

APLICACIÓN MÓVIL PARA LA MONITORIZACIÓN DE PACIENTE DE CÁNCER

APP FOR MONITORING CANCER PATIENTS

Carlos Gómez Amador

GRADO EN INGENIERIA INFORMÁTICA. FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTESNE DE MADRID



Trabajo Fin de Grado en Ingeniería Informática.

Curso 2019/2020
Convocatoria Febrero de 2020

Director:

Iván García-Magariño García

Resumen

Este proyecto tiene como finalidad el desarrollo de un sistema para la recolección de datos en pacientes con cáncer, con el fin de poder determinar enfermedades mentales tales como depresión y ansiedad.

El sistema estará formado por una aplicación android, que recogerá información de forma activa mediante el uso de formularios a los pacientes, y de forma pasiva realizando mediciones en dispositivos wearables conectados al dispositivo móvil, además el sistema dispondrá de un servicio web en el que almacenar los datos recogidos por las pulseras inteligentes y de las respuestas ofrecidas en los formularios para su posterior análisis por parte del personal sanitario.

Palabras clave

Android, BLE, monitorización, cliente-servidor.

Abstract

This project aims to develop a system for collecting data in cancer patients, in order to determine mental illnesses such as depression and anxiety. The system will consist of an android application, which will actively collect information through the use of forms to patients, and passively making measurements on wearables devices connected to the mobile device, in addition the system will have a web service in which to store the data collected by the smart bands and the answers offered in the forms for later analysis by medical staff.

Keywords

Android, BLE, monitoring, client-server.

Índice general

Índice	I
1. Introducción	1
1.1. Motivación	1
1.2. Objetivo	1
1.3. Plan de trabajo	2
2. Estado del arte	3
2.1. Enfermedades oncológicas	3
2.1.1. Factores de riesgo	3
2.1.2. Efectos Emocionales y Psicológicos	4
2.2. Tecnologías actuales	4
2.3. Smartbands	4
2.3.1. Características	5
2.3.2. Estudio de mercado	6
2.4. Aplicaciones similares	9
3. Arquitectura del sistema	13
3.1. Arquitectura global	13
3.2. App android	14
3.2.1. Recogida de datos activa	15
3.2.2. Recogida de datos mediante dispositivos wearables	17
3.2.3. Interfaz gráfica	19
3.3. Api RESTful	25
3.3.1. Securización de las conexiones	26
3.3.2. Base de datos	27
3.3.3. Tecnologías empleadas	29
4. Experimentación	30
4.1. Pruebas de rendimiento	31
4.1.1. Medidas tiempo API	31
4.1.2. Profiling app android	31
4.1.3. Prueba con usuarios	34
5. Conclusiones y trabajo futuro	39
5.1. Conclusiones	39
5.2. Trabajo futuro	40

5. Conclusions and future work	41
5.1. Conclusions	41
5.2. Future work	42
Bibliography	43
A. Manual de usuario de la API	44
A.1. Login	44
A.1.1. Ejemplo de petición	44
A.1.2. Respuestas	44
A.2. Registro	45
A.2.1. Ejemplo de petición	45
A.2.2. Respuestas	45
A.3. Obtener usuario por su id	46
A.3.1. Ejemplo de petición	46
A.3.2. Respuestas	46
A.4. Obtener usuario por su email	46
A.4.1. Ejemplo de petición	47
A.4.2. Respuestas	47
A.5. Listar todos los usuarios	47
A.5.1. Ejemplo de petición	47
A.5.2. Respuestas	48
A.6. Actualizar usuario	48
A.6.1. Ejemplo de petición	48
A.6.2. Respuestas	49
A.7. Obtener los tests disponibles para un usuario	49
A.7.1. Ejemplo de petición	49
A.7.2. Respuestas	50
A.8. Obtener un test por su id	50
A.8.1. Ejemplo de petición	50
A.8.2. Respuestas	50
A.9. Obtener un test por su nombre	51
A.9.1. Ejemplo de petición	52
A.9.2. Respuestas	52
A.10. Guardar respuestas de un test	53
A.10.1. Ejemplo de petición	53
A.10.2. Respuestas	54
A.11. Obtener datos de frecuencia cardíaca de un usuario	54
A.11.1. Ejemplo de petición	54
A.11.2. Respuestas	54
A.12. Obtener datos de pasos de un usuario	55
A.12.1. Ejemplo de petición	55
A.12.2. Respuestas	55

A.13.Subir medición de frecuencia cardíaca	56
A.13.1.Ejemplo de petición	56
A.13.2.Respuestas	56
A.14.Subir medición de pasos	56
A.14.1.Ejemplo de petición	57
A.14.2.Respuestas	57

Capítulo 1

Introducción

1.1. Motivación

Vivimos en un mundo rodeado de tecnologías, donde es frecuente disponer de varios dispositivos por persona conectados a internet, los cuales tratan de hacernos la vida mucho mas sencilla y simple, nos permite estar interconectados con personas de forma instantánea sin importar la distancia que nos separa, así como resolver contratiempos en breves periodos de tiempo.

Con esta idea en mente, me planteé como se podría aprovechar todas las ventajas que nos aportan las tecnologías para ofrecer una herramienta que pudiera servir de utilidad a la sociedad. Fue entonces cuando la lectura de un articulo sobre las consecuencias psicológicas que sufren los pacientes de cáncer⁶² despertó el interés en como poder ayudar a estas personas que tienen que afrontar una enfermedad de este calibre y los efectos secundarios que tienen en ellas los tratamientos actuales.

1.2. Objetivo

El proyecto tiene como objetivo el desarrollo de una aplicación móvil con la que recoger datos sobre pacientes con cáncer de la manera menos intrusiva posible. Para ello deberá recoger las respuestas a cuestionarios y deberá recoger datos sobre dispositivos wearables. La aplicación deberá ser lo mas intuitiva posible con el fin de evitar que la aplicación caiga en desuso por la complejidad de esta.

Además el proyecto deberá incorporar un sistema en el lado del servidor en el que almacenar la información recogida con la aplicación.

Dado el carácter de la información que se manejará en la aplicación se vuelve imprescindible que el sistema tenga una comunicación segura en todo momento, con la finalidad de evitar la fuga de datos.

Capítulo 2

Estado del arte

2.1. Enfermedades oncológicas

Se conoce como enfermedades oncológicas o cáncer a aquellas enfermedades en las que algunos tipos de células del organismo comienzan a dividirse de forma descontrolada y sin detenerse, pudiendo diseminarse a otros tejidos sanos del propio organismo.

El cáncer puede originarse en cualquier parte del organismo del ser humano, el cual esta formado por billones de células, estas en condiciones normales son remplazadas por células nuevas cuando envejecen o se dañan, permitiendo al organismo crecer y continuar sus funciones vitales.

Sin embargo, en el cáncer este proceso se descontrola. Las células comienzan a presentar anomalías, las células viejas o dañadas no mueren, al mismo tiempo que el organismo genera nuevas células para remplazarlas. A su vez estas nuevas células continúan dividiéndose sin morir, dando lugar a masas tumorales.³

2.1.1. Factores de riesgo

Actualmente se desconoce el motivo por el que unas personas padecen de cáncer y otras no. Numerosas investigaciones realizadas, indican el aumento de probabilidad de padecer este tipo de enfermedades con factores de riesgo.

Dentro de los factores nos podemos encontrar con riesgos relacionados con la conducta, como pueden ser fumar, consumo de alcohol, dieta poco saludable; riesgo ambientales, donde nos encontramos exposiciones a elementos propios del entorno como radiación ultravioleta, polución, pesticidas, etc; riesgo biológicos, relacionados directamente con la edad, grupo étnico; y por último nos encontramos con los riesgos hereditarios, en los que destaca la carga genética producto de la herencia que se heredan de los padres.³

2.1.2. Efectos Emocionales y Psicológicos

Ansiedad

La ansiedad es un mecanismo de los seres vivos, que permite poner al individuo en un estado de altera frene a situaciones de peligro o desconocidas. Unos niveles de ansiedad moderados contribuyen a que se mejore la atención y la concentración para hacer frente a la situación adversa, en cambio cuando los niveles de ansiedad se disparan el efecto que se obtiene es justo lo contrario, el individuo tiende a bloquearse y a no reaccionar. En los pacientes con diagnostico de cáncer saben que es una enfermedad peligrosa, por lo que hay una tendencia clara en el desarrollo de este tipo de trastornos¹² *"Casi el 48 % de los pacientes con cáncer divulgan altos niveles de ansiedad y el 18 % demuestra desórdenes relacionados con la ansiedad."*¹²

Depresión

La depresión es un trastorno del estado de ánimo que afecta al estado anímico de la persona, a la calidad del sueño, al desarrollo de la vida cotidiana, a la alimentación, etc.¹² *"Se estima que entre 16 % y 25 % de los pacientes con cáncer desarrollan depresión."* La depresión tiene una mayor tasa de incidencia en pacientes diagnosticados con cáncer frente al resto de la población, siendo las mujeres y personas de edad avanzada los mas propensos a sufrir este tipo de enfermedades.

La depresión de no ser tratada suele presentar numerosos efectos negativos en la salud de los pacientes, puesto que suelen tener un impacto negativo sobre el tratamiento, provocando una recuperación mas lenta, disminuir la probabilidad de supervivencia o conducir al paciente al suicidio, por lo que es especialmente importante atender este trastorno.¹²

2.2. Tecnologías actuales

En este apartado se realizará una introducción a los dispositivos wearables, en especial a las smartbands, puesto que serán los dispositivos empleados para el desarrollo del proyecto. Se va a realizar un estudio de las diferentes bandas de actividad disponibles en el mercado exponiendo sus características y funcionalidad. Por otro lado se analizaran diferentes aplicaciones cuya funcionalidad sea similar a la de nuestro proyecto, donde veremos que ofrece cada una.

2.3. Smartbands

Las smartbands aparentemente son simples pulseras que varían su diseño en función del fabricante, pudiendo disponer de pantalla o no, pero todas se caracterizan por incorporar en su interior varios sensores con los que recoger información del exterior, llegando a encontrarlos en ellas pulsómetros, para realizar mediciones del ritmo cardíaco; GPS, para realizar seguimiento de velocidad y distancia; podómetros, con los que contabilizar los pasos; así

como giróscopos y acelerómetros entre otros sensores.

En el mercado actual existe una gran cantidad de pulseras inteligentes debido a su bajo precio y gran popularidad, aumentando año tras año el número de venta de estos dispositivos. Al ser un campo relativamente nuevo y siendo unos dispositivos destinados a un público general, nos encontramos con unos sensores que carecen de la misma precisión que otros destinados a un ámbito médico o profesional, pero el continuo desarrollo de estos dispositivos hace que día a día su fiabilidad y precisión vayan mejorando.

2.3.1. Características

Como se ha indicado anteriormente no todas las pulseras cuentan con los mismos sensores, generalmente varían en función del precio de cada dispositivo, los sensores más comunes son los siguientes.

- **Pulsómetro:** Este sensor está formado por uno o varios diodos verdes e infrarrojos, cuya función es la de emitir luz; y fotodiodos, cuyo propósito es captar la luz reflejada. El sensor al estar cerca de la piel es capaz de iluminar las venas y arterias más cercanas a la piel, que a su vez emiten un reflejo de vuelta que indica el nivel de sangre en ese lugar en concreto. Nuestro corazón al bombear hace que la sangre varíe decenas de veces por minuto, por lo que con cada latido se produce una variación de la intensidad de percibida por los fotodiodos, y analizando estas variaciones el sensor es capaz de determinar la frecuencia cardíaca.
- **Bluetooth:** todas las pulseras cuentan con conexión bluetooth, en especial Bluetooth BLE, con el que permite conectarse a otros dispositivos para intercambiar información.
- **Acelerómetro:** Existen diversos tipos en función del principio físico que usen, siendo los más comunes los acelerómetros piezoeléctricos. Su cometido es la medición de la fuerza y orientación de la aceleración, con el fin de determinar la intensidad de la actividad física.
- **GPS:** El GPS suele encontrarse en pulseras de alta gama, usualmente permite conectarse a distintas constelaciones de satélites con el objetivo de realizar mediciones de velocidad y muestreo de posiciones.
- **Termómetro:** Es un sensor que permite medir la temperatura de nuestra piel, el aumento de actividad física conlleva un incremento de temperatura en nuestro organismo, por lo que generalmente se utiliza para medir intensidad de las actividades físicas.
- **Sensor de respuesta galvánica de la piel (GSR),** también conocida como conductancia de la piel, mide la variación en las características eléctricas de la piel, generalmente causadas por la sudoración del cuerpo humano, el cual ayuda a la monitorización de las actividades físicas y estados de las personas.

