



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior (Jaén)

Trabajo Fin de Máster

AMPLIACIÓN DE FUNCIONALIDADES DEL SISTEMA WEB UJAMI PARA LA MONITORIZACIÓN DE AMBIENTES INTELIGENTES

Alumno/a: Zafra Romero, Daniel

Tutor/a: Dra. Macarena Espinilla Estévez

Cotutor: Dr. Javier Medina Quero

Dpto.: Informática

Junio, 2018



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Dra. D^a. Macarena Espinilla Estévez,
y Dr. D. Javier Medina Quero, cotutor,
del Trabajo Fin de Master titulado:

**Ampliación de funcionalidades del sistema web UJAml para la
monitorización de ambientes inteligentes,**

que presenta D. Daniel Zafra Romero,
autorizan su presentación para defensa y evaluación en la
Escuela Politécnica Superior de Jaén.

Jaén, abril de 2018

El alumno:

Los tutores:

Daniel Zafra Romero

Macarena Espinilla Estévez

Javier Medina Quero

*En primer lugar, me gustaría expresar mi agradecimiento a los tutores de este trabajo,
Macarena Espinilla Estévez y Javier Medina Quero.
Todo el apoyo, experiencia y recomendaciones han sido uno de los grandes pilares
para que este trabajo fuese posible.*

*A mi familia,
y en especial a mis padres y a Carolina,
por su apoyo incondicional y su gran paciencia.
He llegado hasta aquí gracias a vosotros.*

*Y, por último, no
quisiera olvidarme de dar las gracias a todos mis compañeros,
tanto del máster como del CEATIC por todos los momentos
dentro y fuera de la universidad.
Gracias a todos.*

Índice

1	INTRODUCCIÓN	3
1.1	MOTIVACIÓN	4
1.2	PROPUESTA	8
1.3	OBJETIVOS.....	8
1.4	ESTRUCTURA DE LA MEMORIA	9
1.5	PLANIFICACIÓN TEMPORAL.....	10
1.5.1	<i>Estimación de tiempos</i>	10
1.5.2	<i>Diagrama de Gantt</i>	12
1.6	METODOLOGÍA SOFTWARE	13
1.6.1	<i>Iteraciones</i>	14
1.7	ESTIMACIÓN DE RECURSOS.....	14
1.7.1	<i>Recursos humanos</i>	15
1.7.2	<i>Recursos técnicos</i>	15
1.8	ESTIMACIÓN DE COSTES.....	17
2	AMBIENTES INTELIGENTES Y MINE V1	18
2.1	AMBIENTES INTELIGENTES.....	19
2.1.1	<i>Aplicaciones de los ambientes inteligentes</i>	22
2.1.2	<i>Laboratorio Inteligente (Universidad de Jaén)</i>	24
2.2	MIME V1.....	25
2.2.1	<i>Reconocimientos</i>	26
3	ESTADO DEL ARTE EN AMBIENTES INTELIGENTES	28
3.1	MIDDLEWARES	29
3.2	COMUNICACIÓN	30
3.3	ETIQUETADO DE ACTIVIDADES.....	37
3.3.1	<i>Tecnología NFC</i>	37
3.3.2	<i>Balizas posicionamiento</i>	40
4	INGENIERÍA DE SOFTWARE	49
4.1	DEFINICIÓN DE REQUISITOS	50
4.2	ANÁLISIS DEL SISTEMA.....	56
4.2.1	<i>Casos de uso</i>	56
4.2.2	<i>Escenarios</i>	63
4.3	DISEÑO.....	68
4.3.1	<i>Diseño de clases</i>	68
4.3.2	<i>Diseño de datos</i>	74
4.3.3	<i>Diseño de la interfaz en la aplicación móvil</i>	78
4.4	IMPLEMENTACIÓN.....	88
4.4.1	<i>Arquitectura</i>	88
4.4.2	<i>Tecnologías utilizadas</i>	90
4.4.3	<i>Servicios</i>	100
4.4.4	<i>Herramientas de desarrollo</i>	104
5	CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS	106
6	BIBLIOGRAFÍA	109
7	ANEXOS	112
7.1	CONTENIDO DEL CD-ROM	113

7.2	MANUAL DE INSTALACIÓN	114
7.2.1	<i>Servidor</i>	114
7.2.2	<i>Cliente</i>	118
7.2.3	<i>Aplicación móvil</i>	119
7.3	MANUAL DE USO	120
7.3.1	<i>Portal de usuario</i>	120
7.3.2	<i>Administración</i>	128
7.3.3	<i>Aplicación móvil</i>	133

Índice de tablas

Tabla 1 Estimación de tiempos	11
Tabla 2 Recursos humanos.....	15
Tabla 3 Equipo informático	16
Tabla 4 Costes fijos.....	16
Tabla 5 Costes servidor	17
Tabla 6 Costes licencias.....	17
Tabla 7 Costes totales	17
Tabla 8 Comparativa estándares.....	36
Tabla 9 Requisito no funcional 0	54
Tabla 10 Requisito no funcional 1	54
Tabla 11 Requisito no funcional 3	54
Tabla 12 Requisito no funcional 4	55
Tabla 13 Restricción 1	55
Tabla 14 Restricción 2	56
Tabla 15 Plantilla escenarios	63
Tabla 16 Escenario (Crear una actividad).....	64
Tabla 17 Escenario (Eliminar una actividad)	65
Tabla 18 Escenario (Consultar actividades en tiempo real)	65
Tabla 19 Escenario (Exportar flujo de eventos)	66
Tabla 20 Escenario (Etiquetar una actividad).....	67
Tabla 21 Símbolos del diagrama de clases	68
Tabla 22 Diagrama de paquetes.....	69
Tabla 23 Diagrama de clases	70
Tabla 24 Códigos de estado	100

Tabla de ilustraciones

Ilustración 1 Diagrama de Gantt	12
Ilustración 2 Ciclo de vida SCRUM.....	13
Ilustración 3 Tabla de iteraciones	14
Ilustración 4 Población 2018	20
Ilustración 5 Población 1993	20
Ilustración 6 Población 2056	21
Ilustración 7 Laboratorio Universidad de Jaén.....	24
Ilustración 8 Población 2018	28
Ilustración 9 Población 1993	28
Ilustración 7 Middleware	29
Ilustración 8 Esquema MQTT	31
Ilustración 12 Ejemplo temas MQTT	32
Ilustración 9 Esquema SOAP	33
Ilustración 10 Zonas SOAP.....	33
Ilustración 11 Esquema REST	34
Ilustración 12 Interés por protocolos.....	35
Ilustración 13 Interés por estándares	36
Ilustración 14 Laboratorio Universidad de Jaén.....	37
Ilustración 15 Campo magnético NFC	37
Ilustración 16 Etiqueta NFC.....	38
Ilustración 17 Plano de ejemplo.....	40
Ilustración 18 Ejemplo de posicionamiento mediante beacons	41
Ilustración 22 Baliza nearables.....	42
Ilustración 24 Estimación de proximidad	42
Ilustración 25 Beacons Balizas	43
Ilustración 19 APP DTTJax	44
Ilustración 20 Estimote Mirror	45
Ilustración 21 Posicionamiento WIFI.....	47
Ilustración 22 Triangulación GSM	48
Ilustración 23 Requisitos funcionales (Ampliar funcionalidades)	51
Ilustración 24 Requisitos funcionales (APP Móvil)	52
Ilustración 25 Casos de uso	57
Ilustración 26 Caso de uso: Gestionar actividades.....	57
Ilustración 27 Caso de uso: Gestionar Habitantes	58
Ilustración 28 Caso de uso: Consultar actividades	58
Ilustración 29 Caso de uso: Visualizar histórico	59
Ilustración 30 Caso de uso: Exportar actividades.....	60
Ilustración 31 Casos de uso aplicación móvil	61
Ilustración 32 Caso de uso: Identificarse.....	61
Ilustración 33 Caso de uso: Visualizar estado actual.....	62
Ilustración 34 Caso de uso: Etiquetar actividades	62
Ilustración 35 Caso de uso: Suscribirse	63
Ilustración 36 Diagrama de paquetes	72

Ilustración 37 Diagrama clases (APP Móvil)	73
Ilustración 38 Esquema conceptual modificado	76
Ilustración 39 StoryBoard (Inicio de sesión).....	79
Ilustración 40 StoryBoard (Ambientes disponibles).....	80
Ilustración 41 StoryBoard (Menú).....	80
Ilustración 42 StoryBoard (Estado actual de un ambiente)	81
Ilustración 43 StoryBoard (Etiquetado de actividades).....	82
Ilustración 44 StoryBoard (Notificaciones)	83
Ilustración 45 Fuente ROBOTO	84
Ilustración 46 Paleta de colores	85
Ilustración 54 Logo corporativo UJAMI MONITORING.....	85
Ilustración 47 Metáfora (Estado de un objeto).....	86
Ilustración 48 Metáfora (Volver y ajustes).....	87
Ilustración 49 Metáfora (Menú).....	87
Ilustración 50 Arquitectura cliente – servidor.....	88
Ilustración 51 Equema JDBC.....	91
Ilustración 52 Diagrama Modelo -Vista – Controlador	93
Ilustración 53 Cuota de mercado móvil	94
Ilustración 54 Cuota de mercado Android	95
Ilustración 55 Capas Android	96
Ilustración 56 Ciclos de vida Android	98
Ilustración 57 Topic en firebase	99
Ilustración 58 Ejemplo servicio REST - JSON	103
Ilustración 67 Logo IJ.....	104
Ilustración 60 Logo AS.....	104
Ilustración 61 Logo Atom	105
Ilustración 62 Logo Workbench	105
Ilustración 71 Estructura de la carpeta	113
Ilustración 63 Comprobación de Java. 1.8	114
Ilustración 64 Contraseña MariaDB	115
Ilustración 65 Comprobación de MariaDB	115
Ilustración 66 Panel de control Tomcat	117
Ilustración 67 Aplicación desplegada	117
Ilustración 68 Vista principal de la aplicación	118
Ilustración 69 Instalar dese fuentes desconocidas	119
Ilustración 70 Inicio de la aplicación web	120
Ilustración 71 Estado actual de un ambiente.....	121
Ilustración 72 Actividad sin beacon.....	122
Ilustración 73 Actividad con beacon	123
Ilustración 74 Lista de actividades en modo texto.....	124
Ilustración 75 Filtros de consulta	124
Ilustración 76 Menú rápido de exportación de actividades.....	125
Ilustración 77 Vista modal para exportar actividades.....	125
Ilustración 78 Configuración del fichero de salida	126
Ilustración 79 Ejemplo de fichero CSV	126
Ilustración 80 Configuración del fichero de salida	127
Ilustración 81 Flujo de eventos	127

Ilustración 82 Exportado de actividades	128
Ilustración 83 Panel de administración	128
Ilustración 84 Lista de usuarios	129
Ilustración 85 Contraseña SHA1	129
Ilustración 86 Lista de habitantes	130
Ilustración 87 Lista de actividades	131
Ilustración 88 Etiqueta NFC	131
Ilustración 89 Posicionando actividades	132
Ilustración 90 Inicio de sesión	133
Ilustración 91 Lista de ambientes	134
Ilustración 92 Menú	134
Ilustración 93 Estado actual	135
Ilustración 94 Etiquetado de actividad y configuración de etiquetado	136
Ilustración 95 Actividades realizándose	137
Ilustración 96 Suscripciones a eventos	138
Ilustración 97 Notificación en tiempo real	139

Capítulo 1

1 INTRODUCCIÓN

En este primer capítulo, se van a describir las ideas desarrolladas a lo largo de este trabajo de fin de máster, así como la motivación para la realización del mismo, su estructura y objetivos. También se detallarán convenientemente los aspectos relacionados con la metodología, la planificación temporal y el presupuesto establecido.

1.1 MOTIVACIÓN

Antes de comenzar y adentrarnos en la lectura de este trabajo de fin de máster, conviene indicar la motivación de su desarrollo. En el año 2015, fruto mi trabajo fin de grado, se desarrolló un sistema web para la monitorización de entornos de inteligencia ambiental, denominado MINE (Monitoring INteligent Environment) [1] así como un prototipo de aplicación móvil que permitía la suscripción y la notificación de eventos en un entorno inteligente equipado con sensores.

Dicho sistema gozaba de múltiples funcionalidades, como consultar el estado de los sensores en tiempo real o el histórico de cambios de estado en el histórico, brindado numerosas oportunidades en un sinfín de ámbitos de aplicación como, por ejemplo, seguridad en el hogar o monitorización de personas mayores.

A pesar del sistema tan ambicioso que se desarrolló, el cual quedó desplegado como middleware del apartamento de inteligencia ambiental del Centro de Estudios Avanzados en Tecnologías de la Información y de la Comunicación de la Universidad de Jaén, el sistema quedó limitado en los siguientes aspectos:

- El sistema web sola permitía la lectura de datos procedentes de dos tipos de sensores. Actualmente, el apartamento de inteligencia ambiental cuenta con múltiples tipos de sensores que deberían ser integrados en el sistema web.

- El sistema web permitía la visualización general del panel administrador para todos los entornos inteligentes, sin hacer uso de roles o perfiles más limitados para cada uno de los entornos que estaba registrados en el sistema.
- El sistema web debía de mejorar en ciertos aspectos de seguridad como el cifrado de cierta información.
- La funcionalidad asociada a la exportación de datos
- La aplicación móvil asociada al sistema web para registrar actividades no permitía el uso de múltiples usuarios.

Por estas razones, este trabajo fin de máster se centra en ampliar las funcionales del sistema web para monitorización de ambientes inteligentes basado en una arquitectura cliente-servidor mediante servicios REST y la realización de una nueva aplicación móvil para el etiquetado de actividades dentro de un ambiente o entorno inteligente equipado con sensores.

Así, esta nueva versión superará las limitaciones de la primera versión e incluirá funcionalidad adicional. A modo de resumen, las principales funcionales a implementar son las siguientes:

- Ampliación de servicios en la API REST, anteriormente desarrollada en el trabajo fin de grado.
 - Adaptar servicios para la gestión de usuarios y roles.
 - Gestionar habitantes en cada uno de los ambientes.
 - Desplegar nuevos servicios para interconectar con la nueva aplicación móvil.
 - Adaptar servicios para la gestión y exportación de actividades.
 - Exportado de actividades.
 - Exportado de flujo de eventos.
 - Exportado de vector de características.

-
- Mostrar lecturas recopiladas por cada una de las balizas de un ambiente inteligente:
 - Estimote Beacon.
 - Estimote Stickers.
 - Adaptar servicios para la lectura de los siguientes sensores.
 - Sensores ambientales (Samsung Smarthing).
 - Sensores de contacto (Samsung Smarthing, Fibaro).
 - Sensores de movimiento (Samsung Smarthing, Fibaro).
 - Adaptar los servicios para la lectura de los siguientes actuadores:
 - Luces Philips HUE.
 - Cerradura electrónica (Schage).
 - Aplicación móvil (Android), para etiquetar actividades en cualquier ambiente inteligente y en tiempo real.
 - Inicio de sesión seguro mediante servicios REST.
 - Información detallada de cada uno de los objetos.
 - Etiquetado de actividades.
 - Suscripción y recepción de notificaciones en tiempo real.
 - Enviar lecturas recopiladas por cada una de las balizas de un ambiente inteligente.
 - Ampliación de la persistencia para el posterior estudio de las actividades y los eventos producidos mientras se realizaban dichas actividades.
 - Se debe rediseñar la estructura de la base de datos para gestionar los cambios propuestos anteriormente. Además, se han de añadir nuevas tablas donde se guardará toda la información relacionada con el comportamiento de las

actividades realizadas por cada uno de los habitantes del sistema.

- Cifrado SHA1 en cualquier contraseña guardada en la base de datos.
- Modificar la imagen corporativa del sistema web y adaptar la nueva aplicación móvil, acorde al espacio de la Universidad de Jaén e Inteligencia Ambiental (UJAml) [2] de aplicaciones del departamento de inteligencia ambiental de la Universidad de Jaén.

Gracias a la interoperabilidad que nos brindan estos elementos, la propuesta de este trabajo fin de máster brinda un mayor abanico de funcionalidades a la vez de superar las limitaciones anteriormente descritas.

1.2 PROPUESTA

El propósito de este trabajo fin de máster está centrado en ampliar la funcionalidad de un sistema Web para la monitorización de ambientes inteligentes a través de la gestión y acceso a servicios, además del desarrollo e implementación de un prototipo de aplicación móvil para el etiquetado y monitorización de actividades dentro del ambiente inteligente de la Universidad de Jaén.

1.3 OBJETIVOS

Los objetivos que derivan de la propuesta de este trabajo fin de máster son los siguientes:

- Definir el conjunto de funcionalidades a ampliar en el sistema web.
- Definir el conjunto de funcionalidades de la aplicación móvil que conecte con el sistema web.
- Diseñar e implementar la funcionalidad del sistema web.
- Diseñar e implementar la funcionalidad de la aplicación móvil.
- Redactar una memoria que recoja todo el trabajo desarrollado, así como los manuales de instalación y usuario.

1.4 ESTRUCTURA DE LA MEMORIA

En este punto se describe la estructura en la que está dividida la memoria de este trabajo:

- **Capítulo 1.** En este primer capítulo, se describe brevemente la idea desarrollada, junto con la motivación que ha llevado a realizar dicho trabajo. Además, se define la propuesta perseguida, junto con cada uno de los objetivos que se marcaron en la definición del trabajo fin de master. Por último, detalla la metodología que se ha seguido para alcanzar cada objetivo, la planificación temporal en la que se justificaran los tiempos establecidos para cada punto y un presupuesto en el que se detallaran los costes de implementación del mismo.
- **Capítulo 2.** En este capítulo, se han descrito los conceptos básicos relacionados con los ambientes inteligentes y una descripción de la primera versión del sistema. Así mismo, ya que las pruebas de este trabajo han sido llevadas a cabo dentro del laboratorio de inteligencia ambiental del Centro de Estudios Avanzados en TIC, se describe detalladamente cómo es este laboratorio y con qué tecnología cuenta en la actualidad.
- **Capítulo 3.** En este tercer punto, se especifican las tecnologías utilizadas, así como los protocolos y estándares de comunicación elegidos para el desarrollo de este trabajo.
- **Capítulo 4.** El capítulo cuarto ha sido utilizado para especificar todo el proceso de Ingeniería del Software llevado a cabo. Dentro de este punto se ha establecido la especificación de requisitos, el análisis de la propuesta, su diseño, la implementación de la misma y por último las pruebas llevadas a cabo para asegurarnos que cumple con cada uno de los objetivos propuestos en el inicio del trabajo.

-
- **Capítulo 5.** Para concluir, en este último punto, se han expuesto las conclusiones que nos ha aportado la realización del presente trabajo, junto con las posibles líneas de trabajo futuras en las que se puede seguir trabajando.
 - **Bibliografía.** En la bibliografía incluiremos todo el catálogo de las publicaciones, capítulos de libros o enlaces web que nos han sido de utilidad para desarrollar este trabajo.
 - **Anexos.** En el anexo A, visualizamos el contenido del disco adjunto a esta memoria. Se describirá con detalle cada carpeta y archivo, junto con una copia de esta memoria. Por otro lado, en los siguientes Anexos (B, C y D) se ha incluido los manuales de instalación, administración y utilización de ambos softwares elaborados en este trabajo fin de máster.

1.5 PLANIFICACIÓN TEMPORAL

A continuación, se indica la estimación de tiempos, donde establecemos el inicio y vencimiento de cada una de las tareas. Finalmente, en el diagrama de Gantt representamos las tareas en función del tiempo y sus predecesoras.

1.5.1 Estimación de tiempos

En la siguiente tabla se puede contemplar las tareas en las que está dividido el trabajo acompañado de la duración (en días) de cada una de estas tareas. Por último, obtenemos como resultado un total de 119 días de trabajo hábiles, los cuales han sido repartidos en 4 horas de trabajo diarias. Obteniendo un total de 476 horas de desarrollo para el trabajo fin de grado.

ID	Nombre	Inicio	Vencimiento	Duración
TFM	TFM	01/12/2017	29/03/2018	119d
AF	Ampliar funcionalidades	04/12/2017	08/02/2018	67d
1	Incrementar la seguridad	04/12/2017	06/12/2017	3d
2	Roles usuarios	07/12/2017	13/12/2017	5d
3	Administración de habitantes	14/12/2017	18/12/2017	2d 4h
4	Ambientes con administradores	18/12/2017	22/12/2017	4d
5	Administración actividades	22/12/2017	28/12/2017	4d
6	Visualización de actividades	28/12/2017	03/01/2018	4d
7	Administración beacons	03/01/2018	05/01/2018	2d 4h
8	Visualización de beacons	08/01/2018	11/01/2018	4d
9	Exportar flujo de eventos	12/01/2018	17/01/2018	4d
10	Exportar vectores de características	18/01/2018	24/01/2018	5d
11	Exportar actividades	25/01/2018	30/01/2018	4d
12	Internacionalización del sistema	04/12/2017	08/02/2018	49d
13	Limpiar diseño administración	31/01/2018	02/02/2018	3d
14	Limpiar diseño principal	05/02/2018	08/02/2018	4d
AP	Aplicación móvil	09/02/2018	20/03/2018	40d
1	Inicio de sesión	09/02/2018	16/02/2018	6d
2	Cerrar sesión	19/02/2018	19/02/2018	2h
3	Consultas ambientes disponibles	19/02/2018	21/02/2018	3d
4	Consultar estado actual	22/02/2018	26/02/2018	3d
5	Lectura NFC	19/02/2018	21/02/2018	3d
6	Escaneado de Beacons	19/02/2018	21/02/2018	3d
7	Escaneado de Stickers	19/02/2018	22/02/2018	4d
8	Etiquetado de actividades	26/02/2018	09/03/2018	10d
9	Etiquetado de actividades en segundo plano	12/03/2018	13/03/2018	2d
10	Recibir notificaciones de eventos	16/03/2018	20/03/2018	3d
11	Suscripciones a eventos	12/03/2018	15/03/2018	4d
D	Documentación	01/12/2017	29/03/2018	119d
1	Redactar documentación del trabajo	01/12/2017	29/03/2018	85d

Tabla 1 Estimación de tiempos

1.5.2 Diagrama de Gantt

A continuación, se expone el tiempo y dedicación establecido en cada una de las diferentes tareas que está dividido el trabajo:

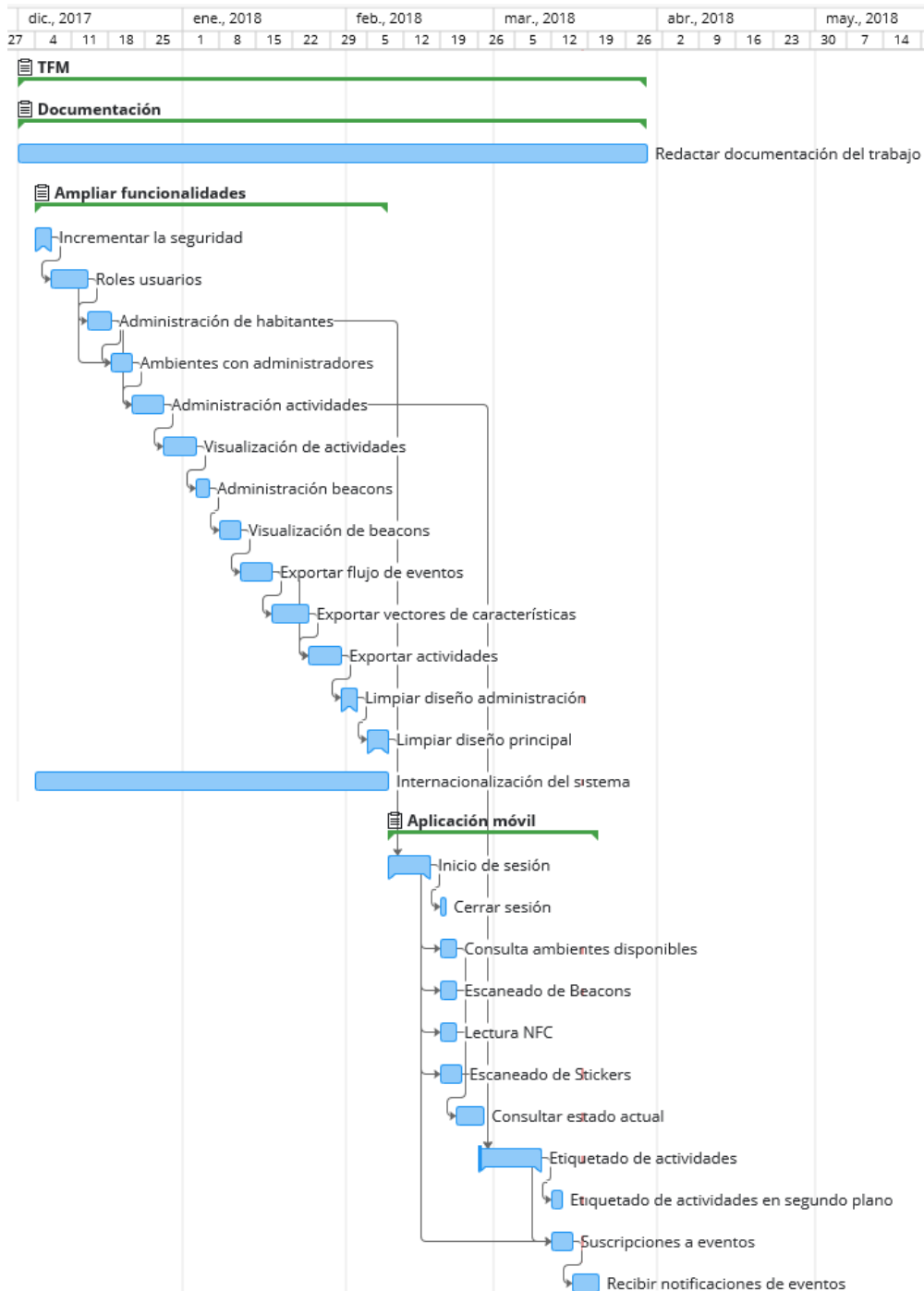


Ilustración 1 Diagrama de Gantt

1.6 METODOLOGÍA SOFTWARE

En este punto, se detallará la metodología de Ingeniería del Software llevada a cabo para el desarrollo de este trabajo fin de máster.

Para ello, se ha utilizado un método iterativo (Scrum) [3], de este modo en cada una de las iteraciones que realicemos, aportaremos un valor añadido al proyecto que puede ser presentado al cliente e ir utilizándose en función del crecimiento. En la figura inferior, podemos ver el ciclo de vida aplicado durante el desarrollo del proyecto.

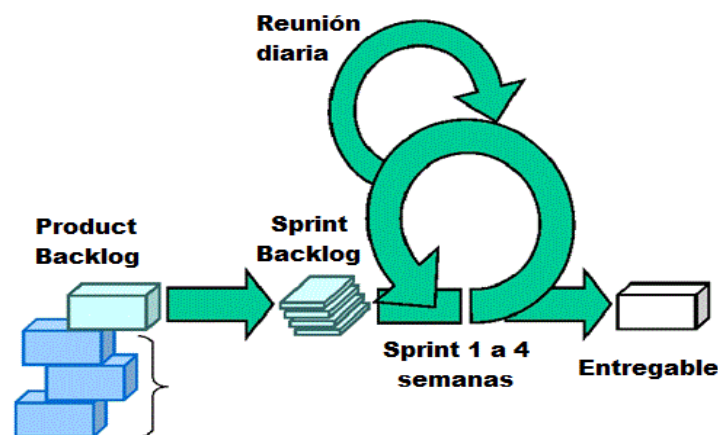


Ilustración 2 Ciclo de vida SCRUM

Uno de los principales problemas que nos encontramos al utilizar Scrum es la imposibilidad de realizar reuniones propuestas por la metodología, ya que el equipo de desarrollo está compuesto por una sola persona. Para solventar esta dificultad, hemos mantenido reuniones en cada iteración con los tutores del trabajo para realizar entregas y valorar el cumplimiento de los objetivos.

1.6.1 Iteraciones

Como hemos detallado en la estimación de tiempos, el desarrollo de este trabajo fin de máster se ha estimado con una duración de cuatro meses. Si estimamos que cada iteración nos va a llevar 2 semanas de tiempo cada una dividiremos el proyecto en un total de ocho iteraciones, estas iteraciones estarán establecidas del siguiente modo.

Tabla de interacciones	
Iteración	Tareas
Iteración 1	AF1, AF2, AF3, AF12, D1
Iteración 2	AF4, AF5, AF12, D1
Iteración 3	AF6, AF7, AF8, AF12, D1
Iteración 4	AF9, AF10, AF11, AF13, D1
Iteración 5	AP1, AP2, AP3, D1
Iteración 6	AP4, AP5, AP6, D1
Iteración 7	AP7, AP8, AP9, D1
Iteración 8	AP10, AP11, D1

Ilustración 3 Tabla de iteraciones

1.7 ESTIMACIÓN DE RECURSOS

Los recursos son los activos o bienes con los que cuenta una organización. Todos los recursos unidos y trabajando entre si hacen posible la realización de un proyecto o servicio. En este trabajo se han dividido los recursos mediante los siguientes dos tipos.

1.7.1 Recursos humanos

Al tratarse de un trabajo fin de master, y estar elaborado por un solo alumno, el recurso humano está limitado a una sola persona. Actualmente la ingeniería informática no cuenta con una regulación que marque unos salarios mínimos, por lo que para este punto nos hemos documentado mediante estadísticas y sueldos medios de la empresa Indeed.

(<https://www.indeed.es/salaries/Informatico/a-Salaries>)

Media salarial ingeniero informático: 18.222€ / Anuales

Concepto	Media salarial	Cuota	Horas	Coste
Ingeniero informático	18,222€/Anual	10,30 € /h	476	4.902,08 €
Cuota SS		31,05 %		1.470,62 €
Total:				6.372,70 €

Tabla 2 Recursos humanos

1.7.2 Recursos técnicos

En este punto, recogeremos cada uno de los elementos físicos y software necesarios para cumplir con cada uno de los objetivos.

