

Grado en Ingeniería en Tecnologías de Telecomunicación
2018-2019

Trabajo Fin de Grado

“Aplicación Móvil para Monitorizar la Actividad de Personas en Riesgo Cardiovascular e Ictus”

Javier Yuste Rodríguez-Escalona

Tutor

David Griol Barres

Leganés, 14 octubre 2019



Esta obra se encuentra sujeta a la licencia Creative Commons **Reconocimiento – No Comercial – Sin Obra Derivada**

RESUMEN

En la actualidad, un gran número de personas sufren enfermedades cardiovasculares. De forma particular, las personas de avanzada edad tienen un mayor factor de riesgo. Por ello, en este trabajo se implementa una solución basada en IoT (*Internet of Things*) para ser utilizada por personas de la tercera edad que viven solas y por su entorno de confianza. En concreto, esta solución se enmarca en el sector de la e-Salud.

El sistema que conforma el proyecto está formado por distintos bloques interconectados entre sí. La aplicación Android pretende ser el punto de encuentro entre las distintas funcionalidades. Por un lado, se utiliza la smartband Xiaomi Mi Band 2 para monitorizar de forma continua la actividad cardíaca del paciente, donde la comunicación con la aplicación se realiza a través del protocolo BLE (*Bluetooth Low Energy*). Los datos del usuario, tanto de configuración como de pulso cardíaco, se guardan en la base de datos de Firebase, servicio que, además, proporciona la autenticación del usuario. En el sistema se ha creado otro bloque complementario formado por una Raspberry Pi 3 con un sensor de distancia ultrasónico, proporcionando información de la actividad del paciente en la vivienda.

El trabajo realizado cumple con los objetivos y requisitos establecidos al inicio del proyecto, siendo una aplicación funcional y fácil de usar por el usuario final.

Palabras clave: IoT, tercera edad, e-Salud, Xiaomi Mi Band 2, BLE, Firebase, Raspberry, aplicaciones móviles, enfermedades cardiovasculares.

ABSTRACT

Currently, a large number of people suffer from cardiovascular disease. In particular, older people have a higher risk factor. Therefore, this work implements an IoT-based solution to be used by the elderly and its family environment. In particular, this solution is an eHealth-based solution.

The project encompasses a system that consists of different blocks interconnected with each other. The Android application aims to be the meeting point between the different functionalities. On the one hand, the Xiaomi Mi Band 2 smartband is used to continuously monitor the patient's heart rate, where communication with the application is done through the BLE (Bluetooth Low Energy) protocol. User data, both configuration and heart rate data, is saved in the Firebase database, a service that also provides user authentication. In the system has been created another complementary block consisting of a Raspberry Pi 3 with an ultrasonic distance sensor, providing information of the patient's activity in the home.

The work carried out agrees with the objectives and requirements established at the beginning of the project, being a functional and easy to use application.

Key words: IoT, elderly, e-Health, Xiaomi Mi Band2, BLE, Firebase, Raspberry, mobile applications, cardiovascular diseases.

DEDICATORIA

Tras la realización de este trabajo, sólo tengo palabras de agradecimiento para mi tutor, David, que ha estado siempre cuando me surgía algún obstáculo en el desarrollo del proyecto. Agradezco infinitamente la paciencia y compañía de mi familia y amigos, especialmente a mis padres Pedro José y María de los Ángeles, que sin ellos no hubiera sido posible. Por último, agradecer a la Universidad Carlos III de Madrid por poner a disposición medios físicos y digitales para el mejor desarrollo del Trabajo Fin de Grado.

ÍNDICE DE CONTENIDOS

1.	INTRODUCCIÓN	1
1.1.	Motivación	1
1.2.	Objetivos	2
1.3.	Entorno socioeconómico.....	2
1.3.1.	Impacto socioeconómico	3
1.3.2.	Presupuesto.....	6
1.4.	Marco regulador.....	8
1.4.1.	LOPD	8
1.4.2.	Regulación Firebase	9
1.4.3.	Marco laboral	10
1.4.4.	Estándares técnicos.....	11
1.5.	Estructura de la memoria.....	12
2.	ESTADO DEL ARTE	13
2.1.	Situación actual	13
2.1.1.	Enfermedades cardiovasculares.....	13
2.1.2.	IoT.....	15
2.1.3.	e-Salud.....	17
2.1.4.	Tecnología wearable	19
2.1.5.	Smartband.....	21
2.1.6.	Sistemas operativos para móviles.....	22
2.2.	Soluciones existentes	23
3.	DESARROLLO DEL PROYECTO	25
3.1.	Metodología	25
3.2.	Análisis de requerimientos.....	25
3.2.1.	Objetivos del sistema	26
3.2.2.	Requisitos Funcionales	27
3.2.3.	Requisitos No Funcionales	29
3.3.	Alternativas de diseño.....	31
3.3.1.	Sistema Operativo	31
3.3.2.	Base de Datos.....	32
3.3.3.	HW para la vivienda	33
3.3.4.	Smartband.....	35
3.4.	Diseño y arquitectura.....	36
3.4.1.	Casos de uso	36

3.4.2.	Diagramas de Actividad.....	40
3.4.3.	Prototipos.....	43
3.4.4.	Diagrama Entidad-Relación.....	45
3.5.	Desarrollo e implementación.....	46
3.5.1.	Aplicación Android.....	46
3.5.1.1.	Android Studio.....	47
3.5.1.2.	Android Manifest.....	48
3.5.1.3.	Firebase: autenticación y BD en tiempo real.....	49
3.5.1.4.	Conexión Smartband.....	56
3.5.2.	Raspberry Pi en la vivienda.....	62
3.5.2.1.	Configuración inicial Raspberry.....	62
3.5.2.2.	Configuración HW del sensor HC-SR04.....	64
3.5.2.3.	Configuración SW del sensor HC-SR04.....	66
3.6.	Pruebas.....	69
3.6.1.	Pruebas manuales.....	69
3.6.2.	Pruebas de aceptación.....	71
4.	PLANIFICACIÓN.....	73
4.1.	Fases principales.....	73
4.2.	Diagrama de Gantt.....	73
5.	CONCLUSIONES Y TRABAJO FUTURO.....	76
5.1.	Revisión de los objetivos.....	76
5.2.	Trabajos futuros.....	76
APÉNDICE A: ENGLISH SUMMARY.....		78
A.1.	Introduction.....	78
A.1.1.	Motivation.....	78
A.1.2.	Objectives.....	78
A.1.3.	Socio-economic environment.....	78
A.1.4.	Regulatory Framework.....	79
A.1.5.	Document Structure.....	79
A.2.	State of the art.....	80
A.2.1.	Current situation.....	80
A.2.2.	Existing solutions.....	81
A.3.	Project development.....	82
A.3.1.	Methodology.....	82
A.3.2.	Requirements analysis.....	82
A.3.3.	Design alternatives.....	82

A.3.4. Design and architecture	83
A.3.5. Development and implementation.....	83
A.3.5.1. Android app	83
A.3.5.2. Raspberry Pi in the home.....	85
A.3.6. Testing.....	85
A.4. Planning	85
A.5. Conclusions and future work	86
BIBLIOGRAFÍA.....	88

ÍNDICE DE FIGURAS

Fig 1.1 Estimación de población hasta 2068 en España [9]	4
Fig 1.2 Porcentaje de personas que han utilizado internet en los últimos tres meses, por grupos de edad, en España, 2017-2018 [9].....	5
Fig 2.1 Tipos de dispositivos wearable [11]	19
Fig 2.2 Estimación del mercado de los dispositivos wearable hasta 2023 [13]	21
Fig 2.3 Distribución mundial de dispositivos móviles según el sistema operativo. Fuente: https://gs.statcounter.com/os-market-share/mobile/worldwide/	22
Fig 3.1 Placa Arduino [17]	33
Fig 3.2 Raspberry Pi 3 Model B [18]	33
Fig 3.3 Sensor de distancia ultrasónico HC-SR04 [19].....	34
Fig 3.4 Vista global del sistema	36
Fig 3.5 Elementos diagrama de actividad.....	41
Fig 3.6 Diagrama de actividad: Registro e inicio de sesión	41
Fig 3.7 Diagrama de actividad: Configuración.....	42
Fig 3.8 Diagrama de actividad: Bluetooth	42
Fig 3.9 Diagrama de actividad: Panel de control Paciente.....	42
Fig 3.10 Diagrama de actividad: Panel de control Supervisor.....	43
Fig 3.11 Prototipo Inicio de sesión y Registro	44
Fig 3.12 Prototipos Configuración y Escáner.....	44
Fig 3.13 Prototipo Panel de Control Paciente y Supervisor	45
Fig 3.14 Prototipo Perfil de Usuario	45
Fig 3.15 Diagrama de Entidad-Relación	46
Fig 3.16 Vista inicial Android Studio	48
Fig 3.17 Android Manifest: declaración de permisos.....	49
Fig 3.18 Asistente de Firebase en Android Studio	49
Fig 3.19 Captura de pantalla: Registro	50
Fig 3.20 Captura de pantalla: Iniciar Sesión	51
Fig 3.21 Vista de los usuarios registrados en la consola de Firebase.....	51
Fig 3.22 Declaración de variables de la clase User.java	52
Fig 3.23 Captura de pantalla: Configuración.....	53
Fig 3.24 Funciones Firebase en la actividad de Configuración.....	54
Fig 3.25 Captura de pantalla: Perfil	54
Fig 3.26 Función Firebase en la actividad Perfil	55

Fig 3.27 Captura de pantalla: HomeSupervisor.....	56
Fig 3.28 Esquema protocolo BLE	57
Fig 3.29 UUIDs del servicio HeartRate.....	58
Fig 3.30 Captura de pantalla: SearchDevice.....	58
Fig 3.31 HomePatient: función startConnecting().....	60
Fig 3.32 Captura de pantalla: Homepatient	60
Fig 3.33 HomePatient: función procesarHeartRate().....	61
Fig 3.34 Captura de pantalla: Mensaje SMS recibido	61
Fig 3.35 VNC Viewer configurado.....	63
Fig 3.36 Escritorio Raspberry.....	64
Fig 3.37 Funcionamiento Sensor HC-SR04 [24]	65
Fig 3.38 Esquema circuito Raspberry con HC-SR04 [25]	65
Fig 3.39 Circuito Raspberry implementado.....	66
Fig 3.40 Código Python para calcular la distancia en Raspberry.....	66
Fig 3.41 Código Python para inicializar la aplicación en Firebase	67
Fig 3.42 Código Python de la función principal.....	68
Fig 3.43 Ejecución programa Python en el terminal de Raspbian.....	68
Fig 3.44 Actualización Punto de control en la consola de Firebase	68
Fig 4.1 Lista de tareas del proyecto.....	74
Fig 4.2 Diagrama de Gantt.....	75

ÍNDICE DE TABLAS

Tabla 1.1 GASTOS DE PERSONAL.....	6
Tabla 1.2 AMORTIZACIÓN DE EQUIPOS INFORMÁTICOS.....	6
Tabla 1.3 GASTOS DE MATERIAL.....	7
Tabla 1.4 GASTOS EN SUMINISTROS.....	7
Tabla 1.5 PRESUPUESTO DEL PROYECTO.....	8
Tabla 1.6 CERTIFICACIÓN DE LOS SERVICIOS DE FIREBASE UTILIZADOS.....	10
Tabla 2.1 COMPARATIVA ENTRE LOS PRINCIPALES PROTOCOLOS DE COMUNICACIÓN IOT.....	16
Tabla 2.2 COMPARATIVA SOLUCIONES ENCONTRADAS.....	24
Tabla 3.1 OBJETIVO 01.....	26
Tabla 3.2 OBJETIVO 02.....	26
Tabla 3.3 OBJETIVO 03.....	26
Tabla 3.4 OBJETIVO 04.....	26
Tabla 3.5 OBJETIVO 05.....	27
Tabla 3.6 REQUISITO FUNCIONAL 01.....	27
Tabla 3.7 REQUISITO FUNCIONAL 02.....	27
Tabla 3.8 REQUISITO FUNCIONAL 03.....	27
Tabla 3.9 REQUISITO FUNCIONAL 04.....	28
Tabla 3.10 REQUISITO FUNCIONAL 05.....	28
Tabla 3.11 REQUISITO FUNCIONAL 06.....	28
Tabla 3.12 REQUISITO FUNCIONAL 07.....	28
Tabla 3.13 REQUISITO FUNCIONAL 08.....	28
Tabla 3.14 REQUISITO FUNCIONAL 09.....	29
Tabla 3.15 REQUISITO FUNCIONAL 10.....	29
Tabla 3.16 REQUISITO FUNCIONAL 11.....	29
Tabla 3.17 REQUISITO NO FUNCIONAL 01.....	30
Tabla 3.18 REQUISITO NO FUNCIONAL 02.....	30
Tabla 3.19 REQUISITO NO FUNCIONAL 03.....	30
Tabla 3.20 REQUISITO NO FUNCIONAL 04.....	30
Tabla 3.21 REQUISITO NO FUNCIONAL 05.....	30
Tabla 3.22 REQUISITO NO FUNCIONAL 06.....	30
Tabla 3.23 COMPARATIVA DISEÑO ANDROID E IOS.....	31
Tabla 3.24 COMPARATIVA ARDUINO VS RASPBERRY PI.....	33
Tabla 3.25 CASO DE USO 01.....	37

Tabla 3.26 CASO DE USO 02	37
Tabla 3.27 CASO DE USO 03	38
Tabla 3.28 CASO DE USO 04	38
Tabla 3.29 CASO DE USO 05	39
Tabla 3.30 CASO DE USO 06	39
Tabla 3.31 CASO DE USO 07	40
Tabla 3.32 CASO DE USO 08	40
Tabla 3.33 PRUEBA MANUAL 01	69
Tabla 3.34 PRUEBA MANUAL 02	69
Tabla 3.35 PRUEBA MANUAL 03	69
Tabla 3.36 PRUEBA MANUAL 04	70
Tabla 3.37 PRUEBA MANUAL 05	70
Tabla 3.38 PRUEBA MANUAL 06	70
Tabla 3.39 PRUEBA MANUAL 07	71
Tabla 3.40 PRUEBA MANUAL 08	71
Tabla 3.41 PRUEBA MANUAL 09	71
Tabla 3.42 PRUEBA MANUAL 10	71
Tabla 3.43 PRUEBA ACEPTACIÓN	72

ABREVIATURAS

API	Interfaz de Programación de Aplicaciones (<i>Application Programming Interface</i>)
AVC	Accidentes Vasculares Cerebrales
ARCO	Acceso, Rectificación, Cancelación y Oposición
BD	Base de Datos
BLE	Bluetooth de baja energía (<i>Bluetooth Low Energy</i>)
bpm	Pulsaciones por minuto (<i>Beats per minute</i>)
ECG	Electrocardiograma
ECV	Enfermedades Cardiovasculares
GPS	Sistema de posicionamiento global (<i>Global Positioning System</i>)
HW	Hardware
IoT	Internet de las cosas (<i>Internet of Things</i>)
IVA	Impuesto sobre el Valor Añadido
LOPD	Ley Orgánica de Protección de Datos
OMS	Organización Mundial de la Salud
RGPD	Reglamento General de Protección de Datos
SW	Software
TIC	Tecnologías de la Información y la Comunicación
UE	Unión Europea
UID	Identificador de Usuario (<i>User Identifier</i>)
XML	Lenguaje de Marcado Extensible (<i>Extensible Markup Language</i>)

1. INTRODUCCIÓN

Este primer capítulo del documento tiene como objetivo presentar una serie de aspectos fundamentales del proyecto. En primer lugar, en la motivación se expone la necesidad de crear una solución ante un problema real de la sociedad. En el segundo epígrafe, se presentan los distintos objetivos perseguidos con el proyecto, tanto personales como de proyecto. Seguidamente, se habla del entorno socioeconómico, incluyendo, entre otros elementos, un presupuesto para la elaboración del proyecto. Por último, se expone el marco regulador a tener en cuenta para el desarrollo.

1.1. Motivación

Según los informes estadísticos de los últimos años, las ECV son la primera causa de mortalidad en España, por encima del cáncer y de las enfermedades respiratorias. Concretamente, las causas más frecuentes en 2016 fueron las relativas a enfermedades isquémicas del corazón y las enfermedades cerebrovasculares, en hombres y mujeres, respectivamente [1]. Ante un caso de problema cardiovascular o cerebrovascular, como puede ser el ictus, es fundamental la detección inmediata, de ahí la importancia de utilizar un análisis de datos en tiempo real.

La situación actual española es la de una sociedad envejecida, con una clara tendencia al alza en los años venideros, debido a que el mayor volumen de población se encuentra en torno a los 40-50 años [2]. Este hecho hace entrever que las inversiones en cuidado de la tercera edad crecerán más en los próximos años.

Además, según datos del 2017, en España existen 4,7 millones de personas que conforman hogares unipersonales, es decir, que viven solas. Más de la tercera parte del total se corresponde a personas de 65 años y más [1].

Por otro lado, en las distintas realidades de la sociedad se tienen cada vez más dispositivos conectados a través de Internet, conformando lo que se conoce como IoT, con unas previsiones de inversión en 2019 de 745 mil millones de dólares [4]. En particular, el sector sanitario es uno de los principales activos de estas tecnologías.

Como respuesta a las premisas anteriormente citadas, este proyecto de investigación y desarrollo surge con el afán de mejorar la prevención y el seguimiento de aquellas personas de avanzada edad que se encuentran en riesgo cardiovascular, para mayor

tranquilidad de sus familias. Utilizando IoT se consigue disponer de los datos recogidos por distintos sensores en tiempo real, en cuestión de milisegundos.

1.2. Objetivos

En este apartado se distinguen dos tipos de objetivos, los personales y del proyecto. Los objetivos personales se corresponden a aquellos cuya finalidad es complementar los conocimientos previos a la realización del proyecto. Por otro lado, los objetivos del proyecto son los que se quieren conseguir al realizar el proyecto.

Los objetivos personales son los siguientes:

- Investigar y conocer las ECV más importantes y su diagnóstico.
- Planificar y ejecutar de forma correcta un proyecto de una envergadura determinada.
- Aprender a desarrollar aplicaciones móviles Android, utilizando Android Studio.
- Aprender nuevos lenguajes de programación como XML y Python.
- Adquirir un conocimiento preciso sobre la tecnología Bluetooth y BLE.
- Utilizar Firebase como BD en tiempo real.
- Conocer el funcionamiento de la Raspberry Pi y componentes.
- Introducirse en el funcionamiento y uso de sensores.
- Diseñar un sistema IoT complejo con procesamiento de datos en tiempo real.

Los objetivos del proyecto son los siguientes:

- Establecer los requisitos mínimos que ha de cumplir el prototipo final.
- Cumplir los requisitos mínimos establecidos previo al inicio del proyecto.
- Crear una aplicación Android con una interfaz de usuario intuitiva, teniendo en cuenta el sector al que va destinado.

1.3. Entorno socioeconómico

En este epígrafe se expone el impacto socioeconómico y el presupuesto para la elaboración de este proyecto. Para analizar el impacto socioeconómico se parte de un análisis económico del mercado de las aplicaciones móviles, seguido de una visión de las necesidades sociales que se pretenden cubrir. Por último, se muestra el presupuesto de la elaboración de este proyecto.

1.3.1. Impacto socioeconómico

Desde un punto de vista económico, el mercado de las aplicaciones móviles en el mundo crece cada año. Los ingresos totales en 2014 fueron cercanos a los 98 mil millones de dólares y en 2018 ya ascendían a los 365 mil millones de dólares. Se prevé que en 2023 esta cifra ascienda hasta los 935 mil millones de dólares, lo que supone un incremento en 10 veces a las cifras de 2014 [15].

En España, en particular, 22 millones de usuarios descargan aplicaciones y se realizan 4 millones de descargas cada día. El tiempo que se dedica a las aplicaciones móviles ha aumentado en los últimos años, donde un tercio de la población consume más de 20 horas semanales.

Por otro lado, el 82% de los españoles sólo se baja aplicaciones de descarga gratuita. Además, lo que más valoran los españoles es que la aplicación cubra una necesidad. En caso de no estar satisfecho con la aplicación se borra inmediatamente, siendo esta la principal causa de desinstalación de aplicaciones [16].

La situación demográfica de España es la de una sociedad envejecida. La natalidad está a unos niveles por debajo del mínimo para garantizar el reemplazo generacional, de 2'1 hijos por mujer. En la Figura 1.1. se puede ver la pirámide de población en el 2018 y las estimaciones realizadas hasta el 2068, quedando reflejado el aumento de población en las edades de 60 años en adelante.

A su vez, en España, un gran número de personas mayores viven solas. En 2017 casi el 30% de las mujeres mayores de 65 años formaban hogares unipersonales, frente al 17'7 % de los hombres. Existen 4'2 plazas de residencia por cada 100 personas mayores de 65 años, lo que hace que el porcentaje de personas mayores que viven solas sea mayor, entre otros factores [9].

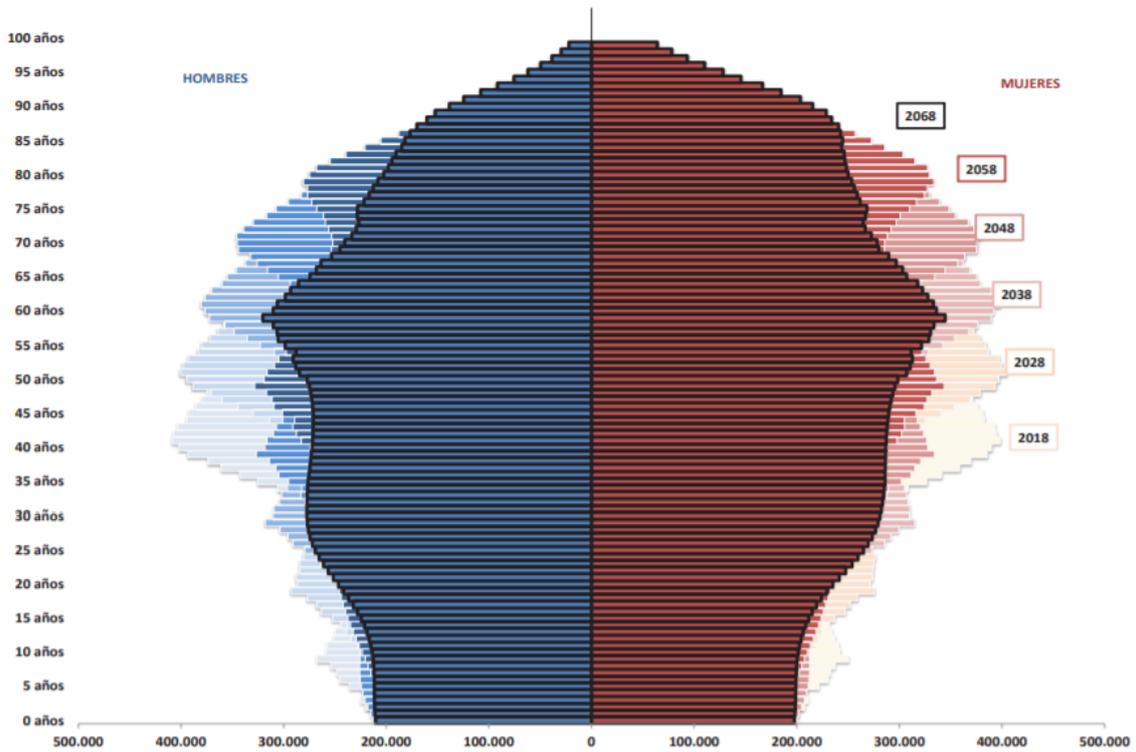


Fig 1.1 Estimación de población hasta 2068 en España [9]

Por otro lado, los familiares de estas personas mayores necesitan tener un control sobre la situación y actividad de sus mayores, para su mayor tranquilidad y bienestar. Debido a los avances técnicos existentes, se han implementado diversas soluciones para paliar esta necesidad, como se verá más adelante.

Por último, la brecha digital en España es un desafío al que se enfrenta la sociedad, ya que la diferencia de uso de las TIC es muy amplia entre el grupo de los 16-24 años y el de 65-74 años. No obstante, esta tendencia está cambiando, como se puede observar en la Figura 2.2.

