



**Prueba de concepto mediante el desarrollo de una APP Android para la  
búsqueda de repuestos de vehículos en dispositivos móviles**

**Wilmer Julian Malagon Rueda**

20441223609

**Universidad Antonio Nariño**

Programa Ingeniería Electrónica

Facultad de Ingeniería Mecánica, Electrónica y Biomédica

Villavicencio, Colombia

2021

**Prueba de concepto mediante el desarrollo de una APP Android para la  
búsqueda de repuestos de vehículos en dispositivos móviles**

**Wilmer Julian Malagon Rueda**

Proyecto de grado presentado como requisito parcial para optar al título de:  
**Ingeniero Electrónico**

Director (a):

Ing. Luis Fernando Castañeda Melo, Ph.D

Línea de Investigación:

Automatización - Programación

**Universidad Antonio Nariño**

Programa Ingeniería Electrónica

Facultad de Ingeniería Mecánica, Electrónica y Biomédica

Villavicencio, Colombia

2021

III

## NOTA DE ACEPTACIÓN

El trabajo de grado titulado  
Prueba de concepto mediante el desarrollo de una APP  
Android para la búsqueda de repuestos de vehículos  
en dispositivos móviles, Cumple con los requisitos  
para optar  
Al título de Ingeniero Electrónico.

---

Firma del Tutor

---

Firma Jurado

---

Firma Jurado

## Contenido

Pág.

Contenido	
<b>Resumen</b> .....	<b>1</b>
<b>Abstract</b> .....	<b>2</b>
<b>Introducción</b> .....	<b>3</b>
<b>1. Marco Teorico</b> .....	<b>4</b>
<b>1.1 Licencias de Uso Libre</b> .....	<b>4</b>
<b>1.1.1 Licencias GPL</b> .....	<b>5</b>
<b>1.1.2 Licencias AGPL</b> .....	<b>5</b>
<b>1.1.3 Licencia BSD</b> .....	<b>6</b>
<b>1.1.4 Licencia Apache</b> .....	<b>6</b>
<b>1.2 Entornos de Desarrollo</b> .....	<b>7</b>
<b>1.2.1 Visual Code</b> .....	<b>8</b>
<b>1.2.2 Xampp</b> .....	<b>9</b>
<b>1.2.3 PhpMyAdmin</b> .....	<b>10</b>
<b>1.2.4 Android Studio</b> .....	<b>10</b>
<b>1.3 Bases de Datos</b> .....	<b>13</b>
<b>1.3.1 Necesidad de gestionar los datos</b> .....	<b>13</b>
<b>1.3.2 Base de datos en la web</b> .....	<b>13</b>
<b>2. Metodologia</b> .....	<b>15</b>
<b>2.1 Arquitectura de la aplicación</b> .....	<b>15</b>
<b>2.2 Diseño de interfaz</b> .....	<b>16</b>
<b>2.2.1 Pantalla Principal</b> .....	<b>17</b>
<b>2.2.2 Pantalla numero 2</b> .....	<b>18</b>
<b>2.3 Código de programación en Android Studio</b> .....	<b>19</b>
<b>2.3.1 Activitys</b> .....	<b>19</b>
<b>2.3.2 Código Interfaz Final</b> .....	<b>22</b>
<b>2.4 Diseño base de datos</b> .....	<b>23</b>
<b>2.4.1 Modelo entidad relación</b> .....	<b>23</b>
<b>2.4.2 Esquema de la base de datos</b> .....	<b>24</b>
<b>2.4.3 Base de datos en phpMyAdmin</b> .....	<b>24</b>
<b>2.5 Conectividad entre aplicación y la base de datos</b> .....	<b>26</b>