En este punto, se diferencian los recursos que ya se contaban con ellos, pero han sido utilizados para desarrollar el trabajo. En este caso realizaremos una amortización lineal del recurso por lo que cuantificaremos el coste de estos recursos dividiendo el precio de su adquisición entre su vida útil entre las horas de trabajo al año. Para ello, supondremos que cada recurso tendrá cero como valor residual y una vida útil de 6 años.

$$\text{Cuota} = ((\text{Precio} / \text{Vida útil})) / (\text{Horas utilizado})$$

Teniendo en cuenta que el equipo informático se utiliza como principal producto de trabajo, calcularemos el número de horas que lo utilizamos cada año.

Para ello, tendremos en cuenta que para el año en el que se realiza este trabajo la jornada laboral es de 1.769 horas anuales de trabajo efectivo, siendo equivalente a cuarenta horas semanales de promedio.

1.7.2.1 Equipo Informático

Se hace uso de un ordenador portátil para el desarrollo, implementación y pruebas. La cuota de uso del portátil se obtiene mediante la siguiente fórmula:

$$\text{Cuota} = ((706,26 / 6)) / 1769 = 0,06 \text{ € / h}$$

Dispositivo móvil para realizar pruebas:

$$\text{Cuota} = ((383,12 / 6)) / 1769 = 0,03 \text{ € / h}$$

Costes variables

Concepto	Adquirido	Cuota	Horas	Coste
DELL Inspiron 15 5000	706,26 €	0,06 € /h	476	28,56 €
Xiaomi Mi6	383,12 €	0,03 €/h	476	11,46 €
Total:				40,02 €

Tabla 3 Equipo informático

Costes fijos

Concepto	Coste
Kit tarjetas NFC	12,16 €
Estimote Beacons	99 €
Estimote Stickers	99 €
Total:	210,16 €

Tabla 4 Costes fijos

1.7.2.2 Servidor dedicado

Para realizar pruebas de desarrollo se ha utilizado un servidor virtual alojado en la empresa OVH. Este servidor tiene las características principales para alojar los servicios, aplicación web y base de datos del proyecto.

Concepto	Coste
VPS OVH SSD	35.88 € /Anual
Dominio .ES	5.43 € /Anual
Total:	41.31 € / Anual

Tabla 5 Costes servidor

1.7.2.3 Licencias software

Para el desarrollo de la aplicación web se ha utilizado un IDE con una suscripción anual. Del mismo modo que con el equipo informático, vamos a calcular la cuota y el coste total en este proyecto.

Concepto	Adquirido	Cuota	Horas	Coste
IntelliJ IDEA Ultimate	149.00 €	0.084 € /h	476	39.99€
Android Studio	0 €	0 €/h	476	0 €
Total:				39.99€

Tabla 6 Costes licencias

1.8 ESTIMACIÓN DE COSTES

En este apartado usamos los datos calculados en los en el anterior punto 2.2. Gracias a esta información podemos estimar el coste total de este trabajo.

Concepto	Precio
Mano de obra	6372,70 € (Sin I.V.A)
Equipo informático	250,18 €
Servidor dedicado	41,31 €
Licencias software	39,99 €
Total:	6662,874 €

Tabla 7 Costes totales

Capítulo 2

2 AMBIENTES INTELIGENTES Y MINE v1

En el siguiente capítulo, se describirán los conceptos básicos relacionados con los ambientes inteligentes y la primera versión del sistema. A continuación, se describirá cómo es el laboratorio de inteligencia ambiental del Centro de Estudios Avanzados en TIC y la tecnología con la que cuenta en la actualidad.

2.1 AMBIENTES INTELIGENTES

En la actualidad, resulta habitual encontrar el término “inteligente” junto a la mayoría de productos convencionales, ya sea un vehículo sobredotado que aparcará por nosotros o un horno que te supera en habilidades culinarias y evita un desastre en esa cena tan importante. En este apartado, sin embargo, exploraremos el verdadero significado de este concepto tan común en estos días, y que, por su innegable importancia en el mundo de las nuevas tecnologías, haremos alusión durante el desarrollo de esta memoria.

La principal pregunta que debemos hacernos es, ¿qué son los ambientes inteligentes? Para responderla, nos vamos a apoyar en la definición que nos proporcionan Alan Steventon y Steve Wright [4]:

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

Extrapolando esta definición a un nivel más técnico, podemos encontrar sistemas electrónicos de diversa índole, de tamaño muy reducido que van a permitir explorar ambientes con una influencia menor en los mismos. Además, debido precisamente a la proliferación de este tipo de elementos y su inclusión en la vida cotidiana, el presupuesto para hacernos con ellos es cada vez más bajo. Por último, encontramos una ventaja indiscutible entre estos dispositivos, y es que tienen la capacidad de comunicarse entre sí, compartiendo información sin que esto afecte a la duración de su batería. Por ello, este tipo de dispositivos son capaces de monitorizar las actividades convencionales que realicemos en la vivienda en la que se

encuentran desplegados y ofrecernos mejores o análisis de las tareas desempeñadas.

Una vez entendida la definición, debemos centrarnos en su aplicabilidad en la sociedad y en las ventajas que nos aporta estudiar estos ambientes inteligentes. Si consultamos cómo se encuentra la pirámide de población en nuestro país, podemos observar que la esperanza de vida es mucho más alta, mientras que nos encontramos con un notable decremento de la natalidad [5].

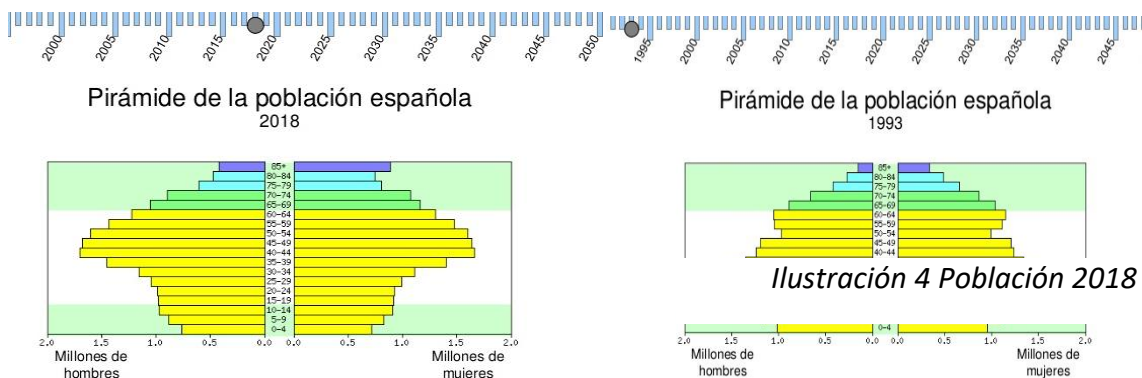
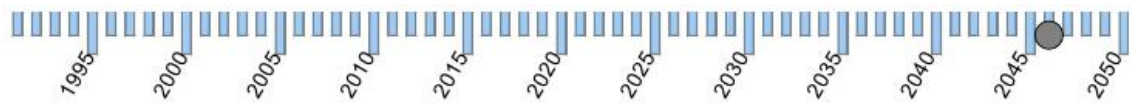


Ilustración 4 Población 2018

En adición, una estimación de la población de nuestro país en el año 2056 indica que la mayor parte de la sociedad estará compuesta por personas mayores.



Pirámide de la población española 56

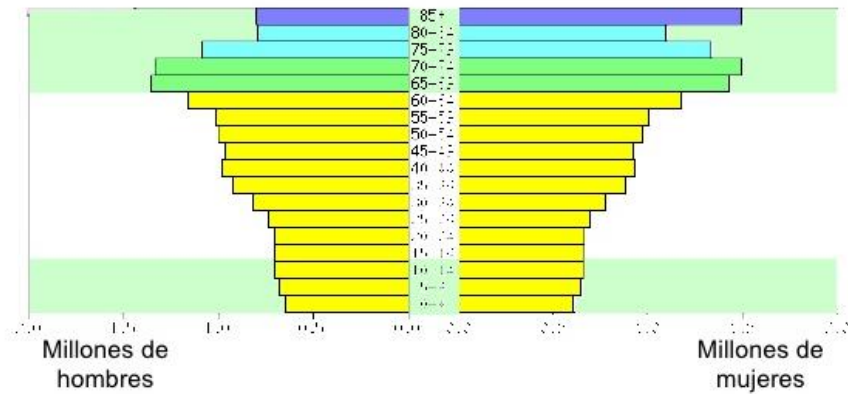


Ilustración 6 Población 2056

Aquí, los ambientes inteligentes juegan un papel muy importante, ya que una de sus principales funcionalidades es hacer frente al envejecimiento general de la población.

Uno de los principales motivos del enfoque de este trabajo fin de máster se ha establecido en el aumento de la funcionalidad de la primera versión del sistema de monitorización, debido a sus carencias.

2.1.1 Aplicaciones de los ambientes inteligentes

A continuación, se indican algunas de las aplicaciones más relevantes que pueden ser desplegadas en los ambientes inteligentes.

- **Asistencia telemática:** La teleasistencia es definida como el conjunto de servicios prestados a través de dispositivos telemáticos [6]. En la medicina, al ofrecer cada uno de estos servicios se mejora la calidad de vida de las personas con algún tipo de dependencia.
- **Seguridad:** En muchas ocasiones, es muy fácil comprar los ambientes inteligentes con los sistemas de domótica, estos sistemas siempre nos han brindado un extra de seguridad en un entorno o vivienda convencional. De este modo, una de las posibles aplicaciones que obtenemos dentro de un ambiente inteligente es aumentar la seguridad. En este ámbito, el ambiente puede detectar automáticamente algunas situaciones e interactuar por sí sólo, a diferencia de los sistemas domóticos, los ambientes inteligentes, son capaces de realizar simulaciones de personas residiendo en una vivienda o de animales ladrado. De este modo cualquier individuo que desee romper la seguridad de la vivienda puede ser disuadido al no saber si se encuentra gente en ella.
- **Seguimiento de actividades:** Como ya hemos podido apreciar en este documento, uno de los puntos más importantes de los ambientes inteligentes es introducir una mejora transparente en las actividades que realiza habitante. Para ello, una aplicación muy importante en este ámbito es el análisis del seguimiento de actividades, esta aplicación, nos facilitará el tratamiento de muchas enfermedades o demencias. Normalmente los dispositivos más utilizados suelen ser balizas (beacons) que detectan la situación del

habitante, etiquetas NFC, pulseras inteligentes y sensores de interrupción.

- **Escenas** En los entornos de inteligencia ambiental también se pueden configurar escenas donde diferentes actuadores se coordinan para facilitar a los habitantes las tareas que realizan. Estas aplicaciones se utilizan con individuos con problemas de sueño o concentración. Tan solo debemos de contar con dispositivos sonoros y luminosos. Estos dispositivos crearán un ambiente específico dependiendo de la actividad que realice, modificando los atributos de la luminosidad y reproduciendo sonidos en cada una de las estancias.
- **Eficiencia Energética**, Por último, se encontrará con el área de aplicación la cual gestionará todo el sistema, dependiendo de la temperatura, luminosidad y valores, tanto externos como internos. De este modo, no hará falta tener una bombilla encendida si no hay nadie en una habitación, o ajustar la temperatura de la calefacción dependiendo de la temperatura exterior y a las preferencias de los usuarios que se encuentran en las estancias.

2.1.2 Laboratorio Inteligente (Universidad de Jaén)



En Centro de Estudios Avanzados en Tecnologías de la Información y la Comunicación (CEATIC) de la Universidad de Jaén posee uno de los primeros laboratorios de inteligencia ambiental en España [7]. Este laboratorio, denominado UJAmI Smartlab, queda englobado en el ecosistema ¹UJAmI que, a su vez, hace referencia la fusión del término AmI (Ambient Intelligence) [2] con la Universidad de Jaén. UJAmI Smartlab se centra principalmente en la atención socio sanitaria a personas mayores o con algún tipo de diversidad funcional.



Ilustración 7 Laboratorio Universidad de Jaén

El laboratorio, se encuentra en la dependencia 109 del edificio C6 del Campus de Las Lagunillas de la Universidad de Jaén. Este laboratorio, está equipado con todos los electrodomésticos que podemos encontrar en una vivienda convencional, facilitando de este modo a los habitantes que realicen actividades dentro de este laboratorio. Además de estos electrodomésticos, UJAmI Smartlab cuenta con sensores ambientales, actuadores, localización de interiores, visión a través de cámaras,

¹ <http://ceatic.ujaen.es/ujami>

dispositivos de salud, interfaces cerebrales, interfaces humano-máquina, robots y dispositivos multimedia.

Un aspecto diferenciador de este ambiente inteligente es que cuenta con el asesoramiento de la fundación Ageig Lab². Esta fundación con sede en la capital Jienense nutre al centro de las necesidades reales de las personas en las que está enfocado dicho laboratorio.

2.2 MIME V1

A lo largo del curso 2015-2016, se desarrolló un sistema de monitorización de ambientes inteligentes junto a un prototipo de aplicación móvil para la suscripción de eventos en un ambiente, como trabajo de fin de grado. Dicho sistema, pretendía proporcionar a las personas con dificultades para realizar tareas cotidianas, una oportunidad para ganar independencia, como puede ser el caso de personas que hayan visto mermadas algunas de sus capacidades, ya sean físicas o psíquicas, y garantizar la tranquilidad a su entorno más cercano, como pueden ser cuidadores o familiares.

Por ello, los objetivos que se propusieron, y finalmente cumplieron, para dicho sistema son los siguientes:

- Definir el conjunto de servicios necesarios para desarrollar el sistema de monitorización de entornos de inteligencia ambiental, también denominados ambientes inteligentes.
- Desarrollar los servicios Web asociados, así como la definición e implementación de la base de datos para el almacenamiento de la información necesaria.
- Desarrollar un prototipo de aplicación web que permita la monitorización de ambientes inteligentes.

² <http://www.ageinglab.com>

- Desarrollar un prototipo de aplicación móvil para la notificación de cambios en el ambiente inteligente.

Sin embargo, pese a que dicho sistema gozaba de numerosas oportunidades para mejorar la calidad de vida de las personas, también constaba de algunas limitaciones como se ha explicado en apartados anteriores.

Una de las limitaciones que podrían resultar más llamativa, es que el Mime v1, pese a ser escalable, tan sólo trabajaba con dos tipos de sensores, lo cual reducía la cantidad y la calidad de la información útil recogida por los usuarios. Concretamente, se utilizaron, por un lado, sensores Sunspot, unos sensores ambientales de Oracle capaces de detectar la temperatura, luminosidad y movimiento y, por otro lado, los sensores Tynetec, de contacto, utilizados para detectar la manipulación en cualquier punto en el que se encuentren instalados.

2.2.1 Reconocimientos

El proyecto Mime v1, ha recibido el reconocimiento de la comunidad científica en varias ocasiones, a través de artículos académicos, congresos, medios de comunicación y, también, siendo galardonado con el Premio Ada Lovelace 2016 a la Mejor Aplicación Software.

A Web System for Managing and Monitoring Smart Environments. In *International Conference on Bioinformatics and Biomedical Engineering* (Zafra, D., Medina, J., Martinez, L., Nugent, C., & Espinilla, M.).

Capítulo 3

3 ESTADO DEL ARTE EN AMBIENTES INTELIGENTES

El tercer punto de este trabajo, se centrará en explicar las tecnologías que se han utilizado y, en los protocolos y estándares de comunicación que se han desarrollado en este trabajo.

3.1 MIDDLEWARES

Un middleware es un conjunto de componentes software, este conjunto es el encargado de integrar múltiples aplicaciones o proceso en un mismo ámbito. De esta forma, cada una de estas integraciones pueden trabajar de manera concurrente y compartir datos entre sí.

En los ambientes inteligentes el middleware es la pieza fundamental, ya que debe trabajar con múltiples sensores de diferentes tipos, múltiples métodos de comunicación y ofrecer servicios a distintos clientes.



Ilustración 10 Middelware

Bloques

Todo middleware encargado de trabajar en un ambiente inteligente comparte los siguientes bloques:

- **Gestión de dispositivos:** Detecta e informa del estado de cada uno de los dispositivos que se encuentran conectados en un ambiente inteligente.

- **Conectividad:** Intercambia datos entre dispositivos (sensores o clientes) de manera transparente para el usuario. Algunas de las comunicaciones más comunes son WIFI, Z-Wave, Zigbee...
- **Persistencia:** Capacidad de almacenar todos los cambios que se producen en el sistema.
- Procesamiento de datos
- **Análisis de datos:** Agrupamiento de datos, y aprendizaje de forma autónoma y transparente.
- **Visualización de datos:** Permite a los usuarios observar todo lo que ocurre en el sistema mediante un cuadro de mando.

Una vez comprendido el concepto, se puede observar que dentro de los ambientes inteligentes es una de las partes más importantes a tener en cuenta. Ya que una buena definición desde la base, nos ahorrará mucho tiempo a la hora de manejar los datos que se están produciendo de forma manual o automática en un ambiente.

3.2 COMUNICACIÓN

Como se ha mencionado en el punto anterior, una de las partes más importantes de los middlewares es la comunicación. Gracias a la comunicación el sistema puede interconectar a usuarios, sensores, actuadores o incluso ordenadores convencionales. Es por esto que dentro de un ambiente inteligente nos podemos múltiples alternativas para intercambiar datos. En esta memoria vamos a diferenciar el intercambio entre aplicaciones software (Protocolos de comunicación) y hardware (estándares de comunicación).

Protocolos de comunicación

La definición proporcionada por el World Wide Web Consortium (W3C) sobre los protocolos de comunicación en los servicios web, es la siguiente.

“Son un sistema software diseñado para soportar una comunicación entre diferentes equipos en red”

Los protocolos más importantes dentro de un ambiente inteligente son los siguientes:

- **MQTT**

Es un protocolo orientado a la comunicación entre dispositivos con recursos limitados. Este protocolo se basa en una arquitectura en estrella. Para ello, su principal funcionamiento reside en un nodo central llamado bróker, el cual por si solo tiene una capacidad de hasta 10.000 clientes simultáneos.

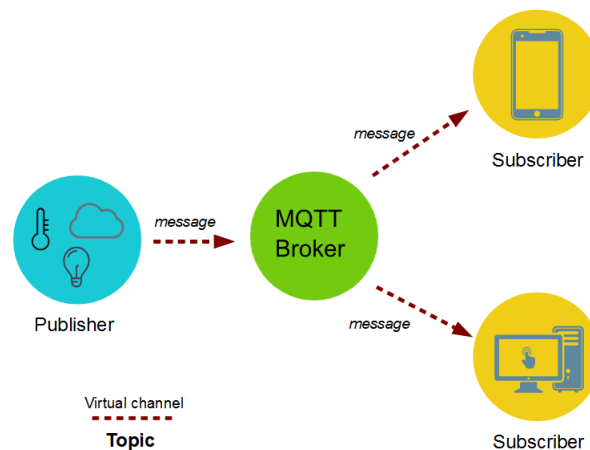


Ilustración 11 Esquema MQTT

Como se puede visualizar en la figura superior, el bróker es el encargado de recibir y transmitir los mensajes a través de la red. Para esto se basa en el uso de mensaje a través de un topic (temas) en el que cada cliente publica la información.

Posteriormente, por cada uno de los topics, pueden suscribirse tantos clientes como se deseen, de este modo en el momento de la publicación recibirán el mensaje publicado en tiempo real.

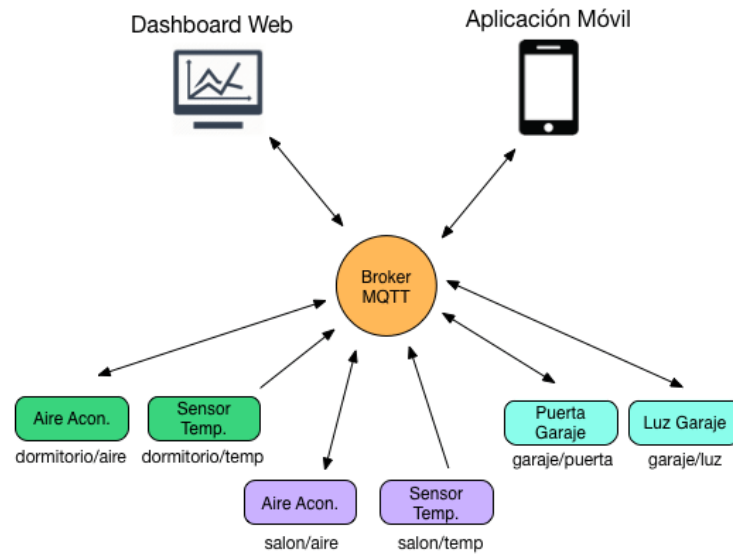


Ilustración 12 Ejemplo temas MQTT

Del modo que se puede visualizar en la ilustración superior, una aplicación móvil o un panel web puede consumir cada evento según la suscripción realizada. Estos topics o temas son los diferentes canales en los que se publican los datos y a los que los diferentes receptores deben suscribirse. Teniendo cada uno de ellos una estructura tipo url donde se van generando niveles con barras.

A modo de ejemplo unos topic válidos podría ser casa/salon/humedad o casa/salon/temperatura.

Lo que hace realmente interesante a MQTT es la facilidad a suscribirnos a todos los eventos utilizando comodines. Por ejemplo, /casa/salón/# o incluso /casa/#, obteniendo de este modo todos los eventos producidos dentro de la vivienda.

- **SOaP**

Es uno de los protocolos más utilizados en los servicios web, este protocolo se encarga del intercambio de objetos y mensajes entre distintas aplicaciones haciendo so del paradigma de mensajería en una dirección sin estado. Su funcionamiento reside en un servidor, este es el encargado de realizar las consultas pertinentes a una fuente de datos y devolverla al cliente encargado de realizar la petición.

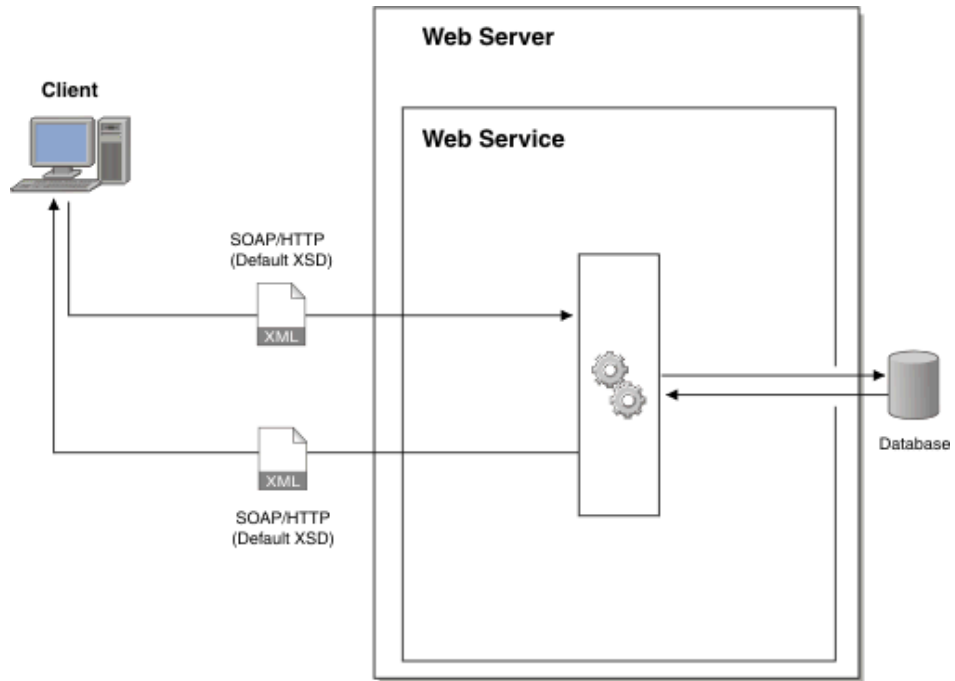


Ilustración 13 Esquema SOAP

Cada mensaje enviado utilizando este protocolo, está dividido en las siguientes zonas:

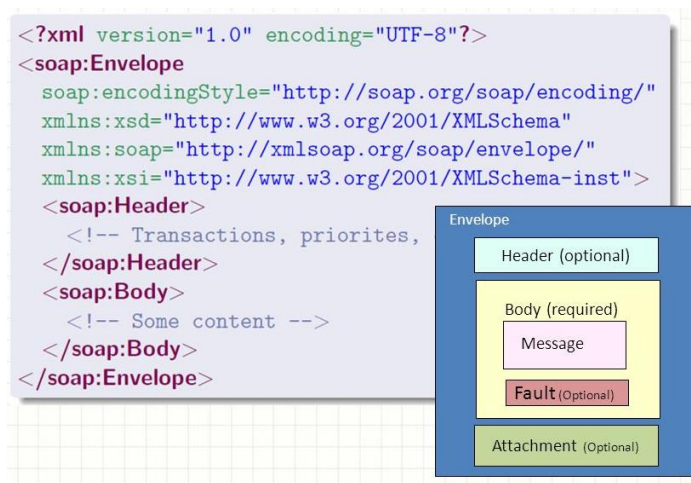


Ilustración 14 Zonas SOAP

- **Envelope** o más conocido como “sobre”, es el encargado de definir que es el contenido que se encuentra en el mensaje y como deberá ser procesado.
- **Header**, nos permite enviar información relativa a cómo debe ser procesado el mensaje.
- **Body**, contiene toda la información relativa a la petición que ha realizado el cliente.
- **Fault**, es el encargado de mostrar los errores que se ha producido en el mensaje.

- **HTTP REST**

REST de las siglas en inglés (REpresentational State Transfer), es un paradigma de intercambio de datos basado en el protocolo HTTP.



Ilustración 15 Esquema REST

De nuevo nos encontramos con un protocolo cliente/servidor que nos permite el intercambio de mensajes/datos sin estado. A diferencia de SOaP, normalmente los mensajes van encapsulados mediante texto plano JSON, aunque del mismo modo REST permite trabajar con documentos XML.

Las operaciones que nos permite realizar REST vienen definidas por el protocolo HTTP:

- POST: Crear contenido
- GET: Consultar contenido
- PUT: Editar contenido
- DELETE: Borrar contenido

Comparativa de uso

Para realizar esta comparativa, nos hemos basado en el interés de los usuarios por dichos protocolos, para ello, utilizaremos el servicio gratuito Google Trends.

En el momento de llevar a cabo esta comparativa, se ha utilizado una ventana temporal de un año, como resultado, se puede inferir que el protocolo más utilizado dentro del ámbito de este trabajo fin de master es MQTT.



Ilustración 16 Interés por protocolos

Estándares de comunicación

Dentro del ámbito europeo, se pueden destacar cuatro estándares de comunicación, entre los dispositivos que se encuentran ubicados en un ambiente inteligente.

- ZWave
- ZigBee
- Wifi
- Bluetooth

Comparativa de características

	Wifi	Bluetooth	ZigBee	Z-Wave
Frecuencia	2.4Ghz	2.4Ghz	2.4Ghz	868Mhz
Tamaño de pila	~1Mb	~1Mb	~20Kb	~20Kb
Transferencia	11Mbps	1Mbps	250Kbps	9.6/ 40 Kbit/s
Canales	11-14	79	16 (2.4Ghz)	1
Distancia	100m	100m	100m	100m
Dispositivos	32	8	255/65535	232
Arquitecturas	Estrella	Estrella	Estrella/Malla	Malla

Tabla 8 Comparativa estándares

Comparativa de uso

En el momento de llevar a cabo esta comparativa, se ha utilizado una ventana temporal de un año, como resultado, se puede inferir que el estándar de comunicación más utilizado dentro del ámbito de este trabajo fin de master es Bluetooth.

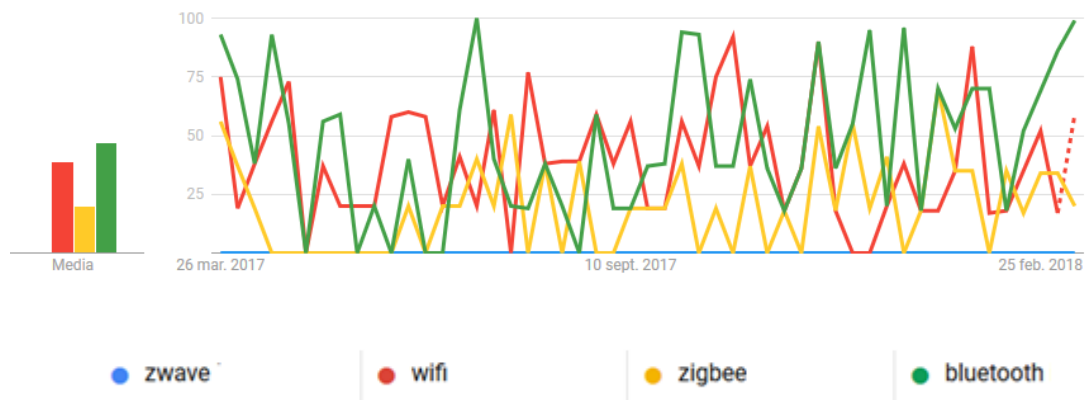


Ilustración 17 Interés por estándares

3.3 ETIQUETADO DE ACTIVIDADES

El etiquetado de actividades consiste en limitar los cambios de los sensores que se producen al realizar determinadas actividades. Los datos recabados servirán para crear dataset y entrenar modelos para realizar reconocimiento de actividades, detección de anomalías, etc. La aplicación móvil desarrollada en este trabajo fin de máster tiene como funcionalidad etiquetar y enviar actividades al middleware. También se ha implementado la funcionalidad para que, desde el middleware se puedan exportar estos datos y así dar paso a la creación del dataset.