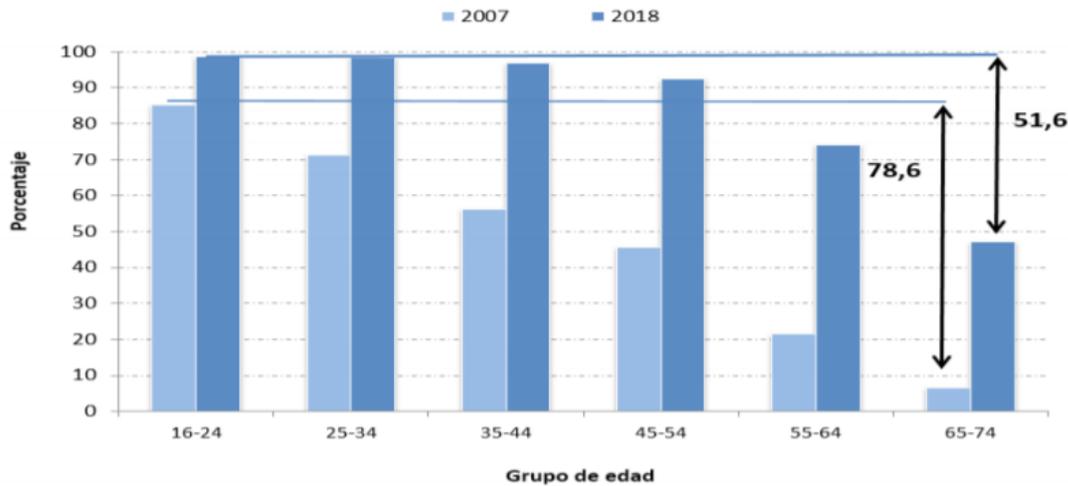


Fig 1.2 Porcentaje de personas que han utilizado internet en los últimos tres meses, por grupos de edad, en España, 2017-2018 [9]

El desarrollo de este proyecto busca cubrir las necesidades presentadas en los párrafos anteriores, principalmente la necesidad que tienen las personas mayores que viven solas de un control por parte de su entorno más cercano. Por otro lado, se fomenta el uso de las nuevas tecnologías en los rangos de más avanzada edad, disminuyendo, de esta forma, la brecha digital.

Como se ha indicado al comienzo de este epígrafe, el mercado de las aplicaciones móviles está en una dinámica positiva lo que hace viable decantarse por un proyecto en este sector. No obstante, deben tenerse en cuenta las consideraciones de la gratuidad y utilidad para que la aplicación pueda llegar a un alto nivel de penetración en la sociedad.

El proyecto será explotado de la siguiente manera:

- **Aplicación gratuita:** la descarga de la aplicación será gratuita. Por tanto, todos los usuarios que dispongan de la smartband Xiaomi Mi Band 2 podrán utilizar la aplicación en su modo control de usuarios. De esta forma, se alcanzará un alto número de usuarios.
- **Venta del HW en la vivienda:** el HW conformado por la placa y los sensores se venderá a un precio asequible para el usuario, proporcionando un margen de beneficio razonable.
- **Sin publicidad:** en un principio no se contempla el incorporar publicidad de terceros en la aplicación. Según avance la vida del proyecto se estudiará la posibilidad de incorporarla, dependiendo en gran medida de la penetración alcanzada.

Por todo lo detallado anteriormente, se llega a la conclusión de que el proyecto aquí presentado tendrá un impacto socioeconómico positivo, ya que contribuye activamente a cubrir una necesidad social en un sector económico en auge.

1.3.2. Presupuesto

Un presupuesto tiene como objetivo presentar de forma ordenada el dinero que se destina para la elaboración de un proyecto. En el caso particular de este proyecto, el presupuesto se basa principalmente en el gasto de personal. También hay que indicar las dotaciones correspondientes para amortizaciones y gastos materiales.

En primer lugar, en la Tabla 1.1, se detallan los gastos relativos al personal. Para ello se ha considerado un salario anual de 25.000 € anuales, para un recién titulado en ingeniería de telecomunicaciones.

Tabla 1.1 GASTOS DE PERSONAL

Puesto de trabajo	Salario anual	Seguridad social (28,30%)	Coste total	Coste hora (1,800 horas al año)
Ingeniero de telecomunicaciones	25.000 €	7.075 €	32.075 €	17,82 € por hora
Duración del proyecto		TOTAL		
300 horas		5.345,83 €		

A continuación, en la Tabla 1.2, se presentan los gastos correspondientes a amortización de equipos informáticos. Los equipos utilizados son un ordenador portátil HP y un teléfono móvil Xiaomi Mi A2 lite.

Tabla 1.2 AMORTIZACIÓN DE EQUIPOS INFORMÁTICOS

Equipo	Coste	Amortización (meses)	Uso (meses)	Coste total
Ordenador portátil HP	850 €	60	4	56,67 €

Smartphone Xiaomi Mi A2 lite	200 €	36	4	22,22 €
Total amortización (Inmovilizado material)				78,89 €

En las Tablas 1.3 y 1.4, se detalla la partida presupuestaria para gastos materiales y suministros, respectivamente. A todos los gastos se les añadirá el IVA al final de forma conjunta con los demás gastos.

Tabla 1.3 GASTOS DE MATERIAL

Descripción	Coste	Cantidad	Total
Raspberry Pi 3	30 €	1	30 €
Xiaomi Mi Band 2	20 €	1	20 €
Pack Cables “Jumpers” para protoboard.	7 €	1	7 €
Pack mini protoboard (x6)	7 €	1	7 €
Resistencia	0,25 €	2	0,50 €
Sensor de distancia ultrasonidos	3 €	1	3 €
TOTAL (sin IVA)			68 €

Tabla 1.4 GASTOS EN SUMINISTROS

Tipo	Coste mensual	Detalle	Meses	Coste total
Internet	30 €	Fibra 50 MB	4	120 €
Luz	4 € (0,139 €/kWh)	28,88 kWh	4	16 €

TOTAL	136 €
-------	-------

Para finalizar, en la Tabla 1.5, se presenta el presupuesto completo, incluyendo las distintas partidas de gasto detalladas previamente.

Tabla 1.5 PRESUPUESTO DEL PROYECTO

Concepto	Referencia	Coste
Gastos de personal	Tabla 1.1	5.345,83 €
Amortización de equipos informáticos	Tabla 1.2	78,89 €
Gastos de material	Tabla 1.3	68 €
Suministros	Tabla 1.4	136 €
TOTAL (sin IVA)		5.628,72 €
TOTAL (con 21% IVA)		6.810,75 €

1.4. Marco regulador

Para el desarrollo de una aplicación móvil se deben tener en cuenta las distintas legislaciones y regulaciones que deben respetar los desarrolladores. En concreto, este proyecto tiene que cumplir con la LOPD y el marco laboral español, además de tener en cuenta la normativa de Firebase.

1.4.1. LOPD

Para el desarrollo de una aplicación móvil se deben tener en cuenta las distintas legislaciones y regulaciones que deben cumplir los desarrolladores.

En primer lugar, se debe cumplir con la LOPD acerca de la protección de datos personales y garantía de los derechos digitales. Los desarrolladores deben respetar las siguientes obligaciones:

- Proteger la privacidad, con un consentimiento previo por parte del usuario tras haber sido informado correctamente de las condiciones.

- Indicar al usuario cuál es el objetivo de la recolección de sus datos y si va a ser cedida a terceros, de una forma entendible para cualquier usuario experto.
- Informar al usuario acerca de sus derechos y los tiempos de retención de sus datos.
- Dejar al usuario que pueda cancelar la instalación de la aplicación.
- Indicar qué información va a ser accedida previamente a la descarga.
- Proporcionar la identidad de la entidad desarrolladora.
- Indicar al usuario el procedimiento para ejercer sus derechos ARCO [5].

Como se ha indicado previamente, el usuario puede acogerse al ejercicio de sus derechos ARCO. Estos derechos pretenden proporcionar al usuario el control sobre sus datos. Estos derechos son los siguientes:

- **Derecho de acceso.** El usuario tiene la posibilidad de solicitar información sobre si sus datos personales están siendo tratados. Además, el responsable dispone de un plazo de 30 días para responder a dicha solicitud.
- **Derecho de rectificación.** Este derecho permite al usuario solicitar el cambio de sus datos por inexactitud o falsedad.
- **Derecho de oposición.** Expone el derecho de un usuario a oponerse al tratamiento de sus datos en unas condiciones concretas.
- **Derecho de cancelación.** Permite al usuario solicitar la supresión de aquellos datos que sean inadecuados [6].

La serie de derechos vistos anteriormente se amplían con el RGPD, proveniente del marco regulador de la UE. El RGPD adopta, entre otros, el derecho al olvido. Este consiste en la eliminación de los datos del usuario cuando, por ejemplo, este retire su consentimiento o en caso de que los datos ya hayan cumplido su finalidad.

1.4.2. Regulación Firebase

El proyecto utiliza algunos servicios de Firebase como BD en tiempo real y autenticación. Por ello, es útil conocer las condiciones a las que está sujeto el desarrollador y los potenciales usuarios de la aplicación.

En protección de datos, Firebase está acorde al RGPD de la UE. Además, los clientes de Firebase suelen controlar los datos de los usuarios de su aplicación, quedando en su responsabilidad el cumplir con el reglamento y la legislación correspondiente a su entorno legal.

Por otro lado, Firebase cumple con los certificados ISO y SOC. Concretamente, los servicios empleados en la realización de la aplicación completaron los procesos de certificación que aparecen en la Tabla 1.6 [7].

Tabla 1.6 CERTIFICACIÓN DE LOS SERVICIOS DE FIREBASE UTILIZADOS [7]

Nombre del servicio	ISO 27001	ISO 27017	ISO 27018	SOC 1	SOC 2	SOC 3
Cloud Storage para Firebase	✓	✓	✓	✓	✓	✓
Autenticación Firebase	✓	✓	✓	✓	✓	✓

1.4.3. Marco laboral

En el apartado 1.3.2, se presentaba el presupuesto de la elaboración de este proyecto. Concretamente, el mayor gasto del presupuesto era el asociado a los gastos en personal, siendo un indicativo de la importancia del marco laboral en este proyecto. A continuación, se presentan las distintas legislaciones y documentos a tener en cuenta:

- **Constitución Española:** se recogen los derechos y deberes de los españoles, también en el ámbito laboral. Estableciendo en el artículo 35 que “La ley regulará un estatuto de los trabajadores” [20].
- **Real Decreto 1462/2018:** se establece un salario mínimo interprofesional de 900€ para el año 2019.
- **Ley del Estatuto de los Trabajadores:** se exponen artículos sobre la jornada laboral, contratos de trabajo, etc.

Por otro lado, en lo recogido en el Régimen General de la Seguridad Social, se establecen las bases de cotización dependiendo del grupo de cotización. El desarrollador de este proyecto pertenece al grupo 2, que contiene a los Ingenieros Técnicos o graduados en Ingeniería. Para este grupo, la base mínima de cotización es de 1,215.90 € al mes, y la máxima de 4,070.10 € al mes [21]. Al desarrollador del proyecto se le pagarán 2,083.33 € brutos, estando dentro de la normativa actual.

1.4.4. Estándares técnicos

En este proyecto se pretende implementar de forma correcta un sistema que integra una aplicación móvil Android, entre otros subgrupos. Para ello, hay que tener en cuenta los estándares de calidad de desarrollo de este tipo de aplicaciones.

La calidad de la aplicación está directamente relacionada con el éxito que pueda llegar a tener en un futuro. Se deben cumplir correctamente los estándares relacionados con los aspectos siguientes:

- **Entorno visual e interacción del usuario:** a través de una serie de recomendaciones, se busca que el usuario encuentre la aplicación accesible e intuitiva. Establece recomendaciones para el uso de notificaciones, navegación y diseño. Por ejemplo, uno de los patrones indica que la aplicación tiene que habilitar el botón “Atrás” (del sistema) para el funcionamiento de la aplicación y evitar la redundancia de otro botón que tenga el mismo propósito.
- **Funcionalidad:** está relacionada con los permisos que necesita la aplicación para realizar sus funcionalidades. Además, establece el criterio para la ubicación de la instalación, la funcionalidad del audio en las aplicaciones, la Interfaz de Usuario. Por ejemplo, se indica que sólo se deben requerir la cantidad mínima de permisos necesarios para el funcionamiento.
- **Compatibilidad, rendimiento y estabilidad:** en lo relacionado con la estabilidad, se indica que la app no debe fallar o cerrarse de forma repentina. Las aplicaciones se deben ejecutar rápidamente, sin esperar mucho tiempo, además de incorporar funcionalidades de ahorro de batería. Para que la aplicación sea compatible, se debe compilar con la última versión disponible.
- **Seguridad:** la app debe guardar todos los datos en la memoria interna y verificar previamente los datos que van a ser accedidos externamente. Las conexiones en red deben protegerse mediante SSL.
- **Configuración de un entorno de prueba:** la app debe ser simulada para garantizar su correcto funcionamiento. Se puede realizar mediante simuladores virtuales, disponibles en Android Studio, o mediante dispositivos físicos.
- **Procedimientos de prueba:** es necesario probar la aplicación en el peor escenario posible para evitar que pueda darse algún fallo inesperado en el futuro. Para ello se propone la habilitación de “Strict Mode” para detectar fallos en el rendimiento, en la red o en las operaciones con archivos [22].

1.5. Estructura de la memoria

La memoria del proyecto está organizada en las siguientes partes:

- **Estado del arte:** se expone la situación actual en cuanto a las ECV, tecnologías utilizadas, IoT, *wearables* (“ponibles”), sensores, e-Salud y proyectos similares.
- **Desarrollo del proyecto:** trata la metodología a seguir, el análisis de requerimientos, el diseño y arquitectura del prototipo a realizar, su posterior desarrollo e implementación y las correspondientes pruebas para comprobar su funcionamiento.
- **Planificación:** incluye la organización temporal seguida en el proyecto.
- **Conclusiones:** se analizan la consecución de los objetivos establecidos al inicio del proyecto junto a la valoración personal del autor.

2. ESTADO DEL ARTE

Para la realización de un proyecto como el presentado en este documento, se requiere investigar acerca de qué conceptos y tecnologías se usan para un proyecto determinado. Además, el análisis de soluciones existentes permite analizar las soluciones de proyectos competidores y la capacidad de implementar mejoras competitivas en el proyecto que lo diferencien de la competencia.

En este apartado se presenta la información acerca de la situación actual de las ECV, la tercera edad en España, las tecnologías utilizadas para detectar este tipo de enfermedades, la e-Salud, la situación del IoT, el uso de *wearables* (“ponibles”) y soluciones existentes.

2.1. Situación actual

En este primer epígrafe, se presenta una panorámica sobre los distintos temas y tecnologías involucradas en el desarrollo de este proyecto. En primer lugar, se expone la situación actual de las ECV, presentando, también, los factores de riesgo cardiovascular. Seguidamente, se introducirán los términos de IoT, e-Salud, Tecnología Wearable y Smartband, términos en los que se basa el sistema realizado. Para terminar, se analizará la situación actual acerca de los distintos sistemas operativos en teléfonos móviles.

2.1.1. Enfermedades cardiovasculares

Las ECV son un conjunto formado por distintas irregularidades del corazón y vasos sanguíneos, entre las que se encuentran la cardiopatía coronaria, las enfermedades cerebrovasculares, las arteriosis periféricas, la cardiopatía reumática, entre otras.

Los paros cardíacos y los AVC están propiciados, generalmente, por la obstrucción de paredes que impiden que la sangre vaya al corazón o al cerebro. Normalmente, son paredes de grasa las que forman estos bloqueos y son generadas, mayoritariamente, por la combinación de factores de riesgo cardiovascular.

Los factores de riesgo cardiovascular aumentan la probabilidad de que se produzca un problema cardiovascular. Entre los más importantes se encuentran:

- **El colesterol:** si este está en niveles de 240 mg/dl, se conoce como hipercolesterolemia, se dobla la probabilidad de sufrir un infarto de miocardio. Lo normal es un nivel inferior a 200 mg/dl.
- **La diabetes mellitus:** enfermedad producida por la no fabricación de suficiente insulina o porque la insulina no consigue que las células respondan a su estímulo.

Se considera un factor de riesgo muy importante porque la glucosa se queda en la sangre, dañando los vasos sanguíneos y aumentando el riesgo de sufrir angina, infarto agudo de miocardio o incluso muerte cardíaca súbita.

- **Enfermedad periodontal:** producida por un deterioro en la salud de las encías. Está demostrado que existe una relación causal con la aparición de enfermedades ECV.
- **Estrés y ansiedad:** es la tensión que se provoca por el sufrimiento de situaciones de agobio elevado. Por ejemplo, está demostrado que tras un caso de alteración emocional significativa se dobla el riesgo de infarto de miocardio.
- **Falta de ejercicio:** considerado uno de los factores de riesgo más importantes. Este factor aumenta la probabilidad de sufrir arterioesclerosis e hipertensión. Además, influye directamente en el aumento de otros factores de riesgo como la obesidad, la hipertensión o el colesterol.
- **La frecuencia cardíaca:** corresponde a la cantidad de veces que el corazón se contrae a lo largo de un minuto. Para que el organismo funcione correctamente, el corazón debe bombear sangre a una presión arterial y frecuencia correctas. En estado de reposo, lo normal es que la frecuencia esté entre 50 y 100 bpm. El control de este factor es importante, ya que está demostrada su relación con el riesgo de muerte.
- **Hipertensión:** se corresponde al incremento mantenido de la presión arterial. La presión arterial es fundamental para que la sangre llegue a los distintos órganos del cuerpo humano. La hipertensión propicia una mayor resistencia para el corazón, pudiendo causar insuficiencia coronaria y angina de pecho. También afecta al cerebro si se rompe alguna arteria, produciendo un AVC hemorrágico o ictus. Los valores normales están entre 120-129 mmHg para la presión arterial sistólica (máxima) y entre 80 y 84 mmHg para la diastólica (mínima).
- **La mala alimentación, obesidad, tabaquismo y consumo de alcohol abusivo:** la relación de estos factores de riesgo con las ECV es directa. Por tanto, se recomienda una alimentación adecuada, con una dieta cardiosaludable, realizar ejercicio físico de forma regular, y evitar el consumo de alcohol y tabaco, especialmente si se sufre alguna ECV.

Por otro lado, la edad tiene un impacto sobre el riesgo cardiovascular. Al igual que todo el cuerpo humano, el corazón también sufre alteraciones en el transcurso del

envejecimiento, siendo la tercera edad la más vulnerable a sufrir las ECV. A partir de los 40-45 años el riesgo de sufrir algún tipo de ECV se duplica, aproximadamente, cada década. No obstante, la prevención en los factores de riesgo sigue siendo eficaz y fundamental a cualquier edad [8].

Una de las bases de este proyecto es el control continuo de la frecuencia cardiaca. A través del hardware utilizado y un algoritmo, el sistema realiza notificaciones cuando el pulso está demasiado bajo o alto de forma continuada. Después, el usuario debe acudir a un profesional sanitario para realizar un correcto diagnóstico.

2.1.2. IoT

El IoT puede considerarse una infraestructura dinámica y global que controla objetos inteligentes de una manera compleja. Esto, a su vez, permite la interconexión de dispositivos IoT que comparten la información para crear nuevas aplicaciones y servicios que mejoran la calidad de vida de las personas.

Desde su primera definición en 1999, por Kevin Ashton, ha adoptado otras muchas en distintas organizaciones internacionales y de estandarización.

El IoT está creciendo significativamente, de forma que desde 2008 existen más dispositivos conectados que personas en el mundo, con unas previsiones de crecimiento hasta alcanzar 50.000 millones de unidades para 2023, siendo la población mundial cercana a los 7.500 millones de personas.

Se pueden identificar una serie de características comunes al IoT:

- **Gran escala:** el número de dispositivos IoT se incrementa en miles de millones de unidades. Estos dispositivos se tienen que poder conectar con el resto.
- **Inteligencia:** gracias a la integración de algoritmos software y hardware, los dispositivos IoT son inteligentes.
- **Uso de sensores:** imprescindibles para las aplicaciones de IoT. Estos sensores perciben cualquier cambio en el ambiente y generan datos que reflejan el estado en el que se encuentran.
- **Sistema complejo:** existen miles de millones de dispositivos IoT. La realización de operaciones es una tarea ardua, debido a las limitaciones asociadas a la memoria, la energía y el tiempo.

- **Entorno dinámico:** los dispositivos IoT están preparados para cambios continuos en el entorno y la entrada de nuevos dispositivos a la red.
- **Cantidad masiva de datos:** existen miles de millones de dispositivos, cada uno a su vez generando datos. Esto expone ciertas amenazas de seguridad y privacidad.
- **Heterogeneidad:** el sistema IoT involucra dispositivos con distintos sistemas operativos, protocolos de comunicación, etc.
- **Energía limitada:** la mayoría de los dispositivos son pequeños y ligeros, estando configurados para consumir poca energía.
- **Conectividad:** habilita el acceso y compatibilidad a la red [14].

Todos los dispositivos IoT deben estar conectados a la red, por lo que deben usar un protocolo de comunicación. En la Tabla 2.1 se presenta una comparativa entre los distintos protocolos de comunicación utilizados.

Tabla 2.1 COMPARATIVA ENTRE LOS PRINCIPALES PROTOCOLOS DE COMUNICACIÓN IOT [14]

Características	ZigBee	Bluetooth	Z-Wave	NFC	6LoWPAN
Estándar	IEEE 802.15.4	IEEE 802.15.1	Z-Wave	ISO/IEC 18000-3	IEEE 802.15.4
Tipo de red	WPAN	WPAN	WPAN	Red P2P	WPAN
Banda de frecuencia	2.4 GHz	2.4 GHz	900 MHz	13.56 MHz	2.4 GHz
Alcance	10-100 m	15-30 m	30-100 m	Hasta 200 m	100 m
Tasa binaria	250 kbps	1 Mbps	100 kbps	100-420 kbps	250 kbps
Consumo de potencia	Bajo	Bajo	Bajo	Bajo	Muy bajo

Seguridad	AES	AES	AES	RSA y AES	AES
Aplicaciones comunes	Control de la vivienda y monitorización	Auriculares inalámbricos, relojes y pulseras	Control de la vivienda y monitorización	Pagos y accesos	Monitorización y control vía Internet

El IoT se utiliza en distintos ámbitos de la vida cotidiana. Mejorar la calidad de vida de las personas es una de las características del IoT, de ahí su presencia en numerosos campos. Existen soluciones para la automatización agrícola, el consumo de energía, hogares inteligentes, ciudades inteligentes, telemedicina y salud, entre muchos otros. En concreto, el sector de la telemedicina y el cuidado de la salud es el correspondiente al interés de este trabajo.

Este proyecto plantea una solución basada en el intercambio de información entre una pulsera inteligente, una aplicación móvil y sensores en la vivienda. Más adelante se explicará la solución concreta adoptada.

2.1.3. e-Salud

En las últimas décadas, se ha observado un incremento en el uso de las TIC en distintos ámbitos de la sociedad y de los negocios. Concretamente, el entorno sanitario es un pilar de estas tecnologías.

Se conoce como e-Salud (en inglés, *e-Health*) al conjunto de herramientas TIC que se emplean en el sector sanitario para prevención, diagnóstico, seguimiento, tratamiento y gestión de la salud.

Dentro de la e-Salud se encuentran aplicaciones para dispositivos móviles, telemedicina, dispositivos wearables, uso de Big Data para gran cantidad de información, sistemas de apoyo a la decisión clínica o, incluso, videojuegos de salud.

Con la primera definición de e-Salud, en 2001, se declararon una serie de principios inherentes al propio término, que añadían más significado a la letra ‘e’ de e-Salud.

- **Eficiencia:** el uso de e-Salud debe reducir costes. Una posible forma de abaratar costes puede ser evitar diagnósticos innecesarios o duplicados.

- **Mejora de la calidad del cuidado:** mejorar la eficiencia no sólo incluye reducir costes, sino también mejorar la calidad.
- **Basado en la evidencia:** es necesario que la efectividad de las aplicaciones de e-Salud esté probada a través de una rigurosa evaluación científica.
- **Potenciación de los consumidores y pacientes:** haciendo accesibles en Internet los conocimientos básicos sobre medicina y cuidado personal. Gracias a ello, los pacientes tendrán más información que ofrecer a los profesionales sanitarios.
- **Fomentar una nueva relación entre el paciente y el profesional sanitario:** las decisiones se realizan de una manera compartida.
- **Educación a través de fuentes digitales:** tanto extender los conocimientos de los profesionales con recursos online, como sobre educación de salud personalizada para cada consumidor.
- **Habilitar el intercambio de información:** para la comunicación entre centros sanitarios.
- **Ampliar el espectro del cuidado sanitario más allá de los límites convencionales:** desde un significado geográfico como conceptual. Es decir, permitir obtener servicios de cualquier proveedor global y ampliar el rango de complejidad de las intervenciones, diagnósticos, etc.
- **Ética:** e-Salud incorpora nuevas formas de interacción entre el paciente y el profesional haciendo frente a nuevos desafíos como el consentimiento informado, la consulta online, privacidad y problemas de equidad.
- **Equidad:** uno de los principales compromisos de la e-Salud es el de hacer más equitativo el cuidado sanitario. No obstante, existe el riesgo de que aumente la desigualdad, debido a que hay gente que no tiene recursos económicos suficientes, conocimientos digitales básicos o a acceso a un ordenador y/o a la red.

Además de estos 10 principios, la e-Salud debe ser fácil de usar, entretenido y emocionante. Estos sistemas deben ser muy sencillos e intuitivos, evitar un uso aburrido de las TIC y aportar un valor añadido para que se usen de forma cotidiana [10].

