceso de transmisión
- Actualización del firmware mediante el sistema operativo de Raspberry Pi
- Administración completa de su red Z-Wave (incluir, excluir, gestión de red)
- Gestión de asociaciones directas entre dispositivos
- Gestión de escenas, scripts y temporizadores usando scripts de JavaScript
- Proporciona API C nativa y Web API JSON Script local basado en el Google V8 Java Script

Una de las grandes ventajas con respecto al resto de dispositivos es el bajo coste que tiene, ya que es solo de 50 € + 40 € de la Raspberry que además permite multitud de posibilidades aparte de esta. Por tanto, para la implantación de este sistema, se ha optado por usar esta tecnología. Por tanto, vamos a detallar un poco en qué consiste la placa en la que se sustenta esta placa.

## 2.3 CONFIGURACIÓN DE LA RED DE SENSORES PARA EL APARTAMENTO DE INTELIGENCIA AMBIENTAL DE LA UNIVERSIDAD DE JAÉN

---

El ambiente inteligente donde se va a realizar el despliegue del sistema se encuentra en la Universidad de Jaén. Concretamente en la dependencia 109 del edificio C6 [10]. Dispone de un espacio de 25 m<sup>2</sup> en los hay espacio para una cocina, dormitorio, salón y entrada.

En la cocina existen multitud de electrodomésticos, típicos de una casa habitual como una lavadora, lavavajillas, frigorífico, microondas y un hervidor. En el dormitorio cuenta una cama, un lavabo, y un inodoro. El salón incorpora una TV de 40", teléfono, sofá, videoconsola y distintos armarios donde almacenar los utensilios del hogar.

A continuación, se muestra el plano del ambiente inteligente:





### 2.3.1 Raspberry Pi

Después de realizar un estudio del mercado actual, se ha optado por escoger el sistema Raspberry Pi + módulo RaZberry, ya que es el que permite mayor libertad a la hora de desarrollar un software adaptado para el propósito de este proyecto, su precio es notablemente inferior en comparación con el resto de productos vistos, y proporciona una forma de descubrir e investigar un sistema tan a la orden del día como lo es Raspberry Pi.

La Raspberry Pi se trata de una diminuta placa base de 85 x 54 milímetros (un poco más grande que una cajetilla de tabaco).

El modelo más actual es Raspberry Pi 3 model B, lanzada en 2016. Dicho modelo cuenta con un SoC Broadcom BCM2835 en el que se incluye tanto la CPU, GPU, RAM y puerto USB). La CPU es un Quad-Core-Core a 1,2 GHz, con arquitectura ARM Cortex A8. La GPU es una VideoCore IV que admite resoluciones de hasta 1080p. La memoria RAM integrada es de 1 GB, los cuales comparte con la GPU. Además, en esta última versión se incorpora Wi-Fi y Bluetooth dentro de la propia Raspberry, por lo que ya no son necesario comprarlos.

También incorpora multitud de conexiones:

- **MicroUSB:** Es por donde se alimenta la Raspberry. Funciona a un voltaje de 5V, que es el empleado en la mayoría de cargadores de móviles, por lo que éstos serán compatibles,

siempre cuando sean capaces de entregar 2 Amperios como mínimo, ya que un amperaje inferior puede hacer que no funcione correctamente. Además, este conector se encarga de alimentar el resto de dispositivos que estén conectados a la placa, como teclados, ratones, discos duros portátiles, etc. También es el que da alimentación al módulo RaZberry.

- **GPIO:** Son un conjunto de 40 pines que permiten que la Raspberry interactúe con multitud de sensores, actuadores, módulos, etc. De estos 40 pines, 26 son de entrada/salida, y el resto de voltaje/masa. En estos pines también es posible conectar módulos como el de RaZberry.
- **USB:** Cuenta con 4 puertos USB para conectar lo que necesitemos. En nuestro caso será necesario usar al menos 1 o 2 puertos para conectar el ratón y el teclado. Los otros pueden servir para conectar memorias externas como discos duros, pendrives, etc.
- **MicroSD:** Situada en la parte trasera, es el “disco duro” de la Raspberry. Es donde se instala el sistema operativo. Por tanto, es más que recomendable que sea una tarjeta de lectura/escritura rápida, ya que de esto depende la velocidad con la que se ejecutará la mayoría de proceso. La tarjeta debe contar con un espacio recomendado de al menos 8GB.
- **HDMI:** A través de un cable HDMI podemos conectar la Raspberry a una televisión o monitor para poder uso de su sistema operativo con interfaz gráfica. Aunque también cabe decir que puede ser controlada a través de SSH remotamente, y por tanto no haría falta utilizar esta conexión.
- **Jack de 3,5 mm:** Por si queremos conectar unos cascos, altavoces u otro tipo de dispositivo de audio.
- **Puerto Ethernet:** Puerto por el que se establece una conexión a internet. Es más seguro y fiable que la conexión Wi-Fi, aunque nos obliga a depender de la longitud del cable hasta el router.
- **Conector display (DSI):** Conector donde se puede enchufar pequeñas pantallas táctiles especialmente preparadas para emplear con la Raspberry Pi. Aun así, no es aconsejable su uso ya que el espacio de maniobra de estas pantallas suele ser muy reducido, debido al escaso tamaño de la pantalla (normalmente 3.5”).
- **Conector camera (CSI):** Conector donde podemos conectar una cámara y que podremos programar gracias a las librerías existentes.

Los sistemas operativos que emplea mayoritariamente están basados GNU/Linux. Raspbian, una distribución derivada de Debian, optimizada para el hardware de Raspberry Pi, se lanzó durante julio de 2012 y es la distribución recomendada por la fundación para iniciarse.



*Ilustración 14 - Raspberry Pi*

Junto a la adquisición de la Raspberry Pi también han sido necesarios los siguientes productos:

- Cargador de 3 Amperios
- Tarjeta SD de 16 Gb
- Carcasa

### 2.3.2 Red de sensores y localización

Del estudio anteriormente realizado, al final los sensores escogidos han sido los siguientes:

- 10 sensores de contacto, de la marca Fibaro, empleados en las siguientes localizaciones:
  - Puerta del microondas
  - Puerta de la lavadora
  - Puerta del lavavajillas
  - Tapa del inodoro
  - Puerta de una de las ventanas
  - Puerta del frigorífico
  - Puerta principal
  - Teléfono
  - Mando de la televisión
  - Hervidor

El motivo por el cual se han escogido estos sensores y no otros, han sido por su reducido tamaño en comparación con el resto y por la calidad y fiabilidad del producto. Además, el precio era bastante similar al del resto por lo que se ha considerado que esta era la mejor opción.

- 6 sensores de presión empleados en:
  - 3 para el sofá (Dividido en: parte izquierda, parte central y parte derecha)
  - 3 para la cama (Dividido en: almohada, parte central, pies de la cama)

Para el despliegue de estos sensores se han empleado dos medidas distintas dependiendo de la ubicación de dichos sensores:

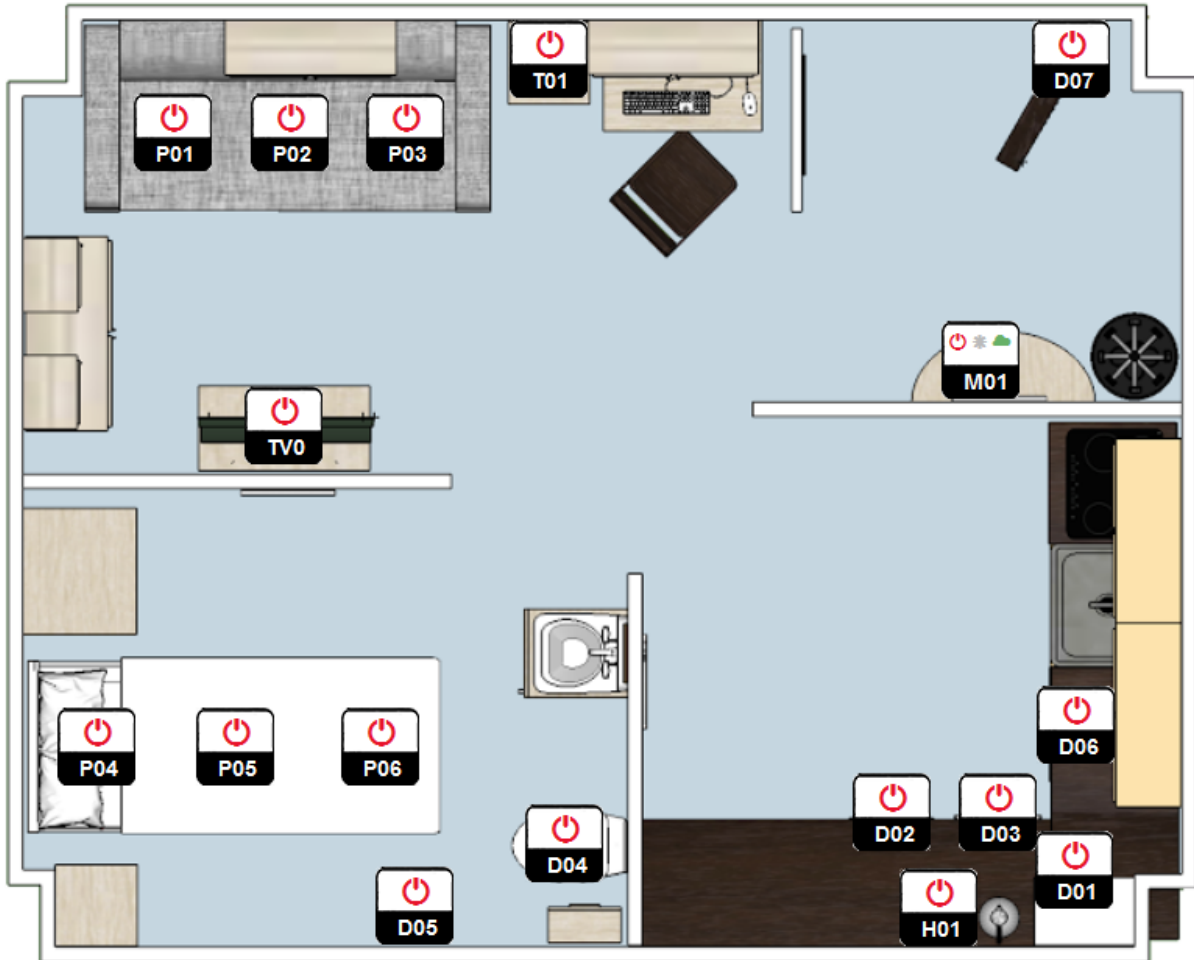
- 595 x 170 x 3mm para los 3 sensores que componen el sofá
- 800 x 560 x 3mm para los 3 sensores que forman la cama

Junto a cada uno de los sensores, han sido necesario un sensor de contacto, ya que, incluyen una entrada binaria que sirve para conectar los cables que salen del sensor de presión, puesto que estos no son inalámbricos por sí mismos.

- 1 sensor de movimiento, de la marca Fibaro, ubicado enfrente de la puerta principal. El motivo por el que he usado este y no otro ha sido por los extras que incorpora, como sensor de temperatura y de luminosidad. Además, su tamaño es muy reducido, (aproximadamente el tamaño de una pelota de ping-pong).

La colocación de estos sensores no puede ser más sencilla. Vienen provistos de una tira adhesiva, la cual permite instalarlos donde más fácil sea. Con lo que retirando la protección de la tira y colocándolos donde queramos, quedarían todos los sensores instalados.

La ubicación final de los sensores quedaría, por tanto, de la siguiente manera:



Una vez desplegados, quedaría concluido el desarrollo de la parte física del sistema. La ubicación de la Raspberry es irrelevante mientras esté dentro del alcance inalámbrico de los sensores.

## 2.4 PRESUPUESTO

Una vez escogidos todos los componentes que componen el sistema, es el momento de realizar un presupuesto de todos los productos necesarios.

<i>Componente</i>	<i>Enlace</i>	<i>Precio</i>	<i>Unidades</i>	<i>Precio total</i>
<i>Raspberry Pi 3</i>	<a href="#">Enlace</a>	39,99 €	1	39,99 €
<i>Módulo RaZberry</i>	<a href="#">Enlace</a>	56,58 €	1	56,58 €

<i>Tarjeta SD clase 10 16GB</i>	<a href="#">Enlace</a>	6,99 €	1	6,99 €
<i>Carcasa + Cargador + Disipadores</i>	<a href="#">Enlace</a>	11,99 €	1	11,99 €
<i>Sensores de contacto</i>	<a href="#">Enlace</a>	40,96 €	10*	409,60 €
<i>Sensor de movimiento</i>	<a href="#">Enlace</a>	48,26 €	1	48,26 €
<i>Sensores de presión</i>	<a href="#">Enlace</a>	13 €	6	78 €
<b><i>Total</i></b>				<b>651,41 €</b>

Tabla 2 - Presupuesto del despliegue

\*Se compraron 10 sensores de contacto, dado que ya había sensores disponibles en el ambiente que fueron usados para los sensores de presión.



# CAPÍTULO 3

## RECONOCIMIENTO DE ACTIVIDADES BASADO EN SENSORES

---



## 3.1 INTRODUCCIÓN

---

El reconocimiento de actividades basados en sensores es uno de los campos cruciales en ambientes inteligentes con el objetivo de fomentar la independencia de las personas mayores.

A medida que la población envejece, la necesidad de invertir en atención sanitaria aumenta y se pone en relieve la necesidad de crear soluciones tecnológicas que permitan mejorar la atención reduciendo costes. El uso de sistemas de monitorización basados en el uso de las TIC y de tecnologías wearables permitirán cubrir esa necesidad.

Y a consecuencia de esto, dentro de este ámbito, un campo altamente estudiado es el reconocimiento de una actividad que nos permita monitorizar qué está realizando la persona, con el objetivo de estudiar sus rutinas e identificar posibles enfermedades como la demencia, TOCs, etc.

Por tanto, los pasos para realizar dicho reconocimiento son los siguientes:

- 1) Elegir y desplegar sensores apropiados a objetos y entornos para monitorizar y capturar el comportamiento de un usuario.
- 2) Recopilar, almacenar y procesar la información obtenida empleando técnicas de análisis de datos.
- 3) Crear modelos de actividad computacional de una manera que permita a los sistemas llegar a conclusiones razonables.
- 4) Seleccionar o desarrollar algoritmos de razonamiento para inferir actividades.

Los avances en la tecnología se han centrado principalmente en la creación de una amplia gama de dispositivos de bajo costo, de poco consumo eléctrico y de reducido tamaño, lo que se traduce en un ahorro de los recursos, ya que estos son limitados. Un ejemplo son las placas de bajo costo, como Raspberry Pi o Arduino. Estos dispositivos permiten leer la información de los sensores, además de procesar los datos que éstos envían, para extraer información relevante de los ambientes inteligentes. Sin embargo, estos dispositivos tienen una capacidad de procesamiento limitada y una baja o nula capacidad de almacenamiento, por lo que, un factor clave a tener en cuenta es reducir la complejidad computacional de cualquier tarea que deban realizar.

Los enfoques utilizados para el reconocimiento de actividades basadas en sensores se dividen en dos categorías principales:

- Enfoques basados en datos (DDA)
- Enfoques basados en conocimiento (KDA).

El primero, DDA, se basa en técnicas de aprendizaje automático en las que se requiere un conjunto de datos preexistente de comportamientos de usuario. Se lleva a cabo un proceso de aprendizaje, por lo general, para construir un modelo de actividad. A continuación, se realiza un proceso de prueba para evaluar la generalización del modelo en la clasificación de actividades que aún no han sido clasificadas [10].

Con KDA, el modelo de actividad se construye a través de la incorporación de conocimientos previos obtenidos del dominio de aplicación, utilizando la ingeniería del conocimiento y técnicas de gestión del conocimiento [11].

En este caso, nos centraremos en el primer caso, DDA, y en concreto en el algoritmo más popular dentro de esta categoría, el vecino más cercano (NN). Este algoritmo destaca por su simplicidad y buenos niveles de precisión en general. Su funcionamiento se basa en el concepto de patrones de similitud que se pueden asignar a la misma clase (en nuestro caso actividad).

Sin embargo, también tiene unos cuantos inconvenientes. Estos tienen que ver principalmente con el tamaño del conjunto de datos, ya que uno muy grande hará que se requiera un almacenamiento elevado y un buen nivel de computo por parte de la máquina donde se ejecute este algoritmo.

La complejidad del método de búsqueda lineal del NN es  $O(n \cdot d)$  donde  $n$  es el tamaño del conjunto de datos y  $d$  es la dimensionalidad, es decir, el número de sensores. En nuestro caso, este hecho tiene gran relevancia ya que, el tamaño de los datos que se manejan es bastante grande debido a la ingente cantidad de eventos que generan los sensores en un ambiente inteligente.

Por esta razón, los algoritmos de Generación de Prototipos (PG) son los adecuados, ya que, una de sus funciones principales es el reconocimiento y generación de un óptimo subconjunto de datos representativos a partir de datos originales. Esto es posible eliminando el posible ruido que pueda existir, y detectando y eliminando información redundante, para, así, reemplazar los datos originales por los que han sido generados.

Por este motivo resulta atractivo emplear algoritmos PG a la hora de generar el modelo de actividad, con el fin de maximizar las ventajas proporcionadas por el NN y evitar los inconvenientes asociados al tamaño de los conjuntos de datos en entornos con recursos limitados.

De esta forma, se emplea algoritmos PG para reducir el tamaño de los datos con el fin de disminuir los requisitos de almacenamiento y complejidad computacional a la hora de aplicar el NN, y para mantener la exactitud de la clasificación.

### 3.1.1 Algoritmos de generación de prototipos

Los algoritmos de generación de prototipos (PG) son una técnica de reducción de datos que ayuda a identificar un subconjunto óptimo del conjunto original, descartando ruido y ejemplos redundantes y modificando el valor de algunas muestras para construir ejemplos artificiales, conocidos como prototipos.

El esquema que siguen estos algoritmos es el siguiente:

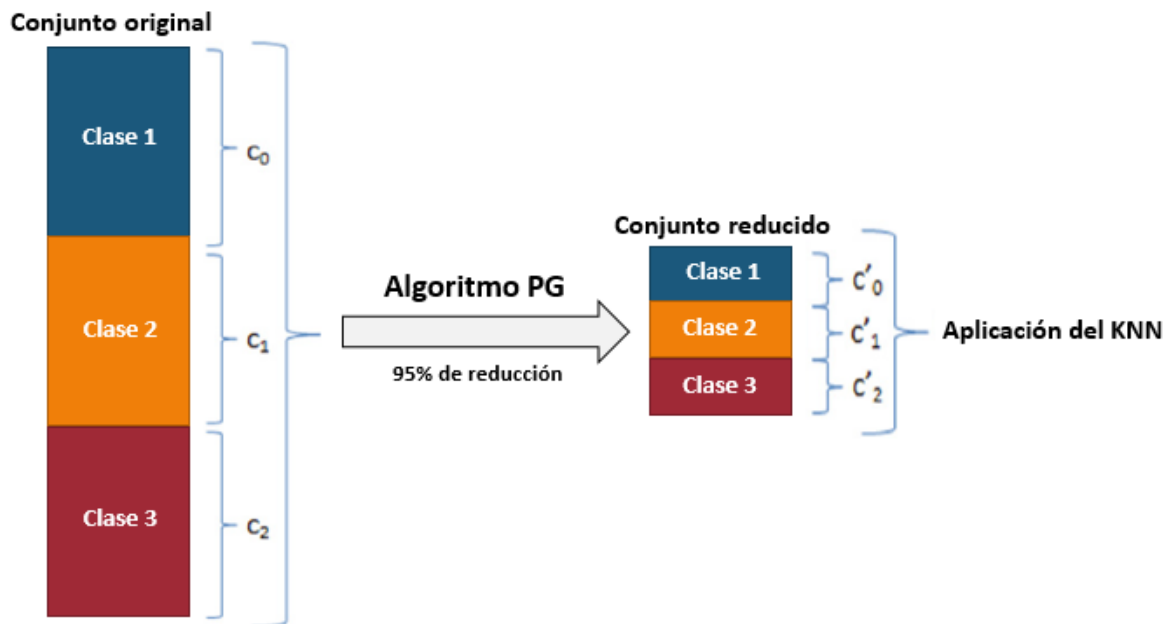


Tabla 3 - Esquema de los algoritmos de generación de prototipos

El hecho de reducir el conjunto final, tiene un gran impacto en el almacenamiento empleado y en la capacidad de computación necesarios a la hora de ejecutar el algoritmo NN. De hecho, muchos de estos algoritmos fueron diseñados para tal propósito.

Dependiendo del mecanismo de generación de prototipos se pueden distinguir cuatro mecanismos distintos:

- **Ajuste de posicionamiento:** Esta técnica corrige la posición de un subconjunto de prototipos del conjunto inicial mediante un procedimiento de optimización. Las nuevas posiciones de los prototipos se obtienen mediante el desplazamiento en un espacio imaginario multidimensional, sumando o restando algunas cantidades a los valores de cada una de las características de los prototipos [12].
- **Reetiquetado de clases:** Este mecanismo de generación de prototipos consiste en cambiar las clases de aquellas muestras que se consideran que contienen errores o que pertenecen a otra clase distinta a la cual habían sido clasificados [13].
- **Basado en centroides:** Estas técnicas están basadas en la generación artificial de prototipos unificando aquellos que pertenecen a la misma clase. Este proceso de unificación normalmente se realiza sobre la media de los valores de los atributos del conjunto escogido para unificar, produciendo los llamados centroides [14].
- **División de espacio:** Esta técnica está basada en el empleo de diferentes heurísticas para dividir el espacio disponible junto con múltiples mecanismos para crear nuevos prototipos. La idea consiste en dividir conjunto original en regiones, que serán sustituidas por muestras representativas que establezcan los límites de decisión asociados con el conjunto original [15].

Además, estos algoritmos pueden ser agrupados en 4 dependiendo del tipo de reducción:

- **Creciente:** Parte de un conjunto reducido o con algunos prototipos representativos de cada clase y realizar una reducción incremental, es decir, añadiendo prototipos que considere adecuados.
- **Decreciente:** Parte del conjunto original y conforme va avanzando, el algoritmo va reduciendo o modificando los prototipos en el conjunto original para dar como resultado el conjunto final.
- **Fijo:** La reducción fija establece el número final de prototipos que el conjunto final deberá tener. Esto se ajusta en un parámetro del algoritmo antes de su ejecución.

- **Mezclado:** Realiza una mezcla de los anteriores, partiendo de un subconjunto ya preestablecido al cual le añade, modifica o elimina prototipos.

[16]

### 3.1.2 Enfoque vecino más cercano (NN)

El enfoque del vecino más cercano es unas de las técnicas más usadas para tareas de clasificación y reconocimiento de patrones. Está basado en el concepto de similitud [17] y en el hecho de que los patrones que son similares, normalmente, pertenecen a la misma clase.

A la hora de analizar el algoritmo, se distinguen dos conjuntos:

- **Conjunto de entrenamiento (Training set):** Es el conjunto del que se sabe la clase a la que pertenece cada uno de las muestras que está formado el conjunto.
- **Conjunto de test (Test set):** Es el conjunto que se pretende etiquetar, y del que se desconoce la clase de cada muestra.

Cada una de las muestras está formada, básicamente, por un vector. En nuestro caso, cada valor corresponde a un sensor existente en el ambiente inteligente. Por tanto, este método está categorizado como aprendizaje pasivo, ya que, etiqueta a partir de las clases existentes en las muestras.

Para realizar el etiquetado del conjunto de test, se fija un valor, denominado  $k$ , que indica el número de muestras más cercanas del conjunto de entrenamiento que tiene en cuenta a la hora de clasificar. El proceso de etiquetado de un nuevo vector se realiza eligiendo la clase más frecuente entre los  $k$  vecinos más cercanos a él. Cuando el valor de  $k$  es igual a 1, la clase etiquetada es la clase del vecino más cercano.

Para calcular la cercanía de un vecino con respecto a una muestra existen multitud de enfoques y opciones, pero la más usada y la que se empleará en este proyecto es la de la distancia euclídea entre ambas muestras. Cuanto menor sea la distancia euclídea entre ellos, más cercanos están.