<b>2.5.1 Iniciar Xampp</b> .....	26
<b>2.5.2 Api–Rest</b> .....	27
<b>3. Resultados</b> .....	<b>29</b>
<b>3.1 Aplicación</b> .....	29
<b>3.1.1 Logo aplicación</b> .....	29
<b>3.1.1 Interfaz final</b> .....	30
<b>3.1.2 Pruebas aplicación</b> .....	32
<b>3.2 Encuesta de satisfacción</b> .....	32
<b>3.3 Base de datos</b> .....	38
<b>3.3.1 Base de datos phpMyadmin</b> .....	38
<b>3.3.2 Api-rest</b> .....	39
<b>Conclusiones</b> .....	<b>40</b>
<b>Anexos</b> .....	<b>41</b>
<b>Referencias Bibliográficas</b> .....	<b>61</b>

## Lista de Figuras

	<b>Pág.</b>
Figura 1-1 Licencia GNU .....	5
Figura 1-2 Licencia AGPL.....	5
Figura 1-3 Licencia Apache.....	6
Figura 1-4 Visual Code.....	8
Figura 1-5 Xampp.....	9
Figura 1-6 Conectividad phpMyAdmin.....	10
Figura 1-7 Actualizaciones Android Studio .....	11
Figura 1-8 Arquitectura Android Studio.....	12
Figura 1-9 Software Utilizados para proyecto .....	12
Figura 1-10 Base de datos en la web .....	14
Figura 2-1 Arquitectura de la aplicación .....	15
Figura 2-2 Formato Json .....	16
Figura 2-3 Diseño interfaz pantalla principal .....	17
Figura 2-4 Diseño interfaz segunda pantalla .....	18
Figura 2-5 Entorno de desarrollo Android studio .....	19
Figura 2-6 Librerías del proyecto .....	20
Figura 2-7 Código para leer formato json.....	21
Figura 2-8 Código para conectar con la api-rest.....	22
Figura 2-9 Grafica identidad relación .....	23
Figura 2-10 Base de datos PhpMyAdmin.....	25
Figura 2-11 Estructura base de datos PhpMyAdmin .....	25
Figura 2-12 Ruta de acceso imágenes para la aplicación .....	26
Figura 2-13 Habilitación de puertos .....	27
Figura 2-14 Código para publicación de Api-Rest .....	28
Figura 2-15 Url Api-Rest.....	28
Figura 3-1 Logo Aplicación.....	30
Figura 3-2 Interfaz final aplicación.....	31
Figura 3-3 Interfaz final segunda pantalla .....	31
Figura 3-4 Grafica pregunta 3.....	34
Figura 3-5 Grafica pregunta 4.....	35
Figura 3-6 Grafica pregunta 5.....	35
Figura 3-7 Grafica pregunta 6.....	36
Figura 3-8 Grafica pregunta 7.....	37
Figura 3-9 Base de datos plataforma PhpMyAdmin.....	38
Figura 3-10 Api-Rest .....	39

## Lista de tablas

	<b>Pág.</b>
Tabla 1-1 Comparación software libre – software propietario .....	4
Tabla 2 Esquema base de datos .....	24
Tabla 3 Personal que realizo la encuesta .....	33

*Dedico este proyecto y la culminación de este título primeramente a los que gracias a su esfuerzo y dedicación me ayudaron con la solución de este trabajo de grado. Quiero agradecer enormemente a mi madre quien siempre estuvo dándome ánimos para no decaer ni renunciar a cumplir este sueño.*

*A mis hijos mi esposa lo cual son pilares fundamentales en mi vida, pero en especial me dedico este triunfo ya que a pesar de todas las adversidades logre sacar el título adelante, pase por momentos difíciles, pero al ver culminada la carrera siento que valió la pena todo ese esfuerzo.*

## Resumen

Las bases de datos y las aplicaciones móviles han venido teniendo un gran auge dentro de las empresas en la actualidad. Es por esto que se podría afirmar que el 95% de los trabajadores tienen en su poder un teléfono inteligente, y es aquí donde las industrias sin invertir en estos artefactos los utilizan buscando automatizar y agilizar los procesos dentro de sus instalaciones generando mayores ganancias y ahorrando recursos.