El proyecto realizado busca, principalmente, ofrecer una solución intuitiva y fácil de usar tanto para el usuario monitorizado, como para el supervisor de dicho paciente. Además, gracias al bajo coste de los componentes utilizados, es una solución accesible para casi cualquier persona con smartphone.

2.1.4. Tecnología wearable

La tecnología wearable consiste en el uso de componentes electrónicos en accesorios de vestir u objetos “vestibles” que se llevan de forma cotidiana. Esto implica que es un subgrupo dentro del IoT, ya que incorpora objetos físicos con sensores y software que permite el intercambio de datos sin la intervención humana.

Para que un dispositivo se considere wearable, además de ser “vestible” y usarse de forma cotidiana, debe tener uno o más procesadores. Como se puede ver en la Figura 2.1, este grupo está formado por pulseras, anillos, camisetas, relojes, gafas... con el fin de mejorar la calidad de vida de las personas.



Fig 2.1 Tipos de dispositivos wearable [11]

Estos dispositivos inteligentes se emplean en distintas aplicaciones:

- **Gafas inteligentes:** te permiten disponer de información que solicites a través de la voz, grabar vídeos o tomar imágenes que estés viendo en cada instante.
- **Accesorios y complementos:** es el caso de los anillos, pulseras (smartband) y relojes (smartwatches), orientados a mejorar la calidad de vida y a facilitar ciertas

tareas. Estos te dan información de los sensores que incorporan, como puede ser la frecuencia cardiaca, el oxígeno en sangre, los pasos realizados en un día, las calorías quemadas... Además, en algunos dispositivos se pueden realizar pagos sin necesidad de tarjeta de crédito, manejar aplicaciones de mensajería como el correo electrónico, o incluso abrir el coche.

- **Seguridad de los Trabajadores:** existen aplicaciones de la tecnología wearable para mejorar la seguridad y las condiciones de algunos profesionales. Por ejemplo, existe un casco de bombero que te proporciona el nivel de oxígeno, la temperatura y la localización por GPS.
- **Industria Textil:** en la actualidad se están realizando prendas que incorporan algunos sensores que permiten mandar señales al smartphone u otros dispositivos conectados.
- **Moda:** existen vestidos que se encogen, cambian de forma o, incluso, se iluminan. Hay otros que poseen pequeñas placas solares y que te permiten cargar la batería del smartphone.
- **Medicina:** la tecnología wearable también está presente en este campo, con sensores que se introducen dentro del cuerpo y permiten obtener información como, por ejemplo, la cantidad de glucosa del paciente y, en este caso, inyectar la insulina necesaria de forma automática [12].

Como hemos visto, existen numerosas aplicaciones con dispositivos wearables que pretenden facilitar la vida a la sociedad. En la Figura 2.4. se observa como el mercado wearable crece cada año, con unas estimaciones positivas de crecimiento. Esto es debido, en gran parte, a las mejoras técnicas y a que los casos de uso amplían las oportunidades de mercado.

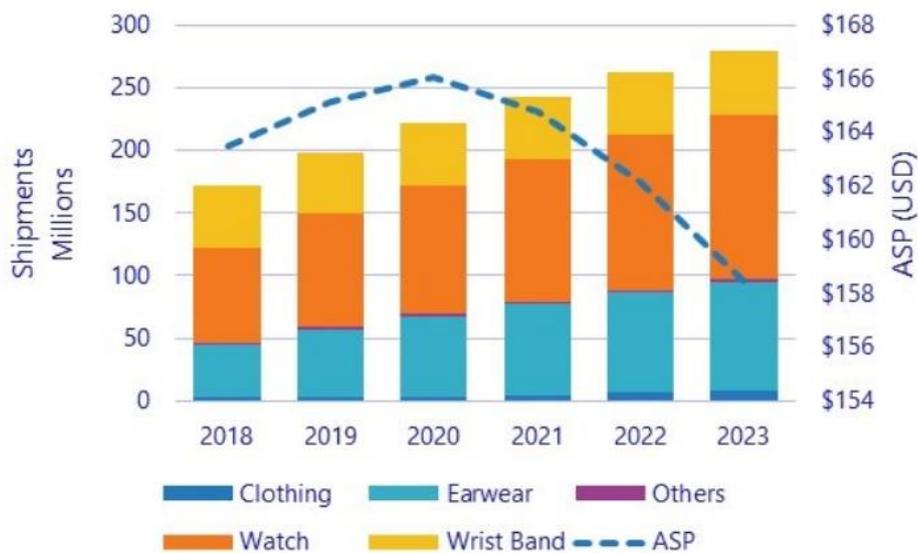


Fig 2.2 Estimación del mercado de los dispositivos wearable hasta 2023 [13]

2.1.5. Smartband

Este proyecto propone una solución basada en una smartband, una de las aplicaciones de la tecnología wearable. A continuación, se expone el estado del arte de estos dispositivos.

Una smartband es un dispositivo que se coloca en la muñeca y que proporciona información acerca de la persona que lo porta, como pueden ser las calorías quemadas, los pasos realizados, entre otras funciones.

Estos dispositivos tienen distintas características y en la mayoría de ellos existen sólo algunas de ellas. A continuación, veremos algunas de ellas:

- **Acelerómetro y GPS:** un acelerómetro o podómetro permite obtener el número de pasos que se realizan a lo largo de una jornada. El GPS ayuda a calcular la distancia recorrida andando o corriendo. Ambos sensores pueden determinar si se está realizando una actividad u otra.
- **Pulso cardiaco:** es una característica que tienen la mayoría de las pulseras. Funciona con un sensor óptico, que ayuda a obtener el pulso cardiaco a través de la luz. El sensor ilumina la piel del usuario, iluminando a su vez los capilares para controlar el flujo sanguíneo y así medir el pulso cardiaco.
- **Sensores UV:** informa sobre la radiación UV que absorbe el sol.
- **Inalámbrico:** la característica más importante de una smartband. Proporciona una experiencia de manos libres y permite al usuario realizar algunas funciones del smartphone sin sostenerlo.

- **Información:** contiene gran cantidad de información. A través de la pulsera, se pueden enviar y recibir mensajes y notificaciones.
- **Asistencia:** una smartband funciona como un asistente para el usuario que la porta, simplificando nuestra vida cotidiana.

Como se puede observar, estas pulseras están destinadas a un público atraído por el fitness y el ejercicio. No obstante, se pueden emplear para otras muchas utilidades.

2.1.6. Sistemas operativos para móviles

Los smartphones son un tipo de teléfono móvil que permite realizar funciones de computación como un ordenador. Esto ha proporcionado numerosas utilidades a lo largo del recorrido de estos dispositivos.

Para su funcionamiento es necesario un sistema operativo que se encargue de los procesos básicos y que permita el funcionamiento de otros programas. Existen números sistemas operativos para estos dispositivos, pero los más importantes son Android e IOS.

En la Figura 2.5 se presenta la distribución mundial de dispositivos según su sistema operativo.

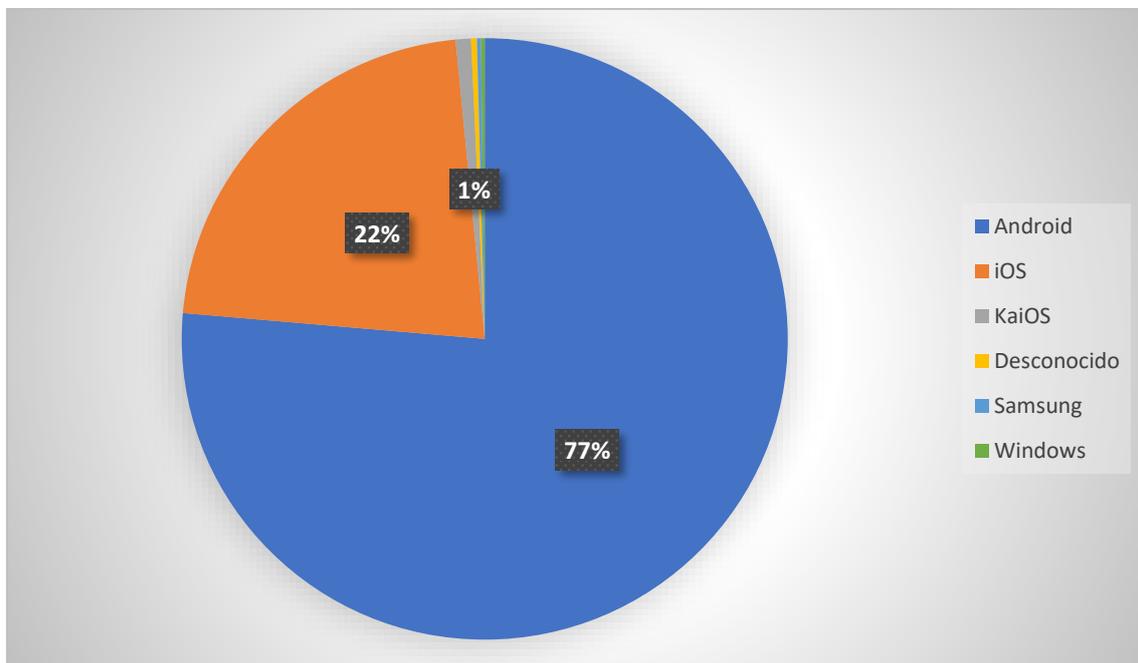


Fig 2.3 Distribución mundial de dispositivos móviles según el sistema operativo [Creación propia [32]]

Como se observa en la gráfica, la mayoría de los dispositivos funcionan bajo el sistema operativo Android. Por tanto, para la realización de este proyecto se ha elegido Android.

A su vez, los desarrolladores de Android han ido realizando actualizaciones desde su primera versión en 2008. Cada una de estas actualizaciones pretende corregir errores y añadir nuevas funcionalidades a la versión anterior.

La versión 1.0 se lanzó en septiembre de 2008 con el nombre de “Apple Pie” y la última hasta la realización de este proyecto es la versión 9.0, “Pie”, lanzada en agosto del 2018 [31].

Como se verá en el próximo capítulo, es necesario ajustar la versión de Android al proyecto que se está realizando, ya que determinadas funcionalidades requieren una versión de Android mínima para funcionar.

2.2. Soluciones existentes

En base al problema planteado, este proyecto busca cubrir las siguientes necesidades de forma conjunta:

- Supervisión continua del pulso cardiaco.
- Alarma en caso de detectarse alguna anomalía en el paciente.
- Seguimiento del paciente en la vivienda a través de sensores.

A continuación, se exponen algunas de las soluciones ya existentes que pretenden cubrir las necesidades anteriormente expuestas:

- **“Rithmi”**: es un wearable que monitoriza el ritmo cardiaco de forma continua, detecta fibrilación auricular, una de las arritmias más comunes y que pueden llegar a provocar un Ictus. No obstante, este proyecto está en búsqueda de inversores y todavía no ha salido al mercado [33].
- **“Monitor de frecuencia cardíaca precisa”**: una aplicación disponible en Play Store y que te permite medir tu pulso cuando tú lo consideres a través de la cámara del teléfono móvil.
- **Teleasistencia en casa**: un producto de Cruz Roja que consta de un pulsador de emergencia, que se lleva en la muñeca o colgado. Cuando el paciente no se encuentra bien presiona el botón y automáticamente se señaliza a los servicios de la entidad para proceder a su asistencia [34].
- **“Apple Watch Series 4”**: este producto de Apple incorpora la posibilidad de realizar un ECG. Una vez realizado el ECG te indica el resultado, detectando

incluso una fibrilación auricular, en cuyo caso es necesario acudir a un profesional sanitario [35].

- **Cámaras de vigilancia para ancianos:** esta solución consiste en colocar cámaras en la vivienda, de tal forma que se puede ver en tiempo real la actividad, a través de alguna aplicación móvil o web.

En la Tabla 2.2 se presenta una comparativa entre las distintas soluciones encontradas.

Tabla 2.2 COMPARATIVA SOLUCIONES ENCONTRADAS

Característica	Rithmi	Monitor de frecuencia cardíaca	Teleasistencia en casa	Apple Watch	Cámaras de vigilancia
Pulso cardíaco	✓	✓		✓	
Continuidad	✓				✓
Alarma	✓		✓		
Seguimiento en la vivienda					✓
Necesidad de dispositivo externo	✓		✓	✓	✓

Por último, tras haberse realizado una comparativa de las soluciones existentes se llega a la conclusión de la necesidad de desarrollar un producto que cubra todas y cada una de las necesidades planteadas en el problema.

3. DESARROLLO DEL PROYECTO

En este capítulo se presentará el desarrollo del proyecto, incluyendo la metodología seguida, las alternativas consideradas antes de realizar el diseño y las distintas partes del proyecto. Antes de realizar la implementación real del sistema a realizar, es necesario realizar el diseño detallado de cada una de las partes.

3.1. Metodología

La metodología seguida para la realización de este proyecto sigue un esquema de desarrollo de software. Concretamente, este proyecto sigue un modelo de prototipos. Primero, se establecen los requisitos mínimos del sistema a desarrollar y luego se suceden las distintas versiones o prototipos que van mejorando la versión anterior, pero siempre cumpliendo los requisitos establecidos.

El proceso de creación de este proyecto se divide en 4 etapas:

1. **Análisis de requerimientos.** Primera etapa del proyecto. Tras haber realizado un estudio de las necesidades a cubrir, se establecen los requisitos que tiene que alcanzar nuestro proyecto. Para poder conseguir los objetivos finales es crucial realizar bien esta fase.
2. **Diseño y estructura.** En esta etapa se realiza un diseño de más alto grado de abstracción, utilizando los Casos de Uso.
3. **Desarrollo e implementación.** Consiste en escribir en código lo que definió en la etapa anterior.
4. **Pruebas.** Consiste en comprobar que la aplicación funciona correctamente y sin errores, según lo establecido en los casos de uso.

A las 4 etapas mencionadas se le añade otra, rediseño del prototipo, que se encarga de incorporar las propuestas de mejora o sugerencias para la siguiente iteración del producto. En este proyecto se ha realizado la primera iteración, siendo posible continuar en el futuro con prototipos mejorados.

3.2. Análisis de requerimientos

En este epígrafe se realiza una descripción de los objetivos y requisitos fundamentales de la aplicación. Estos requisitos están destinados tanto para el cliente como para el desarrollador.

3.2.1. Objetivos del sistema

Primero, se expone la lista de objetivos esenciales de la aplicación para considerar el prototipo como aceptable.

Para definir cada objetivo utilizaremos la siguiente información:

- **Identificador único:** código para identificar el objetivo.
- **Descripción:** breve definición del objetivo.
- **Comentarios:** es un campo opcional. Se puede incluir la prioridad, alguna nota a resaltar, etc.

Tabla 3.1 OBJETIVO 01

SIS-01	Gestión usuarios
Descripción	Para poder utilizar la aplicación es necesario que el usuario se haya registrado. A su vez, los datos manejados en la aplicación deben de estar vinculados al usuario correspondiente.
Comentarios	-

Tabla 3.2 OBJETIVO 02

SIS-02	Gestión de alertas
Descripción	La aplicación debe incorporar un algoritmo para generar alertas en caso de detectarse anomalías en el usuario.
Comentarios	Preferiblemente por SMS.

Tabla 3.3 OBJETIVO 03

SIS-03	Gestión del HR
Descripción	La aplicación debe hacer una lectura del pulso cardiaco a través de la conexión con un sensor externo.
Comentarios	-

Tabla 3.4 OBJETIVO 04

SIS-04	Gestión de un punto de control
Descripción	La aplicación debe indicar cuando el usuario pasa por el punto de control establecido.
Comentarios	-

Tabla 3.5 OBJETIVO 05

SIS-05	Análisis continuo
Descripción	El sistema debe permitir un monitoreo continuo de los parámetros observados.
Comentarios	-

3.2.2. Requisitos Funcionales

Los Requisitos Funcionales (RF) definen las funciones del sistema y sus elementos. De tal forma, en este epígrafe se indica la lista de todas las funciones del sistema con su correspondiente información.

La información se distribuye de la siguiente manera:

- **Identificador único:** código para identificar el requisito.
- **Nombre:** nombre breve y descriptivo.
- **Descripción:** breve descripción del requisito.
- **Comentarios:** campo opcional. Se puede indicar nivel de prioridad o dependencia con otro requisito, si fuera necesario, etc.

Tabla 3.6 REQUISITO FUNCIONAL 01

Identificador único	RF-01
Nombre	Registro de usuario
Descripción	La aplicación tiene que incluir una opción de registro para el usuario y guardar dicho registro en la BD.
Comentarios	-

Tabla 3.7 REQUISITO FUNCIONAL 02

Identificador único	RF-02
Nombre	Inicio de sesión
Descripción	La aplicación tiene que incluir una opción de inicio de sesión para usuarios ya registrados.
Comentarios	-

Tabla 3.8 REQUISITO FUNCIONAL 03

Identificador único	RF-03
Nombre	Cierre de sesión
Descripción	La aplicación tiene que dejar que el usuario cierre de sesión y poder abrir una sesión con un usuario distinto.

Comentarios	-
--------------------	---

Tabla 3.9 REQUISITO FUNCIONAL 04

Identificador único	RF-04
Nombre	Modificar datos del usuario
Descripción	La aplicación debe habilitar la modificación de los datos del usuario.
Comentarios	-

Tabla 3.10 REQUISITO FUNCIONAL 05

Identificador único	RF-05
Nombre	Entrada de contactos de emergencia
Descripción	La aplicación debe habilitar la entrada de contactos de emergencia, los cuales recibirán las alarmas.
Comentarios	-

Tabla 3.11 REQUISITO FUNCIONAL 06

Identificador único	RF-06
Nombre	Alerta a terceros
Descripción	El sistema debe generar enviar una alerta en caso de detectar un problema en el usuario.
Comentarios	-

Tabla 3.12 REQUISITO FUNCIONAL 07

Identificador único	RF-07
Nombre	Lectura del pulso cardiaco
Descripción	La aplicación debe leer el valor del pulso cardiaco obtenido a través de la smartband.
Comentarios	Es necesario haber conectado la smartband previamente.

Tabla 3.13 REQUISITO FUNCIONAL 08

Identificador único	RF-08
Nombre	Actualización del pulso cardiaco
Descripción	Cada nuevo valor del pulso cardiaco debe ser actualizado en la BD.
Comentarios	-

Tabla 3.14 REQUISITO FUNCIONAL 09

Identificador único	RF-09
Nombre	Lectura del punto de control
Descripción	El sistema tiene que leer cuando el usuario pasa por el punto de control en la vivienda
Comentarios	-

Tabla 3.15 REQUISITO FUNCIONAL 10

Identificador único	RF-10
Nombre	Actualización de la información de control
Descripción	Los datos de control deben actualizarse en la BD cada vez que estos sean modificados.
Comentarios	-

Tabla 3.16 REQUISITO FUNCIONAL 11

Identificador único	RF-11
Nombre	Lectura continua de parámetros
Descripción	El sistema debe leer de forma continua los valores tanto del pulso cardiaco como del punto de control.
Comentarios	-

3.2.3. Requisitos No Funcionales

Los Requisitos No Funcionales (RNF) hacen referencia a aspectos indirectos que ha de cumplir la aplicación. Es decir, no son funcionalidades software que ha de tener la aplicación, pero sí aspectos de forma a tener en cuenta.

La información se distribuye de la siguiente manera:

- **Identificador único:** código para identificar el requisito.
- **Nombre:** nombre breve y descriptivo.
- **Descripción:** breve descripción del requisito.
- **Comentarios:** campo opcional. Se puede indicar nivel de prioridad o dependencia con otro requisito, si fuera necesario, etc.

Tabla 3.17 REQUISITO NO FUNCIONAL 01

Identificador único	RNF-01
Nombre	Interfaz sencilla
Descripción	El público al que va dirigida la aplicación es el de personas mayores dependientes y su entorno familiar. De tal forma la interfaz de usuario debe ser fácil de usar e intuitiva.
Comentarios	-

Tabla 3.18 REQUISITO NO FUNCIONAL 02

Identificador único	RNF-02
Nombre	HW para desarrollo
Descripción	PC con sistema operativo Windows (min. Windows 7)
Comentarios	-

Tabla 3.19 REQUISITO NO FUNCIONAL 03

Identificador único	RNF-03
Nombre	HW para ejecución
Descripción	Es necesario disponer de una smartband, un smartphone Android y dispositivo con sensores para la vivienda.
Comentarios	-

Tabla 3.20 REQUISITO NO FUNCIONAL 04

Identificador único	RNF-04
Nombre	Marco legal
Descripción	Cumplir con el RGPD de la UE.
Comentarios	-

Tabla 3.21 REQUISITO NO FUNCIONAL 05

Identificador único	RNF-05
Nombre	Compatibilidad
Descripción	La aplicación debe estar preparada para pantallas de distintas resoluciones.
Comentarios	-

Tabla 3.22 REQUISITO NO FUNCIONAL 06

Identificador único	RNF-06
Nombre	BD confiable

Descripción	La BD a utilizar debe garantizar un correcto funcionamiento del sistema.
Comentarios	-

3.3. Alternativas de diseño

En este apartado se presentan las distintas alternativas de diseño consideradas antes de realizar el diseño y arquitectura.

Las posibles soluciones surgen de la elección del sistema operativo, la base de datos y el HW utilizado para sensor en la vivienda.

3.3.1. Sistema Operativo

En el epígrafe 2.1.5 se presentaba una gráfica dónde se mostraba el porcentaje de dispositivos móviles según su Sistema Operativo. Se observaba que la mayoría de los dispositivos funcionan bajo Sistema Operativo Android, un 77% del total.

Por ello, el sistema operativo elegido es Android, ya que proporciona mayor nivel de penetración, entre otras ventajas. No obstante, el diseño de una aplicación para iOS está dentro de los propósitos a largo plazo, al margen de los objetivos de este trabajo.

En la Tabla 3.22 se observa una comparativa entre el diseño de aplicaciones con Android e iOS.

Tabla 3.23 COMPARATIVA DISEÑO ANDROID E IOS

Característica	ANDROID	IOS
Lenguaje de programación	Java	Objective-C / Swift
Test de la aplicación	Emulador (Máquina Virtual) incluyendo CPU.	Simulador más rápido que el de Android.
Interfaz Gráfica	Archivos XML	Archivos XIB Mayor calidad de las animaciones
Coste	Más barato que iOS, ya que hay mayor demanda de aplicaciones Android.	Algo más caro que Android, debida a la menor demanda de aplicaciones iOS.

App Store	Play Store	App Store
Monetización	La mayoría son de descarga gratuita, incluyendo publicidad.	Descarga no gratuita o compras dentro de la app.
[Creación propia [36]]		

3.3.2. Base de Datos

Para el desarrollo de la aplicación se necesitan almacenar distintos valores de cada usuario. A continuación, se presentan distintas alternativas.

Una posible implementación consiste en usar una base de datos relacional local en el dispositivo, utilizando SQLite. Esta solución permite guardar datos en la memoria local del dispositivo sin necesidad de tener que utilizar servidores para el almacenamiento de estos datos, siendo una solución gratuita. No obstante, utilizar SQLite tiene distintas limitaciones como el acceso a los datos por terceros, ya que no están disponibles en la nube.

Otra posible solución consiste en el uso de una BD cliente-servidor. Esta solución requiere de mayor inversión, ya que se tendría que alquilar o comprar un servidor para los datos. De esta forma se podrían disponer de los datos a través de consultas a la BD.

La última alternativa consiste en utilizar Firebase, un servicio de Google que te cubre todas las necesidades del backend de la aplicación, junto con otras muchas utilidades que te brindan oportunidades de crecimiento. Te proporciona servicio de BD en tiempo real, servicios analíticos, autenticación del usuario, incorporación de anuncios, etc. Además, el coste de este servicio depende de las necesidades a cubrir, pudiendo ser gratuito bajo ciertas circunstancias.

Tras analizar las diferentes soluciones y necesitando disponer de los datos en la nube, se concluye en que Firebase es la plataforma que más conviene para el desarrollo de este proyecto.

3.3.3. HW para la vivienda

Para el sistema a desarrollar se necesita de HW que incorpore sensores para enviar datos en tiempo real a la aplicación. Concretamente, se necesita un sensor de presencia que detecta cuando el usuario pasa por delante del punto de control.

En este epígrafe se plantean dos alternativas de HW para elegir el procesador: Arduino y Raspberry Pi. En la Tabla 3.23 se expone una comparativa entre ambos dispositivos. En las Figuras 3.1 y 3.2 se pueden ver las placas Arduino y Raspberry Pi 3, respectivamente.



Fig 3.1 Placa Arduino [17]



Fig 3.2 Raspberry Pi 3 Model B [18]

Tabla 3.24 COMPARATIVA ARDUINO VS RASPBERRY PI

	ARDUINO	RASPBERRY PI 3
Ventajas	<ul style="list-style-type: none"> • Facilidad para conectar sensores analógicos, motores y otros componentes electrónicos. • Pueden añadirse muchos complementos para mejorar su funcionalidad. • No necesita de una ardua configuración inicial. • Más barata que la placa Raspberry Pi. 	<ul style="list-style-type: none"> • Procesador más robusto y fuerte, multitarea. • Puerto Ethernet, Wi-Fi y Bluetooth disponible. • Se pueden instalar distintos sistemas operativos. • Puertos USB, para cámara, HDMI... • Adecuado para proyectos multitarea y con necesidad de conexión online.