3.3.1 Tecnología NFC

Near Field Communication (NFC) es una tecnología de comunicación por radiofrecuencia de corto alcance. Se encarga de transmitir ondas en la banda 13.56Mhz con velocidades comprendidas entre los 106 Kbits a 425Kbits [8]. Al tratarse de una frecuencia de corto alcance, el rango de trabajo se encuentra entre los cinco centímetros de longitud.

NFC se comunica mediante inducción dentro de un campo magnético creado por dos antenas en forma de espiral, las cuales permiten leer o modificar el contenido de cada tarjeta.

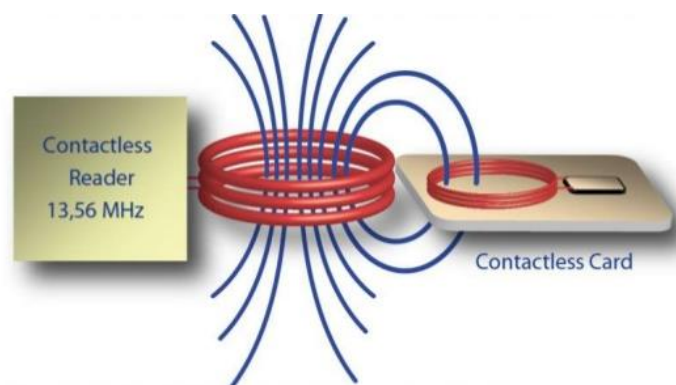


Ilustración 19 Campo magnético NFC

Dependiendo del uso para el que esté pensando, permite trabajar en dos modos distintos.

- **Activo:** Ambos dispositivos crean el campo electromagnético para intercambiar información.
- **Pasivo:** Solo un dispositivo es el encargado de crear el campo electromagnético, por lo que este campo será suficiente para intercambiar la información entre ambos puntos. Normalmente este es el modo más utilizado en dispositivos móviles.

En la definición de NFC, hemos hablado de etiquetas pasivas, tarjetas contact less y dispositivos. Antes de adentrarnos en el uso de NFC, nos queda definir que son las etiquetas NFC o tag, siendo estas un dispositivo pasivo encargado de modificar el campo magnético enviado por un elemento activo y transmitir de este modo información.

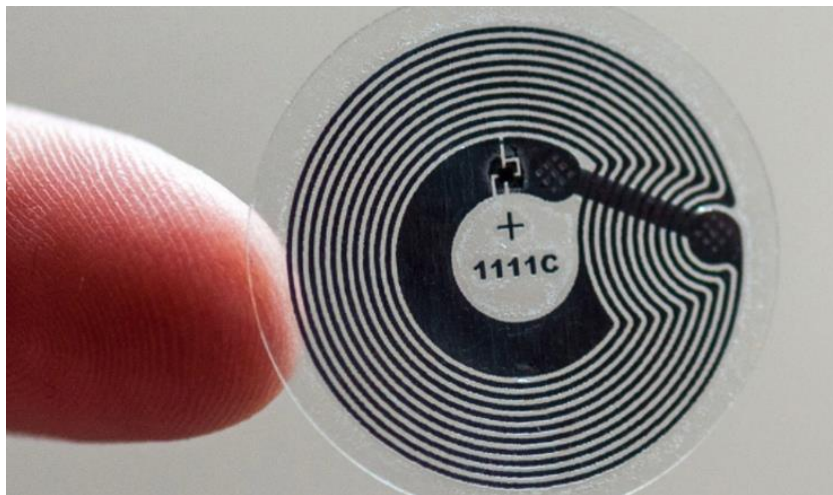


Ilustración 20 Etiqueta NFC

Estas etiquetas contienen una capacidad máxima de memoria, comprendida entre los 64 bytes a los 924, los cuales nos permiten almacenar una longitud entre 40 y 900 caracteres de texto por etiqueta.

En la actualidad la tecnología NFC está en auge, y la mayoría de los dispositivos móviles ya cuentan con ella. El abanico de posibilidades es cada vez más amplio, destacando los siguientes posibles casos de uso:

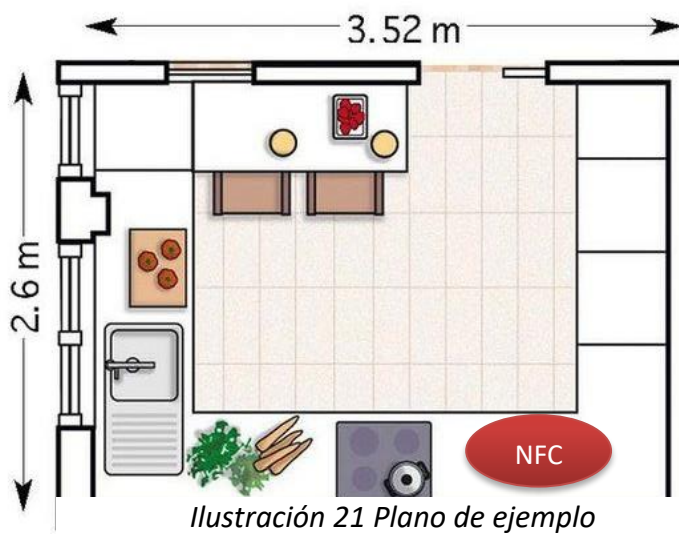
- Pago en transporte publico
- Pago de pequeñas compras si necesidad de introducir PIN
- Pago mediante dispositivos móviles
- Abrir cerraduras electrónicas
- Control de presencia
- Video manuales
- Conectarnos a redes WIFI sin necesidad de mostrar la contraseña
- Configurar parámetros de dispositivos móviles
- Realizar llamadas en dispositivos móviles
- Llevarnos a URLs personalizadas

NFC nos proporciona una de las mejores soluciones para suplir el objetivo de este trabajo fin de máster, ya que, al tratarse de una tecnología de corto alcance, obligan al usuario a estar en el lugar donde se encuentra realizando la actividad.

Dicho de otro modo, se podría realizar una aplicación móvil simple donde el usuario seleccionara que actividad va a realizar antes de comenzarla, pero no tendríamos forma de comprobar si el usuario se encuentra en el sitio adecuado donde verdaderamente realizará la actividad, produciendo falsos eventos de sensores y errores en el posterior análisis.

Esta desventaja la suple con creces la tecnología NFC, tan solo se deberá de hacer un estudio de dónde colocar las etiquetas para que el usuario pueda notificar el comienzo de una actividad.

Ejemplo (Actividad: Cocinar en vitrocerámica)



Tomando como ejemplo el plano de la imagen izquierda, un buen lugar para colocar la etiqueta NFC que nos indique que está cocinando haciendo uso de la vitrocerámica puede ser la marcada con la imagen roja. Ya que todas las acciones recopiladas por todos los

sensores instalados en la cocina del ambiente inteligente, estarán asociadas a la propia actividad.

3.3.2 Balizas posicionamiento

Las balizas, también conocidas como beacon son dispositivos electrónicos de muy bajo consumo. Estas balizas, emiten una señal inalámbrica en la banda de trabajo Bluetooth de bajo consumo (BLE) con una difusión amplia sin destino concreto.

Bluetooth (BLE) es una evolución del estándar Bluetooth [9] la cual consigue reducir notablemente el consumo de la emisión y por ende el aumento de la vida útil de las baterías gracias a minimizar la potencia en la transmisión de la señal de radio. Gracias a este bajo consumo de energía la tecnología BLE es ampliamente utilizada en dispositivos wearables.

Gracias a la señal emitida por cada una de estas balizas, podemos detectar a que distancia aproximada [10] nos encontramos de ellas mediante un simple escaneo de señal bluetooth.

De este modo el sistema desarrollado se nutre totalmente de esta tecnología para así posicionar a de los habitantes que se encuentran realizando una actividad utilizando la aplicación móvil desarrollada.

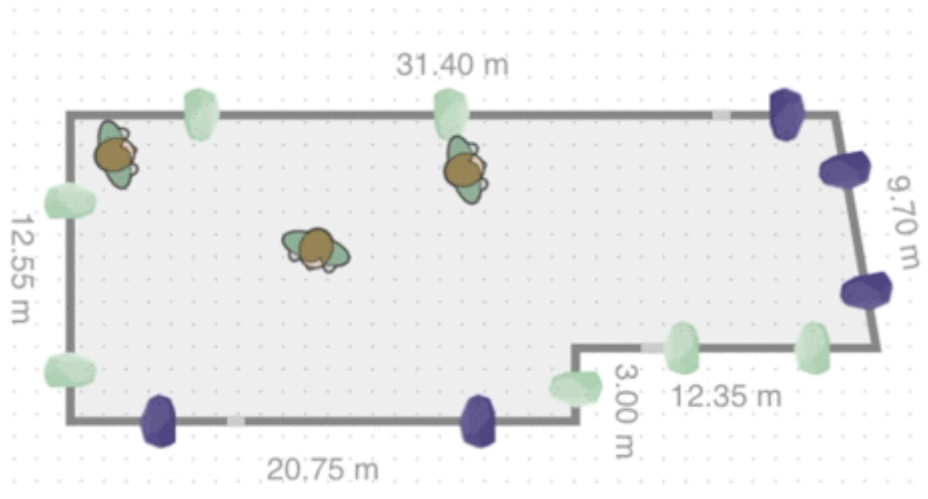


Ilustración 22 Ejemplo de posicionamiento mediante beacons

Como se puede visualizar en la ilustración superior, si colocamos varias balizas en un entorno o vivienda inteligente se puede triangular y por ende posicionar a cualquier persona u objeto dentro de un entorno cerrado.

Gracias al posicionamiento proporcionado por estas balizas, podemos asociar los eventos de los sensores a la persona que los está provocando, solventando de este modo falsas lecturas dentro del etiquetado de actividades.

Nearables

Del mismo modo que hemos hablado en el punto superior, las balizas aportaban un gran abanico de posibilidades en localizaciones estáticas, como museos, laboratorios, aeropuertos etc. Las balizas entienden este concepto a lugares mucho más pequeños como pueden ser los objetos que se encuentran dentro de un entorno o vivienda.



Ilustración 23 Baliza nearables

Por otro lado, estas balizas además de radiar BLE, son capaces de monitorizar movimiento y temperatura, con un tamaño de tan solo 3mm de ancho. Esto hace que puedan ser distribuidas en casi cualquier objeto de una vivienda convencional.

Por otro lado, la empresa Estimote, la cual comercializa estas balizas, nos brinda una API gratuita con la que además de obtener el RSSI de cada baliza, también podemos obtener una estimación de la proximidad. De este modo, el usuario puede saber si se encuentra cerca o lejos de la baliza.



Ilustración 24 Estimación de proximidad

Esta tecnología casa enormemente con los objetivos del trabajo, por lo que además de no tener que llevar un dispositivo extra para posicionar a los habitantes hace que el sistema pueda ser escalable en un futuro a dispositivos más pequeños como relojes o pulseras deportivas.

De manera introductoria, ya que se podrá ver todo el funcionamiento en detalle dentro del capítulo 4 de esta memoria, como podemos apreciar en la ilustración inferior, mientras un habitante se encuentra realizando una actividad etiquetada con el dispositivo móvil, este enviará de forma transparente cada una de las potencias recibidas por cada baliza.



Ilustración 25 Beacons Balizas

En el caso de las balizas beacon, pueden ser vitales para detectar en que área de la vivienda se encuentra, cocina, salón, dormitorio, mientras que con las balizas nearable, identificaremos que objeto está siendo manipulado, puerta, televisión, microondas etc.

3.3.2.1 Aplicaciones

A continuación, mostraremos las aplicaciones donde reside la tecnología beacon en la actualidad. Abriendo un gran abanico entre las siguientes posibilidades.

- **Sector turístico**

En la actualidad, el 70% de los viajeros, visita nuevos destinos sin separarse de su teléfono móvil, gracias a esto, los turistas pueden recibir información en tiempo real de todo lo que encuentran a su alrededor, desde información o publicidad personalizada dentro de un aeropuerto hasta ciudades personalizadas donde van guiando al turista entre los puntos de interés más característicos. Un gran ejemplo de esta última funcionalidad es DTJax, la cual ofrece información sobre tiendas y restaurantes mientras el viajero se va moviendo por la ciudad. Para que esto sea posible la empresa ha instalado una serie de balizas en puntos céntricos.

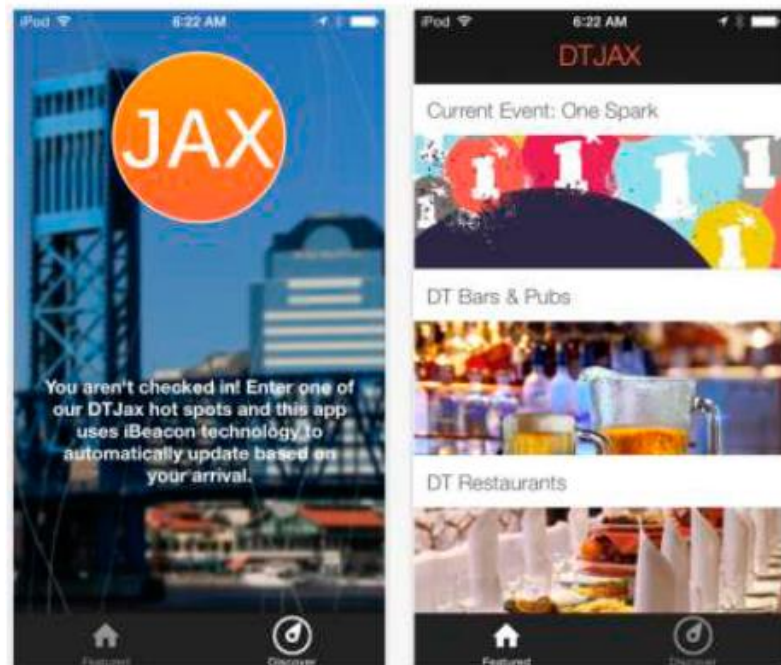


Ilustración 26 APP DTTJax

- **Museos**

Una de las grandes innovaciones de los museos es poder mostrar al visitante información guiada tanto en idioma como geográficamente. Mediante el sistema beacons, ofrece a los visitantes una oportunidad para añadir visitas guiadas e interactivas a sus propios teléfonos móviles, ahorrando grandes cantidades de dinero en guías y material electrónico.

Uno de los primeros museos en incorporar este sistema ha sido el museo nacional de Gales, el cual permite a los viajeros conocer cada una de las colecciones en diferentes idiomas y en el momento preciso.

- **Centros comerciales**

Grandes centros comerciales también están incorporando esta tecnología para enviar publicidad personalizada a sus clientes. De este modo, mientras los clientes caminan por el centro comercial, se pueden mostrar en sus teléfonos móviles alertas personalizadas de los comercios más próximos.

Llevando a un extremo un poco más amplio, nos encontramos con comercios los cuales muestran información precisa del producto que tienes en ese momento en las manos.

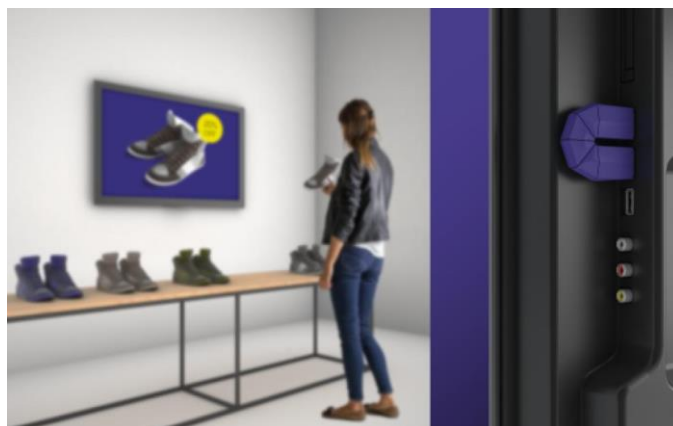


Ilustración 27 Estimote Mirror

- **Grandes eventos**

Por último, otra de las posibilidades donde más se utiliza esta tecnología es en grandes eventos. Donde los gestores de estos eventos pueden facilitar los accesos y los puntos de interés más importantes que pueden encontrar los asistentes. Algunos de los eventos donde se utiliza esta tecnología en la actualidad son:

- Festival de cine de Cannes
- Chicago AutoShow
- Campo de fútbol del FC Barcelona [11]

El Club las está instalando en la zona del Bulevar del Camp Nou, y de esta manera los visitantes que tengan la aplicación oficial del Barça en el móvil recibirán promociones para los restaurantes del bulevar y la explanada. Además, podrán comprar desde el móvil la entrada para el Tour Camp Nou Experience.

3.3.2.2 Otras tecnologías

Uno de los grandes problemas en el etiquetado de actividades multiocupacional es detectar que habitante se encuentra realizando una actividad dentro de un entorno inteligente. Para ello, una funcionalidad añadida a este trabajo fin de máster es guardar la información a la que se encuentran los habitantes de varias marcas repartidas por el entorno inteligente. Gracias a esta información, posteriormente en trabajos futuros se puede triangular y obtener de este modo que sensores estaba manipulando cada uno de los habitantes.

Antes de elegir la tecnología beacon se estudiaron y rechazaron las siguientes tecnologías.

- **Tecnología WIFI**

La tecnología WIFI funciona de manera inalámbrica donde un emisor radia una señal para ser utilizada en modo de conexión en redes locales. Gracias a esta señal, podemos saber a qué distancia aproximada nos encontramos del emisor y por ende conocer de forma aproximada la localización del cliente.

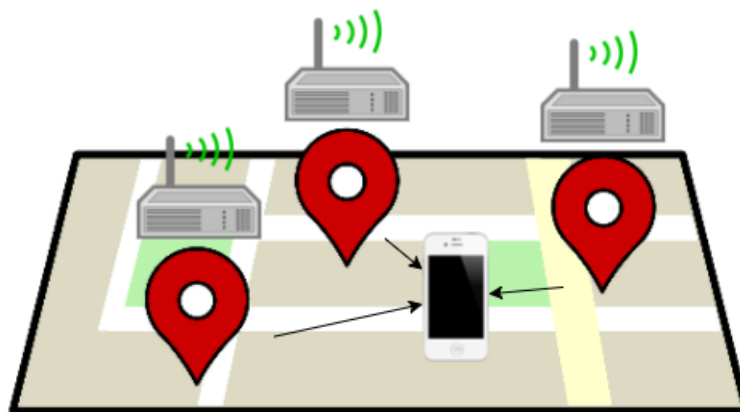


Ilustración 28 Posicionamiento WIFI

Este sistema, cuenta con la ventaja de no necesitar ninguna instalación extra de dispositivos y añade al usuario la funcionalidad de navegar de manera gratuita dentro de la red de internet.

Finalmente pese a esta ventaja, se decidió no utilizar esta tecnología en este trabajo fin de máster por su alto consumo eléctrico comparado con BLE y su poca tolerancia a ruidos entre canales similares.

- **Tecnología GSM/3G/4G**

Cada uno de los teléfonos móviles que utilizamos en la actualidad emite continuamente una serie de mensajes en modo broadcast a las antenas más cercanas, las antenas responden y este se conecta a la más cercana y la que más cobertura ofrece.



Ilustración 29 Triangulación GSM

Gracias a esto, podemos obtener de una forma aproximada en qué punto se encuentra el usuario, se descartó sin duda esta tecnología al no ser nada precisa en interiores.

- **Tecnología RFID**

Por último, se estudió la posibilidad de utilizar etiquetas RFID y un lector UHF para obtener información de cada una de las etiquetas que tengamos próximas. Dentro de interiores es uno de los sistemas más precisos, pero el uso de un dispositivo extra hace que no sea posible utilizar esta tecnología en este trabajo.

Capítulo 4

4 INGENIERÍA DE SOFTWARE

En este capítulo especificaremos el comportamiento y las diferentes funcionalidades del sistema. Además, se representará toda la arquitectura mediante diagramas, estos diagramas están modelados mediante UML [12], ya que es uno de los lenguajes de modelado más utilizados en los sistemas softwares.

4.1 DEFINICIÓN DE REQUISITOS

Los requisitos son todas las condiciones impuestas por el cliente que le usuario o desarrollador debe abarcar para llevar a cabo cada objetivo [13]. Dentro de este trabajo se han diferenciado dos tipos de requisitos:

4.1.1.1 Requisitos funcionales

Estos requisitos se encargan de definir el comportamiento del sistema. Para una representación sencilla y clara, vamos a utilizar una estructura en forma de árbol. De este modo, nos encontraremos con los siguientes niveles:

- **Nivel 1:** Requisitos de cliente. (Incrementan el valor del proyecto)
- **Nivel 2:** Representan funcionalidades
- **Nivel 3:** Representan subfuncionalidades
- **Nivel 4:** Características

Para representar estos requisitos de una forma más sencilla, se han dividido dos partes, requisitos para aumentar la funcionalidad del sistema y requisitos de la aplicación móvil.

Ampliar funcionalidades

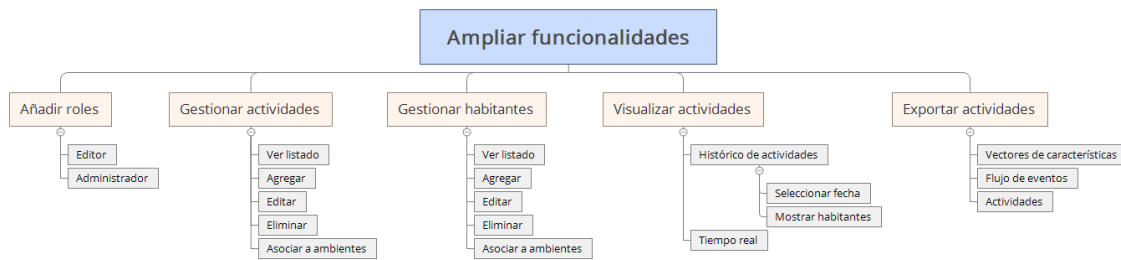


Ilustración 30 Requisitos funcionales (Ampliar funcionalidades)

- **Requisito funcional 1:** Añadir roles

El sistema debe proporcionar la posibilidad de añadir roles a los usuarios, de este modo, serán los encargados de gestionar ambientes o simplemente editar el ambiente que tienen asociado a su cuenta.

- **Requisito funcional 2:** Gestionar actividades

El sistema debe permitir añadir, editar, visualizar o borrar las actividades dentro de un ambiente determinado. Estas actividades tan solo estarán asociadas a un solo ambiente.

- **Requisito funcional 3:** Gestionar habitantes

El sistema debe permitir gestionar habitantes (dar de alta, editar o eliminar). Cada uno de estos habitantes estarán asociados a un ambiente determinado y sus datos serán utilizados como credenciales para identificarse en la aplicación móvil.

- **Requisito funcional 4:** Visualizar actividades

El sistema permitirá a los usuarios consultar las actividades que se están realizando en el momento de la consulta o en una fecha determinada. A su vez, el sistema mostrará información de los habitantes que se encuentran realizando la actividad e información

de posicionamiento mediante lecturas recopiladas por cada una de las balizas desplegadas.

- **Requisito funcional 5:** Exportar actividades

El sistema permitirá exportar las actividades que se han realizado en un ambiente para su posterior estudio o generación de artículos. El exporto dado deberá ser parametrizado por el usuario y contará con tres tipos de exportados distintos.

Como requisito funcional englobando todos los puntos anteriores, el sistema deberá de proporcionar una base de datos modificada para realizar la persistencia de todos los datos necesarios para el correcto funcionamiento del mismo. Además, tanto la aplicación web como la aplicación móvil deberán cumplir con la nueva imagen corporativa de la Universidad de Jaén y el laboratorio de inteligencia ambiental. (UJAMI).

Aplicación móvil

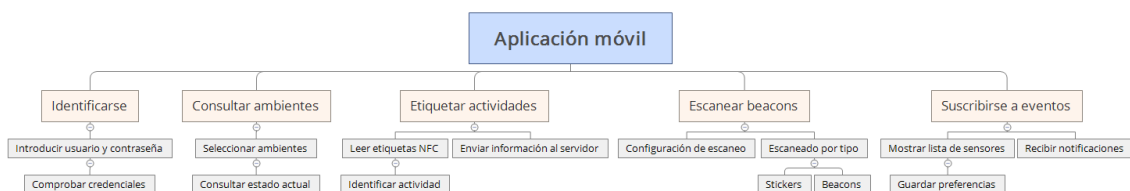


Ilustración 31 Requisitos funcionales (APP Móvil)

- **Requisito funcional 6:** Identificarse

El sistema permitirá a los habitantes de los ambientes identificarse en la aplicación móvil. La identificación debe proporcionar una seguridad acorde a los datos sensibles que proporciona (contraseñas).

- **Requisito funcional 7:** Consultar ambientes

Una vez identificado el habitante en un ambiente concreto, el sistema permitirá consultar todos los ambientes asociados a su perfil. Por cada uno de estos ambientes el habitante/usuario podrá consultar el estado actual en el que se encuentran todos los objetos del ambiente.

- **Requisito funcional 8:** Etiquetar actividades

Mediante la tecnología NFC, el sistema permitirá al habitante etiquetar las actividades que se encuentre realizando en el ambiente inteligente.

- **Requisito funcional 9:** Escenar beacons

Mientras se están realizando actividades, el sistema permitirá el escaneo de balizas (Beacons) las cuales permitirá posicionar al habitante en un lugar físico del ambiente. El tipo de balizas que el sistema es capaz de reconocer son beacons y stickers.

- **Requisito funcional 10:** Suscribirse a eventos

El sistema permitirá a los usuarios suscribirse a los eventos que nos proporcionen los sensores de un ambiente determinado. A su vez, el sistema enviará notificaciones para alertar al usuario de todo lo que sucede en un ambiente.

4.1.1.2 Requisitos no funcionales

A diferencia de los requisitos funcionales, estos requisitos no representan una funcionalidad clara del sistema, o proyecto. Estos requisitos se encargan de describir la calidad que se espera que proporcione un sistema.

ID:	0	Nombre:	Factor de disponibilidad
Descripción:	Tiempo (%) que se encontrará el sistema funcionando de manera continuada		
Requisito:	<ul style="list-style-type: none"> El sistema deberá estar operativo el 98% en una ventana temporal de 365 días. A excepción de actualizaciones o depuración de errores 		

Tabla 9 Requisito no funcional 0

ID:	1	Nombre:	Seguridad
Descripción:	El sistema debe proporcionar seguridad en todas las contraseñas guardadas y no mostrar información sensible a usuarios no autorizados		
Requisito:	<ul style="list-style-type: none"> Contraseñas guardadas en base de datos mediante cifrado SHA1 Panel de administración privado, accesible mediante usuario y contraseña de más de 8 caracteres 		

Tabla 10 Requisito no funcional 1

ID:	3	Nombre:	Viabilidad
Descripción:	Facilidad al usuario para interpretar el sistema		
Requisito:	<ul style="list-style-type: none"> Mensajes de ayudas en sitios con dificultad alta Iconos asociados a eventos y botones para representar la funcionalidad Varios idiomas disponibles para adaptarnos al país del usuario 		

Tabla 11 Requisito no funcional 3

ID:	4	Nombre:	Fortaleza
Descripción:	Adaptarse a situaciones anómalas provocadas por usuarios		
Requisito:	<ul style="list-style-type: none"> • La aplicación debe mostrar mensajes informando de los errores al usuario sin mostrar información del servicio, servicio o aplicación móvil. 		

Tabla 12 Requisito no funcional 4

4.1.1.3 Restricciones

En este punto detallaremos cuales son los requisitos que se deben cumplir para el correcto funcionamiento del sistema.

ID:	1	Nombre:	Software
Descripción:	Todo el software necesario para que entre en funcionamiento todo el sistema desarrollado		
Requisito:	<ul style="list-style-type: none"> • Android 5 o superior • Java 8.0 • Tomcat 8 o superior • Apache2 • MariaDB 5.0 • PHPMyAdmin • Terminal ssh/sftp • Navegador Web 		

Tabla 13 Restricción 1

ID:	2	Nombre:	Dispositivos / Componentes
Descripción:	Cada uno de los dispositivos necesarios para que entre en funcionamiento todo el sistema.		
Requisito:	<ul style="list-style-type: none"> • Servidor dedicado (R. Mínimos): <ul style="list-style-type: none"> ○ 512MB RAM ○ 5GB Disco duro ○ 1 Core • Teléfono Android <ul style="list-style-type: none"> ○ 2GB RAM ○ 16GB Disco duro ○ 2 Core 		

Tabla 14 Restricción 2

4.2 ANÁLISIS DEL SISTEMA

En este punto se pretende identificar y desglosar los posibles casos de uso, así como una representación visual de cómo está construido todo el sistema, sus clases, paquetes y estructura de persistencia de datos.

4.2.1 Casos de uso

Los diagramas de casos de uso muestran el comportamiento del sistema desde el punto de vista del usuario, representando las funciones que un sistema puede ejecutar.

A la hora de elaborar el caso de uso usaremos los siguientes componentes:

- Identificador único
- **Actor:** Usuario principal que será el encargado de utilizar el sistema
- **Condiciones de entrada:** Precondiciones que deben cumplirse antes de realizar una llamada al sistema.
- **Flujo de eventos:** Todas las salidas que nos proporciona realizan una acción.

Vamos a diferenciar dos tipos de caso de uso, uno para el sistema web y otro para la aplicación móvil, por cada uno de estos dos tipos se visualizarán los casos de uso más importantes del sistema.

4.2.1.1 Sistema web

A continuación, mostramos el diagrama de casos de usos de la ampliación del sistema web, diferenciando dos tipos de actores, administrador y usuario.