- Sensor de bioimpedancia, generalmente usado para la medición del porcentaje de oxígeno en sangre, el cual permite realizar un análisis de la respiración e hidratación de la persona.
- Sensor barométrico, usado para medir los cambios de altura en función de la presión atmosférica en un lugar en concreto. A medida que aumentamos la altitud, la presión atmosférica se ve decrementada de forma cuadrática, por lo que partiendo que a nivel del mar nos encontramos con una atmósfera, podemos realizar una estimación de la altitud. Este sensor se ve afectado por el tiempo climático, por lo que sus datos son orientativos.

2.3.2. Estudio de mercado

Dado la gran variedad de fabricantes y modelos de pulseras inteligentes, se ha realizado un estudio de mercado analizando modelos con el fin de obtener que características son las mas comunes, para tratar de hacer nuestro proyecto lo mas abierto posible en cuanto a dispositivos.

Vivosport HR

Esta pulsera del fabricante Garmin, cuenta con sensor cardíaco con el que monitorizar las calorías quemadas y cuantificar la intensidad de las actividades, ofreciendo un resumen del tiempo en el que el usuario ha permanecido en una zona de esfuerzo. Además incorpora un sensor barométrico el cual se emplea para medir el número de pisos subidos.



Figura 2.1: *Pulsera Vivosport HR*

Mi Band 4

Pulsera desarrollada por Xiaomi, en la que podemos encontrar un acelerómetro y giroscopio de 3 ejes con el que detectar la actividad del usuario, junto con un sensor de frecuencia cardíaca con posibilidad de monitoreo continuo, además ofrece la posibilidad de realizar un análisis de la calidad del sueño diferenciando en tres zonas, despierto, sueño ligero y sueño profundo.



Figura 2.2: *Pulsera Xiaomi mi band 4*

Charge 3

Es la pulsera mas avanzada de la marca Fitbit. Cuenta con sensor de frecuencia cardiaca para monitorización continua, además se emplea para realizar una estimación de calorías quemadas a lo largo del dia así como para determinar la fase de actividad en la que se encuentra el usuario (Quema de grasas, Cardio o Pico). Esta pulsera ofrece la posibilidad de la monitorización de la calidad del sueño distinguiendo las fases de sueño ligero, profundo y REM. Un apartado en el que se desmarca es monitorización del periodo del sexo femenino, llegando a establecer predicciones de cuando sucederá el siguiente periodo.



Figura 2.3: *Pulsera Fitbit Charge 3*

Gear Fit2 Pro

Esta pulsera de la marca Samsung, cuenta con acelerómetros, giróscopos y barómetros con los que realiza el seguimiento de la actividad física del usuario, además ofrece sensor de frecuencia cardíaca con el que monitorizar el ritmo cardíaco, si embargo no se especifica si estas mediciones son continuadas o realizadas en intervalos de tiempo. En cambio esta pulsera ofrece la posibilidad de realizar un seguimiento gracias a su GPS, por lo que puede medir distancias de forma precisa y no basada en estimaciones como el resto de las pulseras.



Figura 2.4: Pulsera samsung gear fit2

Band 3 Pro

La pulsera de la marca Huawei ofrece al igual que la pulsera anterior sensor de GPS, a diferencia de otras pulseras utiliza los datos recibidos de este sensor para hacer una estimación del volumen máximo de oxígeno. También ofrece sensor de frecuencia cardíaca con el que monitorizar el ritmo cardíaco, sin especificar si realiza la medición de forma continuada o de forma periódica.



Figura 2.5: Pulsera Huawei Band 3 Pro

Moov Now

A diferencia del resto de pulseras analizadas, la marca moov ha prescindido de una pantalla en la que ofrecer información al usuario, por lo que el usuario debe vincularla a un dispositivo móvil para comprobar sus datos. En cuanto a sensores, carece de sensor de

frecuencia cardíaca integrado, pero ofrece la posibilidad de conectar una banda de pecho para realizar este tipo de seguimiento, ofreciendo una mayor precisión por ser un sensor ECG, el cual utiliza la actividad eléctrica del corazón para obtener el valor. En cambio ofrece 3 giroscopios con los que determina los movimientos del usuario con una gran precisión, permitiendo la reconstrucción de los movimientos en tres dimensiones.



Figura 2.6: *Pulsera Moov Now*

2.4. Aplicaciones similares

Dado el amplio espectro de la tecnología y de la salud, la combinación de ambas da lugar a una infinidad de posibilidades, es por ello que en la actualidad muchas empresas están desarrollando aplicaciones relacionadas con e-health no solo para el uso domestico, si no que la digitalización del sistema sanitario ha propiciado la colaboración de empresas privadas con organismos públicos de sanidad.

i4park

i4park es una app basada para dispositivos android, destinada a pacientes con parkinson, desarrollada por la empresa i4life innovacion y desarrollos SL. Su funcionamiento esta basado en cuestionarios sencillos, que el usuario debe rellenar de forma periódica y que serán utilizados en la consulta médica mas próxima con el objetivo de establecer el estado anímico del paciente. Ofrece un sistema para llevar un registro de la medicación con un sistema basado en alarmas, siendo necesario que el paciente registre la toma de la dosis para llevar el seguimiento completo.



Figura 2.7: *app i4park*

Beneca

Beneca es una app basada en web, cuyo objetivo es la monitorización del balance energético de pacientes con cáncer, está desarrollada por la universidad de Granada. Para realizar el seguimiento el usuario debe realizar un registro con los alimentos que ha tomado y cantidades así como las actividades físicas que ha realizado a lo largo del día, con el fin de obtener un balance energético del paciente.

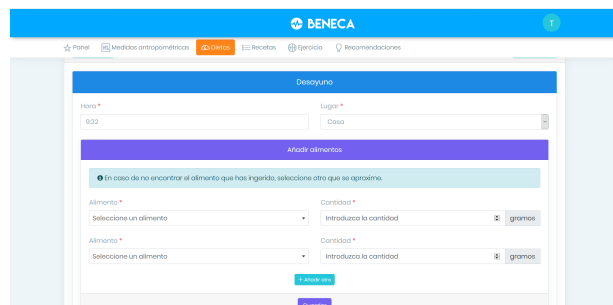


Figura 2.8: *App Beneca*

e-OncoSalud

Se trata de una aplicación para dispositivos móviles android e iOS desarrollada por el servicio de salud de farmacia del hospital Gregorio Marañón, la cual está destinada a pacientes con tratamientos de antineoplásicos orales, con el objetivo de registrar todo su historial de tratamiento, contando con funcionalidades como avisos de toma de medicamento además de ofrecer información sobre consejos nutricionales, síntomas, efectos adversos, etc.

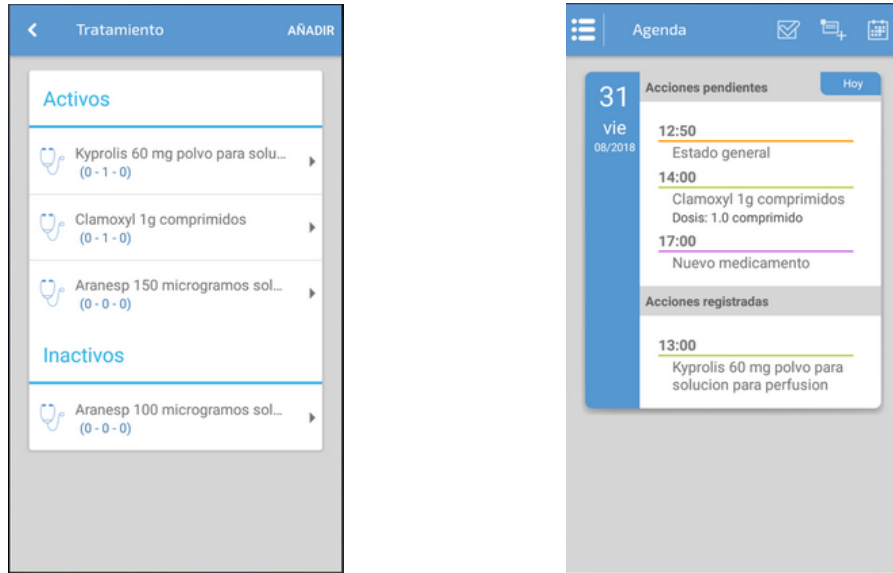


Figura 2.9: App e-OncoSalud

Rotterdam Prostate Cancer Risk

Es una aplicación android desarrollada por el departamento de urología de la universidad de Rotterdam y su validez médica ha sido estudiada en diversos estudios⁹. Su funcionamiento se basa en la recolección de datos procedentes de diversas pruebas realizadas al paciente, el cual debe introducir a mano, y en base a las respuestas y los datos introducidos la aplicación realiza un calculo sobre la probabilidad de tener cáncer de prostata.

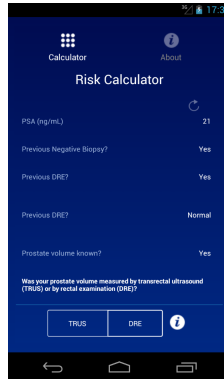


Figura 2.10: Rotterdam Prostate Cancer Risk

My pro health

Es una aplicación desarrollada para estudiar el efecto de los andrógenos en pacientes con cancer prostático¹ la cual recoge mediante formularios el nivel de compromiso del paciente en su propia atención médica, la dieta y la actividad física. Esta aplicación almacena directamente los datos de los formularios en los servidores del hospital.

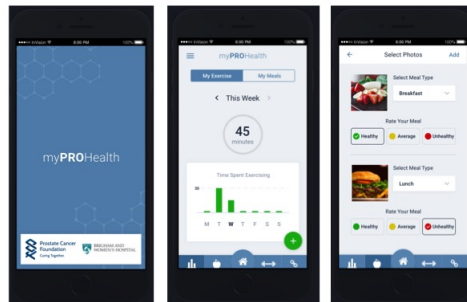


Figura 2.11: My pro health

Capítulo 3

Arquitectura del sistema

En este capítulo se detallará la arquitectura del sistema elegida, así como la arquitectura de los submódulos del sistema así como de las tecnologías empleadas en cada submódulo

3.1. Arquitectura global

Como se aprecia en la figura 3.1 la arquitectura escogida para el sistema ha sido un modelo cliente servidor, la aplicación móvil adquiere el rol de cliente y la API el rol de servidor. La comunicación entre la app y la API se realiza a través de internet, utilizando los verbos del protocolo HTTP (GET, POST, PUT y DELETE) se detallará más en el apartado 3.3

Esta arquitectura nos permite tener en tiempo real las respuestas de los cuestionarios y las mediciones para su tratamiento por parte del personal médico, así como poder añadir nuevos cuestionario de forma individualizada a cada paciente, sin la necesidad de tener que lanzar actualizaciones de la propia app con consiguientes molestias que esto puede ocasionar para los usuarios. Además al tener centralizada la información en el futuro nos permitirá aplicar machine learning para aprender patrones sobre los pacientes mejorando la efectividad del sistema.



Figura 3.1: *Arquitectura del sistema*

3.2. App android

La arquitectura escogida para la app ha sido el Model-View-Presenter⁸, ya que es una arquitectura que se adapta muy bien al framework de android⁴.

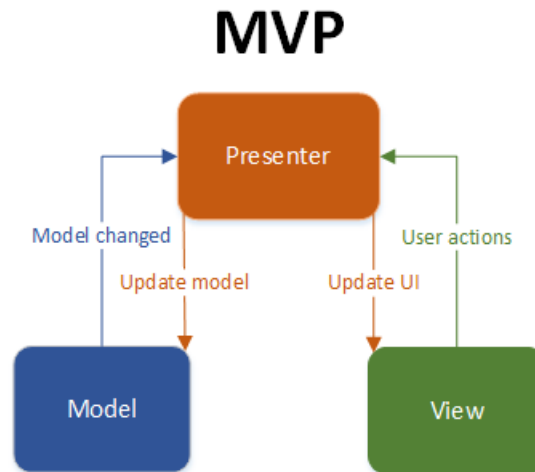


Figura 3.2: *Arquitectura app*

La principal ventaja del patrón MVP es el desacoplamiento de la vista del modelo, de tal forma que la modificación sobre una parte no afecta a la otra. Otra de las principales ventajas de este patrón es la testabilidad, pudiendo automatizar la ejecución de pruebas con la finalidad de comprobar si se cumplen los requisitos del sistema.

En el contexto de android la vista se corresponde con las activities y los fragments, siendo su finalidad la de captar la interacción del usuario sobre la UI y de transmitir estos eventos al presenter, además de mostrar la información asociada a la acción realizada por el usuario. Nótese que en esta capa es donde nos encontraremos toda la lógica del SDK de android.

La capa del presenter o presentador, es la encargada de formatear la información solicitada al modelo y transmitirle esta información a la capa de la vista con el formato adecuado, ya que la vista es pasiva.