La finalidad de este proyecto es la de combinar estas dos herramientas y agilizar un proceso de búsqueda de repuestos para vehículos móviles en una aplicación para teléfonos inteligentes con sistema operativo Android, los usuarios tendrán la oportunidad de tener en sus teléfonos móviles gran cantidad de información de forma rápida y sencilla, así como la opción de buscar algún repuesto y ver información importante de este.

Se busca agilizar la disponibilidad de los repuestos para automotores para su instalación en los vehículos móviles.

## **Abstract**

Databases and mobile applications have been having a great boom within companies today. This is why it could be said that 95% of workers have a smartphone in their possession, and it is here where industries without investing in these devices use them seeking to automate and streamline processes within their facilities, generating greater profits and saving means.

The purpose of this project is to combine these two tools and streamline a search process for spare parts for mobile vehicles in an application for smartphones with Android operating system, users will have the opportunity to have a large amount of information on their mobile phones. quick and easy way as well as the option to find a spare part and see important information about it.

It seeks to speed up the availability of automotive spare parts for installation in mobile vehicles.

## Introducción

Hoy en día un automóvil común está conformado por aproximadamente 70.000 piezas, y los grandes talleres de vehículos han desarrollado múltiples maneras de caracterizar estas piezas, con el fin de llevar un mejor control en sus inventarios, la clasificación más utilizada es por subsistemas, un vehículo automotor se encuentra dividido en 8 subsistemas los cuales son (electricidad, motor, neumático, suspensión, refrigeración, carrocería, rodaje y llantas).

Este método de clasificación ayuda a llevar un inventario más ordenado, sin embargo, cuando los técnicos especializados solicitan un repuesto es un poco engorroso ya que son demasiadas las piezas que se encuentran en estas listas, peor aún si son impresas; además muchas de estas piezas pueden ser utilizadas en distintos subsistemas lo que lleva al técnico a buscar en diferentes listas. Otro gran problema a la hora de la búsqueda de repuestos es que se pueden encontrar piezas con nombres o características físicas similares, lo que retrasa la intervención del vehículo ya que en muchas ocasiones retiran del almacén la pieza equivocada, generando un embotellamiento en los procesos de mantenimiento.

Para el capítulo 1 se estudiará todos los conceptos básicos que se van a utilizar en el desarrollo de este proyecto, podremos entender temas importantes como las licencias de desarrollo web, que tipos de software se requieren para desarrollar este proyecto.

El capítulo 2 se centrará en el proceso para desarrollar la aplicación móvil, así como la base de datos y la publicación en la web de esta. También se explicará en manera general el código de programación que se utilizó para la ejecución del proyecto.

El capítulo 3 se observará los resultados obtenidos la funcionalidad de la aplicación pruebas en diferentes celulares móviles, así como la base de datos ya construida y publicada en la web lista para ser consumida por los usuarios, por último, se tendrán los anexos donde estará todo el código de programación que se utilizó para la ejecución del proyecto.

## 1. Marco Teorico

### 1.1 Licencias de Uso Libre

En los años 80, en el inicio de la era informatica, las empresas comenzaron a crear licencias de manejo de software, todo para protegerse del plagio, o el robo de ideas desarrolladas. En los ultimos 10 años estas licencias han evolucionado dividiendose en software propietario y software libre.

El software propietario se basa especificacmente en que no puede ser redistribuido o modificado sin autorizacion del desarrollo.

El software libre en cambio tiene muchas libertades sobre el mismo, ya que tiene la opcion de ser distribuido, copiado, modificado al antojo del usuario.