Desventajas	<ul style="list-style-type: none"> • No puede realizar múltiples tareas. • Para conectarse a Internet necesita de un complemento (aumento de coste) 	<ul style="list-style-type: none"> - Configuración inicial más compleja, necesitando monitor con HDMI, teclado... - La placa es algo más cara que Arduino.
[Creación propia [37]]		

Tras analizar ambas opciones, se determina que es preferible el uso de la placa de Raspberry Pi para el proyecto que estamos realizando. La principal razón es que se necesita conectar la placa a Internet, siendo más práctica la Raspberry Pi porque ya incorpora el modo de conexión. En segundo lugar, la capacidad multitarea de la Raspberry Pi proporciona al proyecto mayor grado de libertad para incorporar nuevas funcionalidades paralelas. Por ejemplo, mientras un proceso puede estar detectando la presencia en una sala, otro puede encargarse de grabar vídeo a través de una cámara.

Inicialmente, la placa Raspberry sólo incorporará un sensor de distancia ultrasónico HC-SR04, véase la Figura 3.3, que permitirá detectar cuando la persona ha pasado por el



Fig 3.3 Sensor de distancia ultrasónico HC-SR04 [19]

punto de control situado en un lugar la vivienda. Esto se ha determinado de esta manera porque la aplicación está destinada para personas que viven solas en sus hogares. Gracias a este sensor, la aplicación indica al supervisor la última vez que el paciente pasó por ese lugar.

Otra posible implementación, y que podría realizarse sin la necesidad de estar una única persona, sería utilizar sensores RFID también compatibles con la placa de Raspberry Pi. No obstante, por simplicidad de HW se ha decidido usar el sensor ultrasónico citado anteriormente.

3.3.4. Smartband

Uno de los elementos principales del sistema es el dispositivo encargado de proporcionar la información de la actividad cardiaca del paciente. Por ello, es necesario elegir el dispositivo que mejor se adapte a las necesidades.

Para realizar una elección adecuada, es necesario conocer las características que el dispositivo debe tener. Estas características son presentadas a continuación:

- **Lectura del pulso cardiaco:** debido a que se requieren los valores del pulso cardíaco, se necesita un sensor que te los proporcione.
- **Gran autonomía:** el dispositivo será usada por personas mayores, menos familiarizadas con las tecnologías que personas de otros rangos de edad. Además, al realizar un monitoreo continuo el consumo de batería será mayor que el habitual.
- **Bajo coste:** el precio es un factor significativo. Si el precio es más bajo, será más fácil que los usuarios dispongan de estos dispositivos. Además, un precio más bajo reduce el gasto en material en el presupuesto de este proyecto.
- **Disponibilidad de los datos de la aplicación:** para poder intercambiar datos entre la aplicación y el dispositivo, es necesario que exista software libre capaz de realizarlo.

Tras realizar una búsqueda de dispositivos que permitan cumplir con los requerimientos, se presentan las dos alternativas:

- **Polar H7:** banda que se coloca en el pecho, te proporciona datos del pulso cardiaco. Además, existe en la web software disponible para realizar la conexión y el intercambio de datos. La autonomía es cercana a las 8 horas y el precio está en torno a los 50 €.
- **Xiaomi MI Band 2:** smartband de Xiaomi. Existen trabajos realizados que te permiten la conexión y el intercambio de datos entre el dispositivo y la aplicación. La autonomía puede superar los 20 días de uso y su precio está cercano a los 16 €.

Para este proyecto se ha decidido utilizar la Xiaomi MI Band 2 por diversas razones. Principalmente, la smartband es más sencilla de colocar que la banda Polar H7, situada en el pecho. Además, la Xiaomi MI Band 2 es más barata y un gran número de usuarios disponen de ella actualmente, permitiendo un mayor nivel de penetración en el mercado.

3.4. Diseño y arquitectura

Tras haber realizado un análisis de los requerimientos del sistema y planteado las posibles soluciones se presentará el diseño y la arquitectura del proyecto.

En la Figura 3.4 se puede observar el esquema global del sistema, incluyendo los distintos elementos y la interconexión entre ellos.

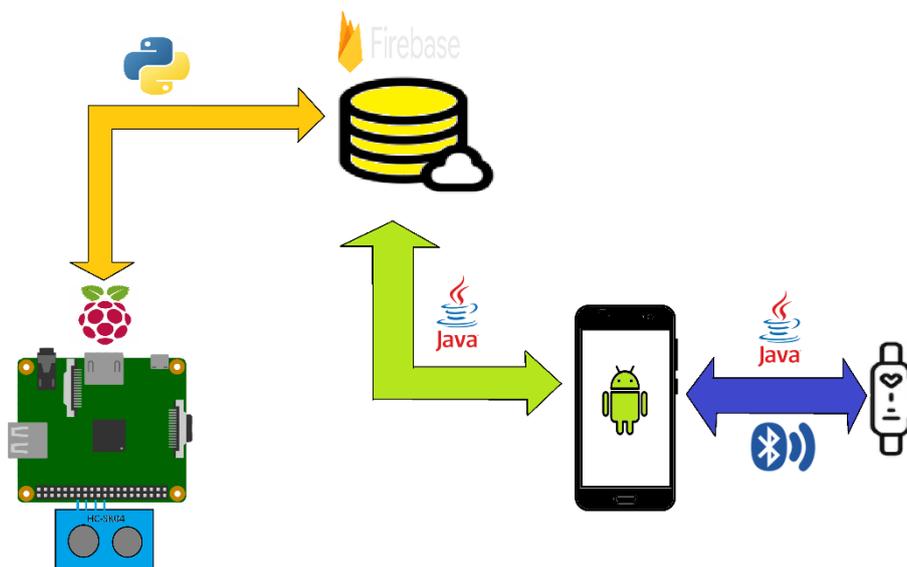


Fig 3.4 Vista global del sistema

La parte de diseño y arquitectura es la última previa al desarrollo e implementación. Esta parte es esencial para relacionar los objetivos del sistema con los casos de uso y de tal forma poder realizar correctamente la siguiente fase del proyecto.

3.4.1. Casos de uso

Un caso de uso es un término de Ingeniería de Software que describe cómo un usuario utiliza el sistema para conseguir un determinado objetivo.

A continuación, se explicará cada uno de los casos de uso de forma detallada. De realizar correctamente cada caso de uso depende directamente el éxito del proceso.

Dentro de cada caso de uso se debe presentar la siguiente información:

- **Identificador único:** código asociado al caso de uso.

- **Objetivo Asociado:** objetivo del sistema, expuesto en el epígrafe 3.2.1, asociado al caso de uso en cuestión.
- **Nombre:** Palabras claves del caso de uso.
- **Precondición:** mínimos necesarios para poder empezar con este caso de uso.
- **Flujo básico:** secuencia de los pasos que se darían en caso de cumplir todos los requisitos (caso ideal).
- **Flujo alternativo:** secuencia de todos los caminos alternos que se podrían dar.
- **Postcondición:** actividad sucesiva a la actual, en caso de acontecerse el flujo básico.

Tabla 3.25 CASO DE USO 01

Identificador único	CDU-01
Objetivo Asociado	-
Nombre	Empezar
Precondición	El usuario tiene que abrir la aplicación.
Flujo básico	Se abre la aplicación.
Flujo alternativo	-
Postcondición	Se muestra la pantalla de inicio de sesión.

Tabla 3.26 CASO DE USO 02

Identificador único	CDU-02
Objetivo Asociado	Gestión usuarios
Nombre	Iniciar sesión
Precondición	El usuario ya tiene una cuenta, pero no ha iniciado sesión aún.
Flujo básico	<ol style="list-style-type: none"> 1. Se introducen los datos. 2. Se inicia sesión.
Flujo alternativo	<ol style="list-style-type: none"> 1. Se introducen los datos de forma errónea. 2. Se indica en la pantalla que no se ha realizado correctamente. 3. El usuario vuelve a introducir sus datos de forma correcta.
Postcondición	<p>Si es la primera vez que se accede a la aplicación, se muestra una configuración inicial.</p> <p>Si no es la primera vez, se va al panel de control correspondiente al modo configurado previamente.</p>

Tabla 3.27 CASO DE USO 03

Identificador único	CDU-03
Objetivo Asociado	Gestión usuarios
Nombre	Registrar usuario
Precondición	El usuario no tiene una cuenta registrada.
Flujo básico	<ol style="list-style-type: none"> 1. Se introducen los datos 2. Se registra el usuario 3. Se inicia sesión automáticamente.
Flujo alternativo	<ol style="list-style-type: none"> 1. Se introducen datos no válidos. 2. Se indica en la pantalla que no se ha podido realizar el registro. 3. El usuario vuelve a introducir sus datos de forma correcta.
Postcondición	<p>Si es la primera vez que se accede a la aplicación, se muestra una configuración inicial.</p> <p>Si no es la primera vez, se va al panel de control correspondiente al modo configurado previamente.</p>

Tabla 3.28 CASO DE USO 04

Identificador único	CDU-04
Objetivo Asociado	Gestión de alertas
Nombre	Configuración
Precondición	El usuario ha iniciado sesión
Flujo básico	<ol style="list-style-type: none"> 1. El usuario elige el modo. Puede ser modo supervisor o modo usuario. 2. El usuario añade los teléfonos de emergencia. 3. Se guardan los datos en Firebase.
Flujo alternativo	<ol style="list-style-type: none"> 1. El usuario no ha elegido ningún modo. 2. Se indica en la pantalla que se debe elegir un modo u otro. 3. El usuario elige rellena los datos correctamente.
Postcondición	<p>Si el modo elegido es usuario, se vería una lista de los dispositivos bluetooth cercanos.</p> <p>Si el modo elegido es supervisor, se va al panel de control del supervisor.</p>

Tabla 3.29 CASO DE USO 05

Identificador único	CDU-05
Objetivo Asociado	<ul style="list-style-type: none"> • Gestión del HR • Gestión de un punto de control
Nombre	Panel de control Supervisor
Precondición	El usuario ha seleccionado el modo supervisor en la configuración.
Flujo básico	<ol style="list-style-type: none"> 1. El usuario refresca los datos. 2. Se leen los datos de Firebase y se muestran en la pantalla. <ul style="list-style-type: none"> • Se muestra la fecha y hora en la que el usuario pasó por última vez por el punto de control. • Se muestra el valor actual del HR.
Flujo alternativo	<ol style="list-style-type: none"> 1. El usuario no pulsa el botón de refrescar los datos. 2. Los datos mostrados puede que no sean los últimos guardados en Firebase. 3. El usuario refresca los datos correctamente.
Postcondición	El usuario puede acceder a otras funcionalidades de la aplicación, como la configuración o cerrar sesión.

Tabla 3.30 CASO DE USO 06

Identificador único	CDU-06
Objetivo Asociado	-
Nombre	Listado de dispositivos BLE
Precondición	El usuario ha seleccionado el modo usuario y ha aceptado los permisos requeridos por la aplicación.
Flujo básico	<ol style="list-style-type: none"> 1. Visualización de la lista de dispositivos. 2. El usuario selecciona el correspondiente a Mi Band 2. 3. Se guarda la dirección MAC para posteriormente poder conectarse a la smartband.
Flujo alternativo	<ol style="list-style-type: none"> 1. El usuario selecciona un dispositivo BLE distinto a Mi Band 2. 2. Se muestra en la pantalla que el dispositivo seleccionado no es correcto y se mantiene la misma pantalla.

	3. El usuario selecciona correctamente la banda MiBand2
Postcondición	Se abre el panel de control del modo usuario.

Tabla 3.31 CASO DE USO 07

Identificador único	CDU-07
Objetivo Asociado	<ul style="list-style-type: none"> • Gestión del usuario • Gestión de un punto de control • Análisis continuo
Nombre	Panel de control Usuario
Precondición	Se tiene que haber seleccionado correctamente la smartband Mi Band 2 del listado de dispositivos BLE.
Flujo básico	<ol style="list-style-type: none"> 1. Se conecta la pulsera 2. Se pulsa la opción de comenzar monitoreo continuo. 3. Empiezan a leerse continuamente datos de la smartband. 4. Se actualizan los datos en la base de datos.
Flujo alternativo	-
Postcondición	El usuario puede terminar la medición y salir de la aplicación cuando quiera.

Tabla 3.32 CASO DE USO 08

Identificador único	CDU-08
Objetivo Asociado	Gestión usuarios
Nombre	Cerrar sesión
Precondición	El usuario ha iniciado previamente una sesión.
Flujo básico	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de cerrar sesión. 2. Se cierra la sesión del usuario.
Flujo alternativo	-
Postcondición	Se vuelve a la ventana de inicio de sesión.

3.4.2. Diagramas de Actividad

Los diagramas de actividad son representaciones gráficas de flujos de tareas. En UML, los diagramas de actividad pretenden modelar tanto los procesos computacionales, los procesos organizativos como los flujos de datos intercambiados entre las distintas actividades. Aunque los diagramas de actividad muestran principalmente el flujo de

control, también pueden incluir elementos que muestran el intercambio de datos entre distintas actividades y fuentes de datos.

En la Figura 3.5 se presentan los distintos elementos que se utilizan para elaborar los diagramas de actividad de este proyecto. Los rectángulos blancos indican las acciones que realiza internamente la aplicación. Por otro lado, los rectángulos rosados presentan las acciones realizadas por el usuario que dan lugar a otras actividades. Por último, los rectángulos azules representan la pantalla concreta de la aplicación en la que se estaría en ese momento de la secuencia.



Fig 3.5 Elementos diagrama de actividad

A continuación, en las siguientes Figuras, se muestran los diagramas de actividad que componen el funcionamiento general de la aplicación. En la Figura 3.6, se presenta en diagrama de actividad correspondiente a las tareas de registro e inicio de sesión por parte del usuario. En la Figura 3.7, se muestra el diagrama a cerca de la configuración inicial del usuario, dónde este debe seleccionar el modo e insertar los valores requeridos por la aplicación. En la siguiente Figura, 3.8, viene el diagrama correspondiente al scanner bluetooth, entre otras tareas. Por último, en las Figuras 3.9 y 3.10, se presentan los diagramas de actividad correspondientes al panel de control del paciente y del supervisor, respectivamente.

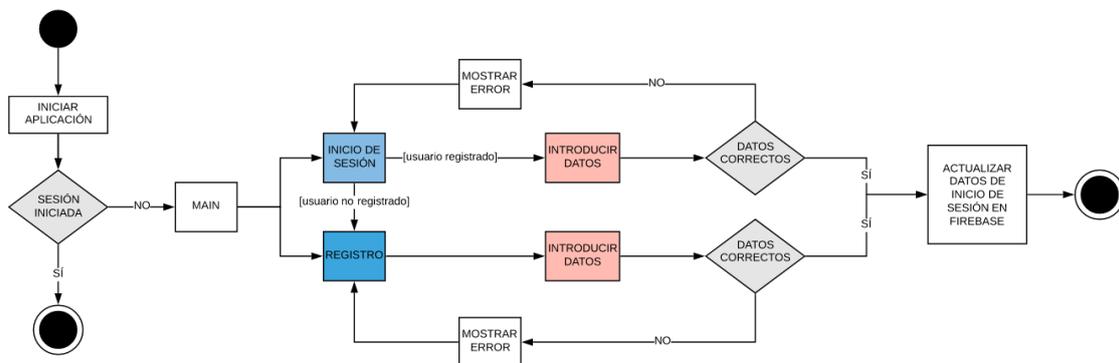


Fig 3.6 Diagrama de actividad: Registro e inicio de sesión

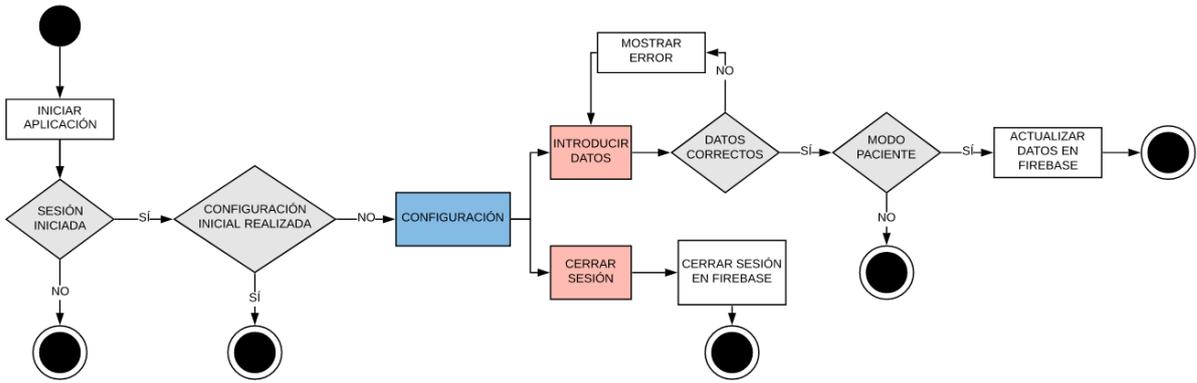


Fig 3.7 Diagrama de actividad: Configuración

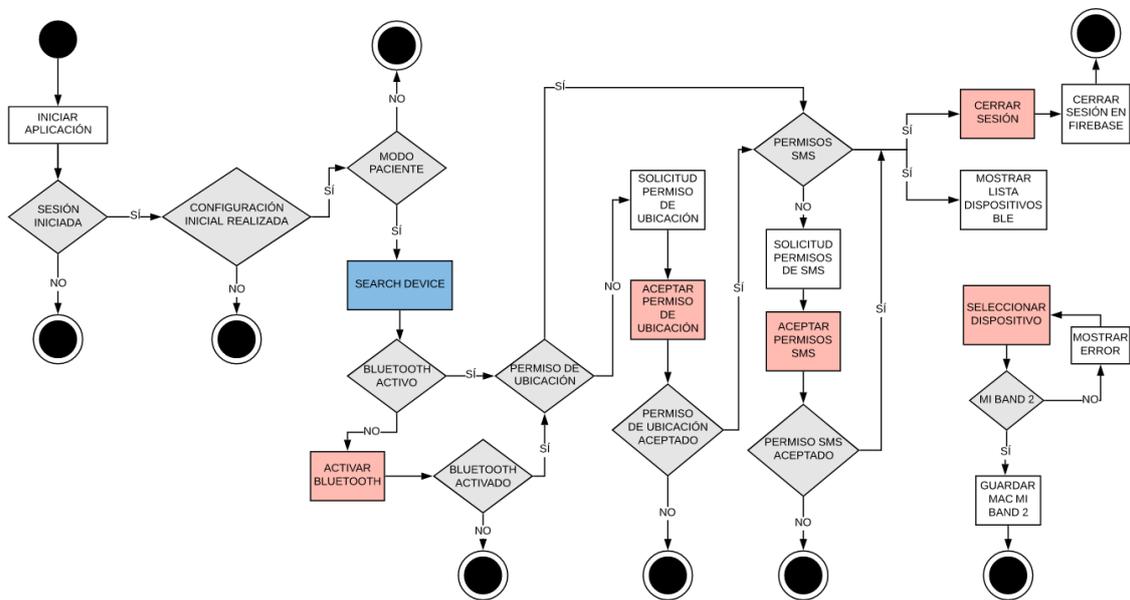


Fig 3.8 Diagrama de actividad: Bluetooth

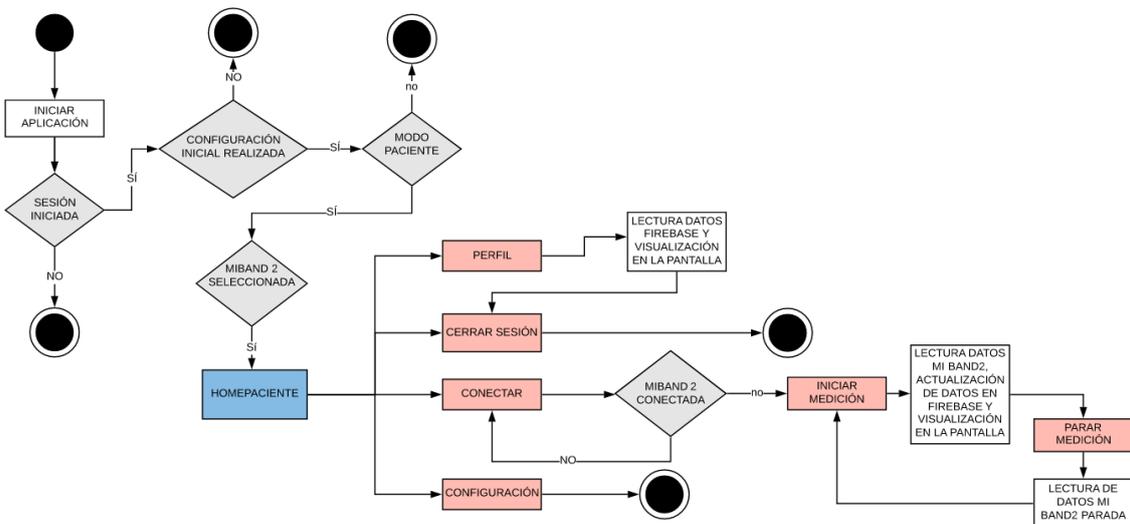


Fig 3.9 Diagrama de actividad: Panel de control Paciente

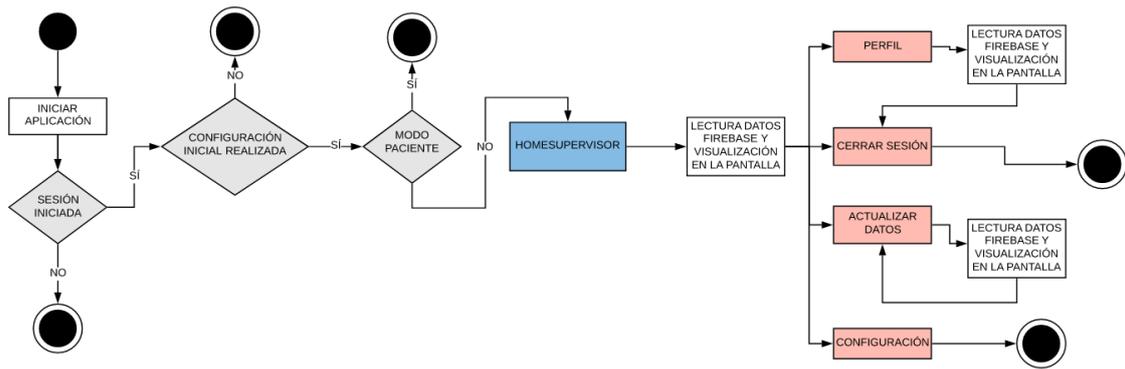


Fig 3.10 Diagrama de actividad: Panel de control Supervisor

3.4.3. Prototipos

En este epígrafe se presentan los prototipos de la aplicación Android. Estos prototipos están basados en las especificaciones y en los casos de uso, y gracias a ellos los desarrolladores disponen de una mayor claridad para realizar la implementación.

A continuación, se muestran los prototipos de las distintas actividades de la aplicación. En la parte izquierda de la Figura 3.11 se muestra el prototipo de la pantalla de inicio de sesión y a la derecha, a su vez, la pantalla de registro del usuario. Una vez se ha iniciado sesión, si es la primera vez que se utiliza la aplicación se presenta una vista de configuración, cuyo prototipo se corresponde a la Figura 3.12 (izquierda). Tras completar la configuración tenemos dos posibilidades:

- **Si se ha pulsado el Modo Paciente:** la siguiente pantalla que se vería sería la de la Figura 3.12 (derecha), seguida de la Figura 3.13 (izquierda), siempre que se haya seleccionado correctamente un dispositivo de la lista.
- **Si se ha pulsado el Modo Supervisor:** la siguiente pantalla que se vería sería la de la Figura 3.13 (derecha), correspondiente al panel de control del supervisor.

Cuando el usuario se encuentra tanto en la pantalla de control del modo paciente o modo supervisor, tiene la opción de visualizar los datos de su perfil, cuyo prototipo se corresponde a la Figura 3.14.