La capa del modelo es la encargada de la representación de la información con la cual el sistema opera, siendo en esta capa donde se realizarán los accesos y modificaciones de los datos, esta capa es equivalente a la del modelo del patrón MVC.

3.2.1. Recogida de datos activa

La principal función de la aplicación es la de la recolección de datos a través de formularios establecidos por el personal médico, estos formularios son recogidos del servidor web y son mostrados al usuario con la intención de que los rellene diariamente.

Dada la posibilidad de tener una gran cantidad de preguntas que mostrar al usuario, la opción que mejor encajaba en el proyecto, era la creación de un formulario en un formato de lista. Partiendo de esta premisa y teniendo en cuenta las limitaciones que tienen los dispositivos móviles se ha decidido implementarlo con un RecyclerView junto con el patrón viewHolder.

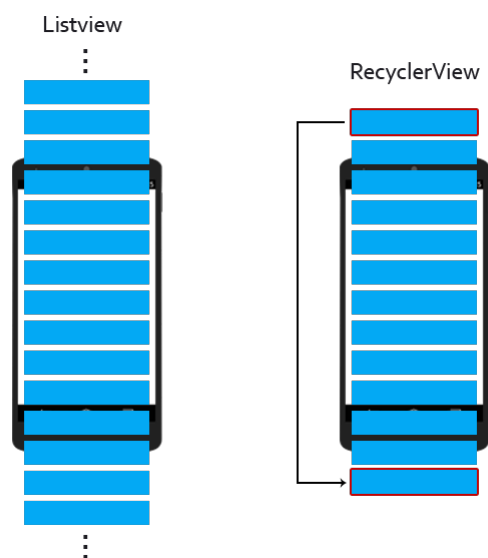


Figura 3.3: *Listview vs RecyclerView*

La diferencia entre un ListView y un RecyclerView, es la forma en la que gestionan los datos, la primera genera una vista para cada elemento que contiene la lista que se desea mostrar, de modo que para listas de tamaño muy grande tendremos guardados en memoria todas las vistas de todos los elementos, por el contrario RecyclerView solo genera las vistas que alberga la pantalla, de esta forma cuando un elemento de la lista sale de la pantalla, se reutiliza este elemento modificando su contenido y se inserta al final de los elementos a mostrar. Con esta vista, se mejora en la fluidez a la hora de hacer scroll, el consumo de memoria y en uso de CPU del dispositivo móvil.

Tanto si usamos la vista RecyclerView como ListView es necesario la utilización de un objeto adaptador, este será el encargado de actuar como un enlace entre la lista de datos y la vista, y aquí es donde entra en juego el patrón viewHolder.

Después de que un adaptador cree o reutilice una vista, busca el lugar dentro del layout donde tiene que mostrar la información localizando el id del componente de la vista. Con el patrón View Holder conseguimos que esa referencia se establezca cuando se crea la vista y se guarde para no tener que volver a buscar. Básicamente lo que se consigue con este patrón es evitar la búsqueda del id de componente cada vez que se tenga que mostrar un nuevo elemento de la lista, y dado que la búsqueda del id se realiza tantas veces como componentes tenga la vista por cada elemento de la lista, conseguimos aumentar el rendimiento del scroll y la experiencia del usuario.

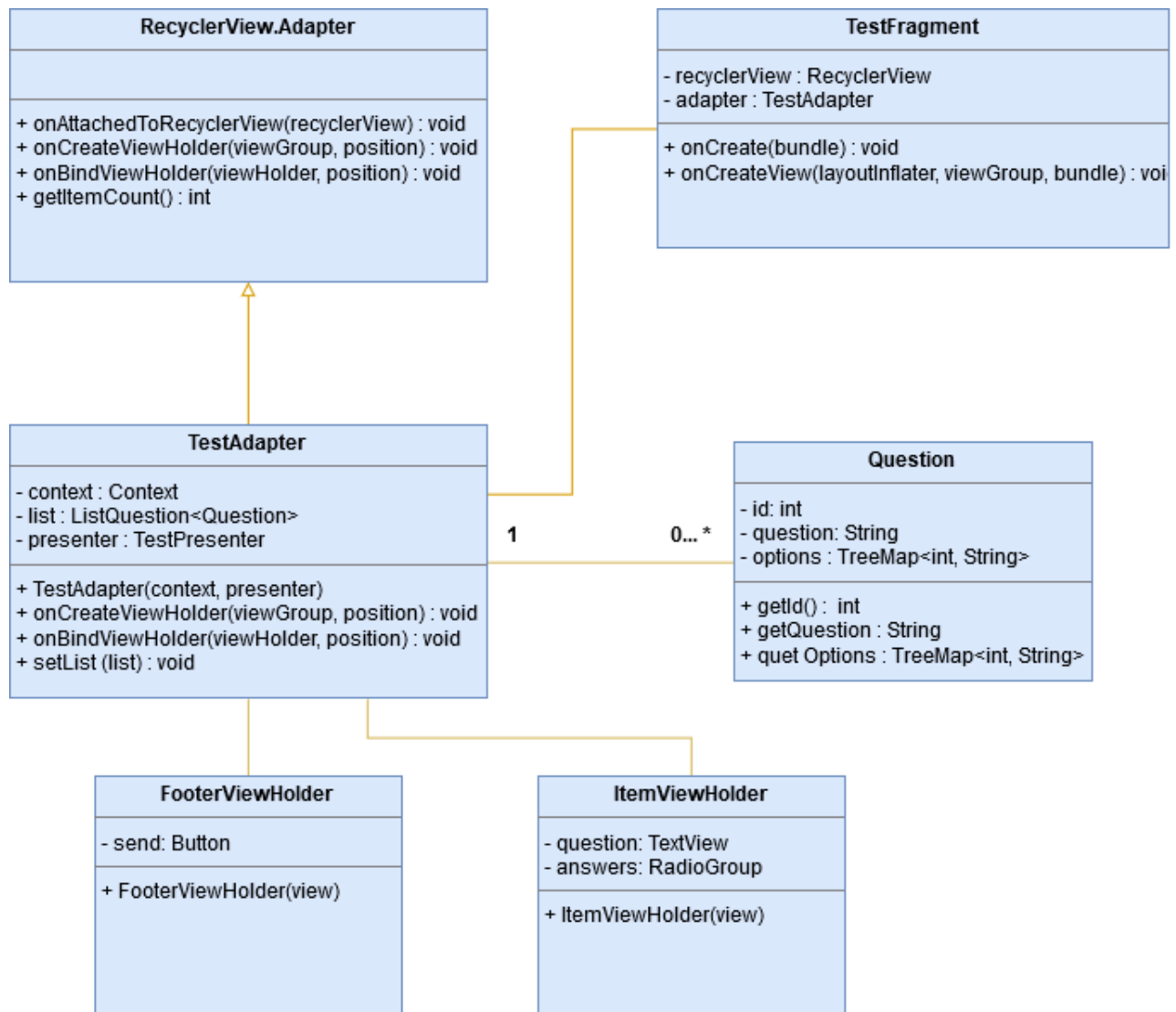


Figura 3.4: UML *recyclerView* y *viewHolder*

3.2.2. Recogida de datos mediante dispositivos wearables

La otra funcionalidad de la aplicación es la recogida de datos de dispositivos wearables, para ello se ha implementado la comunicación con estos dispositivos vía Bluetooth Low Energy (BLE)¹¹.

Estas lecturas se realizan de forma periódica y en segundo plano sin que el usuario sea consciente de cuando se realizan, con el fin de que el consumo de batería sea el mas bajo posible se ha implementado mediante el uso de alarmas. Puesto que se realizan lecturas tanto del ritmo cardíaco como de los pasos realizados por el usuario, se han usado dos alarmas independientes, la alarma de la frecuencia cardíaca se activa cada dos minutos y otra alarma para la lectura de los pasos al final del día, activándose una única vez.

Al registrar una alarma, estamos asociando un evento (intent) a esa alarma, por lo que cuando esta expira, el sistema operativo manda un broadcast a todas las aplicaciones suscritas a este evento. Para suscribirse basta con generar una clase que atienda estos eventos y registrarlo en el manifest de la aplicación, son los conocidos recivers, siendo aquí donde nos conectamos al dispositivo y lanzamos la lectura de los datos necesarios.

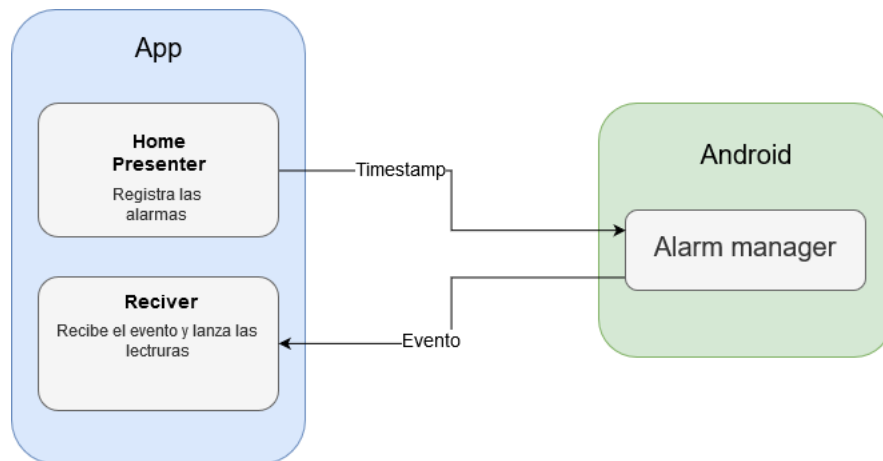


Figura 3.5: *Lecturas BLE segundo plano*

También se ha considerado necesario mostrar información al usuario sobre el dispositivo asociado a la aplicación, nivel de batería y pasos que lleva a lo largo del día, por ello se ha implementado la siguiente arquitectura con el aliciente de que sea escalable y en el futuro se puedan asociar varios dispositivos a la aplicación.

Todas las operaciones que sean necesarias sobre un dispositivo se realizaran sobre el Ble-Manager, puesto que al ser un singleton nos aseguramos que creamos una única conexión sobre el dispositivo y cuando se añadan varios dispositivos asociados a la aplicación este

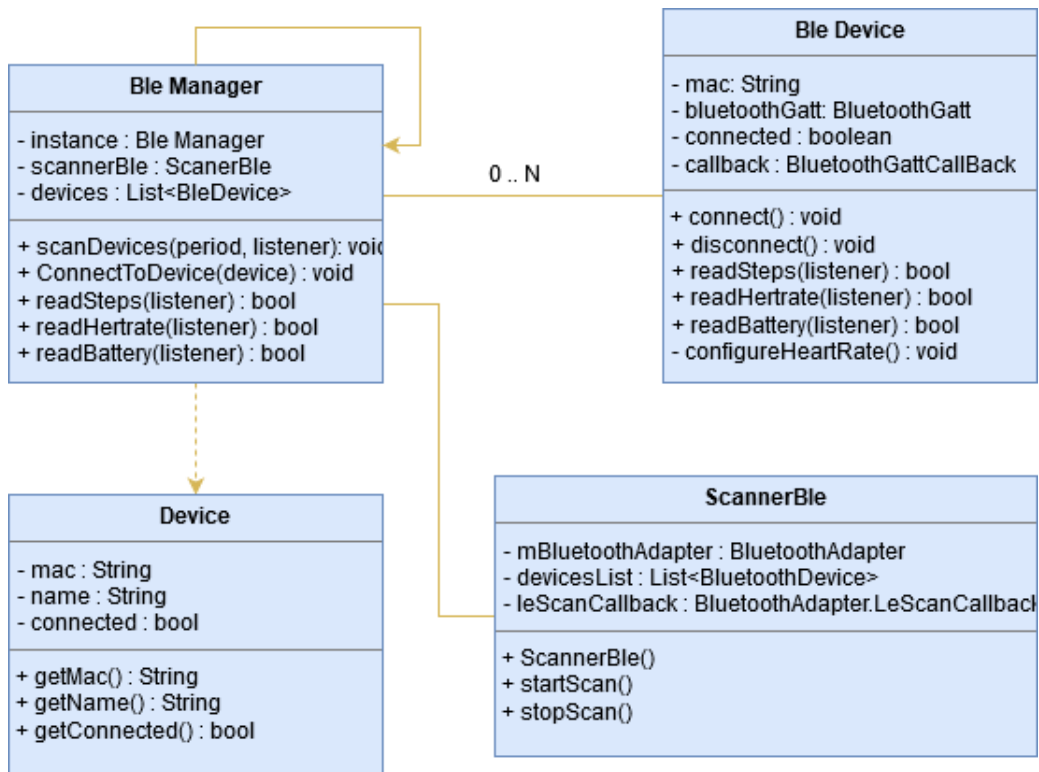


Figura 3.6: *UML BLE*

será el encargado de decidir sobre cual realiza la operación.

