

# **ESCUELA POLITÉCNICA NACIONAL**

## **ESCUELA DE FORMACIÓN DE TECNÓLOGOS**

### **DESARROLLO DE APLICACIÓN MÓVIL CON DISPOSITIVO OBD2 PARA DIAGNOSTICAR EL ESTADO DE UN VEHÍCULO**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO  
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR  
EN DESARROLLO DE SOFTWARE**

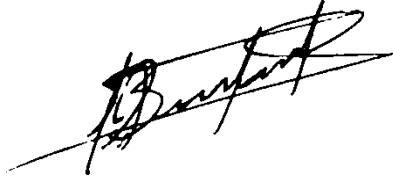
**BYRON SEBASTIAN HUARACA SALAZAR**

**DIRECTOR: ING. JUAN PABLO ZALDUMBIDE PROAÑO, MSC.**

**DMQ, febrero 2021**

## CERTIFICACIONES

Yo, BYRON SEBASTIAN HUARACA SALAZAR declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



---

**Byron Sebastian Huaraca Salazar**

[byron.huaraca@epn.edu.ec](mailto:byron.huaraca@epn.edu.ec)

[byron.huaraca.04@gmail.com](mailto:byron.huaraca.04@gmail.com)

Certifico que el presente trabajo de integración curricular fue desarrollado por BYRON SEBASTIAN HUARACA SALAZAR, bajo mi supervisión.



---

**Ing. Juan Pablo Zaldumbide, MSc.**

**DIRECTOR**

[juan.zaldumbide@epn.edu.ec](mailto:juan.zaldumbide@epn.edu.ec)

## **DECLARACIÓN DE AUTORÍA**

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

BYRON SEBASTIAN HUARACA SALAZAR

# ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
ÍNDICE DE CONTENIDO.....	III
RESUMEN .....	V
ABSTRACT .....	VI
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1 Objetivo general .....	1
1.2 Objetivos específicos .....	1
1.3 Alcance .....	2
1.4 Marco teórico .....	2
Bluetooth.....	2
Wi-fi.....	3
On Board Diagnostics II (OBD2).....	4
Computadora automotriz .....	5
Android Studio .....	5
2 METODOLOGÍA.....	6
2.1 Metodología de Desarrollo .....	6
Roles.....	6
Artefactos.....	7
2.2 Diseño de interfaces (mockups).....	10
Herramienta utilizada para el diseño .....	11
Aplicación Móvil .....	11
2.3 Diseño de la arquitectura .....	11
Arquitectura de la solución.....	11

Aplicación Móvil .....	12
2.4 Herramientas de desarrollo .....	12
Aplicación Móvil .....	13
3 RESULTADOS .....	14
Sprint 0. Configuración del ambiente de desarrollo .....	14
Sprint 1. Importación de datos y Creación de hipervínculos .....	15
Sprint 2. Módulos de estados .....	17
Sprint 3. Módulos de fallas.....	17
Sprint 4. Módulo de almacenamiento .....	18
Sprint 5. Pruebas .....	19
4 CONCLUSIONES .....	26
5 RECOMENDACIONES.....	27
6 REFERENCIAS BIBLIOGRÁFICAS .....	28
7 ANEXOS.....	31

## RESUMEN

El presente proyecto surge como idea de ayuda a los mecánicos de un taller automotor, pues, aunque se posea el conocimiento necesario para realizar el mantenimiento preventivo o correctivo del vehículo, muchas de las veces pueden llegar a pasarse por alto ciertos puntos o fallos mínimos que a futuro puedan ocasionar accidentes automovilísticos.

Teniendo en consideración los accidentes automovilísticos, en varias ocasiones no solo se dan por un mal mantenimiento correctivo o preventivo, sino también, por el descuido del propietario del vehículo. En ciertos casos, dicho descuido no solo puede ocasionar un accidente, también ocasiona un mal para el propietario del vehículo, esto debido a que los vehículos no pasan las revisiones anuales y esto repercute en el pago de una multa.

En este sentido se ha desarrollado una Aplicación Móvil que le permite al mecánico de un taller automotor y al dueño de un vehículo conocer los códigos de falla de un automóvil, así como también, una redirección a un sitio web permitiendo visualizar en mayor detalle el código de falla obtenido.

Este proyecto está desarrollado con el uso de diferentes tecnologías, destacando el IDE de desarrollo de *Android Studio* permitiéndonos construir una aplicación netamente nativa para el usuario final y *firebase* como sistema gestor de Base de Datos (NoSQL). Además, se hace uso de la metodología ágil *Scrum*, misma que aporta beneficios tales como; obtención de requerimientos, flexibilidad entorno al cambio, roles definidos y reuniones con el usuario, esto permite que el desarrollo se concluya en los tiempos establecidos. Para concluir, en la última sección de este informe, se detallan las conclusiones y recomendaciones que se han obtenido a lo largo del desarrollo del presente proyecto.

**PALABRAS CLAVE:** *Android Studio, firebase, NoSQL, Scrum, automóvil.*

## **ABSTRACT**

This project arises as an idea to help the mechanics of an automotive workshop, because, although they have the necessary knowledge to carry out the preventive or corrective maintenance of the vehicle, many times certain points or minor failures that may cause car accidents in the future.

Considering car accidents, on several occasions they are not only due to poor corrective or preventive maintenance, but also to the carelessness of the owner of the vehicle. In certain cases, said carelessness can not only cause an accident, it also causes damage to the owner of the vehicle, this is because the vehicles do not pass the annual inspections and this affects the payment of a fine.

In this sense, a Mobile Application has been developed that allows the mechanic of an automotive workshop and the owner of a vehicle to know the fault codes of a car, as well as a redirection to a website allowing the code to be viewed in greater detail of fault obtained.

This project is developed with the use of different technologies, highlighting the Android Studio development IDE allowing us to build a purely native application for the end user and firebase as a Database management system (NoSQL). In addition, the agile Scrum methodology is used, which provides benefits such as obtaining requirements, flexibility around change, defined roles and meetings with the user, this allows the development to be completed within the established times. To conclude, in the last section of this report, the conclusions and recommendations that have been obtained throughout the development of this project are detailed.

**KEYWORDS:** *Android Studio, firebase, NoSQL, Scrum, car.*

# **1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO**

Un dispositivo de On Board Diagnostics II (OBD2) permite diagnosticar casi por completo el motor de un auto y de la misma forma monitoriza la parte del chasis y otros componentes del vehículo, además, estos dispositivos usan protocolos de comunicación como lo es el protocolo ISO 9141-2, esto porque se debe incluir adaptadores para la velocidad de transferencia de información [1, 2]. Manejar medios digitales para proporcionar un seguimiento adecuado del vehículo; además, obtener una retroalimentación sobre sus componentes beneficia al mecánico de un vehículo [3].

Hoy en día, algunas de las razones por las que los vehículos no aprueban los controles vehiculares es porque no logran pasar el control de emisión de gases y hay fallas en los sistemas de frenos, principalmente por la falta de mantenimiento del vehículo [4, 5], en este sentido, la instalación de un sistema de OBD2 y su uso mediante una aplicación móvil permitirá que se conozca sobre el estado del vehículo.

Se ha desarrollado una aplicación móvil haciendo uso de Android Studio con la finalidad de otorgar al mecánico de un vehículo automotor una lectura detallada del diagnóstico del vehículo, mismo que es obtenido mediante la conexión al dispositivo OBD-II y mediante una aplicación externa se realiza el escaneo del vehículo y se realiza la exportación de un archivo de texto.

El archivo de texto exportado permitirá que la aplicación móvil importe dichos fallos obtenidos a una base de datos. Se realiza la lectura de los fallos almacenados permitiendo hacer una redirección hacia una página web donde se tendrá el detalle del fallo leído esto permitirá que el mecánico de un vehículo no erre sus decisiones al momento de realizar el mantenimiento correctivo o preventivo del automóvil.

## **1.1 Objetivo general**

Desarrollar una aplicación móvil bajo la plataforma Android para el diagnóstico del estado de un vehículo.

## **1.2 Objetivos específicos**

1. OE1: Identificar los requerimientos de la aplicación móvil para la conectividad al dispositivo de OBD2.



2. OE2: Diseñar la arquitectura de datos e interfaz de usuario de la aplicación móvil y del dispositivo OBD2.
3. OE3: Implementar los módulos de interfaz de la aplicación móvil, el esquema y la capa de datos.
4. OE4: Probar la aplicación móvil.

### **1.3 Alcance**

Un dispositivo On Board Diagnostics II (OBD2) permite diagnosticar elementos específicos tales como; el sistema de encendido, motor, aceite y otros componentes de un automóvil. El presente proyecto propone el desarrollo de una aplicación móvil que permita la detección de fallos físicos, como el estado del motor y frenos del vehículo [1].

La propuesta beneficia de forma directa a mecánicos de un taller automotor, debido a que contarán con una herramienta que les permitirá obtener información precisa acerca del fallo físico del motor y frenos de un vehículo.

Se toma en consideración el uso de Android Studio debido a que nos permite el desarrollo de aplicaciones móviles con una interfaz cómoda para el usuario final, a su vez, permite el uso de un dispositivo físico para poder realizar pruebas acerca del funcionamiento de la aplicación móvil [6].

El presente proyecto será desarrollado en base a la metodología ágil Scrum la cual se basa en la creación de ciclos cortos denominados Sprints permitiendo el cumplimiento de los requerimientos obtenidos mediante reuniones previas con los posibles usuarios de la aplicación [7]. Scrum agiliza el proceso, permite el trabajo de forma organizada y es flexible al cambio lo que le permite aceptar posibles cambios que surjan durante el desarrollo del proyecto [7, 8].