Sin embargo, este algoritmo a pesar de ser bastante bueno en sus resultados, tiene una serie de desventajas que son enumeradas a continuación:

- Altos requisitos en almacenamiento con el fin de almacenar el conjunto de entrenamiento.

- Alta complejidad computacional, ya que, por cada muestra a etiquetar, es necesario recorrer todo el conjunto de entrenamiento.
- Baja tolerancia al ruido, debido a que tiene en cuenta todo el conjunto de entrenamiento, incluso cuando pueda existir datos erróneos.

[18]

Debido a esto, es más que recomendable emplear algoritmos de generación de prototipos, explicados anteriormente, con el fin de evitar estos inconvenientes.

## 3.2 EXPERIMENTACIÓN PARA EL MODELADO DE RECONOCIMIENTO DE ACTIVIDADES

---

Una vez explicados los algoritmos de generación de prototipos y el KNN, es el momento de detallar las experimentaciones realizadas con el fin de obtener un modelo de datos que sea óptimo para el KNN.

Para poder realizar estas experimentaciones, he contado con un repositorio de datasets ya preexistente, que pueden ser descargados gratuitamente. El repositorio en cuestión se llama CASAS (Center for Advanced Studies in Adaptive Systems), y cuenta con 36 datasets para realizar experimentaciones. También disponía de otro conjunto de datasets previo que también fue incluido en las experimentaciones. Estos datasets, posteriormente, han sido usados en el software “KEEL”, donde ejecutando una serie de algoritmos de generación de prototipos se obtenían unos resultados que serán analizados más adelante.

Los datasets en su mayoría están compuestos por varias columnas:

- Fecha del evento
- Sensor que lanza el evento, formado por 3 caracteres
- Valor del evento (ON, OFF, Valor de temperatura, etc.)
- Comienzo/Fin de la actividad que se estaba realizando

Para que puedan ser útiles, estos han de estar etiquetados, es decir, que tengan marcadas las actividades que se realizan a lo largo del dataset. Por ello después de realizar esta comprobación, de los 36 datasets iniciales, únicamente han quedado 33.

Los datasets válidos son los siguientes:

- **Adlnormal:** Este dataset representa los eventos recogidos en el apartamento inteligente WSU en un banco de pruebas. Contiene 120 instancias que fueron generadas usando 39 sensores llegando a identificar 5 actividades.
- **Adlerror:** Este dataset representa los eventos recogidos en el apartamento inteligente WSU en un banco de pruebas. Contiene 100 instancias que fueron generadas usando 39 sensores llegando a identificar 5 actividades. (Similar al anterior, pero sin añadido de errores).
- **Adlinterweave:** Este dataset representa los eventos recogidos en un apartamento, donde se realizaban tareas simultáneamente. Contiene 167 instancias que fueron generadas usando 76 sensores llegando a identificar 7 actividades.
- **Twor - Spring 2009:** Este dataset representa los eventos de sensor recogidos en el apartamento durante la primavera de 2009. El apartamento aloja dos residentes, R1 y R2 que llevan a cabo sus actividades diarias normales. Contiene 1681 instancias que fueron generadas usando 81 sensores llegando a identificar 15 actividades.
- **Twor - Summer 2009:** Este dataset representa los eventos de sensor recogidos en el apartamento durante el verano de 2009. El apartamento aloja dos residentes, R1 y R2 que llevan a cabo sus actividades diarias normales. Contiene 743 instancias que fueron generadas usando 76 sensores llegando a identificar 11 actividades.
- **Twor 2010:** Este dataset representa los eventos de sensor recogidos en el apartamento durante el verano de 2009. El apartamento aloja dos residentes, R1 y R2 que llevan a cabo sus actividades diarias normales. Contiene 743 instancias que fueron generadas usando 92 sensores llegando a identificar 23 actividades.
- **Cairo:** Este dataset contiene los datos del sensor que se recogió en la casa de una pareja de adultos voluntarios. Los residentes en el hogar eran un hombre, una mujer y un perro. Los hijos de la pareja también visitaron la casa en al menos una ocasión. Contiene 560 instancias que fueron generadas usando 27 sensores llegando a identificar 13 actividades.
- **Milan:** Este conjunto de datos contiene los datos de los sensores que se recogieron en la casa de un adulto voluntario. Los residentes en el hogar eran una mujer y un perro. Los hijos de la mujer la visitaron en varias ocasiones. Contiene 1544 instancias que fueron generadas usando 31 sensores llegando a identificar 15 actividades.

- **Aruba:** Este conjunto de datos contiene los datos de los sensores que se recogieron en la casa de un adulto voluntario. Los residentes en el hogar era una mujer. Los hijos y nietos de la mujer la visitaron en varias ocasiones. Contiene 3392 instancias que fueron generadas usando un número indeterminado de sensores llegando a identificar 11 actividades.
- **hh102:** Este conjunto de datos contiene los datos del sensor que se recogió en la casa de un adulto. Contiene 2084 instancias que fueron generadas usando 61 sensores llegando a identificar 30 actividades.
- **hh104:** Este conjunto de datos contiene los datos del sensor que se recogió en la casa de un adulto. Contiene 3139 instancias que fueron generadas usando 60 sensores llegando a identificar 30 actividades.
- **hh107:** Este conjunto de datos contiene los datos del sensor que se recogió en la casa de un adulto. Contiene 591 instancias que fueron generadas usando 50 sensores llegando a identificar 20 actividades.
- **hh110:** Este conjunto de datos contiene los datos del sensor que se recogió en la casa de un adulto. Contiene 833 instancias que fueron generadas usando 41 sensores llegando a identificar 20 actividades.
- **hh112:** Este conjunto de datos contiene los datos del sensor que se recogió en la casa de un adulto. Contiene 4345 instancias que fueron generadas usando 37 sensores llegando a identificar 27 actividades.
- **hh113:** Este conjunto de datos contiene los datos del sensor que se recogió en la casa de un adulto. Contiene 22015 instancias que fueron generadas usando 61 sensores llegando a identificar 30 actividades.
- **hh115:** Este conjunto de datos contiene los datos del sensor que se recogió en la casa de un adulto. Contiene 33367 instancias que fueron generadas usando 49 sensores llegando a identificar 34 actividades.
- **hh116:** Este conjunto de datos contiene los datos del sensor que se recogió en la casa de un adulto. Contiene 1832 instancias que fueron generadas usando 23 sensores llegando a identificar 30 actividades.
- **hh117:** Este conjunto de datos contiene los datos del sensor que se recogió en la casa de un adulto. Contiene 9039 instancias que fueron generadas usando 37 sensores llegando a identificar 30 actividades.



- **hh118:** Este conjunto de datos contiene los datos del sensor que se recogió en la casa de un adulto. Contiene 1601 instancias que fueron generadas usando 25 sensores llegando a identificar 30 actividades.
- **hh120:** Este conjunto de datos contiene los datos del sensor que se recogió en la casa de un adulto. Contiene 2280 instancias que fueron generadas usando 40 sensores llegando a identificar 30 actividades.
- **hh122:** Este conjunto de datos contiene los datos del sensor que se recogió en la casa de un adulto. Contiene 1368 instancias que fueron generadas usando 47 sensores llegando a identificar 30 actividades.
- **hh123:** Este conjunto de datos contiene los datos del sensor que se recogió en la casa de un adulto. Contiene 167 instancias que fueron generadas usando 24 sensores llegando a identificar 25 actividades.
- **hh124:** Este conjunto de datos contiene los datos del sensor que se recogió en la casa de un adulto. Contiene 167 instancias que fueron generadas usando 39 sensores llegando a identificar 20 actividades.
- **Apartment:** Este conjunto de datos representa los eventos de sensor recogidos en un banco de pruebas inteligente durante varios días de junio de 2008. Uno de los participantes vivía en el apartamento durante este tiempo. Contiene 35 instancias que fueron generadas usando 51 sensores llegando a identificar 4 actividades.
- **Tulum 2009:** Este conjunto de datos representa los eventos de sensor recogidos en el apartamento de abril a julio del 2009. El apartamento aloja dos residentes, R1 y R2 que llevan a cabo sus actividades diarias normales. Contiene 1307 instancias que fueron generadas usando 18 sensores llegando a identificar 10 actividades.
- **Tulum 2010:** Este conjunto de datos representa los eventos de sensor recogidos en el apartamento durante el año académico 2009-2010. El apartamento aloja dos residentes que llevan a cabo sus actividades diarias normales. Contiene 10402 instancias que fueron generadas usando 31 sensores llegando a identificar 15 actividades.
- **VanKasteren:** Este dataset fue creado en una casa y contiene 245 instancias que fueron generadas a partir de 14 sensores binarios identificando 7 tipos de actividades.

- **CASAS:** Este dataset fue recolectado del ambiente inteligente de la Universidad de Washington y contiene 121 instancias que fueron generadas usando 39 sensores llegando a identificar 5 actividades.
- **CASAS\_ERROR:** Este dataset fue recolectado del ambiente inteligente de la Universidad de Washington y contiene 100 instancias que fueron generadas usando 39 sensores llegando a identificar 5 actividades. (Similar al anterior, pero con errores)
- **ODI\_Ulster1 y ODI\_Ulster2:** Estos datasets fueron generados a partir de una herramienta de simulación de ambientes inteligentes. El primer ODI contiene 308 instancias generadas a partir de 21 sensores binarios dando lugar a un total de 11 tipos de actividades. El segundo ODI contiene 616 instancias generadas también con 21 sensores binarios y con el mismo tipo de actividades que el primero.
- **ULSTER2015:** Contiene 364 instancias que fueron generadas usando 15 sensores llegando a identificar 9 actividades.
- **UJA2016:** Contiene 131 instancias que fueron generadas usando 25 sensores llegando a identificar 25 actividades.

El siguiente paso ha sido realizar un pre-procesamiento de los mismos, ya que casi ninguno disponía del formato necesario para que KEEL los reconozca. Este pre-procesamiento ha consistido en crear tantos ficheros CSV (Comma-Separated Values) como datasets existentes, en los que cada una de las instancias del fichero es un vector de tamaño  $d + 1$  componentes. Las  $d$  primeras corresponden al valor de los  $d$  sensores existentes en el ambiente inteligente. La última componente,  $d + 1$ , corresponde a la actividad realizada. A continuación, se muestra un ejemplo:

```
0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,guest_bathroom
0,0,1,0,0,0,0,0,0,0,0,1,1,1,0,1,1,1,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,kitchen_activity
1,1,1,1,1,0,0,0,0,0,0,0,1,0,1,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,read
```

De esta forma, ya puede ser leído por el software KEEL.

### 3.2.1 Experimentación con distintos algoritmos de generación de prototipos

Aunque el número de algoritmos de generación de prototipos era superior, se han considerado 15, mostrados en la tabla de a continuación, puesto que el resto u obtenían resultados muy pésimos o no funcionaban correctamente.

El objetivo de realizar esta experimentación es identificar el algoritmo que más reduce la complejidad computacional a la hora de realizar el reconocimiento de actividades basados en sensores.

	<i>Algo ritmo</i>	<i>Mecanismo de generación de prototipos</i>	<i>Tipo de reducción</i>
3	LVQ	Ajuste de posicionamiento	Fija
M	DS	Ajuste de posicionamiento	Fija
	VQ	Ajuste de posicionamiento	Fija
	LVQ	Ajuste de posicionamiento	Mezclada
TC	MS	Ajuste de posicionamiento	Mezclada
	HYB	Ajuste de posicionamiento	Mezclada
E	LVQ	Ajuste de posicionamiento	Fija
	PSC	Ajuste de posicionamiento	Mezclada
SA	ENP	Ajuste de posicionamiento	Mezclada
	BTS	Basado en centroides	Decreciente
3			

Tabla 4 – Algoritmos escogidos para la experimentación

Cada algoritmo requiere un conjunto de parámetros. Los valores de los parámetros han sido sacados de [19], gracias a los exitosos resultados obtenidos en experimentaciones previas.

Dado que todos los datasets están etiquetados con la actividad, somos capaces de calcular la precisión del subconjunto de prototipos obtenido de cada algoritmo. Para calcular esta precisión se usa el dataset completo y se evalúa usando el KNN para un prototipo dado. Esta precisión viene a indicar la proporción de resultados acertados entre el número total de clases examinadas.

El conjunto de algoritmos mostrados anteriormente en la Tabla 4, han sido ejecutados usando el software KEEL, como se mencionó previamente. Este software es una herramienta de código abierto con que emplea técnicas basadas en aprendizaje evolutivo y soft-computing para distintos tipos de problemas de minería de datos. Para evaluar el rendimiento de cada uno de los algoritmos se ha empleado un Cross-Validation de 10-folds. La principal ventaja de esta validación es que todas las instancias del dataset son empleadas tanto para el conjunto de entrenamiento como para el de test [20]. De esta manera la división de los datos no tiene tanta importancia.

A continuación, se muestra una serie de tablas con los resultados obtenidos en cada uno de algoritmos por cada dataset:

<i>Algoritmo</i>	<i>Dataset</i>	<i>Precisión</i>	<i>Reducción</i>
LVQ3	Adlerror	90	95
LVQ3	Adlinterwave	88,08823529	95
LVQ3	Adlnormal	94,16666667	95
LVQ3	Aparment	68,33333333	95
LVQ3	Aruba	88,44421308	95
LVQ3	Cairo	57,14285714	95
LVQ3	Hh102	39,1576187	95
LVQ3	Hh104	40,01139578	95
LVQ3	Hh107	41,45480226	95
LVQ3	Hh110	42,61761331	95
LVQ3	Hh112	35,19084697	95
LVQ3	Hh113	41,2349503	95
LVQ3	Hh115	30,02385595	95
LVQ3	Hh116	42,13946306	95
LVQ3	Hh117	29,25132792	95
LVQ3	Hh118	47,28610248	95
LVQ3	Hh120	38,90350877	95
LVQ3	Hh122	36,40510949	95

LVQ3	Hh123	26,72243952	95
LVQ3	Hh124	48,67647059	95
LVQ3	Milan	65,08713867	95
LVQ3	Tulum2009	76,4386377	95
LVQ3	Tulum2010	63,69919826	95
LVQ3	Twor.2009	63,1254902	95
LVQ3	Twor.2010	83,34636517	95
LVQ3	Twor.summer.2009	72,28108108	95
LVQ3	Casas	95,83333333	95
LVQ3	Odi_Ulster1	49,92473118	95
LVQ3	Odi_Ulster2	59,55843469	95
LVQ3	VanKasteren	82,7	95
LVQ3	CASAS_Error	86	95
LVQ3	UJA2016	72,41758242	95
LVQ3	ULSTER2015	70,10510511	95

Tabla 5 – Experimentación con algoritmo LVQ3

<i>Algoritmo</i>	<i>Dataset</i>	<i>Precisión</i>	<i>Reducción</i>
DSM	Adlerror	89	95
DSM	Adlinterwave	88,05147059	95
DSM	Adlnormal	87,5	95
DSM	Aparment	74,16666667	95
DSM	Aruba	87,8539823	95
DSM	Cairo	57,5	95
DSM	Hh102	36,99760766	95
DSM	Hh104	34,85327934	95

DSM	Hh107	39,60451977	95
DSM	Hh110	38,52553069	95
DSM	Hh112	28,92859791	95
DSM	Hh113	38,62420313	95
DSM	Hh115	27,4043085	95
DSM	Hh116	39,89843193	95
DSM	Hh117	30,14711287	95
DSM	Hh118	40,72593168	95
DSM	Hh120	31,84210526	95
DSM	Hh122	31,58490769	95
DSM	Hh123	27,21787629	95
DSM	Hh124	55,69852941	95
DSM	Milan	63,27314621	95
DSM	Tulum2009	71,30299472	95
DSM	Tulum2010	52,56692899	95
DSM	Twor.2009	56,74901961	95
DSM	Twor.2010	84,29874613	95
DSM	Twor.summer.2009	68,77297297	95
DSM	Casas	91,66666667	95
DSM	Odi_Ulster1	43,43010753	95
DSM	Odi_Ulster2	47,8979376	95
DSM	VanKasteren	72,61666667	95
DSM	CASAS_Error	79	95
DSM	UJA2016	70,16483516	95
DSM	ULSTER2015	29,15165165	95

Tabla 6 - Experimentación con algoritmo DSM

<i>Algoritmo</i>	<i>Dataset</i>	<i>Precisión</i>	<i>Reducción</i>
VQ	Adlerror	86	95
VQ	Adlinterwave	84,41176471	95
VQ	Adlnormal	85,83333333	95
VQ	Aparment	59,16666667	95
VQ	Aruba	89,71186882	95
VQ	Cairo	53,92857143	95
VQ	Hh102	41,12555208	95
VQ	Hh104	39,34311471	95
VQ	Hh107	45,85310734	95
VQ	Hh110	43,81956397	95
VQ	Hh112	39,885852	95
VQ	Hh113	43,00399483	95
VQ	Hh115	32,79001372	95
VQ	Hh116	43,34372773	95
VQ	Hh117	33,67619244	95
VQ	Hh118	46,78338509	95
VQ	Hh120	42,28070175	95
VQ	Hh122	37,86764706	95
VQ	Hh123	27,46994061	95
VQ	Hh124	40,36764706	95
VQ	Milan	64,70004189	95
VQ	Tulum2009	68,86024662	95
VQ	Tulum2010	54,57606407	95
VQ	Twor.2009	58,54117647	95
VQ	Twor.2010	82,75077487	95
VQ	Twor.summer.2009	73,47567568	95

VQ	Casas	88,33333333	95
VQ	Odi_Ulster1	56,13978495	95
VQ	Odi_Ulster2	57,78159704	95
VQ	VanKasteren	72,26666667	95
VQ	CASAS_Error	86	95
VQ	UJA2016	70,93406593	95
VQ	ULSTER2015	78,10810811	95

Tabla 7 - Experimentación con algoritmo VQ

<i>Algoritmo</i>	<i>Dataset</i>	<i>Precisión</i>	<i>Reducción</i>
LVQTC	Adlerror	95	95
LVQTC	Adlinterwave	92,79411765	95
LVQTC	Adlnormal	94,16666667	95
LVQTC	Aparment	79,16666667	95
LVQTC	Aruba	84,34556655	95
LVQTC	Cairo	56,78571429	95
LVQTC	Hh102	38,72216599	95
LVQTC	Hh104	36,76359862	95
LVQTC	Hh107	45,3559322	95
LVQTC	Hh110	47,90590935	95
LVQTC	Hh112	46,44345569	95
LVQTC	Hh113	45,44668472	95
LVQTC	Hh115	46,18079196	95
LVQTC	Hh116	41,97760751	95
LVQTC	Hh117	43,40075853	95
LVQTC	Hh118	46,72166149	95



LVQTC	Hh120	45,74561404	95
LVQTC	Hh122	37,86335337	95
LVQTC	Hh123	34,00985079	95
LVQTC	Hh124	60,51470588	95
LVQTC	Milan	62,37494763	95
LVQTC	Tulum2009	73,07222548	95
LVQTC	Tulum2010	56,46991613	95
LVQTC	Twor.2009	70,52156863	95
LVQTC	Twor.2010	82,2749366	95
LVQTC	Twor.summer.2009	69,04684685	95
LVQTC	Casas	94,16666667	95
LVQTC	Odi_Ulster1	69,38709677	95
LVQTC	Odi_Ulster2	55,510312	95
LVQTC	VanKasteren	88,5	95
LVQTC	CASAS_Error	95	95
LVQTC	UJA2016	77,8021978	95
LVQTC	ULSTER2015	89,95495495	95

Tabla 8 - Experimentación con algoritmo LVQTC

Algoritmo	Dataset	Precisión	Reducción
MSE	Adlerror	94	46
MSE	Adlinterwave	94,04411765	47
MSE	Adlnormal	93,33333333	55
MSE	Aparment	51,66666667	45
MSE	Aruba	92,06940829	99

<i>MSE</i>	Cairo	64,46428571	86
<i>MSE</i>	Hh102	46,5458226	98
<i>MSE</i>	Hh104	43,48263161	98
<i>MSE</i>	Hh107	51,77966102	95
<i>MSE</i>	Hh110	44,9168101	95
<i>MSE</i>	Hh112	49,43381535	99
<i>MSE</i>	Hh113	50,82984502	96
<i>MSE</i>	Hh115	44,86767344	97
<i>MSE</i>	Hh116	49,39831314	97
<i>MSE</i>	Hh117	52,24016062	99
<i>MSE</i>	Hh118	53,2189441	97
<i>MSE</i>	Hh120	47,36842105	99
<i>MSE</i>	Hh122	40,71167883	98
<i>MSE</i>	Hh123	35,62436622	99
<i>MSE</i>	Hh124	45,58823529	92
<i>MSE</i>	Milan	69,43443653	94
<i>MSE</i>	Tulum2009	74,36758661	97
<i>MSE</i>	Tulum2010	67,76576701	100
<i>MSE</i>	Twor.2009	73,30196078	88
<i>MSE</i>	Twor.2010	88,93420682	99
<i>MSE</i>	Twor.summer.2009	76,57837838	92
<i>MSE</i>	Casas	93,33333333	63
<i>MSE</i>	Odi_Ulster1	82,48387097	94
<i>MSE</i>	Odi_Ulster2	80,85933369	97
<i>MSE</i>	VanKasteren	89,33333333	93
<i>MSE</i>	CASAS_Error	92	68
<i>MSE</i>	UJA2016	62,63736264	75

<i>MSE</i>	ULSTER2015	96,4039039	95
------------	------------	------------	----

Tabla 9 - Experimentación con algoritmo MSE

<i>Algoritmo</i>	<i>Dataset</i>	<i>Precisión</i>	<i>Reducción</i>
<i>HYB</i>	Adlerror	91	95
<i>HYB</i>	Adlinterwave	94,63235294	95
<i>HYB</i>	Adlnormal	92,5	95
<i>HYB</i>	Aparment	77,5	95
<i>HYB</i>	Aruba	90,77225403	95
<i>HYB</i>	Cairo	62,67857143	95
<i>HYB</i>	Hh102	42,2757177	95
<i>HYB</i>	Hh104	38,64441098	95
<i>HYB</i>	Hh107	50,09039548	95
<i>HYB</i>	Hh110	50,78600115	95
<i>HYB</i>	Hh112	46,44303194	95
<i>HYB</i>	Hh113	43,53290315	95
<i>HYB</i>	Hh115	22,00381469	95
<i>HYB</i>	Hh116	49,45236398	95
<i>HYB</i>	Hh117	46,5533644	95
<i>HYB</i>	Hh118	49,72127329	95
<i>HYB</i>	Hh120	49,60526316	95
<i>HYB</i>	Hh122	37,71683126	95
<i>HYB</i>	Hh123	40,64247429	95
<i>HYB</i>	Hh124	53,45588235	95
<i>HYB</i>	Milan	67,09635526	95
<i>HYB</i>	Tulum2009	71,38990018	95

<i>HYB</i>	Tulum2010	61,49773701	95
<i>HYB</i>	Twor.2009	73,3254902	95
<i>HYB</i>	Twor.2010	88,51824458	95
<i>HYB</i>	Twor.summer.2009	75,9045045	95
<i>HYB</i>	Casas	93,33333333	95
<i>HYB</i>	Odi_Ulster1	68,51612903	95
<i>HYB</i>	Odi_Ulster2	72,08884188	95
<i>HYB</i>	VanKasteren	93,85	95
<i>HYB</i>	CASAS_Error	90	95
<i>HYB</i>	UJA2016	70,93406593	95
<i>HYB</i>	ULSTER2015	99,16666667	95

Tabla 10 - Experimentación con algoritmo HYB

<i>Algoritmo</i>	<i>Dataset</i>	<i>Precisión</i>	<i>Reducción</i>
<i>LVQPRU</i>	Adlerror	93	95
<i>LVQPRU</i>	Adlinterwave	94,63235294	95
<i>LVQPRU</i>	Adlnormal	94,16666667	95
<i>LVQPRU</i>	Aparment	54,16666667	95
<i>LVQPRU</i>	Aruba	91,18558043	95
<i>LVQPRU</i>	Cairo	60,35714286	95
<i>LVQPRU</i>	Hh102	43,66534781	95
<i>LVQPRU</i>	Hh104	38,51569972	95
<i>LVQPRU</i>	Hh107	50,42655367	95
<i>LVQPRU</i>	Hh110	38,7751004	95
<i>LVQPRU</i>	Hh112	47,38831506	95
<i>LVQPRU</i>	Hh113	48,47326306	95