La clase BleDevice encapsula todo el comportamiento que puede realizar un dispositivo físico, y sobre esta clase se generan unos métodos para realizar las operaciones sobre el.

La clase Device es una clase del modelo, que recoge los datos guardados del dispositivo en el almacenamiento interno de la aplicación, nuestra aplicación interactuará con el Ble Manager con estas clases.

La clase ScannerBle es la encargada de realizar la operación de búsqueda de dispositivos y de pararla cuando transcurre el tiempo predefinido. Los resultados son devueltos al objeto que invoca la búsqueda mediante un callback.

3.2.3. Interfaz gráfica

A continuación se detallarán las vistas implementadas para la aplicación con las que el usuario interactuará.

Login

El login es la primera vista que el usuario se encontrará cuando inicie nuestra aplicación por primera vez. En ella deberá introducir su email y su contraseña para iniciar la sesión, si uno de ellos es incorrecto mostrará un mensaje indicando el error producido, y en caso de que la información proporcionada sea correcta, su sesión se quedará guardada en el sistema hasta que use la función de logout o elimine los datos de la aplicación desde la ventana de opciones de android.

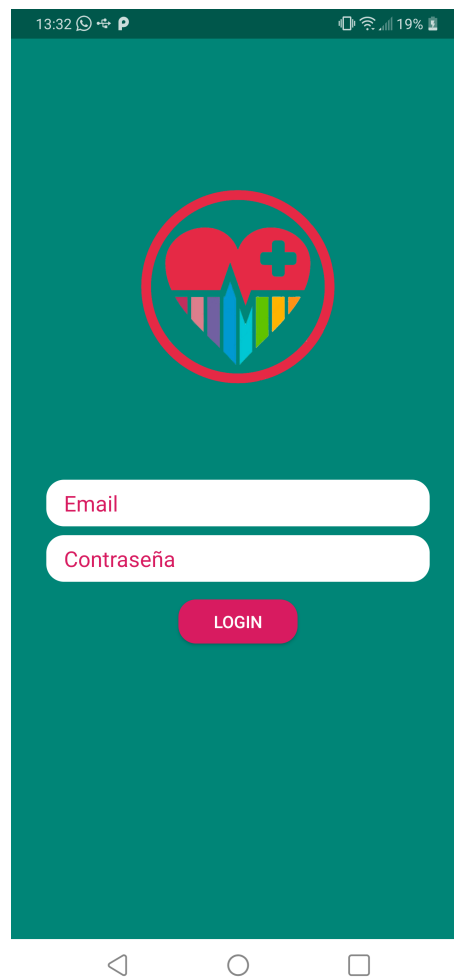


Figura 3.7: *vista login*

Solicitud de permisos

Una vez que se inicia sesión es necesario solicitar el permiso de ubicación, puesto que desde la versión 5.0 de Android es obligatorio el uso de dicho permiso para realizar búsquedas de dispositivos Bluetooth. Si el permiso es concedido por parte del usuario, la aplicación comprueba si tiene el módulo de localización activo, en caso de no tenerlo, se le pide al usuario si quiere activarlo, en caso de que la respuesta sea afirmativa, se le lleva a la ventana de ubicación del dispositivo Android, donde deberá ser él quien active el módulo.

En caso de que alguna de las dos solicitudes al usuario sean denegadas la aplicación se cerrará, puesto que es necesario la concesión del permiso y habilitar la localización para que la aplicación pueda vincular un dispositivo BLE a la aplicación.

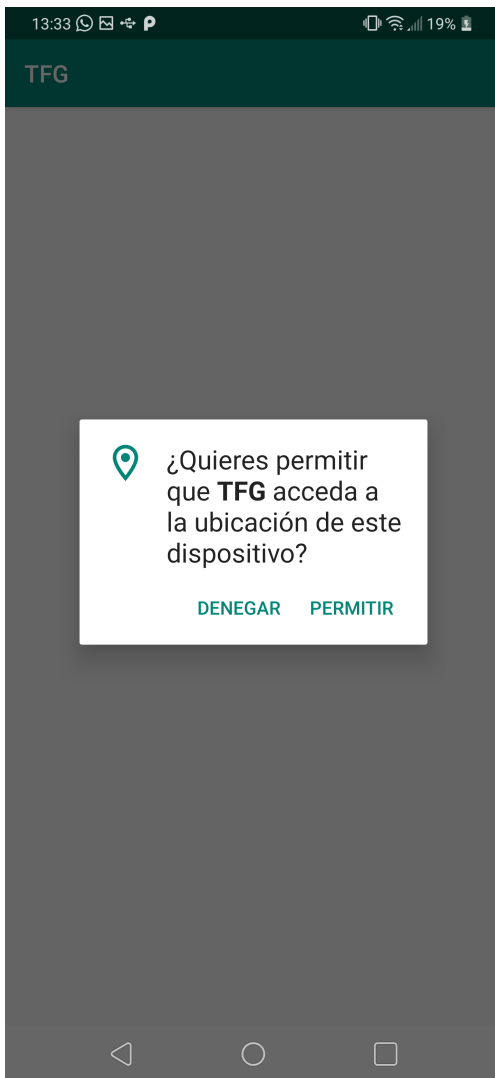


Figura 3.8: Vista solicitud permisos

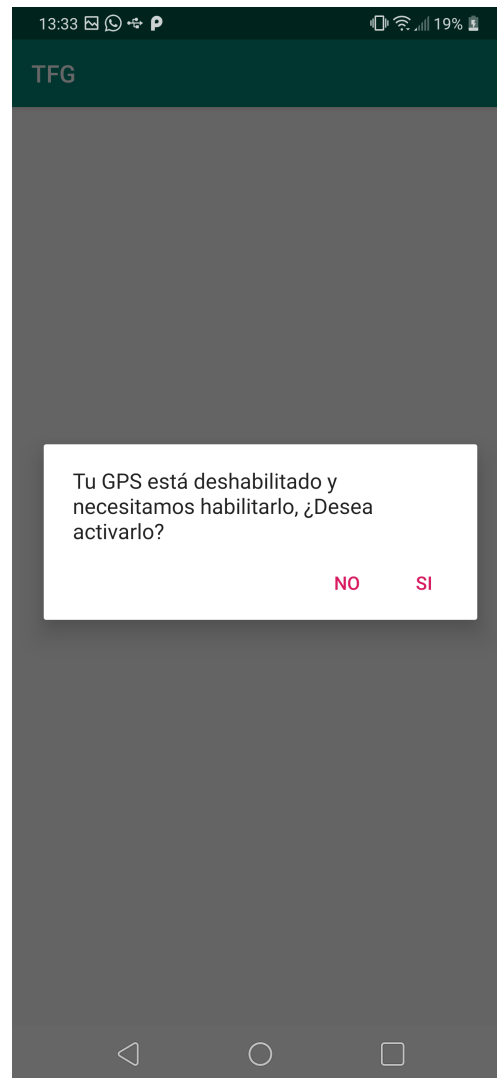


Figura 3.9: Vista habilitar GPS

Búsqueda de dispositivos

Una vez concedidos los permisos y activada la localización, la aplicación empieza a buscar dispositivos BLE cercanos, y tras un periodo de tiempo nos mostrará en forma de lista los distintos dispositivos encontrados, siendo presentados en dos tonalidades, verdes aquellos dispositivos que están emparejados al dispositivo y que la aplicación puede usar, y en rojo aquellos que no se pueden vincular a la aplicación. Para vincular el dispositivo a la app, basta con pulsar sobre en un dispositivo presentado en verde, en caso de pulsar sobre uno de color rojo, nos mostrará un mensaje de error.

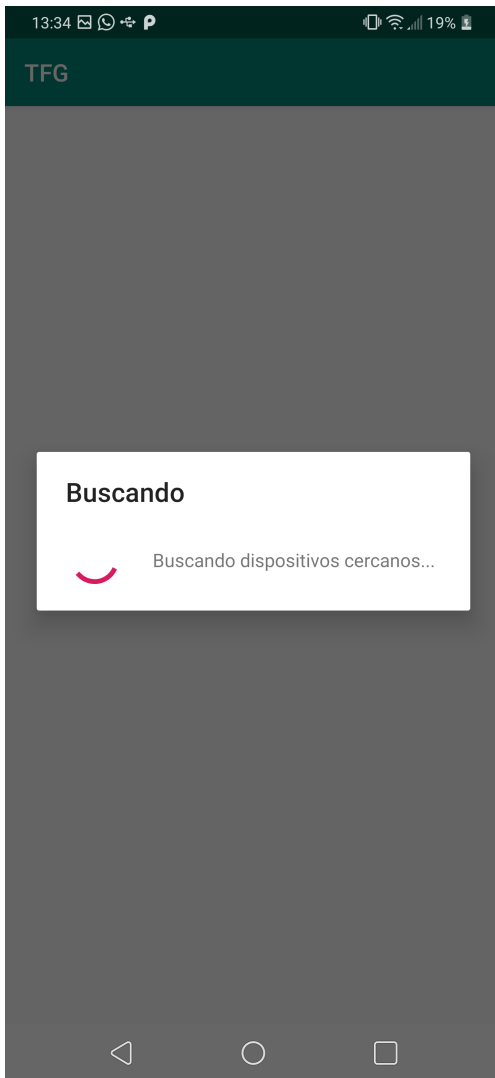


Figura 3.10: *Vista búsqueda*

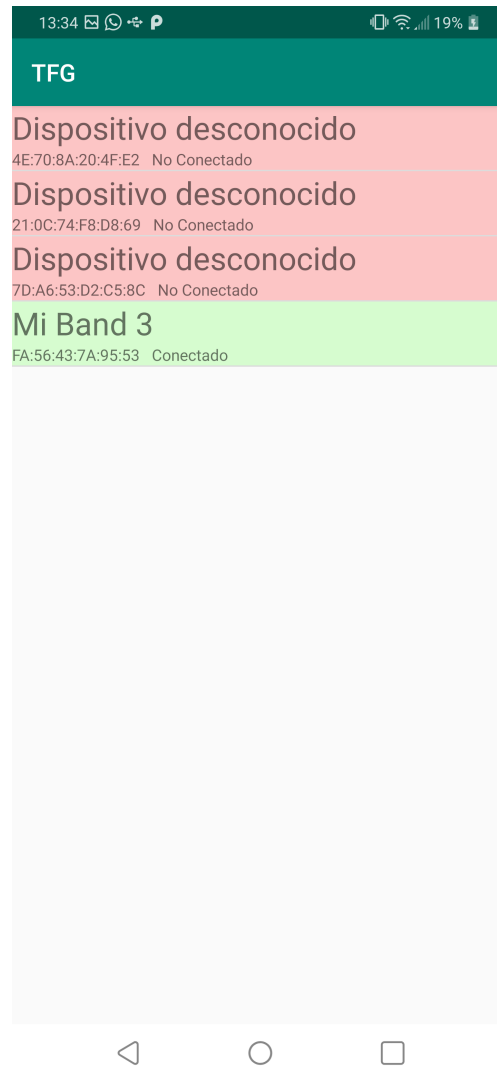


Figura 3.11: *Vista resultados*

Vista principal

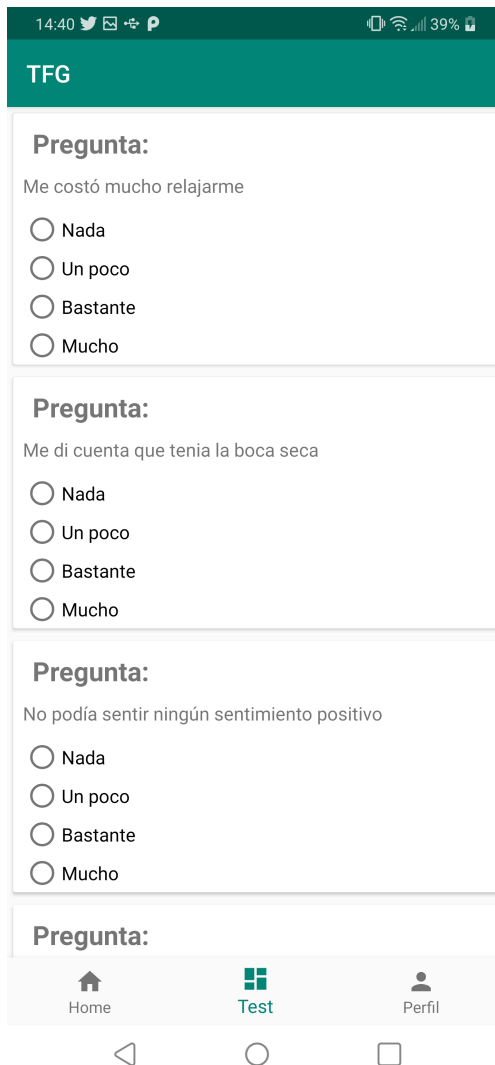
Tras asociar un dispositivo bluetooth, la aplicación nos redirige a la vista principal de la aplicación, esta vista también será la que el usuario verá la próxima vez que inicie la aplicación. Desde esta vista podemos movernos por las diferentes funcionalidades de la aplicación usando el menú situado en la parte inferior.