El alcance de este proyecto comprende la implementación de una aplicación móvil que haciendo uso de la conexión bluetooth hacia un dispositivo de OBD2 se obtengan especificaciones que permitan la detección de fallos físicos.

### **1.4 Marco teórico**

#### **Bluetooth**

Bluetooth es una red inalámbrica personal, usada para realizar la conexión entre dispositivos sin la necesidad de hacer uso de un cable que permita la transmisión de información. Una gran diferencia con otros dispositivos que permiten la transmisión de información es que los dispositivos que utilizan Bluetooth no hacen uso de la tecnología IrDA, es decir, no usa una tecnología de conexión infrarrojo. Estos dispositivos no hacen uso de una línea recta para poder comunicarse, por lo que la transmisión de información es más flexible, permitiendo realizar comunicaciones hasta cierta distancia, por ejemplo; de una habitación a otra.

La tecnología Bluetooth puede realizar la transferencia de información con velocidades de 1 Mbps, teniendo en cuenta esto, se tendría un alcance de hasta 10 metros, sin embargo, los valores de alcance pueden variar dependiendo la clase de transmisor que se utilice, teniendo en cuenta esto se puede llegar hasta tener un alcance de hasta 100 metros si se utiliza un transmisor de clase 1 [9].

Un tema para tener en cuenta es que la tecnología Bluetooth hace uso de un estándar:

- IEEE 802.15.1 define Bluetooth 1.x, esto determina que la velocidad que puede alcanzar es de 1 Mbps, a su vez, esta presenta una red de área personal (WPAN). Este estándar maneja dos enlaces físicos SCO y ACL cada uno con 12 paquetes distintos. Se divide cada paquete de enlace haciendo uso de TYPE cada uno con 4-bits; el primero posee 4 fajas de control, el segundo para fajas de un solo slot, el tercero para fajas de tres slots y el cuarto para fajas de 5 slots de tiempo [10].

## **Wi-fi**

Es un sistema que permite la conexión y comunicación mediante un punto de acceso de red inalámbrica, el cual posee un alcance limitado, mismo que puede ser mayor en espacios exteriores a interiores [11]. El término nace por la idea de establecer un método para conectar varios dispositivos a una red común manejando el estándar IEEE 802.11 el cual es un estándar de red inalámbrica.

Un punto de acceso de red inalámbrica permite la conexión a una de red inalámbrica, este punto de acceso usa el ancho de banda del router y expande dicha señal para que más dispositivos puedan conectarse a esta red [12], cabe resaltar que el ancho de banda depende de la que sea brindada al router, puede ser más potente o débil de ser el caso.

## **On Board Diagnostics II (OBD2)**

Este protocolo es un sucesor del OBD y OBDI detecta fallos físicos, eléctricos y químicos en el vehículo. Un dispositivo OBD2 almacena el detalle de la falla que se dio y la almacena para poder hacer revisión de en qué condiciones ocurrió el fallo, generalmente el puerto de conexión para hacer uso de este dispositivo se encuentra localizado en el lado del conductor, específicamente bajo el volante. En ciertos casos puede encontrarse localizado en el asiento del copiloto [13]. Los estándares de OBD son aplicables en automóviles que hagan uso de combustible o diésel, a su vez, estos dispositivos pueden hacer uso de bluetooth, Wifi o un cable USB de esta forma se nos permite hacer uso de este mediante una aplicación web o móvil o haciendo la instalación de una aplicación para el computador. Existen varios tipos de OBD en el mercado, tales como:

- Scanner BlueDriver Professional Bluetooth OBDII.
- Scanner Automotriz
- Escáner ELM327.

### **Escáner ELM327 (Bluetooth)**

Este escáner permite hacer la obtención de datos del vehículo mediante la conexión al puerto OBD2 y conectándose con la Unidad de Control Electrónico (ECU) del vehículo enviando los datos obtenidos mediante la conexión bluetooth de un teléfono o computador.

Se debe tener en cuenta que la información que se obtenga del dispositivo OBD2 varía en función del tipo de vehículo al cual se esté realizando el escaneo, esto es debido al tipo de bus de OBD que tenga el automóvil [14]. Este escáner maneja varios protocolos, tales como:

- ISO 9141-2 (5 baud init, 10.4 kbaud)
- ISO 14230-4 KWP (fast init, 10.4 kbaud)
- ISO 15765-4 CAN-BUS

Destacando ISO 9141-2 el cual es utilizado en vehículos europeos o asiáticos y el protocolo ISO 15765-4 CAN-BUS del cual hacen uso la mayoría de los vehículos después del año 2008, cabe resaltar que este proporciona un conjunto limitado de parámetros del automóvil.

### **Escáner ELM327 (Wifi)**

Funciona de la misma forma que el escáner mencionado anteriormente, con gran diferencia que este trabaja mediante una conexión wifi, a diferencia que el escáner que trabaja por bluetooth, este permite hacer un análisis de valores de oxígeno y los sensores en el motor.

Gracias al método de conexión que posee, este permite hacer la lectura y corrección de fallos registrados en la ECU del vehículo y borra la luz de advertencia “revisar motor” del tablero de instrumentos. Cabe resaltar que una de las grandes ventajas que posee este dispositivo, es que posee la lectura de datos en tiempo real [13].

### **Computadora automotriz**

Muchas de las funciones que realiza el vehículo se encuentran controladas por la Unidad de Control del Motor (ECU) del vehículo, este procesa la información o una serie de tareas que pueda ejecutar el vehículo, tales como; monitorear la luz de encendido, regular el inyector de combustible, etc. Este proceso garantiza el funcionamiento óptimo del motor.

La ECU del vehículo permite obtener datos sobre un diagnóstico avanzado, y, a su vez reduce la cantidad de cables necesarios en un automóvil, gracias al diagnóstico avanzado del vehículo mediante la ECU de este se almacenan los códigos de falla (DTC) que permiten identificar el tipo de problema que se obtuvo, de esta forma dar solución al mismo [15].

### **Android Studio**

Es creado por Google y basado en IntelliJ, posee una estructura simple e intuitiva permitiendo el desarrollo en tiempo real, de igual forma, ofrece plantillas para el desarrollo de diferentes elementos que se desee. Aunque hace uso del lenguaje *Java* o *Kotlin* permite la compilación de aplicaciones dejando un archivo *.apk* que puede ser ejecutado en un emulador exterior como *Bluestacks* [16].

Esta herramienta usa un sistema de compilación flexible haciendo uso de *Gradles*, además, posee un emulador cargado de funcionalidades y un entorno que permite la conexión de un dispositivo físico para compilación de aplicaciones [6].

## 2 METODOLOGÍA

Las metodologías ágiles permiten que se adapte la forma de trabajar a las necesidades del proyecto, existe una rápida adaptación en la documentación solicitada, a su vez, es flexible al cambio, es decir, Scrum acepta posibles cambios que puedan surgir durante el desarrollo de software, esto con el fin de adaptarse a las necesidades y requerimientos del cliente [8, 7].

La metodología ágil Scrum se basa en la creación de ciclos cortos denominados Sprints [8]. Al inicio de cada Sprint se realizará una pequeña reunión con los usuarios de la aplicación (conductores de vehículos con un motor de combustión) para definir las actividades a realizarse durante la semana y al final del sprint se presentará un pequeño avance al cliente con el fin de satisfacer dudas con respecto al desarrollo del proyecto y alcanzar los objetivos planteados en el tiempo determinado [17].

### 2.1 Metodología de Desarrollo

Scrum facilita la comunicación con el *Product Owner*, a su vez, facilita la obtención de información, codificación, desarrollo de prototipos, realización de prueba y despliegue en entornos de trabajo [18], de igual forma Scrum está diseñada para proyectos a realizarse en entornos complejos donde se necesite obtener resultados inmediatos de las entregas totales o parciales del producto final [19].

#### Roles

Scrum está basado en roles, *Product Owner*, *Scrum Master* y *Development Team*, donde la participación de cada uno es explícita, debido a que el desarrollo del proyecto depende de la buena participación, la buena planificación y desarrollo de actividades asignadas a cada integrante [18].

**Product Owner:** Es el encargado de decidir cuál será la actividad para realizarse, además, define las necesidades que se solventarán en el proyecto y garantiza una comunicación clara sobre el proyecto [20]. El rol lo ejecuta un conductor de un vehículo con motor de combustión, quien proporciona información clara y precisa mediante una serie de reuniones, de esta manera se identifica los requerimientos para el desarrollo de la aplicación móvil.

**Scrum Master:** Facilita la interacción del equipo de desarrollo como motivante, además, adopta una actitud de liderazgo y garantiza el cumplimiento de los principales aspectos de la metodología Scrum [21]. El rol lo ejecuta el director del proyecto, mismo que posee los

conocimientos necesarios para ayudar al equipo de desarrollo a resolver dudas existentes durante el desarrollo del proyecto.

**Development Team:** Es el grupo de personas aptas y con los conocimientos necesarios para el desarrollo de los requerimientos y las historias de usuario establecidas o pendientes en cada Sprint [20]. El rol lo ejecuta el desarrollador del proyecto, mismo que cumple con los conocimientos para dar desarrollo a cada una de las fases de la metodología Scrum.

El equipo Scrum se encuentra dividido como se presenta en la **TABLA I**.