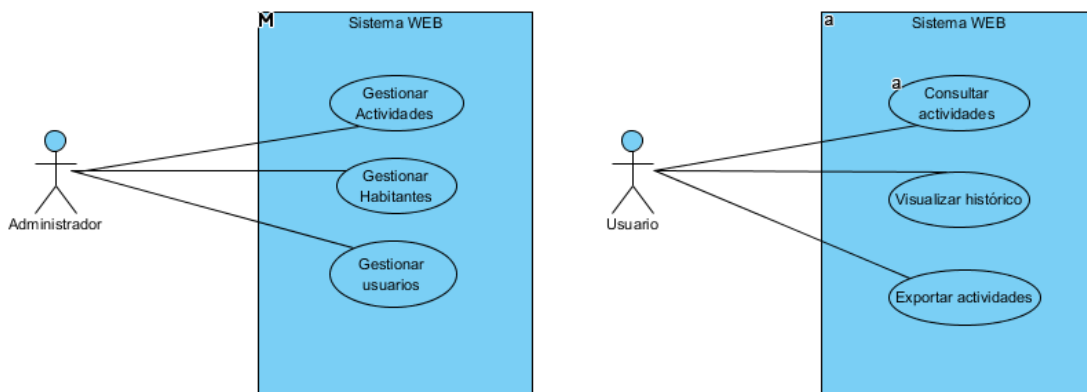


Ilustración 32 Casos de uso

- **Gestionar actividades**

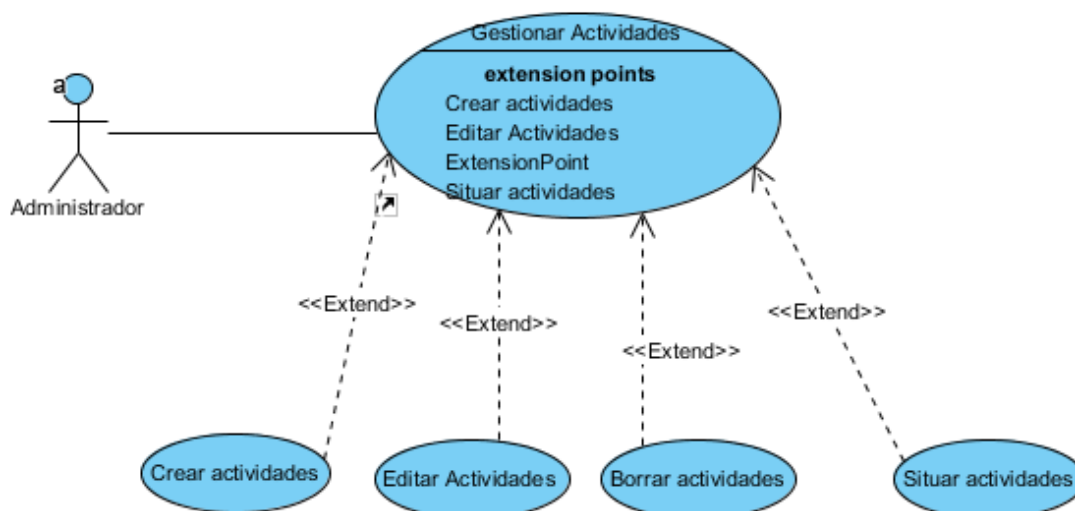


Ilustración 33 Caso de uso: Gestionar actividades

Gestionar habitantes

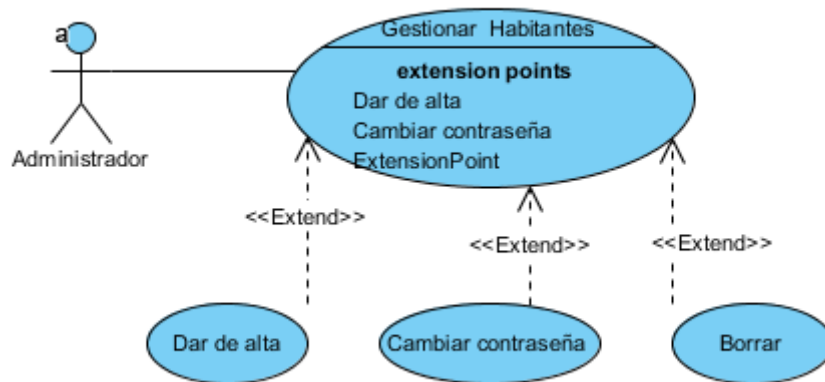


Ilustración 34 Caso de uso: Gestionar Habitantes

- Consultar actividades

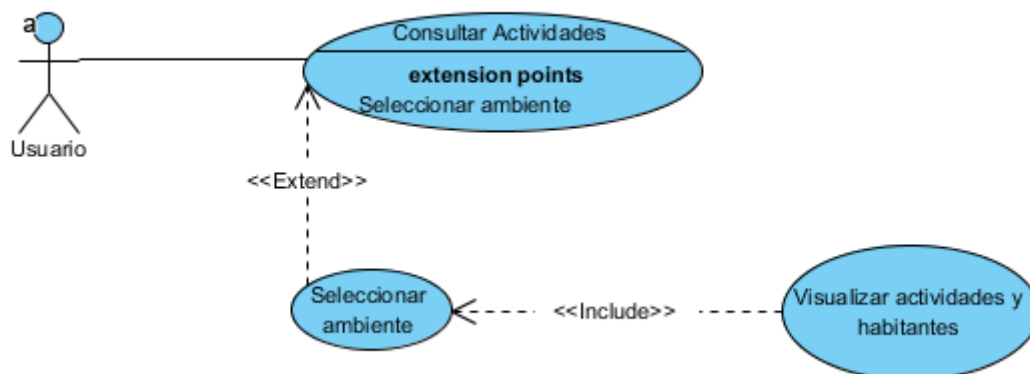


Ilustración 35 Caso de uso: Consultar actividades

- Visualizar histórico

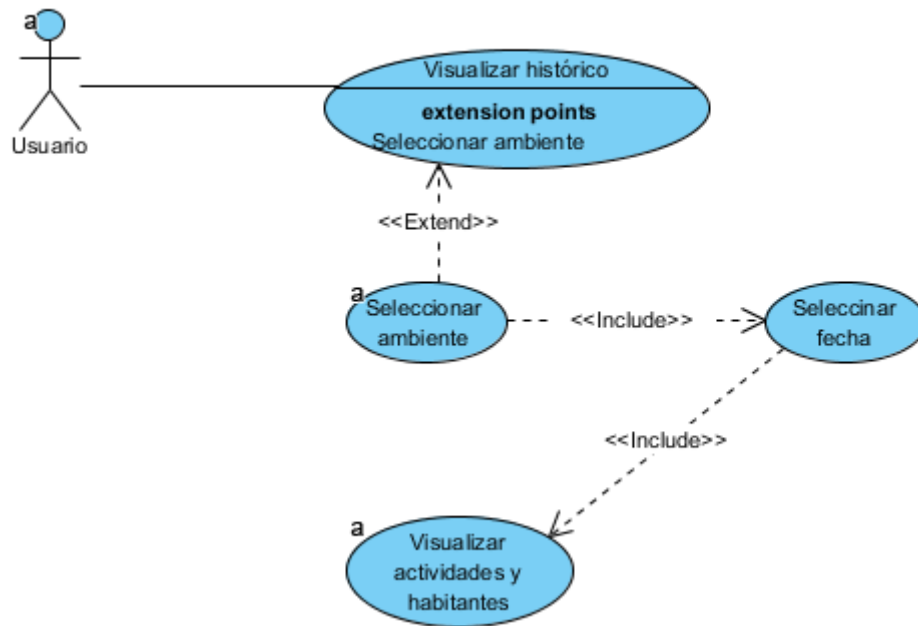


Ilustración 36 Caso de uso: Visualizar histórico

- Exportar actividades

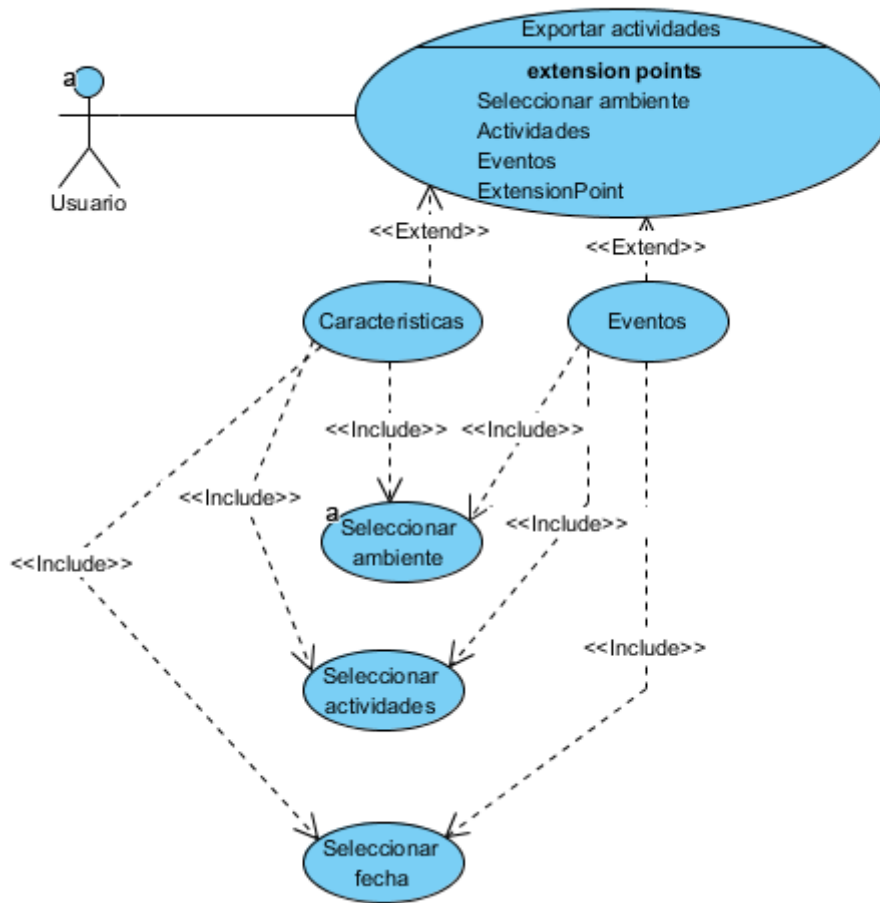


Ilustración 37 Caso de uso: Exportar actividades

4.2.1.2 Aplicación móvil

A continuación, mostramos los diagramas de casos de uso para la aplicación móvil. En este caso tan solo utilizaremos el actor usuario para realizar todas las acciones posibles.

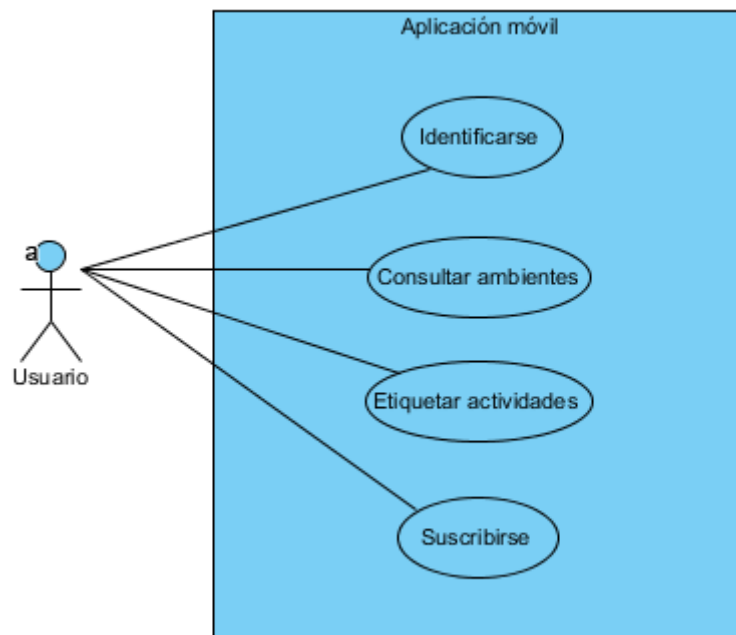


Ilustración 38 Casos de uso aplicación móvil

- **Identificarse**

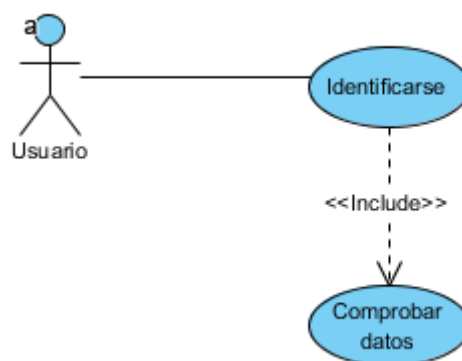


Ilustración 39 Caso de uso: Identificarse

- Consultar ambientes

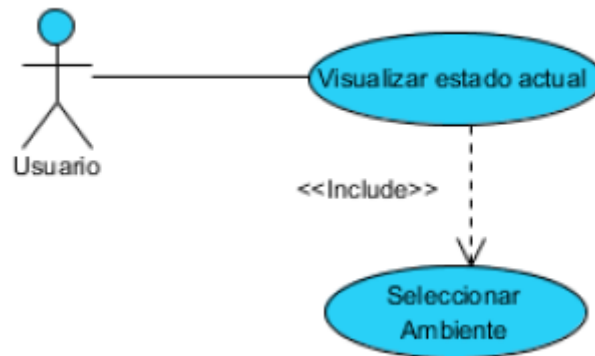


Ilustración 40 Caso de uso: Visualizar estado actual

- Etiquetar actividades

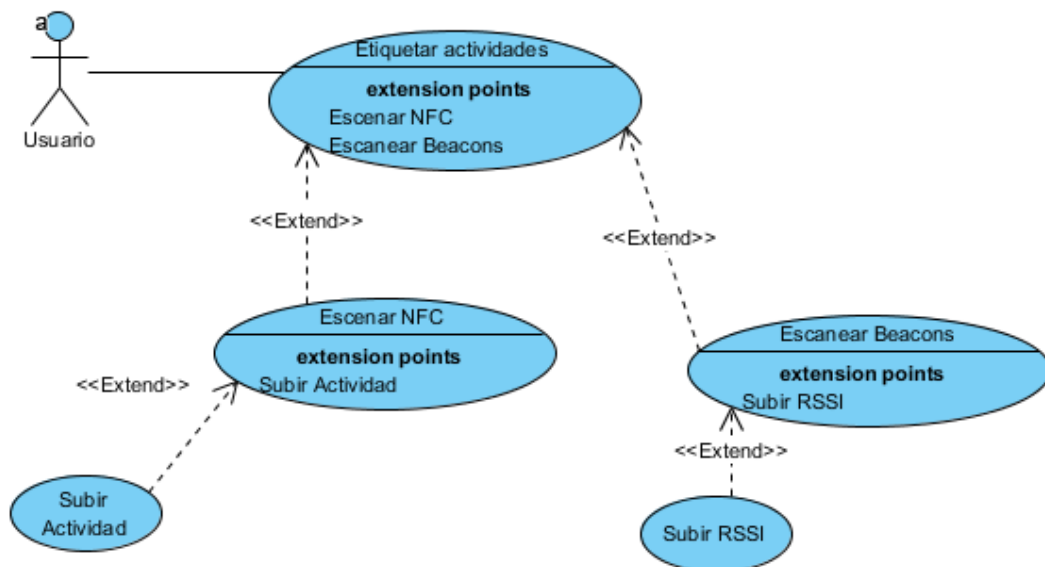


Ilustración 41 Caso de uso: Etiquetar actividades

- **Suscribirse**

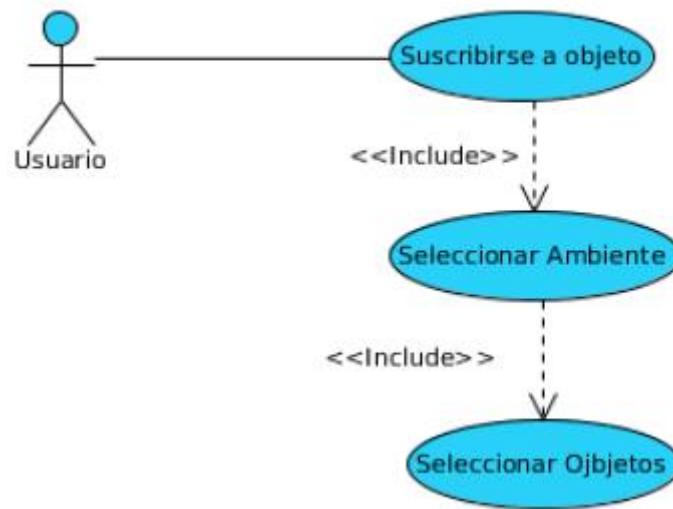


Ilustración 42 Caso de uso: Suscribirse

4.2.2 Escenarios

Antes de adentrarnos en este punto definiremos los escenarios como una secuencia de pasos, de cómo el sistema se comporta dentro de una situación o acción concreta.

A la hora de elaborar cada uno de los escenarios utilizaremos la siguiente plantilla:

Identificador:	
Actores:	
Descripción	
Eventos:	

Tabla 15 Plantilla escenarios

A continuación, se presentarán los escenarios que se han sido considerados más importantes tanto para la aplicación del sistema web como para la aplicación móvil.

4.2.2.1 Sistema web

Identificador:		Crear una actividad
Actores:	Administrador, Sistema web, Servidor	
Descripción	El administrador encargado de gestionar el ambiente “Casa Dani”, desea dar de alta cuatro nuevas actividades, las cuales podrán etiquetarse desde su teléfono móvil.	
Eventos:	<ol style="list-style-type: none"> 1. Administrador pulsa el botón Actividades en el ambiente deseado. 2. El servidor consulta todas las actividades 3. La aplicación muestra una vista con todas las actividades 4. Administrador pulsa el botón nueva actividad 5. La aplicación devuelve un formulario para que introduzca el nombre y descripción de la actividad 6. Administrador finalmente rellena los datos y acepta. 7. El servidor comprueba que no existía dicha actividad y devuelve la vista actualizada. 	

Tabla 16 Escenario (Crear una actividad)

Identificador:	Eliminar una actividad
Actores:	Administrador, Sistema web, Servidor
Descripción	El administrador encargado de gestionar el ambiente “Casa Dani”, desea eliminar una actividad.
Eventos:	<ol style="list-style-type: none"> 1. Administrador pulsa el botón Actividades en el ambiente deseado. 2. El servidor consulta todas las actividades 3. La aplicación muestra una vista con todas las actividades 4. Administrador pulsa el icono de papelera correspondiente a la actividad. 5. La aplicación muestra un panel confirmando la eliminación. 6. El administrador acepta. 7. El servidor devuelve la lista actualizada de actividades.

Tabla 17 Escenario (Eliminar una actividad)

Identificador:	Consultar actividades (Tiempo real)
Actores:	Usuario, Sistema web, Servidor
Descripción	El supervisor habitual del ambiente (Usuario) desea comprobar que actividades se están realizando en el momento de la consulta dentro del ambiente que él monitoriza.
Eventos:	<ol style="list-style-type: none"> 1. Usuario selecciona el ambiente deseado 2. El servidor consulta el estado de los objetos y actividades 3. El sistema presenta un plano con los objetos y actividades que se están realizando

Tabla 18 Escenario (Consultar actividades en tiempo real)

Identificador:	Exportar flujo de eventos
Actores:	Usuario, Sistema web, Servidor
Descripción	El supervisor habitual del ambiente (Usuario) desea exportar las actividades que se realizaron en una fecha comprendida entre el 1/02/2018 al 1/03/2018.
Eventos:	<ol style="list-style-type: none"> 1. Usuario selecciona el ambiente deseado 2. Usuario pulsa sobre el botón “Actividades” 3. El sistema devuelve la lista de actividades disponibles. 4. Usuario selecciona las actividades que desea exportar y elige la consulta de tipo fecha, introduciendo la fecha origen y la fecha fin. 5. El servidor se encarga de buscar todos los eventos producidos en ese intervalo de tiempo. 6. El sistema muestra en una tabla todos los eventos ocurridos en el intervalo fechado. 7. El usuario pulsa sobre exportar. 8. El sistema devuelve una nueva vista donde tiene tres formas de exportar actividades. 9. El usuario pulsa sobre el botón “Exportar flujo de eventos” 10.El sistema devuelve una nueva vista con las distintas opciones del formato de salida 11.El usuario selecciona XLS y pulsa sobre el botón exportar. 12.El servidor devuelve un archivo XLS con el flujo de eventos dentro del intervalo establecido.

Tabla 19 Escenario (Exportar flujo de eventos)

4.2.2.2 Aplicación móvil

Identificador:	Etiquetar una actividad
Actores:	Habitante, APP Móvil, Servidor
Descripción	Un habitante del laboratorio de inteligencia ambiental desea etiquetar la actividad “Ver la televisión” sin hacer uso de tecnología de posicionamiento indoor.
Eventos:	<ol style="list-style-type: none"> 1. El habitante inicia sesión en la aplicación 2. El servidor comprueba si las credenciales son correctas 3. La aplicación móvil devuelve todos los ambientes disponibles para el usuario 4. El habitante selecciona el ambiente “laboratorio de inteligencia ambiental” 5. El habitante pulsa dentro del menú, el botón “Actividades” 6. La aplicación móvil, quedará a la espera de escanear una nueva etiqueta NFC 7. El habitante desliza el teléfono por la etiqueta “Ver TV” situada en el salón de la vivienda 8. La aplicación envía los datos de la actividad al servidor 9. El servidor guarda los datos del habitante y la actividad en la base de datos 10. El habitante disfruta de su descanso viendo la televisión.

Tabla 20 Escenario (Etiquetar una actividad)

4.3 DISEÑO

El diseño software [13] es una amplia descripción, en la cual se identifican los objetivos finales del sistema. Además, en este punto, dentro del diseño, se planean cada una de las estrategias llevadas a cabo para alcanzar todos los requisitos de la implementación.

Dentro del diseño, para este trabajo fin de máster, lo vamos a dividir en dos partes, diseño de clases y diseño de datos. Del mismo modo que en los puntos anteriores. Vamos a diferenciar entre la aplicación de funcionalidades del sistema web y la aplicación móvil.

4.3.1 Diseño de clases

Dentro de la ingeniería del software el lenguaje de modelado más utilizado para realizar el diseño es clases es el lenguaje unificado de modelado (UML), este tipo de lenguaje se encarga de estructurar el sistema, mostrando las clases, atributos y métodos.

Por cada una de las clases que representemos, especificaremos la visibilidad de cada atributo o método del siguiente modo.

Símbolo	Descripción
+	Público
-	Privado
#	Protegido

Tabla 21 Símbolos del diagrama de clases

4.3.1.1 Ampliación del sistema web

Al tratarse de una aplicación de un trabajo realizado en el trabajo fin de grado, vamos a partir del diseño realizado anteriormente.

El sistema web, está dividido en tres paquetes, los cuales albergan patrones, clases o servicios del siguiente modo.

- **Services:** Se utilizará como patrón fachada, es el encargado de recibir y ofrecer las peticiones REST de todos los servicios.

- **Patrón DAO:** El patrón DAO (Data Access Object) se encarga del acceso a los datos. Este patrón, contiene todos los métodos CRUD (Create, Reader, Update y Delete) posibles dentro de una base de datos relacional.
- **POJOs:** Definición del objeto, tan solo cuenta con las acciones GET y SET.

Finalmente, en la imagen inferior se puede visualizar el estado de los paquetes. Por motivos de espacio, se visualizará este diagrama de paquetes sin atributos ni métodos.

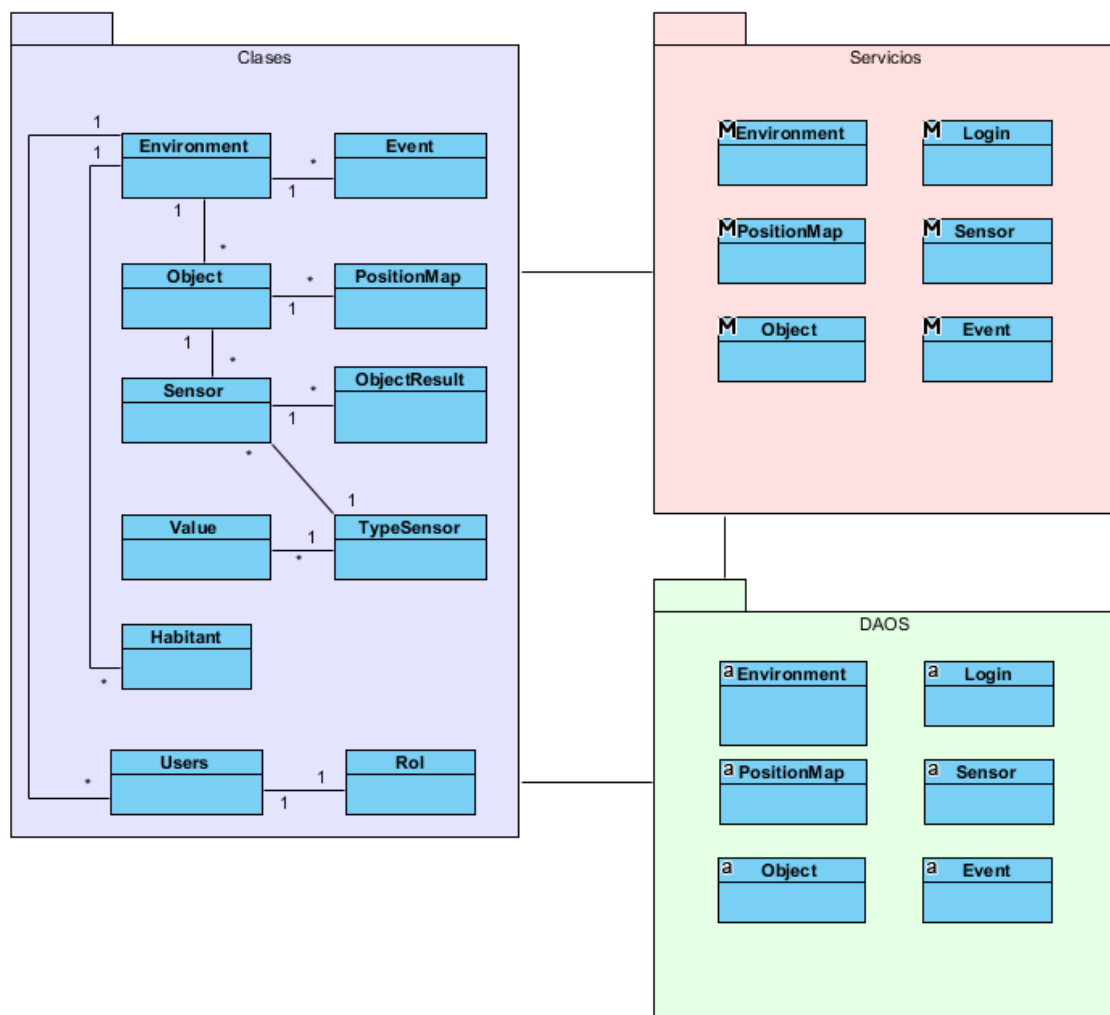


Tabla 22 Diagrama de paquetes

Finalmente, ya que conocemos como están distribuidos los paquetes, se puede mostrar cada uno de los atributos de las clases o POJOs de nuestro sistema.

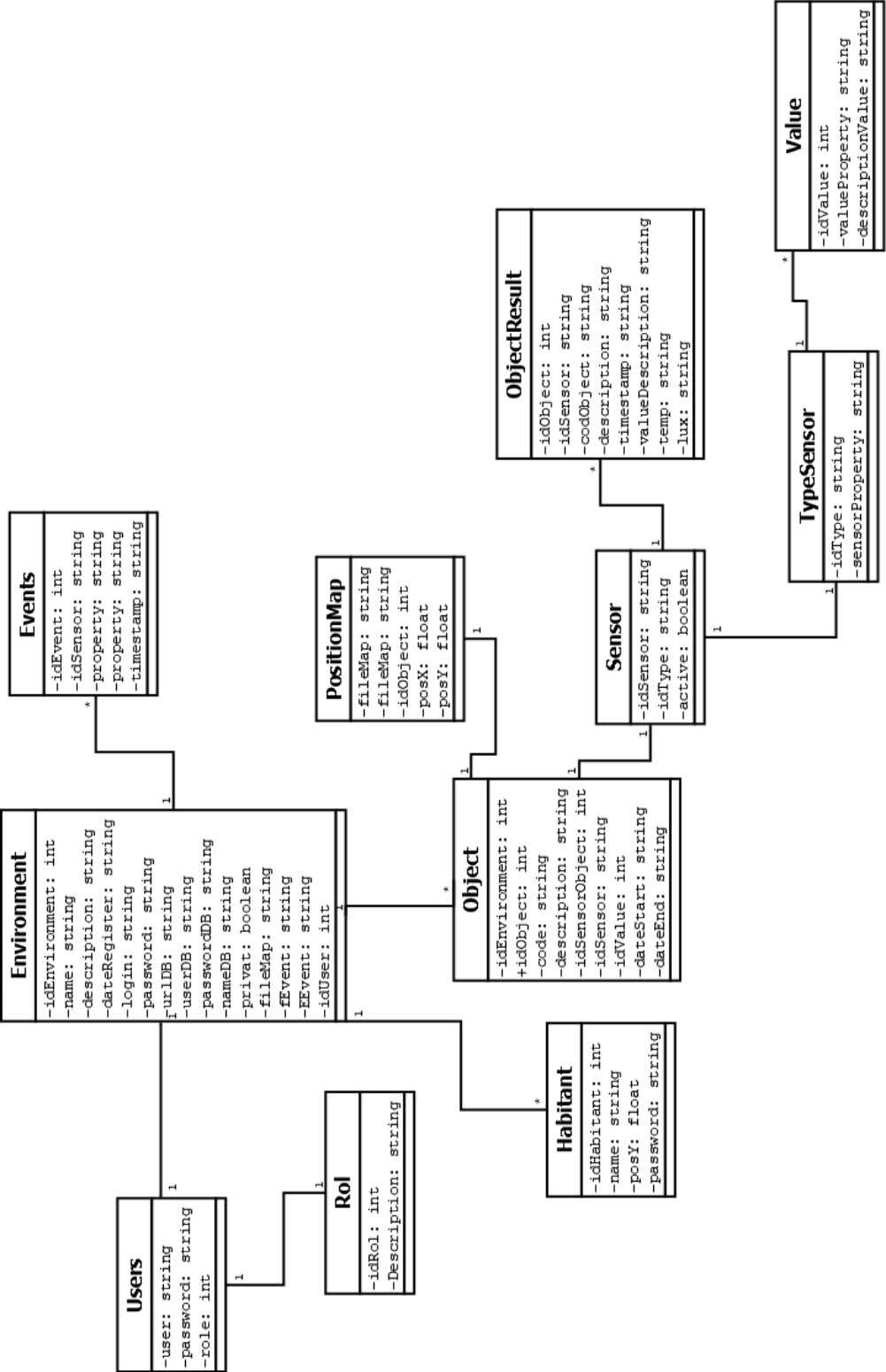


Tabla 23 Diagrama de clases

4.3.1.2 Aplicación móvil

Del mismo modo que agrupamos las clases de la ampliación del sistema web, para la aplicación móvil se han agrupado en los siguientes cinco paquetes.