Tabla 1-1  
Comparación software libre – software propietario

SOFTWARE LIBRE	SOFTWARE PROPIETARIO
El código puede ser modificado	No tiene acceso al código
No requiere de actualizaciones	Requiere Actualizaciones
Sin costo	Tiene Costo
Orientación mediante foros, blogs, artículos	Orientación mediante paginas oficiales, propietarios y manuales
Libertad de uso	Limita el uso del usuario

Fuente: Imagen propia

En la actualidad la legislacion ha venido prestando gran importancia en el campo de los derechos de autor o propiedad intelectual, basicamente lo que explica es que no se puede hacer o desarrollar casi nada si el autor no lo autoriza. Las licencias de uso libre nos dan cierto permiso para poder ya sea distribuir, comercializar o y modificar los programas diseñados por parte de nosotros. El tipo de licencia a utilizar en cada programa es elegido por el autor o en este caso del programador, es importante recalcar que cualquier actualizacion que se haga en dicho programa podra ser modificada la licencia.

Las licencias de uso libre mas utilizadas son:

### 1.1.1 *Licencias GPL*

Figura 1-1 Licencia GNU



Fuente: Internet

La licencia Publica General es una de las licencias mas utilizadas en la actualidad, esta permite al usuario libre distribucion modificacion y uso, pero siempre y cuando el nuevo desarrollo que se genere quede con el mismo tipo de licencia, inclusive si solo una parte de un proyecto se desarrolla con un software libre que contenga esta licencia, automaticamente todo el programa quedaria con licencia GPL.

La licencia GPL se diseño para dar libertad de codigo esto quiere decir que una vez un programa sea licenciado bajo esta normativa nunca podra ser echo propietario, ni se podra nunca modificar la licencia asi se le hagan modificacion futuras al codigo, esque quiere decir que una vez tenga licencia GPL nunca se podra modificar.

### 1.1.2 *Licencias AGPL*

Figura 1-2 Licencia AGPL



Fuente: Imagen propia

La licencia AGPL o licencia Affero AGPL es una licencia tipo copyleft derivada de la licencia GPL con la diferencia de que obliga a distribuir el código fuente

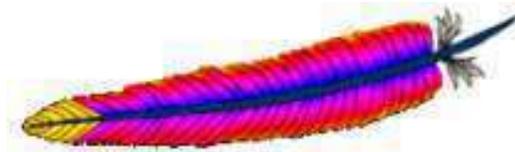
modificado en aplicaciones que solo funcionan cuando son ejecutadas en internet. Esto quiere decir que si se realiza alguna modificación al código fuente y no se distribuye no es obligatorio proporcionar el código modificado a nadie y utilizarlo de forma privada.

### **1.1.3 Licencia BSD**

La licencia BSD (Berkeley Software Distribution) es una de las más libres, solo se reserva los derechos de autor y la propiedad intelectual, solo exige que el autor sea mencionado correctamente. A diferencia de las otras licencias, esta permite la utilización de código fuente en software no libre. Esta licencia es considerada como la antagonista de la licencia GPL ya que si utiliza un tipo de licencia diferente no es obligación que siga siendo BSD.

### **1.1.4 Licencia Apache**

Figura 1-3 Licencia Apache



Fuente: Apache página oficial

La licencia Apache tiene como principal función preservar los derechos de autor aparte de algunos permisos que otorga como lo son:

El programa diseñado puede ser usado comercialmente, si el programa es modificado puede añadir un copyright adicional, además de que siempre tienen que ser incluidas en el software, y los cambios deben ser documentados.

- El software puede ser distribuido sin ningún tipo de regalías.
- El software puede ser usado y modificado en software privado.
- Así como otorga permisos, también tiene limitaciones como son:
- Si hay algún tipo de daño en el software, no habrá responsables de este.
- Prohíbe el uso de marcas registradas.
- La licencia no proporciona ninguna garantía.

El uso de las licencias es bastante importante en la actualidad y es por eso que, a la hora de diseñar, elaborar o crear algún programa software o código de programación es recomendable verificar las licencias que utilizas los softwares en los cuales vamos a trabajar, también es importante el objetivo o la función que va a cumplir ese programa, ya que si es para uso comercial la licencia que se debe utilizar no va ser igual que para uso plenamente didáctico.