**TABLA I: Asignación de roles**

<b>NOMBRE</b>	<b>ROL</b>
Sr. Eduardo León	<i>Product Owner</i>
Ing. Juan Pablo Zaldumbide, Msc.	<i>Scrum Master</i>
Sr. Byron Huaraca	<i>Development Team</i>

### **Artefactos**

Garantizan la claridad de la información evitando el mal manejo y comunicación de esta en la entrega del proyecto [22].

### **Recopilación de Requerimientos**

Permite la documentación de los requerimientos de la aplicación móvil basado en el cumplimiento de las necesidades de los interesados [23]. El cumplimiento de esta actividad va de la mano con una serie de reuniones efectuadas con el *Product Owner*, de esta forma se obtiene las funcionalidades a cumplirse de la aplicación móvil. Esta lista se puede observar en la sección Recopilación de Requerimientos (Pág. 1) del Anexo II.

### **Historias de Usuario**

Su escritura es a través de fichas en un formato específico, son descripciones simples que detallan una característica o funcionalidad dicha desde la perspectiva del usuario o cliente [24]. Desde la **TABLA II** hasta la **TABLA VI** se listan las Historia de Usuario, de esta forma se logra la organización de los requerimientos obtenidos para previamente su codificación y resolución.

**TABLA II: Historia de usuario 1 – Creación de hipervínculos**

<b>HISTORIA DE USUARIO</b>	
<b>Identificador (ID):</b> HU001	<b>Usuario:</b> Mecánico
<b>Nombre Historia:</b> Creación de hipervínculos	
<b>Prioridad en Negocio:</b> Media	<b>Riesgo en Desarrollo:</b> Media
<b>Iteración Asignada:</b> 1	
<b>Responsable:</b> Byron Huaraca	
<b>Descripción:</b> El usuario Mecánico de un vehículo está en la posibilidad de visualizar una página web para conocer en mayor medida el tipo de fallo de DTC.	
<b>Observación:</b> Si el usuario mecánico de un vehículo desea hacer visualización de una página web con el fallo de DTC más especificado. Al realizar la lectura de los fallos obtenidos, el mecánico de un vehículo deberá pulsar el código de fallo, de esta forma se hará redirección hacia una página web donde se detallará en mayor medida el fallo de DTC.	

**TABLA III: Historia de usuario 2 - Visualización de estadísticas**

<b>HISTORIA DE USUARIO</b>	
<b>Identificador (ID):</b> HU002	<b>Usuario:</b> Mecánico
<b>Nombre Historia:</b> Recuento de datos	
<b>Prioridad en Negocio:</b> Alta	<b>Riesgo en Desarrollo:</b> media
<b>Iteración Asignada:</b> 2	
<b>Responsable:</b> Byron Huaraca	
<b>Descripción:</b> El usuario mecánico de un vehículo está en la posibilidad de visualizar el recuento de datos del automóvil.	
<b>Observación:</b> Si el usuario mecánico de un vehículo desea visualizar un resumen de los fallos mayoritarios del automóvil este deberá presionar en un módulo denominado “estadísticas” donde se podrá visualizar un resumen de cuáles son los fallos mayoritarios obtenidos al realizar el escaneo del vehículo.	

**TABLA IV: Historia de usuario 3 - Obtención de fallos**

<b>HISTORIA DE USUARIO</b>	
<b>Identificador (ID):</b> HU003	<b>Usuario:</b> Mecánico
<b>Nombre Historia:</b> Obtención de fallos	
<b>Prioridad en Negocio:</b> Alta	<b>Riesgo en Desarrollo:</b> Alta
<b>Iteración Asignada:</b> 3	
<b>Responsable:</b> Byron Huaraca	
<b>Descripción:</b> El usuario mecánico de un vehículo está en la posibilidad de visualizar en detalle el fallo obtenido a través de la aplicación móvil.	
<b>Observación:</b> Si el usuario mecánico de un vehículo desea saber en detalle el tipo de fallo, este encontrará un módulo denominado “fallos” donde se visualizarán los detalles obtenidos, ya sean, fallos en el motor del vehículo o los frenos de este.	

**TABLA V: Historia de usuario 4 - Fallos guardados**

<b>HISTORIA DE USUARIO</b>	
<b>Identificador (ID):</b> HU004	<b>Usuario:</b> Mecánico
<b>Nombre Historia:</b> Fallos guardados	
<b>Prioridad en Negocio:</b> Media	<b>Riesgo en Desarrollo:</b> Media
<b>Iteración Asignada:</b> 4	
<b>Responsable:</b> Byron Huaraca	
<b>Descripción:</b> El usuario mecánico de un vehículo está en la posibilidad de ver los datos guardados de análisis previos.	
<b>Observación:</b> Si el usuario mecánico de un vehículo desea consultar los detalles de fallos previos, podrá encontrarlos en un módulo denominado “Datos congelados” donde se visualizarán los fallos obtenidos de escaneos previos.	



**TABLA VI: Historia de usuario 5 – Importación de datos**

<b>HISTORIA DE USUARIO</b>	
<b>Identificador (ID):</b> HU005	<b>Usuario:</b> Mecánico
<b>Nombre Historia:</b> Importación de datos	
<b>Prioridad en Negocio:</b> Alta	<b>Riesgo en Desarrollo:</b> Media
<b>Iteración Asignada:</b> 1	
<b>Responsable:</b> Byron Huaraca	
<b>Descripción:</b> El usuario mecánico de un vehículo está en la posibilidad de realizar la importación de datos hacia la base de datos desde la aplicación móvil.	
<b>Observación:</b> Si el usuario mecánico de un vehículo desea hacer la carga de datos hacia la base de datos desde la aplicación móvil, deberá existir un botón dentro de la aplicación que permita importar los datos del archivo de texto obtenido de la aplicación “Car Scanner”.	

### ***Product Backlog***

Es una lista que contienen todos los requerimientos a implementar en el producto, se encuentra ordenado dando prioridad a los temas que brindan más valor en el momento de especificar los requerimientos [20]. El *Product Backlog* puede ser observado en la respectiva sección (Pág. 1) del Anexo II.

### ***Sprint Backlog***

Es una lista de tareas emplea para llevar a cabo las historias de usuario de cada *Sprint* [8]. Conforme a la recopilación de requerimientos, historias de usuario y *Product Backlog* se ha establecido 4 *Sprints* de desarrollo, mismos que pueden ser observados en la respectiva sección (Pág. 2) del Anexo II.

## **2.2 Diseño de interfaces (mockups)**

Una vez determinadas las necesidades para este proyecto y establecido los *Sprints*, a continuación, se define la herramienta que se ha empleado en el diseño de la Aplicación Móvil.

## Herramienta utilizada para el diseño

El diseño de las interfaces para la Aplicación Móvil está determinado por prototipos (mockups), mismos que se han realizado con la ayuda de la herramienta *Figma*. Esta herramienta que permite diseñar prototipos mediante el uso de componentes simples [25], en este caso se ha hecho uso de los componentes de diseño orientados a las aplicaciones móviles.

## Aplicación Móvil



**Figura 1: Pantalla de inicio**

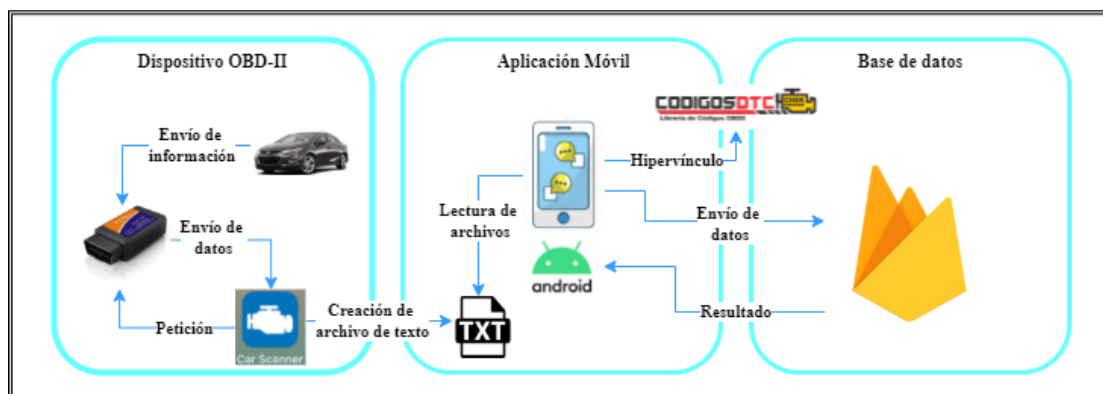
## 2.3 Diseño de la arquitectura

Una vez definidos los *Sprints* para el proyecto, y, usando el diseño de las interfaces, se establece el patrón arquitectónico implementado en el desarrollo de la Aplicación Móvil propuesta.

### Arquitectura de la solución

La arquitectura de la solución de la aplicación móvil está basada en una única colección simple, misma que se usa para poder llevar un registro sencillo de las transacciones o datos, y de esta forma mantener una estructura organizada.

En la **Figura 2** se observa la arquitectura de la solución que se ha empleado en el desarrollo de la aplicación móvil permitiéndonos tener un producto escalable a medida que lo requiera.