Figura 3.12: *vista principal*

Vista tests

En la vista de test, aparecerán los diferentes tests que el usuario disponga para contestar a lo largo del día, una vez completado el test para guardar los resultados el usuario debe utilizar el boton enviar situado en el final del test. Una vez enviado y en el caso de que haya otros tests disponibles, la vista cargará un nuevo test, pudiendo contestarse en ese momento o cuando el usuario disponga de tiempo.



14:40 [social icons] [signal icons] 39%

TFG

Pregunta:
Me costó mucho relajarme

Nada
 Un poco
 Bastante
 Mucho

Pregunta:
Me di cuenta que tenia la boca seca

Nada
 Un poco
 Bastante
 Mucho

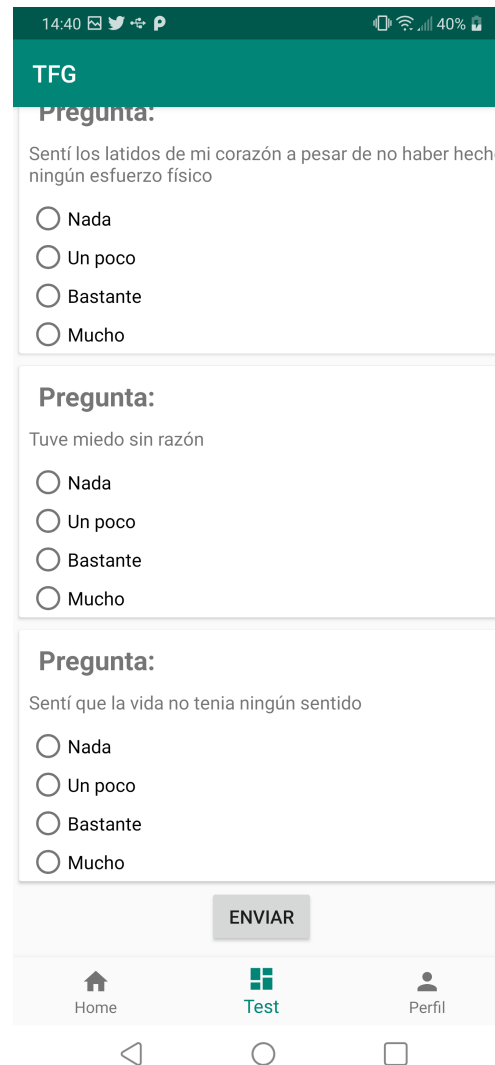
Pregunta:
No podía sentir ningún sentimiento positivo

Nada
 Un poco
 Bastante
 Mucho

Pregunta:

Home Test Perfil

Figura 3.13: Vista test



14:40 [social icons] [signal icons] 40%

TFG

Pregunta:
Sentí los latidos de mi corazón a pesar de no haber hecho ningún esfuerzo físico

Nada
 Un poco
 Bastante
 Mucho

Pregunta:
Tuve miedo sin razón

Nada
 Un poco
 Bastante
 Mucho

Pregunta:
Sentí que la vida no tenia ningún sentido

Nada
 Un poco
 Bastante
 Mucho

ENVIAR

Home Test Perfil

Figura 3.14: Vista test

Vista perfil

En el perfil, el usuario puede visualizar todo lo relacionado con sus datos, desde su información personal, información del dispositivo vinculado y el historial de mediciones realizadas a través de gráficos. Es en esta vista donde el usuario puede cerrar su sesión, llevándolo directamente a la vista de login o cambiar de dispositivo bluetooth.



Figura 3.15: Vista perfil



Figura 3.16: Vista perfil

3.3. Api RESTful

La api responde a una arquitectura REST, dicha arquitectura se basa en un sistema sin estado, es decir, entre dos peticiones independientes, el servidor pierde los datos generados en cada petición; por ejemplo, al loguearse en el sistema, en la siguiente petición el servicio no recordará al usuario logueado, por ello se debe pasarle toda la información necesaria para que procese cada petición. La arquitectura REST se apoya en los verbos HTTP (GET, POST, PUT, DELETE, etc), GET obtendrá un recurso del sistema, POST creará un recurso, PUT actualiza un recurso y DELETE elimina un recurso. Entre las principales ventajas...

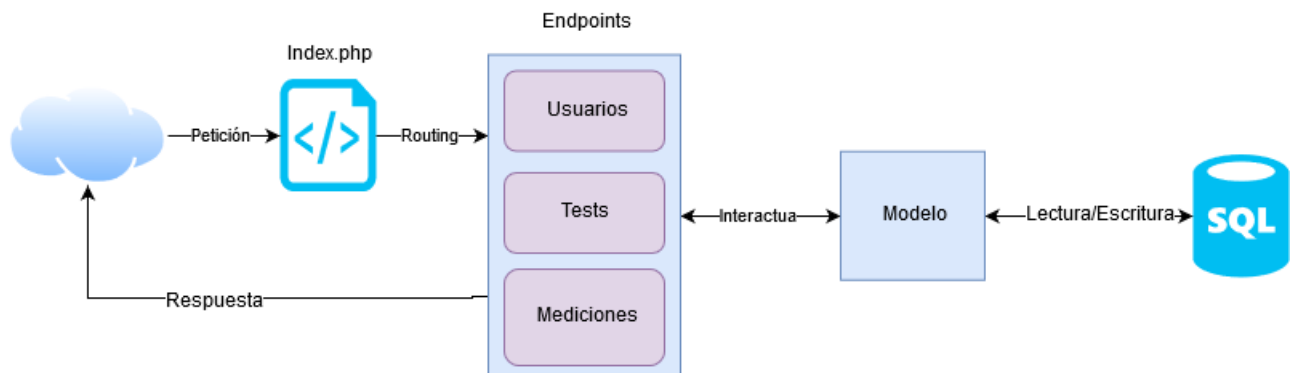


Figura 3.17: *Arquitectura API*

Por defecto PHP proporciona el siguiente formato de URLs `http://hostname?variable=valor`, el cual dista del formato de las APIs REST `http://hostname/recurso/valor`. Para realizar la transformación de las urls es necesario configurar un fichero `.htaccess`, estableciendo en él unas directivas hacer la reescritura de las urls con el fin de obtener el formato deseado. Con la nueva URL, el framework Slim se encarga de hacer el routing correspondiente hacia el endpoint correspondiente.

Los endpoints son callbacks a los que asociamos un método HTTP y la url con la que accederemos al servicio, el framework slim realiza un mapeo interno el cual usará para realizar el enrutamiento de la petición hasta el servicio solicitado. Una de las ventajas de este sistema es que permite agrupar endpoints en función de la estructura de la URL, facilitando mucho la escalabilidad del sistema.

El modelo lo conforman las clases destinadas al tratamiento de datos (usuarios, tests y mediciones) y el conjunto de objetos DTO que nos ayudan a desacoplar la comunicación entre los endpoints y el modelo, incorporando la funcionalidad de serializaciónJSON. Para el acceso a la base de datos, se ha utilizado un patrón singleton con el fin de tener solo una instancia a la base de datos y así evitar tener múltiples conexiones a la base de datos abierta.

3.3.1. Securitización de las conexiones

Dado el carácter de los datos que se manejan en la API, es necesario dotar seguridad al servicio web con la finalidad de proteger quien puede acceder a la información, por ello se ha utilizado el sistema middleware que ofrece el framework slim⁷.

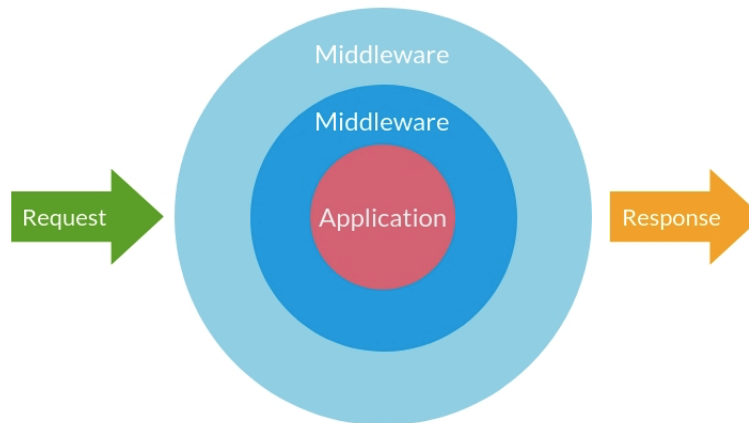


Figura 3.18: *Sistema middleware*

Los middleware son capas que envuelven a la aplicación de forma concéntrica, de tal forma que interceptan tanto la petición antes de entrar al servicio web, como las respuestas a la salida. Aprovechando esta funcionalidad se ha implementado un sistema de autenticación basado en tokens JWT.

Los Tokens no son más que una firma cifrada que permite a nuestra API identificar al usuario cada vez que se realiza una petición, ya que este token contiene información a cerca del usuario propietario.

La estructura de los tokens está formada por tres campos: un encabezado o header, un contenido o payload, y una firma o signature.

- El header contiene generalmente dos campos, el tipo o type, el cual es jwt, y el algoritmo de firma, como por ejemplo HMAC SHA256 or RSA. Una vez formado el header este es cifrado en base64
- El payload contiene la información relacionada con la fecha de creación del token, espiración y datos del usuario, en el proyecto en cuestión se ha incluido los claims iss (Issuer) que identifica al proveedor de identidad que emitió el token, iat (Issued at) el cual es una marca temporal de la fecha en la que se generó el token, sub (Subject) identifica al usuario al cual le fue emitido el token y exp (Expiration time) siendo una marca temporal a partir de la cual el servidor no aceptará peticiones con ese token. Al igual que el header, una vez creado el payload, este es cifrado en base64.

- Por último, el tercer campo es la firma(signature) este campo está formado la cadena de caracteres del header y payload cifrado en base64 separadas por un punto. A esta cadena resultante se le aplica el cifrado elegido en el header, siendo necesario una contraseña para aplicar este cifrado, la cual será utilizada posteriormente para la decodificación del token. En el proyecto se ha utilizado el algoritmo de cifrado HS256 con SHA256

3.3.2. Base de datos

En esta sección veremos en profundidad los distintos elementos que componen nuestra bases de datos.

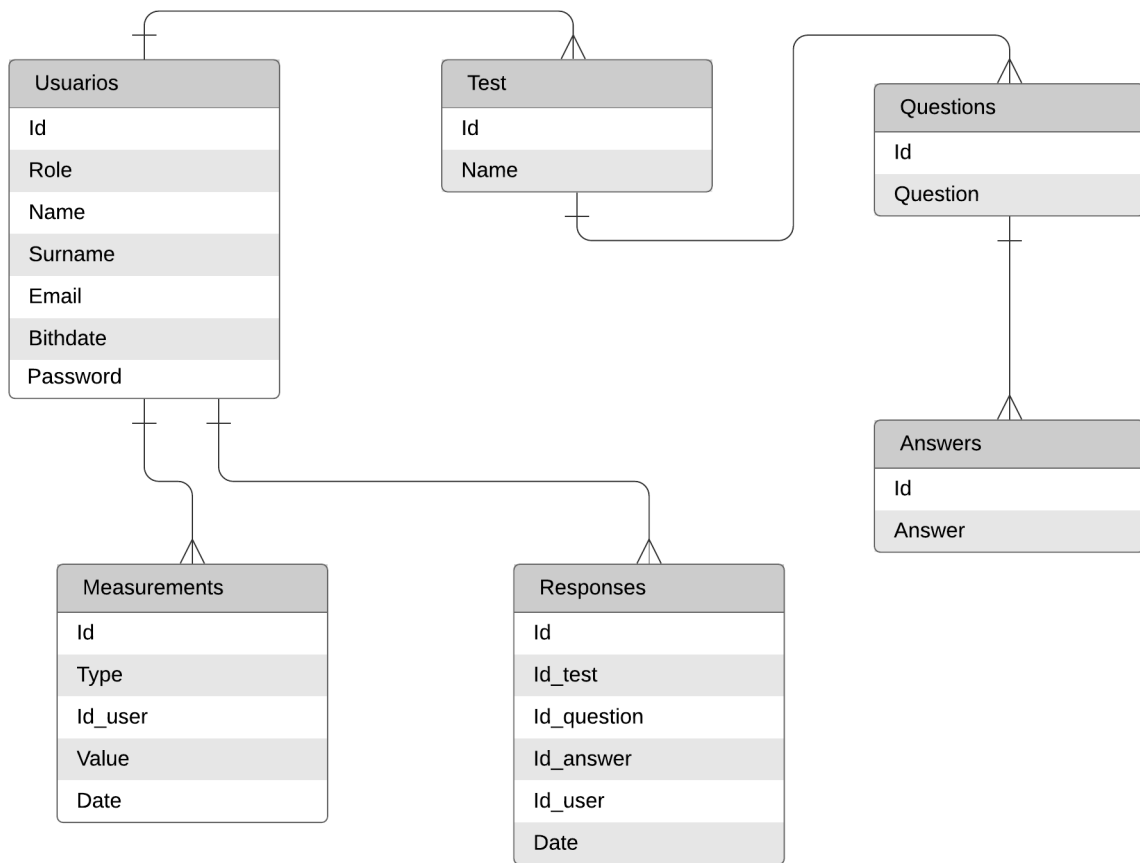


Figura 3.19: *Entidad relación base de datos*

Las tablas necesarias para almacenar la información en el servidor son:

- **users:** En esta tabla se almacena toda la información personal de los usuarios de nuestro sistema.