## 1.2 Entornos de Desarrollo

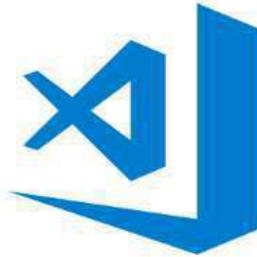
Los entornos de desarrollo han agilizado y dinamizado el trabajo de los programadores actualmente, gracias a estos hemos podido de una forma más didáctica crear programas y o aplicaciones móviles que nos facilitan el modo de vivir, debido a la pandemia del covid 19 el acercamiento entre personas se ha disminuido drásticamente y es por esto que los programadores se están centrando en el diseño de aplicaciones móviles para el uso en el entorno laboral.

El lenguaje binario (1,0), es el método en el cual las computadoras realizan todas las acciones que nosotros ordenamos desde los periféricos (teclado, mouse, impresora, etc). Diseñar en este tipo de códigos es bastante difícil, por no asegurar imposible, es por ello que la función principal de estos entornos de desarrollo es la de traducir las acciones que digitamos en los códigos de programación, cada entorno de desarrollo tiene un lenguaje de programación diferente, pero la lógica es la misma. Es como decir en los países de habla hispana casa, y en los países de habla inglesa house, así mismo es en los entornos de desarrollo, en unos se utiliza lenguajes java, en otros, c, también c++, php ect.

En este proyecto se utiliza entornos de desarrollo con lenguajes de programación basados en java y php como son Visual Code, Xampp, phpMyAdmin y Android Studio, los cuales se hará una breve descripción para tener una idea de su manejo y las características y función principal en este trabajo.

### 1.2.1 Visual Code

Figura 1-4 Visual Code



Fuente: Visual code página oficial

Visual Studio Code es un editor de texto gratuito creado por Microsoft con soporte para Linux, Windows, macos. Incluye soporte para depuración, control de Git integrado, resaltado de sintaxis, finalización de código inteligente, fragmentos de código y refactorización de código. También es personalizable, de modo que los usuarios pueden cambiar el tema del editor, los métodos abreviados de teclado y las preferencias. Es gratuito y de código abierto.

Visual Studio Code fue anunciada en el año 2015, su principal característica es la combinación de la interfaz de usuario optimizada de un editor moderno con asistencia y navegación de código enriquecido y una experiencia de depuración integrada, sin la necesidad de un IDE completo. Visual Studio Code, cuenta con herramientas de Debug hasta opciones para actualización en tiempo real de nuestro código en la vista del navegador y compilación en vivo de los lenguajes que lo requieran (por ejemplo, en el caso de SASS a CSS). Además de las extensiones, tendremos la posibilidad de optar por otros temas o bien configurarlo a nuestro gusto.

Las principales ventajas de este entorno de desarrollo son:

- Es una herramienta que tiene soporte nativo para gran variedad de lenguajes.
- La interfaz puede ser configurada a gusto del programador, así se tiene la oportunidad de los proyectos guardarlos en carpetas y en forma ordenada.
- Tiene la opción de cambiar el tema en la interfaz, color de la pantalla, tamaño de la letra etc.

## 1.2.2 Xampp

Figura 1-5 Xampp



Fuente: Xampp página oficial

XAMPP es una extensión de Apache que contiene el desarrollo web más común tecnologías. Usualmente es utilizado para el desarrollo de aplicaciones en PHP Y MySQL, Su uso e implementación es completamente gratuita y fácil de descargar.

Existen dos paquetes diferentes de Xampp una es la convencional que es la que utilizaremos en este proyecto, y la otra es Xampp Lite un paquete un poco más liviano que tiene menos opciones de conectividad. En el siguiente cuadro se evidenciará las diferencias entre los dos paquetes y que software son compatibles con cada uno de ellos.