- **Adapters:** Se encargan de proporcionar objetos para partes específicas de las vistas.
- **Services:** Se encarga de realizar las llamadas GET y POST a los servicios proporcionados por la Api del sistema web. Se ha separado en varias clases para realizar tareas asíncronas en hilos distintos.
- **Persistence;** Son las clases encargadas de guardar toda la información de la aplicación en la base de datos del teléfono móvil.
- **Pojos:** Definición del objeto, tan solo cuenta con las acciones GET y SET.
- **Activities:** Cada una de las vistas lleva asociada una clase encargada de manejar los métodos y atributos.

Una vez definidos todos los paquetes, en la figura inferior se puede visualizar dichos paquetes junto con sus clases y atributos.

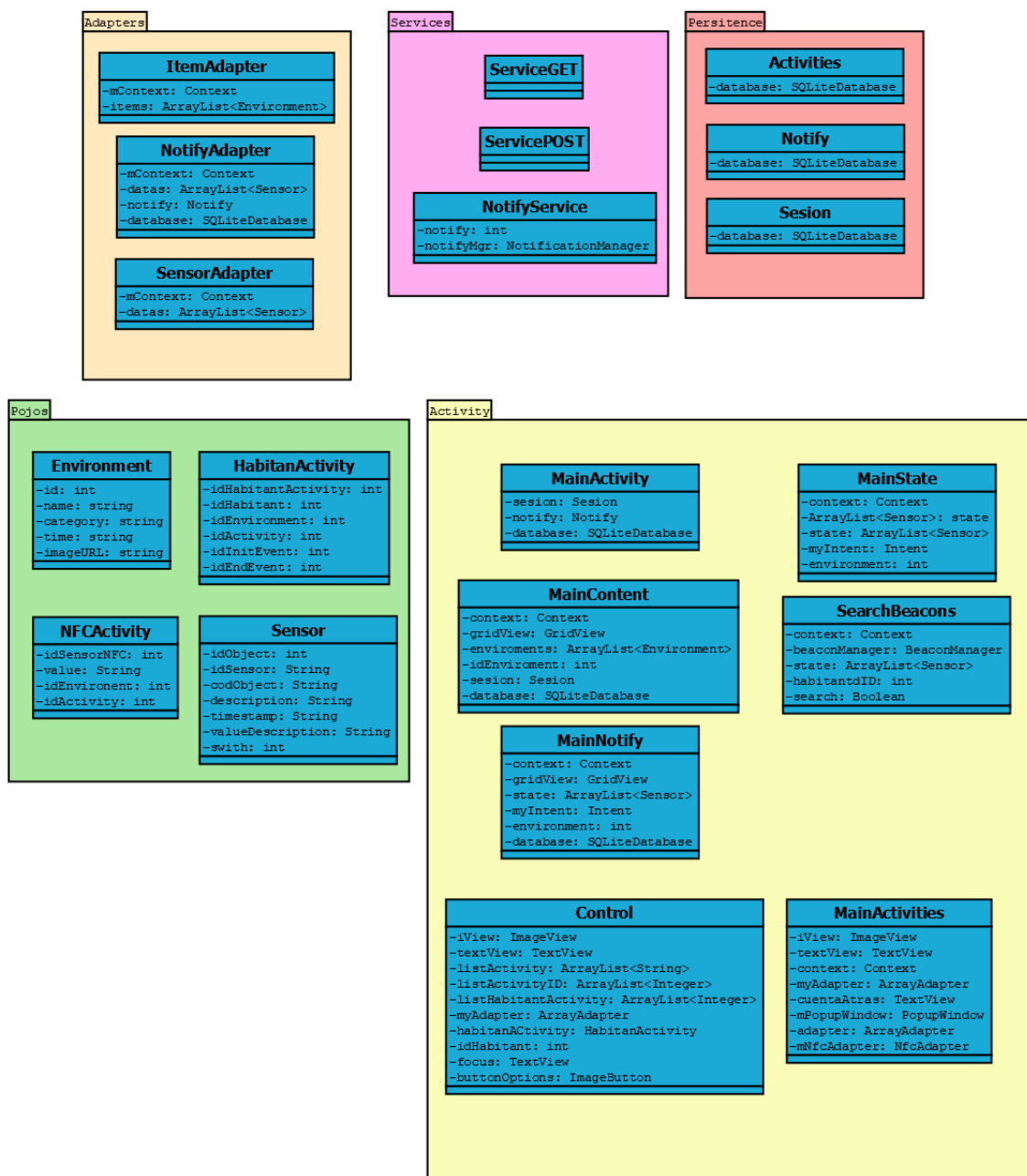


Ilustración 43 Diagrama de paquetes

Tras conocer los paquetes, la visualización del diagrama de clases se puede ver en la siguiente figura.

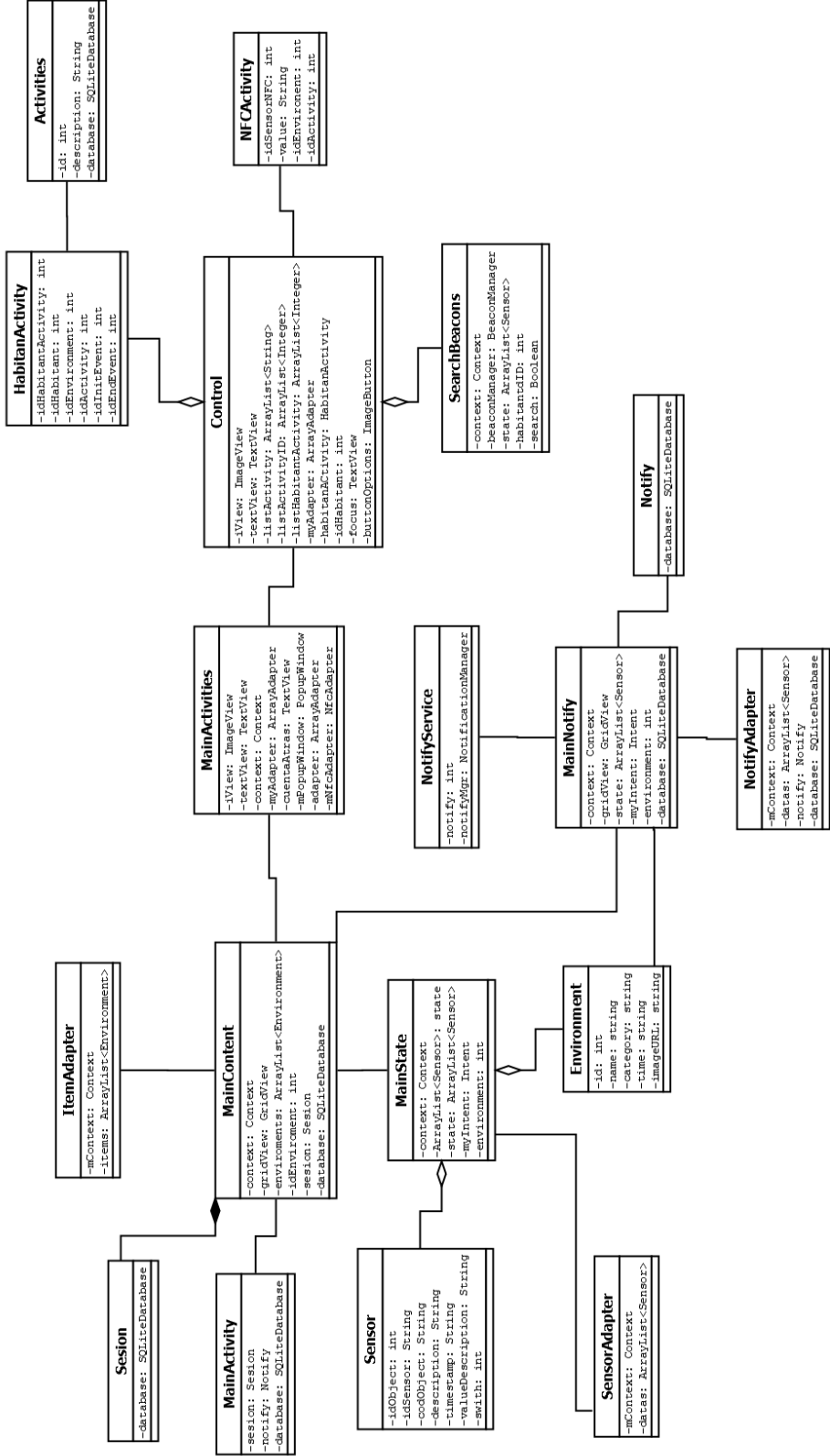


Ilustración 44 Diagrama clases (APP Móvil)

4.3.2 Diseño de datos

Dentro de este punto, se va a presentar la estructura utilizada para realizar la persistencia de todos los datos importantes de nuestro sistema.

Antes de entrar en detalle, en nuestro diseño de datos, vamos a diferenciar entre dos alternativas de base de datos que podían ser utilizadas en el desarrollo del proyecto.

Base de datos relacionales

Son bases de datos, las cuales cumplen con el modelo relacional, es decir, estas bases de datos establecen conexiones o relaciones entre los datos, los cuales se encuentran guardados en forma de tablas. A través de estas conexiones, los datos pueden relacionarse entre sí.

Ventajas

- Los datos deben cumplir requisitos de integridad
- Debido a su largo tiempo en el mercado, tienen una mayor compatibilidad
- Información estructurada
- Jerarquía entre datos

Desventaja

- Escalabilidad
- Estructura centralizada

Base de datos no relacionales

Actualmente nos encontramos con un tipo de base de datos el cual se encuentra en auge. El principal motivo de este auge reside en las redes sociales [14], ya que el crecimiento continuo al que se enfrentan requiere proporcionar información procesada en grandes volúmenes de datos. Es por esto que se le ha dado prioridad a una respuesta rápida de datos y no a la estructura de como la información es guardada.

Para esto, estas bases de datos utilizan un diseño orientado a propiedades, por lo que en el caso de añadir una nueva propiedad no es necesario añadirla dentro de una tabla ni realizar modificaciones en el diseño. Esto es posible ya que la estructura se encuentra organizada mediante grafos.

Ventajas

- Flexibilidad
- Escalabilidad
- Estructura distribuida

Desventajas

- Falta de estandarización
- Soporte multiplataforma

Tras conocer los dos tipos de base de datos, para el desarrollo de este sistema se ha utilizado una base de datos relacional SQ, ya que nos brinda una consistencia mucho más elevada. Además de tratarse de un sistema cerrado el cual el crecimiento nunca será tan grande como para llegar a tener que descentralizar la estructura o información.

Una vez elegida la base de datos, el esquema utilizado para la persistencia de los datos es el siguiente.

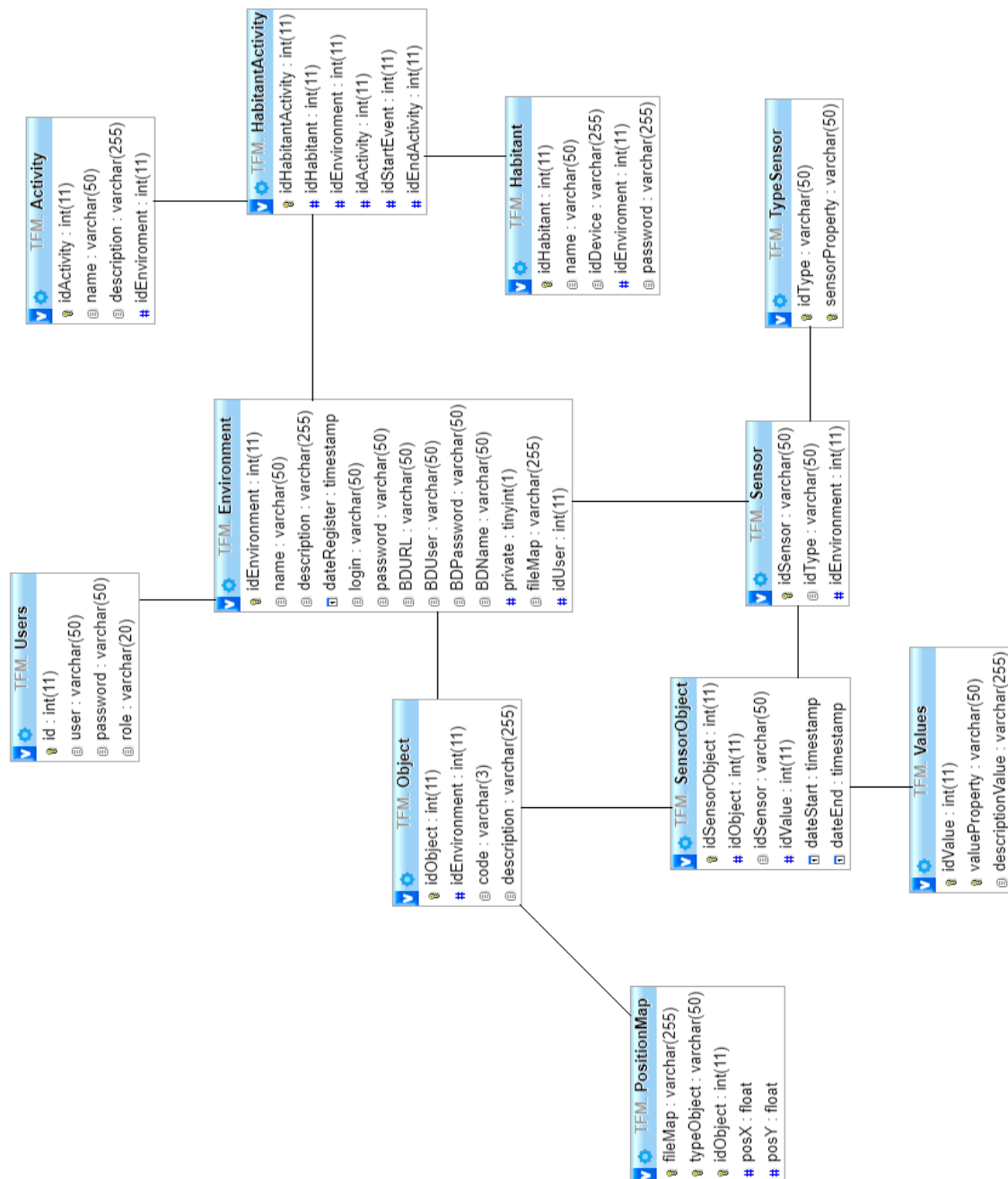


Ilustración 45 Esquema conceptual modificado

4.3.2.1 Descripción de tablas

A continuación, se van a definir cada una de las tablas mostradas en la ilustración superior.

- **Users:** Se encarga de la información de los usuarios que pueden acceder al sistema mediante el inicio de sesión. La contraseña proporcionada por el sistema será guardará mediante cifrado SHA1.
- **Activity:** Es la encargada de almacenar la información de las actividades que pueden desarrollarse en cualquier ambiente inteligente.
- **HabitantActivity:** Actúa de unión entre la actividad que se encuentra realizando un habitante. Gracias a esta tabla podemos obtener el inicio y fin de la actividad.
- **Habitant:** Realiza la persistencia de los datos del habitante, estos datos son necesarios para obtener información de que persona se encuentra realizando cada una de las posibles actividades.
- **Environment:** Cuenta con toda la información relevante del ambiente. Además, cuenta con las credenciales para conectarse a la base de datos donde se encuentra toda la información de los sensores.
- **Object:** Da conocimiento a la información proporcionada por el sensor.
- **PositionMap:** Se encarga de posicionar un objeto en el mapa del ambiente.
- **SensorObject:** Actúa de unión entre el sensor físico y el objeto virtual.
- **Values:** Almacena los valores que puede tomar un objeto.
- **Sensor:** Persiste todos los sensores en cada uno de los ambientes que se encuentran dados de alta.

- **TypeSensor:** Almacena los tipos de sensores que pueden trabajar en el sistema junto con cada una de sus propiedades y características.

4.3.3 Diseño de la interfaz en la aplicación móvil

Dentro de esta fase, definiremos cual será la apariencia visual que tendrá la aplicación móvil desarrollada. Esta fase es tan importante como una buena definición de la arquitectura, ya que el usuario final el que estará con total contacto y sin un buen diseño, todo el trabajo de arquitectura puede fracasar.

En esta fase se presentarán las ilustraciones de cada pantalla, estilos y metáforas.

4.3.3.1 Story board

A continuación, se va a presentar un guion gráfico en forma de ilustraciones donde se mostrarán las principales pantallas de la aplicación móvil junto con la funcionalidad de cada una de ellas.



Iniciar sesión

Nada más abrir la aplicación, nos encontramos con la vista principal donde nos pide nuestras credenciales para acceder al sistema.

Dichas credenciales serán dadas de alta desde el panel de administración del ambiente, implementado en el sitio web del sitio, por lo que en la aplicación móvil no tendremos posibilidad de registrarnos automáticamente.

Ilustración 46 StoryBoard (Inicio de sesión)

Tus ambientes

Después de iniciar sesión, el usuario puede visualizar una nueva vista, en esta, se mostrará al usuario todos los ambientes a los que su perfil tiene permiso para acceder.

Por defecto, nos encontramos con el primer ambiente seleccionado, pero podemos seleccionar manualmente cada uno de ellos para ejecutar cualquier acción desde el menú de la aplicación.

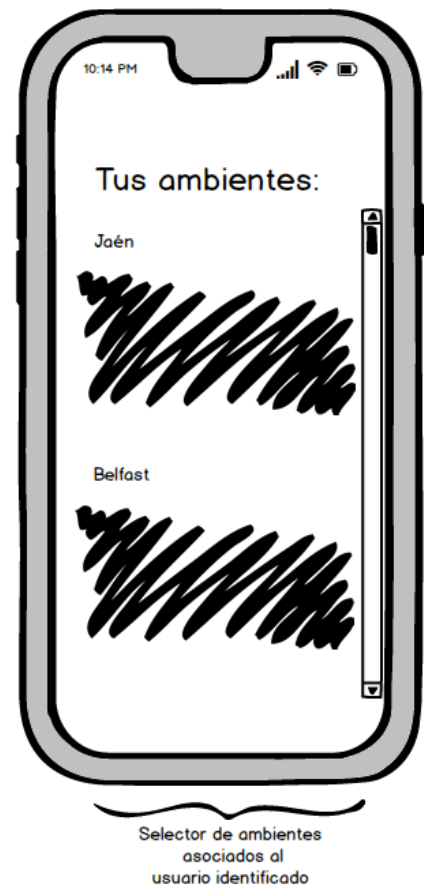


Ilustración 47 StoryBoard (Ambientes disponibles)



Ilustración 48 StoryBoard (Menú)

Menú

El menú de la aplicación se encuentra en el margen derecho de la pantalla, siendo este el lugar por defecto que nos proporciona la plantilla de Android.

En este menú, aparte de mostrar una pequeña imagen que identifique la aplicación, el usuario puede acceder directamente a cada uno de las opciones que se están detallando en este punto.

Estado actual

Mediante esta pantalla, el usuario puede visualizar de una manera sencilla y rápida el estado en el que se encuentra cada uno de los objetos del ambiente que está consultando.

Se ha utilizado el color verde para representar el estado, abierto/encendido, y el rojo para representar su opuesto.

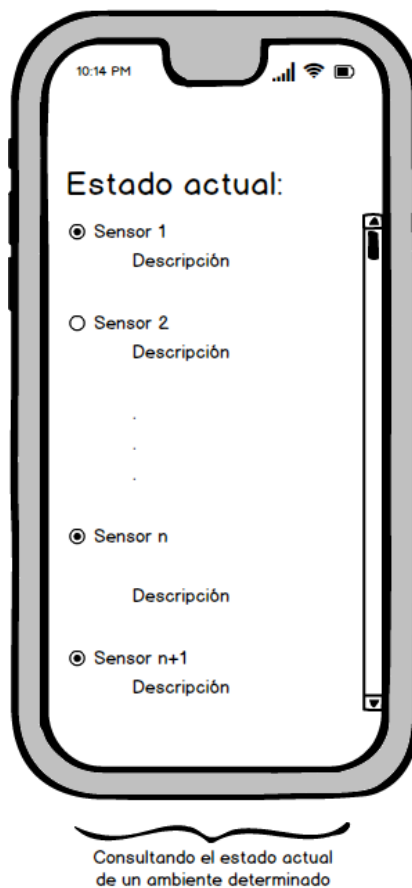


Ilustración 49 StoryBoard (Estado actual de un ambiente)

Actividades



Ilustración 50 StoryBoard (Etiquetado de actividades)

Desde esta vista, el usuario puede etiquetar actividades manualmente, para ello, hace uso de la tecnología NFC, por lo que el procedimiento de etiquetado es muy simple. Una vez el usuario se encuentre en la vista, si podrá deslizar el teléfono por una etiqueta NFC. Si el teléfono cuenta con la tecnología NFC y la etiqueta tiene asociada una actividad, el usuario verá una nueva tabla donde se le indicará todas las actividades que se encuentra realizando de manera simultánea.

Por otro lado, si el teléfono no cuenta con dicha tecnología se le notificará al usuario mediante mensajes informativos, indicándole que no puede usar dicha funcionalidad.

Notificaciones



Ilustración 51 StoryBoard (Notificaciones)

Por último, la aplicación móvil, ofrece al usuario suscribirse a eventos producidos por cualquier objeto que se encuentre activo en el ambiente seleccionado, mediante un simple check, el usuario puede suscribirse al objeto.

Una vez suscrito, cada vez que se produzca un cambio en el ambiente, el usuario recibirá una notificación push al sistema, indicando la hora, y estado en el que se encuentra el objeto.

4.3.3.2 Estilo

Al tratarse de una aplicación móvil desarrollada en Android, se va a definir una guía de estilo que englobe todos los elementos de la interfaz, creando de este modo una fuerte consistencia.

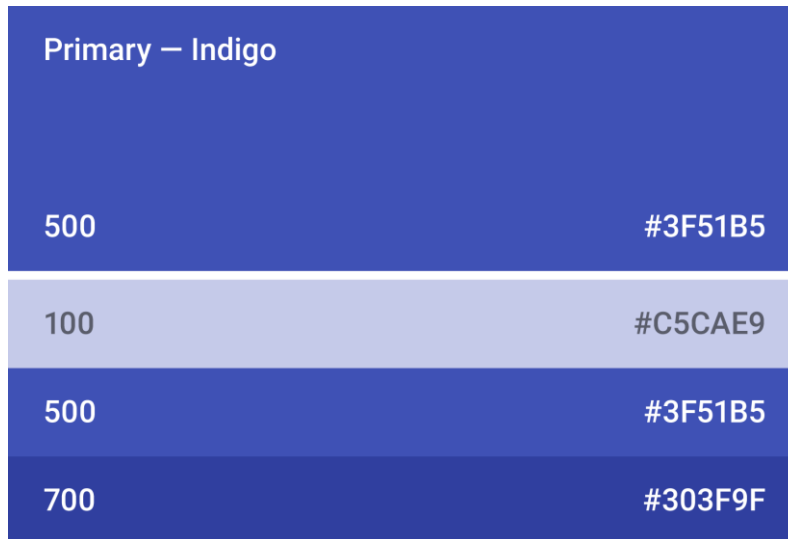
Desde 2014, Google recomienda utilizar un estilo denominado “Material Design” [15], el cual define una serie de patrones de diseño enfocados plenamente a la visualización, destacando en el uso de colores vibrantes sobre planos más apagados.

- **Iconos:** Cada icono utilizado, está enfocado a representar mediante una forma gráfica simple una posible acción por parte del usuario. Para ello hemos utilizado los iconos que nos proporciona el estilo Material Design [16].
- **Tipografía:** En todos los textos de la aplicación se ha utilizado la fuente Roboto, ya que es la principal fuente que recomienda Google para la escritura proveniente del latín. En cuanto a tamaños hemos definido texto 12pt, títulos 20, 16 y 14, en función de la importancia de cada punto.

abcdefghijklmnopqrstvwxyz
ABCDEFGHIJKLMNPOQRSTUVWXYZ
0123456789 (!@#\$%&.,?::)

Ilustración 52 Fuente ROBOTO

- **Colores:** Como hemos comentado, Material Design nos proporciona una serie de paletas donde podemos elegir el color principal de nuestra aplicación. En su desarrollo nos hemos apoyado por el color azul, obteniendo la paleta de color que se puede visualizar en la imagen inferior.



Primary – Indigo	
500	#3F51B5
100	#C5CAE9
500	#3F51B5
700	#303F9F

Ilustración 53 Paleta de colores

- **Logo corporativo**

El logo utilizado en la aplicación web y móvil de este trabajo fin de máster debe ser el proporcionado por el ecosistema de aplicaciones UJAMI.



Ilustración 54 Logo corporativo UJAMI MONITORING

4.3.3.3 Metáforas

Gracias a las metáforas comunicaremos a los usuarios conceptos mediante imágenes o formas. De este modo nos ahorramos de introducir texto en las vistas. Gracias a esto, creamos aplicaciones mucho más sencillas y con una apariencia más cuidada y minimalista.

Normalmente en el desarrollo software, las metáforas, pueden ser, verbales, visuales o globales, para el desarrollo de esta aplicación móvil, las metáforas visuales utilizadas han sido visuales, representándose en forma de iconos.

A continuación, mostraremos algunas de las más importantes:

- **Estado de un objeto**

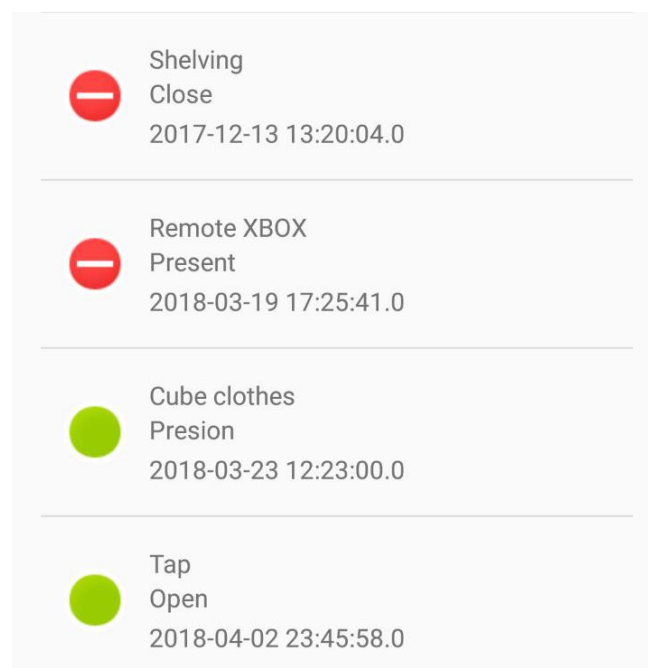


Ilustración 55 Metáfora (Estado de un objeto)

En este punto, representamos el estado de un objeto mediante un círculo de color rojo para indicar que se encuentra cerrado/apagado y un icono verde para su antagónico.

- **Volver atrás y ajustes**



Ilustración 56 Metáfora (Volver y ajustes)

Se ha utilizado una flecha indicando que se puede volver atrás y una herramienta para cambiar los ajustes de la vista. Esta metáfora, está totalmente aceptada por el diseño material design.

- **Accesos del menú**

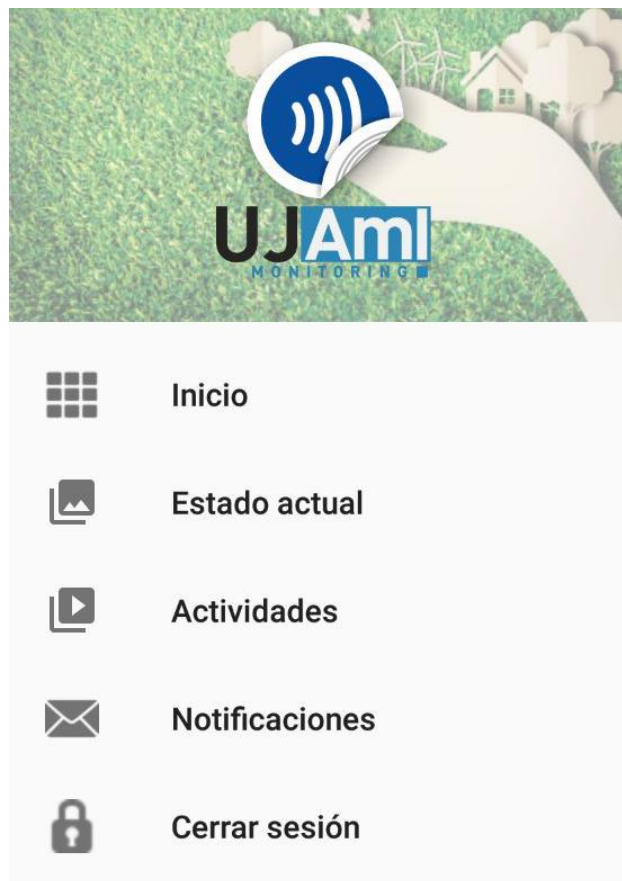


Ilustración 57 Metáfora (Menú)

Por cada enlace del menú, se ha asociado un icono para poder identificar rápidamente la acción que desea realizar el usuario.