- id (integer): identificador único de cada usuario del sistema.
 - role (enum): rol que desempeña el usuario en el sistema (usuario, médico o administrador)
 - name (varchar): nombre del usuario.
 - surname (varchar): apellido del usuario.
 - email (varchar): email del usuario, este campo es único para cada usuario, por lo que no pueden existir dos usuarios con el mismo email.
 - birthdate (date): fecha de nacimiento.
 - pass (varchar): hash de la contraseña del usuario.
- **test:** En esta tabla se almacena
 - id (integer): identificador único de cada test.
 - name (varchar): nombre del test.
- **questions:** En esta tabla se almacenan las preguntas de los test
 - id (integer): identificador único de la pregunta
 - question (varchar): texto correspondiente al enunciado de la pregunta.
- **answers:**
 - id (integer): identificador único de una respuesta.
 - answer (varchar): texto correspondiente a la opción de la respuesta.
- **response:** En esta tabla se almacenan todas las posibles opciones de las preguntas.
 - id (integer): id único que identifica a una respuesta de un usuario.
 - id_test (integer): id del test al que corresponde la respuesta
 - id_question (integer): id de la pregunta a la que corresponde la respuesta.
 - id_answer (integer): id de la opción a la que corresponde la respuesta.
 - id_user (integer): id del usuario que ha contestado.
 - date (timestamp): fecha de en la que fue respondida.
- **measures:** En esta tabla se almacenan todos los datos de las mediciones de los dispositivos wearables
 - id (integer): id unico para identificar a una medición realizada.
 - type (enum): tipo de medición realizada (pasos o pulsaciones)
 - id_user (integer): id del usuario al que pertenece la medición
 - value (integer): valor de la medición.
 - date (timestamp): fecha en la que se realizó la medición

3.3.3. Tecnologías empleadas

Durante el desarrollo del proyecto ha sido necesario disponer de una pila LAMP en la que poder probar el código tanto por parte de la API como en la aplicación móvil, por ello se ha utilizado XAMPP para el entorno de desarrollo.

En el desarrollo de la parte del servidor se ha empleado PHP, ya que es un lenguaje de código libre ampliamente utilizado y con una documentación muy detallada, para el almacenamiento de datos se ha utilizado MySQL¹⁰ como gestor de base de datos, dada la integración nativa que dispone con PHP⁵. En cuanto a la transmisión de datos entre el servidor y la aplicación se ha utilizado JSON, ya que es un sistema muy ligero que en nuestro caso cuenta soporte nativo tanto PHP como android.

En cuanto a la parte del desarrollo de la app android, al ser un desarrollo nativo se ha utilizado JAVA como lenguaje, ya que es el lenguaje oficial por parte de Google para el desarrollo de este tipo de aplicaciones y por disponer de una amplia comunidad que aportan librerías de terceros que facilitan el tiempo de desarrollo.

Capítulo 4

Experimentación

La última fase del proyecto a consistido en desplegar el servidor con el fin de poder probar la aplicación durante varios seguidos y así poder detectar posibles fallos que de otra forma no pudieran manifestarse en el entorno de desarrollo.

En primer lugar ha sido necesario configurar una pila LAMP para proceder al despliegue de la API, dado el poco volumen de usuarios que iban a estar conectados al mismo tiempo durante la fase de pruebas, se ha decido montar un servidor en una Raspberry pi 3 con raspbian como SO.

Por otro lado ha sido necesario configurar un cliente para un servicio de DNS dinámico, puesto que el proveedor ISP no proporciona direcciones IPs fijas. Este cliente nos va a permitir poder acceder desde cualquier parte de internet a nuestra api a través de un nombre de dominio, y para ello el cliente enviará al servidor DNS nuestra IP de forma periódica, con el objetivo de que si nuestro proveedor de internet nos cambia la IP este cambio sea notificado.

4.1. Pruebas de rendimiento

4.1.1. Medidas tiempo API

Para comprobar la eficiencia de nuestra api, se ha procedido a realizar unas mediciones del tiempo desde que un cliente envía una petición al servidor hasta que este recibe una respuesta.

El procedimiento de prueba ha consistido en realizar 5 mediciones de tiempo de forma no consecutiva, es decir, nunca se han realizado 5 peticiones a un mismo recurso seguidas para evitar cualquier acceso a datos por caché.

Petición	Tiempo medio
login	61.6ms
Registro	86.2ms
Obtener usuario (id)	42.4ms
Obtener usuario (email)	36.4ms
Obtener usuarios	54.2ms
Actualizar usuario	59.4ms
Tests disponibles para un usuario	40.2ms
Obtener test (id)	59.6ms
Obtener test (nombre)	46.6ms
Guardar respuestas	77.8ms
Obtener frecuencia cardiaca	36.8ms
Obtener pasos	36.2ms
Añadir medición frecuencia cardiaca	43.2ms
Añadir medición pasos	42.4ms

Cuadro 4.1: *test*

4.1.2. Profiling app android

Con el fin de determinar el rendimiento de nuestra aplicación android se ha procedido a realizar un profiling de la aplicación, para obtener unas medidas de uso de CPU, consumo de memoria ram y el consumo energético que tiene el uso de la app en la batería de nuestro dispositivo móvil. Para realizar esta parte del proyecto se ha utilizado la herramienta de profiling integrada en Android Studio.

El procedimiento ha consistido en ejecutar la aplicación 5 veces desde arranque en frío, es decir sin que hubiese nada de ella cargada en la memoria con el fin de obtener los casos más desfavorables de la aplicación, y para determinar los resultados se han utilizado los peores

resultados de las cinco ejecuciones. Las medidas de rendimiento se han realizado sobre un dispositivo LG G7 thinQ con las siguientes características:

- Procesador: Snapdragon 845 de ocho núcleos
- RAM: 4GB
- Sistema operativo: Android 9

Cabe destacar que las medidas de rendimiento pueden variar de un dispositivo a otro, e incluso en un mismo dispositivo con diferente sistema operativo, por lo que la reproducción del procedimiento puede dar lugar a variaciones significativas en cuanto al uso de procesador y tiempos de carga.

Los datos que se mediran serán: tiempo de carga, siendo el tiempo desde que el actividad/fragment es lanzado hasta que se encuentra en el estado resumed; uso maximo de CPU, siendo este el porcentaje de carga generada por la app en el procesador; uso máximo de RAM, en el cual se engloba el uso de Java, Nativa, Gráficos, Pila, Código y otros; el uso de red, el cual nos ofrecerá la tasa máxima de uso de ancho de banda.

Login

tiempo de carga	280ms
Uso max. CPU	22 %
Uso max. RAM	93.5MB
Uso de red	4.6KB/s / 3.6KB/s

Cuadro 4.2: *Medidas rendimiento login*

La ventana del login es una de las que mas tardan en cargar, puesto que es la entrada a nuestra app, y el sistema operativo android, debe solicitar y reservar todos los recursos para poder lanzarla.

Escaneo de dispositivos

tiempo de carga	160ms
Uso max. CPU	8.6 %
Uso max. RAM	133MB
Uso de red	0KB/s / 0KB/s

Cuadro 4.3: *Medidas rendimiento Escaneo BLE*

Las operaciones realizadas sobre el escaneo de dispositivos bluetooth no conlleva apenas uso del procesador, puesto que es el propio sistema operativo el que realiza las tareas sobre el dispositivo bluetooth.

Home

tiempo de carga	171ms
Uso max. CPU	14.9 %
Uso max. RAM	131MB
Uso de red	5,4KB/s /3.4KB/s

Cuadro 4.4: *medidas rendimiento home*

Las operaciones sobre el home no suponen un gran uso de recursos, puesto que realiza un petición al servidor con el fin de obtener los tests necesarios y una lectura sobre el dispositivo bluetooth.

Test

tiempo de carga	251ms
Uso max. CPU	17 %
Uso max. RAM	137 MB
Uso de red	34,2KB/s /4KB/s

Cuadro 4.5: *Medidas rendimiento test*

Las operaciones sobre la vista test es una de las que mas usa la conexión de red, puesto que debe solicitar el cuestionario al servidor web, aún así el uso de datos es muy bajo.

Perfil

tiempo de carga	274ms
Uso max. CPU	18,3 %
Uso max. RAM	142.2MB
Uso de red	9.3KB/s /6.8KB/s

Cuadro 4.6: *Medidas rendimiento perfil*

La vista del perfil es la que mayor uso de memoria ram tiene y del procesador, esto es debido a la necesidad de pintar los gráficos incorporados en él, puesto que debe almacenar en ram todos los datos para mostrarlos.

4.1.3. Prueba con usuarios

Uno de los aspectos más importantes a la hora de desarrollar una aplicación, es pensar que debe ser intuitiva y fácil de utilizar por ello se ha realizado unas pruebas con usuarios para comprobar que son capaces de utilizarla sin indicaciones de terceros y para recibir un feedback de aspectos de la aplicación a mejorar.

La prueba ha consistido en proporcionarles la aplicación junto con una smartband xiaomi mi band 3, para pedirles hacer login en la aplicación, emparejar la mi band, contestar el test disponible, ver su historial y por ultimo cerrar sesión. Durante la prueba del dispositivo se ha monitorizado las pulsaciones que realizaban en la aplicación con el objetivo de detectar problemas de usabilidad, y por ultimo se les ha pedido que rellenasen un formulario USE junto con unas preguntas abiertas.

Formulario

1. Creo que usaría esta aplicación frecuentemente.
2. Encuentro esta aplicación innecesariamente compleja.
3. Creo que la aplicación fue fácil de usar.
4. Creo que necesitaría ayuda de una persona con conocimientos técnicos para usar esta aplicación.
5. Las funciones de esta están bien integradas.
6. Creo que la aplicación es muy inconsistente.
7. Imagino que la mayoría de la gente aprendería a usar esta aplicación de forma muy rápida.
8. Encuentro que la aplicación es muy difícil de usar.
9. Me siento confiado al usar esta aplicación.
10. Necesité aprender muchas cosas antes de ser capaz de usar esta aplicación.

Cuestiones

1. ¿Te sientes cómodo con el menú situado en la parte inferior o prefieres un menú deslizante en el lateral izquierdo?
2. ¿Qué es lo que menos te gusta de la aplicación?
3. ¿Echas en falta alguna funcionalidad?

Jorge

Es una persona de 24 años de edad que cuenta con estudios superiores, esta habituado al uso de tecnología ya que su día a día interactúa con multitud de tecnologías por motivos laborales.

Durante las operaciones que se solicitó al usuario no se apreció ningún comportamiento extraño y no hubo que hacerle ninguna precisión o indicación para realizar las tareas pedidas.

Respuestas formulario

Pregunta	Respuesta (0-5)
1	4
2	1
3	5
4	1
5	4
6	1
7	5
8	1
9	4
10	1

Cuadro 4.7: *Test USE Jorge.*

Respuestas cuestiones

1. Considera que le es indiferente el uso del tipo de menú, al ser un menú pequeño considera que es muy manejable.
2. Menciona los colores y el tamaño de la tipografía de los gráficos, ya que es un color que no se aprecia bien y el tamaño de la letra es un poco pequeño
3. A priori considera que no.

Sergio

Sergio de 43 años de edad, cuenta con estudios medios y generalmente usa el móvil y un ordenador personal en el trabajo.

Durante la prueba con la aplicación se apreció que el usuario pulsó sobre la notificación de la vista home, cuando trataba de llegar a los tests pensando que también sería posible acceder desde la notificación. En el resto de operaciones el usuario no tuvo ningún problema.

Respuestas formulario

Pregunta	Respuesta (0-5)
1	3
2	1
3	4
4	1
5	4
6	2
7	4
8	1
9	4
10	1

Cuadro 4.8: *Test USE Sergio.*

Respuestas cuestiones

1. Comenta que no encuentra molesto ni incomodo el menú en la parte de abajo de la pantalla.
2. Nota que le cuesta leer los números de los gráficos.
3. Considera que podría ser interesante añadir un calendario con las citas médicas y que estas fueran notificadas.