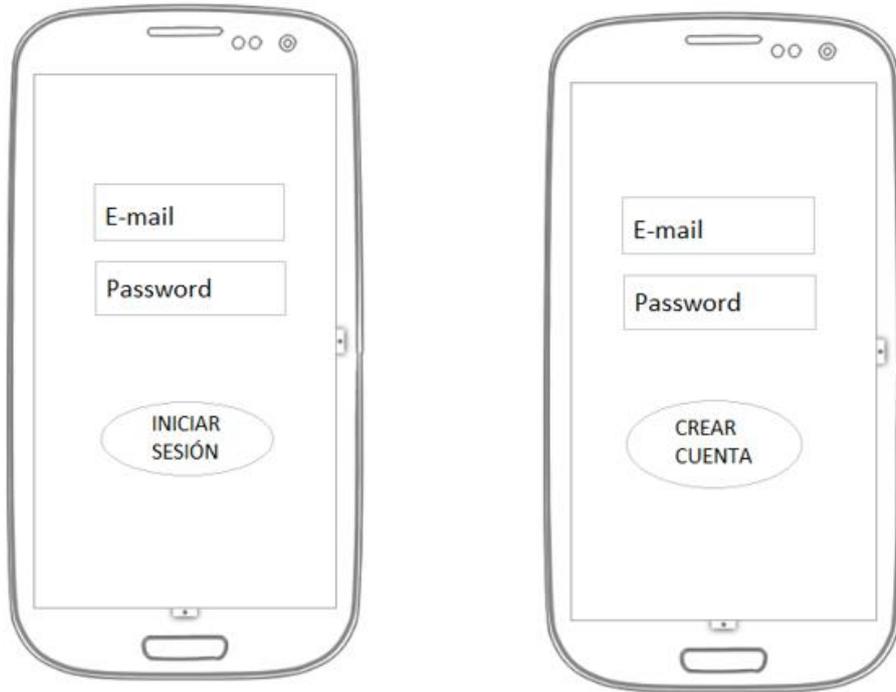


Fig 3.11 Prototipo Inicio de sesión y Registro



Fig 3.12 Prototipos Configuración y Escáner

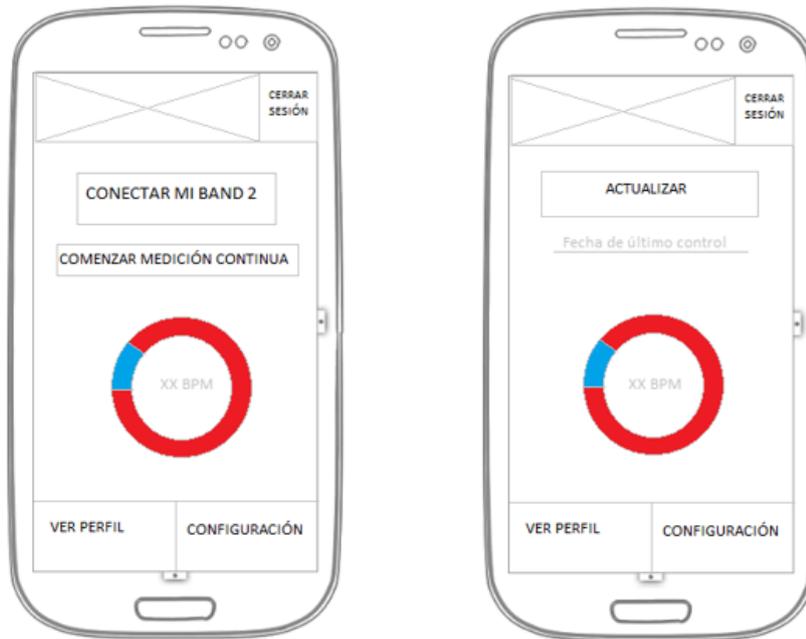


Fig 3.13 Prototipo Panel de Control Paciente y Supervisor



Fig 3.14 Prototipo Perfil de Usuario

3.4.4. Diagrama Entidad-Relación

Un diagrama de Entidad-Relación permite visualizar la estructura de los distintos datos guardados en la BD del sistema. Cada conjunto representa una idea u objeto del que se

derivan distintos valores. A su vez, este diagrama sirve para mostrar la relación existente entre los distintos bloques.

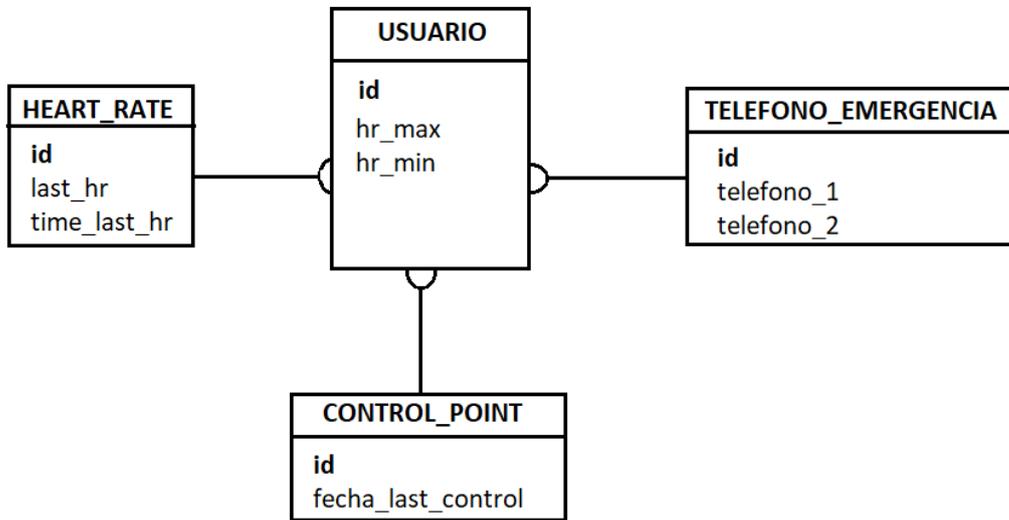


Fig 3.15 Diagrama de Entidad-Relación

En la Figura 3.15 se muestra el diagrama de Entidad-Relación de nuestro sistema. Como se puede ver, los distintos bloques confluyen en que todos los datos brindan información acerca del usuario. Esto no podría ser de otra manera, ya que la aplicación que se presenta en este trabajo tiene como misión proporcionar los datos del paciente a gente de su entorno para su mejor control.

3.5. Desarrollo e implementación

En este apartado se presenta el desarrollo e implementación del proyecto. Una vez se ha realizado el diseño falta llevarlo a la práctica. El sistema se divide en dos partes bien diferenciadas, la aplicación Android y la Raspberry con sensor. En primer lugar, se trata el bloque de la aplicación Android, desde unas pautas introductorias hasta el diseño final. En segundo lugar, se presenta la implementación del circuito con sensor ultrasónico en la Raspberry. Una vez realizados ambos bloques se da por finalizado el desarrollo.

3.5.1. Aplicación Android

El primer bloque del sistema a implementar es el formado por la aplicación Android. Este bloque está formado por las conexiones aplicación-smartband y aplicación-firebase. A continuación, se introducen aspectos acerca de Android Studio, la configuración de Firebase, la conexión con la smartband y la parte gráfica de la aplicación.

3.5.1.1. Android Studio

El primer paso para desarrollar la aplicación en Android es disponer de acceso al SDK de Android, que permite la comunicación entre el SW desarrollado y el sistema operativo. Este SDK está disponible para versiones desde Windows 7 y MacOS 10.6 y ofrece el uso de Android Studio.

Android Studio es el IDE oficial de Android, construido sobre otro IDE, denominado IntelliJ IDEA y diseñado específicamente para desarrollo Android. Este programa está disponible para Windows, macOS y Linux y reemplaza a Eclipse como SW para el desarrollo de este tipo de aplicaciones. La última versión de Android Studio (v3.5) fue lanzada en Agosto de 2019, siendo la versión anterior (v3.4) la utilizada para el desarrollo de este proyecto. Los lenguajes de programación disponibles para versiones superiores a la v3.0 son Kotlin y Java [26]. En este proyecto se ha optado por usar Java, debido a que existe más documentación en este lenguaje, a pesar de que “Kotlin” sea actualmente el lenguaje oficial. Por otro lado, para la parte gráfica de la aplicación se utiliza XML, aunque Android Studio dispone de la posibilidad de realizarla manualmente, sin necesidad de escribir el código directamente.

Una vez instalado Android Studio, ya se puede realizar la implementación de la aplicación Android. Al iniciar el programa se ve una pantalla como la Figura 3.16. En la parte superior tenemos los distintos menús (File, Edit, View...) que proporcionan distintas funcionalidades. En la parte izquierda se pueden ver los elementos del proyecto que se está realizando, por ejemplo, en el directorio “java” es donde se encuentra nuestra parte lógica. En la parte central-derecha tenemos el editor de texto, donde se escribe el código. Por último, en la parte inferior se nos mostrará el progreso de las distintas tareas y los mensajes de depuración, entre otras funcionalidades.

La versión de API Android, indica la versión mínima que los dispositivos deben tener para poder utilizar la aplicación. En este caso se ha utilizado la API 23, la cual nos permite disponer de las funcionalidades utilizadas en esta aplicación, como servicios de Firebase. En el momento en el que se desarrolla este documento, el nivel de API 23 permite ser usado en el 62,6 % de los dispositivos, según Android Studio.

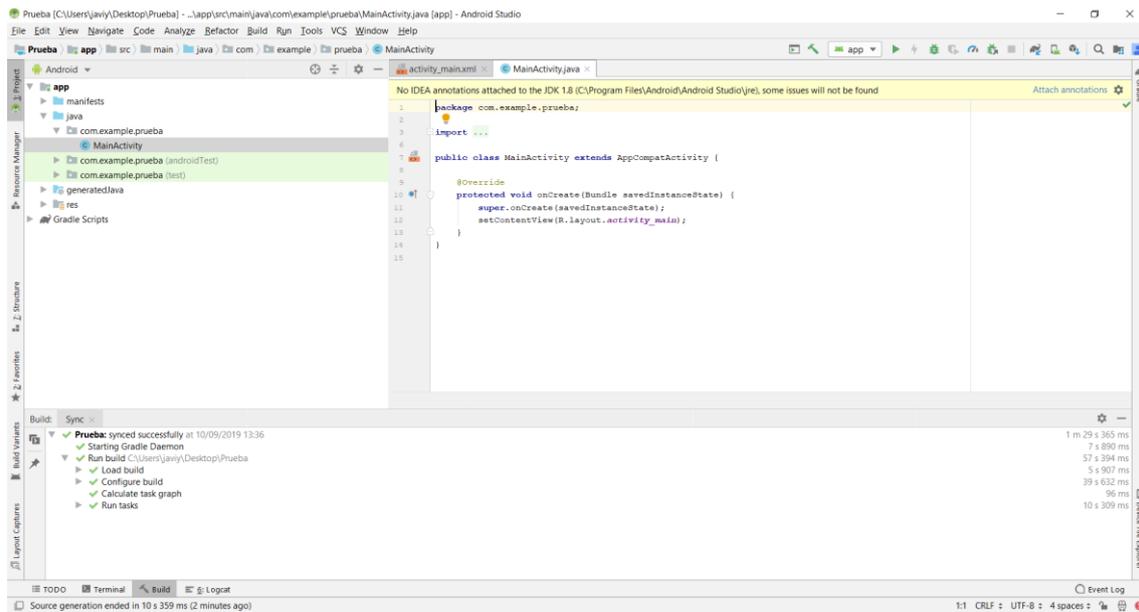


Fig 3.16 Vista inicial Android Studio

3.5.1.2. Android Manifest

Un proyecto Android está formado por numerosos archivos, desde los que proporcionan la lógica o gráfica de la aplicación hasta los que indican distintas propiedades de la aplicación. El archivo “AndroidManifest.xml” proporciona información en formato XML acerca de la aplicación, las distintas actividades o incluso de los permisos que debe tener para su correcto funcionamiento [27].

Una de las principales funciones de este archivo es declarar los permisos que necesita la aplicación para poder entrar en alguna de las partes protegidas del sistema. En la Figura 3.17 se muestra un fragmento del archivo de la aplicación desarrollada. En primer lugar, nuestra aplicación necesita tener acceso a Internet para poder disponer de los servicios de autenticación y base de datos de Firebase, de ahí el primer permiso. Por otro lado, para poder realizar la conexión con la pulsera es necesario tener los permisos asociados a la utilización del protocolo Bluetooth y el permiso de localización para poder realizar el escáner de dispositivos. Por último, para que la aplicación pueda enviar mensajes de texto SMS se requiere del permiso que lo habilita.

No obstante, en el archivo AndroidManifest.xml hay más información aparte de los permisos, como el nombre del paquete del proyecto o la actividad principal que ha de ser lanzada primero.

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.SEND_SMS" />
```

Fig 3.17 Android Manifest: declaración de permisos

3.5.1.3. Firebase: autenticación y BD en tiempo real

Como se ha visto a lo largo de este documento, una de las partes que conforman este proyecto consiste en utilizar algunos de los servicios de Firebase. En particular, para la aplicación implementada se requiere de las funcionalidades de autenticación y base de datos en tiempo real.

Gracias a la autenticación se consigue identificar a los distintos usuarios y poder tener una referencia a cada uno de ellos, gracias al UID que se genera automáticamente. La base de datos en tiempo real es necesaria para almacenar los distintos datos del usuario en la nube y que el supervisor pueda acceder a ellos desde otro terminal. Para poder incorporar estas funciones en la aplicación se necesita disponer de un proyecto en Firebase. No obstante, Android Studio facilita el uso de este servicio. En la parte superior del programa se encuentra la pestaña “Tools” que incorpora un asistente de Firebase. Este asistente presenta cada uno de los servicios Firebase que se pueden implementar, aunque en el caso de este proyecto sólo se ha usado “Authentication” y “Realtime Database”. Como se puede observar en la Figura 3.18, es necesario conectar la aplicación a Firebase y luego añadir el servicio correspondiente. Al hacer click sobre “Connect to Firebase” hay que iniciar sesión con una cuenta de Google para poder crear un proyecto nuevo. En caso de ya tener algún proyecto creado en Firebase, también se puede vincular directamente con la aplicación. El segundo paso se hace automáticamente, es decir, Android Studio se encarga de realizar todas las configuraciones necesarias en el proyecto tras pulsar el botón. Este paso se repite para los dos servicios que se usan.

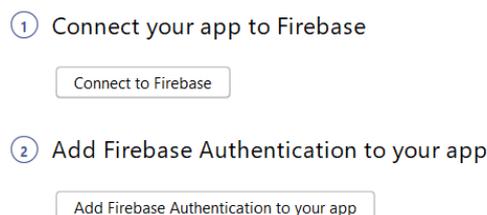


Fig 3.18 Asistente de Firebase en Android Studio

En primer lugar, se presenta la implementación relativa al servicio de autenticación. Para que el usuario pueda autenticarse e iniciar sesión en Firebase es necesario estar previamente registrado. En las clases Register.java y Login.java se han implementado las lógicas de las actividades que se encargan del registro e inicio de sesión, respectivamente. En las Figuras 3.19 y 3.20 se pueden ver las pantallas de registro e inicio de sesión, respectivamente.

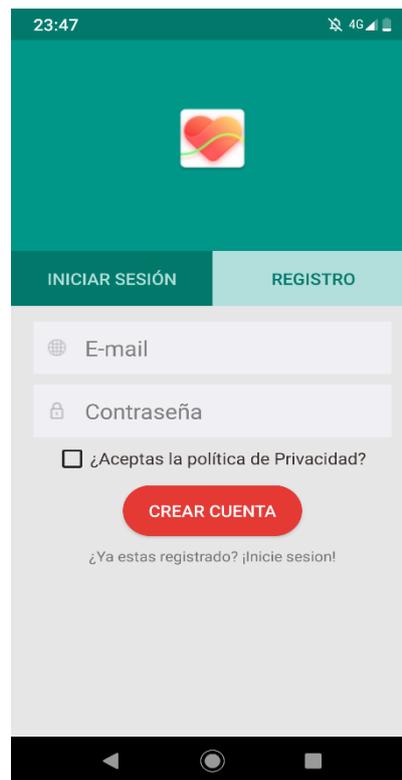


Fig 3.19 Captura de pantalla: Registro

Para las distintas tareas de autenticación es necesario declarar e inicializar una variable de la clase FirebaseAuth, la cual permite llamar a las funciones de crear usuario e iniciar sesión. Para registrar el usuario con un email y contraseña, como se ve en la Figura 3.19, se llama a la función “createUserWithEmailAndPassword (String email, String passwd)” cuando se pulsa el botón rojo y, además, los campos han sido rellenados correctamente. En caso contrario, si los campos no han sido rellenados o no se ha aceptado la política de privacidad, se muestra un mensaje de error y no se realiza el registro. Para realizar el inicio de sesión, véase la Figura 3.20, se hace una llamada a la función “signInWithEmailAndPassword(String email, String passwd)” cuando los campos se han rellenado correctamente, al igual que para el registro [28].

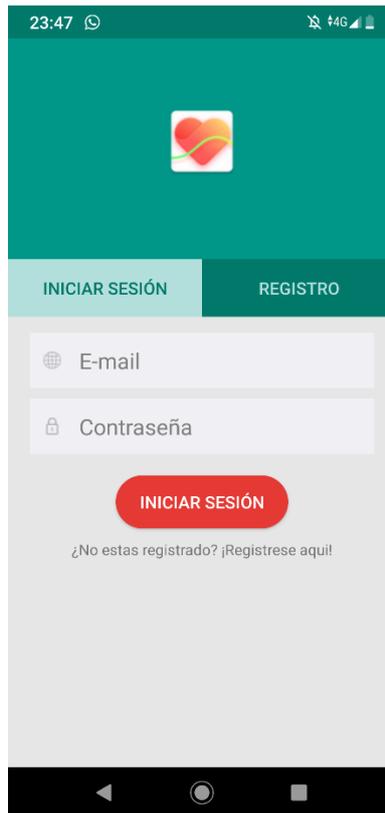


Fig 3.20 Captura de pantalla: Iniciar Sesión

Cada usuario que se registra en Firebase se puede ver desde la consola, donde se pueden eliminar usuarios registrados o añadir otros manualmente. En la Figura 3.21 se muestra la información que proporciona Firebase acerca de los usuarios registrados.

Identificador	Proveedores	Fecha de creación	Inicio de sesión	UID de usuario ↑
javiyus.97.jy@gmail.com	✉	5 sept. 2019	7 sept. 2019	EKa0aerUUxbdzYJE3eaMt1eYB3
100346810@alumnos.uc3m....	✉	5 sept. 2019	5 sept. 2019	nYBXP0CjjbgMIQMtcF11UrwaA8v2
javi.yuste@hotmail.com	✉	4 sept. 2019	6 sept. 2019	socQFXU2IWUFZU0JPetQS3B8ww...

Filas por página: 50 1-3 de 3

Fig 3.21 Vista de los usuarios registrados en la consola de Firebase

Por último, para cerrar sesión con la cuenta de Firebase se ha implementado en las distintas pantallas un botón (flecha). Este botón está situado en la parte superior-derecha de cada actividad de la aplicación (excepto Registro e Inicio de Sesión) y realiza una llamada a la función “`FirebaseAuth.getInstance().signOut()`”. Tras cerrar sesión, la aplicación vuelve a la actividad principal (`Main.java`).

El otro servicio de Firebase implementado en la aplicación es el de BD en tiempo real. Las operaciones de este servicio se pueden resumir en lectura y escritura de datos en la BD. A diferencia de una BD de SQL, en Firebase no hay ni Tablas ni registros y los datos se guardan como objetos JSON [29].

En este proyecto, las llamadas de lectura y escritura se han utilizado en las actividades “Config”, “HomePatient”, “HomeSupervisor” y “Perfil”, correspondientes a la configuración, las pantallas de control del paciente, supervisor y la vista de los datos del perfil, respectivamente.

En el epígrafe 3.4.4 se introducía el diagrama de entidad-relación para la estructuración de los datos de la aplicación. Para ello, se ha implementado una clase “User” en la que se declaran todas las variables que se guardan en la BD de Firebase. En la Figura 3.22 se visualiza la declaración de todas las variables, las cuales deben llamarse de la misma forma que las referencias en Firebase. Esto es necesario para que se puedan realizar correctamente las operaciones de lectura, llamando a los métodos “Getter()” de estas variables.

```
private String emerPhone1;  
private String emerPhone2;  
private int hr_max;  
private int hr_min;  
private int last_hr;  
private String time_last_hr;  
private String fecha_last_control;
```

Fig 3.22 Declaración de variables de la clase User.java

La actividad “Config”, véase Figura 3.23, permite al usuario elegir el modo de uso de la aplicación. En caso de seleccionar el modo “Paciente” se habilita un cuadro de entrada con distintos EditText, donde se deben introducir los 2 teléfonos de emergencia y los pulsos mínimo y máximo del paciente. Una vez rellenados los campos de forma correcta, tal y como se detalla en las especificaciones, es necesario guardar los datos del paciente en la BD de Firebase.

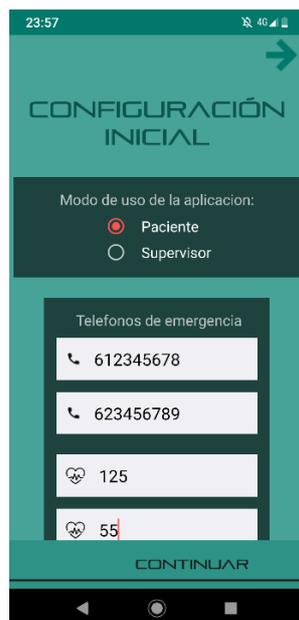


Fig 3.23 Captura de pantalla: Configuración

En la Figura 3.24 se muestran las dos funciones creadas para subir los datos a Firebase y estructurarlo de la forma que se había diseñado. Ambas funciones son similares, a través de la referencia al nodo principal de la BD se crean los distintos hijos: “Telefono_Emergencia” y “Usuario”. Dentro de cada hijo, se crea otro hijo identificado con el UID del usuario, de tal forma que los datos sean asociados a su usuario correspondiente. Para guardar cada dato se utiliza un objeto de la clase “HashMap” que permite asociar una clave a cada dato guardado. Es fundamental que la clave tenga el mismo nombre que la variable declarada en la clase User, de lo contrario no se podría leer de la BD.

Por tanto, en el nodo “Usuario” de la BD sólo se guardan el pulso máximo y mínimo, mientras que en “Teléfono_Emergencia” se guardan los distintos teléfonos a los que notificar si hay una emergencia. Las funciones explicadas en el párrafo anterior se invocan cuando se han rellenado los campos adecuadamente y se ha pulsado el botón “CONTINUAR”.

```

private void subirDatosTelefono(String data1, String data2) {

    Map<String, Object> datoUsuario0 = new HashMap<>();
    datoUsuario0.put( k: "emerPhone1", data1);
    mRootReference.child("Telefono_Emergencia").child(userID).updateChildren(datoUsuario0);

    Map<String, Object> datoUsuario1 = new HashMap<>();
    datoUsuario1.put( k: "emerPhone2", data2);
    mRootReference.child("Telefono_Emergencia").child(userID).updateChildren(datoUsuario1);
}

private void subirDatosUsuario(int data1, int data2) {

    Map<String, Object> datoUsuario0 = new HashMap<>();
    datoUsuario0.put( k: "hr_max", data1);
    mRootReference.child("Usuario").child(userID).updateChildren(datoUsuario0);

    Map<String, Object> datoUsuario1 = new HashMap<>();
    datoUsuario1.put( k: "hr_min", data2);
    mRootReference.child("Usuario").child(userID).updateChildren(datoUsuario1);
}

```

Fig 3.24 Funciones Firebase en la actividad de Configuración

Como hemos podido ver previamente, en la configuración solo se realizan funciones de escritura de datos. A continuación, se explica de qué forma se utilizan operaciones de lectura en las actividades “Perfil” y “HomeSupervisor”. Ambas pantallas se pueden ver en las Figuras 3.25 y 3.27, respectivamente.



Fig 3.25 Captura de pantalla: Perfil

La actividad “Perfil” tiene el objetivo de mostrar al usuario cual es la configuración que tiene guardada, es decir, mostrar los valores de pulso máximo, pulso mínimo y teléfonos de emergencia. Para leer los datos se han creado dos funciones similares: “ObtenerTelefonosFirebase()” y “ObtenerValoresPulso()”. En la Figura 3.26 se muestra sólo la primera función porque la otra sigue la misma estructura. Al igual que para escribir datos, es necesario tener una referencia al nodo principal (“mRootReference”) y a la vez avanzar en el árbol JSON hasta situarse en el campo que queremos leer. Dentro del método “onDataChange(...)” se inicializa un objeto de la clase User, explicada previamente, y a través de los métodos Getter() se consiguen los teléfonos de emergencia guardados en la BD. Una vez obtenidos dichos valores, los colocamos en elementos TextView para poder visualizarlos en la pantalla.