4.4 IMPLEMENTACIÓN

Tras conocer todo el proceso de ingeniería del software nos adentramos en el punto más amplio de nuestro desarrollo software. El proceso de implementación detalla desde los pilares del proyecto hasta su puesta en producción en la etapa final, por lo que una buena elección en este punto ahorrara tiempo en futuros trabajos o actualizaciones que se quieran llevar a cabo.

Dentro de este trabajo fin de máster, al tratarse de ampliar funcionalidades a un trabajo existente, muchas de las tecnologías debían de mantenerse para no modificar la estructura del proyecto.

4.4.1 Arquitectura

La arquitectura utilizada tanto para la aplicación de funcionalidades como para el desarrollo de una nueva aplicación móvil, se apoya en el modelo cliente – servidor.

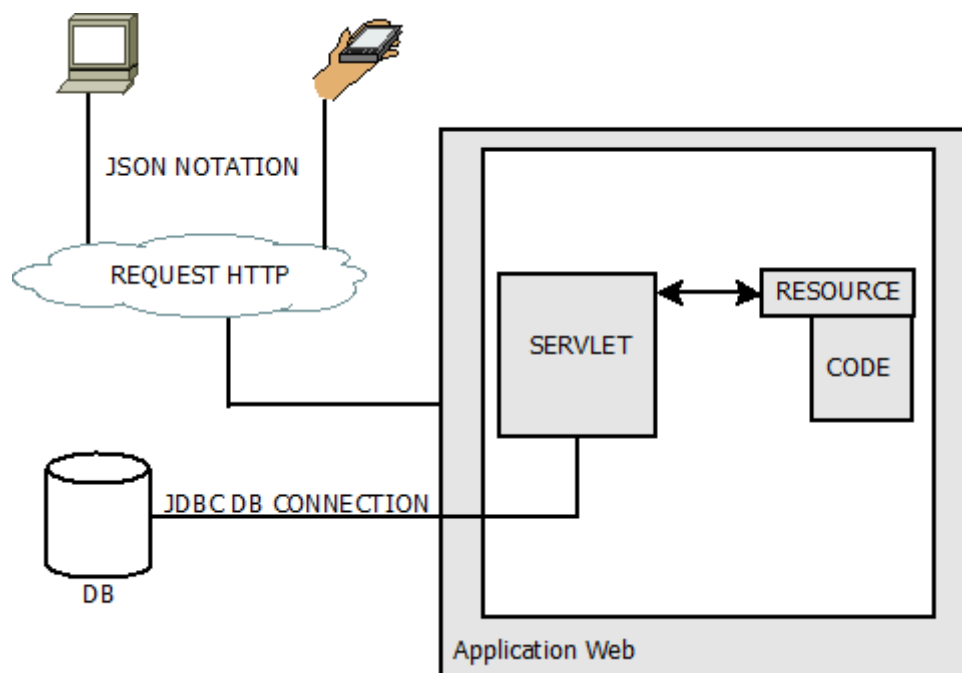


Ilustración 58 Arquitectura cliente – servidor

Del mismo modo que podemos visualizar en la imagen superior, nos encontramos con cuatro objetos que se interconectan entre si para mostrar

toda la información. A continuación, se procederá a describir cada uno de estos objetos.

- **Base de datos**

El sistema cuenta con varias bases de datos, las cuales corresponden a dos tipos de información, por un lado, tenemos la definición de sensores, ambientes y configuración del sistema mientras que, por otro lado, el sistema tiene la posibilidad de conectarse a base de datos externas para obtener los datos proporcionados por los sensores. Toda esta información será ofrecida únicamente al servidor mediante la aplicación desarrollada.

- **Servidor**

En este objeto reside la aplicación desarrollada, también llamada a lo largo de esta memoria como sistema. Mediante una API (Interfaz de programación de aplicaciones), la aplicación desplegada en el servidor obtiene toda la información que se encuentra persistida en las bases de datos, la procesa mediante todas las fases desarrolladas y una vez organizada es ofrecida a cualquier cliente mediante un modelo de datos en texto llamado JSON. Este intercambio de información entre el servidor y los distintos clientes se lleva a cabo mediante el protocolo REST el cual definimos en el capítulo 2.

- **Clientes**

El último punto al que llega la información procesada, es a cada uno de los clientes, al tratarse de una API, dicha información puede ser consumida independientemente del sistema operativo o el tipo de dispositivo, es por esto que tan solo no es necesario que el protocolo HTTP esté operativo. Como se ha comentado a lo largo de esta memoria, en este trabajo fin de máster, los clientes desarrollados han sido una aplicación web y una aplicación móvil.

4.4.2 Tecnologías utilizadas

A continuación, mostraremos las tecnologías utilizadas para el desarrollo de este trabajo fin de máster. De modo que se puedan visualizar de una forma más sencillas, han sido agrupadas entre el servidor y los distintivos tipos de clientes que se han desarrollado.

4.4.2.1 Servidor

- **Java**

Java es un lenguaje de programación orientado a objetos creado en 1995 por la empresa Sun Microsystems [17]. La filosofía que ha llevado a Java a ser uno de los lenguajes de programación más utilizados en la actualidad es WORA. Esta filosofía permite a los desarrolladores software desplegar las aplicaciones en cualquier sistema operativo sin necesidad de modificar el código de la aplicación.

Actualmente Java es propiedad de Oracle, siendo adquirida en el año 2009 por dicha empresa por la cantidad de 5.710 millones de dólares [18].

Las principales ventajas que han hecho que en este trabajo se hiciera uso de este lenguaje son la siguientes:

- Lenguaje OO (Orientado a objetos)
- Multiplataforma
- Código fuente abierto

- **Jersey**

Jersey es un framework [19]orientado al desarrollo de aplicaciones web y APIs en el lenguaje de programación JAVA.

Jersey nos proporciona soporte para trabajar con JAX-RS, ahorrándonos muchas líneas de código y aumentando la seguridad y funcionalidad.

La principal razón por utilizar Jersey y no otro framework ha sido el gran soporte con el que cuenta y su robustez.

- **JDBC**

JDBC (Java DataBase Connectivity) es una API de conexión a base de datos JAVA el cual proporciona una serie de funciones las cuales se encargan de intercambiar datos entre el servidor y la base de datos.

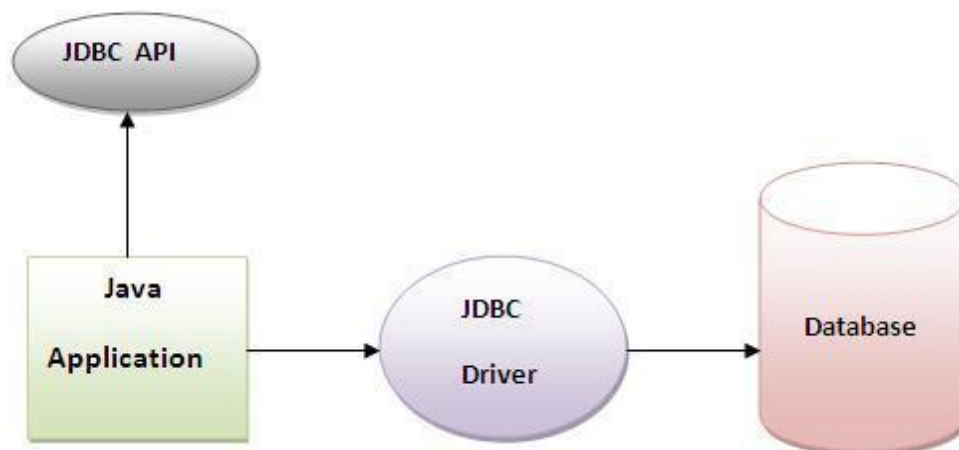


Ilustración 59 Esquema JDBC

JDBC [20] cuenta con un esquema de funcionamiento muy simple, y se encarga de traducir las consultas a cualquier lenguaje SQL independientemente de la base de datos que estemos utilizando.

- **MariaDB**

MariaDB nace como una alternativa libre y gratuita a la conocida base de datos MySQL tras la compra de Oracle. MariaDB cuenta con licencia LGPL y con motor MyISAM, por lo que de una manera muy rápida podemos convertir una base de datos MySQL a MariaDB.

4.4.2.2 Cliente Web

- **HTML5**

HTML5 Nace como estándar entre los principales navegadores web. En su actualización número cinco, HTML [21] nos proporciona una mayor integración entre distintos navegadores, apoyándose en los nuevos dispositivos móviles y los distintos tipos de resoluciones.

Una de las nuevas novedades que incorpora HTML5, frente a sus versiones anteriores es la incorporación de contenido multimedia directamente en la web.

- **CSS3**

CSS3 Es un lenguaje de marcas [21] el cual define el diseño de una web mediante una serie de hojas de estilo. Estas marcas se pueden aplicar mediante ID, clases o etiquetas HTML.

- **Bootstrap 3**

Bootstrap 3 es un framework de programación web desarrollado por la empresa Twitter, el cual define una serie de vistas divididas en rejillas. Gracias a estas rejillas, podemos adaptar el contenido a cualquier resolución, adaptándose y haciéndose la navegación mucho más intuitiva y amigable.

Mientras se llevaba a cabo el desarrollo de este trabajo fin de máster ha sido publicada Bootstrap 4, incorporando esta versión nuevos complementos a las vistas.

- **Angular**

Angular es un framework JavaScript, el cual ha sido desarrollado por Google. Este framework, permite a los desarrolladores crear aplicaciones en la parte cliente de una aplicación web, se basa principalmente en el lenguaje JavaScript [22], mientras que en su segunda versión trabaja mediante TypeScript.

Angular se apoya en el patrón MVC (Modelo – Vista – Controlador)

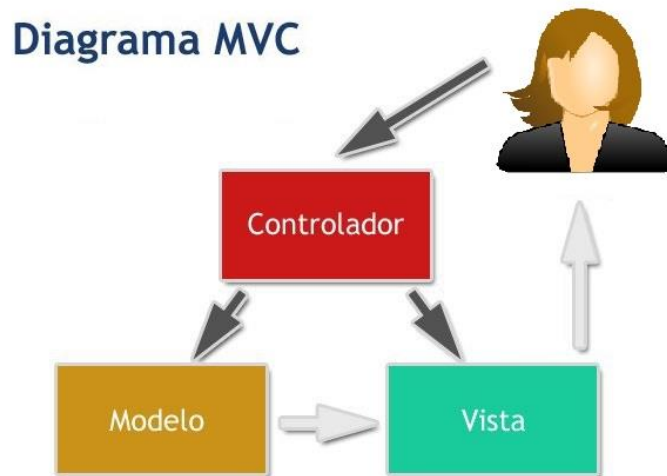


Ilustración 60 Diagrama Modelo -Vista – Controlador

- **Vista:** Todo el contenido y representación de la información.
- **Controlador:** Se encarga de toda la lógica del cliente, encarga de realizar peticiones al servidor y presentar los datos a la vista.
- **Modelo:** Además de utilizarse como parte del diseño de datos que se mostrará en la vista, Angular nos permite definir ámbitos en los que la información puede ser presentada filtrada o procesada.

El principal motivo por el que se ha elegido trabajar con Angular en la parte cliente de la aplicación web se encuentra en la reusabilidad de código, y la gran facilidad a la hora de filtrar los datos en las vistas presentadas al usuario.

4.4.2.3 Cliente Móvil

En la fase de análisis de este trabajo fin de máster, se estudiaron las alternativas ofrecidas en sistemas operativos móviles, como iOS, Android y Windows Phone, para tomar una decisión, se estudió la cuota de mercado ofrecida por cada uno de estos sistemas operativos.

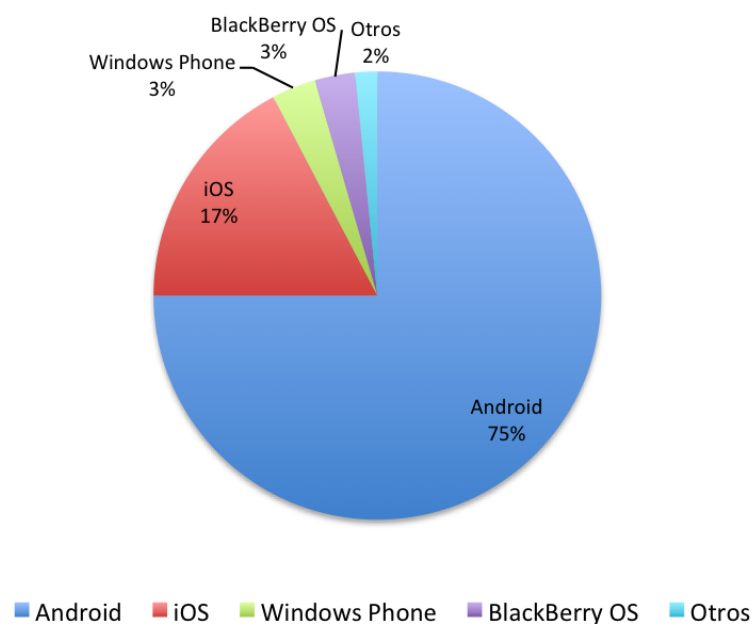


Ilustración 61 Cuota de mercado móvil

Como se puede observar en la imagen superior, en el estudio realizado por la web de tecnología MyComputer [23], en el año 2017, Android contaba con un 75% de la cuota de mercado, al tener una cuota mayor, es más fácil que la aplicación desarrollada en este trabajo fin de máster sea compatible con un mayor número de usuarios.

- **Android**

Android es un sistema operativo móvil con núcleo Linux, pensando principalmente para dispositivos móviles táctiles. En sus inicios Android fue desarrollado por Android Inc, posteriormente en el año 2005 esta empresa fue absorbida por la gigante Google. Desde su compra, cada año Google presenta una nueva versión de Android, asociándole a esta el nombre en clave de un postre azucarado, en la actualidad (2018), la última versión presentada es la 8.0, llamada Oreo. A esta versión se le asocia un número de API, en los que los desarrolladores debemos apoyarnos para implementar aplicaciones.

En este punto, otra de las decisiones tomadas en este trabajo, ha sido elegir la versión mínima funcional en la que trabajará la aplicación de etiquetado de actividades. En este punto volvemos a estudiar la cuota de mercado, pero esta vez fijándonos en las versiones más utilizadas en el año de desarrollo [24] y que estas cuenten con la tecnología NFC, la cual está en funcionamiento desde la API número 9 correspondiente a diciembre de 2010.

Version	Codename	API	Distribution
2.3.3-2.3.7	Gingerbread	10	0.3%
4.0.3-4.0.4	Ice Cream Sandwich	15	0.4%
4.1.x	Jelly Bean	16	1.7%
4.2.x		17	2.6%
4.3		18	0.7%
4.4	KitKat	19	12.0%
5.0	Lollipop	21	5.4%
5.1		22	19.2%
6.0	Marshmallow	23	28.1%
7.0	Nougat	24	22.3%
7.1		25	6.2%
8.0	Oreo	26	0.8%
8.1		27	0.3%

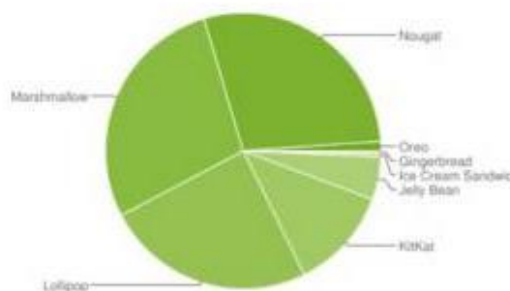


Ilustración 62 Cuota de mercado Android

Como se puede observar en la ilustración superior, la versión Android más dominante en el mercado actual se trata de Nougat, correspondiente con la versión 7.0. Esta versión es precedida por Lollipop, que, aun siendo lanzada en el año 2015, aun cuenta con un importante de usuarios activos.

Tras conocer dichos datos, se decide desarrollar la aplicación con una interfaz de aplicación mínima 21, dicho de otro modo, nuestra aplicación será compatible con dispositivos Lollipop versión 5.1 o superiores.

Una vez realizada esta introducción, es de vital importancia conocer la estructura en la que se basa Android.

Estructura del sistema operativo Android

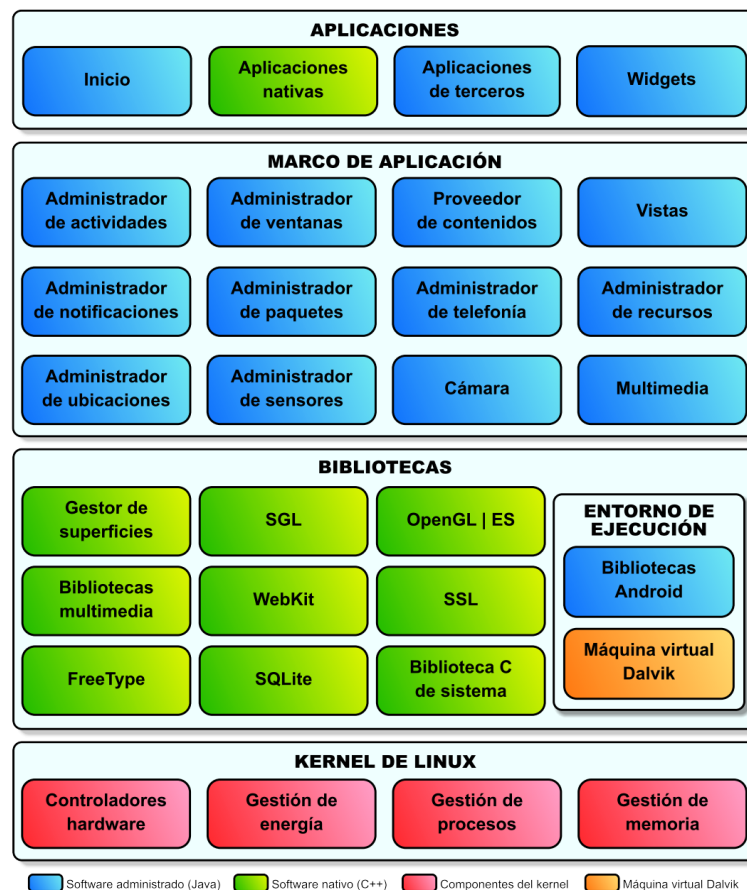


Ilustración 63 Capas Android

- **Kernel:** Como se ha comentado en el inicio de este punto, Android cuenta con un núcleo Linux, concretamente basado en la versión 2.6. Este núcleo puede trabajar en distintas

arquitecturas, aunque la más usada en dispositivos móviles es ARM.

- **Bibliotecas:** Dentro del ámbito de la programación software, en esta capa nos encontramos con todas las librerías que pueden ser utilizadas en cualquier sistema operativo Android. Estas librerías vienen compiladas desde código C o C++ y proporcionan toda la funcionalidad básica del dispositivo. Entre las más importantes nos encontramos con OpenGL, SQLite, SSL, WebKit o bibliotecas multimedia. Las cuales han sido plenamente utilizadas en el desarrollo de la aplicación móvil.
- **Marco de aplicación:** En esta capa, Android nos proporciona un conjunto de clases y servicios en forma de API para acceder a todas las librerías descritas en la anterior capa. Gracias a esta capa, el desarrollador puede utilizar todo el hardware del dispositivo sin necesidad de conocer la información del fabricante o sus necesidades.
- **Aplicaciones:** Por último, nos encontramos en a la capa final, en ella se presentan todas las aplicaciones del dispositivo, independientemente de si son aplicaciones con interfaz o en segundo plano. En esta capa, nos encontramos con todas las aplicaciones que vienen preinstaladas en el dispositivo y que no pueden ser eliminadas desde el gestor de aplicaciones.

Ciclo de vida

En la fase de desarrollo o implementación de esta aplicación Android, se ha tenido muy en cuenta los ciclos de vida por los que pasa nuestra aplicación mientras se está utilizando. Antes de entrar en detalle en estos motivos, definiremos cada uno de los ciclos de vida por los que pasa la aplicación desarrollada. [25].

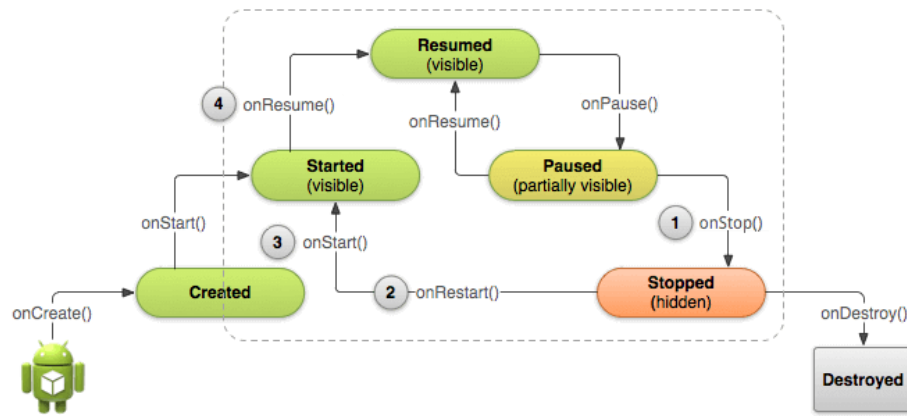


Ilustración 64 Ciclos de vida Android

- **onCreate:** Es invocado en el momento de abrir la actividad nada más instalarse o reiniciarse el terminal o dispositivo móvil.
- **onStart:** Es importante no confundir este método con onCreate, este método es invocado en el momento de abrir la actividad una vez ya abierta anteriormente.
- **onRestart:** Es invocado cuando la actividad previamente ha sido parada y desea abrirse de nuevo.
- **onResume:** La aplicación entra en este método en el momento de terminar de cargarse.
- **onPause:** El sistema realiza actividades simultaneas y deja en un segundo papel la aplicación sin llegar a cambiar la vista del usuario.
- **onStop:** La aplicación pase a estar en segundo plano.
- **onDestroy:** La aplicación es destruida y liberada su memoria.

Una vez conocidos todos los métodos, ha sido importante centrarnos en el método onStop, ya que mientras el usuario se encuentra realizando actividades es muy importante que pueda seguir utilizando el dispositivo móvil con el que esta trabajando, de este modo nunca perderemos ningún dato de la actividad.

- **Firestore**

Firestore es una plataforma de desarrollo en la nube proporcionada por Google. Firestore nos proporciona una API para guardar y sincronizar los datos en la nube en tiempo real. Pese a tener una amplia gama de servicios, en este trabajo fin de máster, solo se ha utilizado para enviar notificaciones push a los dispositivos móviles, ahorrándonos el tener que fabricar un servicio GCM para dicho fin.

Firestore nos permite crear TOPICS (Una serie de temas personalizados) a los que los usuarios pueden suscribirse y recibir alertas cuando se produzca un cambio en el ambiente inteligente que están monitorizando. La estructura de este topic está establecida del siguiente modo.

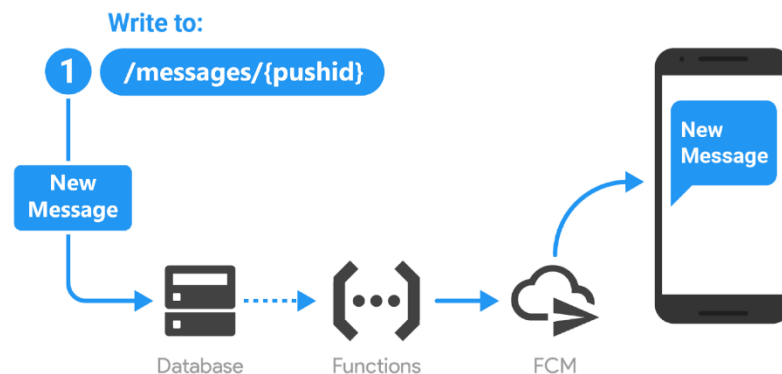


Ilustración 65 Topic en firestore

De este modo, podemos realizar llamadas a la API proporcionada por firestore, y los dispositivos suscritos al topic recibirán el mensaje concreto.

En la aplicación móvil, cada tema estará asociado a un objeto único, el cual recibirá mediante el servidor eventos proporcionados por el sensor. Finalmente, estos eventos serán recibidos en cada uno de los dispositivos que suscritos al tema.

Para que el servidor envíe los eventos a firestore, se ha definido un servicio REST, el cual se encarga de enviar la información automáticamente en cada evento.

Dicho servicio envía la siguiente información:

URL: <https://fcm.googleapis.com/fcm/send>

Método: POST

Cuerpo:

```

1 {
2   "data": {
3     "title": "Puerta abierta",
4     "detail": "Dormitorio dani: La puerta de la habitación ha sido abierta"
5   },
6   "to" : "/topics/94AE"
7 }

```

En el cuerpo del mensaje podemos identificar tres puntos clave en el envío de información al dispositivo móvil.

Title: Título de la notificación.

Detail: Texto largo de la notificación.

To: Tema, 94AE, Identificador único del objeto que ha producido el evento.

4.4.3 Servicios

Todos los servicios implementados en este trabajo fin de máster son proporcionados mediante una API y el protocolo HTTP. Para identificar el estado de la petición que hacemos a cada uno de estos servicios es importante conocer los códigos de respuesta posibles que podemos obtener al hacer uso de ellos.

Código de estado	Descripción
200	Apropiado
201	Datos creados
304	Datos no modificados
400	Petición anómala
401	No autorizado para usar el servicio
403	Sesión expirada
404	Petición inexistente
422	Modelo no procesado
500	Error del servidor

Tabla 24 Códigos de estado

Como se ha descrito anteriormente, una de las partes de este trabajo fin de grado ha consistido en ampliar la funcionalidad de un trabajo realizado por el mismo autor en el anterior trabajo fin de grado. En esta memoria, tan solo se podrán algunos de los servicios ampliados más importantes.

- **Identificarse como habitante**
 - **Descripción:** Permite a los usuarios identificarse desde un medio externo. Utilizado en la aplicación móvil desarrollada.
 - **URL:** /services/login
 - **Método:** POST
 - **Parámetros:**

- **Crear un habitante**
 - **Descripción:** Crea un nuevo habitante en el sistema. Este habitante será asociado a un ambiente que esté previamente dado de alta.
 - **URL:** /services/login
 - **Método:** POST
 - **Parámetros:** user, password (Cifrada SHA1)

- **Crear un usuario**
 - **Descripción:** Crea un nuevo usuario, este puede estar asociado a un solo ambiente o tener rol de administrador y gestionar todos los ambientes del sistema.
 - **URL:** /services/login/newuser
 - **Método:** POST
 - **Parámetros:** user,password,rol,environments(ids)

- **Crear una actividad**
 - **Descripción:** Permite a un usuario o editor de un ambiente dar de alta nuevas actividades. Estas actividades estarán asociadas a un solo ambiente.
 - **URL:** /services/activity/newactivity
 - **Método:** POST
 - **Parámetros:** name,description,idenvironment

- **Iniciar actividad**
 - **Descripción:** Desde un dispositivo móvil, envía la información necesaria proporcionada mediante el escaneo de una etiqueta NFC.
 - **URL:** /services/activity/initactivity
 - **Método:** POST
 - **Parámetros:** nfctag,i dhabitant, idenvironment, idactivity, idstart, idevent

- **Consultar actividades en tiempo real**
 - **Descripción:** Devuelve el estado de todas las actividades asociadas a un ambiente inteligente.
 - **URL:** /services/activity/currentstatus
 - **Método:** GET
 - **Parámetros:** -

- **Consultar histórico entre dos fechas**

- **Descripción:** Devuelve el estado de cada una de las actividades enviadas dentro de una ventana de tiempo establecida entre dos fechas.
- **URL:** /services/activity/datehistory
- **Método:** POST
- **Parámetros:** idenvironment, idActivities, dateInit, dateEnd

Todos los parámetros necesarios han de ser encapsulados en formato de texto plano JSON, ya que la API está configurada para que el mimetype sea en este formato.

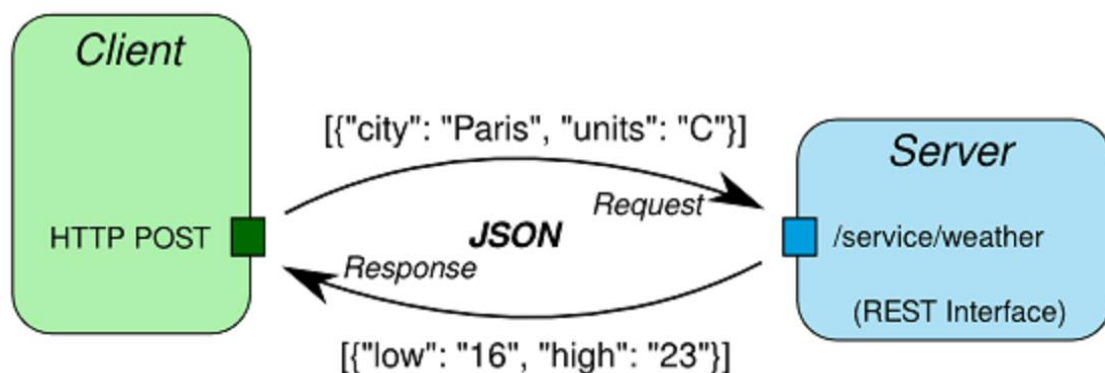


Ilustración 66 Ejemplo servicio REST - JSON

4.4.4 Herramientas de desarrollo

Para llevar a cabo toda la implementación de este trabajo fin de grado se han utilizado las siguientes herramientas de desarrollo software.