LVQPRU	Hh115	37,87879017	95
LVQPRU	Hh116	49,23408173	95
LVQPRU	Hh117	47,71576309	95
LVQPRU	Hh118	53,15411491	95
LVQPRU	Hh120	47,36842105	95
LVQPRU	Hh122	41,44750966	95
LVQPRU	Hh123	39,69288715	95
LVQPRU	Hh124	53,27205882	95
LVQPRU	Milan	66,06409719	95
LVQPRU	Tulum2009	74,98062243	95
LVQPRU	Tulum2010	65,80465344	95
LVQPRU	Twor.2009	68,12156863	95
LVQPRU	Twor.2010	85,24936602	95
LVQPRU	Twor.summer.2009	74,42882883	95
LVQPRU	Casas	96,66666667	95
LVQPRU	Odi_Ulster1	84,39784946	95
LVQPRU	Odi_Ulster2	84,58487573	95
LVQPRU	VanKasteren	93,81666667	95
LVQPRU	CASAS_Error	94	95
LVQPRU	UJA2016	80,16483516	95
LVQPRU	ULSTER2015	99,16666667	95

Tabla 11 - Experimentación con algoritmo LVQPRU

Algoritmo	Dataset	Precisión	Reducción
PSCSA	Adlerror	88	94
PSCSA	Adlinterwave	80,91911765	94

PSCSA	Adlnormal	90,83333333	95
PSCSA	Aparment	60,83333333	87
PSCSA	Aruba	81,54823876	99
PSCSA	Cairo	41,96428571	97
PSCSA	Hh102	29,32278248	98
PSCSA	Hh104	29,08569219	97
PSCSA	Hh107	42,13841808	96
PSCSA	Hh110	32,39816408	97
PSCSA	Hh112	35,18740399	98
PSCSA	Hh113	0	
PSCSA	Hh115	0	
PSCSA	Hh116	41,97344975	98
PSCSA	Hh117	0	
PSCSA	Hh118	35,53920807	97
PSCSA	Hh120	36,75438596	98
PSCSA	Hh122	35,38589523	98
PSCSA	Hh123	31,2921918	98
PSCSA	Hh124	55,73529412	86
PSCSA	Milan	36,9124424	98
PSCSA	Tulum2009	45,91837933	99
PSCSA	Tulum2010	0	
PSCSA	Twor.2009	52,59607843	97
PSCSA	Twor.2010	66,43949	99
PSCSA	Twor.summer.2009	54,21621622	98
PSCSA	Casas	86,66666667	95
PSCSA	Odi_Ulster1	94,11827957	96
PSCSA	Odi_Ulster2	91,22686409	98

PSCSA	VanKasteren	95,91666667	96
PSCSA	CASAS_Error	87	94
PSCSA	UJA2016	68,07692308	91
PSCSA	ULSTER2015	99,72972973	97

Tabla 12 - Experimentación con algoritmo PSCSA

Algoritmo	Dataset	Precisión	Reducción
ENPC	Adlerror	92	85
ENPC	Adlinterwave	93,41911765	86
ENPC	Adlnormal	90	85
ENPC	Aparment	71,66666667	67
ENPC	Aruba	91,18505986	96
ENPC	Cairo	61,25	71
ENPC	Hh102	42,99549135	89
ENPC	Hh104	46,25811441	89
ENPC	Hh107	47,89548023	78
ENPC	Hh110	51,026965	70
ENPC	Hh112	51,7841517	94
ENPC	Hh113	58,40764503	93
ENPC	Hh115	0	
ENPC	Hh116	52,29240912	87
ENPC	Hh117	0	
ENPC	Hh118	48,09704969	95
ENPC	Hh120	0	
ENPC	Hh122	45,53563761	96
ENPC	Hh123	34,93553527	95
ENPC	Hh124	46,10294118	82
ENPC	Milan	69,36614998	72

ENPC	Tulum2009	74,21961245	84
ENPC	Tulum2010	65,39100717	88
ENPC	Twor.2009	71,89803922	79
ENPC	Twor.2010	95,12292195	94
ENPC	Twor.summer.2009	76,44504505	78
ENPC	Casas	93,33333333	89
ENPC	Odi_Ulster1	65,91397849	93
ENPC	Odi_Ulster2	68,02221047	96
ENPC	VanKasteren	85,28333333	95
ENPC	CASAS_Error	89	88
ENPC	UJA2016	71,7032967	78
ENPC	ULSTER2015	99,16666667	97

Tabla 13 - Experimentación con algoritmo ENPC

Algoritmo	Dataset	Precisión	Reducción
BTS3	Adlerror	20	95
BTS3	Adlinterwave	11,98529412	95
BTS3	Adlnormal	20,83333333	95
BTS3	Aparment	31,66666667	95
BTS3	Aruba	44,83966684	95
BTS3	Cairo	52,32142857	95
BTS3	Hh102	7,342887376	95
BTS3	Hh104	9,844020268	95
BTS3	Hh107	0,84180791	95
BTS3	Hh110	11,40275387	95
BTS3	Hh112	6,651305684	95



BTS3	Hh113	5,012350833	95
BTS3	Hh115	10,49520348	95
BTS3	Hh116	12,06313851	95
BTS3	Hh117	3,661859681	95
BTS3	Hh118	3,872282609	95
BTS3	Hh120	10,30701754	95
BTS3	Hh122	8,55302705	95
BTS3	Hh123	2,717658989	95
BTS3	Hh124	9,522058824	95
BTS3	Milan	63,14578969	95
BTS3	Tulum2009	65,41573693	95
BTS3	Tulum2010	57,81587601	95
BTS3	Twor.2009	19,91372549	95
BTS3	Twor.2010	46,51979431	95
BTS3	Twor.summer.2009	3,632432432	95
BTS3	Casas	20,83333333	95
BTS3	Odi_Ulster1	9,075268817	95
BTS3	Odi_Ulster2	9,085140137	95
BTS3	VanKasteren	13,93333333	95
BTS3	CASAS_Error	20	95
BTS3	UJA2016	9,89010989	95
BTS3	ULSTER2015	65,90840841	95

Tabla 14 - Experimentación con algoritmo BTS3

Estas tablas muestran los resultados de las experimentaciones realizadas con los distintos algoritmos junto con el enfoque KNN siendo  $K = 1$ . Se escogió este valor de  $k$  porque proporciona baja tolerancia al ruido.

A continuación, en la Tabla 15, se puede observar la media obtenida para cada uno de los algoritmos empleados.

<i>Algoritmo</i>	<i>Media precisión</i>	<i>Media reducción</i>
<i>LVQ3</i>	59,5687245	95
<i>DSM</i>	55,0611132	95
<i>VQ</i>	58,45849033	95
<i>LVQTC</i>	63,73916641	95
<i>MSE</i>	66,45508073	87,63636364
<i>HYB</i>	65,3220659	95
<i>LVQPRU</i>	66,4231216	95
<i>PSCSA</i>	59,57685968	96,03448276
<i>ENPC</i>	68,32392865	86,3
<i>BTS3</i>	20,88190033	95

*Tabla 15 - Media de precisión y reducción de las experimentaciones*

Los datos analizados son el porcentaje de precisión junto con el porcentaje de reducción en términos de complejidad computacional para cada algoritmo de GP y dataset. Por tanto, la reducción representa el porcentaje de instancias que existen en el modelo de actividad del conjunto original. Por ejemplo, en el ODI\_Ulster2 contiene 616 eventos, si el conjunto es reducido al 95%, el conjunto de prototipos generados es de tan solo 31 eventos que pasan a ser evaluado por el KNN. Además, por cada algoritmo de GP se establece la media de ambos valores, precisión y reducción.

Con esto conseguimos nuestro objetivo, reducir el tamaño del conjunto original, de manera que se reduce la complejidad computacional cuando se esté clasificando una nueva actividad basada en sensores binarios mediante el KNN, con  $K = 1$ .

Sin embargo, existen algoritmos en los que la reducción del conjunto final implica una reducción de la precisión, por ello, no son capaces de reducir hasta el 95%, que es el valor ideal de reducción. En estos casos, en caso de reducir más de lo apropiado, se generarán inconsistencias, incoherencias o ruido que empeorarán el resultado a la hora de clasificar actividades.

Analizando los resultados obtenidos, podemos deducir que algoritmos como PSCSA o ENPC obtienen resultados bastantes buenos en la mayoría de datasets (sin contar datasets que no funcionaron).

### 3.2.2 Elección del algoritmo de GP para el ambiente inteligente

Una vez realizado la experimentación con un conjunto amplio de datasets, nos centraremos en el dataset de la UJA, basado en el ambiente inteligente de la Universidad de Jaén. Una vez más, realizaremos una experimentación con los algoritmos estudiados anteriormente para obtener el mejor algoritmo de generación de prototipos con este dataset.

<i>Algoritmo</i>	<i>Precisión (%)</i>	<i>Reducción (%)</i>
<i>LVQ3</i>	82,7	95
<i>DSM</i>	72,61666667	95
<i>VQ</i>	72,26666667	95
<i>LVQTC</i>	88,5	95
<i>MSE</i>	89,33333333	93
<i>HYB</i>	93,85	95
<i>LVQPRU</i>	93,81666667	95
<b><i>PSCSA</i></b>	<b>95,91666667</b>	<b>96</b>
<i>ENPC</i>	85,28333333	95
<i>BTS3</i>	13,93333333	95

Tabla 16 - Tabla de algoritmos para el dataset UJA

Como se puede observar en la Tabla 16, el algoritmo PSCSA es el que mejor puntuación obtiene con respecto al resto, además de ofrecer un 96% de reducción del tamaño del dataset, por lo que se decide usar este para generar el modelo de actividad que servirá posteriormente para el algoritmo KNN.

A continuación, se ha realizado una prueba para comprobar la eficacia del modelo de actividad obtenido, empleando el algoritmo KNN.

De esta prueba, se ha obtenido un total de 167 actividades correctas de 209, lo que equivale a un 80% de acierto. A continuación, se muestra una matriz de confusión donde se puede comprobar dichos resultados:

	Salir de casa	Dormir	Beber	Preparar desayuno	Ir al WC	Preparar cena
Salir de casa	27	0	5	1	1	0
Dormir	0	4	0	0	18	0
Beber	0	0	17	0	2	1
Preparar desayuno	0	0	0	14	0	6
Ir al WC	0	2	2	0	99	0
Preparar la cena	0	0	1	0	3	6

*Tabla 17 - Matriz de confusión de las actividades reconocidas*

En la tabla 17, las filas determinan las actividades que realizó, y las columnas las actividades que se predijeron.

Es normal que los resultados obtenidos no sean totalmente óptimos ya que, no se dispone un algoritmo que esté optimizado para el dataset empleado, aun así, los resultados son más que aceptables y deseables.



# CAPÍTULO 4

## APLICACIÓN MÓVIL DE MONITORIZACIÓN DE AMBIENTES INTELIGENTES

---

## 4.1 CONTEXTO DE LA APLICACIÓN MÓVIL

---

Una vez explicado todo el proceso teórico de reconocimiento de actividades, por el cual se identifica qué tarea está realizando un usuario en un ambiente inteligente, es el momento de entrar en profundidad en el desarrollo de este proyecto.

A grandes rasgos, el proyecto consiste en el desarrollo de una aplicación móvil desde la que se monitoriza un ambiente inteligente, en este caso, el laboratorio 109 del edificio C6 de la Universidad de Jaén. Junto a esta monitorización también se incluye un reconocimiento de las actividades dentro de ese ambiente inteligente, motivo por el cual se ha explicado el funcionamiento del mismo en los capítulos anteriores. Este proyecto consta de varios pilares básicos sobre los que se sustenta:

- **Raspberry Pi:** Dispositivo que recoge los eventos producidos por los sensores y los almacena en una base de datos.
- **Base de datos:** Lugar de almacenamiento de los eventos recogidos por la Raspberry Pi.
- **Servicio web:** Interfaz que sirve de comunicación entre la base de datos y la aplicación.
- **Aplicación móvil:** Lugar donde se visualizan todos los eventos y actividades almacenadas en la base de datos.

En forma de esquema, quedaría de la siguiente manera:





*Ilustración 15 - Esquema del sistema*

En este caso, una vez abordado el despliegue de sensores, de la Raspberry, y la implementación del reconocimiento de actividades, es el momento de detallar el desarrollo de la aplicación móvil y del servidor web, que será el lugar donde se almacenen todos los eventos recogidos por la Raspberry.

Para ello, haremos uso de la Ingeniería del Software. Ésta se compone de las siguientes actividades que deben ser llevadas a cabo:

- **Especificación de requisitos:** Se obtiene el propósito del sistema, así como las características y restricciones impuestas sobre el mismo.
- **Análisis del sistema:** Se obtiene un modelo del sistema completo, consistente, claro y verificable.
- **Diseño del sistema:** Se establecen los objetivos del proyecto y las estrategias a llevar a cabo para conseguirlos.
- **Implementación:** Consiste en la traducción del modelo lógico del sistema a código fuente.
- **Pruebas:** Consiste en verificar y validar el sistema.

## 4.2 ESPECIFICACIÓN DE REQUISITOS

---

El primer paso en el proceso de Ingeniería del Software debe ser determinar el propósito del proyecto, las propiedades que debe satisfacer y las restricciones a las que ha sido impuesto. Este es, sin



duda, un paso de vital importancia dentro del desarrollo de un proyecto software, ya que sin conocer el propósito del mismo ni establecer las diferentes limitaciones que debe afrontar, será realmente difícil realizar una aplicación software capaz de cumplir dicho propósito.

En nuestro caso el propósito que se desea conseguir es la realización de un sistema de bajo coste que realice una monitorización de un ambiente inteligente e incorpore reconocimiento de actividades realizadas dentro del mismo.

Este propósito no se puede alcanzar sin antes haber especificado los requisitos del mismo. Estos se definen como el conjunto de propiedades o restricciones, que, detallados con gran precisión, ha de satisfacer el sistema. Existen dos tipos requisitos: funcionales y no funcionales.

#### 4.2.1 Requisitos funcionales

Son aquellos requisitos que describen detalladamente las funcionalidades que nuestro sistema ha de proporcionar a los usuarios para alcanzar el propósito por el cual ha sido diseñado.

En nuestro caso un usuario esperaría de nuestra aplicación las siguientes funcionalidades:

- **Monitorizar en tiempo real el ambiente inteligente:** Se debe de actualizar la lista de eventos cada vez que se registre un evento nuevo.
- **Ver el histórico de eventos pasados:** Se debe de mostrar una lista con todos los eventos ocurridos, permitiendo filtrar por último día, última semana, último mes o una fecha en concreto.
- **Mostrar las actividades realizadas en el ambiente inteligente:** Se debe de realizar un listado de todas las actividades realizadas a lo largo del tiempo.
- **Posibilidad de notificar al usuario de nuevos eventos:** Se debe de avisar al usuario mediante una notificación, si él lo desea activando o desactivando cada sensor, de que ha ocurrido un evento nuevo.
- **Filtrar los eventos por sensor:** Se debe mostrar únicamente los eventos que un sensor ha producido.

Adicionalmente, podría resultar deseable ser capaces añadir, modificar y/o eliminar sensores, pero es algo que se sale de los propósitos de este proyecto y por tanto se ha decidido no implementarlo.

## 4.2.2 Requisitos no funcionales

Los requisitos no funcionales no están relacionados con las funcionalidades del sistema en sí, sino que son los requisitos que ha de cumplir el sistema para poder llevar a cabo los requisitos funcionales. Son tan importantes como los funcionales y en muchos casos pueden ser críticos para que el sistema funcione correctamente.

Estos requisitos están orientados a aspectos de software, hardware y otro tipo de necesidades como puedan ser la definición de la interfaz que usará dicho sistema. Estos estándares son normalmente impuestos por el cliente. En nuestro caso nos ajustaremos a lo que se nos ha pedido en este trabajo.

### 4.2.2.1 *Requisitos hardware*

Son los requisitos que han de cumplir tanto la parte del cliente como la parte del servidor para poder procesar y mostrar la información en la aplicación móvil. Por tanto, se distinguen dos tipos:

- **Requisitos del dispositivo móvil:** Para que esta aplicación pueda ejecutarse correctamente el usuario ha de disponer de un dispositivo con el sistema operativo Lolipop 5.0 o superior. También como requisito imprescindible, es necesaria una conexión a internet con el fin de obtener toda la información desde el servidor web. Como requisito deseable, es aconsejable que tenga como mínimo una pantalla de 4.5" para que la información se muestre adecuadamente.
- **Requisitos del servidor web:** Para que el cliente pueda tener una navegación deseable en su aplicación móvil y pueda acceder a todo el contenido sin retardos, es imprescindible que el servidor cuente con los siguientes aspectos:
  - Equipo lo suficientemente potente para soportar la carga de usuarios que accederán a nuestro sistema. También ha de tener una conexión de Internet para soportar todo el tráfico que genere el afluente de personas que accedan a la aplicación y soliciten datos, sobretodo priorizando una elevada cuota de subida, ya que en estos casos es más importante que la cuota de bajada.
  - Deberá de contar con una cantidad de RAM generosa, para poder gestionar con rapidez y eficacia la carga de las distintas peticiones a la BBDD. Estimo que con unos 4 GB sería más que suficiente para mantener estable tanto el SO, como las aplicaciones que hacen posible la ejecución del proyecto.

- Es conveniente, pero no imprescindible contar con un procesador de gama media/alta para que no haya problemas a la hora de procesar todas las peticiones y pueda desenvolverse con soltura en todos los procesos implicados. Sería conveniente que fuera multi-núcleo para poder ejecutar varias tareas concurrentemente y así acelerar la rapidez de cómputo.
- En el apartado gráfico, no es para nada necesario, tan solo una gráfica que permita mostrar la interfaz de usuario del Sistema Operativo instalado.
- En cuanto al almacenamiento disponible, es algo a no tener muy en cuenta, ya que en nuestro caso no almacenamos más allá de los eventos registrados por los sensores.
- En lo referente a la pantalla, cualquiera disponible es suficiente, ya que solo se necesitaría para la instalación y para tareas de mantenimiento u optimización.
- También es conveniente que el equipo esté convenientemente refrigerado, ya que debe de ofrecer el servicio web las 24 horas del día los 7 días de la semana (también conocido como 24/7).

#### 4.2.2.2 *Requisitos de la interfaz*

En este apartado pasaremos a explicar cómo debe de ser la interfaz de usuario de nuestro sistema. Pero antes pasemos a describirla:

*“La interfaz de usuario es el medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo.”*

La interfaz de usuario, aunque pueda parecer algo superfluo, puede, en muchos casos, determinar el éxito o fracaso de un proyecto. Una interfaz debe de ser:

- **Fácil de aprender:** es decir, que el tiempo necesario desde la primera toma de contacto con la aplicación hasta su uso con productividad sea mínimo. Para conseguir esto se debe cumplir que sea:
  - **Sintetizable:** El usuario debe ser consciente de los cambios surgidos en la interfaz a consecuencia de una acción tomada.
  - **Familiar:** Debe existir una relación entre los conocimientos que posee el usuario y los que son necesarios para la interacción con dicha aplicación.

- **Flexible:** Deben existir múltiples formas o caminos en los que usuario y el sistema intercambien información. Para poder medir la flexibilidad existe una serie de parámetros:
  - **Control de usuario:** Determina la cantidad de acciones que puede realizar el usuario en el sistema.
  - **Migración de tareas:** Determina la posibilidad de transferir el control de las tareas entre el usuario y el sistema.
  - **Capacidad de sustitución:** Determina la capacidad que tiene el sistema de permitir que los valores que existen previamente puedan ser sustituidos por otros.
- **Adaptable:** El sistema se debe poder adaptar al uso que el usuario hace del mismo y a los cambios que el usuario puede realizar en el dispositivo, como girar la pantalla.
- **Consistente:** Todos los mecanismos, interfaces, diseños, etc., deben de seguir un patrón común, para que al usuario le resulte fácil de usar.
- **Robusta:** El sistema debe ser capaz de soportar situaciones no esperadas de manera eficiente sin alterar la experiencia del usuario.
- **Recuperable:** El sistema debe dar opción al usuario de poder deshacer una acción realizada erróneamente.
- **Con tiempo de respuesta mínimo:** El tiempo de espera debe de ser el mínimo posible para que el usuario no se altere pensando que la aplicación no funciona o el dispositivo va demasiado lento.
- **Adecuada en las tareas:** El sistema debe permitir realizar al usuario las tareas previstas de la manera que desea.
- **Con poca carga cognitiva:** El usuario debe sentirse cómodo cada vez que use la aplicación, es decir, ha de ser intuitiva.

## 4.3 ANÁLISIS DEL SISTEMA

---

Tras haber recopilado los principales requisitos del sistema, es necesario realizar un análisis del sistema. Es aquí donde se especifican y estructuran y plasman todas las condiciones, necesidades y restricciones recogidas en los requisitos del sistema.

De esta forma, se puede enfocar el desarrollo de una manera u otra. De esta etapa depende directamente el diseño del sistema, ya que un mal análisis del sistema dará como resultado un mal diseño del mismo.

Este análisis del sistema debería seguir los siguientes pasos:

- Identificar las fuentes de información (usuarios) relevantes para el proyecto y planificar actividades de investigación.
- Realizar preguntas correctas para entender lo que nos pide el usuario.
- Analizar la información para reconocer aquellos aspectos que no queden del todo claros.
- Establecer escenarios y flujos de casos de usos
- Verificar los casos de uso con los usuarios interesados

#### 4.3.1 Modelos de caso de uso

Un caso de uso puede ser definido como la representación de una situación o proceso de interacción del usuario con la aplicación o sistema. Sirven para mostrar de manera simple, los requisitos del sistema y comprobar la reacción ante eventos que surgen en el propio sistema o en su ámbito. [21]

Visualmente están compuestos por dos nodos:

- **Actor:** Representa cualquier agente que intercambia información con el sistema, por lo que no pertenece a este. Puede ser tanto una persona, como una máquina.
- **Caso de uso:** Secuencia de transacciones que se encuentran relacionadas por su comportamiento.

Un caso de uso está compuesto de la siguiente información:

- Nombre del caso de uso
- Sistema al que pertenece
- Actores que participan
- Requisitos o condiciones previas
- Operaciones que se realizan
- Alternativas que surgen

En nuestro caso, el único actor será el usuario que use la aplicación.

Para empezar, hay que definir el diagrama frontera, que será el que muestre de forma genérica cómo funciona el sistema:

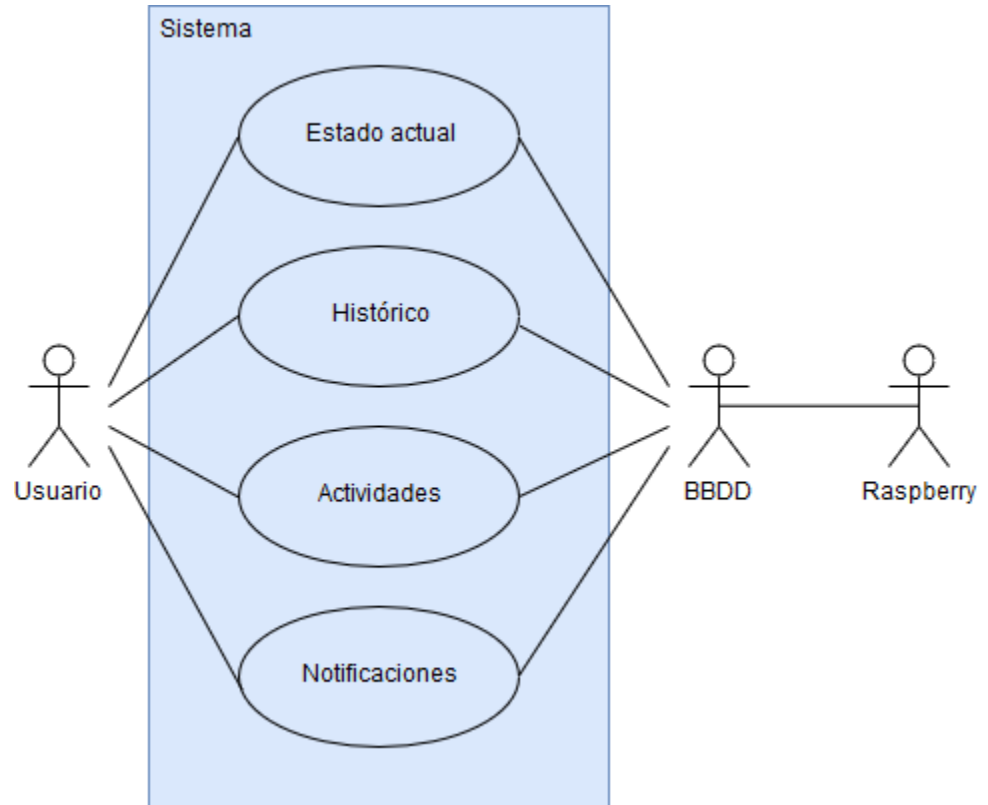


Ilustración 16 - Diagrama frontera

Una vez tenemos definido este diagrama frontera voy a definir las funcionalidades que posee el sistema mediante distintos casos de uso:

**Caso de uso 1:** Comprobar el estado actual del ambiente inteligente

*Actores participantes:* Usuario, BBDD.