```
private void ObtenerTelefonosFirebase() {
    mRootReference.child("Telefono_Emergencia").child(userID).addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            User user = dataSnapshot.getValue(User.class);
            try {
                telefono1 = user.getEmerPhone1();
            } catch (NullPointerException e) {}
            try {
                telefono2 = user.getEmerPhone2();
            } catch (NullPointerException e) {}

            Log.e( tag: "Telefono emergencia 1", msg: "" + telefono1);
            Log.e( tag: "Telefono emergencia 2", msg: "" + telefono2);
            tv_tele1.setText("Nº 1 :"+ telefono1);
            tv_tele2.setText("Nº 2 :"+ telefono2);

        }

        @Override
        public void onCancelled(@NonNull DatabaseError databaseError) {

        }

    });
}
```

Fig 3.26 Función Firebase en la actividad Perfil

La actividad “HomeSupervisor” pretende ser la pantalla de supervisión para la persona que está controlando la actividad del paciente. Como se ve en la Figura 3.27, se muestra el último valor del pulso cardiaco junto con la fecha y hora de la medición. Además, el supervisor tiene, en un rectángulo blanco, la fecha y hora en la que el paciente pasó por última vez por el sensor de la vivienda. Por ello, al igual que en la actividad “Perfil”, solo se realizan operaciones de lectura de la BD. En este caso, se han creado dos funciones llamadas “obtenerValoresHeartRate()” y “obtenerValoresControlPoint()”. La primera función se encarga de leer tanto la última medición de pulso guardada (last_hr) como la fecha de dicha medición (time_last_hr). La segunda función solo lee la fecha del último control (fecha_last_control). Estas funciones muestran los valores obtenidos en la

pantalla, desde el pulso cardiaco en el arco de progreso hasta los distintos tiempos en los correspondientes elementos TextView.



Fig 3.27 Captura de pantalla: HomeSupervisor

La única actividad restante, que usa llamadas a la BD, es “HomePatient”. Esta actividad realiza lecturas y escrituras en la BD de la misma forma que las funciones vistas en este epígrafe, por lo que se presentará en el siguiente epígrafe dedicado a la comunicación aplicación-smartband.

3.5.1.4. Conexión Smartband

Una vez explicada la interacción de la aplicación con Firebase, falta hablar sobre la interacción entre la aplicación y la smartband. La smartband utilizada, como ya se introdujo, es la Xiaomi MI Band 2, un dispositivo de los denominados BLE. En este epígrafe se introduce la tecnología BLE junto a la implementación de todas las funcionalidades de la interacción aplicación-smartband.

BLE es una tecnología inalámbrica diseñada por Bluetooth SIG que se usa en aplicaciones de IoT, entre otras muchas utilidades. Comparado con el Bluetooth clásico, BLE reduce considerablemente el consumo de potencia manteniendo un rango de cobertura similar. Android y otros sistemas operativos como Windows 10 soportan BLE. Esta tecnología funciona en frecuencias 2.4 GHz, al igual que el Bluetooth clásico [30].

Para la integración de BLE en la aplicación se necesita un nivel API 18 en adelante. En la Figura 3.28 se muestra un esquema de los elementos básicos de este protocolo. A continuación, se presentan los distintos elementos de BLE:

- **Generic Attribute Profile (GATT):** El perfil GATT es una especificación general para enviar y recibir trozos pequeños de datos conocidos como atributos sobre un enlace BLE. Todos los perfiles de aplicación de baja energía están basados en GATT. Un dispositivo puede implementar más de un perfil, como el monitor de pulso y el detector del nivel de batería.
- **Attribute Protocol (ATT):** GATT está arriba del protocolo ATT. Este protocolo está optimizado para funcionar en dispositivos BLE. Cada atributo está identificado de forma particular por el Identificador Único Universal (UUID), de 128 bits. Los atributos transportados por ATT se formatean como características y servicios.
- **Característica:** una característica contiene un único valor y de 0 a 'n' descriptores que describen el valor de la característica.
- **Descriptor:** los descriptores se definen como atributos que describen una característica. Por ejemplo, un descriptor puede especificar un rango válido para el valor de una característica o la unidad de medida que tiene el valor de una característica.
- **Servicio:** Un servicio es una colección de características. Por ejemplo, el servicio de monitorización del pulso cardíaco incluye características como la medición del pulso cardíaco [3].

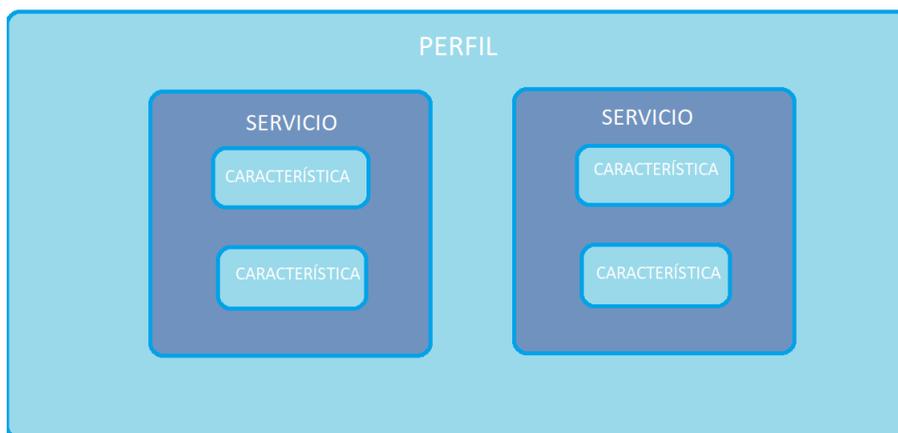


Fig 3.28 Esquema protocolo BLE

En el proyecto Android hay una clase llamada “CustomBluetoothProfile” donde están definidos los distintos UUID necesarios. En concreto, los UUID utilizados son los relacionados al servicio HeartRate. En la Figura 3.29 se ven los distintos elementos.

```
public static class HeartRate {  
    public static UUID service = UUID.fromString("0000180d-0000-1000-8000-00805f9b34fb");  
    public static UUID measurementCharacteristic = UUID.fromString("00002a37-0000-1000-8000-00805f9b34fb");  
    public static UUID descriptor = UUID.fromString("00002902-0000-1000-8000-00805f9b34fb");  
    public static UUID controlCharacteristic = UUID.fromString("00002a39-0000-1000-8000-00805f9b34fb");  
}
```

Fig 3.29 UUIDs del servicio HeartRate

BLE se usa en dos actividades de la aplicación: en SearchDevice, que realiza el escaneo de dispositivos; y en HomePatient, que es donde se conecta la smartband y se leen valores de ella.

En primer lugar, SearchDevice realiza una búsqueda de los distintos dispositivos BLE que están en la zona de cobertura y los muestra en pantalla, véase la Figura 3.30. Cada elemento de la lista contiene el nombre del dispositivo, en caso de conocerse, y la dirección MAC de dicho dispositivo.



Fig 3.30 Captura de pantalla: SearchDevice

Para poder realizar el escáner BLE es necesario usar un objeto de la clase `BluetoothAdapter`. El `BluetoothAdapter` es necesario para todo tipo de actividad con Bluetooth y representa el adaptador Bluetooth del propio dispositivo. Para inicializar dicho objeto se necesita de una instancia de la clase `BluetoothManager`, conseguida a través de la llamada a la función `“getSystemService(...)”`. A través del `BluetoothAdapter` podemos comprobar si este está encendido, empezar y parar el escaneo de dispositivos. Por otro lado, para representar el listado de dispositivos en la pantalla se crea una clase `LeDeviceListAdapter` que extiende de la clase `BaseAdapter`. Esta clase permite sobrescribir la función `“getView(...)”` para mostrar correctamente los datos de cada dispositivo en la pantalla. `SearchDevice` también realiza la comprobación de los permisos de la aplicación, ya que sólo se necesitan estos permisos en caso de estar usando el modo paciente. Si es la primera vez que se lanza la aplicación, aparecen mensajes para aceptar los permisos de ubicación y de SMS, necesarios para el escaneo y la monitorización. Una vez aceptados, se muestran los distintos dispositivos y sólo permite utilizar los dispositivos con nombre `“MI BAND 2”`. Si el dispositivo es válido, la aplicación guarda la dirección MAC en un objeto `Intent`, para que pueda ser leída en `HomePatient`. En caso de no ser un dispositivo válido se muestra un error a través de un mensaje `Toast`.

La actividad `HomePatient` es la que contiene el mayor número de funcionalidades del sistema. Es la encargada de conectarse a la smartband, de iniciar y parar la monitorización y de gestionar las alertas SMS. En la Figura 3.32 se muestran distintas pantallas de la actividad `HomePatient`. Para poder iniciar la medición del pulso cardiaco, es necesario haber conectado la smartband previamente. La parte izquierda de la Figura muestra la situación antes de que la pulsera haya sido conectada, por lo que en caso de pulsarse el botón rojo este no haría nada, ya que se encontraría inhabilitado. Para conectar la pulsera hay que pulsar el botón `CONECTAR` y esperar hasta que en la parte inferior se muestra el mensaje `“Mi Band 2 conectada”`, como se ve en la parte central de la Figura 3.32. Una vez conectada, el usuario puede iniciar la medición continua del pulso cardiaco. Para conectar la pulsera hay que utilizar objetos de las clases `BluetoothDevice`, `BluetoothAdapter` y `BluetoothGatt`. En la Figura 3.31 se muestra la función `startConnecting()` que utiliza estos elementos para conectarse al dispositivo. En primer lugar, la variable local `address` guarda la dirección que se obtuvo en la actividad `SearchDevice`. A su vez, se utiliza para inicializar el objeto `BluetoothDevice` que luego es necesario para conectar el dispositivo con un objeto de la clase `BluetoothGatt`.

```

void startConnecting() {

    String address = getIntent().getStringExtra(EXTRAS_DEVICE_ADDRESS);
    bluetoothDevice = bluetoothAdapter.getRemoteDevice(address);
    Log.v( tag: "test", msg: "Connecting to " + address);
    Log.v( tag: "test", msg: "Device name " + bluetoothDevice.getName());
    bluetoothGatt = bluetoothDevice
        .connectGatt( context: this, autoConnect: true, bluetoothGattCallback);
}

```

Fig 3.31 HomePatient: función startConnecting()

Por otro lado, una vez se ha establecido la conexión y se pulsa el botón de iniciar medición, se habilita un temporizador. Cada vez que se vence el temporizador y no se ha pulsado el botón “DETENER MEDICIÓN” se reinicia la función “getHeartMonitor2()”, encargada de mandar los bytes de iniciar o parar la medición continua, dependiendo del estado de la variable booleana “monitoreo”. Si “monitoreo” es “true” se envían los bytes 21, 1 y 1 (empezar medición continua). En caso contrario se mandan los bytes 21, 1 y 0 (detener medición continua).



Fig 3.32 Captura de pantalla: Homepatient

Para poder leer en la aplicación los distintos valores del pulso cardiaco hay que sobrescribir la función “onCharacteristicChanged(...)”. Cada vez que la smartband manda un nuevo valor del pulso cardiaco, éste es guardado en un array llamado heartRateBurst. Cuando el array llega a tamaño igual a 5, la aplicación realiza el promedio

de los distintos valores y lo almacena en la variable global `heartRateMedio`. El valor promedio obtenido se introduce como atributo en una función llamada “`procesarHeartRate(...)`”, la cual se encarga de detectar anomalías en el pulso cardiaco. En la Figura 3.33 se muestra el código de esta función. Como se puede ver, si el valor actual del pulso (promedio) es mayor o menor que el pulso máximo y mínimo, respectivamente, se incrementa un contador llamado `numAlerts`. Cuando este contador llega a 3 se envía un SMS a los números de teléfono introducidos en la actividad de configuración. Al implementar la función promedio y el contador de alertas se pretende mejorar la fiabilidad de las alertas y disminuir el número de falsas alarmas.

```
private void procesarHeartRate(int heartRate) {
    if ((heartRate < hr_min || heartRate > hr_max) && numAlerts >= 3) {
        sendAlertSMS();
        numAlerts = 0;
    } else if (heartRate < hr_min || heartRate > hr_max) {
        numAlerts++;
        Log.d( tag: "NumAlerts", String.valueOf(numAlerts));
    } else {
        Log.d( tag: "NumAlerts", String.valueOf(numAlerts));
    }
}
}
```

Fig 3.33 HomePatient: función `procesarHeartRate()`

Como se puede ver en la Figura anterior, se ha implementado una función llamada “`sendAlertSMS()`”, la cual se encarga de enviar un mensaje SMS a los números de emergencia. Los números de emergencia y los valores de pulso máximo y mínimo se leen de la BD de Firebase nada más iniciar la actividad, por lo que a la hora de mandar los SMS estos ya están disponibles. En la Figura 3.34 se muestra una captura de pantalla del mensaje SMS recibido por los contactos de emergencia.

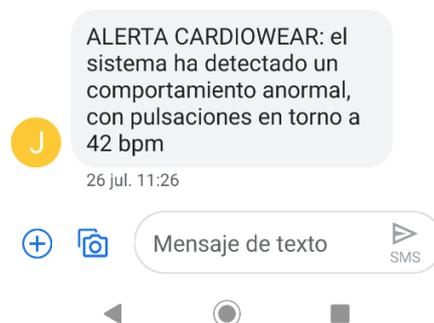


Fig 3.34 Captura de pantalla: Mensaje SMS recibido

3.5.2. Raspberry Pi en la vivienda.

El sistema desarrollado consta de varios subsistemas, como vimos previamente. El que ocupa el desarrollo de este epígrafe es el relativo a la implementación con Raspberry Pi de un sensor de distancia en la vivienda.

El propósito de este bloque es proporcionar una funcionalidad que permita tener un control del paciente en su vivienda. El sensor se debe instalar a la entrada de una de las habitaciones más transitadas de la vivienda, por ejemplo, el comedor. De esta forma, cada vez que el paciente sea detectado por el sensor, se notificará en la aplicación. Gracias a esta funcionalidad, el supervisor puede percibir si se está produciendo un problema, en caso de llevar mucho tiempo sin pasar por ese lugar. A continuación, se dan los detalles del desarrollo.

3.5.2.1. Configuración inicial Raspberry

Raspberry Pi es un dispositivo que no trae instalado ningún Sistema Operativo de fábrica. Por ello, para poder crear un entorno de trabajo hay que instalar un Sistema Operativo previamente. Además, para poder manejar el dispositivo a distancia utilizaremos el software “VNC Viewer”.

El sistema operativo recomendado por el fabricante es Raspbian, una distribución Linux que se desarrolló específicamente para la Raspberry Pi. Para la instalación de Raspbian se han seguido los siguientes pasos:

- **Descarga de NOOBS:** software que facilita la instalación de Raspbian en la tarjeta SD.
- **Formateo de la tarjeta SD:** se usa el programa “SD Formatter” para borrar todos los archivos existentes en la tarjeta.
- **Extracción NOOBS del archivo zip:** se extraen los archivos del zip descargado y posteriormente se copian en la tarjeta SD.
- **Conexión de la Raspberry:** se introducen la tarjeta SD y el cable de alimentación por sus puertos correspondientes. Para la configuración inicial, hasta que se haya realizado correctamente la virtualización del dispositivo, necesitamos de un teclado y ratón USB, y de un cable HDMI, conectados en sus puertos asociados.
- **Encender Raspberry:** este dispositivo no dispone de un interruptor de encendido, sino que cada vez que se conecta a la alimentación arranca. Por tanto, sólo hay que conectar el cable a una fuente de alimentación [23].

- **Configuración Raspberry:** se elige la configuración para el sistema operativo. Una vez realizada la configuración se visualiza el Escritorio.
- **Virtualización:** para poder trabajar con la Raspberry desde otro PC, es necesario que ambos dispositivos se encuentren en la misma red. Gracias a que el modelo de Raspberry usado dispone de interfaz Wifi, se tiene que configurar la red Wifi que esté, a su vez, en el PC.
- **VNC Viewer:** este programa se puede descargar de forma gratuita y permite virtualizar otros dispositivos en el PC. Para ello, primero hay que instalar “VNC Server” en la Raspberry, para que después se pueda introducir la dirección proporcionada por “VNC Server” en el PC. Cuando se ha configurado correctamente, se visualiza una pantalla como la Figura 3.35. Al hacer doble click sobre el dispositivo configurado se inicia otra pantalla donde se ve el escritorio de la Raspberry, como en la Figura 3.36.

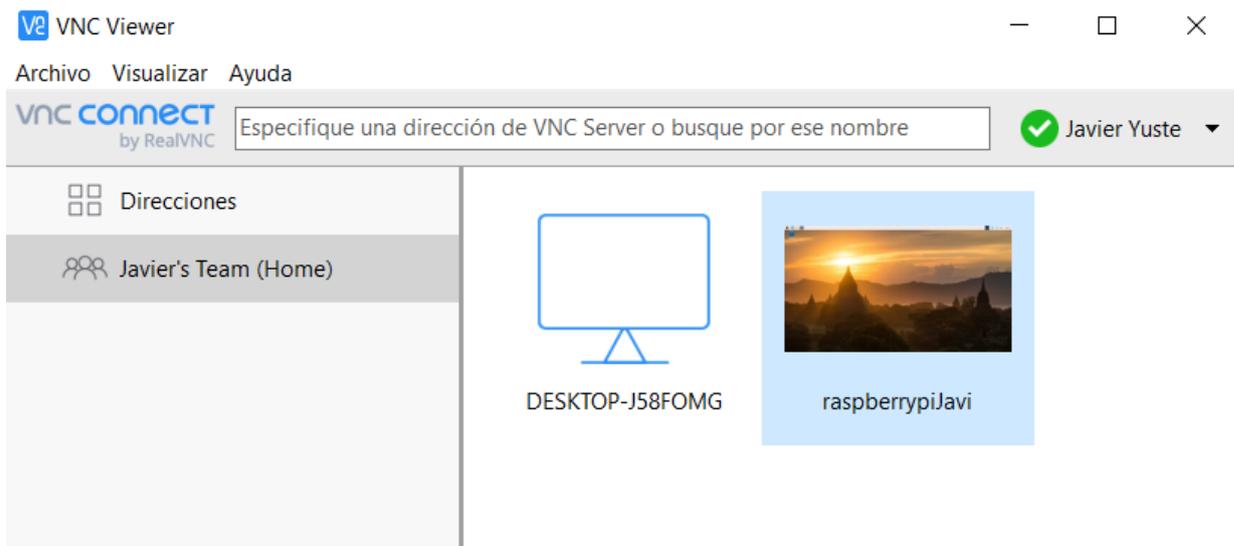


Fig 3.35 VNC Viewer configurado

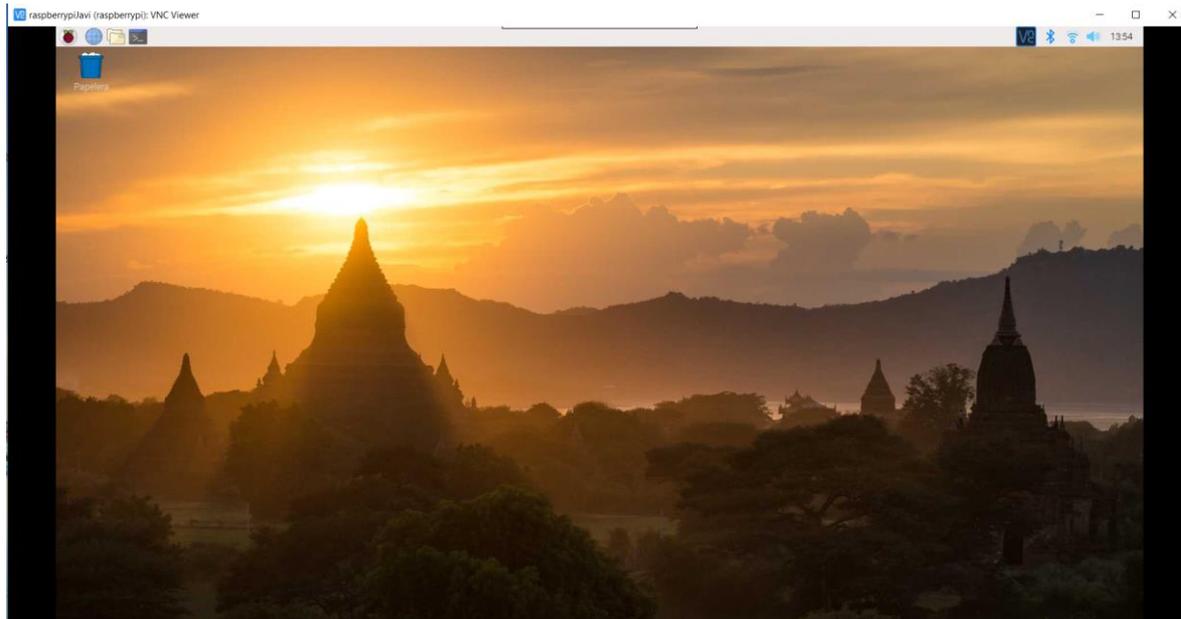


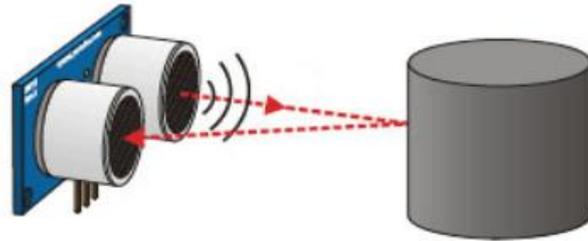
Fig 3.36 Escritorio Raspberry

Cuando se realizan todos los pasos citados se da por finalizada la configuración inicial. Como se ve en la Figura 3.36, el entorno de trabajo es similar a otras distribuciones Linux, pudiendo usar el Terminal para ejecutar comandos.

3.5.2.2. Configuración HW del sensor HC-SR04

Una vez detallada la configuración de la Raspberry, se presenta el circuito HW para el funcionamiento del sensor de ultrasonidos HC-SR04. En este epígrafe se detalla el funcionamiento teórico del sensor junto con el esquemático del circuito.

En primer lugar, un sensor de ultrasonidos es un dispositivo utilizado para medir distancias. Envía un pulso a alta frecuencia y mide el tiempo que tarda en volver, al rebotar con el obstáculo. La velocidad del sonido es de 343 m/s bajo condiciones normales. Por tanto, la distancia (m) es igual a la velocidad del sonido (m/s) multiplicada por el tiempo que tarda en volver el pulso, dividido por dos. La división por dos se realiza para sólo tener en cuenta una mitad del trayecto de ida y vuelta, ya que la distancia con el obstáculo sólo es la mitad de dicho trayecto. En la Figura 3.37 se visualiza este funcionamiento [24].



$$\text{Tiempo} = 2 * (\text{Distancia} / \text{Velocidad})$$

$$\text{Distancia} = \text{Tiempo} \cdot \text{Velocidad} / 2$$

Fig 3.37 Funcionamiento Sensor HC-SR04 [24]

Para realizar el circuito se necesita una protobard, un módulo HC-S04, dos resistencias de 330Ω y 470Ω , y cables “jumper” para las conexiones entre la Raspberry y la protoboard.

En la Figura 3.38 se puede ver la estructura del circuito. La conexión “VCC” del sensor se conecta al pin 2 (VCC), “GND” al pin 6 (GND) y “TRIG” al pin 12 (GPIO18). Por otro lado, la resistencia de 330Ω se conecta al “ECHO” del sensor y por el otro lado al pin 18 (GPIO 18) y a la resistencia de 470Ω , conectada al pin 6 (GND). Esta configuración es necesaria porque los pines GPIO trabajan a sólo 3.3V, de ahí el divisor resistivo formado por ambas resistencias. Es necesario poner una referencia a masa (GND) para tener una señal en GPIO24 cuando se envía un pulso, de lo contrario la entrada sería aleatoriamente ‘0’ o ‘1’ [25].

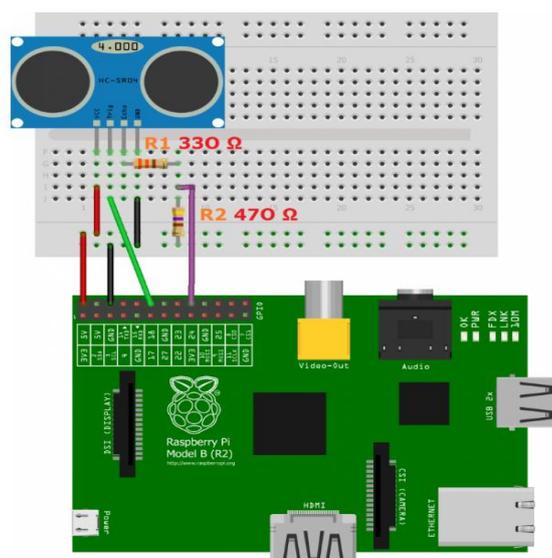


Fig 3.38 Esquema circuito Raspberry con HC-SR04 [25]

Por último, en la Figura 3.39, se muestra la configuración implementada del esquemático.

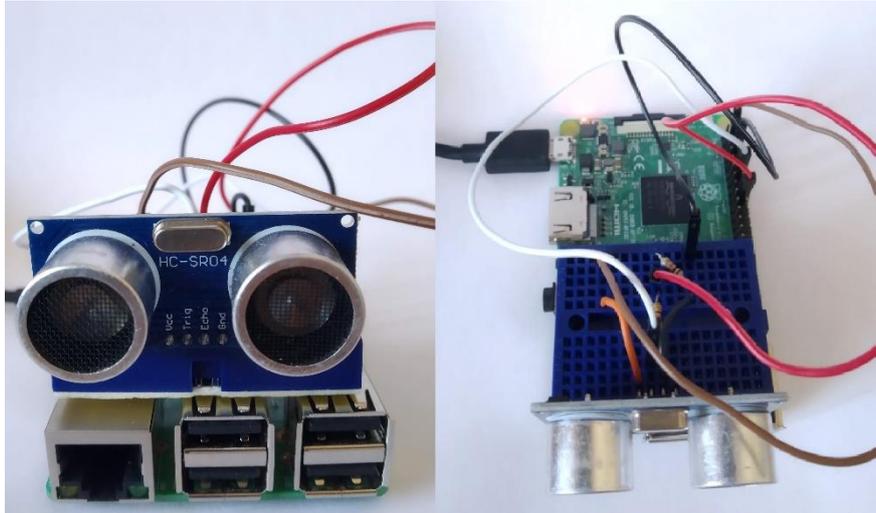


Fig 3.39 Circuito Raspberry implementado

3.5.2.3. Configuración SW del sensor HC-SR04

Siguiendo con la configuración del subsistema formado por el sensor HC-SR04, falta presentar el SW que realiza tanto la medición de la distancia, como el envío de información a la base de datos de nuestro sistema en Firebase.

En la Figura 3.40 se presenta la parte del código que realiza el cálculo de la distancia. Para ello, es necesario importar las librerías “RPi.GPIO” (manejo de la Raspberry) y la librería “time” para funciones relativas a la medición de tiempos. La función realiza los pasos descritos en el apartado anterior, relativos al funcionamiento teórico de este sensor.