**Figura 2: Arquitectura de la Solución – Aplicación móvil**

### **Aplicación Móvil**

Como se observa en la **Figura 2**, la arquitectura de la solución presenta como se realiza la interacción con la aplicación móvil, es decir; en primer lugar, se tiene al dispositivo OBDII que se encuentra conectado al socket OBDII del vehículo, de esta forma, se puede realizar la conexión con la ECU del vehículo para obtener información directamente del cerebro del automóvil. En segundo lugar, una vez conectado el dispositivo al vehículo, este dispositivo realizará el envío de información a través de una petición realizada por una aplicación móvil exterior llamada "Car Scanner", una vez finalizado el escaneo del vehículo se exportan los datos y se genera un archivo de texto para ser leído por nuestra aplicación móvil, se realiza la lectura de este archivo de texto y se cargan los datos a la base de datos firebase. En el módulo denominado "fallos" se obtendrá una pequeña descripción que servirá de ayuda al usuario para poder tener en consideración cada código de DTC obtenido, seguido a la página informativa se visualizará los fallos escaneados y al hacer "clic" en el fallo obtendremos un hipervínculo que nos redirigirá a una página web que permitirá conocer al usuario el detalle de dicho fallo.

Para finalizar, desde la aplicación móvil se realiza el envío de datos obtenidos gracias al archivo de texto exportado, de esta forma, dichos datos se almacenan para ser visualizados más adelante en un módulo denominado "Fallos Guardados". La base de datos enviará el resultado de la información y esta se presenta en el módulo nombrado anteriormente.

## **2.4 Herramientas de desarrollo**

El establecimiento de herramientas de desarrollo que sean directamente proporcionales con el proyecto facilita el desarrollo y optimiza el rendimiento de esta [26].

Una vez definidos los *Sprints* para el proyecto, y, usando el diseño de las interfaces, se describe cada una de las herramientas que se han utilizado en el presente proyecto.

### **Aplicación Móvil**

**TABLA VII: Herramientas para el desarrollo de la Aplicación Móvil**

<b>Herramienta</b>	<b>Justificación</b>
<b><i>Firebase</i></b>	Permite el desarrollo y creación de aplicaciones de forma rápida, es decir, optimiza el tiempo de desarrollo debido a que posee funciones que permitan detectar de errores y realizar pruebas [27].
<b><i>Cloud Firestore</i></b>	Facilita la gestión de información, las actualizaciones en tiempo real las realiza de la misma forma como si se estuviera haciendo uso de <i>Realtime Database</i> , sin embargo, esta herramienta usa un modelo de datos más intuitivo [28].
<b><i>Android Studio</i></b>	Esta herramienta usa un sistema de compilación flexible haciendo uso de <i>Gradles</i> , además, posee un emulador cargado de funcionalidades y un entorno que permite la conexión de un dispositivo físico para compilación de aplicaciones [6].

### 3 RESULTADOS

A continuación, se describe las tareas y los resultados conseguidos en cada Sprint de desarrollo. Además, se detalla las pruebas realizadas a la Aplicación Móvil y su despliegue.

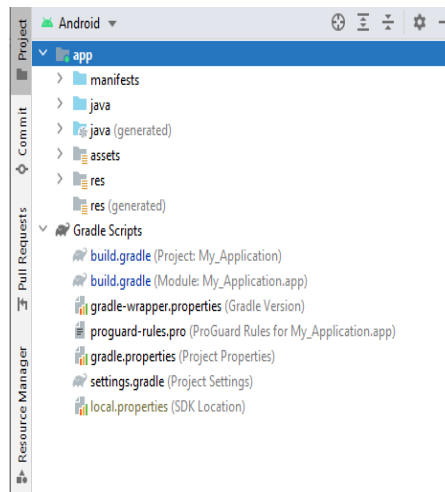
#### **Sprint 0. Configuración del ambiente de desarrollo**

Conforme a la planificación, los resultados del Sprint 0 son:

- Creación del proyecto en Android Studio
- Creación de la base de datos
- Usuarios

#### **Creación del proyecto en Android Studio**

Para crear el entorno de desarrollo se ha instalado las herramientas SDK y el editor de código Android Studio, lo que ha permitido crear directorios bajo el patrón arquitectónico MER. La **Figura 3** muestra la estructura del proyecto.



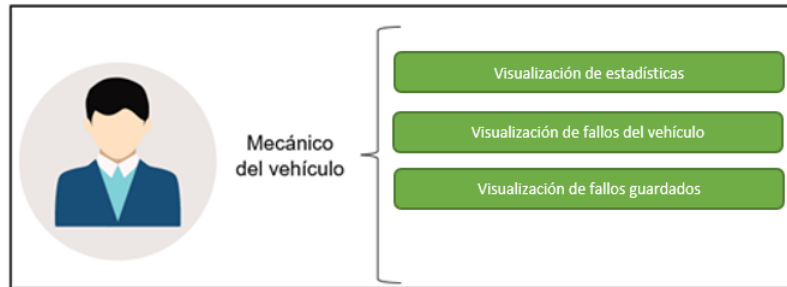
**Figura 3: Estructura de directorios para la aplicación móvil**

#### **Creación de la base de datos**

La base de datos NoSQL se encuentra gestionada mediante *Cloud Firestore* el mismo que es un servicio de la plataforma *Firebase*, en esta se almacena la información relacionada a las solicitudes de los fallos guardados por perfil de vehículo. Las colecciones empleadas y definidas para la Aplicación Móvil pueden ser observadas en la sección “Diseño de la Base de Datos” (Pág. 4) del Anexo II.

## Usuarios

En la **Figura 4** se ilustran los usuarios que pueden interactuar con la Aplicación Móvil, cada uno según el rol que desempeñan con los permisos otorgados a los diferentes módulos asignados una vez realizada la respectiva autenticación.



**Figura 4: Usuarios que interactúan con la aplicación móvil**

### Sprint 1. Importación de datos y Creación de hipervínculos

Conforme a la planificación, los resultados del Sprint 1 son:

- Importación de datos
- Creación de hipervínculos

#### Importación de datos

La **Figura 5** muestra el correspondiente botón de la aplicación móvil, mismo que permite realizar la importación de datos a la base de datos de firebase, una vez se importen los datos se podrá visualizar esos mismos datos en la base, en este caso se ha creado una colección definida para el vehículo escaneado, además se visualiza únicamente los fallos con la descripción del código de DTC. En la **Figura 6** se puede visualizar la colección creada una vez hecha la importación de datos.



**Figura 5: Botón importación de datos**

obd-ii-cdb11	vehículo	Kia
+ Iniciar colección	+ Agregar documento	+ Iniciar colección
vehículo >	Kia >	+ Agregar campo
		P0327: "Knock sensor (KS) 1, bank 1 - low input"
		P0339: "Crankshaft position (CKP) sensor - circuit intermittent"
		P0626: "Alternator field terminal - circuit high"

**Figura 6: Colección con datos almacenados**

## Creación de hipervínculos

La **Figura 7** muestra la implementación de una dirección URL que permite al usuario conocer en mayor medida el tipo de fallo obtenido, la implementación de esta funcionalidad permite tener información detallada del tipo de código de DTC que se haya obtenido, de esta forma se evitan fallos al momento de realizar el mantenimiento preventivo o correctivo del vehículo.

```
String URL = "https://codigosdtc.com/";
String hipervinculo = URL + parent.getItemAtPosition(position).toString();
intent.setData(Uri.parse(hipervinculo));
startActivity(intent);
```

**Figura 7: Código de implementación**



**Figura 8: Hipervínculo resultante**

En este caso la **Figura 8** muestra la redirección, resultado de la implementación del código de la **Figura 7** en este sentido se aprecia el detalle del código P0339.

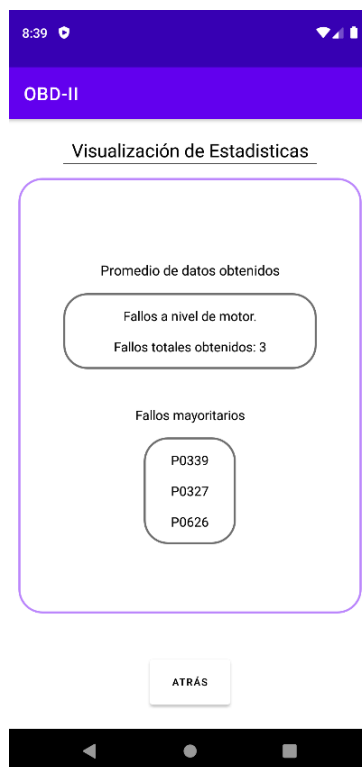
## **Sprint 2. Módulos de estados**

Conforme a la planificación, los resultados del Sprint 2 son:

- Recuento de datos.

### **Implementación del módulo estadísticas**

Si el usuario desea visualizar el recuento de fallos obtenidos del vehículo este lo encontrará haciendo “click” en el módulo estadísticas. Este módulo otorgará información acerca del promedio de datos y fallos mayoritarios encontrados tal y como se muestra en la **Figura 9**.