*Sistema:* Aplicación móvil.

*Condiciones previas:* Ninguna.

*Flujo de eventos:*

1. Entra en la aplicación.

2. Espera a que carguen los eventos producidos por los sensores (A-1), (A-2).

*Condiciones de salida:* Listado de último estado de los sensores del ambiente inteligente.

*Alternativas:*

1. Si no hay eventos que mostrar, aparece un mensaje informándolo.
2. En caso de no haber conexión a internet se muestra un mensaje informando al usuario.

### **Caso de uso 2:** Comprobar el histórico del ambiente inteligente

*Actores participantes:* Usuario, BBDD.

*Sistema:* Aplicación móvil.

*Condiciones previas:* Ninguna.

*Flujo de eventos:*

1. Entra en la aplicación.
2. Abrir el menú lateral izquierdo
3. Pulsar la sección "Histórico"
4. Espera a que cargue el histórico de eventos producidos por los sensores (A-1), (A-2).

*Condiciones de salida:* Listado de histórico de los sensores del ambiente inteligente.

*Alternativas:*

1. Si no hay eventos que mostrar, aparece un mensaje informándolo.
2. En caso de no haber conexión a internet se muestra un mensaje informando al usuario.

### **Caso de uso 3:** Ordenado del histórico

*Actores participantes:* Usuario, BBDD.

*Sistema:* Aplicación móvil.

*Condiciones previas:* Estar en la vista "Histórico".

*Flujo de eventos:*

1. El usuario hace clic en el primer botón superior derecho
2. Selecciona el ordenado que desea realizar
3. La aplicación actualiza el contenido del histórico mostrando los eventos en el orden establecido.

*Condiciones de salida:* Listado del histórico ordenado

*Alternativas:* Ninguna

#### **Caso de uso 4:** Filtrado del histórico

*Actores participantes:* Usuario, BBDD.

*Sistema:* Aplicación móvil.

*Condiciones previas:* Estar en la vista “Histórico”.

*Flujo de eventos:*

1. El usuario hace clic en el segundo botón superior derecho
2. Selecciona el filtrado que desea realizar
3. La aplicación actualiza el contenido del histórico mostrando únicamente los eventos que coinciden con ese filtrado (A-1)

*Condiciones de salida:* Listado del histórico filtrado

*Alternativas:*

1. Si no hay eventos que mostrar, aparece un mensaje informándolo.

#### **Caso de uso 5:** Comprobar las actividades reconocidas

*Actores participantes:* Usuario, BBDD.

*Sistema:* Aplicación móvil.



*Condiciones previas:* Ninguna.

*Flujo de eventos:*

1. Entra en la aplicación.
2. Abrir el menú lateral izquierdo
3. Pulsar la sección “Actividades”
4. Espera a que cargue las actividades reconocidas (A-1), (A-2).

*Condiciones de salida:* Listado de actividades reconocidas del ambiente inteligente.

*Alternativas:*

1. Si no hay actividades que mostrar, aparece un mensaje informándolo.
2. En caso de no haber conexión a internet se muestra un mensaje informando al usuario.

#### **Caso de uso 6:** Seleccionar/Deseleccionar sensores de los que recibir notificaciones

*Actores participantes:* Usuario, BBDD.

*Sistema:* Aplicación móvil.

*Condiciones previas:* Ninguna.

*Flujo de eventos:*

1. Entra en la aplicación.
2. Abrir el menú lateral izquierdo
3. Pulsar la sección “Notificaciones”
4. Espera a que cargue los sensores disponibles (A-1), (A-2).
5. Pulsar el botón de la derecha de cada sensor para activar/desactivar sus notificaciones.

*Condiciones de salida:* Listado de sensores disponibles en el ambiente inteligente.

*Alternativas:*

1. Si no hay actividades que mostrar, aparece un mensaje informándolo.
2. En caso de no haber conexión a internet se muestra un mensaje informando al usuario.

**Caso de uso 7:** Visualizar los créditos de la aplicación

*Actores participantes:* Usuario.

*Sistema:* Aplicación móvil.

*Condiciones previas:* Ninguna.

*Flujo de eventos:*

1. Entra en la aplicación.
2. Abrir el menú lateral izquierdo
3. Pulsar la sección “Acerca de”
4. Se muestra la información relativa al creador de la aplicación

*Condiciones de salida:* Visualización de la información del creador

*Alternativas:* Ninguna

**Caso de uso 8:** Entrar a la aplicación por medio de una notificación

*Actores participantes:* Usuario, BBDD.

*Sistema:* Aplicación móvil.

*Condiciones previas:* Ninguna.

*Flujo de eventos:*

1. Aparece una notificación en el dispositivo.
2. El usuario hace clic en ella.
3. La aplicación se abre y aparece el estado actual de los sensores. (A-1) (A-2)

*Condiciones de salida:* Apertura de la aplicación.

*Alternativas:*

1. Si no hay actividades que mostrar, aparece un mensaje informándolo.

2. En caso de no haber conexión a internet se muestra un mensaje informando al usuario.

**Caso de uso 9:** Actualizar contenido de la vista “Estado actual” manualmente

*Actores participantes:* Usuario, BBDD.

*Sistema:* Aplicación móvil.

*Condiciones previas:* Estar en la vista “Estado Actual”.

*Flujo de eventos:*

1. El usuario realiza un gesto hacia abajo en la pantalla del dispositivo
2. La aplicación actualiza el contenido del estado actual de los sensores. (A-1) (A-2)

*Condiciones de salida:* Listado de último estado de los sensores del ambiente inteligente

*Alternativas:*

1. Si no hay actividades que mostrar, aparece un mensaje informándolo.
2. En caso de no haber conexión a internet se muestra un mensaje informando al usuario.

**Caso de uso 10:** Actualizar contenido de la vista “Estado actual” automáticamente

*Actores participantes:* Usuario, BBDD.

*Sistema:* Aplicación móvil.

*Condiciones previas:* Estar en la vista “Estado Actual”.

*Flujo de eventos:*

1. Se produce un nuevo evento en el ambiente inteligente.
2. La aplicación actualiza el contenido del estado actual de los sensores. (A-2)

*Condiciones de salida:* Listado de último estado de los sensores del ambiente inteligente

*Alternativas:* En caso de no haber conexión a internet se muestra un mensaje informando al usuario.

### 4.3.2 Escenarios

Un escenario es una descripción detallada y ficticia del comportamiento de un sistema en una determinada situación. A pesar de estar acotados a un determinado comportamiento, describen todo el contexto que involucra a esa actividad: recursos del sistema, objetivos de los usuarios, contexto social en que se desarrolla, entidades involucradas. Proveen un “retrato” de como esa actividad se lleva a cabo. Los escenarios describen situaciones teniendo en cuenta aspectos de uso, permitiendo: conocer el problema, unificar criterios, ganar compromiso con clientes / usuarios, organizar los detalles involucrados y entrenar a nuevos participantes.

Gracias a ellos facilita a los diseñadores a tener en cuenta los usuarios a los que va enfocado el sistema y qué actividades van a desarrollarse en él.

Un escenario está formado por los siguientes elementos:

- Nombre único
- Descripción
- Actores que participan en él
- Flujo de los eventos que tienen lugar en el escenario

Para caso de uso es posible la existencia de múltiples escenarios. En este caso se van a definir unos cuantos escenarios para los casos de uso más importantes de nuestro sistema.

*Escenario 1*

*Nombre:* Comprobación del estado actual

*Descripción:* Un usuario quiere comprobar el estado actual en el que se encuentran los sensores del ambiente inteligente.

*Actores participantes:* Usuario y BBDD.

*Flujo de eventos:*

1. El usuario abre la aplicación.
2. La aplicación muestra la ventana principal, que coincide con el estado actual que el usuario quiere comprobar.
3. En esta ventana aparece un mensaje: "Obteniendo datos...".
4. El sistema envía una petición al servidor solicitándole el estado de los sensores.
5. El servidor web devuelve la respuesta con el estado actual de los sensores.
6. La aplicación procesa la información recibida y la muestra al usuario.

*Escenario 2*

*Nombre:* Comprobación del histórico

*Descripción:* Un usuario quiere comprobar el histórico de los sensores del ambiente inteligente.

*Actores participantes:* Usuario y BBDD.

*Flujo de eventos:*

1. El usuario abre la aplicación.
2. La aplicación muestra la ventana principal.
3. EL usuario selecciona el botón de Menú y se despliega una lista de opciones.
4. El usuario hace clic sobre "Histórico".
5. En esta ventana aparece un mensaje: "Obteniendo datos...".
6. El sistema envía una petición al servidor solicitándole el histórico de los sensores.
7. El servidor web devuelve la respuesta con el histórico de los sensores.
8. La aplicación procesa la información recibida y la muestra al usuario.

*Escenario 3*

*Nombre:* Activación de notificaciones

*Descripción:* Un usuario quiere activar las notificaciones para uno o más sensores del ambiente inteligente.

*Actores participantes:* Usuario y BBDD.

*Flujo de eventos:*

1. El usuario abre la aplicación.
2. La aplicación muestra la ventana principal.
3. EL usuario selecciona el botón de Menú y se despliega una lista de opciones.
4. El usuario hace clic sobre “Notificaciones”.
5. En esta ventana aparece un mensaje: “Obteniendo datos...”.
6. El sistema envía una petición al servidor solicitándole los sensores disponibles.
7. El servidor web devuelve la respuesta con los sensores disponibles.
8. La aplicación procesa la información recibida y la muestra al usuario.
9. El usuario hace clic en el botón que se encuentra al lado de cada uno de los sensores y activa los que considera oportunos.
10. El sistema almacena esta información y actualiza la información en el servidor.

*Escenario 4*

*Nombre:* Visualizar el histórico de un único sensor

*Descripción:* Un usuario quiere conocer el estado por el que ha pasado uno de los sensores disponibles

*Actores participantes:* Usuario y BBDD.

*Flujo de eventos:*

1. El usuario abre la aplicación.
2. La aplicación muestra la ventana principal.
3. En esta ventana aparece un mensaje: "Obteniendo datos...".
4. El sistema envía una petición al servidor solicitándole el estado actual de los sensores.
5. El servidor web devuelve la respuesta con el estado actual de los sensores.
6. La aplicación procesa la información recibida y la muestra al usuario.
7. El usuario hace clic en el sensor del que desea conocer información.
8. El sistema envía una petición al servidor solicitándole el histórico del sensor del que sea obtener información.
9. El servidor web devuelve la respuesta con el histórico del sensor.
10. La aplicación procesa la información recibida y la muestra al usuario.



*Escenario 5*

*Nombre:* Visualización de actividades

*Descripción:* Un usuario quiere conocer las actividades que han sido reconocidas por el sistema

*Actores participantes:* Usuario y BBDD.

*Flujo de eventos:*

1. El usuario abre la aplicación.
2. La aplicación muestra la ventana principal. EL usuario selecciona el botón de Menú y se despliega una lista de opciones.
3. El usuario hace clic sobre “Actividades”.
4. El sistema envía una petición al servidor solicitándole el listado de actividades.
5. El servidor web devuelve la respuesta con las actividades reconocidas.
6. La aplicación procesa la información recibida y la muestra al usuario.

## 4.4 DISEÑO DEL SISTEMA

---

El diseño es una de las fases más importante del proceso de creación de un software, ya que, es donde se aplican distintas técnicas y principios que permiten la posterior implementación del software. Es una tarea ardua y delicada a la vez, puesto que, de no realizarse correctamente, a la hora de realizar la implementación, habrá aspectos que resultarán más complicados o imposibles de resolver, y por tanto será necesario revisar de nuevo esta fase. Además, las estrategias a seguir para traducir el modelo a código son bastante complejas y muy variadas, lo que puede complicar bastante la labor de realizar la implementación.

Se puede decir, pues, que el diseño del sistema es la actividad de la Ingeniería del Software en la que se identifican los objetivos finales del sistema y se plantean las diversas estrategias para alcanzarlos en la actividad de implementación [22].

En esta etapa se especifican el diseño y estructura de los datos y el diseño de la interfaz. Estas van realizándose paralelamente ya que en parte una depende de la otra y de esta forma ambas están listas a la par.

### 4.4.1 Diseño de los datos

En este apartado se especifica de manera exacta la estructura que van a tener los datos que serán almacenados en el sistema. Esta estructura será la definitiva a la hora de poner en marcha el sistema.

Para ello, se hace uso de la herramienta modelo entidad-relación, que permite representar las entidades relevantes de nuestro sistema, junto con las asociaciones que tienen entre ellas y sus atributos.

Este modelo se basa en una percepción del mundo real en la que los objetos que lo conforman son denominados entidades y estas entidades tienen una serie de propiedades denominadas atributos. Estas entidades se relacionan entre ellas. Por tanto:

- **Entidades:** cosas u objetos del mundo real, que son distintas del resto de elementos del mundo real. Estas entidades pueden ser tan variadas como, por ejemplo: coche, persona, animal,

habitación, pensamiento, etc. Su representación en el modelo entidad relación viene definido por un rectángulo como el siguiente:



- **Atributos:** Son las propiedades o características que definen a una entidad. Pueden existir multitud de atributos para una entidad, y cada uno de ellos pueden ser de distintos tipos como, por ejemplo: int, double, date, boolean, etc. Gracias a estos atributos, es posible diferenciar distintas instancias de una entidad. Por ejemplo, puede existir varios tipos de coche dentro de la entidad “Coche”. Su representación en el modelo E/R es la siguiente:



- Relaciones: Las relaciones son las que permiten la asociación entre dos o más entidades. Estas entidades pueden ser de distintos tipos:
  - Relación 1-1: Es aquella relación en la que una instancia de una entidad le corresponde otra instancia de la otra entidad.
  - Relación 1-N: Es aquella relación en la que una instancia de una entidad le corresponden muchas instancias de la otra entidad.
  - Relación N-M: Es aquella relación en la que muchas instancias de una entidad corresponden con muchas instancias de la otra entidad. Esta relación da lugar a una nueva tabla, que será la unión de las dos entidades, más atributos propios (si los tuviera).

Las relaciones están determinadas por el símbolo:



Además de los elementos descritos, existen otro tipo de elementos que completan a los anteriores como puede ser:

- **Claves:** Son un subconjunto de los atributos de una entidad que identifican de forma inequívoco cada uno de los elementos que conforman la entidad. A su vez se pueden diferenciar en distintos tipos de claves:
  - Superclave: Es un subconjunto de atributos que diferencian las instancias de las entidades entre sí.
  - Clave candidata: Es un atributo que será superclave en el caso de que la superclave de esa entidad desaparezca.
  - Clave primaria: Es la clave candidata para identificar las instancias dentro de las entidades.
- **Entidades:** Podemos encontrar dos tipos de entidades:
  - Fuertes: Es aquella que puede identificada como única.
  - Débiles: Son aquellas que no pueden existir sin que exista una relación con otra entidad superior, es decir, no puede ser identificada como única.
- **Herencia:** Es un tipo especial de relación en el que las entidades “hijas” de una entidad padre heredan los atributos del padre y además añaden los suyos propios.
- **Agregación:** Se trata de una abstracción en el que las relaciones son tratadas como entidades superiores. Expresan una relación entre relaciones o entre entidades y relaciones. Se representan con un triángulo englobando la relación a abstraer.

Una vez conocidos los elementos que forman parte de un diagrama entidad-relación podemos empezar a desarrollar el modelo entidad-relación. Los pasos a seguir son los siguientes:

1. Convertir los elementos necesarios del sistema en un Esquema Conceptual.
2. Convertir este Esquema Conceptual en un Esquema Conceptual Modificado en el que se realizan algunos cambios, como eliminación de relaciones de N-M.
3. Obtener las tablas que formarán parte de la base de datos a partir del Esquema Conceptual Modificado.

#### 4.4.1.1 *Esquema conceptual*

En nuestro caso, se parte de un proyecto ya existente, por lo que el esquema conceptual ya está previamente creado y, por tanto, no se realizarán modificaciones.

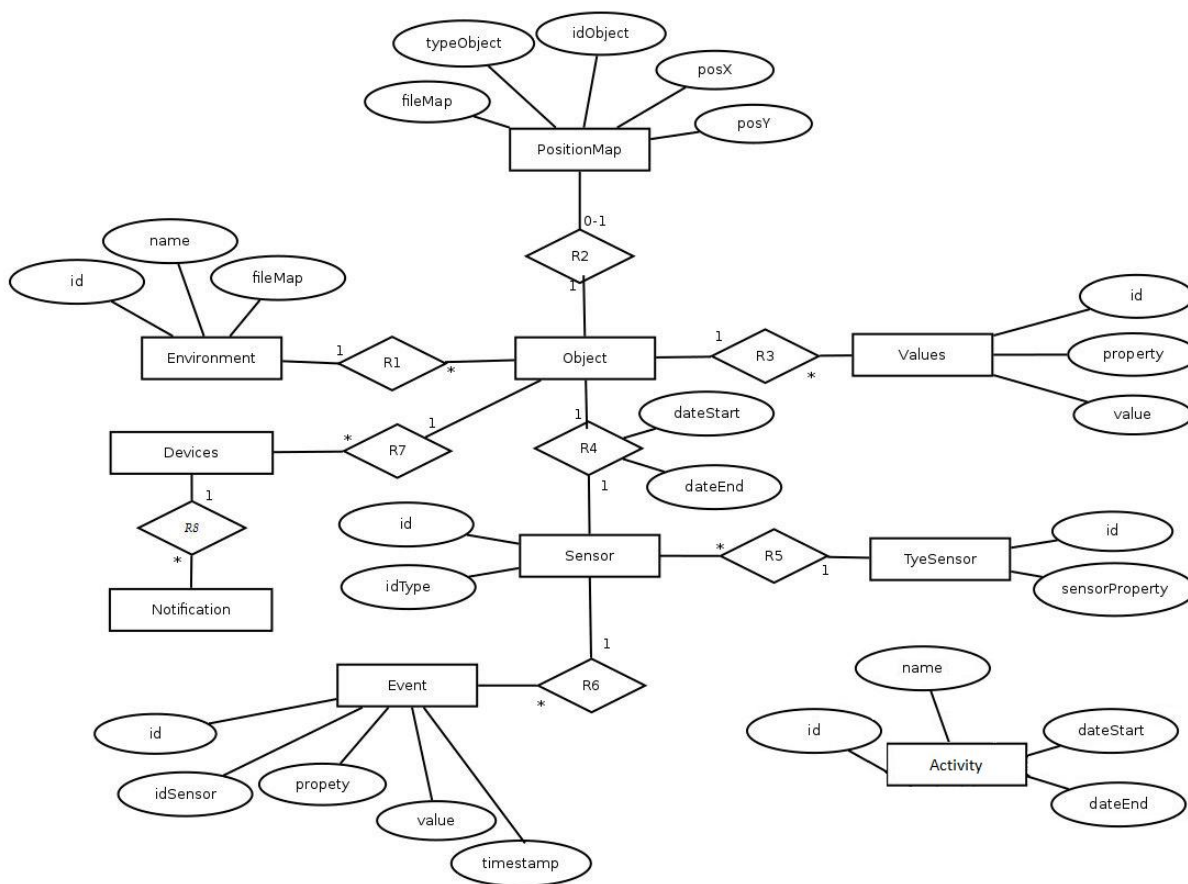


Ilustración 17 - Esquema conceptual

Donde las relaciones son las siguientes:

- R1: A un ambiente le pertenecen muchos objetos.
- R2: Un objeto puede tener una o ninguna sola posición en el mapa.
- R3: En un objeto se definen uno o varios valores.
- R4: Un objeto pertenece a un único sensor.
- R5: Un tipo de sensor puede tener muchos sensores.
- R6: Un sensor envía múltiples eventos.
- R7: Un objeto puede estar en muchos dispositivos móviles
- R8: Un dispositivo móvil puede tener muchas notificaciones y/o sensores.

#### 4.4.1.2 Esquema conceptual modificado

Para la obtención del Esquema Conceptual Modificado a partir del Esquema Conceptual se deben hacer las siguientes modificaciones:

- Eliminar todas las entidades débiles.
- Eliminar las relaciones de muchos a muchos.
- Eliminar las relaciones con atributos existentes en el Esquema Conceptual.

En nuestro caso, no es necesaria la eliminación de entidades débiles, ni eliminar relaciones muchos a muchos. Sin embargo, si es necesario eliminar las relaciones que contienen atributos. Por lo que el

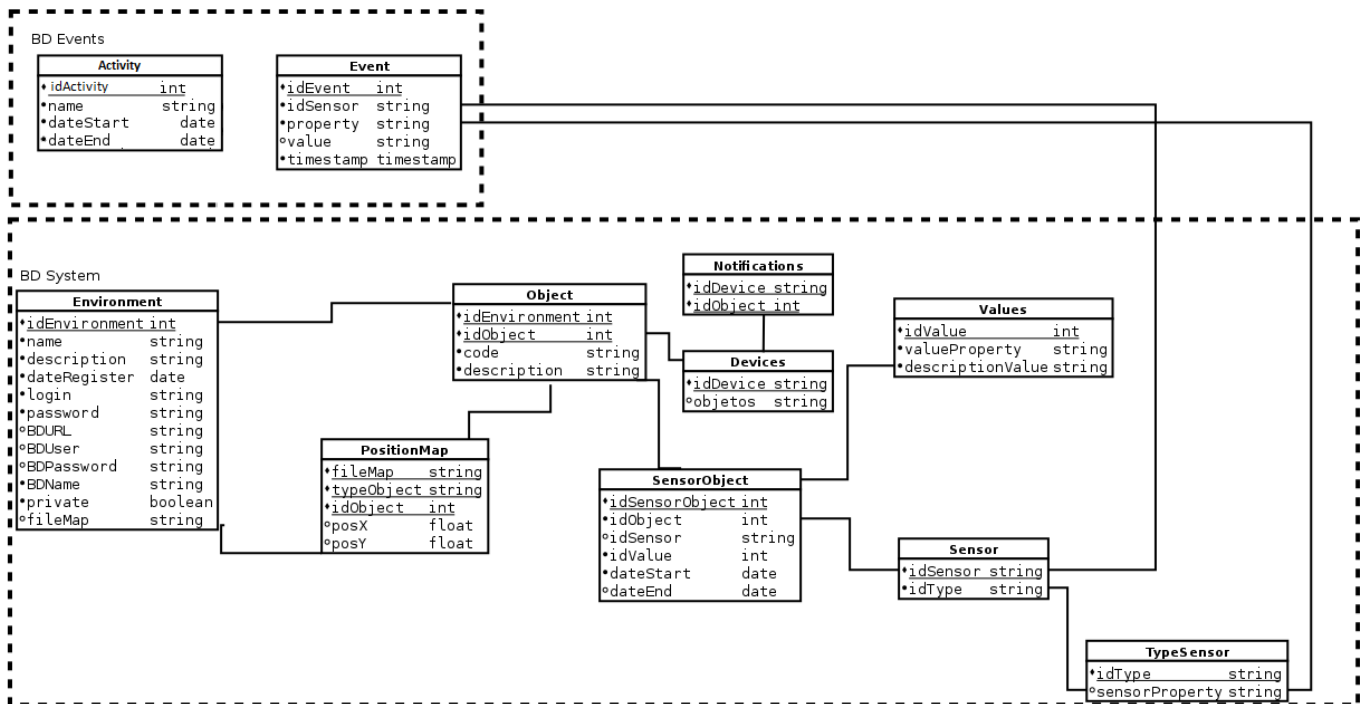


Ilustración 18 - Esquema resultante

esquema, finalmente quedaría de la siguiente forma:

#### 4.4.1.3 Tablas resultantes

Una vez obtenido el esquema conceptual, es necesario desglosar las tablas que compondrán la base de datos del sistema:

- **Event:** En esta tabla se almacenará toda la información que los sensores recogen continuamente del ambiente. Al ser una tabla con un volumen de datos muy alto, esta tabla ha de estar optimizada para que la carga de los datos sea rápida. Sus atributos son los siguientes:
  - *idEvent*: Identificador numérico auto-incremental que identifica cada evento.
  - *idSensor*: Identificador único de cada sensor.
  - *property*: Propiedad del valor que emite el sensor. (Luminosidad, Temperatura, etc.)
  - *value*: Valor de la propiedad.
  - *timestamp*: Marca temporal.
  