Tabla 1-2 Diferencia Xampp - Xampp lite

Technology	Xampp	Xampp Lite
Apache HTTP Server	X	X
Php	X	X
MySQL	X	X
PhpMyAdmin	X	X
Openssl	X	X
Sqlite	X	X
FileZilla FTP Server	X	
Pear	X	
Adodb	X	
Mercury Mail Transport System	X	
Webalizer	X	
Zend Optimizer	X	
Xampp Control Panel	X	
Xampp Security	X	

Fuente: Tabla propia

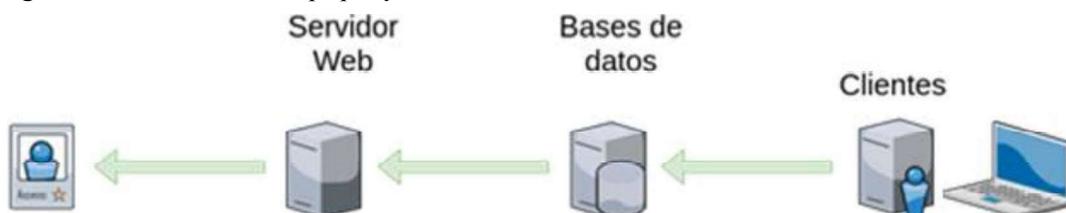
Su función principal es la conexión con los principales gestores de bases de datos como lo es phpMyAdmin, se encarga de habilitar los puertos para poder tener esa conexión.

### 1.2.3 *PhpMyAdmin*

PhpMyAdmin es un gestor para base de datos diseñado en php, es utilizado para la administración de MySQL a través de la web, su nombre viene de la combinación de PHP lo cual es el lenguaje de programación en el cual fue diseñado y MySQL como la base de datos que administra, su primera versión fue lanzada en el año de 1998 y desde ese año hasta la fecha ha tenido 35 actualizaciones, los cuales las mejoras son bastante significativas.

Su interfaz es bastante amigable con el usuario y su código o lenguaje de programación es muy simple, además de también tener la posibilidad de manejar las bases de datos por medio de su interfaz.

Figura 1-6 Conectividad phpMyAdmin



Fuente: Imagen propia

### 1.2.4 *Android Studio*

Android estudio es un entorno de desarrollo integrado oficial para el desarrollo de aplicaciones móviles para el sistema aplicativo Android algunas características de este software son:

- Sistema de compilación flexible
- Emulador rápido y cargado de funciones

- Un entorno en el cual puede desarrollar para todos los dispositivos de Android.
- Integración con GitHub y plantillas de código
- Variedad de marcos de trabajo y herramientas de prueba
- Compatibilidad con C++ y NDK

A continuación, se mostrarán algunas de las versiones que se han publicado de Android estudio.

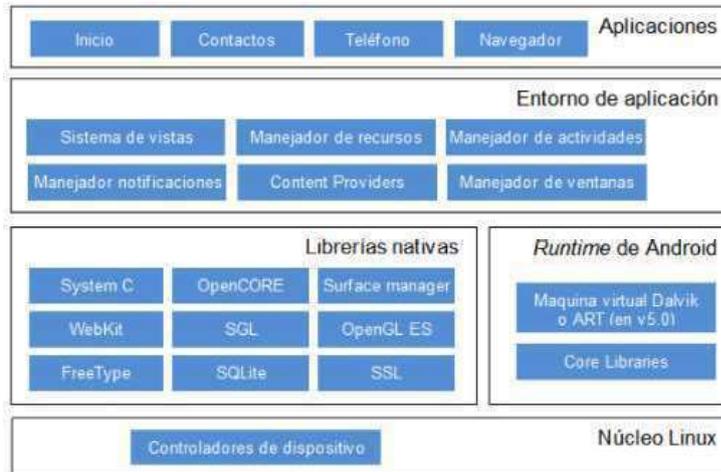
Figura 1-7 Actualizaciones Android Studio



La versión actual de Android studio es Android 12 anunciado el 18 de febrero del año 2021.

La arquitectura de Android se muestra en el siguiente cuadro.

Figura 1-8 Arquitectura Android Studio



Fuente: Universidad politécnica de valencia página oficial

Los entornos de desarrollo utilizados para este proyecto tienen licencia de código abierto lo que significa que no presenta ningún tipo de inconveniente al ser utilizados para este propósito, a continuación, se mostrara un cuadro para explicar mejor el tipo de licencia que abarca cada uno.