**Figura 9: Módulo estadísticas**

## **Sprint 3. Módulos de fallas**

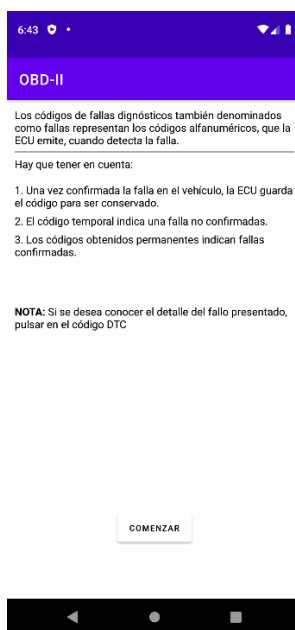
Conforme a la planificación, los resultados del Sprint 3 son:

- Implementación del módulo fallas

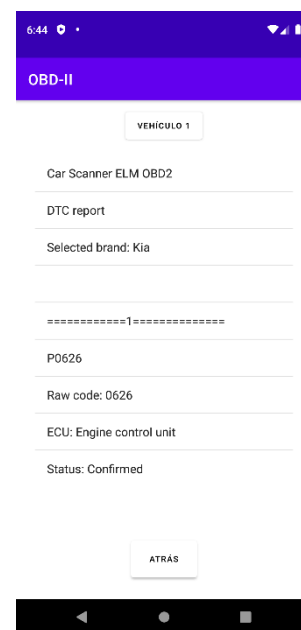


## Implementación del módulo fallas

La **Figura 10** muestra un texto descriptivo, mismo que permite conocer cada uno de los puntos que puede realizar el dispositivo OBD-II. El punto 1 mencionado indica que el dispositivo realiza el almacenamiento del código encontrado, el punto 2 indica fallas halladas, pero no confirmadas y el punto 3 hace alusión a fallas que fueron detectadas y a su vez se encuentran confirmadas por el dispositivo OBD-II. Para avanzar de actividad y realizar el diagnóstico del vehículo realizamos “click” en el botón situado al final de la pantalla. El botón “Comenzar” permitirá a la aplicación móvil realizar la actividad que nos permitirá hacer la obtención de datos del dispositivo OBD-II.



**Figura 10: Texto informativo**



**Figura 11: Fallos de DTC**

La **Figura 11** muestra el diagnóstico realizado mediante el dispositivo, se nos otorga en detalle el tipo de falla en el vehículo. Cabe recalcar que se hizo la colocación de un botón “Atrás” mismo que nos llevará nuevamente a la pantalla de inicio de la aplicación móvil.

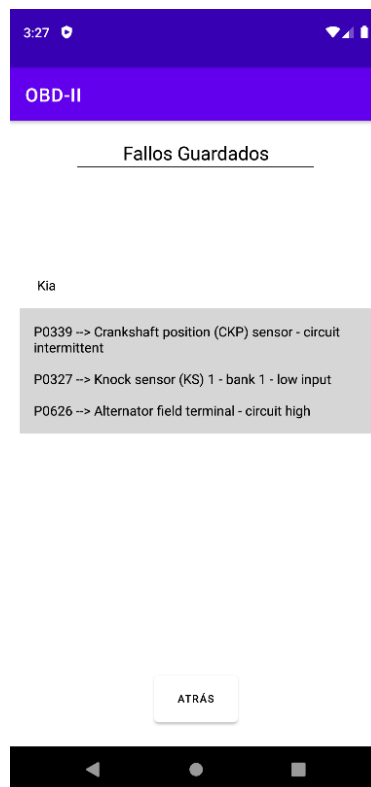
### Sprint 4. Módulo de almacenamiento

Conforme a la planificación, los resultados del Sprint 2 son:

- Implementación del módulo “Datos Guardados”
- Obtención de datos de la BDD

## Implementación del módulo “Datos Guardados”

La **Figura 12** muestra los fallos guardados en la base de datos, permitiendo al usuario de la aplicación tener un control más detallado del vehículo, esto quiere decir que, si el usuario desea visualizar los fallos previos antes del mantenimiento del vehículo, podrán ser encontrados en este apartado.



**Figura 12: Módulo fallos guardados**

### **Sprint 5. Pruebas**

Conforme a la planificación, los resultados del Sprint 2 son:

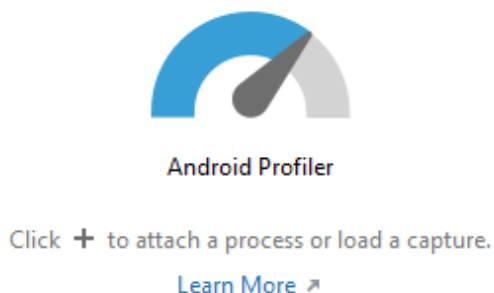
- Pruebas de rendimiento
- Pruebas de aceptación

#### **Pruebas de rendimiento**

Las pruebas de rendimiento tienen el objetivo de poner a prueba un software con la finalidad de medir su estabilidad, velocidad en condiciones de trabajo [29].

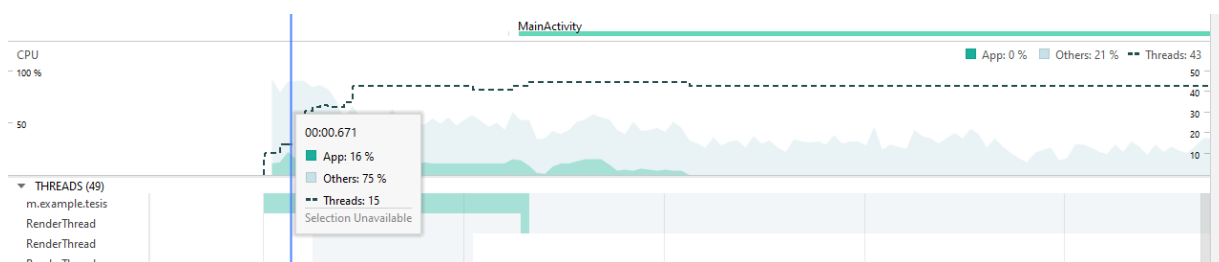
Con el objetivo de conseguir un resultado de rendimiento en módulos específicos de la Aplicación Móvil se realiza una serie de pruebas dentro del entorno de desarrollo de Android Studio. El valor de consumo en CPU esperado debe ser igual o inferior al 30%.

La **Figura 13** muestra el despliegue de la herramienta “*Android Profiler*”, esta nos permite llevar un control adecuado del consumo de CPU en cada una de las ejecuciones de los módulos.

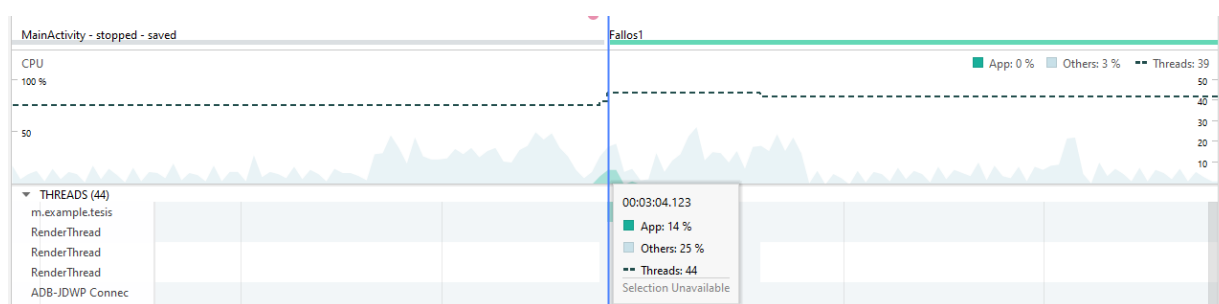


**Figura 13: Herramienta de pruebas**

Desde la **Figura 14** hasta la **Figura 18** se muestran los resultados de consumo de CPU en las actividades MainActivity, Fallos1, Falls2, Estadísticas y FallosGuardados. La **Figura 19** muestra el consumo de red utilizado para realizar la lectura de datos desde Firebase.