```

16 def distance():
17     # activa la señal TRIGGER
18     GPIO.output(GPIO_TRIGGER, True)
19
20     # tras 0.01ms desactiva el TRIGGER
21     time.sleep(0.00001)
22     GPIO.output(GPIO_TRIGGER, False)
23
24     StartTime = time.time()
25     StopTime = time.time()
26
27     # guarda el tiempo de comienzo
28     while GPIO.input(GPIO_ECHO) == 0:
29         StartTime = time.time()
30
31     # guarda el tiempo de llegada
32     while GPIO.input(GPIO_ECHO) == 1:
33         StopTime = time.time()
34
35     # diferencia de tiempo
36     TimeElapsed = StopTime - StartTime
37     # multiplicación por la velocidad de la luz(34300 cm/s)
38     # y división por dos, por la ida y la vuelta
39     distance = (TimeElapsed * 34300) / 2
40
41     return distance

```

Fig 3.40 Código Python para calcular la distancia en Raspberry

Para realizar la conexión con Firebase es necesario descargarse un certificado con las credenciales de la aplicación y su clave privada. Para ello, dentro de “Configuración” en la consola de Firebase, en el apartado “Cuentas de servicio” hay que pulsar en “Generar nueva clave privada”. De esta forma, se genera un archivo de tipo “json”, en este caso “credv2.json”, que se necesita referenciar en el código para poder enviar datos a la BD de Firebase. Para poder utilizar las funciones de Firebase hay que importar la librería “firebase_admin” y de ella “credentials” y “db”, para poder interpretar el certificado y realizar operaciones en la BD, respectivamente. En la Figura 3.41 se observa la parte del código que inicializa la aplicación, proporcionando privilegios de administrador. Como se puede ver, aparte de referenciar la dirección del archivo “credv2.json”, es necesario proporcionar la URL de la BD. En este caso el nombre de la aplicación en Firebase se denomina “cardiowearv2”, de ahí el nombre de la URL.

```
13 cred = credentials.Certificate('/home/pi/IoT/credv2.json')
14
15 #Inicializar la app con una cuenta de servicio, proporcionando privilegios de administrador
16
17 firebase_admin.initialize_app(cred, {
18     'databaseURL': 'https://cardiowearv2.firebaseio.com/'
19 }
```

Fig 3.41 Código Python para inicializar la aplicación en Firebase

A continuación, se presenta el código que proporciona la estructura en la BD tal y como se determinó en el apartado 3.4.4. Primero, se crea una referencia en la raíz de la BD, llamada “Control_Point”, ya que este bloque funciona como un punto de control. De esa referencia se crea un hijo con la id (única) del usuario, proporcionada por Firebase al registrar un usuario nuevo. A su vez, se crea otro hijo llamado “fecha_last_control” que guardará una cadena de caracteres con la fecha del último paso por el punto de control. En la Figura 3.42 se presenta la función principal, donde mide continuamente la distancia y en caso de detectarse un obstáculo a menos de un metro se actualiza la fecha en la base de datos.

```

66 if __name__ == '__main__':
67     try:
68         while True:
69             dist = distance()
70             print ("Measured Distance = %.1f cm" % dist)
71             if dist <100:
72                 ref2.set(time.strftime("%a, %d %b %Y %H:%M:%S"))
73                 time.sleep(1)
74
75             # Reset by pressing CTRL + C
76         except KeyboardInterrupt:
77             print("Measurement stopped by User")
78             GPIO.cleanup()

```

Fig 3.42 Código Python de la función principal

El archivo del código se denomina “ultrasonic_distance.py”, siendo “.py” la extensión de los códigos Python. Para ello, y tras instalar Python en el terminal, el archivo se ejecuta como se ve en la Figura 3.43. Tras ejecutar con el comando “sudo Python ultrasonic_distance.py” el sensor empieza a tomar mediciones de distancia y cuando recibe una medición con un valor inferior a 100, como vimos anteriormente, actualiza la BD de Firebase. En la Figura 3.44 se ve el momento en el que la base de datos es actualizada, mostrando un color anaranjado cuando un valor se modifica.