Figura 1-9 Software Utilizados para proyecto

	Framework	Licencia	Características
	Visual Studio Code	GPL (Licencia Publica General)	Esta licencia permite el uso y la distribución de los programas sin necesidad de pagarle a alguien, si se puede cobrar cuando es usted mismo quien entrega la copia, no se puede exigir ningún pago cuando alguna otra persona es quien entrega la copia. Esta es la licencia mas utilizada en la programación ya que es la que mas manejan los software de uso libre.
	Xampp	GPL (Licencia Publica General)	
	PhpMyAdmin	GPL (Licencia Publica General)	
	Android Studio	Licencia Apache	Esta licencia es creada por apache permitiendo a los usuarios utilizarla para cualquier proposito (distribuir, modificar)si tener que preocuparse por regalías.

Fuente: Imagen propia

### 1.3 Bases de Datos

Las bases de datos son conjuntos de datos almacenados en memorias externas que están organizados mediante una estructura de datos. Una base de datos se diseña para contener información de una empresa o algún tipo de organización está conformada por datos relevantes ya sea de personas, o elementos.

Una base de datos se puede percibir como un gran almacén de datos que se define y se crea una sola vez, y que se utiliza al mismo tiempo por distintos usuarios. En una base de datos todos los datos se integran con una mínima cantidad de duplicidad. De este modo, la base de datos no pertenece a un solo departamento, sino que se comparte por toda la organización. Además, la base de datos no sólo contiene los datos de la organización, también almacena una descripción de dichos datos. Esta descripción es lo que se denomina metadatos, se almacena en el diccionario de datos o catálogo y es lo que permite que exista independencia de datos lógica-física.

#### 1.3.1 Necesidad de gestionar los datos

Hoy en día se ve la cada vez más la necesidad de gestionar las bases de datos de una forma ágil y rápida, ya que podemos encontrar grandes empresas u organizaciones que contienen una inmensa cantidad de datos en su poder todo esto gracias a el auge del internet y la aparición de los dispositivos móviles.

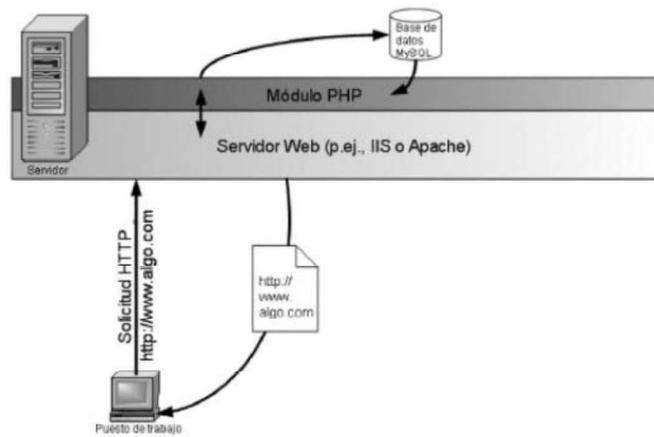
La informática se ha enfocado en proporcionar herramientas que faciliten esa gestión de datos, ya que antes de que apareciera las empresas tenían que guardar esa información en carpetas y cajones convirtiéndolo en un proceso manual, y el tiempo utilizado para gestionar esa información era enorme, es por esto que cada vez es importante el manejo de bases de datos por medio de sistemas informáticos.

#### 1.3.2 Base de datos en la web

La mayoría de las bases de datos son gestionadas por medio de la web con protocolo http, esto se hace digitando en la barra de dirección del navegador la dirección o url de la base de datos, lo que hace el navegador es llamar el sitio web que se encuentra detrás del dominio. si la página llamada contiene código que se deba ejecutar en el servidor, el servidor web llama al interprete necesario, que

ejecuta el código y devuelve el código HTML producido, es entonces que se mostrara en el navegador la base de datos solicitada.