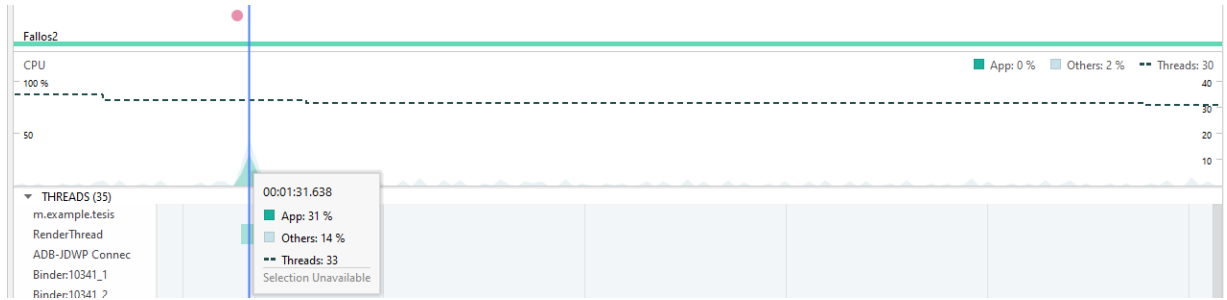


**Figura 14: Resultado de la actividad MainActivity**

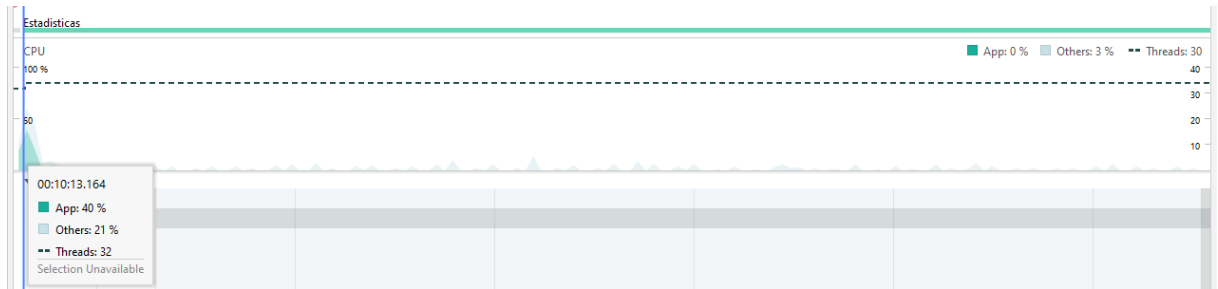


**Figura 15: Resultado de la actividad Fallos1**

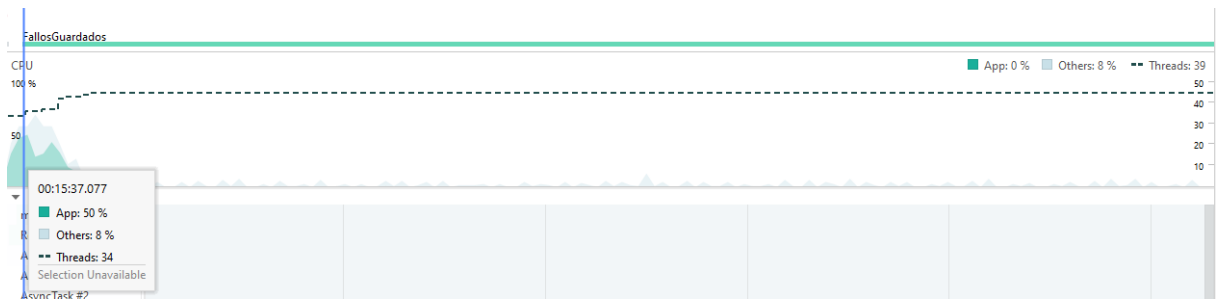
En la **Figura 16** se puede denotar un punto con tonalidad rojiza y un consumo mayor al 30%, dicho punto representa una sentencia ejecutada, en este caso, se denota un “clic” en un botón para realizar la lectura de un archivo.



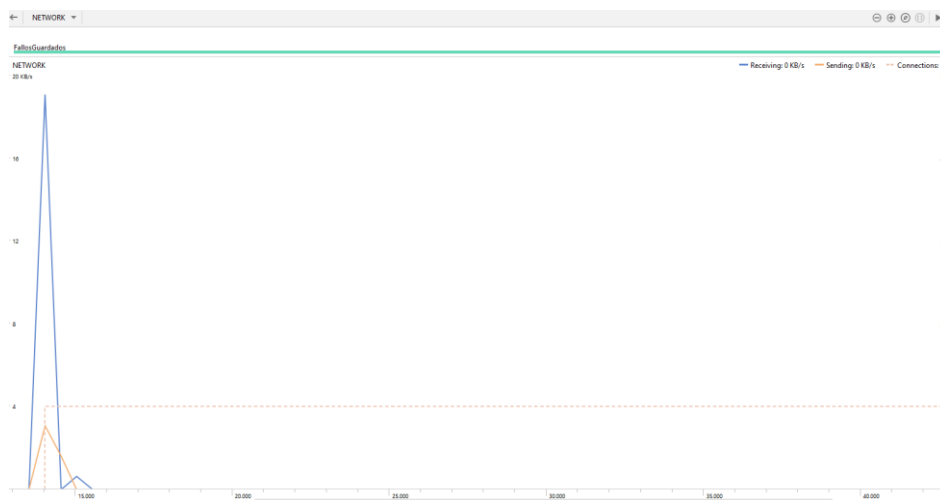
**Figura 16: Resultado de la actividad Fallos2**



**Figura 17: Resultado de la actividad Estadísticas**



**Figura 18: Resultado de la actividad FallosGuardados**



**Figura 19: Resultado del consumo de red en la lectura de datos**

Otra prueba de rendimiento realizada a la Aplicación Móvil puede ser observada en la sección correspondiente (Pág. 4) del Anexo II.

### Pruebas de aceptación

El objetivo de realizar pruebas de aceptación es verificar que la Aplicación Móvil cumpla con los objetivos tomando en cuenta el punto de vista de los usuarios finales. De esta forma cumple con los requerimientos establecidos en los requerimientos iniciales [30].

Desde la **TABLA VIII** hasta la **TABLA XII** se muestran las pruebas de aceptación que se han realizado en el proyecto.

**TABLA VIII: Prueba de aceptación 1 - Importación de datos a Firebase**

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Identificador de prueba (ID):</b> PA001	<b>Identificador historia de usuario:</b> HU005
<b>Nombre prueba de aceptación:</b> Importación de datos a Firebase	
<b>Descripción:</b> El usuario mecánico de un vehículo tiene la posibilidad de importar los datos a la base de datos.	
<b>Condiciones de prueba:</b> <ul style="list-style-type: none"> <li>• Tener conexión a internet</li> <li>• Realizar escaneo previo con la aplicación “Car Scanner”</li> </ul>	
<b>Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>• Exportar los datos desde “Car Scanner”</li> <li>• Presionar el botón “Cargar datos”</li> </ul>	
<b>Resultado deseado:</b> La aplicación móvil cargó los datos los datos a la base de datos en firebase ubicándolos en una única colección con el nombre del vehículo escaneado.	
<b>Evaluación de la prueba:</b> Se ha comprobado el resultado esperado. Aprobación del cliente 100%	

**TABLA IX: Prueba de aceptación 2 – Creación de hipervínculos**

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Identificador de prueba (ID):</b> PA002	<b>Identificador historia de usuario:</b> HU001
<b>Nombre prueba de aceptación:</b> Creación de hipervínculos	
<b>Descripción:</b> El usuario mecánico de un vehículo tiene la posibilidad de acceder al hipervínculo creado visualizando una página web.	
<b>Condiciones de prueba:</b> <ul style="list-style-type: none"> <li>• Tener conexión a internet</li> <li>• Visualizar los códigos de fallo</li> </ul>	
<b>Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>• Leer los datos del módulo fallos</li> <li>• Presionar el ítem con el código de DTC</li> </ul>	
<b>Resultado deseado:</b> La aplicación móvil crea un hipervínculo y redirecciona al usuario hacia una página web con el detalle en profundidad del código de DTC seleccionado.	
<b>Evaluación de la prueba:</b> Se ha comprobado el resultado esperado. Aprobación del cliente 100%	

**TABLA X: Prueba de aceptación 3 – Recuento de datos**

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Identificador de prueba (ID):</b> PA003	<b>Identificador historia de usuario:</b> HU002
<b>Nombre prueba de aceptación:</b> Recuento de datos	
<b>Descripción:</b> El usuario mecánico de un vehículo tiene la posibilidad de visualizar un resumen general de cuáles son los fallos mayoritarios obtenidos.	
<b>Condiciones de prueba:</b> <ul style="list-style-type: none"> <li>• Tener conexión a internet</li> <li>• Visualizar los códigos de fallo</li> </ul>	
<b>Pasos de ejecución:</b>	

<ul style="list-style-type: none"> <li>• Leer los datos del módulo estadísticas</li> </ul>
<p><b>Resultado deseado:</b></p> <p>La aplicación móvil muestra un pequeño resumen de los datos obtenidos, teniendo así un recuento de cuáles son los fallos mayoritarios obtenidos.</p>
<p><b>Evaluación de la prueba:</b></p> <p>Se ha comprobado el resultado esperado.</p> <p>Aprobación del cliente 100%</p>

**TABLA XI: Prueba de aceptación 4 - Obtención de fallos**

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Identificador de prueba (ID):</b> PA004	<b>Identificador historia de usuario:</b> HU003
<b>Nombre prueba de aceptación:</b> Obtención de fallos	
<p><b>Descripción:</b></p> <p>El usuario mecánico de un vehículo tiene la posibilidad de visualizar los fallos exportados desde una aplicación externa accediendo al módulo denominado fallos.</p>	
<p><b>Condiciones de prueba:</b></p> <ul style="list-style-type: none"> <li>• Visualizar los códigos de fallo</li> <li>• Archivo de texto exportado</li> </ul>	
<p><b>Pasos de ejecución:</b></p> <ul style="list-style-type: none"> <li>• Exportar archivo de texto desde “Car Scanner”</li> <li>• Cargar los datos desde la aplicación móvil pulsando el botón “Cargar datos”</li> <li>• Acceder al módulo de “Fallos”</li> <li>• Comenzar la lectura pulsando el botón “Comenzar”</li> <li>• Visualizar los códigos de fallo</li> </ul>	
<p><b>Resultado deseado:</b></p> <p>La aplicación móvil muestra el reporte exportado de la aplicación externa almacenado en un botón denominado “vehículo 1”.</p>	
<p><b>Evaluación de la prueba:</b></p> <p>Se ha comprobado el resultado esperado.</p> <p>Aprobación del cliente 100%</p>	

**TABLA XII: Prueba de aceptación 5 - Fallos guardados**

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Identificador de prueba (ID):</b> PA005	<b>Identificador historia de usuario:</b> HU004
<b>Nombre prueba de aceptación:</b> Fallos guardados	
<b>Descripción:</b> El usuario mecánico de un vehículo tiene la posibilidad de visualizar los fallos guardados del último vehículo escaneado desde una aplicación externa accediendo al módulo denominado “fallos guardados”.	
<b>Condiciones de prueba:</b> <ul style="list-style-type: none"><li>• Conexión a internet</li></ul>	
<b>Pasos de ejecución:</b> <ul style="list-style-type: none"><li>• Ingresar al módulo</li></ul>	
<b>Resultado deseado:</b> La aplicación móvil muestra los fallos guardados del escaneo del último vehículo realizado, y los presenta mediante una separación con el nombre del vehículo y sus respectivos códigos.	
<b>Evaluación de la prueba:</b> Se ha comprobado el resultado esperado. Aprobación del cliente 100%	



## 4 CONCLUSIONES

En esta sección final se listan las conclusiones que se han ido obteniendo a lo largo del proyecto.