Laura

Laura de 35 años de edad, cuenta con estudios superiores relacionados con la tecnología y esta muy habituada al uso de las tecnologías.

En el transcurso de las operaciones que se le solicitó que hiciera no se apreció ninguna dificultad ni ninguna

Respuestas formulario

Pregunta	Respuesta (0-5)
1	4
2	1
3	5
4	1
5	4
6	1
7	4
8	1
9	5
10	1

Cuadro 4.9: *Test USE Laura.*

Respuestas cuestiones

1. Le parece indiferente, ya que comenta que muchas apps cuentan con ambos sistemas de menús y que ambos le parecen cómodos.
2. Comenta que los colores podrían ser algo mas alegres.
3. Comenta que sería interesante la opción de registrar las horas de sueño.

Juan

Es una persona de 69 años de edad que cuenta con estudios básicos, generalmente solo usa el móvil para hacer llamadas y hablar por whatsapp, por lo que no es un usuario muy activo en lo relacionado con las tecnologías.

Durante la fase de vincular la pulsera al dispositivo fue necesario ayudarlo en la pantalla del sistema para activar la geolocalización, ya que no sabía que había que hacer en dicha pantalla. También pulsó sobre la notificación de los tests disponibles para acceder a los tests.

Respuestas formulario

Pregunta	Respuesta (0-5)
1	3
2	2
3	3
4	3
5	4
6	2
7	4
8	3
9	3
10	2

Cuadro 4.10: *Test USE Jorge.*

Respuestas cuestiones

1. Comenta que prefiere el menú en la parte de abajo o arriba ya que está acostumbrado al uso de la aplicación whatsapp y que prefiere no tener que estar abriendo otras cosas.
2. Comenta que no es capaz de leer los valores de los gráficos
3. Añadir recordatorios para las medicaciones.

Valoración

Con los resultados de las pruebas con usuarios, valoramos que la usabilidad de la aplicación es buena, aunque es necesario retocar los gráficos puesto que presenta ciertas dificultades de lectura, también se hará accesible la vista de los tests al pulsar sobre la notificación de la vista home, ya que varios usuarios han interactuado con esta esperando acceder a dicha vista.

Capítulo 5

Conclusiones y trabajo futuro

5.1. Conclusiones

Una vez finalizado el desarrollo del proyecto es el momento de realizar una valoración sobre el trabajo desarrollado y lo aprendido hasta la fecha.

El objetivo del proyecto era el desarrollo de una aplicación móvil con la que recoger información de pacientes con cáncer, ante esto podemos afirmar que el objetivo ha sido cumplido puesto que la aplicación es capaz de recoger la información de pulseras de actividad (pulso y pasos) y también sobre cuestionarios diarios a los pacientes.

Ante el objetivo de desarrollar un sistema para almacenar la información recolectada por la aplicación ha sido cumplido con el desarrollo de una API restful, a la cual la aplicación android es capaz de conectarse y mandar la información recolectada, así como recuperar la información del usuario.

Por otro lado, las comunicaciones debían ser seguras por los datos que se manejan, este objetivo se ha cumplido mediante el uso de autenticación basada en tokens, siendo este sistema usado para que solo puedan acceder a los datos las personas identificadas en el sistema.

Por último la aplicación debía ser intuitiva de cara a los usuarios, y mediante la prueba de la aplicación en usuarios y los resultados arrojados por los cuestionarios el objetivo ha sido cumplido.

Gran parte del desarrollo de este proyecto ha sido gracias a ciertas asignaturas estudiadas a lo largo de la carrera, como pueden ser ingeniería del software y tecnología de la programación, para el desarrollo de las arquitecturas software, aplicaciones Web y Base de datos para el desarrollo de la parte del servidor, programación de aplicaciones para dispositivos móviles para el desarrollo de la aplicación android y otras asignaturas de forma indirecta.

5.2. Trabajo futuro

En esta sección se describirán posibles ideas que añaden mas valor al proyecto, pero que por complejidad y por tiempo disponible no ha sido posible desarrollarlas.

- Desarrollo de chat en tiempo real, con esto se podría tener una comunicación en tiempo real entre paciente y médico con el fin de poder recibir asesoramiento directo por parte del médico en situaciones que los datos indiquen que el paciente necesita de atención urgente. Para realizarlo se puede optar por la integración del servicio de firebase que ofrece google con la api y la aplicación android.
- Entrada por voz a los cuestionarios, con esta funcionalidad se añade la posibilidad del análisis de emociones por voz, añadiendo una mayor fiabilidad a las respuestas ofrecidas por el paciente detectado la tristeza, alegría, nerviosismo, etc.
- Gran parte de la comunicación de los seres humanos se realiza mediante la comunicación no verbal, por lo que sería de gran interés realizar un análisis de las expresiones faciales del usuario mientras está haciendo usos del dispositivo, gracias a la cámara frontal.

Capítulo 5

Conclusions and future work

5.1. Conclusions

Once the development of the project is finished, it is time to make an assessment about the work developed.

The objective of the project was the development of a mobile application with which to collect information from cancer patients, we can say that the objective has been fulfilled since the application is able to collect the information of activity wristbands (pulse and steps) and also on daily questionnaires to patients.

The objective of developing a system to store the information collected by the application has been fulfilled with the development of a restful API, to which the application android is able to connect and send the collected information, as well as recover the user information.

On the other hand, communications should be secure because of the data that is handled within the system, this objective has been fulfilled through the use of token-based authentication, this system being used so that only the people identified in the system can access the data.

Finally, the application should be intuitive for users, and with the test of the application in users and the results produced by the questionnaires the objective has been fulfilled

Much of the development of this project has been thanks to certain subjects studied throughout the career, such as software engineering and programming technology, for the development of software architectures, Web applications and Database for server part development, application programming for devices phones for the development of the android application and other subjects indirectly.

5.2. Future work

This section will describe possible ideas that add more value to the project, but that due to complexity and available time it has not been possible to develop them.

- Real-time chat development, with this you could have a communication between the patient and the doctor with the possibility of receiving direct advice from of the doctor in situations where the data indicates that the patient needs attention. This can be done with the integration of the google firebase service with the api and the Android application.
- Voice input to the questionnaires, with this functionality can be analyze the emotions, adding greater reliability responses offered by the patient detected sadness, joy, nervousness, etc.
- Much of the communication of human beings is done through non-verbal communication, so it would be interesting to perform an analysis of the user's facial expressions while using the device, thanks to the front camera.

Bibliografía

- [1] A.P. Cole, D.T. Pucheril, M. Sun, S.A. Fletcher, D.L. Berry, P.L. Nguyen, C. Sweeney, M. Menon, A.S. Kibel, and Q-D. Trinh. 639 - a smartphone-based mobile health app to address the adverse effects of androgen deprivation therapy in men with prostate cancer. *European Urology Supplements*, 18(1):e849 – e850, 2019. Abstracts EAU19 – 34th Annual EAU Congress.
- [2] Juan Antonio Cruzado, Ziortza González, and Manuel Hernández. Problemas psicológicos en pacientes con cáncer: Dificultades de detección y derivación al psicooncólogo. *Psicooncología*, 4, 01 2007.
- [3] Instituto Nacional del Cáncer de EE. UU. ¿qué es el cáncer. url <https://www.cancer.gov/espanol/cancer/naturaleza/que-es>.
- [4] Google. Android developer. url <https://developer.android.com/>.
- [5] The PHP Group. Documentación php. url <https://www.php.net/manual/es/index.php>.
- [6] Elena Ibáñez and José Pastor. Intervención psicológica en enfermos de cáncer : planteamientos del presente, deseos de futuro. 01 2008.
- [7] Josh Lockhart, Andrew Smith, Rob Allen, and Pierre Bérubé. Documentación slim. url <http://www.slimframework.com/docs/v2/>.
- [8] Robert C. Martin. *Clean Architecture*. Prentice-Hall, 2017.
- [9] Cosimo De Nunzio, Riccardo Lombardo, Giorgia Tema, Fabiana Cancrini, Giorgio Ivan Russo, Rodrigo Chacon, Eduard Garcia-Cruz, Maria Jose Ribal, Giuseppe Morgia, Antonio Alcaraz, and Andrea Tubaro. Mobile phone apps for the prediction of prostate cancer: External validation of the coral and rotterdam apps. *European Journal of Surgical Oncology*, 45(3):471 – 476, 2019.
- [10] Oracle. Documentación mysql. url <https://dev.mysql.com/doc/>.
- [11] Kevin Townsend and Akiba Carles Cuf and, Robert Davidson. *Getting Started with Bluetooth Low Energy book*. O’Reilly, 2014.
- [12] Emory University. Efectos emocionales y psicológicos del cáncer. url <https://www.cancerquest.org/es/para-los-pacientes/problemas-psicosociales>.

Apéndice A

Manual de usuario de la API

A.1. Login

El login nos permite acceder al sistema con nuestro email y contraseña, al loguearnos la API devuelve un token JWT que será necesario añadir a la cabecera de las peticiones

Método	POST
URL	http://localhost/api/public/auth/login
Parámetros	email y contraseña

A.1.1. Ejemplo de petición

```
curl -X POST -i http://localhost/api/public/auth/login --data  
'{"email":"test","pass":"test"}'
```

A.1.2. Respuestas

Código **200** si el usuario se ha podido loguear correctamente

```
1  {  
2  "user": {  
3    "id": "19",  
4    "role": "user",  
5    "name": "Roberto",  
6    "surname": "Delgado",  
7    "email": "test23",  
8    "birthdate": 1566252000  
9  },  
10 "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.  
    eyJpc3MiOiJURkciLCJpYXQiOiJlNzY2ODc4ODEsInN1YiI6InRlc3QyMyJ9.  
    .LnkK752VwKGue-aORBej7z3RCZ3fZOKPLo3cY2K1S7U"  
11 }
```

Código **401** si el usuario no ha podido loguearse correctamente

```
1 {"error": "Invalid username or password."}
```

A.2. Registro

El registro permite crear un nuevo usuario en el sistema. No será posible crear un usuario cuyo email exista previamente en el sistema. Una vez creado el usuario el sistema nos devolverá la información del usuario registrado junto con su token.

Método	POST
URL	http://localhost/api/public/auth/register
Datos	role, name, surname, email, birthdate y pass

A.2.1. Ejemplo de petición

```
curl -X POST -i http://localhost/api/public/auth/register --data '{
  "role": "user",
  "name": "Roberto",
  "surname": "Delgado",
  "email": "TeST2",
  "birthdate": 1566252000,
  "pass": "test"
}'
```

A.2.2. Respuestas

Código **200** si el usuario se ha podido registrar correctamente

```
1 {
2   "user": {
3     "id": "23",
4     "role": "user",
5     "name": "Roberto",
6     "surname": "Delgado",
7     "email": "TeST2",
8     "birthdate": 1566252000
9   },
10  "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJURkciLCJpYXQiOiJlNzY2OTUwNTgsInN1YiI6IiRlR1U1QyIn0.tC1dr1BADBQdeGEcbSx7GXGqXqa0ur3hk1-_lEnzH4k"
11 }
```

Código **409** si existe un usuario en el sistema con el email suministrado.

```
1 {
2   "error": "The user already exists"
3 }
```

A.3. Obtener usuario por su id

Mediante el uso de esta petición se puede obtener un usuario por su id único.

Método	GET
URL	http://localhost/api/public/api/user/<id>
Parámetros	id: valor entero
Cabecera	Authorization: Bearer <JWT>

A.3.1. Ejemplo de petición

```
curl -X GET -H 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.
eyJpc3MiOiJURkciLCJpYXQiOiJlNzY2OTE0MTksInN1YiI6InRlc3QifQ.ZNy
-1RE4Pcs2DXLWUyi7ypbIAoMsRfljhoB_4mUcJvs' -i http://localhost/
api/public/api/user/1
```

A.3.2. Respuestas

Respuesta con código HTTP **200** junto con los datos del usuario.

```
1 {
2   "id": "1",
3   "role": "admin",
4   "name": "Roberto",
5   "surname": "Delgado",
6   "email": "test",
7   "birthdate": 1566252000
8 }
```

Respuesta con código HTTP **200** junto con el mensaje de error indicando el motivo.

```
1 {"error": "There is no user with this ID."}
```

A.4. Obtener usuario por su email

Esta petición permite obtener la información de un usuario dado por su email.