- **Activity:** Tabla donde se almacenará todas las actividades que vayan siendo reconocidas por el sistema. Sus atributos son:
  - *idActivity*: Identificador número auto-incremental que reconoce cada actividad.
  - *name*: Nombre de la actividad reconocida.
  - *dateStart*: Fecha de inicio de la actividad
  - *dateEnd*: Fecha fin de la actividad
  
- **Environment:** Esta tabla almacena toda la información relevante al ambiente inteligente que se va a monitorizar, así como el acceso a la base de datos donde se almacenan los eventos generados por los sensores. Sus atributos son:
  - *idEnvironment*: Identificador numérico auto-incremental que identifica un ambiente de forma inequívoca.
  - *name*: Nombre del ambiente inteligente.
  - *description*: Descripción breve del ambiente inteligente.
  - *dateRegister*: Fecha de registro del ambiente inteligente.
  - *login*: Usuario principal del ambiente inteligente.
  - *password*: Contraseña del usuario.
  - *BDURL*: Dirección de la base de datos donde se almacenarán los eventos.
  - *BDUser*: Usuario de la base de datos de eventos.
  - *BDPassword*: Contraseña del usuario de la base de datos de eventos.
  - *BDName*: Nombre de la base de datos donde se almacenan los eventos.
  - *private*: Valor que indica si el ambiente es público o privado.

- *fileMap*: URL que indica la ubicación del plano del ambiente inteligente
- **Sensors**: Tabla que almacena los sensores dados de alta en el ambiente inteligente.  
Posee los siguientes atributos:
  - *idSensor*: Identificador único del sensor
  - *idType*: Identificador correspondiente al tipo de sensor.
- **TypeSensor**: Tabla que junto con la anterior permite que un sensor pueda recoger varios tipos de eventos (temperatura, luminosidad, etc.) Sus atributos son los siguientes:
  - *idType*: Identificador correspondiente al tipo de sensor.
  - *sensorProperty*: Propiedad que mide el sensor (temperatura, luminosidad, etc.).
- **PositionMap**: Tabla que se usa para aspectos no relacionados con este proyecto.
- **Object**: Tabla que almacena los objetos existentes en el ambiente. Además, esta tabla es la encargada de asociar los objetos con el ambiente correspondiente. Sus atributos son los siguientes:
  - *idEnvironment*: Identificador del ambiente inteligente donde se encuentra el objeto.
  - *idObject*: Identificador numérico auto-incremental que identifica un objeto.
  - *code*: Identificador corto que sirve para nombrar un objeto.
  - *description*: Descripción del objeto.
- **SensorObject**: Tabla que se encarga de enlazar un sensor y el objeto que representa en un ambiente. Consta de los siguientes atributos:
  - *idEnvironment*: Identificador del ambiente inteligente donde se encuentra el objeto.
  - *idObject*: Identificador numérico auto-incremental que identifica un objeto.
  - *idSensor*: Identificador del sensor
  - *idValue*: Identificador del valor.
  - *dateStart*: Fecha de registro del objeto
  - *dateEnd*: Fecha de baja del objeto.



- **Values:** Tabla que se encarga de traducir los valores enviados por los sensores a lenguaje natural. Consta de los siguientes atributos:
  - *idValue*: Identificador numérico auto-incremental que sirve para identificar un valor.
  - *valueProperty*: Propiedad a traducir al lenguaje natural
  - *descriptionValue*: Descripción del valor traducido.
- **Device y Notifications:** Tablas que se emplean para usos no relacionados con este proyecto.

#### 4.4.2 Diseño de la interfaz

En este apartado se define como va a ser la apariencia visual de la aplicación, es decir, la interfaz entre el usuario y la aplicación. Esta es una de las tareas más influyentes de cara a la elección una aplicación u otra por parte del usuario, puesto que, una aplicación que no sea atractiva visualmente o fácilmente entendible, por muy efectiva y potente que sea, poco éxito tendrá.

Normalmente es una actividad multidisciplinar que involucra a varias ramas es decir al diseño y el conocimiento como el diseño gráfico, industrial, web, de software y la ergonomía; y está implicado en un amplio rango de proyectos, desde sistemas para computadoras, vehículos hasta aviones comerciales.

Su objetivo es que las aplicaciones o los objetos sean más atractivos y, además, hacer que la interacción con el usuario sea lo más intuitiva posible, conocido como el diseño centrado en el usuario. En este sentido las disciplinas del diseño industrial y gráfico se encargan de que la actividad a desarrollar se comunique y aprenda lo más rápidamente, a través de recursos como la gráfica, los pictogramas, los estereotipos y la simbología, todo sin afectar el funcionamiento técnico eficiente. [23]

Por tanto, podemos decir que se centra en hacer la experiencia del usuario con respecto a la aplicación, servicio, etc., lo más fácil e intuitiva posible. Por tanto, con el diseño de la interfaz se pretende que las aplicaciones sean más atractivas de cara al usuario.

##### 4.4.2.1 Guía de estilo

Por tanto, si se pretende que nuestra aplicación sea lo más intuitiva posible y de rápido aprendizaje, tenemos que seguir unas pautas que nos guíen de alguna forma, a la hora de diseñar la aplicación web. Estas pautas aseguran que el diseño será uniforme a lo largo de la aplicación y por tanto el usuario se hará mucho más rápido con el diseño y con su uso. Y estas pautas vienen a ser llamadas guías de estilo. Por tanto, definamos que es una guía de estilo:

*“Es un documento que recoge normativas y patrones básicos relacionados con el "aspecto" de un interfaz para su aplicación en el desarrollo de nuevas "pantallas" dentro de un entorno concreto. (sitio web de contenidos, aplicaciones móviles, etc).” [24]*

Esto facilita la labor de los diseñadores, ya que no tienen que preocuparse demasiado sobre que diseño emplear, ya que está definido y por tanto solo han de seguir dichas pautas. Y por parte de los programadores, les resultará más fácil saber cuáles son los elementos que han de programar.

En este caso, al estar desarrollando una aplicación Android, Google, su creador, tiene una guía de estilos para sus aplicaciones móviles llamada “Material Design”. Fue anunciado en la conferencia Google I/O celebrada el 25 de junio de 2014. Material se integró en Android Lollipop como reemplazo de Holo, anteriormente utilizado desde Android 4 y sucesores. [25]

Esta guía de estilos define completamente el estilo que las aplicaciones han de seguir, por ejemplo, colores, formas, posiciones de elementos, etc.

A continuación, se muestran algunos de los aspectos que se indica en la guía de estilos y que se han tenido en cuenta a la hora de plantear el diseño de la interfaz de la aplicación:

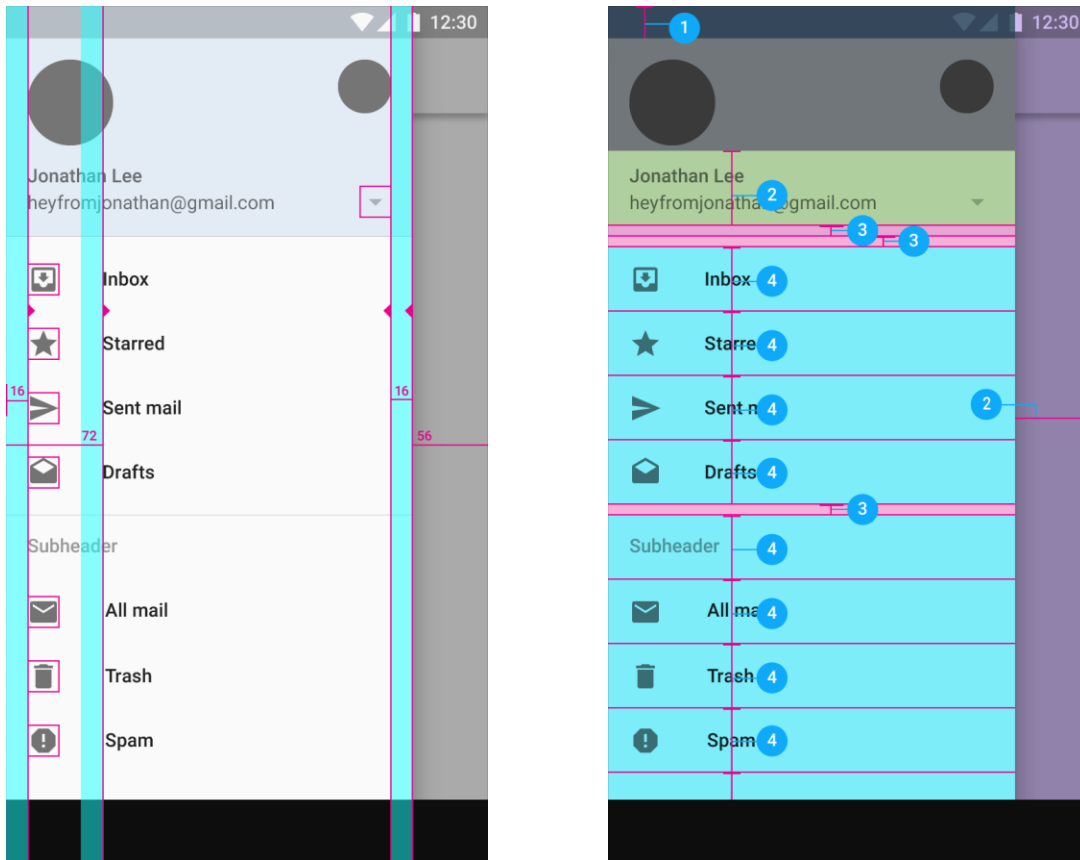


Ilustración 19 - Guía de estilos del menú lateral

## Menú lateral

Google recomienda que se dejen una serie de márgenes a la hora de definir cada una de las secciones del menú. También especifica el tamaño que han de tener los botones que componen el menú.

## Listas

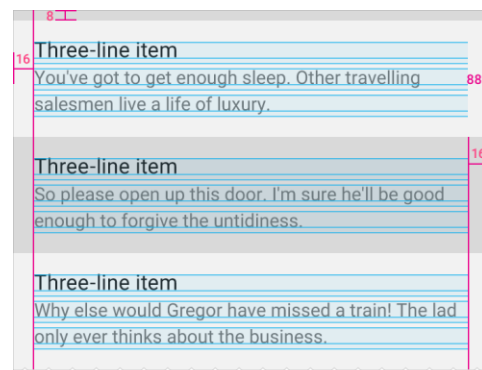
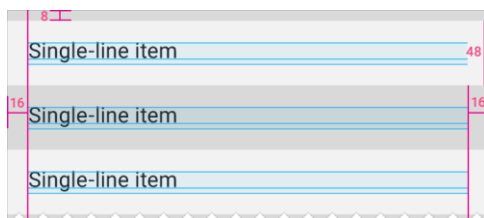


Ilustración 20 - Guía de estilos de las listas

En este caso, es específica como de grandes han de ser cada una de las filas que componen una lista, y qué márgenes deben de respetar. Se muestran tanto para filas de 2 como de 3 renglones.

## Botones

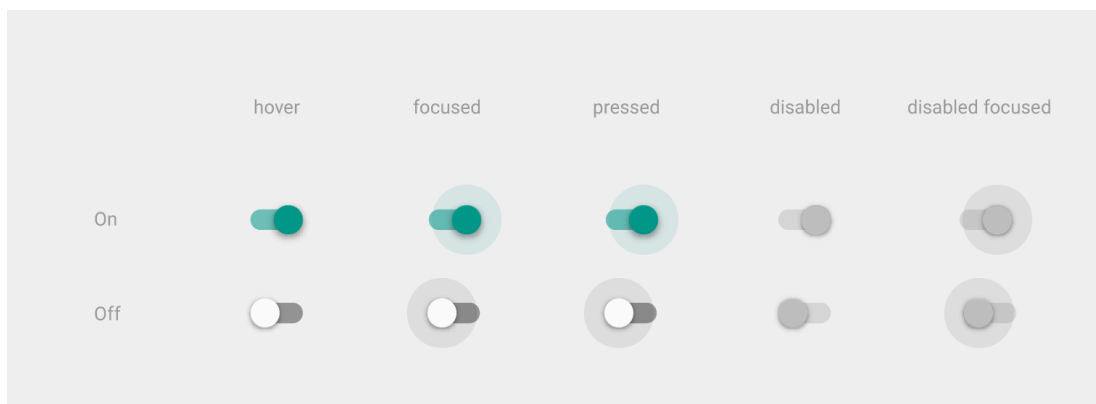


Ilustración 21 - Guía de estilo de los botones

Se indica los distintos estados de los botones: normal, con foco, presionado, desactivado y desactivado con foco.

## Barra de herramientas

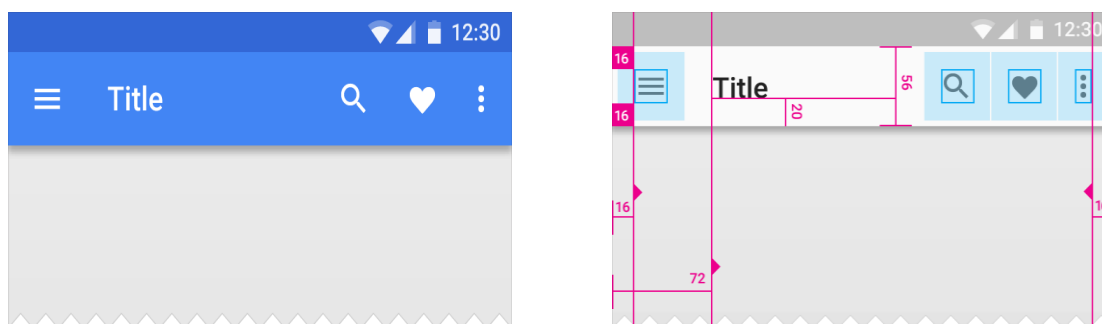


Ilustración 22 - Guía de estilos de la barra de herramientas

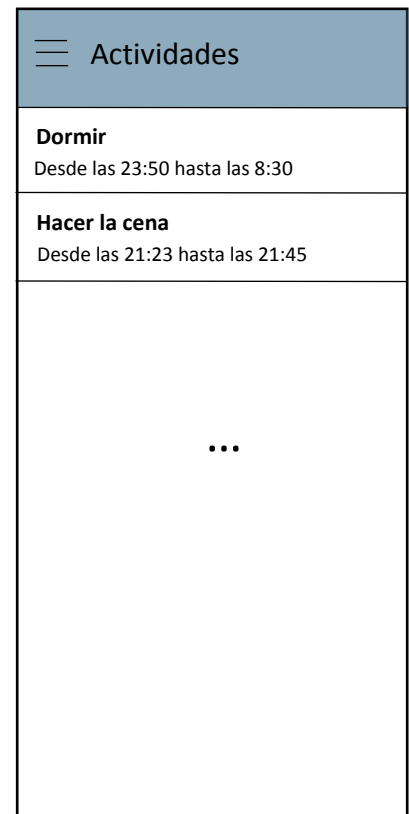
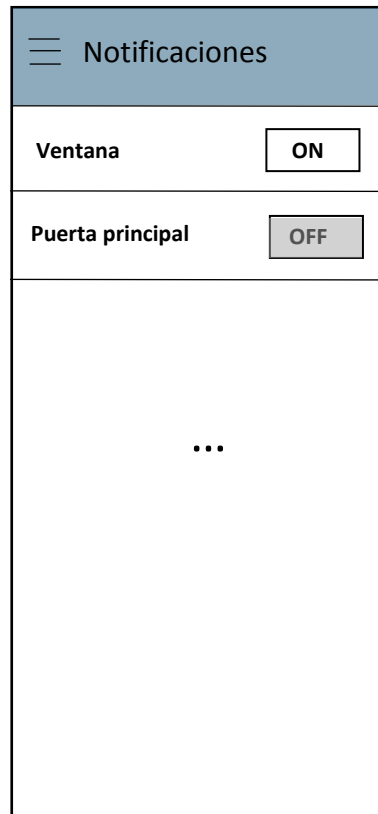
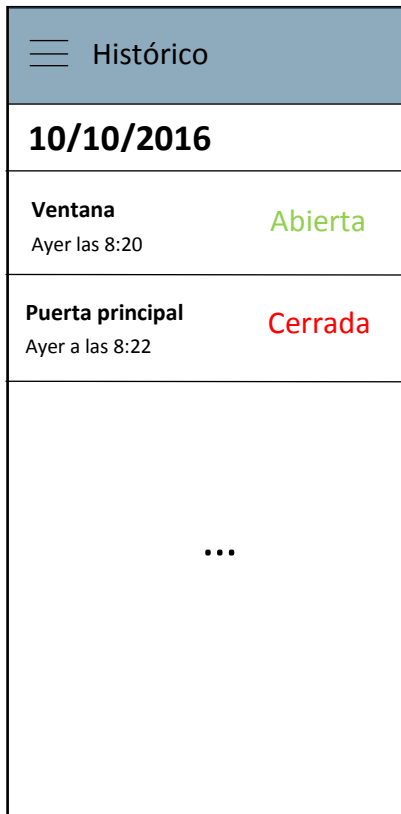
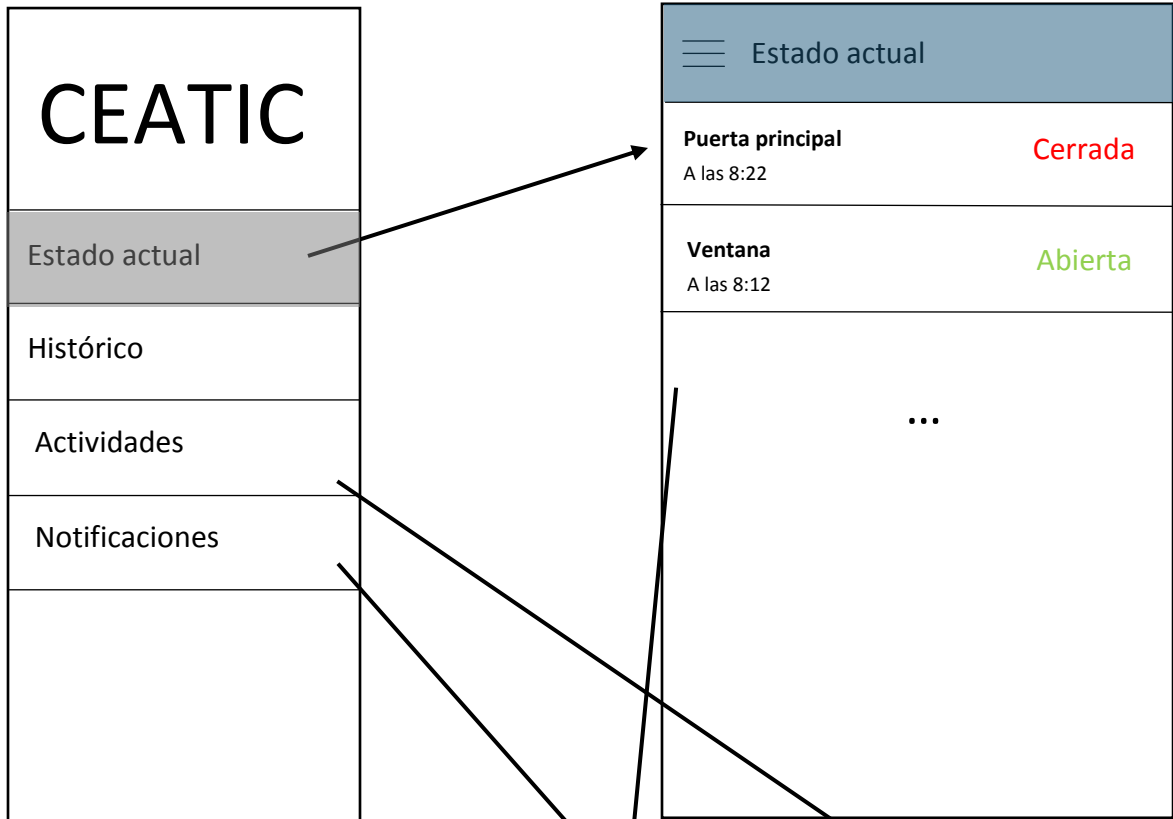
En este caso se especifican como han de ser los botones de la barra de herramientas, el tamaño que han de tener y márgenes que se han de respetar.

#### 4.4.2.2 *Storyboard*

Un storyboard es una técnica en la que se presenta una sucesión de bocetos del diseño que tendrá la interfaz. El propósito es servir como guía para poder seguir una navegación a la hora de desarrollar una aplicación.

Se usó por primera vez en los estudios de Disney en la década de 1930 y su objetivo fue intentar mostrar un conjunto de viñetas de forma que la secuenciación de todas ellas formase una animación. Hoy en día se usa en múltiples campos para esbozar de manera preliminar el resultado final del trabajo a desarrollar.

En nuestro caso, como he mencionado antes, permite tener una idea previa de lo que se quiere diseñar y, sobre todo, como se quiere que sea el resultado de manera aproximada. De esta forma se podrá enseñar al cliente que esté interesado en la aplicación y podrá dar su visto bueno, o corregir aquellos aspectos en los que no esté de acuerdo. Para su desarrollo, veremos el paso de una pantalla a otra mediante flechas, dependiendo del elemento sobre el que interactúe el usuario. Se mostrarán varios casos mediante storyboards.



## 4.5 IMPLEMENTACIÓN DEL SISTEMA

---

En la implementación del sistema, las especificaciones de diseño, anteriormente vistas, sirven como base para la construcción del sistema. Es donde el análisis y el diseño anteriormente descritos, toman partida y a partir de ellos comienza a desarrollarse la aplicación, asegurándose de que se satisfacen los requisitos que fueron especificados. Esta es la parte más costosa de todas, ya que es donde se involucra a más personal y donde se consume más tiempo de realización.

Esto implica una serie de pasos o fases, que se han de seguir hasta llegar a una completa implementación.

### 4.5.1 Arquitectura de la aplicación

La arquitectura llevada a cabo en este sistema es del tipo cliente-servidor. Este tipo de arquitectura es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta.

Consta de una serie de ventajas:

- **Centralización del control:** los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Esta centralización también facilita la tarea de poner al día datos u otros recursos.
- **Escalabilidad:** se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).
- **Fácil mantenimiento:** al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio (o se afectarán mínimamente). Esta independencia de los cambios también se conoce como encapsulación.

Aunque también posee una serie de desventajas:

- **Congestión del tráfico:** Cuando una gran cantidad de clientes envían peticiones simultáneas al mismo servidor, puede ser que cause muchos problemas para éste (a mayor número de clientes, más problemas para el servidor). Esto puede transformarse en una ralentización a la hora de realizar peticiones al servidor.
- **Inoperatividad en caso de caída:** Cuando un servidor está caído, las peticiones de los clientes no pueden ser satisfechas.
- **Especificaciones superiores para el servidor:** Normalmente se necesita software y hardware específico para satisfacer el trabajo. [26]

#### 4.5.2 Lenguajes de programación

Los lenguajes empleados en la implementación del sistema han sido variados. Por lo que voy a categorizarlos dependiendo de cada una de las partes del sistema.