- El producto cumple con los objetivos establecidos, permite que los usuarios, en este caso, el mecánico de un vehículo disponga de un medio tecnológico que le permita visualizar los códigos de fallos en el vehículo correspondiente, y prevenir algún accidente debido a la falta de mantenimiento en el automóvil.
- El producto permite que, no solo los mecánicos de un vehículo automotor sino también las personas puedan realizar un escaneo de su vehículo, y consultar el detalle del fallo, sin la necesidad de ir a un taller automotor.
- La metodología ágil Scrum a lo largo del proyecto permite particionar el proyecto, de esta manera, se logra obtener avances funcionales que dan lugar a la culminación exitosa del mismo.
- Las herramientas empleadas para el desarrollo como Cloud Firestore, el IDE de Android Studio, entre otros, permiten obtener una Aplicación Móvil que cumple con la arquitectura de la solución planteada.
- El uso de Cloud Firestore permite almacenar y manipular los datos de forma eficiente para las peticiones que se realizan para la Aplicación Móvil, posibilitando que toda la transaccionalidad de la información pueda obtenerse de la mejor forma sin causar alteraciones y modificaciones en los datos obtenidos.
- El uso del IDE de desarrollo de Android Studio posibilita el desarrollo del proyecto de la mejor forma posible, debido a su composición y método de compilación bajo gradles, cabe resaltar que, para hacer la importación de ciertos modelos usados en el proyecto, se debe de implementar en el archivo de gradle a nivel de módulo, ya que el IDE de desarrollo no lo implementa de forma automática, es decir, funciona de forma similar a cuando se debe realizar un *"npm install"*, con gran diferencia de que el programador debe colocar la implementación deseada en el archivo de gradle.

## 5 RECOMENDACIONES

- Se recomienda, mantener activo una conexión a internet, ya sea, mediante Wifi o el uso de datos móviles, esto debido a que las dos actividades principales de la aplicación móvil se encuentran directamente relacionadas con el envío y lectura a una base de datos.
- Se recomienda realizar el escaneo previo del vehículo y la exportación de datos del archivo de texto mediante la aplicación externa “Car Scanner” debido a que la aplicación móvil desarrollada trabaja con dicho archivo.

## 6 REFERENCIAS BIBLIOGRÁFICAS

- [1] L. A. Aguilar, 2016. [En línea]. Available: <http://repositorio.unsa.edu.pe/bitstream/handle/UNSA/3236/MCagcala04.pdf?sequence=1&isAllowed=y>. [Último acceso: 05 julio 2021].
- [2] J. Ortiz, febrero 2014. [En línea]. Available: [http://biblioteca.usac.edu.gt/tesis/08/08\\_0365\\_EO.pdf](http://biblioteca.usac.edu.gt/tesis/08/08_0365_EO.pdf). [Último acceso: 05 julio 2021].
- [3] J. I. C. Grande y M. C. Ramilo Araujo, de *Definiendo la e-administración: las páginas web de las administraciones públicas*, 2001.
- [4] «Mecánicos y talleres avalados, un pendiente en el país,» *El Telégrafo*, 28 octubre 2018. [En línea]. Available: <https://www.eltelegrafo.com.ec/noticias/quito/1/mecanicos-talleres-avalados-ecuador>. [Último acceso: 27 junio 2021].
- [5] AMT, «Agencia Metropolitana de Tránsito,» [En línea]. Available: <http://www.amt.gob.ec/index.php/servicios/revision-tecnica/centros-de-revision-tecnica-vehicular.html>. [Último acceso: 05 julio 2021].
- [6] «Introducción a Android Studio,» Developers, [En línea]. Available: <https://developer.android.com/studio/intro?hl=es-419>. [Último acceso: 23 agosto 2021].
- [7] M. T. Gallego. [En línea]. Available: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>. [Último acceso: 05 julio 2021].
- [8] K. Schwaber y J. Sutherland, julio 2016. [En línea]. Available: <https://scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Spanish.pdf>. [Último acceso: 16 julio 2021].
- [9] C. L. Jurado, «Bluetooth: historia, características y estándares,» *CCM*, 12 febrero 2021. [En línea]. Available: <https://es.ccm.net/contents/70-bluetooth>. [Último acceso: 02 enero 2022].
- [10] A. R. Castellano, de *Bluetooth. Introducción a su funcionamiento*, Madrid, 2012, pp. 1-16.
- [11] J. M. R. Chillida, «¿Qué es WiFi?,» *InformeTICfacil*, [En línea]. Available: <https://www.informeticplus.com/que-es-wifi>. [Último acceso: 29 enero 2022].
- [12] «¿Qué es Wi-Fi?,» Cisco, [En línea]. Available: [https://www.cisco.com/c/es\\_mx/products/wireless/what-is-wifi.html](https://www.cisco.com/c/es_mx/products/wireless/what-is-wifi.html). [Último acceso: 29 enero 2022].
- [13] «Todo sobre el protocolo OBD2,» *CODIGOSDTC*, [En línea]. Available: <https://codigosdtc.com/obd2/>. [Último acceso: 02 enero 2022].

- [14] J. C. G, «On Board Diagnostics (OBD2),» GPSTotal , 31 julio 2021. [En línea]. Available: <https://gpstotal.org/es/que-es-obdii>. [Último acceso: 02 enero 2022].
- [15] «¿Qué es una computadora automotriz y para qué sirve?,» KIA, 05 diciembre 2020. [En línea]. Available: <https://www.kia.com/pe/util/news/que-es-computadora-automotriz-funciones.html>. [Último acceso: 02 enero 2022].
- [16] «La historia de Android Studio,» Android Studio Facs, 31 mayo 2016. [En línea]. Available: <https://androidstudiofaqs.com/conceptos/android-studio-historia>. [Último acceso: 02 enero 2022].
- [17] C. Rodríguez y R. Dorado, 10 junio 2015. [En línea]. Available: <https://journal.universidadean.edu.co/index.php/Revistao/article/download/1253/1218>. [Último acceso: 05 julio 2021].
- [18] «Scrum,» Sinnaps, [En línea]. Available: <https://www.sinnaps.com/blog-gestion-proyectos/metodologia-scrum>. [Último acceso: 10 noviembre 2021].
- [19] «Qué es SCRUM,» proyectosagiles, [En línea]. Available: <https://proyectosagiles.org/que-es-scrum/>. [Último acceso: 10 noviembre 2021].
- [20] «Conoce los principales roles de Scrum,» Integrait, [En línea]. Available: <https://integrait.com.mx/blog/roles-de-scrum/>. [Último acceso: 11 noviembre 2021].
- [21] «Los tres principales roles en Scrum,» Proyectum, 19 octubre 2016. [En línea]. Available: <https://www.proyectum.com/sistema/blog/los-tres-principales-roles-en-scrum/>. [Último acceso: 11 noviembre 2021].
- [22] «¿Qué son los artefactos Scrum?,» Kaizenia, [En línea]. Available: <https://kzi.mx/que-son-los-artefactos-de-scrum/>. [Último acceso: 11 noviembre 2021].
- [23] «Recopilación de requisitos,» Proyectum, 01 mayo 2013. [En línea]. Available: <https://www.proyectum.com/sistema/blog/recopilacion-de-requisitos/>. [Último acceso: 11 noviembre 2021].
- [24] «Historia de usuario,» Scrum Manager, 26 abril 2011. [En línea]. Available: [https://www.scrummanager.net/bok/index.php/Historia\\_de\\_usuario](https://www.scrummanager.net/bok/index.php/Historia_de_usuario). [Último acceso: 11 noviembre 2021].
- [25] K. Bracey, «¿Qué es figma?,» envatotuts, 26 noviembre 2018. [En línea]. Available: <https://webdesign.tutsplus.com/es/articles/what-is-figma--cms-32272>. [Último acceso: 09 diciembre 2021].
- [26] D. d. Silva, «Herramientas de desarrollo,» Zendesk, 12 enero 2021. [En línea]. Available: <https://www.zendesk.com.mx/blog/herramientas-software-comunmente-utilizadas/>. [Último acceso: 12 diciembre 2021].

- [27] «Firebase, ¿qué es y para qué sirve?,» IEBS, 14 octubre 2016. [En línea]. Available: <https://www.iebschool.com/blog/firebase-que-es-para-que-sirve-la-plataforma-desarrolladores-google-seo-sem/>. [Último acceso: 12 diciembre 2021].
- [28] «Cloud Firestore,» Firebase, [En línea]. Available: <https://firebase.google.com/docs/firestore/rtdb-vs-firestore?hl=es>. [Último acceso: 12 diciembre 2021].
- [29] C. L. Herrera, «¿Qué es una prueba de rendimiento de Software?,» openwebinars, 14 junio 2019. [En línea]. Available: <https://openwebinars.net/blog/que-es-prueba-de-rendimiento-software/>. [Último acceso: 30 enero 2022].
- [30] «Pruebas de aceptación: el qué y el por qué,» digite, [En línea]. Available: <https://www.digite.com/es/agile/pruebas-de-aceptacion/>. [Último acceso: 30 enero 2022].
- [31] L. Carvajal, Metodología de la Investigación Científica. Curso general y aplicado, 28 ed., Santiago de Cali: U.S.C., 2006, p. 139.
- [32] «Modelo Entidad Relacion,» Lucidchart, [En línea]. Available: <https://www.lucidchart.com/pages/es/que-es-un-diagrama-entidad-relacion>. [Último acceso: 12 diciembre 2021].