```

pi@raspberrypi:~/IoT $ sudo python ultrasonic_distance.py
Measured Distance = 198.3 cm
Measured Distance = 197.9 cm
Measured Distance = 197.9 cm
Measured Distance = 198.8 cm
Measured Distance = 197.4 cm
Measured Distance = 198.3 cm
Measured Distance = 198.8 cm
Measured Distance = 198.7 cm
Measured Distance = 198.7 cm
Measured Distance = 198.3 cm
Measured Distance = 197.9 cm
Measured Distance = 39.9 cm
Measured Distance = 197.8 cm
Measured Distance = 198.3 cm
Measured Distance = 198.3 cm
Measured Distance = 198.4 cm
Measured Distance = 198.3 cm

```

Fig 3.43 Ejecución programa Python en el terminal de Raspbian

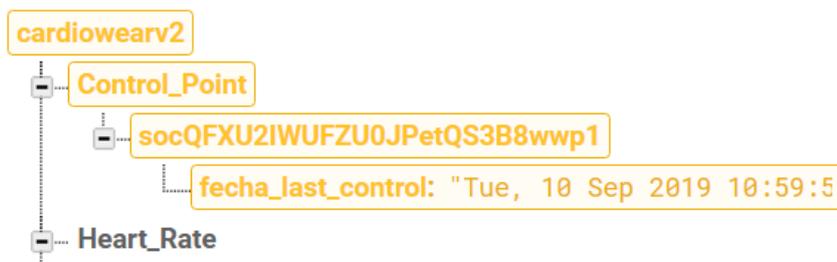


Fig 3.44 Actualización Punto de control en la consola de Firebase

El bloque formado por la Raspberry y el sensor de distancia conforman una parte esencial de este proyecto, proporcionando un valor añadido a la aplicación que se está desarrollando.

3.6. Pruebas

Para determinar si el sistema implementado cumple con los requisitos establecidos y funciona correctamente es necesario llevar a cabo una serie de comprobaciones. Para validar el aspecto técnico de la aplicación se han realizado las pruebas manuales, presentadas a continuación. Por otro lado, las pruebas de aceptación indican si el sistema ha sido aceptado por la entidad o persona supervisora.

3.6.1. Pruebas manuales

Tabla 3.33 PRUEBA MANUAL 01

Identificador	PM-01
Nombre	Funcionamiento correcto al abrir y navegar por la aplicación.
Descripción	Se comprueba si la aplicación se abre correctamente y al navegar a través de ella, sin necesidad de haber conectado el smartband, no da ningún error o cierre inesperado.
Resultado	Éxito

Tabla 3.34 PRUEBA MANUAL 02

Identificador	PM-02
Nombre	Funcionamiento correcto del registro de usuarios.
Descripción	Se comprueba si la aplicación muestra errores cuando no se han rellenado los campos requeridos para el registro. Además, se verificará que los usuarios registrados coinciden con los vistos en la consola de Firebase.
Resultado	Éxito

Tabla 3.35 PRUEBA MANUAL 03

Identificador	PM-03
Nombre	Funcionamiento correcto de la pantalla de configuración.
Descripción	Se comprueba que esta pantalla sólo se muestra en el primer uso de la aplicación.

	<p>En posteriores usos, esta pantalla se omite al iniciar la aplicación.</p> <p>No obstante, los datos se pueden modificar en cualquier momento, accediendo a configuración.</p> <p>También se comprueba que los datos insertados por el usuario en la configuración se guardan correctamente en Firebase.</p>
Resultado	Éxito

Tabla 3.36 PRUEBA MANUAL 04

Identificador	PM-04
Nombre	Solicitud correcta de los permisos.
Descripción	Se comprueba que la primera vez que se inicia la aplicación se muestran los permisos necesarios para utilizarla.
Resultado	Éxito

Tabla 3.37 PRUEBA MANUAL 05

Identificador	PM-05
Nombre	Funcionamiento correcto del escáner de dispositivos BLE.
Descripción	<p>Se comprueba que la pantalla en la que se muestran los dispositivos BLE funciona correctamente.</p> <p>Se comprueba que los dispositivos muestran la MAC correspondiente.</p> <p>Se comprueba que sólo deja conectarse a los dispositivos con nombre “Mi Band 2”.</p>
Resultado	Éxito

Tabla 3.38 PRUEBA MANUAL 06

Identificador	PM-06
Nombre	Conexión exitosa e intercambio de datos con la smartband.
Descripción	<p>Se comprueba que la aplicación se conecta exitosamente con la smartband.</p> <p>Se comprueba que se leen datos correctamente del dispositivo.</p>
Resultado	Éxito

Tabla 3.39 PRUEBA MANUAL 07

Identificador	PM-07
Nombre	Datos de la interfaz gráfica actualizados.
Descripción	Se comprueba que los datos presentados en los distintos paneles de control, tanto del usuario como del supervisor, se corresponden a los valores actualizados de Firebase.
Resultado	Éxito

Tabla 3.40 PRUEBA MANUAL 08

Identificador	PM-08
Nombre	Funcionamiento correcto del Punto de Control.
Descripción	Se comprueba que el sensor en la vivienda detecta correctamente cuando el usuario pasa delante de él. Se comprueba que la fecha subida a Firebase es la correspondiente a ese instante temporal.
Resultado	Éxito

Tabla 3.41 PRUEBA MANUAL 09

Identificador	PM-09
Nombre	Funcionamiento correcto del sistema de alarmas.
Descripción	Se comprueba que la aplicación envía un SMS a los números de emergencia establecidos por el usuario.
Resultado	Éxito

Tabla 3.42 PRUEBA MANUAL 10

Identificador	PM-10
Nombre	Funcionamiento de la aplicación en segundo plano.
Descripción	Se comprueba que la aplicación sigue monitorizando al paciente incluso cuando la ésta se queda en segundo plano.
Resultado	Éxito

3.6.2. Pruebas de aceptación

Tabla 3.43 PRUEBA ACEPTACIÓN

Identificador	PA-1
Nombre	Validación del proyecto por el tutor.
Descripción	Se le entrega el proyecto al tutor y éste comprueba que cumple con las expectativas establecidas al inicio del proyecto.
Resultado	Éxito

Las pruebas se han realizado satisfactoriamente, determinando que el proyecto cumple con los requerimientos técnicos y con los objetivos establecidos por el tutor.

4. PLANIFICACIÓN

En este apartado se presenta información acerca de un aspecto fundamental de cualquier proyecto: la planificación. La planificación de un proyecto consiste en realizar de una forma sistemática y ordenada una descripción de las distintas tareas a realizar, junto con la relación temporal de cada una de estas tareas.

Cada tarea ha de ser llevada a cabo por un agente del proyecto en cuestión, en el caso de este proyecto en particular, todas las tareas serán realizadas por el autor del proyecto bajo la supervisión del tutor.

4.1. Fases principales

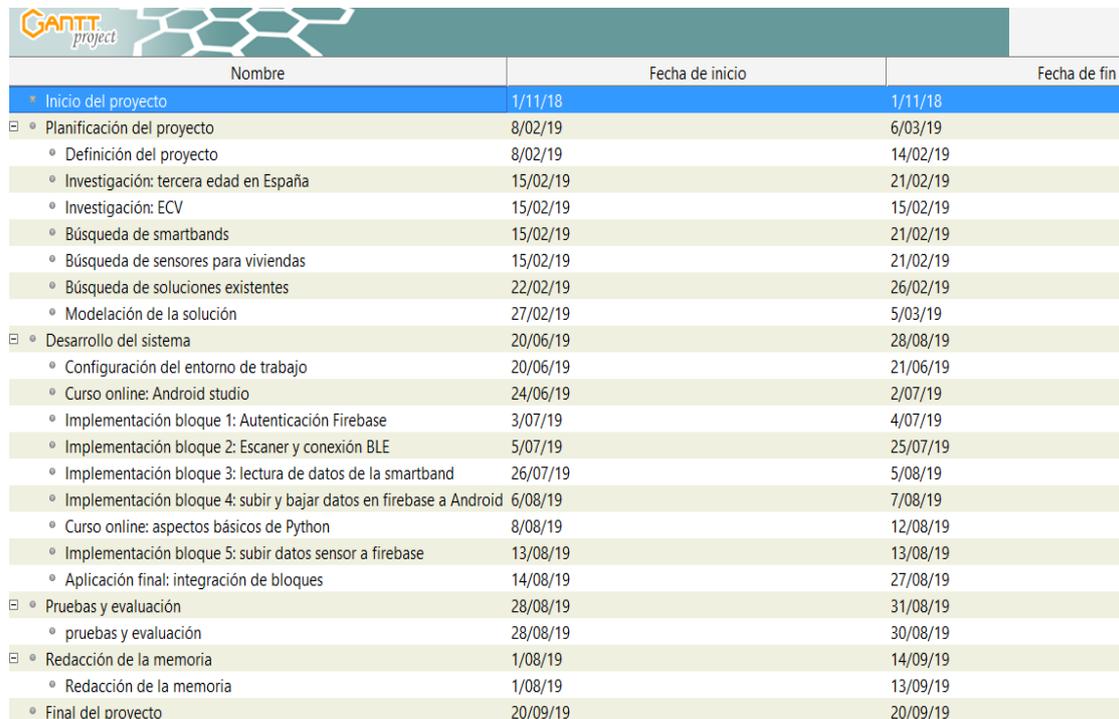
En primer lugar, se presentan las distintas fases del proyecto. Cada fase acoge a un número de tareas por las cuales se llega al término de dicha fase. Las fases principales de este proyecto son las siguientes:

- **Planificación del proyecto:** esta fase es la primera de todas y recoge las tareas desde la propia definición del proyecto hasta la modelación de una solución, antes de ser implementada. Las fases posteriores están condicionadas al resultado de esta primera fase.
- **Desarrollo del sistema:** este proyecto tiene como uno de sus objetivos principales la implementación real de un sistema. Por tanto, la fase de desarrollo ocupa una gran parte del espacio temporal de todo el proyecto. Esta fase incorpora las tareas desde la configuración de un entorno de trabajo hasta la realización del prototipo final.
- **Pruebas y evaluación:** al igual que todo proyecto de software, la parte de evaluación y pruebas conforma un apartado clave en la línea temporal del proyecto. Este apartado también incorpora los cambios realizados en caso de haber encontrado cualquier tipo de irregularidad.
- **Redacción de la memoria:** fase donde se escribe este documento.

4.2. Diagrama de Gantt

Una vez descritas las distintas fases del proyecto, queda indicar las distintas tareas pertenecientes a cada fase. Para poder visualizar la evolución temporal de cada una de las tareas se hace uso de una herramienta utilizada para la gestión de proyectos, el diagrama de Gantt.

Para ello, se utiliza un programa gratuito para PC, GanttProject, que nos permite realizar una declaración de cada tarea con sus atributos correspondientes. Además, este software genera automáticamente un diagrama de Gantt. En la siguiente Figura 4.1 se indican todas las tareas que conforman el proyecto, a su vez distribuidas en cada una de las fases descritas en el epígrafe anterior.



Nombre	Fecha de inicio	Fecha de fin
Inicio del proyecto	1/11/18	1/11/18
Planificación del proyecto	8/02/19	6/03/19
Definición del proyecto	8/02/19	14/02/19
Investigación: tercera edad en España	15/02/19	21/02/19
Investigación: ECV	15/02/19	15/02/19
Búsqueda de smartbands	15/02/19	21/02/19
Búsqueda de sensores para viviendas	15/02/19	21/02/19
Búsqueda de soluciones existentes	22/02/19	26/02/19
Modelación de la solución	27/02/19	5/03/19
Desarrollo del sistema	20/06/19	28/08/19
Configuración del entorno de trabajo	20/06/19	21/06/19
Curso online: Android estudio	24/06/19	2/07/19
Implementación bloque 1: Autenticación Firebase	3/07/19	4/07/19
Implementación bloque 2: Escaner y conexión BLE	5/07/19	25/07/19
Implementación bloque 3: lectura de datos de la smartband	26/07/19	5/08/19
Implementación bloque 4: subir y bajar datos en firebase a Android	6/08/19	7/08/19
Curso online: aspectos básicos de Python	8/08/19	12/08/19
Implementación bloque 5: subir datos sensor a firebase	13/08/19	13/08/19
Aplicación final: integración de bloques	14/08/19	27/08/19
Pruebas y evaluación	28/08/19	31/08/19
pruebas y evaluación	28/08/19	30/08/19
Redacción de la memoria	1/08/19	14/09/19
Redacción de la memoria	1/08/19	13/09/19
Final del proyecto	20/09/19	20/09/19

Fig 4.1 Lista de tareas del proyecto

Una vez realizada la lista de las distintas tareas del proyecto, queda representar de forma gráfica la distribución de dichas tareas en el tiempo. De tal forma, en la Figura 4.2, se representa el diagrama de Gantt de este proyecto. El tiempo está expresado en días y en cada fila está situada la distribución temporal de cada tarea del listado previo. Por otro lado, las distintas flechas indican que una actividad tiene que haber terminado para que empiece la siguiente.

En total, el número de días dedicado al proyecto se estimó en 100 días, con una media de 3 horas al día. No obstante, esta estimación que se realizó no se ha cumplido en algunas de las tareas intermedias, pero sí en su conjunto, finalizando el proyecto en el tiempo establecido.

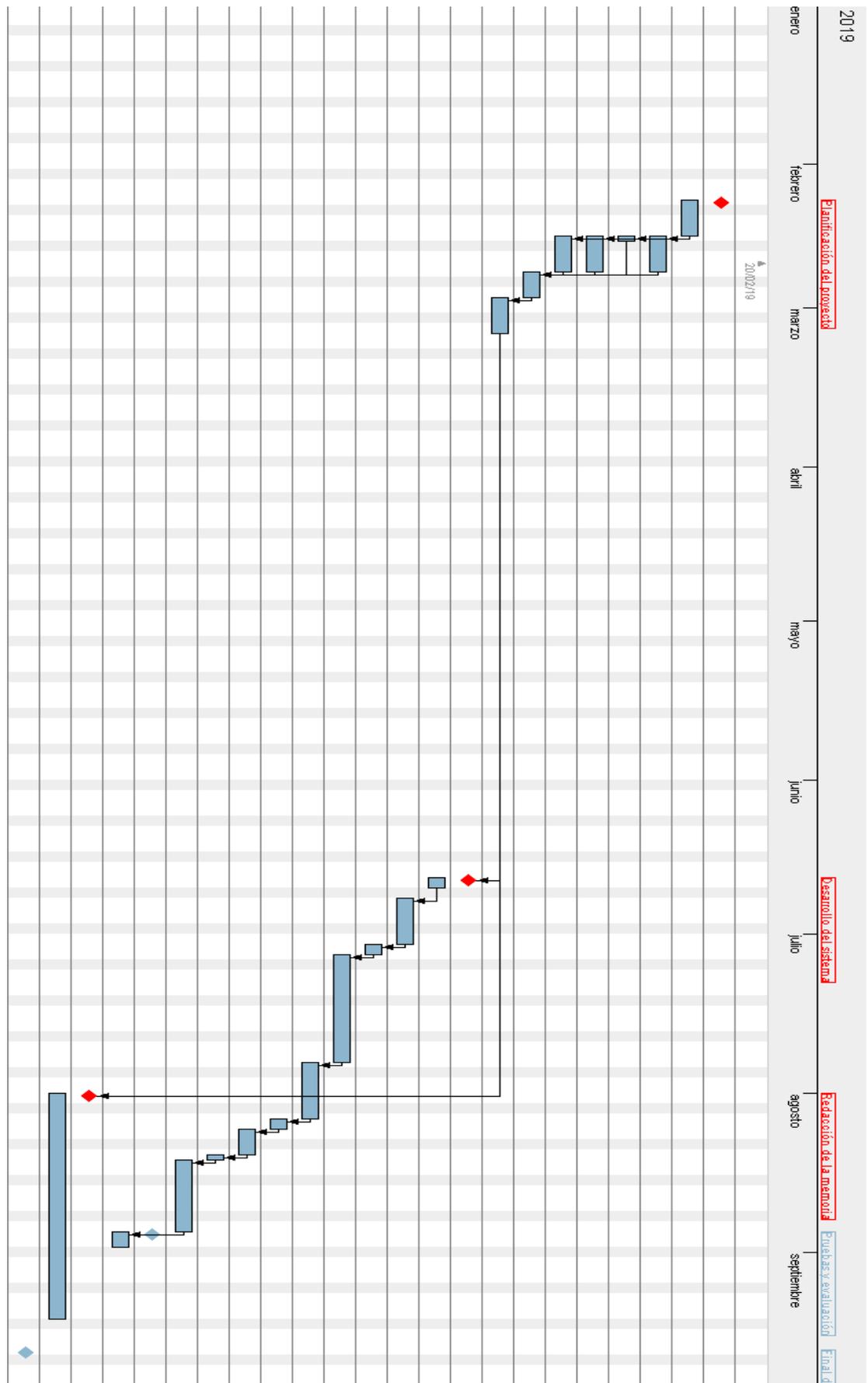


Fig 4.2 Diagrama de Gantt con la planificación temporal del proyecto

5. CONCLUSIONES Y TRABAJO FUTURO

Como se ha visto a lo largo de este documento, este proyecto ha consistido en dar e implementar una solución a un problema que se observa en la sociedad. La sociedad española está envejecida, con un gran volumen de personas mayores que viven en viviendas unipersonales. Gracias al sistema desarrollado, personas de avanzada edad pueden tener una supervisión en tiempo real por parte de familiares o gente de su entorno de confianza, aparte de recibir alarmas en caso de detectarse algún patrón anormal.

5.1. Revisión de los objetivos

Como se introdujo en el apartado de Objetivos (epígrafe 1.2), los objetivos se dividían en los objetivos personales y los del proyecto. Los objetivos personales incluían el aprendizaje de la herramienta Android Studio o, incluso, la gestión de un proyecto de gran envergadura. Estos objetivos se han conseguido paralelamente a los objetivos del proyecto, los más relevantes.

El proyecto desarrollado ha seguido una metodología de proyectos de SW, en los que una fase consistía en el análisis de requerimientos. En esta fase se establecían los objetivos del sistema y los requisitos funcionales y no funcionales. Como se veía en el apartado de pruebas (epígrafe 3.6), los resultados a las distintas pruebas han sido correctos, incluyendo la aceptación por parte del tutor del proyecto. Uno de los objetivos más importantes era incorporar una interfaz gráfica fácil de usar, teniendo en cuenta que iba a ser usada por personas mayores, entre otros. Como se detalla en el capítulo 3, las distintas pantallas de la aplicación están realizadas con un simple diseño, cumpliendo con este objetivo. El otro objetivo principal, establecido en la introducción, consistía en cumplir los requisitos y objetivos del sistema que fueran detallados. Todos estos requisitos funcionales y no funcionales se han cumplido correctamente.

5.2. Trabajos futuros

A pesar de haberse cumplido los distintos objetivos establecidos, el proyecto puede incorporar nuevas funcionalidades y mejoras. Gracias al diseño modular del proyecto, la incorporación de nuevas funcionalidades no debe suponer ningún reajuste de lo previamente realizado. A continuación, se presentan distintas mejoras que pueden implementarse:

- Compatibilidad con distintos modelos de smartband: la aplicación, a día de hoy, solo permite el uso del modelo Xiaomi MI Band2. Si se implementa una mejora que haga compatible la aplicación con otros tipos de smartband, como los otros modelos superiores de Xiaomi MI Band, el nivel de penetración de la aplicación será mucho mayor.
- Incorporar nuevos sensores en la vivienda: en el bloque formado por la Raspberry, solo se ha implementado el funcionamiento de un sensor de ultrasonido. Se pueden implementar numerosas soluciones paralelas para un mayor control en la vivienda. Por ejemplo, los sensores MQ detectan distintos tipos de gases como el monóxido de carbono, altamente peligrosos. Al sistema de alarma configurado con el pulso cardiaco se le podría añadir otro de gases en la vivienda. No obstante, este proyecto está más volcado con los problemas cardiovasculares.
- Incorporación de IA a la aplicación: a través del sensor de distancia se pretende tener un control de la actividad del paciente en la vivienda. Las personas mayores, especialmente si han sufrido un ictus, suelen seguir patrones de actividad. Gracias a la IA, se puede implementar un algoritmo que realice predicciones de posibles problemas, en función de la actividad recogida por el sensor de distancia.
- Internalización: al término de este proyecto, la aplicación sólo está desarrollada para ser utilizada en lengua castellana. No obstante, Android Studio facilita una incorporación sencilla de nuevos idiomas. En particular, un primer paso sería añadir el inglés.

Las posibles mejoras son solo algunas de las muchas implementaciones que se podrían llevar a cabo tras este proyecto. Como opinión subjetiva, este Trabajo Fin de Grado puede ser mejorado en muchos aspectos funcionales. No obstante, pretende ser una base para futuros desarrollos y, sobre todo, ayudar a un gran número de personas mayores y sus familias.

APÉNDICE A: ENGLISH SUMMARY

A.1. Introduction

In this section are presented some important aspects of the project. It is composed by the motivation, the objectives the project tries to cover, the socio-economic environment and the regulatory framework.

A.1.1. Motivation

This project tries to solve some problems that exist in the world and, more precisely, in Spain.

Following some statistic's results of the last years it is said that cardiovascular disease is the main cause of death in Spain. Also, it is known that a fast detection is one of the most important aspects to consider.

The Spanish demography shows that in the coming years the biggest group of people will be the elderly. Therefore, it is predicted that healthcare inversions will be increased in the near future. Also, one third of the people who live alone are formed by over 65.

From a technological point of view, each year increases the number of IoT devices and the money invested in healthcare IoT-based solutions.

As a result, this development's project tries to help elderly people and their families by improving the detection of cardiovascular disease.

A.1.2. Objectives

In this project are considered separately the personal objectives and the project objectives, whether they are related to new knowledge or project goals, respectively.

On the one hand, some of the personal projects are planning a big project and learning Android Studio as the development framework for Android apps. On the other hand, the project objectives are establishing the minimum requirements needed, fulfilling these requirements and creating an intuitive Android app.

A.1.3. Socio-economic environment

From an economical point of view, the world's mobile apps market grows every year with very high revenues. In Spain, in particular, 22 million of users download apps and 4 million of downloads are made every day. A third of the population spend more than 40 hours per week.

In Spain, as presented in the motivation, there are a lot of elderly people that lives alone and because of this, their families have the necessity of taking more control of the situation and activity of them. Also, in Spain there exists a digital breach between young and elderly people, but it is being smaller in the last years. The development of this work tries to cover the necessities presented before.

As seen before, the mobile apps market is growing so that it is viable to make a project in this sector. In order to have higher probability of success the download of the app will be free and the app will not show advertisements in it. Revenues will be obtained by selling the HW with sensors.

The project has a budget of 6.810,75€ due to some outlays. The main outlay is intended to cover the personnel salary. However, the budget includes material payment, light and internet supply, and computer equipment.

A.1.4. Regulatory Framework

In order to develop a mobile app, there have to be considered the different laws and regulations that developers should accomplish.

Firstly, the LOPD related to the personal data protection and digital rights guarantee should be fulfil. Among others, developers should protect users' privacy with a previous consent and indicate which will be the use of their personal data.

The implemented app uses some Firebase services like real time database and authentication. Firebase indicates that their clients are responsible of accomplishing the regulatory framework of their legal environment.

The app quality is directly related with the future success in revenues. Therefore, there are some standards that should be taken into account. For example, the visual environment and user interaction should follow some android recommendations in order to be intuitive and easy to use. There are other standards related to security, compatibility, stability or proof mechanisms.

A.1.5. Document Structure

This document is organized in four parts: State of the art, project development, planification and conclusions with future work.

A.2. State of the art

In this section are presented some important concepts and technologies related to this work. Also, the existing solutions analysis enable the ability of making competitive advantages.

A.2.1. Current situation

Cardiovascular diseases are a set of various irregularities of the heart and blood vessels, including coronary heart disease and cerebrovascular diseases. Cardiovascular risk factors increase the probability of cardiovascular disease. The most important are cholesterol, diabetes mellitus, periodontal disease, stress, anxiety, lack of exercise, heart rate, hypertension, poor diet, obesity, smoking and alcohol consumption. Also, the age has an impact over cardiovascular risk because similar to the whole human body, the heart changes while the ageing. However, the prevention in cardiovascular risks is always effective. This project is based in the heart rate monitoring and an alarm system used when heart rate is too high or too low continuously.

IoT could be consider as a dynamic and global infrastructure that controls intelligent devices in a complex manner. This allows IoT devices' interconnection that share data to create new applications and services for improving quality of life. In 2023 will be 50.000 million of connected devices in the world, being the world population of 7.500 million of people. The common features in IoT are large scale, smart devices, sensor-based systems, complex systems, dynamic environment, big amounts of data, heterogeneity, limited power and connectivity.

Another important term in this project is e-Health, known as the set of information technologies (ICT) used for prevention, diagnostic, monitoring, treatment, and health management in health sector. E-health has some inherent terms related to its definition: efficiency, enhancing quality of care, evidence-based, empowerment of consumers and patients, extending health care beyond its common boundaries, ethics and equity, between others.

Wearables are daily-life devices that people wear that have one or more processors. This group is included in IoT group and is formed by bands, glasses, smartwatches... with the purpose of improving people quality of life. These devices are being used in almost every sector in society, as healthcare, fashion, textile industry...Due to the technical

improvements wearable market is growing every year and making possible new applications.

This project proposes a smartband based solution. A smartband is a device located in the wrist and retrieves information from the person that wears the device. These devices provide GPS and accelerometer information, heart rate, UV-sensors' data, wireless connectivity and user's assistance.

Smartphone is a type of mobile phone that enables computation functionalities as a computer. These devices need an operating system to work in order to manage processes and let other programs to run. Android is the most used operating system (77% of world devices) and is followed by iOS (22%). Due to this, the project app works over Android environment.

A.2.2. Existing solutions

This project is aiming to cover these necessities in a joint form:

- Continuous heart rate monitoring.
- Alarm in case of detecting some problem in the patient.
- Home tracking with sensors.

There exist some works that tries to cover these cited ideas:

- “Rithmi”: a wearable that makes a continuous heart rate monitoring in order to detect auricular fibrillation, one of the most common arrhythmia that can produce an Ictus.
- “Home teleassistance”: A Red Cross product formed by an emergency button that people wear on the wrist or hanged in the neck.
- “Apple Watch Series 4”: this Apple product join the possibility of making and electrocardiogram. It can detect auricular fibrillation where patient should visit a doctor.
- “Vigilance cameras for elderly”: solution based in setting cameras in the home. Therefore, families can follow the activity in real time throw a mobile or web app.

After comparing the different found solutions, it is considered that there is a necessity of covering all the demands together because any of them covers the three demands at the same time.

A.3. Project development

This section includes the methodology, the design alternatives and the different parts of the project. Also, before doing the implementation is necessary to design every subsystem.

A.3.1. Methodology

The project follows a software methodology in which there are different periods that aims to create a prototype. Once the first prototype is created, the project iterates in order to improve the previous prototype. The whole process is divided in four stages: requirements analysis, design and structure, development and implementation and testing.

A.3.2. Requirements analysis

First of all, it is presented a list of essential goals the application should achieve in order to consider an acceptable prototype. These objectives are user management, Alert management, hear rate management, Control Point management and continuous monitoring. Also, the system has some functional and non-functional requirements. On the one hand, functional requirements are user register, login, logout, user data modification, read of emergency contacts, alerts, heart rate read, heart rate update, control point read, control information update and continuous. On the other hand, non-functional requirements are simple interface, development HW, execution HW, regulatory framework, compatibility and database reliability.

A.3.3. Design alternatives

In the process of designing the final prototype has been considered different alternatives. First of all, is important to decide in which operating system the app will run. The two considered alternatives have been Android and IOS and due to the fact that Android runs in the 77% of the devices it has been decided to use Android in the project app. Another important aspect of the design is to decide what to do with user data. In the system it is needed cloud access so there have been considered two alternatives: a client-server database where data is stored and accessed throw SQL language or Firebase database service. Firebase supplies some other features apart from real-time database, such as Authentication services. Therefore, Firebase is the selected option.

Related to the HW, it is needed to consider the alternatives for the HW in the home and the smartband. It has been considered two type of devices for the home: Arduino and

Raspberry Pi. The selected one has been Raspberry Pi because it enables multitask work and incorporates the internet module while Arduino does not support these features. The smartband selected has been Xiaomi MI Band 2 but it was also considered the use of the Polar H7 device.

A.3.4. Design and architecture

After making a requirements analysis and considering possible solutions for the system, now, it is presented the project design and architecture.

The global system consists in the interconnection of some blocks such as smartband-app, app-firebase and Raspberry-firebase. Design and architecture constitute the last part before the system implementation so it is essential not to make mistakes here.

Use cases describe how the user uses the system in order to achieve an objective. Therefore, almost every use case is related to one general objective, defined before. There have been considered eight use cases: Start, Login, Register, Configuration, Supervisor Control Panel, BLE devices' list, User Control Panel and logout. Every use case and its corresponding activity diagram is described in detail in the epigraphs 3.4.1 and 3.4.2 of the document. In the following epigraph are presented the application prototypes, one for each activity the app contains.

Last design part consists in creating a structure for database storage. There are four groups of information: user, heart-rate, emergency number and control point. Each category is related to its corresponding user through an identifier.

A.3.5. Development and implementation

Once the design has been done is necessary to make it in practice. The system is divided in two separate parts: Android app and Raspberry with sensor.

A.3.5.1. Android app

The first implemented part is formed by the Android app. This subsystem is composed by app-smartband and app-firebase interconnections. Now, there are going to be introduced some concepts about Android Studio, Firebase configuration, smartband connection and the graphical part of the app.

In order to create the Android app, it is required the access to Android SDK, that enables communication between the operating system and the developed SW. Android Studio is the official Android IDE, built over IntelliJ IDEA, design specifically for Android

development. In this project is used version 3.4 and the logic part is written with Java language. The API version shows the minimum required version that devices should support in order to use the app. In this case, it has been used API 23 which enables functionalities such as Firebase authentication and real-time database. At the moment of writing this document, API 23 can be used in the 62,6% of the Android devices.

An Android project is composed by numerous files, some of them for the logical and graphical part and others used for different properties of the app. The file “AndroidManifest.xml” gives info XML about the app, the different activities and the required permissions needed for a correct working. In the created app, there are needed permissions for accessing internet, bluetooth, Bluetooth admin properties and location. Also, in order to send SMS alarms is required the permission associated for sending SMS messages.

As seen in this summary, one of the parts that complete the system is the use of some Firebase services. In particular, there are used the Authentication and real-time database services. Thanks to authentication it is obtained a reference to each individual user and makes possible the user management throw Firebase console. Firebase database provides reading and writing features in a cloud database. Thank to these properties, it is possible to save information about the different users in the cloud and, also, it is possible to access to the same information from any place and any device with the same authentication. In the project there are two activities dedicated for the login and register tasks. Also, in the configuration, profile and control panels activities there are used operations in the firebase database, read or write.

The smartband Xiaomi Mi Band 2, used in the system, is a BLE device. BLE is a wireless and low-power consumption technology used in IoT applications, between others. The main BLE elements are the Generic Attribute Profile (GATT), the Attribute Protocol (ATT), Characteristics and Descriptors. These elements form the BLE protocol from a general point of view. BLE is used in two app’s activities: SearchDevice and HomePatient. In SearchDevice activity it is done the BLE scanner and only the MI Band 2 devices are able to connect with the application in the HomePatient activity. For the BLE scanner it is needed a BluetoothAdapter Java object and one BluetoothManager instance, between others. However, the activity with more functionalities is the HomePatient activity. It connects the smartband when connect button is pressed and is the responsible of reading the descriptor values continuously. Also, in this activity is

implemented the SMS alarms management. When there are some irregular values, a function called “sendAlertSMS” is used. This function uses the emergency numbers saved in Firebase and send a SMS message including the heart rate.

A.3.5.2. Raspberry Pi in the home

The developed system is formed by some subsystems, as presented before. This epigraph is related to the Raspberry implementation with an ultrasonic sensor. The purpose of this block is to provide a control over the monitored patient. This sensor should be set over the door of one of the most used rooms in the home of the patient so that each time the person is detected by the sensor it notifies the application. Thanks to this, the supervisor can consider when a problem has occurred if it takes a long time since the patient passed from the sensor.

First of all, it is necessary to install the Raspbian OS in the Raspberry Pi. Throw VNC Server and Viewer is possible to work remotely in the device from any laptop. Once Raspbian is installed is time for HC-SR04 sensor, in charge of detecting patient activity. This sensor measures distances by sending a high frequency pulse and computing the time it takes to come back. There are used some jumper cables and resistances for configuring the connections between the sensor and the GPIO pins of the Raspberry Pi. Once sensor circuit is set up is necessary to write the program that measures the distance and send some data to the Firebase database. When presence detected, variable “fecha_last_control” is changed in Firebase by overwriting the time and date the patient has passed under the control point. This part adds an aggregate value to the application.

A.3.6. Testing

In order to test the application, there have been handmade and acceptance tests. Handmade tests include checking the correct app working while open it, registering users, configuring the parameters, asking for permissions, between others. There is one acceptance test that includes the project validation weather the processor agrees with the presented project. All tests have been carried out successfully.

A.4. Planning

In this section it is presented information related to the project planning, an important aspect of any project. Planning consist of making in a systematic and ordered form a relation between tasks that have to be done and the time each task is going to take.

There are presented the different phases of the project. Each phase is composed by some tasks and when they are finished the end of that phase is reached. The main phases of this project are:

- Project planning: first phase of the project. It is composed by the tasks since the project definition until the solution design, before implementation.
- System development: real implementation of the system. It is done once first phase ends.
- Testing and evaluation: this part incorporates the app testing and the tasks related to making changes in the system.
- Writing the report: phase in which this document is written up.

Once phases have been described, it is necessary to indicate the tasks that correspond to any of the phases. GanttProject is a free program that enables the task declaration and representation in a simple way. In chapter 4 are presented all tasks in detail.

In total, the number of working days was estimated in 100 days and 3 hours per day in average. However, this planning has not been accomplished in some intermediate tasks, but ending up in the established time.

A.5. Conclusions and future work

As presented in this report, this project has tried to implement a solution for an observed problem in society. Thanks to the developed system, elderly people may have a real-time monitoring by their families and the reception of alarms in case of detecting any irregularity.

Project and personal objectives have been accomplished. As seen in the previous section, all tests were done successfully which means that all established requirements have been achieved.

Despite the fact that all objectives and requirements have been achieved, the project can incorporate new functionalities and improvements. Thanks to the modular design, new functionalities addition does not entail in a readjust of the work done. Some of the future improvements are:

- Compatibility with different smartband models: to date, the system developed until only works with Xiaomi Mi Band 2. Whether it makes compatible other types of smartband the penetration level will be much higher.

- Incorporate new sensors in the home: to date, only ultrasonic sensor is implemented in the Raspberry. There can be added some MQ sensors that detects different types of gas such as carbon monoxide, extremely dangerous in homes.
- Artificial Intelligence addition: throw the distance sensor it is pretended to take control of the patient activity in the home. Elderly people, specially if they have suffered an ictus, often follow a determinate activity pattern. Thanks to Artificial Intelligence, it can be implemented an algorithm that performs predictions of problems by learning the activity pattern of the patient.
- Internalization: to date, only Spanish language is available in the app. Next step should offer more languages like English. Android Studio makes easier the procedure of adding new languages to the app.

These feature improvements are some of the possible implementations that could be carried out after this project. As a personal opinion, this project could be improved in a lot of functional aspects. However, it pretends to be the base for feature developments and, above all, it tries to help a lot of elderly people and their families.

BIBLIOGRAFÍA

- [1] *España en cifras 2018*. Madrid: Edita INE, 2019, pp. 17-24.
- [2] "Cifras de población", *Ine.es*, 2019. [En línea]. Disponible: https://www.ine.es/infografias/infografia_dia_poblacion.pdf. [Acceso: 02-Ago- 2019].
- [3] "Bluetooth low energy overview | Android Developers", *Android Developers*. [En línea]. Disponible: <https://developer.android.com/guide/topics/connectivity/bluetooth-le>. [Acceso: 17- Ago- 2019].
- [4] "IDC Forecasts Worldwide Spending on the Internet of Things to Reach \$745 Billion in 2019, Led by the Manufacturing, Consumer, Transportation, and Utilities Sectors", *IDC: The premier global market intelligence company*, 2019. [En línea]. Disponible: <https://www.idc.com/getdoc.jsp?containerId=prUS44596319> . [Acceso: 04-Ago- 2019].
- [5] "BOE.es - Documento BOE-A-2018-16673", *Boe.es*, 2018. [En línea]. Disponible: <https://www.boe.es/buscar/doc.php?id=BOE-A-2018-16673> . [Acceso: 03- Ago- 2019].
- [6] A. ximdex, "index", *Aepd.es*, 2019. [En línea]. Disponible: <https://www.aepd.es/reglamento/derechos/index.html> . [Acceso: 05- Ago- 2019].
- [7] "Privacy and Security in Firebase | Firebase", *Firebase*, 2019. [En línea]. Disponible: <https://firebase.google.com/support/privacy/> . [Acceso: 09- Ago- 2019].
- [8] "Factores de riesgo - Fundación Española del Corazón", *Fundaciondelcorazon.com*. [En línea]. Disponible: <https://fundaciondelcorazon.com/prevencion/riesgo-cardiovascular.html> . [Acceso: 06- Ago- 2019].
- [9] A. Abellán García, P. Aceituno Nieto, J. Pérez Díaz, D. Ramiro Fariñas and R. Pujol Rodríguez, *Un perfil de las personas mayores en España, 2019. Indicadores estadísticos básicos*. Madrid, 2019.
- [10] G. Eysenbach, "What is e-health?", 2019. Disponible: <https://www.jmir.org/2001/2/e20/?newDesign> . [Acceso 9 Agosto 2019].
- [11] "Educational infographic: Ojo, que esto está ya cerquita: La tecnología wearable que nos espera - InfographicNow.com | Your Number One Source For daily infographics & visual creativity", *InfographicNow.com | Your Number One Source For daily infographics & visual creativity*, 2019. [En línea]. Disponible: <https://infographicnow.com/educational/languages/educational-infographic-ojo-que->

esto-esta-ya-cerquita-la-tecnologia-wearable-que-nos-espera-wearabl/ . [Acceso: 12- Ago- 2019].

[12] "¿Que es Wearable? - Los dispositivos vestibles", *Dispositivos Wearables*, 2014. [En línea]. Disponible: <http://www.dispositivoswearables.net/> . [Acceso: 12- Ago- 2019].

[13] "IDC Forecasts Steady Double-Digit Growth for Wearables as New Capabilities and Use Cases Expand the Market Opportunities", *IDC: The premier global market intelligence company*, 2019. [En línea]. Disponible: <https://www.idc.com/getdoc.jsp?containerId=prUS44930019> . [Acceso: 11- Ago- 2019].

[14] H. F. Atlam, R. Walters and G. Wills, "Internet of Things: State-of-the-art, Challenges, Applications, and Open Issues", vol. 9, no. 3, 2018. Disponible: https://www.researchgate.net/publication/327845273_Internet_of_Things_State-of-the-art_Challenges_Applications_and_Open_Issues . [Acceso 12 Agosto 2019].

[15] J. Clement, "Mobile app revenues 2014-2023 | Statista", *Statista*, 2019. [En línea]. Disponible: <https://www.statista.com/statistics/269025/worldwide-mobile-app-revenue-forecast/> . [Acceso: 15- Ago- 2019].

[16] *Informe ditrendia: Mobile en España y en el Mundo 2018*. 2019, pp. 41-51.

[17] "Saber Es MAS Que Aprender ¡Aca Vas A Saber Hacer! | Blog Arduino Y Tecnología", *PatagoniaTec*, 2019. [En línea]. Disponible: <https://saber.patagoniatec.com/2014/03/arduino-uno-r3-2-arduino-argentina-ptec/> . [Acceso: 14- Ago- 2019].

[18] *Raspberrypi.org*, 2019. [En línea]. Disponible: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/> . [Acceso: 12- Ago- 2019].

[19] "Sensor de Distancia de Ultrasonido HC-SR04 - Electronilab", *Electronilab*. [En línea]. Disponible: <https://electronilab.co/tienda/sensor-de-distancia-de-ultrasonido-hc-sr04/> . [Acceso: 13- Ago- 2019].

[20] "BOE.es - Documento consolidado BOE-A-1978-31229", *Boe.es*, 1978. [En línea]. Disponible: <https://www.boe.es/buscar/act.php?id=BOE-A-1978-31229> . [Acceso: 12- Aug- 2019].

[21] "Seguridad Social: Cotización / Recaudación de Trabajadores", *Seg-social.es*, 2019. [En línea]. Disponible: <http://www.seg-social.es/wps/portal/wss/internet/Trabajadores/CotizacionRecaudacionTrabajadores/36537> . [Acceso: 12- Ago- 2019].

- [22] "Core app quality | Android Developers", *Android Developers*. [En línea]. Disponible: <https://developer.android.com/docs/quality-guidelines/core-app-quality> . [Acceso: 15- Ago- 2019].
- [23] "Setting up your Raspberry Pi", *Projects.raspberrypi.org*. [En línea]. Disponible: <https://projects.raspberrypi.org/en/projects/raspberry-pi-setting-up/5> . [Acceso: 15- Ago- 2019].
- [24] "Medir distancia con Arduino y sensor de ultrasonidos HC-SR04", *Luis Llamas*, 2015. [En línea]. Disponible: <https://www.luisllamas.es/medir-distancia-con-arduino-y-sensor-de-ultrasonidos-hc-sr04/> . [Acceso: 15- Ago- 2019].
- [25] "Using a Raspberry Pi distance sensor (ultrasonic sensor HC-SR04)", *Raspberry Pi Tutorials*. [En línea]. Disponible: <https://tutorials-raspberrypi.com/raspberry-pi-ultrasonic-sensor-hc-sr04/> . [Acceso: 14- Ago- 2019].
- [26] "Android Studio", *En.wikipedia.org*, 2019. [En línea]. Disponible: https://en.wikipedia.org/wiki/Android_Studio . [Acceso: 12- Ago- 2019].
- [27] "Descripción general del manifiesto de una app | Android Developers", *Android Developers*. [En línea]. Disponible: <https://developer.android.com/guide/topics/manifest/manifest-intro?hl=es-419> . [Acceso: 16- Ago- 2019].
- [28] "FirebaseAuth | Google APIs for Android | Google Developers", *Google Developers*. [En línea]. Disponible: <https://developers.google.com/android/reference/com/google/firebase/auth/FirebaseAuth> . [Acceso: 17- Ago- 2019].
- [29] "Estructura tu base de datos | Firebase Realtime Database", *Firebase*. [En línea]. Disponible: <https://firebase.google.com/docs/database/web/structure-data?hl=es-419> . [Acceso: 16- Ago- 2019].
- [30] "Bluetooth Low Energy", *En.wikipedia.org*, 2019. [En línea]. Disponible: https://en.wikipedia.org/wiki/Bluetooth_Low_Energy . [Acceso: 12- Ago- 2019].
- [31] "Android", *Es.wikipedia.org*, 2019. [En línea]. Disponible: <https://es.wikipedia.org/wiki/Android> . [Acceso: 7- Ago- 2019].
- [32] "Mobile Operating System Market Share Worldwide | StatCounter Global Stats", *StatCounter Global Stats*, 2019. [En línea]. Disponible: <https://gs.statcounter.com/os-market-share/mobile/worldwide/> . [Acceso: 13- Ago- 2019].

- [33] A. Ribes, "Rithmi - Tecnología wearable para la salud del corazón", *Rithmi*, 2019. [En línea]. Disponible: <https://rithmi.com/> . [Acceso: 13- Ago- 2019].
- [34] "Teleasistencia - Teleasistencia en casa - Cruz Roja", *Cruzroja.es*, 2019. [En línea]. Disponible: <https://www.cruzroja.es/principal/web/teleasistencia/teleasistencia-en-casa> . [Acceso: 13- Ago- 2019].
- [35] "Hacer un electrocardiograma con la app ECG del Apple Watch Series 4", *Apple Support*, 2019. [En línea]. Disponible: <https://support.apple.com/es-es/HT208955> . [Acceso: 13- Ago- 2019].
- [36] "Creating apps: All the differences between Android and iOS", *En.yeeply.com*, 2019. [En línea]. Disponible: <https://en.yeeply.com/blog/creating-apps-differences-android-ios/> . [Acceso: 15- Ago- 2019].
- [37] "Raspberry Pi 3 vs. Arduino Uno Rev3", *Arrow.com*, 2018. [En línea]. Disponible: <https://www.arrow.com/es-mx/research-and-events/articles/comparing-arduino-uno-and-raspberry-pi-3> . [Acceso: 13- Ago- 2019].