##### *Raspberry Pi*

**Javascript y Python:** El software que incorpora el módulo de RaZberry está preparado para realizar una gestión completa de los sensores, ya que es posible añadir, configurar y/o eliminar sensores emparejados al módulo. En este software es posible añadir scripts basados en JavaScript para ejecutar tareas adicionales. En el caso de este sistema, como era necesario insertar en una base de datos ajena, se ha desarrollado un pequeño script en JavaScript, puesto que, el software encargado de controlar el módulo está prácticamente escrito en Javascript. Este código, que será explicado más detalladamente en el manual de instalación, realiza una llamada a otro script, esta vez, basado en Python, que es el encargado de insertar los eventos en la base de datos.

Javascript es un lenguaje de programación ligero, interpretado y orientado a objetos. Es el más usado en la creación de páginas webs, aunque también puede ser usado en otros ámbitos. Está basado en prototipos, y su uso es principalmente en el lado del cliente, permitiendo mejoras en la interfaz de usuario y permitiendo páginas webs dinámicas. [27]

Python, por su parte, es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor



medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma. [28]

### ***Servidor web***

**PHP:** El servidor web está basado en Apache, que será explicado en el siguiente apartado. Apache soporta entre otros lenguajes, PHP, que es el que se ha empleado a la hora de programar el servidor web, junto con un framework también programado en dicho lenguaje, que también será explicado más adelante.

PHP es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página web resultante. [29]

Gracias a su enorme expansión, puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno.

Permite la conexión a diferentes tipos de servidores de bases de datos tanto SQL como NoSQL tales como MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, SQLite o MongoDB. [30]

### ***Aplicación móvil***

**Java:** La aplicación móvil a desarrollar está basada en la plataforma Android, como se mencionó anteriormente, y esta plataforma basa su desarrollo en el lenguaje Java. Es cierto que existen aplicaciones que pueden ser ejecutadas en C++, pero la gran mayoría, están implementadas en Java.

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo, lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para ejecutarse en otra. [31]

### 4.5.3 Entornos de programación

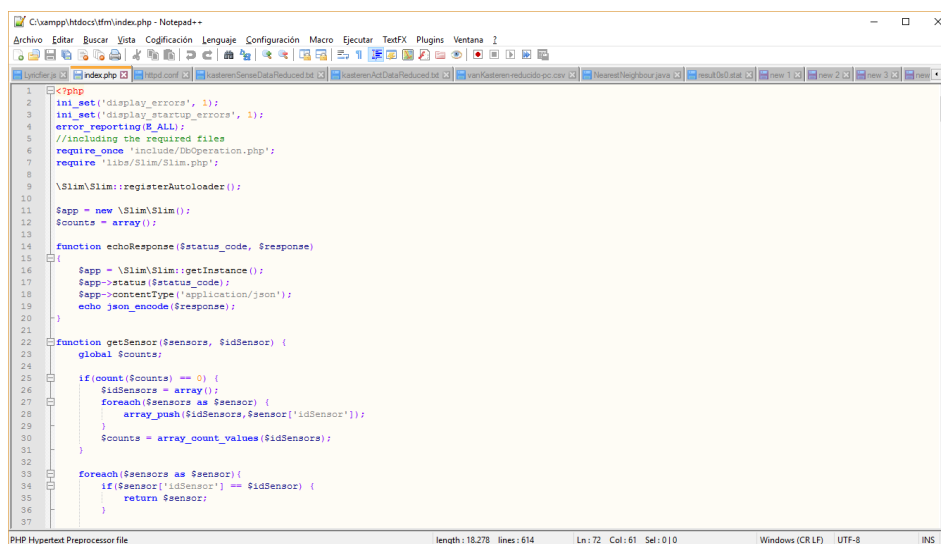
Un entorno de desarrollo integrado es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software.

Normalmente, un IDE consiste de un editor de código fuente, herramientas de construcción automáticas y un depurador. La mayoría de los IDE tienen auto-completado inteligente de código. Algunos IDE contienen un compilador, un intérprete, o ambos, tales como NetBeans y Eclipse.

Algunos IDE están dedicados específicamente a un lenguaje de programación, permitiendo que las características sean lo más cercanas al paradigma de programación de dicho lenguaje. Por otro lado, existen muchos IDE de múltiples lenguajes tales como Eclipse, NetBeans, Microsoft Visual Studio, etc. [32]

En mi caso dependiendo del lenguaje empleado, he utilizado distintos IDE. Son los siguientes:

- Notepad++ para el desarrollo del script en Javascript de la Raspberry y de la parte de PHP del servidor. Este IDE es multi-lenguaje y gracias a ello, podemos programar en múltiples



```
1 <?php
2 ini_set('display_errors', 1);
3 ini_set('display_startup_errors', 1);
4 error_reporting(E_ALL);
5 //including the required files
6 require_once 'include/DbOperation.php';
7 require 'libs/Slim/Slim.php';
8
9 \Slim\Slim::registerAutoloader();
10
11 $app = new \Slim\Slim();
12 $counts = array();
13
14 function echoResponse($status_code, $response)
15 {
16     $app = \Slim\Slim::getInstance();
17     $app->status($status_code);
18     $app->contentType('application/json');
19     echo_json_encode($response);
20 }
21
22 function getSensor($sensors, $idSensor) {
23     global $counts;
24
25     if(count($counts) == 0) {
26         $idsensors = array();
27         foreach($sensors as $sensor) {
28             array_push($idsensors, $sensor['idSensor']);
29         }
30         $counts = array_count_values($idsensors);
31     }
32
33     foreach($sensors as $sensor) {
34         if($sensor['idSensor'] == $idSensor) {
35             return $sensor;
36         }
37     }
38 }
```

Ilustración 24 - IDE Notepad++

lenguajes, adaptándose a cada uno de ellos.

- Spyder para la implementación del script en Python. Este IDE es específico para la programación de Python siendo indispensable su uso.

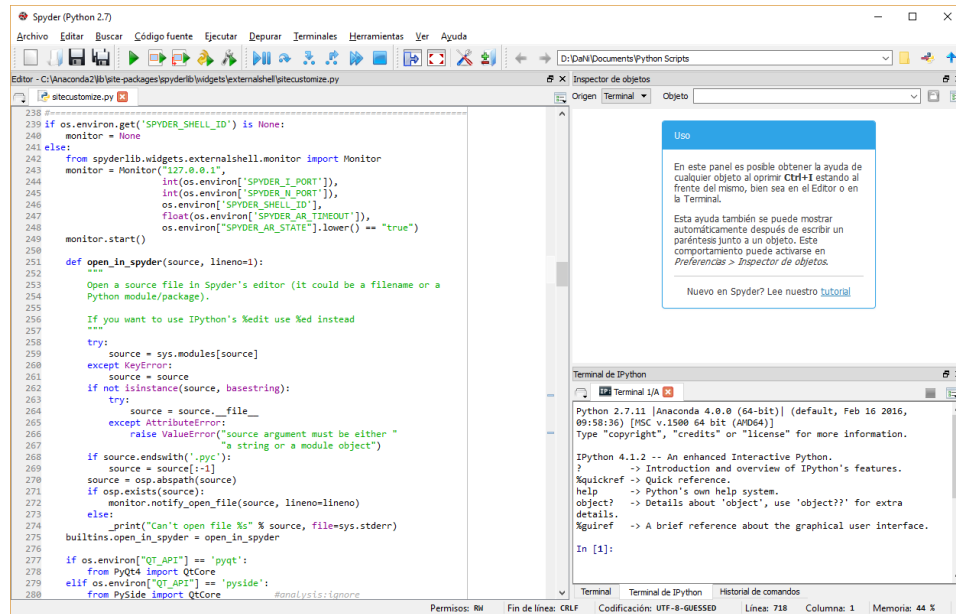


Ilustración 25 - IDE Spyder

- Android Studio para toda la implementación de la aplicación Android. Es un IDE muy completo, cómodo y fácil de usar, y que, proporciona todas las herramientas necesarias para una implementación de una aplicación en Android.

#### 4.5.4 Servidor web

XAMPP es un software libre que contiene varias herramientas como el servidor web Apache, el

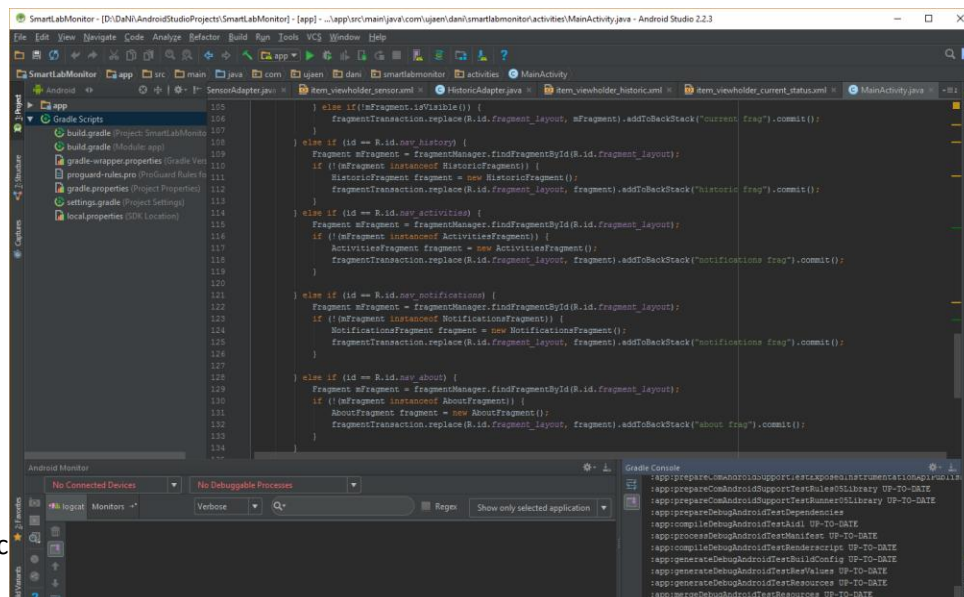


Ilustración 26 - Android Studio

gestor de bases de datos MySQL, Tomcat, y los intérpretes para PHP y Perl. [33]

En el caso de este sistema, únicamente ha sido necesario el uso del servidor Apache, el gestor de base de datos MySQL y el intérprete de PHP.

Apache es un servidor web HTTP desarrollado colaborativamente con el objetivo de crear un servidor HTTP robusto, de uso gratuito, y de fácil configuración. El proyecto es administrado conjuntamente por un grupo de voluntarios ubicados en todo el mundo, utilizando Internet para comunicar, planificar y desarrollar el servidor y la documentación relacionada. Este proyecto es parte de "Apache Software Foundation". Además, cientos de usuarios han aportado ideas, código y documentación al proyecto. [34]

Apache sirve de base para la instalación del servicio web, que es el encargado de suministrar la información necesaria a la aplicación. Pero primero veamos que es un servicio web.

#### *4.5.4.1 Servicios web*

Un servicio web se podría definir como un conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la Web. Estas aplicaciones o tecnologías intercambian datos entre sí con el objetivo de ofrecer unos servicios. Estos servicios proporcionan mecanismos de comunicación estándares entre diferentes aplicaciones, que interactúan entre sí para presentar información dinámica al usuario. Para proporcionar interoperabilidad y extensibilidad entre estas aplicaciones, y que al mismo tiempo sea posible su combinación para realizar operaciones complejas, es necesaria una arquitectura de referencia estándar.

Las ventajas de usar un servicio web son las siguientes: [35]

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.

#### 4.5.4.1.1 REST

En el caso de nuestro sistema se ha optado por usar la arquitectura REST que, haciendo uso del protocolo HTTP, proporciona una API para poder realizar diferentes operaciones entre la aplicación que ofrece el servicio web y el cliente, en este caso, la aplicación móvil.

REST tiene una serie de diseños clave que la hacen ser la escogida en la mayoría de APIs existentes y son los siguientes:

- Un protocolo cliente-servidor sin estado: cada mensaje HTTP contiene toda la información necesaria para comprender la petición. Como resultado, ni el cliente ni el servidor necesitan recordar ningún estado de las comunicaciones entre mensajes. Sin embargo, en la práctica, muchas aplicaciones basadas en HTTP utilizan cookies y otros mecanismos para mantener el estado de la sesión.
- Posee un conjunto de operaciones bien definidas que se aplican a todos los recursos de información: HTTP en sí define un conjunto pequeño de operaciones, las más importantes son POST, GET, PUT y DELETE. Con frecuencia estas operaciones son equivalentes a las operaciones CRUD en bases de datos.
- Una sintaxis universal para identificar los recursos. En un sistema REST, cada recurso es direccionable únicamente a través de su URI.
- El uso de hipermedios, ya que, la información transmitida en un sistema REST es típicamente HTML o XML. Como resultado de esto, es posible navegar de un recurso REST a muchos otros, simplemente siguiendo enlaces sin requerir el uso de registros u otra infraestructura adicional.

#### 4.5.4.2 *Framework PHP*

Para diseñar el servicio web anterior es necesario un framework que proporcione una forma clara y organizada de crear la API necesaria para implantar el servicio web. El framework escogido ha sido Slim.

Slim es un micro-framework, dado que prescinde de muchas de las características que, por ejemplo, Symfony o CodeIgniter incluyen. Sin embargo, para sistemas donde lo único que se desea implementar es una API, es una solución a tener muy en cuenta, como en este proyecto.

Emplea el patrón Modelo-Vista-Controlador (MVC), el cual separa los datos, la lógica y la interfaz de una aplicación creando tres componentes distintos, modelo, vista y controlador respectivamente. De esta forma se busca facilitar el desarrollo de aplicaciones y el posterior mantenimiento.

En el caso de nuestro sistema, la componente de la interfaz no es necesaria, ya que solo queremos construir una API que sirva de comunicación con la aplicación móvil.

Sus principales características son:

- Creador de rutas potente
  - Soporta métodos HTTP estándares y personalizados.
  - Redirección de rutas
  - Parámetros de ruta personalizables
- Vistas personalizadas
- Caché
- Login de usuarios
- Gestión de errores
- Encriptación de cookies

#### 4.5.4.3 MySQL

Por último, la pieza clave del sistema es el servidor de base de datos, donde se almacenan toda la información del sistema. Como se ha mencionado anteriormente, utiliza el gestor de base de datos MySQL, el cual es administrado por la herramienta PhpMyAdmin.

Esta herramienta fue creada en el año 1998 por Tobias Ratschiller. Fue escrita en su totalidad en PHP, ya que se basaba en el servidor Apache, del que antes hablamos, y hoy en día sigue basándose.

Tiene como ventajas, que posee una interfaz gráfica muy intuitiva y sencilla para el usuario medio, ya que prácticamente la creación de una base de datos y configuración de sus tablas y atributos, se pueden realizar de manera visual, sin tener que conocer la sintaxis del lenguaje MySQL.

Como características que tiene PhpMyAdmin podemos destacar las siguientes:

- Tiene una intuitiva interfaz web
- Realiza importaciones de datos desde CSV y SQL
- Exporta datos a varios formatos: CSV, SQL, XML, PDF, Word, Excel y otros

- Tiene posibilidad de administrar múltiples servidores
- Crea gráficos PDF del diseño de la base de datos
- Crea consultas complejas usando Query-by-Example (QBE)
- Realiza búsquedas globales en una base de datos o un subconjunto de esta
- Transforma datos almacenados a cualquier formato usando un conjunto de funciones predefinidas, tal como BLOB.
- Posee herramientas para monitorizar las actividades del servidor MySQL tales como conexiones, procesos, uso de CPU/Memoria, etc.

#### 4.5.5 Aplicación móvil desarrollada en Android

Como se ha especificado varias veces a lo largo de la documentación, se optó por realizar una aplicación Android desde la que se pudiera monitorizar el ambiente inteligente. El motivo de esta decisión viene marcado por varios hechos, como que, a fecha del tercer cuatrimestre de 2016, Android contaba con el 87,5 % de cuota de mercado en comparación con su principal competidor, iOS, que únicamente posee un 12,1 % [36]. También, es importante remarcar que, recientemente desarrollé una aplicación Android para la asignatura “Tecnologías y desarrollo en dispositivos móviles”, de manera que tengo mucho más reciente el uso de Android que el de iOS.

Por tanto, describamos qué es Android. Es un sistema operativo basado en el núcleo Linux. Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tablets o tabléfonos; y también para relojes inteligentes, televisores y automóviles. Inicialmente fue desarrollado por Android Inc., empresa que Google respaldó económicamente y más tarde, en 2005, compró.

Tiene una gran comunidad de desarrolladores creando aplicaciones para extender la funcionalidad de los dispositivos. A la fecha, se ha llegado ya al 1.000.000 de aplicaciones disponibles para la tienda de aplicaciones oficial de Android: Google Play, sin tener en cuenta aplicaciones de otras tiendas no oficiales para Android como la tienda de aplicaciones Samsung Apps de Samsung, y Amazon Appstore de Amazon.

Una de las funciones estrella de los dispositivos móviles y que poco a poco están llegando a otros ámbitos como los ordenadores personales son las notificaciones. Éstas son un sistema el cual permite alertar al usuario de un suceso ocurrido en una aplicación determinada.

Estas son posibles gracias al sistema que incorpora Google para gestionarlas, denominado Firebase Cloud Messaging.

Firebase Cloud Messaging (FCM) para Android es un servicio que permite enviar datos desde tu servidor a dispositivos Android y también recibir mensajes de dispositivos en la misma conexión. Hereda la infraestructura principal de GCM (Google Cloud Messaging), su predecesor.

El funcionamiento de este consiste en que existen múltiples temas a los que una aplicación puede suscribirse. En el momento que se quiera enviar una notificación al dispositivo, se envía a través de este tema, y todos los dispositivos que estén suscritos al tema en cuestión, recibirán la notificación.

FCM e incorpora una serie de mejoras con respecto a éste:

- Simplifica el desarrollo del cliente. Ya no es necesario escribir la lógica de suscripción a un tema.
- Habilita una nueva forma de enviar notificaciones sin necesidad de emplear un servidor
- Con FCM, cualquier persona puede enviar notificaciones a un grupo específico de personas.
- GCM soporta hasta un millón de suscriptores. FCM no tiene esta limitación.
- Es posible enviar mensajes a los temas a través de una consola de notificaciones desde la página web de FCM.

En el caso de nuestro sistema, funciona de la siguiente manera:

- La Raspberry envía un aviso a FCM cada vez que ocurre un evento, distinguiendo cada sensor con un tema distinto y único.
- FCM recibe el aviso y envía la notificación al tema especificado.
- Los dispositivos suscritos a dicho tema recibirán la notificación.



El esquema del funcionamiento es el siguiente:

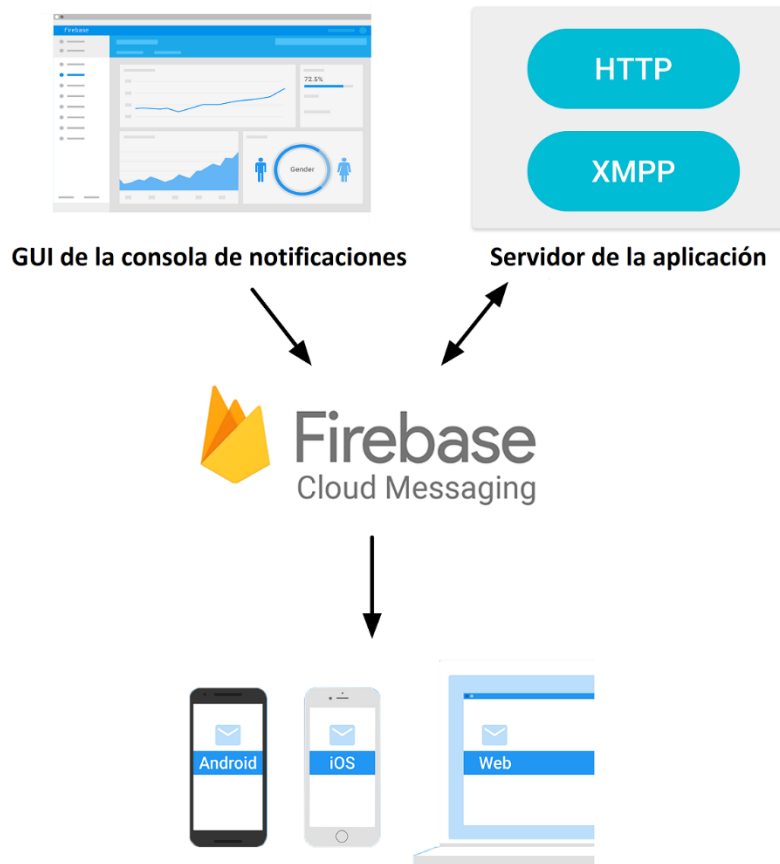


Ilustración 27 - Esquema de FCM

#### 4.5.6 Desarrollo de la implementación del sistema

Una vez realizada la experimentación mencionada en el capítulo 3, se ha procedido a implementar el resto del sistema. Para ello, dado que ya teníamos los sensores, la Raspberry Pi con sus accesorios y el módulo RaZberry, vistos en el capítulo 2, el siguiente paso ha sido la puesta en marcha de los sensores, así como de la propia Raspberry, ya que por defecto no incorpora ningún sistema operativo instalado. Se ha elegido para instalar el sistema operativo “Raspbian”, ya que, es el único sistema operativo oficial de todos los existentes.

Después de tener instalado el SO en la Raspberry, ha sido necesario instalar el software que permite el uso del módulo RaZberry. Gracias a este software ha sido posible vincular los sensores.

Una vez vinculados todos los sensores, teniendo los scripts de JS y Python, anteriormente mencionados, ya programados, ya estaría lista la primera parte de la implantación del sistema.

El siguiente paso ha consistido en el desarrollo de la aplicación móvil, la cual consta de los siguientes apartados:

- **Estado actual:** Se muestran todos los sensores que están instalados en el sistema, el estado en el que se encuentran actualmente y cuando fue la última vez que cambió su estado.
- **Histórico:** Se muestra un listado de todos los eventos ocurridos desde la instalación de los sensores.
- **Actividades:** Se muestran todas las actividades que han sido reconocidas por el sistema, junto con la fecha de inicio y de fin de dicha actividad.
- **Notificaciones:** Apartado donde se puede activar o desactivar la notificación de que un sensor ha cambiado de estado.



# CAPÍTULO 5

## CONCLUSIONES

---

En este trabajo fin de máster se ha propuesto un sistema de monitorización, basado en sensores de bajo coste, de un ambiente inteligente junto a un sistema de reconocimiento de actividades con baja carga computacional. Las ventajas que aporta un sistema de monitorización en el área de envejecimiento son diversas, entre ellas, permiten a las personas del entorno estar más cerca de los suyos y también como soporte a los cuidadores informales. A día de hoy, el precio del sistema de monitorización completo oscila entre 1000 y 1500 €.

El sistema se ha implementado ha contado con una red de sensores desplegada en un ambiente inteligente, en nuestro caso, el laboratorio de inteligencia ambiental de la Universidad de Jaén. Dicho sistema recoge los cambios que se producen en el laboratorio, permitiendo conocer las iteraciones que se realizan como, por ejemplo, abrir puertas, tirar de la cisterna, sentarse en el sofá, etc. El sistema que se ha desplegado cuenta con sensores bajo coste inalámbricos que envían la información a una Raspberry Pi.

Además, este TFM no solamente ha propuesto un sistema de monitorización de ambientes, si no que ha ido más allá, ya que ha tratado el problema de reconocer una actividad a partir de una secuencia de eventos. Para ello, se ha propuesto el uso del algoritmo KNN como algoritmo de clasificación de la actividad junto a un algoritmo de generación de prototipos para reducir el coste computacional, aumentando la autonomía del sistema.