## **7 ANEXOS**

ANEXO I. Turnitin

ANEXO II. Manual técnico

ANEXO III. Manual de usuario

ANEXO IV. Manual de instalación

## Anexo 1. Certificado de Originalidad

### CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 23 de febrero de 2022

De mi consideración:

Yo, Juan Pablo Zaldumbide Proaño, en calidad de director del Trabajo de Integración Curricular titulado DESARROLLO DE APLICACIÓN MÓVIL CON DISPOSITIVO OBD2 PARA DIAGNOSTICAR EL ESTADO DE UN VEHÍCULO asociado al proyecto DESARROLLO DE APLICACIÓN MÓVIL CON DISPOSITIVO OBD2 PARA DIAGNOSTICAR EL ESTADO DE UN VEHÍCULO, elaborado por el estudiante, HUARACA SALAZAR BYRON SEBASTIAN de la carrera TECNOLOGÍA SUPERIOR EN DESARROLLO DE SOFTWARE, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito completo, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 11%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presentedocumento para los trámites de titulación.

NOTA: Se adjunta el informe generado por la herramienta Turnitin en el Quipux correspondiente.

Atentamente,



Ing. Juan Pablo Zaldumbide Proaño, M.Sc.

Docente ESFOT

## ANEXO II

### 1 Recopilación de requerimientos

La **TABLA I** Lista los requerimientos que han sido recopilados de acuerdo con las necesidades del cliente, mismos que se han definido a través de reuniones

**TABLA I: Recopilación de requerimientos**

RECOPIACIÓN DE REQUERIMIENTOS		
TIPO DE SISTEMA	ID - RR	ENUNCIADO DEL ÍTEM
APLICACIÓN MÓVIL	RR001	Como usuario mecánico de un vehículo, necesito visualizar en detalle el tipo de fallo físico obtenido tales como: <ul style="list-style-type: none"><li>• Fallas en motor</li><li>• Fallas en los frenos</li></ul>
	RR002	Como usuario mecánico de un vehículo, deseo visualizar datos de fallas registradas anteriormente.
	RR003	Como usuario mecánico de un vehículo, necesito visualizar un recuento de fallos mayoritarios en el vehículo.
	RR004	Como usuario mecánico de un vehículo, necesito visualizar una página donde se denoten las características de del fallo de DTC.

### 2 Product Backlog

La **TABLA II** presenta la priorización de cada una de las funcionalidades que se han implementado en el desarrollo de la Aplicación Móvil, las cuales están ordenadas en base a las necesidades del negocio.

**TABLA II: Product Backlog**

ELABORACIÓN DEL <i>PRODUCT BACKLOG</i>				
ID - HU	HISTORIA DE USUARIO	ITERACIÓN	ESTADO	PRIORIDAD
HU005	Importación de datos	1	Completado	Media
HU001	Creación de hipervínculos	1	Completado	Alta
HU002	Recuento de datos	2	Completado	Media
HU003	Obtención de fallos	3	Completado	Alto
HU004	Fallos guardados	4	Completado	Alto



### 3 Sprint Backlog

La **TABLA III** presenta los *Sprints* en los que se ha desarrollado la Aplicación Móvil describiendo las actividades a realizarse en un tiempo determinado para cumplir con los entregables establecidos.

**TABLA III:** *Sprint Backlog*

ELABORACIÓN DEL <i>SPRINT BACKLOG</i>					
ID - SB	NOMBRE	ID - HU	HISTORIA DE USUARIO	TAREAS	TIEMPO ESTIMADO
SB000	Configuración del entorno de desarrollo	N / A	N / A	<ul style="list-style-type: none"> <li>Creación del proyecto en Android Studio</li> <li>Creación de la base de datos</li> <li>Usuarios</li> </ul>	10 H
SB001	Importación de datos y Módulo perfil	HU005	Importación de datos	<ul style="list-style-type: none"> <li>Importación de datos a la BDD</li> </ul>	35 H
		HU001	Creación de hipervínculos	<ul style="list-style-type: none"> <li>Implementación de la redirección</li> <li>Implementación de la BDD</li> </ul>	
SB002	Módulos estados	HU002	Recuento de datos	<ul style="list-style-type: none"> <li>Recuento de daños en función de los datos leídos</li> </ul>	55 H
SB003	Módulo de fallas	HU003	Obtención de fallos	<ul style="list-style-type: none"> <li>Implementación de módulo fallas</li> </ul>	60 H
SB004	Módulo de almacenamiento	HU004	Fallos guardados	<ul style="list-style-type: none"> <li>Implementación del módulo “Datos Guardados”</li> <li>Obtención de datos de la BDD</li> </ul>	
SB005	Pruebas	N / A	N / A	<ul style="list-style-type: none"> <li>Realizar pruebas funcionales               <ul style="list-style-type: none"> <li>○ Pruebas de rendimiento</li> <li>○ Pruebas de aceptación</li> </ul> </li> </ul>	20 H
Documentación					30 H
					240 H

## 4 Prototipo de la aplicación

A continuación, desde la **Figura 1** Error! No se encuentra el origen de la referencia. a la **Figura 5** se ilustran los prototipos de la Aplicación Móvil, permitiendo definir las características que deben tener cada una de las interfaces.



**Figura 1: Pantalla de inicio**



**Figura 4: Módulo de fallos guardados**



**Figura 2: Módulo de estadísticas**



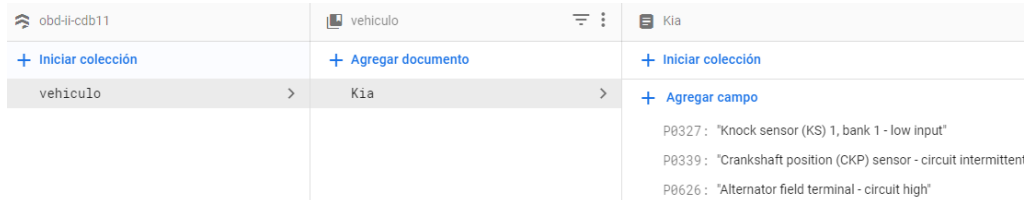
**Figura 5: Módulo de fallos**



**Figura 3: Redirección a página web**

## 5 Diseño de la base de datos

La **Figura 6**, ilustra una colección principal, misma que almacena los documentos generados una vez se realice la importación del archivo de texto obtenido mediante la aplicación “Car Scanner”, de esta forma se mantiene un único uso de una colección que almacenará todos los documentos organizados por vehículos.

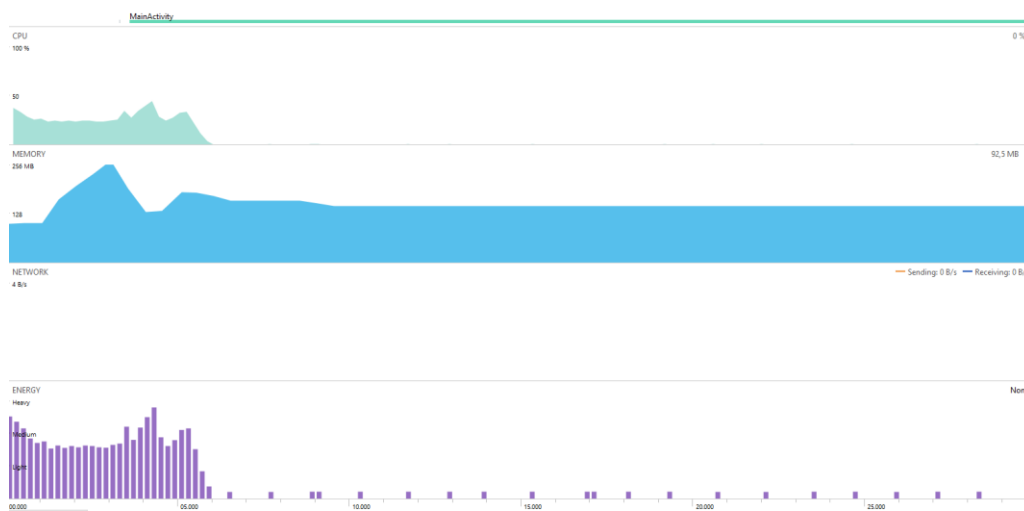


**Figura 6: Diseño de la base de datos - Colección principal**

## 6 Pruebas de rendimiento

La prueba de rendimiento sobre un dispositivo físico o un emulador sobre el computador es realizada con el objetivo de tener resultados precisos sobre la cantidad de recursos que consume la Aplicación Móvil. Con la realización de esta prueba se obtiene un indicador gráfico del rendimiento de la Aplicación Móvil en consumo de memoria, CPU y consumo de internet.

La **Figura 7** muestra el resultado de la prueba de rendimiento realizada en Android Studio obteniendo consumo de CPU, memoria, internet y energía.



**Figura 7: Resultado de la prueba de rendimiento realizado en toda la Aplicación Móvil**

## **ANEXO III**

El manual de usuario lo podemos encontrar en:

<https://youtu.be/axyWCgdhdk>

## **ANEXO IV**

El manual de instalación se encuentra en el archivo Readme.md del repositorio de GitHub:

<https://github.com/Byron040601/TesisProyect/blob/master/README.md>