4.4.4.1 IntelliJ IDEA



Ilustración 67 Logo IJ

IntelliJ es un IDE (Ambiente de desarrollo) diseñado por la empresa JetBrains para el desarrollo de software en el lenguaje de programación JAVA. Actualmente IntelliJ soporta una gran cantidad de plugin que posibilitan trabajar con diferentes lenguajes o framework elaborados por la propia empresa y una gran comunidad.

4.4.4.2 Android Studio



Ilustración 68 Logo AS

Android Studio es el IDE oficial para desarrollar en el sistema operativo Android. Este IDE se basa principalmente en el descrito anteriormente IntelliJ. Está disponibles para Windows MacOS y Linux, cuenta con una licencia Apache 2.0 la cual hace que se gratuito de utilizar.

4.4.4.3 Atom



Ilustración 69 Logo Atom

Atom es un editor de código fuente desarrollado por la empresa GitHub. Actualmente es uno de los editores más utilizados para el desarrollo de aplicaciones web, ya que cuenta con una gran cantidad de paquetes que facilitan el trabajo del desarrollador.

4.4.4.4 MySQL Workbench



Ilustración 70 Logo Workbench

Workbench es una herramienta de diseño y administración de base de datos MySQL, cuenta con una versión opensource y comercial. Además de estar disponible para cualquier sistema operativo.

Capítulo 5

5 CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS

Tras la finalización de este trabajo fin de máster, es de destacar la grata oportunidad de poner en práctica cada uno de los conocimientos adquiridos en cada una de las asignaturas impartidas a lo largo del máster, tanto en la gestión de proyectos como en métodos y tecnologías de desarrollo software.

Revisando todo el trabajo realizado durante los meses de desarrollo, como conclusión técnica personal he de realizar hincapié en el trabajo mediante servicios, ya que a la hora de aumentar las funcionalidades de un proyecto personal anteriormente desarrollado como al conectar nuevos dispositivos y aplicaciones ha supuesto un gran ahorro de trabajo, facilitando a su vez las conexiones y aumentando la seguridad en ellas.

Por otro lado, en cuanto a los objetivos, una vez finalizado todo el trabajo y realizando una valoración justa, se puede dar por cumplidos dentro del plazo estimado de tiempo gestionando las posibles variaciones de tiempo de una forma rápida y concisa.

Es cierto que una vez finalizado todo el trabajo fin de máster, se pueden apreciar un gran abanico de posibilidades donde distintas líneas de trabajo futuras hacen que se pueda mejorar tanto el funcionamiento como la funcionalidad de la aplicación desarrollada.

Entre las principales funcionales implementadas, destaca el posicionamiento de un habitante en un entorno inteligente gracias a los datos recolectados en este trabajo por las balizas bluetooth o la anotación de actividades por diferentes usuarios, su traspaso al middleware con el que cuenta el laboratorio de inteligencia ambiental de la Universidad de Jaén y la explotación de datos para crear dataset los cuales pueden ser estudiados desde aplicaciones externas, en las que además como líneas de trabajo en el futuro en esta área, se podrían incluir métodos para realizar la detección automática de actividades o detección de anomalías en las actividades que se estén realizando.

En cuanto a la tecnología, una gran mejora es desarrollar el etiquetado de actividades en dispositivos wereables para de este modo realizar de forma automática el etiquetado de las actividades. Como trabajo futuro en esta área, se podría dar soporte a otros sistemas operativos móviles como iOS o Windows Phone, los cuales no tienen tanta repercusión en el mercado, pero de este modo se pude abarcar a un público más amplio.

En definitiva, gracias a todo lo aprendido en este proyecto, se ha abierto un nuevo camino en mi etapa profesional donde podré seguir aprendiendo y formándome sobre los ambientes inteligentes.

Capítulo 6

6 BIBLIOGRAFÍA

- [1] M. E. J. M. D. Zafra, Sistema para la monitorización de Ambientes inteligentes a través de la gestión y acceso a servicios, 2016.
- [2] «UJAml,» [En línea]. Available: <http://ceatic.ujaen.es/ujami/en>.
- [3] K. & B. M. Schwaber, Agile software development with Scrum, 2002.
- [4] Y. Benkler, El Pingüino y el Leviatán: Por qué la cooperación es nuestra arma más valiosa para mejorar el bienestar de la sociedad, 2012.
- [5] «International Data Base U.S. Census Bureau,» 2017. [En línea]. Available: <http://www.census.gov/ipc/www/idbpyr.html>.
- [6] J. R. G. J. J. R. I. a. C. S. M. Pérez, «Telemedicina,» de *Ingeniería biomédica*, 56, 2009.
- [7] «UJAMI LAB,» 2018. [En línea]. Available: <http://ceatic.ujaen.es/ujami/>. [Último acceso: 2018].
- [8] A. & L.-M. I. Anaya-Cantellán, «La tecnología NFC en teléfonos celulares, sus retos y aplicaciones.,» *Advances in Artificial Intelligence and its Intelligent Applications*, vol. 97.
- [9] D. M.-F. A. d. C.-P. H. G.-V. S. M. J. & L. D. R. Barba, «Uso de Bluetooth de Baja Energía en Aplicaciones de Localización Indoor.».
- [10] S. M. T. U. T. T. I. & M. H. Kajioka, «Experiment of indoor position presumption based on RSSI of Bluetooth LE beacon.,» *Consumer Electronics*, 2014.
- [11] FCBarcelona, «<https://www.fcbarcelona.es/club/noticias/2015-2016/el-fc-barcelona-despliega-una-red-de-tecnologia-beacons>,» 2018. [En línea].
- [12] «Online Shopping - UML Class Diagram Example,» [En línea]. Available: <http://www.uml-diagrams.org/examples/online-shopping-domain-uml-diagram-example.html?context=cls-examples>. [Último acceso: 2018].
- [13] I. B. G. & R. J. Jacobson, El proceso unificado de desarrollo de software/The unified software development process, 2000.
- [14] V. & B. J. Abramova, «NoSQL databases: MongoDB vs cassandra. In Proceedings of the international C* conference on computer science and software engineering,» pp. 14-22.
- [15] G. INC. [En línea]. Available: <https://material.io/guidelines/>. [Último acceso: 2018].
- [16] <https://material.io/icons/>, «Iconos Material Design,» [En línea]. Available: 2018.

- [17] H. M. & D. P. J. Deitel, Como programar en Java, Pearson Educación..
- [18] «Oracle adquiere Sun Microsystems por 5.710 millones,» El Pais, 2009. [En línea]. Available:
https://elpais.com/tecnologia/2009/04/20/actualidad/1240216080_850215.html.
- [19] R. Santamaría, «REST avanzado,» [En línea]. Available:
<http://vis.usal.es/rodrigo/documentos/soa/REST%20avanzado.pdf>.
- [20] J. & E. A. Melton, SQL y Java: guía para SQLJ, JDBC y tecnologías relacionadas., Ra-Ma, 2002.
- [21] C. Aubry, HTML5 y CSS3-Revolucione el diseño de sus sitios web., ENI, 2012.
- [22] S. & P.-A. G. U. R. Y. Ollivier, AngularJS: Desarrolle hoy las aplicaciones web de mañana., ENI, 2016.
- [23] «Cuota de mercado Android,» muycomputer, [En línea]. Available:
<https://www.muycomputer.com/2017/12/12/android-oreo-cuota-mercado/amp/>.
- [24] «¿Qué versión de Android es la más usada en 2018?,» 2018. [En línea]. Available:
https://as.com/betech/2018/02/06/portada/1517907941_631405.html.
- [25] J. T. Gironés, El gran libro de Android, Marcombo, 2012.
- [26] J. G. Velasco, Energías renovables, Editorial Reverte, 2009.
- [27] A. S. a. S. Wright, Intelligent Spaces: The Application of Pervasive, Springer-Verlag, 2006.

Capítulo 7

7 ANEXOS

7.1 CONTENIDO DEL CD-ROM

En el presente anexo se visualiza en la ilustración inferior el contenido del CD-ROM adjunto en esta memoria.






Nombre	Fecha de modifica...	Tipo	Tamaño
 Aplicaciones Finales	23/06/2018 13:39	Carpeta de archivos	
 Base de datos	23/06/2018 13:39	Carpeta de archivos	
 Código Fuente	23/06/2018 13:39	Carpeta de archivos	
 Demostración.mp4	04/07/2016 17:56	VLC media file (.m	125.921 KB
 Memoria.pdf	04/07/2016 17:56	Documento Adob...	4.634 KB

Ilustración 71 Estructura de la carpeta

- **Aplicaciones finales**
 - Cliente HTML
 - Cliente APK Android
 - Servidor .WAR Tomcat 8
- **Base de datos**
 - Fichero .sql el cual contiene una copia actual de la base de datos del sistema web.
- **Código de fuente**
 - Código fuente de todos los proyectos elaborados en este trabajo fin de máster. Cada uno de estos proyectos será abierto con IntelliJ y Atom.
- **Memoria PDF**
- **Video ilustrativo**

7.2 MANUAL DE INSTALACIÓN

A continuación, se detalla como instalar cada una de las aplicaciones adjuntadas en el CD-ROM dentro de la carpeta “Aplicaciones finales”.

En este manual se ha utilizado el sistema operativo Linux apoyándonos en la conocida distribución Debian 9.

7.2.1 Servidor

Antes de comenzar con el despliegue de la aplicación desarrollada, debemos de realizar la instalación del siguiente software en nuestro sistema.

7.2.1.1 Java 8

Como se ha descrito en el capítulo 4 de Ingeniería del Software, la aplicación servidor se ha desarrollado con un framework de Java, por lo que es de vital importancia que este, esté instalado en el sistema.

Para ello, abrimos una terminal en el servidor y tecleamos la siguiente sentencia.

```
apt install oracle-java8-installer
```

Al acabar la instalación, podemos comprobar si tenemos java funcionando en nuestro sistema escribiendo el siguiente comando.

```
java -version
```

```
[root@dazaro ~]# java -version
java version "1.8.0_131"
Java(TM) SE Runtime Environment (build 1.8.0_131-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.131-b11, mixed mode)
[root@dazaro ~]# █
```

Ilustración 72 Comprobación de Java. 1.8

7.2.1.2 MariaDB

En cuanto a la persistencia de todos los conjuntos de datos hemos utilizado una base de datos relacional SQL. Para ello, hemos utilizado MariaDB, ya que se puede instalar en cualquier sistema operativo y cuenta con una licencia de uso gratuita.

Para llevar a cabo la instalación debemos abrir la consola del sistema y teclear la siguiente cadena.

```
apt install mariadb-server mariadb-client
```

Uno de los pasos más importantes en la instalación es introducir la contraseña de administrador para poder realizar las futuras conexiones o crear nuevos usuarios.



Ilustración 73 Contraseña MariaDB

Del mismo modo que hicimos con la instalación de Java, podemos comprobar que la instalación ha sido satisfactoria con el siguiente comando.

```
mysql -V
```

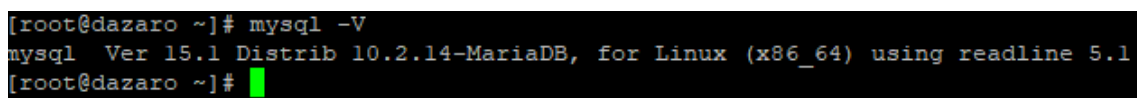


Ilustración 74 Comprobación de MariaDB

7.2.1.3 Tomcat 8

Por último, tan solo nos queda instalar Tomcat 8 para poder desplegar la aplicación servidor desarrollada. Del mismo modo que en los puntos anteriores tecleamos el siguiente comando:

```
apt install tomcat8 apt-get install tomcat8-admin
```

Tran finalizar la instalación debemos configurar un usuario y contraseña para identificarnos en el panel de administración y desplegar aplicaciones. Esto se puede hacer de una manera rápida y sencilla utilizando el editor de textos nano.

```
nano /etc/tomcat8/tomcat-users.xml
```

Al final del fichero abierto se debe añadir la siguiente línea

```
<tomcat-users>
```

```
<user username="USUARIO" password="CONTRASEÑA"  
roles="manager-gui,admin-gui"/>
```

```
</tomcat-users>
```

Una vez añadidas, reiniciamos el servicio Tomcat.

```
service tomcat8 restart
```

Si llegado a este punto no encontramos ningún error, ya tenemos operativo todo el software requerido para que funcione nuestra aplicación servidora desarrollada en este trabajo.

En el CDROM Anexo, encontraremos el archivo compilado dentro de Aplicaciones Finales. Para desplegarlo en tomcat se ha de entrar en la siguiente dirección web.

<http://servidor:8080/manager/html>

Tras introducir las credenciales configuradas en el punto anterior obtenemos la siguiente vista.

The screenshot shows the Tomcat Manager interface. At the top, there are logos for 'The Apache Software Foundation' and 'Tomcat'. The main heading is 'Gestor de Aplicaciones Web de Tomcat'. Below this, there is a message box and navigation links: 'Listar Aplicaciones', 'Ayuda HTML de Gestor', 'Ayuda de Gestor', and 'Estado de Servidor'. The 'Aplicaciones' section contains a table with columns: Trayectoria, Versión, Nombre a Mostrar, Ejecutándose, Sesiones, and Comandos. The table lists applications like 'manager', 'host-manager', and 'docs'. Below the table, there are sections for 'Desplegar' (Deploy) and 'Archivo WAR a desplegar' (WAR file to deploy). The 'Archivo WAR a desplegar' section has a red circle around the 'Seleccionar archivo' button and the text 'Ningún archivo seleccionado'.

Ilustración 75 Panel de control Tomcat

Seleccionamos el fichero .war ubicado en el CDROM y desplegamos la aplicación. De modo que se visualizará como en la imagen inferior.

/SmartLab-2.0	Ninguno especificado	Restful Web Application	true	0	<input type="button" value="Arrancar"/> <input type="button" value="Parar"/> <input type="button" value="Recargar"/> <input type="button" value="Replegar"/> <input type="button" value="Expirar sesiones"/> sin trabajar > 30 minutos
---------------	----------------------	-------------------------	------	---	---

Ilustración 76 Aplicación desplegada

Una vez desplegada, ya podemos acceder a todos los servicios que ofrece nuestra aplicación mediante la dirección <http://servidor:8080/SmartLab2.0/>

7.2.2 Cliente

Para desplegar la aplicación cliente, es necesario tener un servidor web donde se quiera desplegar la aplicación. Por facilidad de uso y licencia en este manual vamos a utilizar apache2. Para su instalación tan solo debemos teclear en un terminal la siguiente sentencia.

```
apt install apache2
```

Una vez instalado, el despliegue es muy simple ya que tan solo debemos copiar la carpeta Cliente la cual se encuentra dentro del CDROM anexo a esta memoria dentro de la carpeta /var/www/html del servidor donde hemos realizado la instalación de apache.

Una vez copiado todo el contenido si tecleamos la dirección de dicho servidor en un navegador nos encontramos con la siguiente aplicación desplegada.



Ilustración 77 Vista principal de la aplicación

7.2.3 Aplicación móvil

Para instalar la aplicación móvil es necesario disponer de un dispositivo Android, el cual cuente con tecnología NFC. Para instalar dicha aplicación en un dispositivo, tan solo debemos de ejecutar el fichero .apk ubicado dentro de la carpeta aplicaciones finales. Al instalarlo es importante otorgar al terminal de permisos para instalar desde fuentes desconocidas, ya que actualmente no tenemos la aplicación subida a Play Store.

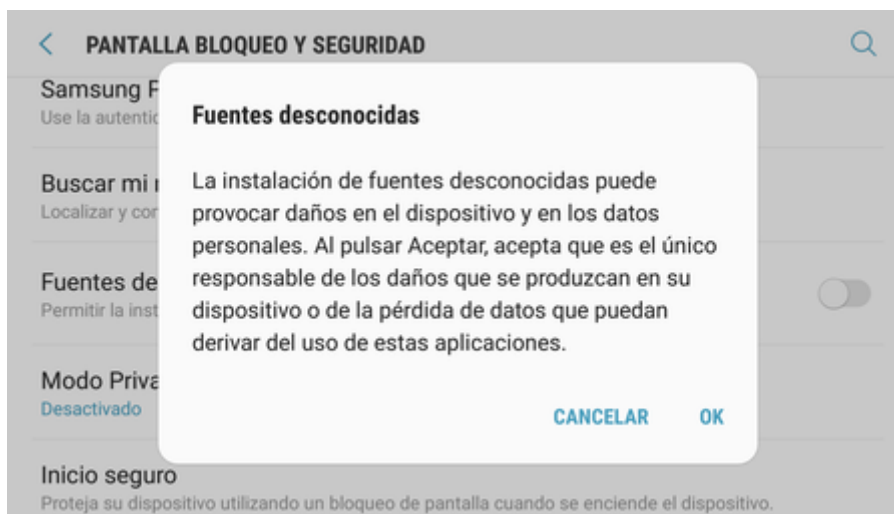


Ilustración 78 Instalar dese fuentes desconocidas

Tras finalizar la instalación ya podemos hacer uso de la aplicación móvil.

7.3 MANUAL DE USO

A continuación, en los siguientes puntos se detallará como se utiliza cada una de las funcionalidades ampliadas en este trabajo.

7.3.1 Portal de usuario



Ilustración 79 Inicio de la aplicación web

La primera vista que encontramos al acceder a la aplicación es un panel simple donde podemos monitorizar cualquier ambiente inteligente que se encuentre dado de alta en el sistema o entrar al panel de administración.

A continuación, detallaremos todos los movimientos posibles los cuales el usuario puede realizar dentro del panel de usuario.

7.3.1.1 Consultar estado actual

Nada más pulsar en uno de los ambientes, el usuario puede consultar de una manera rápida y sencilla como se encuentra cada uno de los sensores que tiene desplegados en el ambiente.

UJAml Monitoring Estado Actual Histórico Eventos Actividades Video Administración Contacto

Información del ambiente

Nombre: Smart Lab
 Fecha alta ambiente: 2015-10-25
 Fecha primer evento: 2015-09-02
 Fecha último evento: 2018-05-17

Últimos valores Actualizar

M01 TV0 D01 D02 D03

Nota: Puedes consultar el estado de cada sensor dejando el cursor sobre el objeto.

Mapa Ver Eventos (Modo texto)

Leyenda

Seleccionar todos

Ocultar	Código	Descripción
<input type="checkbox"/>	M01	Door
<input type="checkbox"/>	TV0	TV
<input type="checkbox"/>	D01	Refrigerator
<input type="checkbox"/>	D02	Microwave
<input type="checkbox"/>	D03	Wardrobe clothes
<input type="checkbox"/>	SM1	Sensor kitchen novent
<input type="checkbox"/>	M02	Bedroom window
<input type="checkbox"/>	L01	Hall light

Ilustración 80 Estado actual de un ambiente

En cuanto a funcionalidades extra, el usuario puede ocultar los sensores que desee para obtener una visualización mas minuciosa de los sensores que les interesan en el momento de la consulta.

Por otro lado, se han añadido los 5 últimos valores en la cabecera de la página, de este modo el usuario podrá saber cuales son los últimos cambios que se están produciendo dentro de un ambiente.

7.3.1.2 Consultar actividades en tiempo real

Del mismo modo que en el punto anterior podíamos visualizar el estado de cada uno de los sensores, en la misma vista se mostrará información en tiempo real de las actividades que se encuentran realizando cada uno de los habitantes el ambiente que estamos monitorizando.

Recordando la funcionalidad de la aplicación móvil, el usuario podría etiquetar una actividad escaneando balizas beacon o sin ellas. En las imágenes inferiores podemos ver como se muestra la información de las actividades haciendo uso de la tecnología NFC y Beacons.

- Sin escaneo de beacon

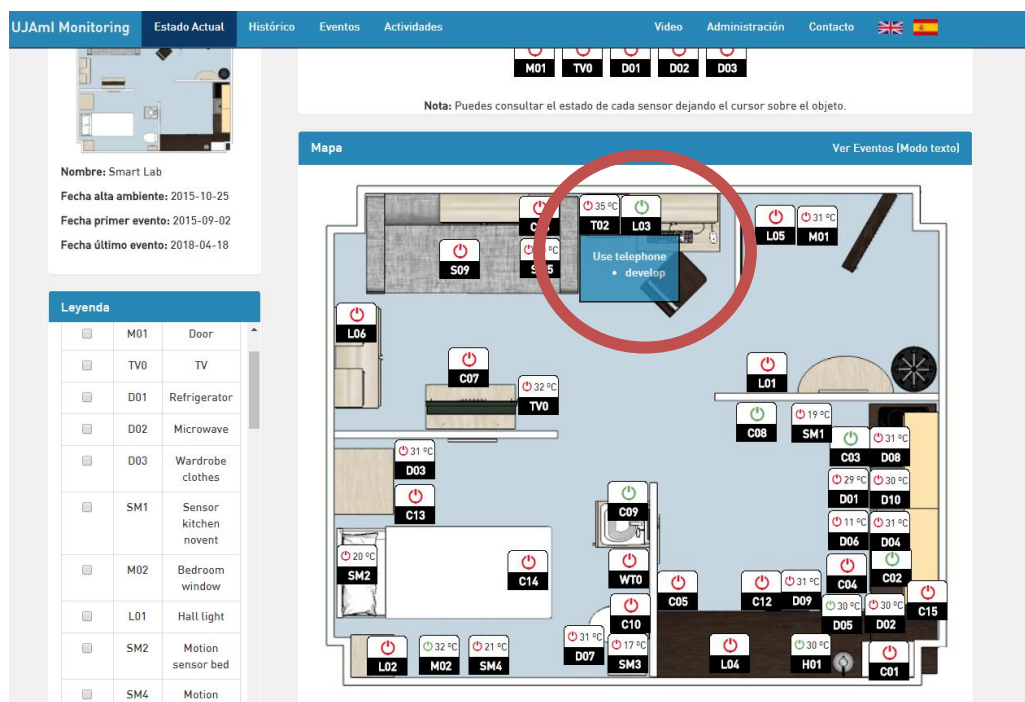


Ilustración 81 Actividad sin beacon

- **Con escaneo de beacons**

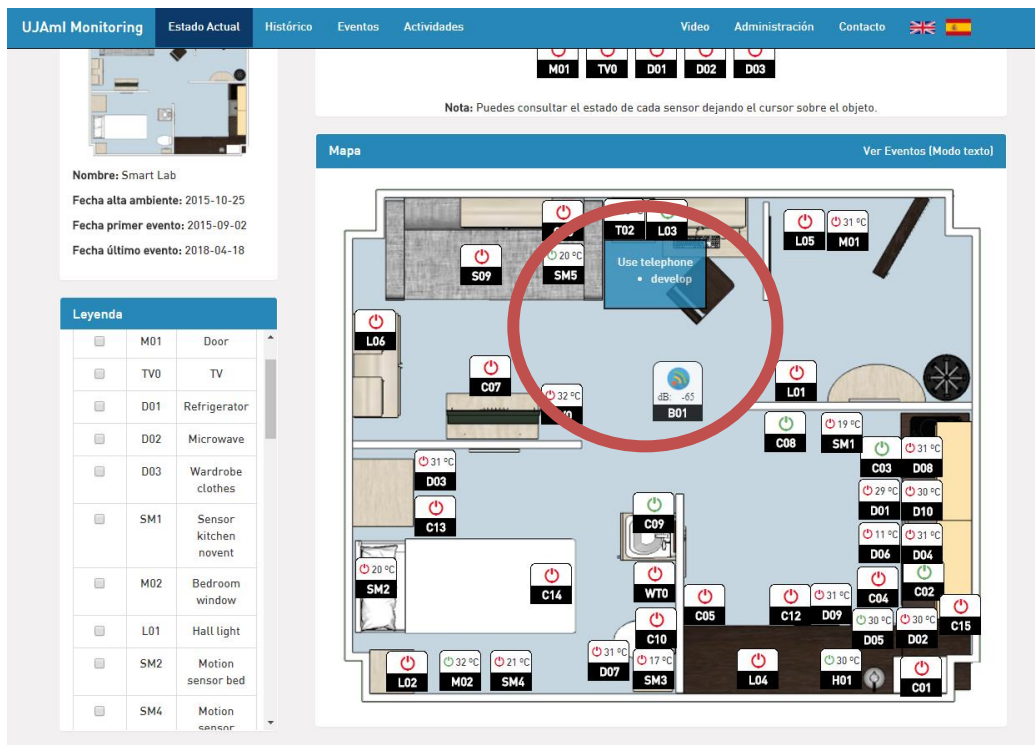


Ilustración 82 Actividad con beacon

Como se puede apreciar en ambas imágenes, en cada uno de los dos modos, mostramos información de la actividad, indicando el nombre de ella y del habitante que se encuentra realizándola.

En cuanto a la parte de escaneo de balizas, el usuario podrá visualizar en todo momento el RSSI de cada baliza la cual se irá actualizando de forma transparente para el usuario.

7.3.1.3 Actividades

Del mismo modo que el usuario podía consultar todos los eventos en modo texto, se ha realizado una vista similar donde el usuario puede realizar consultas por tiempo de cada una de las actividades.

The screenshot shows the 'Actividades' section of the UJAml Monitoring interface. The top navigation bar includes 'UJAml Monitoring', 'Estado Actual', 'Histórico', 'Eventos', 'Actividades', 'Video', 'Administración', 'Contacto', and language flags. The left sidebar contains 'Información del ambiente' with a floor plan and details for 'Smart Lab' (Nombre: Smart Lab, Fecha alta ambiente: 2015-10-25, Fecha primer evento: 2015-09-02, Fecha último evento: 2018-04-18), and a 'Histórico' section with a frequency of 29 and a list of activities to consult, all of which are checked.

The main content area displays a table of activity history with the following data:

Actividad	Fecha Inicio	Fecha Fin	Habitante	Rango
Breakfast	2018-02-05 11:19:20	2018-02-05 11:22:01	Manuel	Ver Eventos
Brush your teeth	2018-02-05 17:20:01	2018-02-05 17:20:50		Ver Eventos
Cooking	2016-05-25 10:25:51	2016-05-25 10:26:45	Amy	Ver Eventos
Dinner	2018-02-05 17:15:46	2018-02-05 17:19:40		Ver Eventos
Do the dishes	2018-02-01 13:35:51	2018-02-01 13:37:27	Manuel	Ver Eventos
Get cold drink	2016-06-02 17:32:08	2016-06-02 17:32:48	Colin Shewell	Ver Eventos
Get dressed	2018-02-05 17:36:51	2018-02-05 17:37:55		Ver Eventos
Get hot drink	2016-06-02 17:28:35	2016-06-02 17:29:06	Colin Shewell	Ver Eventos
Get up from bed	2018-02-05 11:10:15	2018-02-05 11:11:59		Ver Eventos
Go home	2018-02-05 11:16:05	2018-02-05 11:16:43	Manuel	Ver Eventos
Go to bed	2018-02-05 17:38:06	2018-02-05 17:39:28		Ver Eventos
Leave House	2018-02-05 17:31:59	2018-02-05 17:32:34	Manuel	Ver Eventos
Lunch	2018-02-05 13:00:49	2018-02-05 13:07:13		Ver Eventos
Play a videogame	2018-02-07 16:26:21	2018-02-07 16:30:35	Manuel	Ver Eventos
Prepare Breakfast	2018-02-05 11:17:53	2018-02-05 11:18:55	Manuel	Ver Eventos
Prepare Dinner	2018-02-07 16:28:53	2018-02-07 16:30:12		Ver Eventos
Prepare lunch	2018-02-05 12:56:00	2018-02-05 12:59:55		Ver Eventos
Put washing machine	2018-02-01 13:36:51	2018-02-01 13:37:47	Alexa	Ver Eventos
Read a book	2018-02-01 13:03:07	2018-02-01 13:20:22	Alexa	Ver Eventos

Ilustración 83 Lista de actividades en modo texto

En cuanto a la búsqueda de actividades el usuario dispone de los siguientes filtros.

Tipo de consulta:
Fecha: Tiempo: Cambios:

Desde:

Hasta:

Frecuencia:
Actividades que quieres consultar:

Seleccionar todos

Cooking

Watch TV

Go to bed

Use toilet

Prepare Breakfast

Prepare Dinner

Ilustración 84 Filtros de consulta

7.3.1.4 Exportar actividades

Dentro de la vista de actividades, el usuario tiene la posibilidad de exportar todas las actividades que se encuentran almacenadas en la aplicación para un posterior estudio o análisis externo de estas. Para ello, el usuario puede pulsar el botón Exportar situado en el margen superior derecho de la ventana o seleccionado directamente el tipo de exportación que quiere dentro de este botón.



Ilustración 85 Menú rápido de exportación de actividades.

En cambio, si pulsamos el botón exportar, obtenemos la siguiente vista para seleccionar que tipo de actividad queremos exportar.