Se ha realizado una experimentación de algoritmos de generación de prototipos con diferentes dataset de reconocimiento de actividades que pertenecen a diversas configuraciones de ambientes inteligentes con el objetivo de identificar el mejor algoritmo para cada configuración de ambiente inteligente. Teniendo en cuenta la configuración del laboratorio de inteligencia ambiental de la Universidad de Jaén, se ha escogido el algoritmo PSCSA para ser utilizado, ya que es el algoritmo que mejores resultados de precisión proporcionaba para la configuración más cercana a la de Jaén. Dicho algoritmo ha sido evaluado en el laboratorio de inteligencia ambiental proporcionando una precisión de acierto de un 80%.

Finalmente, se ha realizado el análisis, diseño y desarrollo de una aplicación móvil en Android donde un usuario puede monitorizar el ambiente inteligente y las actividades que se realizan en él en cualquier lugar y en cualquier momento a través de su dispositivo inteligente.

Por tanto, en este trabajo se ha realizado un estudio para llevar a cabo un sistema de monitorización que pueda ser desplegado en los hogares de personas mayores a un precio muy competitivo con el objetivo de monitorizar su interacciones y actividades para permanecer el mayor tiempo posible en sus hogares, a la vez, que sirve de ayuda a los cuidadores informales de estos.

## 5.1 CONCLUSIONES PERSONALES

---

La realización de este proyecto ha sido un reto muy interesante y satisfactorio, puesto que he disfrutado realizándolo y aprendiendo aspectos nuevos del Internet de las Cosas, del cual desconocía.

El Internet de las Cosas es aún un mundo de posibilidades por explotar y que irá expandiéndose año tras año hasta convertirse en algo tan usual como lo son nuestros smartphones de hoy en día. En un futuro no muy lejano podremos controlar prácticamente todo nuestro hogar desde nuestro dispositivo inteligente, y poder interactuar con la mayoría de dispositivos que tengamos en la casa de manera unificada, cosa hasta ahora prácticamente imposible. Espero con grandes ganas que llegue ese momento, puesto que es un tema que personalmente me apasiona y gracias a este trabajo he podido conocer a fondo el funcionamiento de una parte de este mundo como es el Internet de las Cosas.

Como desarrollador, he adquirido nuevos conocimientos relativos a cómo funcionan los sensores que componen el ambiente inteligente, como configurarlos, entender como transmiten la información al módulo receptor y como procesarla. He aprendido lo que son los algoritmos de generación de prototipos y para qué se usan. He comprendido el funcionamiento del algoritmo de clasificación KNN y he aprendido a usar herramientas de aprendizaje automático como KEEL. He conocido el mundo de Raspberry, que, si bien sabía de su existencia, nunca había tratado con una, y me ha gustado tanto, que he terminado comprando una para uso personal. Probablemente de no haber realizado este proyecto no habría llegado a descubrir la inmensa cantidad de funcionalidades que este mini-ordenador es capaz de ofrecer.

Por otro lado, el poder realizar la instalación física y comprobar que efectivamente todo funcionaba correctamente, ha sido algo gratificante, puesto que, no siempre se disponen de instalaciones físicas donde poder realizar estos proyectos.

En cuanto a la aplicación Android, he afianzado los conocimientos que adquiriré en la asignatura del Máster, que fue de las que más me gustaron, y ha sido un gusto poder desarrollar la aplicación en un sistema tan famoso y conocido como Android. Es otro mundo que también me entusiasma.

En general, estoy muy satisfecho con la labor desempeñada y sería interesante llevar algún día a cabo un sistema parecido en mi hogar, añadiendo otros sensores como de temperatura, humedad, luminosidad, etc., y poder realizar históricos con los datos obtenidos.

## 5.2 TRABAJOS FUTUROS

---

Respecto a los futuros trabajos que derivan del presente trabajo se podrían encaminar en dos vertientes, una investigadora y una técnica.

Respecto a la vertiente investigadora, se encuentra el estudio de la segmentación de actividades. Este problema es un gran reto en la comunidad científica que consiste en delimitar el tiempo en el que se produce la actividad a través del estudio del tamaño de ventana donde se encuentran los eventos que serán incluidos en el reconocimiento de actividades.

Respecto a la vertiente técnica, una característica muy interesante para su ampliación es la internacionalización. Ésta consiste en la traducción de las frases existentes en la interfaz de la aplicación a múltiples idiomas, de manera que, la aplicación adapta el lenguaje empleado dependiendo del idioma en el que se ejecute. Además, aporta una serie de ventajas:

Entre sus ventajas:

- Se usa el mismo ejecutable en los distintos países.
- El mercado es mayor
- No es necesario efectuar un desarrollo internacional del producto una vez acabada la primera versión
- Se utilizan los recursos más eficientemente
- El mantenimiento del código y la inclusión de nuevos idiomas es menos costoso

Gracias a éste es posible traducir todas las cadenas de frases existentes en la aplicación desde una sencilla interfaz, en la que se muestran los distintos idiomas y la cadena original. Pero esta tarea

necesita su tiempo y, además, los datos recogidos del servidor únicamente estaban en español, por lo que dificultaba en gran medida esta internacionalización.





# ANEXO I

# MANUAL DE INSTALACIÓN

---

El propósito de este manual es explicar la instalación del proyecto realizado, junto con las aplicaciones necesarias, además de la configuración para su correcto uso.

Estará dividido en varios apartados correspondientes al componente que se desea instalar, ya que este proyecto consta de varios de ellos.

## A.1 RASPBERRY PI

---

### A.1.1 Puesta en marcha

Cuando adquirimos una Raspberry Pi, ésta viene sin ningún sistema operativo instalado, por lo que, el primer paso a dar para ponerla en marcha es instalarle su sistema operativo en la tarjeta microSD que hemos adquirido junto con la Raspberry Pi. En mi caso, como ya mencioné anteriormente, se instalará Raspbian que puede ser descargado de [aquí](#).

Una vez descargado, es necesario extraer el archivo que está en el interior del .zip, que será el que se escribirá en la tarjeta. Para poder escribir en la tarjeta, es necesario el software “Win32 Disk Imager”, que puede ser descargado de [aquí](#). Al abrirlo se mostrará una interfaz como la siguiente:

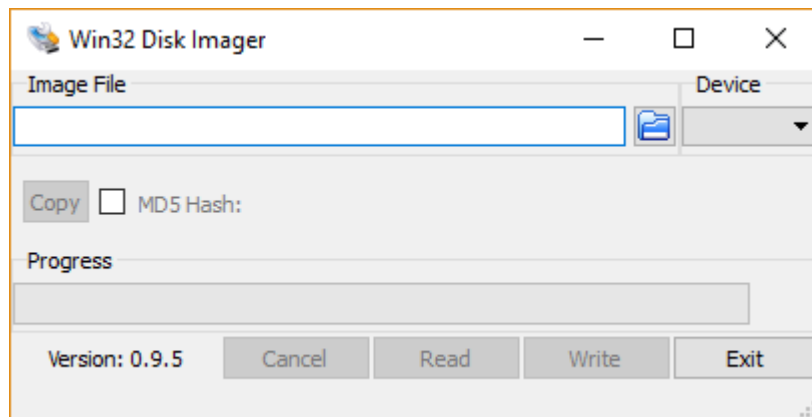


Ilustración A. 1 – Win32 Disk Imager

En Image File deberemos seleccionar el fichero .img extraído anteriormente y en Device, la unidad donde esté insertada la tarjeta microSD, y a continuación, pulsar en el botón Write.

Una vez completado el proceso, ya podemos insertar la tarjeta microSD en la Raspberry Pi.

### A.1.2 Instalación del módulo RaZberry

Antes de arrancar la Raspberry Pi, es recomendable instalar el módulo RazBerry, para dejarlo de esta manera, ya preparado. Para realizar esto es muy importante tener en cuenta donde se coloca el módulo, ya que una mala instalación física puede dejar inservibles ambos dispositivos.

Los GPIO de la Raspberry son los siguientes:

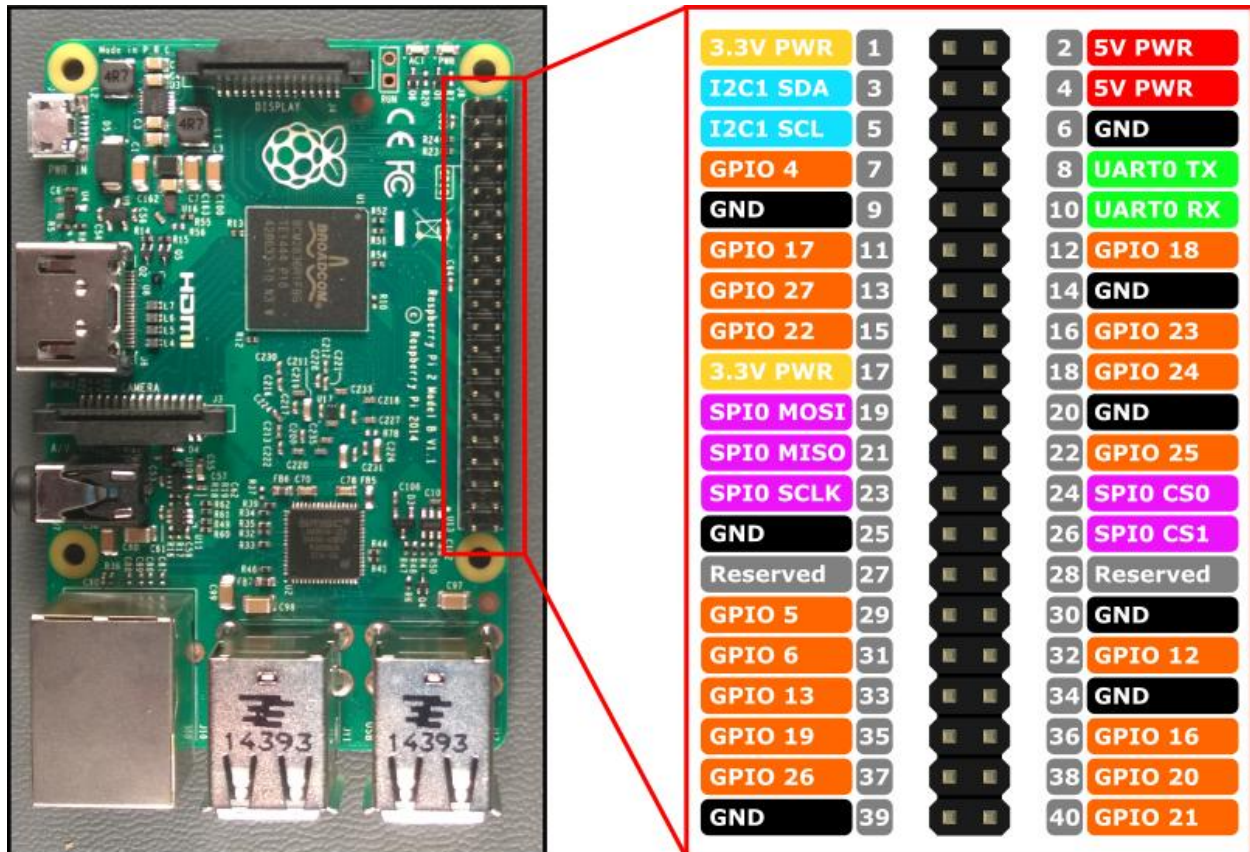


Ilustración A. 2 - Conexiones GPIO

El módulo deberá ser instalado justo empezando en el pin 1 hasta el pin 10, de manera que quede así:

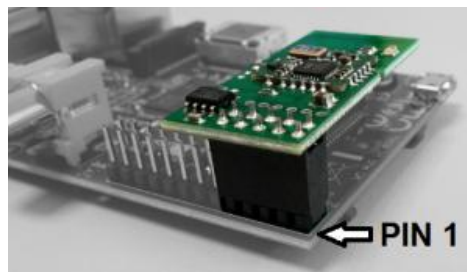


Ilustración A. 3 - Instalación del módulo en los GPIO

Tras ser instalado, es el momento de encender la Raspberry, conectándole el cargador de 5V en el conector microUSB disponible en la placa. Hecho esto, la Raspberry Pi arrancará y se mostrará el sistema operativo.<sup>1</sup>

Una vez configurado el Wi-Fi, no explicado, puesto que es un paso trivial, es el momento de descargar el software de Raspberry Pi. Es importante mencionar que es necesario ser root para realizar esta instalación. El usuario es “pi” y la contraseña por defecto “raspberrypi”, sin comillas.

Para ello ejecutaremos la siguiente instrucción en la terminal:

```
wget -q -O - raspberrypi.org/install | sudo bash
```

Tras ejecutarla, empezará un proceso que puede tardar unos minutos de duración. Una vez terminado el proceso, ya tendremos listo el software que controla el módulo Raspberry Pi.

### A.1.3 Vinculación de los sensores

El siguiente paso a realizar es la vinculación de los sensores con el módulo Raspberry Pi. Para lograr esto, es tan sencillo como acceder a la siguiente página desde la Raspberry:

```
http://localhost:8083/smarthome/
```

Una vez entremos<sup>2</sup>, nos aparecerá la siguiente interfaz:

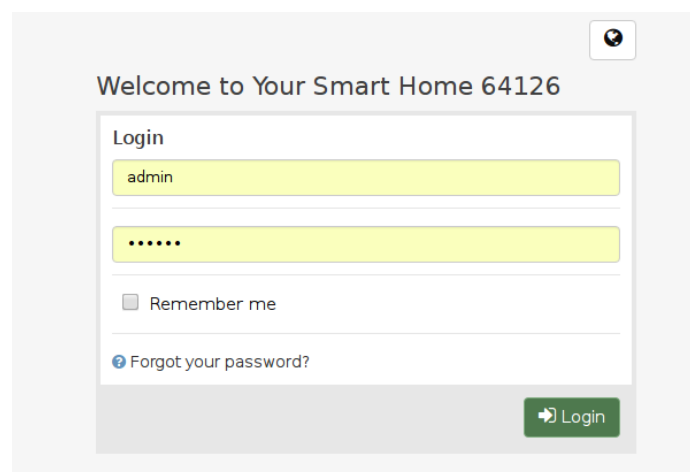


Ilustración A. 4 - Login del software

<sup>1</sup> Es imprescindible disponer de un monitor con conexión HDMI para poder visualizar el sistema operativo de la Raspberry Pi.

<sup>2</sup> Para entrar al sistema los datos son los siguientes: Usuario: admin, Contraseña: ceatic

Al iniciar sesión nos aparecerá la siguiente interfaz:

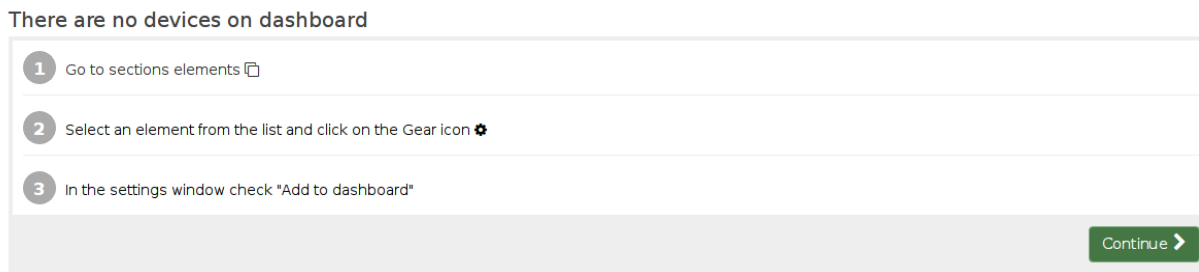


Ilustración A. 5 - Página principal

Al pulsar Continue, se mostrarán todos los sensores que tengamos vinculados con el sistema. A continuación, se muestran algunos de ellos:

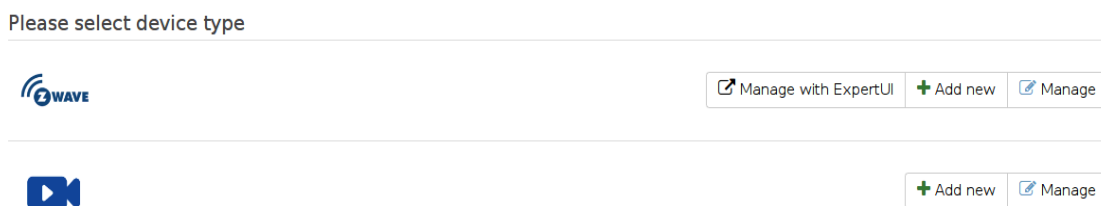


Ilustración A. 6 - Sensores vinculados


Para vincular un sensor, es necesario realizar una serie de pasos:

- Pulsamos en el engranaje de la parte superior derecha de la pantalla y a continuación en Devices.

Se mostrará la siguiente interfaz:



- Hacemos clic en Add new en la fila correspondiente a Z-Wave. Y se nos mostrará el siguiente botón:

 Add new Z-Wave Device and identify it automatically

- A continuación, se nos mostrará la siguiente pantalla:

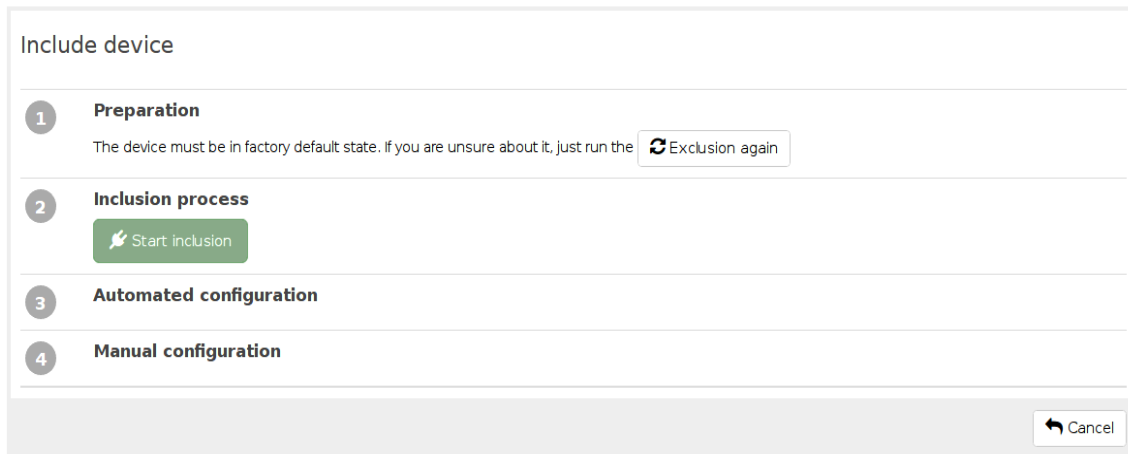


Ilustración A. 7 - Ventana de inclusión de un sensor

En ella se nos dan una serie de pasos, simplemente hemos de hacer clic en “Start Inclusion”, y el proceso de vinculación comenzará. Es importante mencionar que el sensor no debe de estar vinculado previamente, sino no funcionará el proceso de vinculación. En caso de estarlo, primero hemos de pulsar el botón de “Exclusion again”.

Una vez vinculados todos los sensores, es imprescindible realizar estos pasos, ya que, sin ellos, no podremos enviar los eventos al servidor web.

- Copiamos los dos archivos incluidos en la carpeta “RASPBERRY PI” en la siguiente ruta de la Raspberry Pi:

```
/opt/z-way-server/automation
```

- En el fichero main.js, antes de cerrar la llave “}”, insertamos el siguiente código:

```
executeFile('bind_events.js');
```

- Reiniciamos el servidor RazBerry con el siguiente comando:

```
sudo /etc/init.d/z-way-server restart3
```

Una vez realizado estos pasos, ya tendremos listos la puesta en marcha del módulo RaZberry.

---

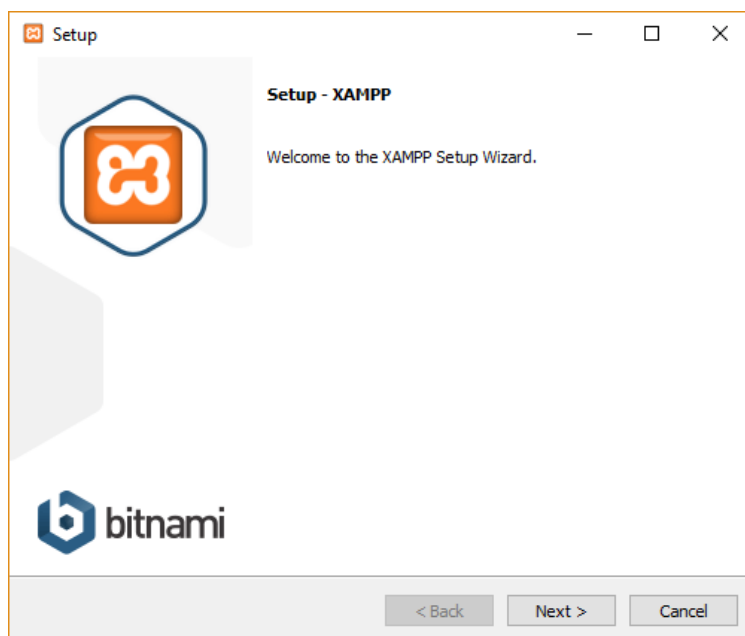
<sup>3</sup> Al ejecutar en modo superusuario nos pide identificarnos. El usuario es “pi” y la contraseña es “raspberry”.

## A.2 INSTALACIÓN DEL SERVIDOR WEB <sup>4</sup>

---

A continuación, se explicarán los pasos necesarios para instalar el servidor web que permitirá el uso de los servicios web.

En primer lugar, es necesario descargar el paquete XAMPP. Puede ser descargado de [aquí](#). Una vez descargado, lo ejecutamos y aparecerá una primera pantalla como esta, donde pulsamos Next.



*Ilustración A. 8 - Vista principal de instalación de XAMPP*

Al presionar el botón Next nos saldrá la siguiente imagen, donde debemos seleccionar qué componentes deseamos instalar:

---

<sup>4</sup> La instalación del servidor web se ha realizado sobre el sistema operativo Windows, por lo que, las imágenes expuestas y tutorial a seguir estará explicado para tal sistema operativo.



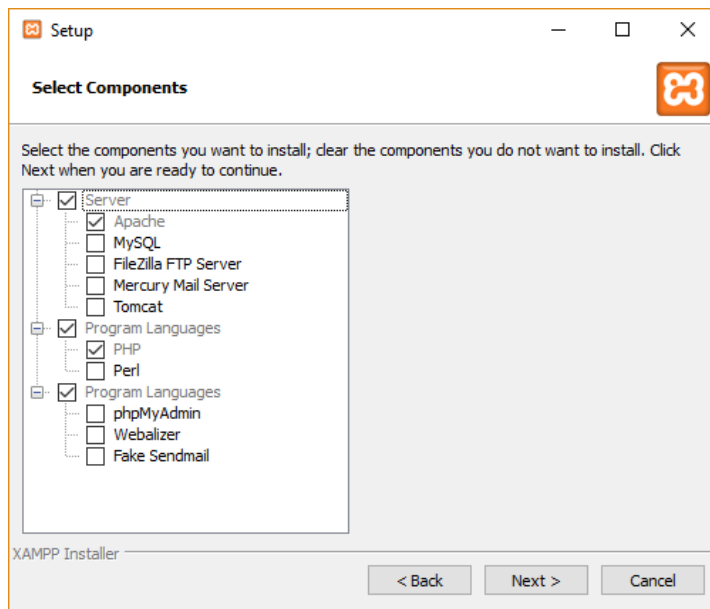


Ilustración A. 9 - Selección de los componentes

Con seleccionar únicamente los componentes de Apache y PHP es suficiente, ya que, la base de datos no se encuentra alojada en este servidor web. Una vez seleccionados, hacemos clic en Next y nos aparecerá la siguiente imagen, donde se configura la carpeta de instalación:

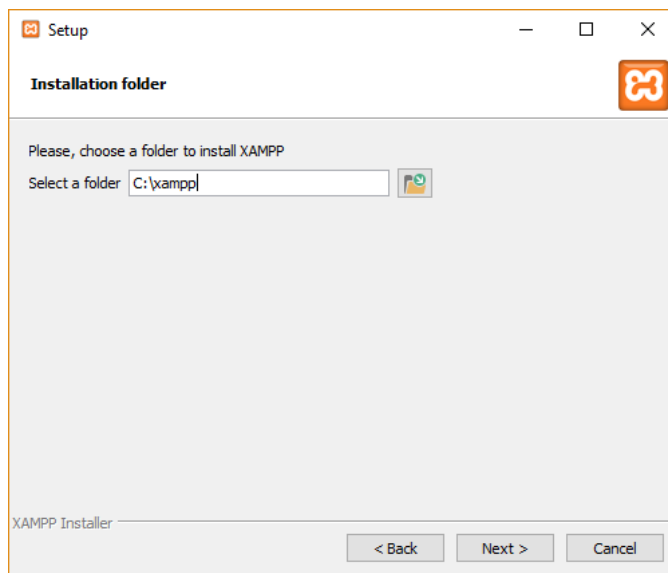


Ilustración A. 10 - Selección de la carpeta de instalación

La dejamos por defecto y hacemos clic en Next. En la siguiente ventana:

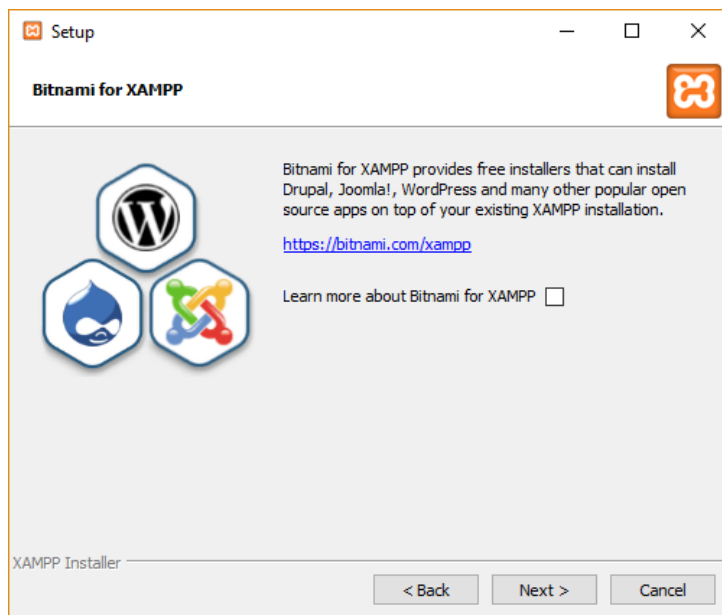


Ilustración A. 11 - Ventana de información

Deseleccionamos la opción que viene marcada por defecto y hacemos clic en Next.