Método	GET
URL	http://localhost/api/public/api/user/<email>
Parámetros	email: string
Cabecera	Authorization: Bearer <JWT>

A.4.1. Ejemplo de petición

```
curl -X GET -H 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.
eyJpc3MiOiJURkciLCJpYXQiOiE1NzY2OTE0MTksInN1YiI6InRlc3QifQ.ZNy
-1RE4Pcs2DXLWUyi7ypbIAoMsRf1jhoB_4mUcJvs' -i 'http://localhost/
api/public/api/user/test@gmail.com'
```

A.4.2. Respuestas

Respuesta con código **200** con la información del usuario con el email suministrado.

```
1 {
2   "id": "17",
3   "role": "user",
4   "name": "user",
5   "surname": "user",
6   "email": "test@gmail.com",
7   "birthdate": 703116000
8 }
```

Respuesta con código **200** con un mensaje de error indicando el motivo.

```
1 {
2   "error": "There is no user with this email."
3 }
```

A.5. Listar todos los usuarios

Esta petición permite obtener la información de todos los usuarios del sistema.

Método	GET
URL	http://localhost/api/public/api/user/
Cabecera	Authorization: Bearer <JWT>

A.5.1. Ejemplo de petición

```
curl -X GET -H 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.
eyJpc3MiOiJURkciLCJpYXQiOiJlNzY2OTE0MTksInN1YiI6InRlc3QifQ.ZNy
-1RE4Pcs2DXLWUyi7ypbIAoMsRfljhoB_4mUcJvs' -i http://localhost/
api/public/api/user/
```

A.5.2. Respuestas

Respuesta con código **200** con la información de los usuarios

```
1
2 [{
3   "id": "1",
4   "role": "admin",
5   "name": "Roberto",
6   "surname": "Delgado",
7   "email": "test",
8   "birthdate": 1566252000
9 },
10 ...
11 {
12   "id": "23",
13   "role": "user",
14   "name": "Roberto",
15   "surname": "Delgado",
16   "email": "TeST2",
17   "birthdate": 1566252000
18 }]
```

A.6. Actualizar usuario

Esta petición permite actualizar la información almacenada en el sistema.

Método	PUT
URL	http://localhost/api/public/api/user/<id>
Parámetros	id: valor entero
Cabecera	Authorization: Bearer <JWT>

A.6.1. Ejemplo de petición

```
curl -X PUT -H 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.
eyJpc3MiOiJURkciLCJpYXQiOiJlNzY2OTE0MTksInN1YiI6InRlc3QifQ.ZNy
```

```
-1RE4Pcs2DXLWUyi7ypbIAoMsRfljhoB_4mUcJvs' -i http://localhost/
api/public/api/user/23 --data '{
"name": "Fernando",
"surname": "Romero",
"email": "luiro@gmail.com",
"birthdate": 1576694507
}'
```

A.6.2. Respuestas

Respuesta con código **200** si se ha podido modificar el usuario junto con su información actualizada.

```
1 {
2   "id": "23",
3   "role": "user",
4   "name": "Fernando",
5   "surname": "Romero",
6   "email": "luiro@gmail.com",
7   "birthdate": 1576623600
8 }
```

Respuesta con código **200** con un mensaje de error

```
1 {
2   "error": "The user has not been updated."
3 }
```

A.7. Obtener los tests disponibles para un usuario

Petición que devuelve el nombre de los test que el usuario tiene pendientes de contestar ese día

Método	GET
URL	http://localhost/api/public/api/user/<id>/availabletests
Parámetros	id: valor entero
Cabecera	Authorization: Bearer <JWT>

A.7.1. Ejemplo de petición

```
curl -X GET -H 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.
eyJpc3MiOiJURkciLCJpYXQiOiJlNzY2OTE0MTksInN1YiI6InRlc3QifQ.ZNy
-1RE4Pcs2DXLWUyi7ypbIAoMsRfljhoB_4mUcJvs' -i http://localhost/
api/public/api/user/1/availabletests
```

A.7.2. Respuestas

Respuesta con código **200** junto con un listado de los nombres de los tests disponibles.

```
1 {
2   "availableTests": ["dass21", "testRo12"]
3 }
```

A.8. Obtener un test por su id

Esta petición permite obtener todas las preguntas y sus posibles respuestas de un test identificado por su id único en el sistema.

Método	GET
URL	http://localhost/api/public/api/test/<id>
Parámetros	id: valor entero
Cabecera	Authorization: Bearer <JWT>

A.8.1. Ejemplo de petición

```
curl -X GET -H 'Authorization: Bearer
eyJ0eXAIoiJKV1QiLCJhbGciOiJIUzI1NiJ9.
eyJpc3MiOiJURkciLCJpYXQiOiJlNzY3MDU2MDksInN1YiI6InRlc3QifQ.
X6FvgYK1BFuegXhVrIMeV17f4OdRyhnjc6tPatvIYEA' -i http://
localhost/api/public/api/test/1
```

A.8.2. Respuestas

Respuesta con código **200** con todo el contenido del test

```
1 {
2   "name": "dass21",
3   "questions": [{
4     "id": "1",
5     "question": "Me costo mucho relajarme",
6     "options": [{
7       "id": 1,
8       "option": "Nada"
9     }, {
10      "id": 2,
11      "option": "Un poco"
12     }, {
13      "id": 3,
```

```

14     "option": "Bastante"
15   }, {
16     "id": 4,
17     "option": "Mucho"
18   }, {
19     "id": 5,
20     "option": "test"
21   }]
22 },
23
24 ...
25
26 {
27   "id": "21",
28   "question": "Senti que la vida no tenia ningun sentido",
29   "options": [{
30     "id": 1,
31     "option": "Nada"
32   }, {
33     "id": 2,
34     "option": "Un poco"
35   }, {
36     "id": 3,
37     "option": "Bastante"
38   }, {
39     "id": 4,
40     "option": "Mucho"
41   }]
42 }]
43 }

```

Respuesta con código **200** con mensaje de error, por no encontrarse en el sistema un test con ese id.

```

1 {
2   "error": "Invalid id"
3 }
4 }

```

A.9. Obtener un test por su nombre

Esta petición permite obtener todas las preguntas y sus posibles respuestas de un test identificado por su nombre en el sistema.

Método	GET
URL	http://localhost/api/public/api/test/<nombre>
Parámetros	nombre: string
Cabecera	Authorization: Bearer <JWT>

A.9.1. Ejemplo de petición

```
curl -X GET -H 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.
eyJpc3MiOiJURkciLCJpYXQiOiE1NzY3MDU2MDksInN1YiI6InRlc3QifQ.
X6FvgYK1BFuegXhVrIMeVl7f4OdRyhnc6tPatvIYEA' -i http://
localhost/api/public/api/test/dass21
```

A.9.2. Respuestas

Respuesta con código **200** con todo el contenido del test

```
1 {
2   "name": "dass21",
3   "questions": [{
4     "id": "1",
5     "question": "Me costo mucho relajarme",
6     "options": [{
7       "id": 1,
8       "option": "Nada"
9     }, {
10      "id": 2,
11      "option": "Un poco"
12     }, {
13      "id": 3,
14      "option": "Bastante"
15     }, {
16      "id": 4,
17      "option": "Mucho"
18     }, {
19      "id": 5,
20      "option": "test"
21     }
22   ]},
23
24   ...
25
26   {
27     "id": "21",
```

```

28     "question": "Senti que la vida no tenia ningun sentido",
29     "options": [{
30         "id": 1,
31         "option": "Nada"
32     }, {
33         "id": 2,
34         "option": "Un poco"
35     }, {
36         "id": 3,
37         "option": "Bastante"
38     }, {
39         "id": 4,
40         "option": "Mucho"
41     }
42 ]
43 }

```

Respuesta con código **200** con mensaje de error, por no encontrarse en el sistema un test con ese id.

```

1 {
2   "error": "Invalid name"
3 }
4 }

```

A.10. Guardar respuestas de un test

Esta petición permite guardar las respuestas de un test dadas por un usuario.

Método	PUT
URL	http://localhost/api/public/api/test
Data	json con: testName(String), userId(int), timestamp(long) y responses(lista)
Cabecera	Authorization: Bearer <JWT>

A.10.1. Ejemplo de petición

```

curl -X PUT -H 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.
eyJpc3MiOiJURkeiLCJpYXQiOiE1NzY3MDU2MDksInN1YiI6InRlc3QifQ.
X6FvgYKlBFuegXhVrIMeVl7f4OdRyhnjc6tPatvIYEA' -i http://
localhost/api/public/api/test --data '{
  "testName": "dass21",
  "userId": 1,

```

```

    "timestamp": 1577554570,
    "responses": [{
      "questionId": 1,
      "responseId": 3
    }, {
      "questionId": 2,
      "responseId": 7
    }]
  },
}
```

A.10.2. Respuestas

Respuesta con código **200** con mensaje.

```

1 {
2   "message": "responses uploaded correctly"
3 }
```

Respuesta con código **400** con mensaje.

```

1 {
2   "message": "Invalid json content"
3 }
```

A.11. Obtener datos de frecuencia cardíaca de un usuario

Esta petición permite obtener las mediciones de la frecuencia cardíaca de un usuario.

Método	GET
URL	http://localhost/api/public/api/measurement/hearttrate/<id>
Parámetros	id: integer
Cabecera	Authorization: Bearer <JWT>

A.11.1. Ejemplo de petición

```

curl -X GET -H 'Authorization: Bearer
eyJ0eXAIoiJKV1QiLCJhbGciOiJIUzI1NiJ9.
eyJpc3MiOiJURkciLCJpYXQiOiE1NzY3MDU2MDksInN1YiI6InRlc3QifQ.
X6FvgYK1BFuegXhVrIMeV17f4OdRyhnjc6tPatvIYEA' -i http://
localhost/api/public/api/measurement/hearttrate/1
```

A.11.2. Respuestas

Respuesta con código HTTP **200** con un listado de todas las mediciones ordenadas por fecha.

```

1 {
2   "measurements": [{
3     "idUser": "1",
4     "value": "63",
5     "timestamp": 1569582000
6   },
7     ...
8   {
9     "idUser": "1",
10    "value": "140",
11    "timestamp": 1569582360
12  }]
13 }
14 }

```

A.12. Obtener datos de pasos de un usuario

Esta petición permite obtener las mediciones de pasos de un usuario.

Método	GET
URL	http://localhost/api/public/api/measurement/steps/<id>
Parámetros	id: integer
Cabecera	Authorization: Bearer <JWT>

A.12.1. Ejemplo de petición

```

curl -X GET -H 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.
eyJpc3MiOiJURkciLCJpYXQiOiE1NzY3MDU2MDksInN1YiI6InRlc3QifQ.
X6FvgYK1BFuegXhVrIMeV17f4OdRyhnjc6tPatvIYEA' -i http://
localhost/api/public/api/measurement/steps/1

```

A.12.2. Respuestas

Respuesta con código HTTP **200** con un listado de todas las mediciones ordenadas por fecha.

```

1 {
2   "measurements": [{
3     "idUser": "1",
4     "value": "10358",
5     "timestamp": 1569571200
6   },

```

```

7     ...
8     {
9         "idUser": "1",
10        "value": "200",
11        "timestamp": 1570003200
12    }
13 }

```

A.13. Subir medición de frecuencia cardíaca

Petición que permite almacenar una medición de frecuencia cardíaca de un usuario.

Método	POST
URL	http://localhost/api/public/api/measurement/hearttrate
Data	json con: idUser(int), timestamp(long) y value(int)
Cabecera	Authorization: Bearer <JWT>

A.13.1. Ejemplo de petición

```

curl -X POST -H 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.
eyJpc3MiOiJURkeiLCJpYXQiOiE1NzY3MDU2MDksInN1YiI6InRlc3QifQ.
X6FvgYK1BFuegXhVrIMeV17f4OdRyhnjc6tPatvIYEA' -i http://
localhost/api/public/api/measurement/hearttrate --data '{
"idUser": "1",
"timestamp": 1576708574,
"value": "105"
}'

```

A.13.2. Respuestas

Respuesta con código HTTP **200** indicando que la información se ha almacenado correctamente

```

1 {
2   "message": "Data stored correctly"
3 }

```

A.14. Subir medición de pasos

Petición que permite almacenar una medición de pasos de un usuario.

Método	POST
URL	http://localhost/api/public/api/measurement/steps
Data	json con: idUser(int), timestamp(long) y value(int)
Cabecera	Authorization: Bearer <JWT>

A.14.1. Ejemplo de petición

```
curl -X POST -H 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.
eyJpc3MiOiJURkciLCJpYXQiOiE1NzY3MDU2MDksInN1YiI6InRlc3QifQ.
X6FvgYK1BFuegXhVrIMeV17f4OdRyhnc6tPatvIYEA' -i http://
localhost/api/public/api/measurement/steps --data '{
"idUser": "1",
"timestamp": 1576708574,
"value": "10560"
}'
```

A.14.2. Respuestas

Respuesta con código HTTP **200** indicando que la información se ha almacenado correctamente

```
1 {
2   "message": "Data stored correctly"
3 }
```