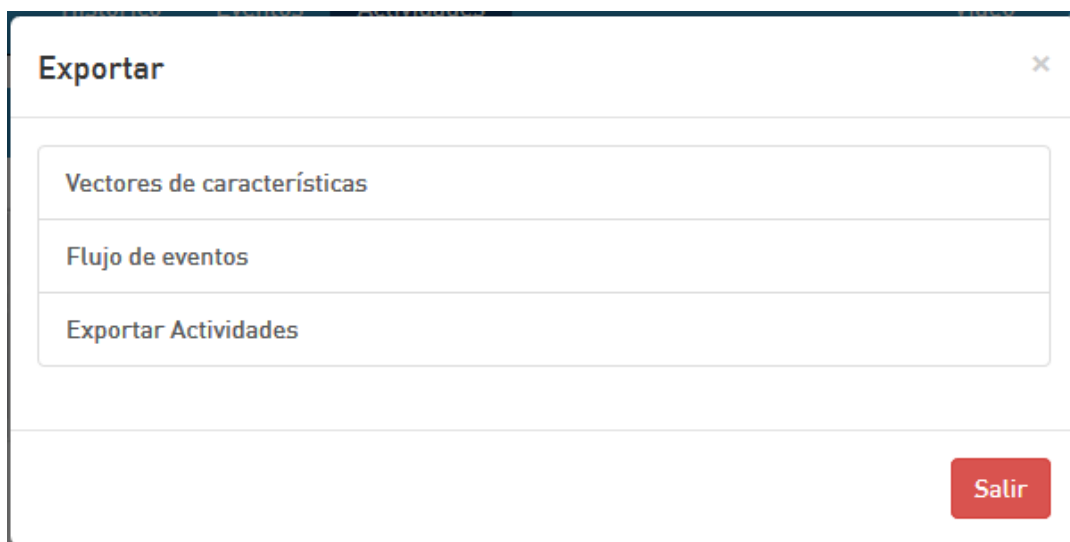


Ilustración 86 Vista modal para exportar actividades

7.3.1.5 Vector de características

Al exportar el vector de características de las actividades consultadas, el sistema genera un archivo en forma de matriz donde las filas son las actividades y las columnas cada uno de los sensores que se han manipulado mientras se realizaban cada una de ellas. Dentro de la configuración del fichero, podemos indicar si queremos mostrar los habitantes que se encontraban realizando dichas actividades o si queremos codificarlo las repeticiones de las actividades de forma numérica o binaria.

Ilustración 87 Configuración del fichero de salida

Una vez exportado, si abrimos con un editor de calculo el fichero CSV podemos ver la matriz generada por el sistema.

	AP	AQ	AR	AS	
	S09	B01	B99	Activity	
0	0	0	106	Leave House	
0	0	22	0	Use telephone	
0	0	20	0	Use telephone	
0	0	0	0	Use telephone	

Ilustración 88 Ejemplo de fichero CSV

7.3.1.6 Flujo de eventos

El flujo de eventos nos muestra todos los eventos que se han producido mientras se realiza una actividad. Este flujo puede ser configurado mostrando información del habitante que se encontraba realizando dicha actividad o mostrando su nombre.

Toda la información estará ordenada en el tiempo y agrupada por actividades.

The screenshot shows a configuration window titled 'Flujo de eventos'. It has two main sections: 'Formato de datos' and 'Archivo'. In 'Formato de datos', there are two checkboxes: 'Actividades' (checked) and 'Habitante' (unchecked). In 'Archivo', there are two radio buttons: 'TXT' (selected) and 'CSV' (unselected). Below these sections is a blue 'Exportar' button. At the bottom right of the window is a red 'Salir' button.

Ilustración 89 Configuración del fichero de salida

Obteniendo el siguiente fichero

	A	B	C	D	E
1	Date	Object	State	Activity	Habitant
2	2018-04-18 17:41:23.0	SM5	Movement	Use telephone	develop
3	2018-04-18 17:41:58.0	SM5	No movement	Use telephone	develop
4	2018-04-18 17:41:59.0	SM5	Movement	Use telephone	develop
5	2018-04-18 17:42:19.0	SM5	No movement	Use telephone	develop
6	2018-04-18 17:42:20.0	SM5	Movement	Use telephone	develop
7	2018-04-18 17:42:40.0	SM5	No movement	Use telephone	develop
8	2018-04-18 17:42:41.0	SM5	Movement	Use telephone	develop
9	2018-04-18 17:43:02.0	SM5	Movement	Use telephone	develop
10	2018-04-18 17:43:02.0	SM5	No movement	Use telephone	develop
11	2018-04-18 17:43:40.0	SM5	No movement	Use telephone	develop
12	2018-04-18 17:43:52.0	SM5	Movement	Use telephone	develop
13	2018-04-18 17:44:01.0	M01	Open	Use telephone	develop
14	2018-04-18 17:44:05.0	SM4	Movement	Use telephone	develop
15	2018-04-18 17:44:18.0	M01	Close	Use telephone	develop
16	2018-04-18 17:44:21.0	SM4	No movement	Use telephone	develop
17	2018-04-18 17:44:22.0	SM5	No movement	Use telephone	develop

Ilustración 90 Flujo de eventos

7.3.1.7 Actividades

Por último, en este punto tan solo generaremos un fichero con las actividades y su marca temporal que se han realizado en la búsqueda seleccionada por el usuario.

	A	B	C	D
1	Activity	Date Init	Date End	Habitant
2	Leave House	2018-04-25 18:35:56.0	2018-04-25 18:40:13.0	develop
3	Use telephone	2018-04-18 17:44:24.0	2018-04-18 17:45:17.0	develop
4	Use telephone	2018-04-18 17:42:15.0	2018-04-18 17:43:17.0	develop
5	Use telephone	2018-04-18 17:40:35.0	2018-04-18 17:41:18.0	develop
6				

Ilustración 91 Exportado de actividades

7.3.2 Administración

Una vez terminado con todas las posibilidades con las que puede interactuar el usuario, nos pasamos al panel de administración.

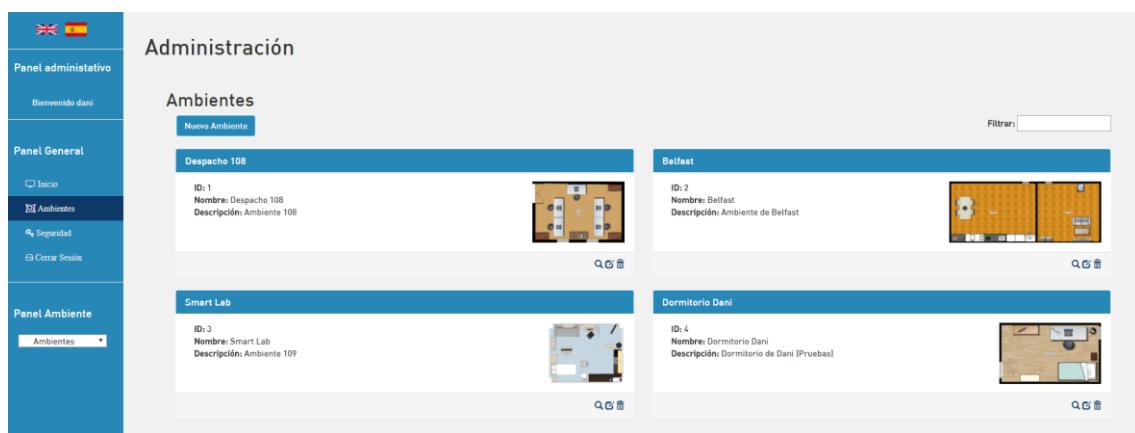


Ilustración 92 Panel de administración

A continuación, detallaremos cuales son las partes en las que el administrador puede realizar acciones.

7.3.2.1 Seguridad

Del modo que comentamos en los objetivos del proyecto, una nueva funcionalidad es poder tener administradores generales de la aplicación los cuales pueden gestionar cualquier ambiente y todo el conjunto del sistema. A su vez, estos superusuarios, tiene la capacidad de generar editores de ambientes, los cuales solo pueden realizar acciones dentro del ambiente o ambientes a los que tengan permisos.



Ilustración 93 Lista de usuarios

En el momento de dar de alta un nuevo usuario, la contraseña será persistida en la base de datos con el algoritmo SHA1, de este modo será difícil poder obtener la contraseña real que el usuario ha tecleado.

6	daniel	b1b3773a05c0ed0176787a4f1574ff0075f7521e	user
7	prueba	711383a59fda05336fd2ccf70c8059d1523eb41a	user

Ilustración 94 Contraseña SHA1

7.3.2.2 Habitantes

Ya que tenemos los administradores dados de alta dentro de un ambiente inteligente o del sistema. Tan solo nos queda dar de alta a los habitantes, estos habitantes tienen la funcionalidad de realizar actividades dentro de un ambiente inteligente a través de la aplicación móvil desarrollada en este trabajo.

The screenshot displays the 'Administración' section of a web application. On the left is a blue sidebar with navigation links: 'Inicio', 'Ambientes', 'Seguridad', 'Crear Sesión', and 'Habitantes'. The main content area is titled 'Administración' and 'Habitantes'. It features a 'Nuevo Habitante' button and a search filter. Below is a table listing existing inhabitants:

ID	Nombre	ID Dispositivo	
22	Manuel	79782e923bf9cb29	
24	develop	aac5626953f8963c	

Ilustración 95 Lista de habitantes

El usuario y contraseña que utilizemos al crear el habitante dentro de un ambiente, será el que debe de utilizar en la aplicación móvil para iniciar sesión y etiquetar actividades.

Del mismo modo que con el panel de seguridad, las contraseñas de todos los habitantes serán cifradas con el algoritmo SHA1 y se asociará el último ID (Identificador) del dispositivo con el que han iniciado sesión en el sistema.

7.3.2.3 Actividades

Por último, para finalmente el usuario puede etiquetar actividades, tan solo nos queda darlas de alta en el sistema y posicionarlas en el mapa para su posterior visualización.

ID	Name	Descripción
20	Dinner	Dinner
1	Cooking	Proceso de cocinar
7	Go to bed	Go to bed
8	Use toilet	Use toilet
9	Prepare Breakfast	Prepare Breakfast
10	Prepare Dinner	Prepare Dinner
11	Leave House	Leave House
12	Get cold drink	Get cold drink
13	Get hot drink	Get hot drink
14	Get dressed	Get dressed
15	Use telephone	Use telephone
16	Take the medicine	Take the medicine
17	Prepare lunch	Make the lunch. 2 types- pasta or omelette.
18	Breakfast	Breakfast
19	Lunch	Lunch

Ilustración 96 Lista de actividades

Una vez dada de alta la aplicación en el sistema, es importante guardar e identificador generado por el sistema, ya que este identificador es el que deberá introducirse en la etiqueta NFC para su posterior lectura desde la aplicación móvil.



Ilustración 97 Etiqueta NFC

- **id:** Identificador NFC, NFCXXXX
 - Es utilizado para detectar que es una etiqueta del sistema.
- **Value:** F (Actividad Fin) S (Actividad Inicio)
- **IdE:** Identificador del ambiente donde reside la etiqueta
- **idA:** Identificador de la actividad generado por el sistema

Finalmente, tan solo debemos de posicionar la actividad en el mapa para su posterior lectura desde el panel de usuario.

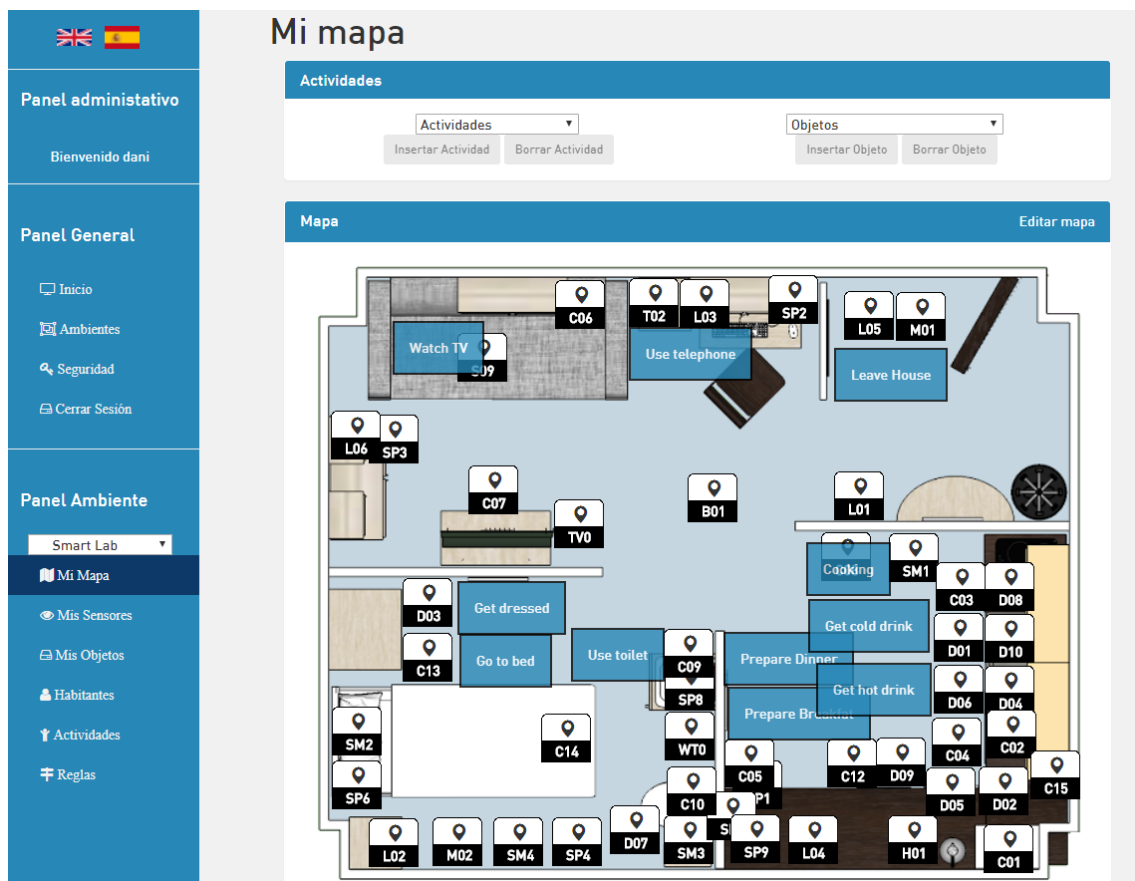


Ilustración 98 Posicionando actividades

7.3.3 Aplicación móvil

A continuación, se muestra el manual de uso de la aplicación móvil.

7.3.3.1 Inicio de sesión

Nada más realizar la instalación, al abrirla nos encontramos con un pequeño formulario en el que introduciremos el usuario y contraseña del habitante. Recordamos que este usuario ha sido generado por un administrador del ambiente dentro del panel de administración web.

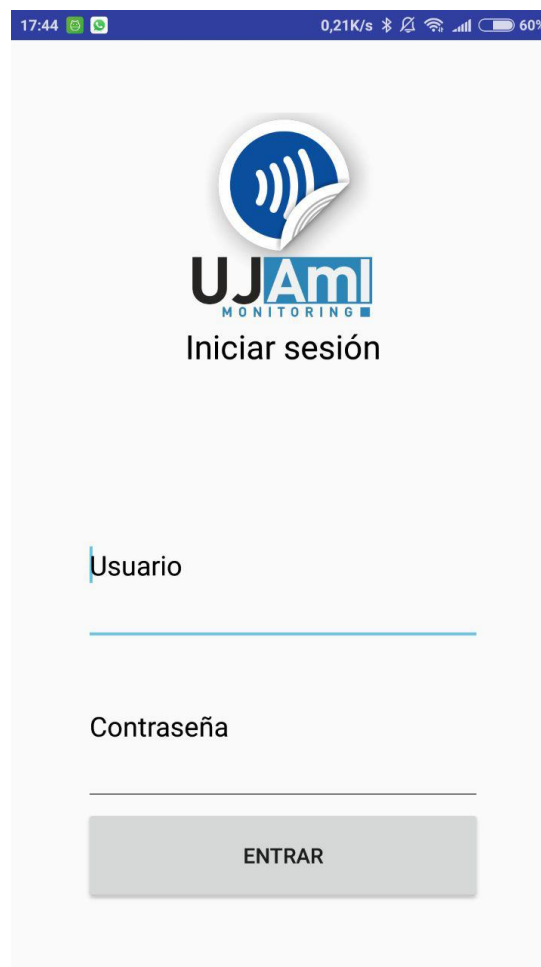
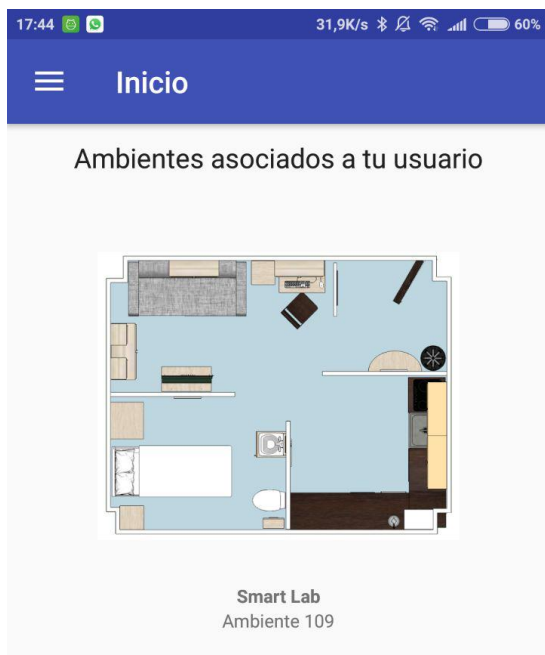


Ilustración 99 Inicio de sesión

Si el inicio de sesión es correcto, visualizaremos el ambiente al que está asociado dicho perfil, en cambio si el usuario no es correcto, la aplicación no nos permitirá logarnos en ella.

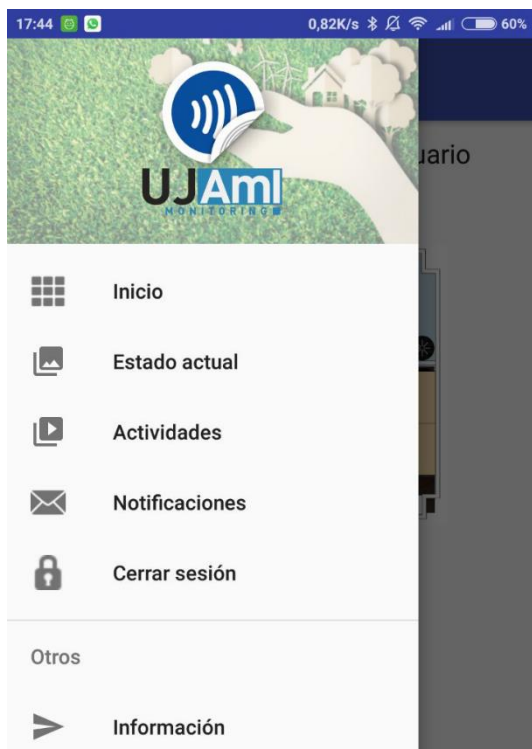
7.3.3.2 Ambientes disponibles



Tras realizar un inicio de sesión satisfactorio, el sistema muestra al usuario de la aplicación el o los ambientes a los que tiene permiso su usuario. Pulsamos con el ambiente que deseamos trabajar y ya podemos utilizar el menú de la aplicación.

Ilustración 100 Lista de ambientes

7.3.3.3 Menú



Dentro del menú de la aplicación, el usuario puede realizar acciones como consultar el estado actual del ambiente, etiquetar actividades, realizar suscripciones a eventos para recibir notificaciones o cerrar la sesión iniciada en el sistema.

Ilustración 101 Menú

7.3.3.4 Consultar estado actual

Al consultar el estado actual de un ambiente ,este mostrará la hora a la que se ha producido el último evento y un icono indicando el estado en e lque se encuentra cada uno de los sensores u objetos asociados al ambiente.

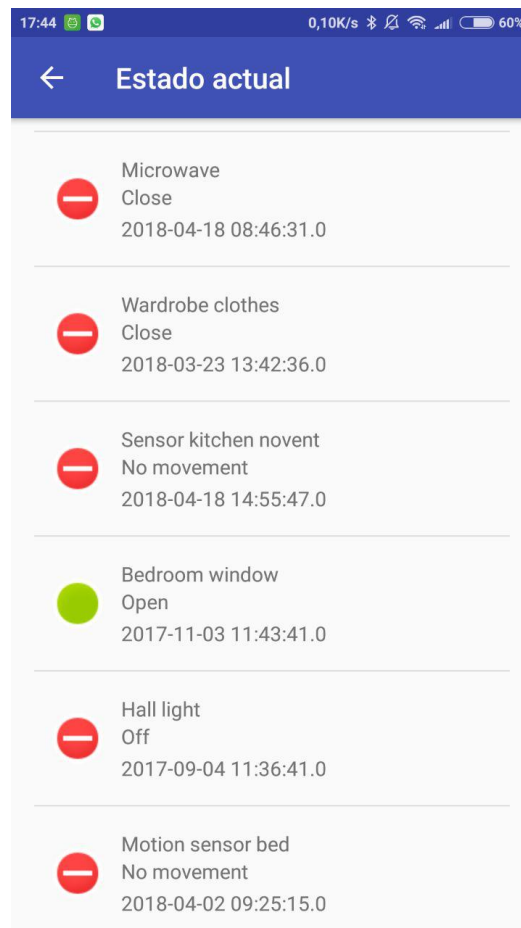


Ilustración 102 Estado actual

Como se detalla dentro de las metáforas de esta memoria. El color rojo se ha utilizado para un estado de cerrado o inactivo, mientras que el color verde ha sido utilizado par un estado de abierto o activo.

7.3.3.5 Etiquetado de actividades

El etiquetado de actividades es uno de los puntos más fuertes de esta aplicación móvil, nada más pulsar el botón de etiquetado de la aplicación, el sistema mostrará la primera ilustración que se puede ver en la imagen inferior.

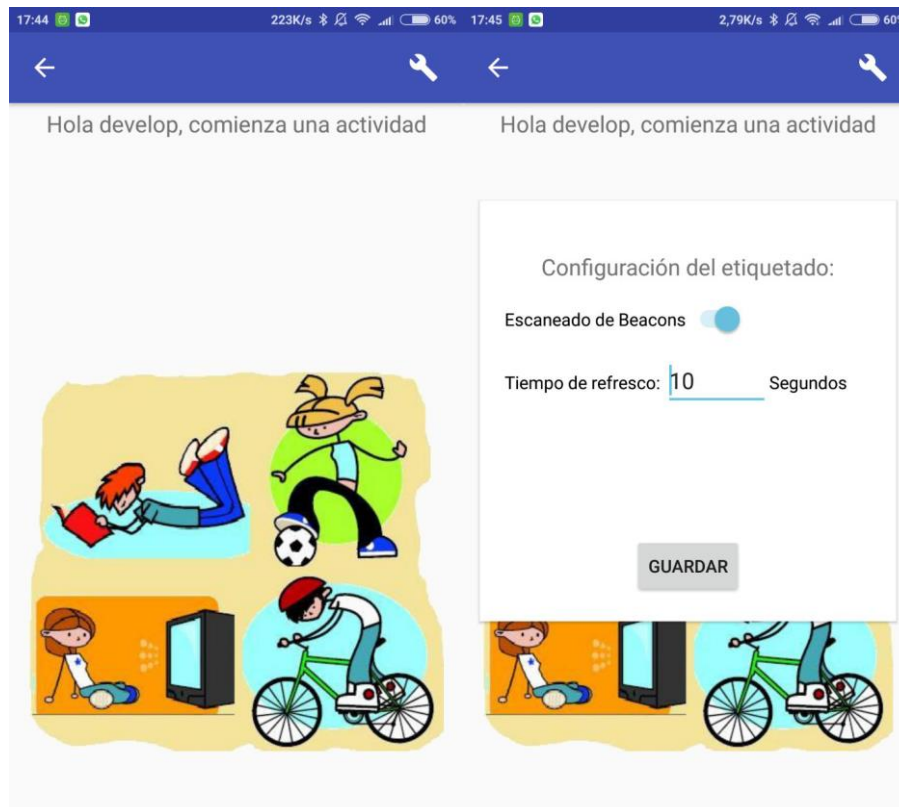


Ilustración 103 Etiquetado de actividad y configuración de etiquetado

Si pulsamos en el icono de la llave, podemos configurar si queremos escanear balizas beacon mientras realizamos cada una de las actividades y además podemos seleccionar cada cuanto tiempo queremos realizar dicho escaneo.

En el momento de etiquetar una actividad, la aplicación móvil mostrará al usuario una lista con todas las actividades que se encuentra realizando en dicho momento.

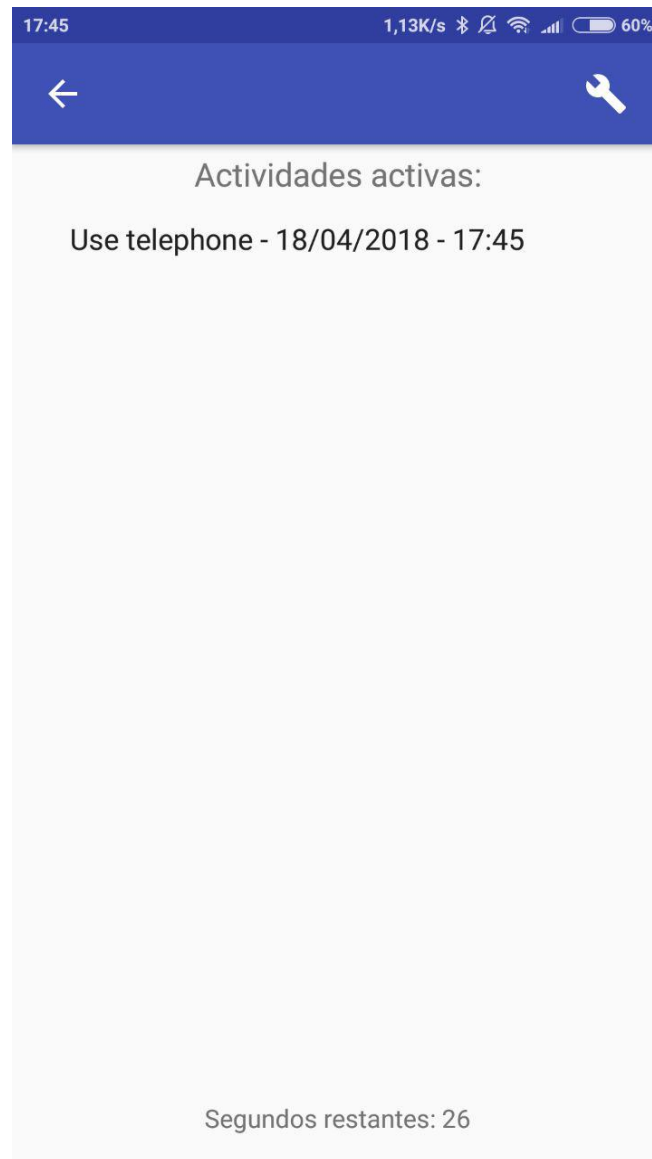


Ilustración 104 Actividades realizándose

7.3.3.6 Suscribiendonos a eventos

Por último, otra de las funcionalidades que tiene esta aplicación móvil es poder suscribirnos a cualquier objeto o sensor del ambiente inteligente que estamos monitorizando. En dicha vista, seleccionaremos cada objeto y cuando este realice un evento en el sistema, nos alertará mediante una notificación push en nuestro teléfono móvil.

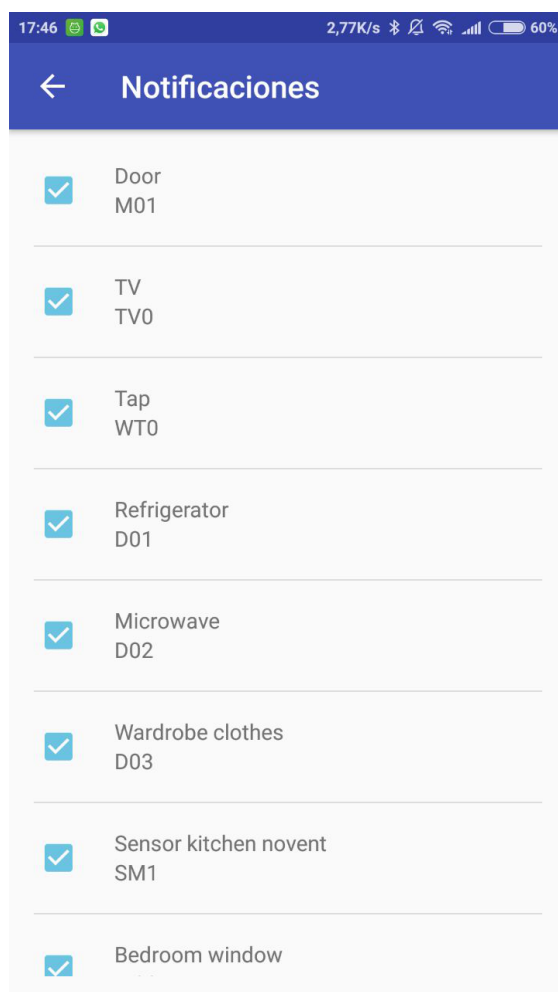


Ilustración 105 Suscripciones a eventos

Gracias a estas notificaciones podemos estar al corriente de todo lo que sucede en el sistema sin necesidad de estar comprobando continuamente el estado de los sensores.

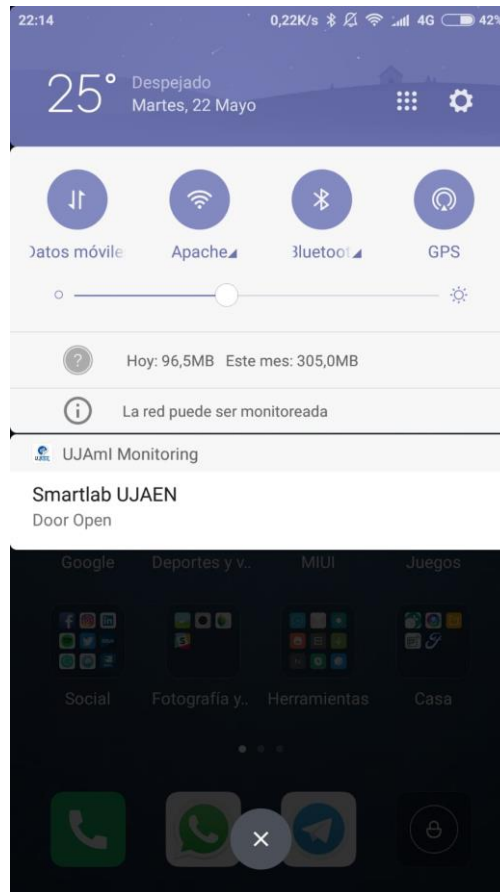


Ilustración 106 Notificación en tiempo real