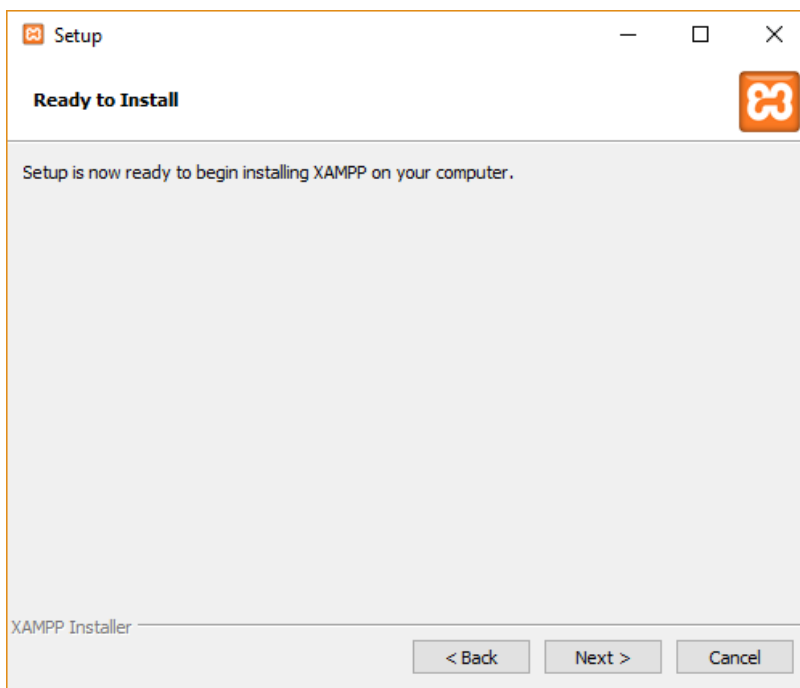


Ilustración A. 12 - Ventana previa a la instalación

Y volvemos a pulsar Next donde empezará la instalación:

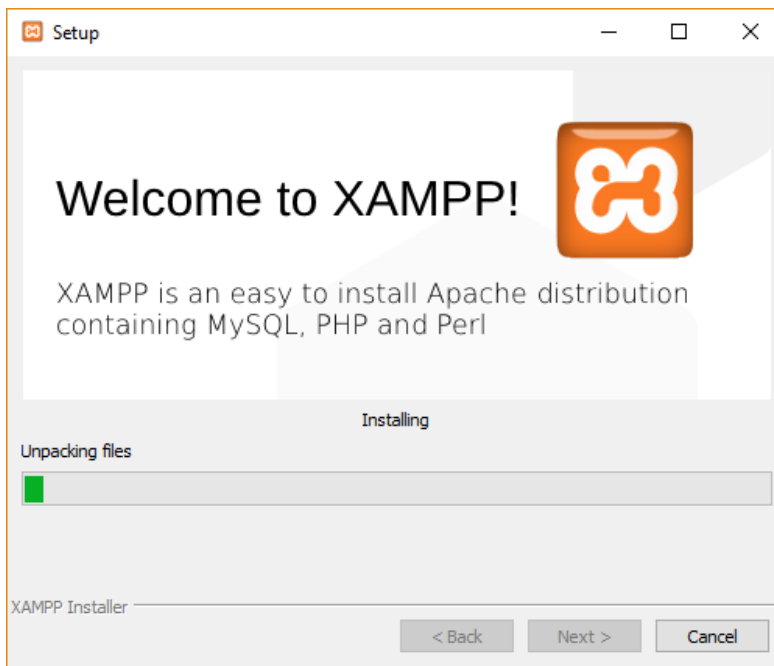


Ilustración A. 13 - Proceso de instalación

Una vez finalice la instalación, nos aparecerá la siguiente ventana:

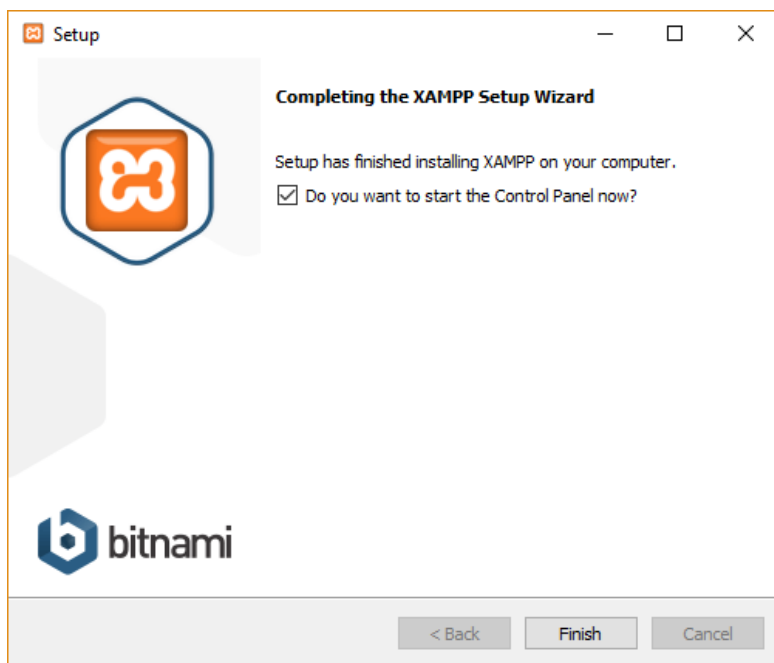


Ilustración A. 14 - Finalización de la instalación

Donde haremos clic en Finish y se cerrará el instalador y abrirá el panel de control de XAMPP:

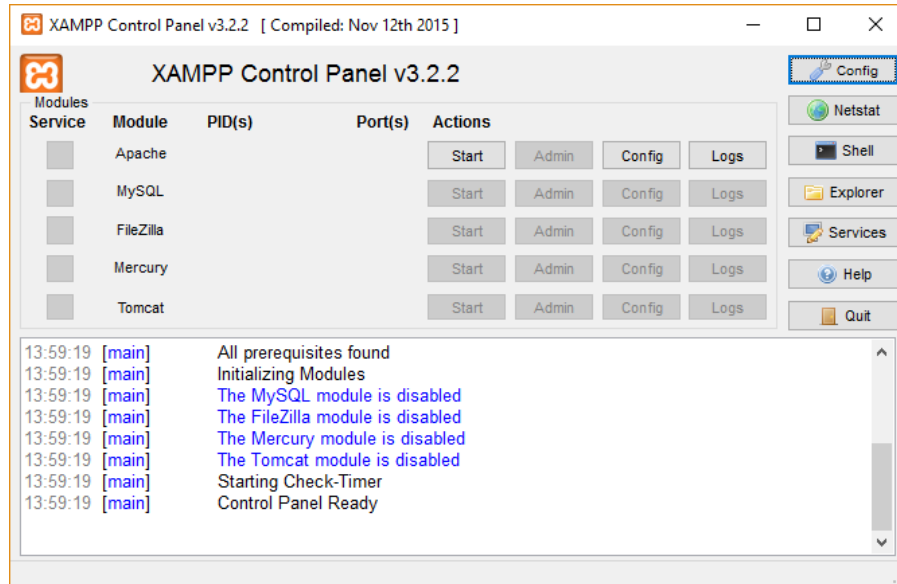


Ilustración A. 15 - Panel de control de XAMPP

Como únicamente hemos instalado el servidor web, solo podemos hacer clic en Start en el módulo de Apache.

El último paso a realizar será copiar el contenido de la carpeta “SERVIDOR WEB” en la siguiente ruta:

*C://xampp/htdocs.*

Con esto quedaría configurado el servidor web y listo para funcionar.



# ANEXO II

# MANUAL DE USUARIO

---

En este anexo se muestra, a modo de visita guiada, el manual de usuario de la aplicación. Se pretende, así, que el usuario aprenda a usar la aplicación.

Nada más entrar en ella, nos aparecerá la siguiente pantalla:

*Ilustración A. 16 - Pantalla principal*



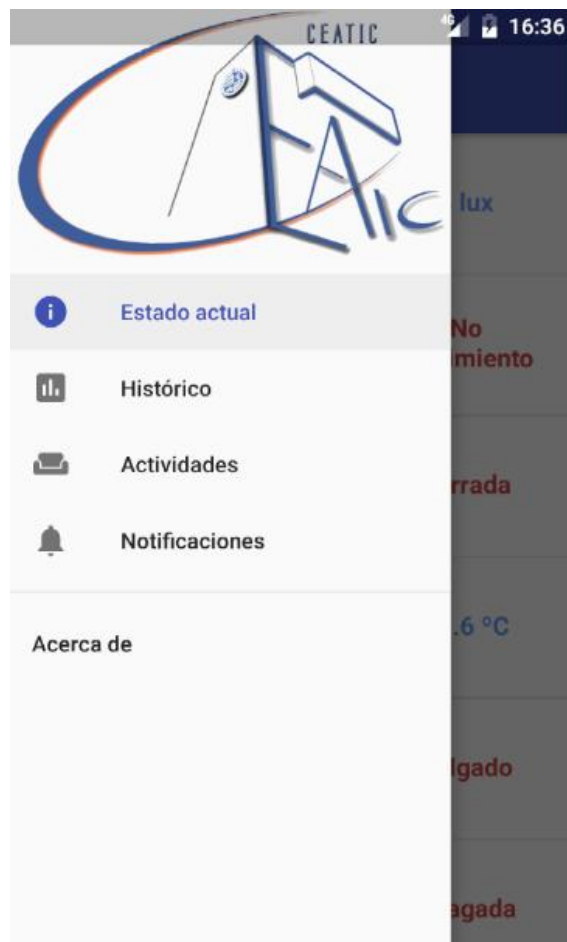
The screenshot shows a mobile application interface with a blue header bar containing a menu icon and the text 'Estado actual'. The main content area is a list of sensor data items, each with a title, ID, and status. The status is highlighted in blue or red. The items are: 'Sensor mov. entrada (Luz)' with ID TFD0007 and status '4 lux'; 'Sensor mov. entrada (Mov)' with ID TFD0007 and status 'No movimiento'; 'Puerta principal' with ID TFD0017 and status 'Cerrada'; 'Sensor mov. entrada (Temp)' with ID TFD0007 and status '20.6 °C'; 'Teléfono' with ID TFD0012 and status 'Colgado'; and 'Televisión' with ID TFD0016 and status 'Apagada'. Each item also includes a timestamp of the last event.

Sensor	ID	Estado	Fecha y hora
Sensor mov. entrada (Luz)	TFD0007	4 lux	Hoy a las 16:15:48
Sensor mov. entrada (Mov)	TFD0007	No movimiento	El 23/02/2017 a las 08:11:30
Puerta principal	TFD0017	Cerrada	El 23/02/2017 a las 08:09:22
Sensor mov. entrada (Temp)	TFD0007	20.6 °C	El 22/02/2017 a las 22:16:20
Teléfono	TFD0012	Colgado	El 17/02/2017 a las 13:24:21
Televisión	TFD0016	Apagada	El 16/02/2017 a las 19:50:20

En esta pantalla se muestran todos los sensores instalados en el ambiente inteligente con el último evento y fecha en la que se registró ese evento. También se muestra el nombre del sensor.



Si pinchamos en el botón de menú (los tres guiones de la parte superior izquierda), se nos despliega



*Ilustración A. 17 - Menú de la aplicación*

el siguiente menú:

Como podemos ver, aparecen 4 apartados, los cuales se subdivide la aplicación. Si nos vamos a Histórico nos aparecerá la siguiente pantalla:

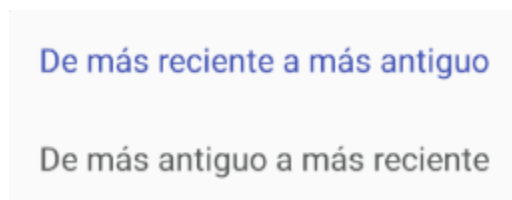
*Ilustración A. 18 - Pantalla de Histórico*



HOY	
Sensor mov. entrada (Luz) A las 16:15:48	4 lux
Sensor mov. entrada (Luz) A las 15:15:33	3 lux
Sensor mov. entrada (Luz) A las 14:15:16	4 lux
Sensor mov. entrada (Luz) A las 13:15:00	2 lux
Sensor mov. entrada (Luz) A las 12:14:45	3 lux
Sensor mov. entrada (Luz) A las 11:14:30	2 lux
Sensor mov. entrada (Luz)	3 lux

Aquí se nos muestra un listado de todos los eventos que han sido registrados, junto con el valor de cada uno de ellos y la fecha en la que fueron registrados.

En la parte superior a la derecha disponemos de dos botones; uno de ellos sirve para ordenar la lista; mientras que el otro sirve para filtrarla. Si pinchamos en el primer botón nos aparecerán estas



*Ilustración A. 19 - Listado de opciones de ordenación*

opciones:

La primera opción nos mostrará en la parte superior los eventos más recientes, mientras que la segunda opción mostrará los eventos más antiguos en la parte superior.

Si pinchamos en el segundo botón se nos muestra un listado de opciones:

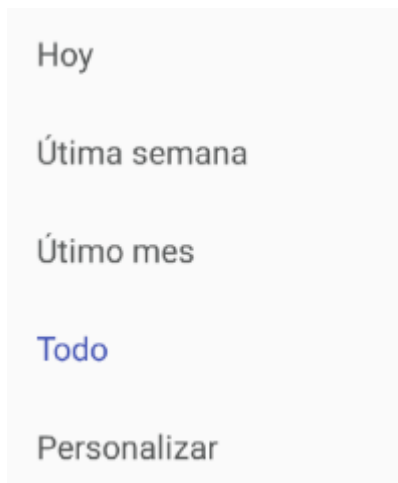


Ilustración A. 20 - Listado de opciones de filtrado

Estas opciones nos determinarán la cantidad de elementos que se mostrarán en el histórico. Es posible personalizar un día concreto sobre el que se quiere consultar los eventos existentes. Para ello, solo hay que pulsar en el botón “Personalizar” y se mostrará la siguiente ventana:



Ilustración A. 21 - Ventana de selección de fecha

Con lo que pulsando en el día que queramos consultar y aceptando, se nos mostraría los eventos de ese día, si los hay.

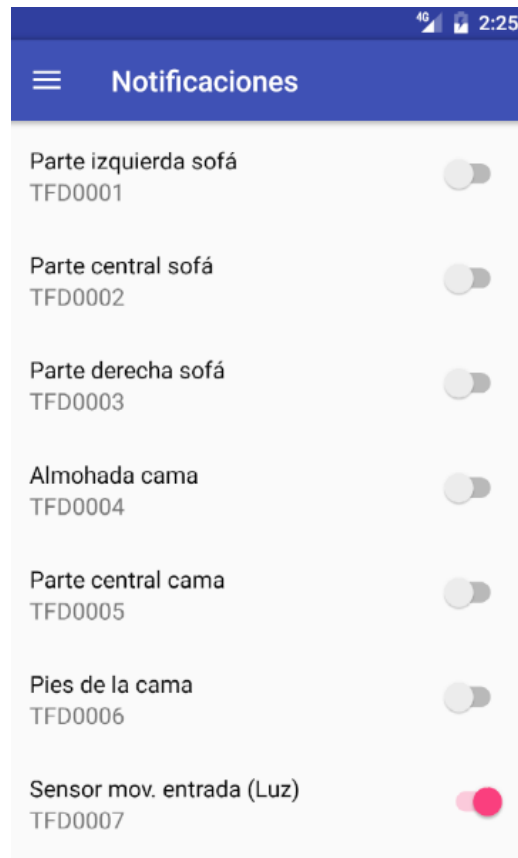
En el apartado de actividades, disponemos de un listado que muestra las actividades realizadas. Es posible ver la fecha de comienzo y de fin de las mismas, como se puede ver en el siguiente ejemplo:



*Ilustración A. 22 - Pantalla de actividades*

Al igual que en el apartado de Histórico, aquí también podemos realizar un filtrado y/u ordenación de la misma forma explicada anteriormente.

En el apartado de Notificaciones, disponemos de un listado de los sensores instalados en el ambiente inteligente, con un botón por cada sensor, como se puede observar en la siguiente imagen:



*Ilustración A. 23 - Pantalla de notificaciones*

Cada interruptor al lado de un sensor significa que se recibirá notificaciones cuando el sensor cambie de estado.

Por último, se dispone del apartado Acerca de, donde se muestra información relativa al creador de la aplicación:



*Ilustración A. 24 - Pantalla de Acerca de*

# BIBLIOGRAFÍA Y REFERENCIAS

---



- [1] P. Droege, *Intelligent Environments - Spatial Aspect of the Information Revolution*, Holanda: Oxford: Elsevier , 1997.
- [2] U. Nations, «United Nations - Department of Economic and Social Affairs,» 2013. [En línea]. Available: <https://www.un.org/development/desa/en/>.
- [3] R. e. al, «Life Extension Possibilities,» 2014. [En línea]. Available: <https://srconstantin.wordpress.com/2017/01/24/life-extension-possibilities/>.
- [4] «Economist Intelligence Unit,» 2011. [En línea]. Available: [http://www.eiu.com/public/topical\\_report.aspx?campaignid=DemocracyIndex2011](http://www.eiu.com/public/topical_report.aspx?campaignid=DemocracyIndex2011).
- [5] R. S. S. C. Ph.D. y S. B. Ph.D., *Older Adults and the Adoption of Healthcare Technology*, Springer US, 2013.
- [6] A. H. a. V. K. T. B. Moeslund, «A survey of advances in vision-based human motion capture and analysis,» *Comput. Vis. Image*, 2006.
- [7] L. B. a. S. Intille, de *Activity recognition from user-annotated acceleration data*, *Proc. Pervasive*, 2004, pp. 1-17.
- [8] A. P. a. N. G. Bourbakis, «“A survey on wearable sensor-based systems for health monitoring and prognosis,» *IEEE Trans. Syst.*, 2010.
- [9] N. U. Mushtaq, «Z-Wave,» [En línea]. Available: <https://en.wikipedia.org/wiki/Z-Wave>.
- [10] T. W. L. W. Z. T. X. L. J. Gu, «A pattern mining approach to sensor-based human activity recognition.,» 2011.
- [11] L. N. C. Chen, «Ontology-based activity recognition in intelligent pervasive environments,» 2009.
- [12] S. S. J. Geva, «Adaptive nearest neighbor pattern classifier.,» 1991.
- [13] J. B. T. Koplowitz, «On the relation of performance to editing in nearest neighbor rules.,» *Pattern*

- Recognit., 1981.
- [14] C. Chang, «Finding prototypes for nearest neighbor classifiers.,» 1974.
- [15] T. L. C. Raicharen, «A divide-and-conquer approach to the pairwise opposite class-nearest neighbor (POC-NN) algorithm.,» 2005.
- [16] S. D. J. C. J. H. F. Garcia, «Prototype selection for nearest neighbor classification: taxonomy and empirical study.,» 2012.
- [17] Y. G. E. G. M. R. A. C. L. Chen, «Similarity-based classification: concepts and algorithms.,» 2009.
- [18] I. M. K. L. D. Kononenko, «Introduction to Principles and Algorithms.,» Horwood Publishing Limited, Chichester, 2007.
- [19] S. D. J. C. J. H. F. Garcia, «Prototype selection for nearest neighbor classification: taxonomy and empirical study.,» 2012.
- [20] P. K. J. Devijver, «Pattern Recognition: A Statistical Approach.,» Prentice Hall, London, 1982.
- [21] I. P. J. M. C. a. G. O. Jacobson, «Un acercamiento a través de los casos de uso,» de *Ingeniería de Software Orientada a Objetos*, 1992.
- [22] E. S. K. B. B. Freeman, *Head First Design Patterns*, O'Reilly, 2008.
- [23] B. Roe, «Diseño de Interfaces de Usuario Usables,» 2004. [En línea]. Available: <http://mundogeek.net/traduccion/interfases-usuario-usables/gui.html>.
- [24] L. Villa, «Guías de estilo: diseño, normalización y usabilidad,» 2004. [En línea]. Available: <https://desarrolloweb.com/articulos/1513.php>.
- [25] «Material Design,» 2014. [En línea]. Available: [https://es.wikipedia.org/wiki/Material\\_design](https://es.wikipedia.org/wiki/Material_design).
- [26] «Cliente-Servidor,» 2016. [En línea]. Available: <https://es.wikipedia.org/wiki/Cliente-servidor#Ventajas>.
- [27] P. Netter, «About JavaScript,» 2015. [En línea]. Available: <https://developer.mozilla.org/en->

US/docs/Web/JavaScript/About\_JavaScript.

[28] «Python,» 2016. [En línea]. Available: <https://es.wikipedia.org/wiki/Python>.

[29] «PHP,» [En línea]. Available: <https://es.wikipedia.org/wiki/PHP>.

[30] P. Group, «Extensiones de bases de datos,» [En línea]. Available:  
<http://php.net/manual/es/refs.database.php>.

[31] «Java (lenguaje de programación),» [En línea]. Available:  
[https://es.wikipedia.org/wiki/Java\\_\(lenguaje\\_de\\_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n)).

[32] I. Ramos Salavert y M. D. Lozano Pérez, «Entornos de Desarrollo Integrados,» de *Ingeniería del software y bases de datos: tendencias actuales*, Universidad de Castilla La Mancha, 2000.

[33] «XAMPP,» 2016. [En línea]. Available: <https://es.wikipedia.org/wiki/XAMPP>.

[34] A. S. Foundation, «Apache,» 2015. [En línea]. Available:  
[http://httpd.apache.org/ABOUT\\_APACHE.html](http://httpd.apache.org/ABOUT_APACHE.html).

[35] W3C, «Guía Breve de Servicios Web,» 2014. [En línea]. Available:  
<http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>.

[36] S. Analytics, «Global Smartphone OS Market Share by Region: Q3 2016,»  
<https://www.strategyanalytics.com/access-services/devices/mobile-phones/smartphone/smartphones/market-data/report-detail/global-smartphone-os-market-share-by-region-q3-2016#.WLDrExpjJhF>.