Figura 1-10 Base de datos en la web

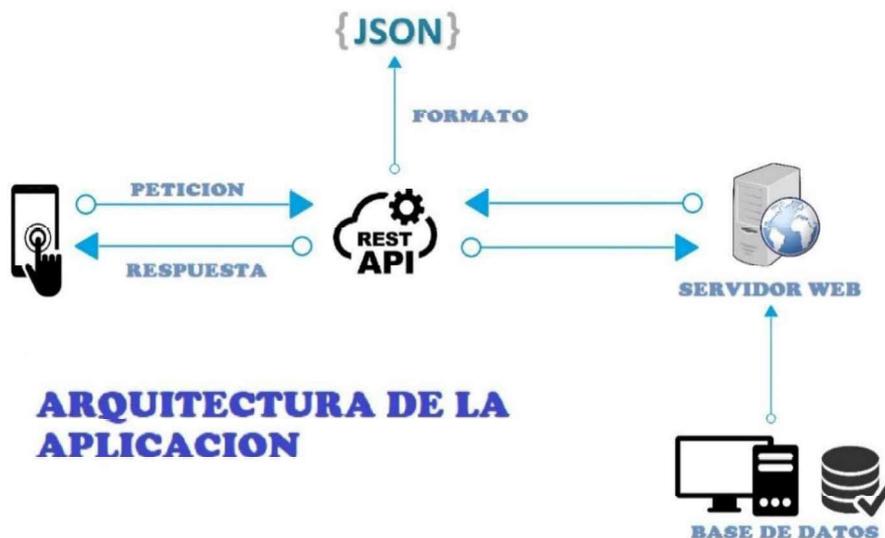


## 2. Metodología

Este capítulo se centra en el procedimiento que se eligió para el diseño de la aplicación búsqueda de repuestos para vehículos, desde su arquitectura, hasta los códigos de programación que se utilizaron para que la aplicación funcione correctamente en dispositivos móviles con sistema operativo Android, de igual forma se profundizara en el desarrollo de la base de datos, desde su diseño hasta el método de conexión con la aplicación, en los anexos se encontraran el manual de manejo básico para crear bases de datos en el entorno de desarrollo llamado phpMyAdmin.

### 2.1 Arquitectura de la aplicación

Figura 2-1 Arquitectura de la aplicación



Fuente: Imagen propia

Como podemos observar en la imagen, la arquitectura de este proyecto se centra en la cliente servidor. La aplicación será el cliente el cual solicitará información a la base de datos, esta será diseñada desde una plataforma basada en php llamada phpMyAdmin, con ayuda de un pequeño código de programación creado también en este lenguaje se diseñó una APIREST que mostrará la base de datos en formato JSON y es de donde se suministra la información para la aplicación móvil.

El formato JSON es un formato que en la actualidad ha venido teniendo gran importancia para el manejo y visualización de las bases de datos, ya que es bastante sencillo de entender como lo vemos en la siguiente imagen.

Figura 2-2 Formato Json

```
[
  {
    "rep": "ELE11369",
    "nombre": "Baterías 4D",
    "subsistema": "Electricidad",
    "tipologia": "Todos",
    "descripcion": "La bater\u00eda de un coche es un\u00a0acumulador\u00a0que se encarga
arrancar y el sistema el\u00e9ctrico general del veh\u00edculo para funcionar.\r\n",
    "imgResource": "bateria_4d.jpg"
  }
]
```

Fuente: Imagen propia

Es importante aclarar que se eligió la comunicación por medio de una APIREST ya que es un método seguro de proteger el servidor que para este proyecto será un computador portátil.

Cuando el cliente inicia la aplicación el dispositivo móvil por medio de datos de internet se conecta a la API – REST descargando toda la información de la base de datos, esto quiere decir que cada vez que se actualice un repuesto, se añada o se elimine, es necesario cerrar la aplicación y volverla abrir para poder verse los cambios que se realizaron.

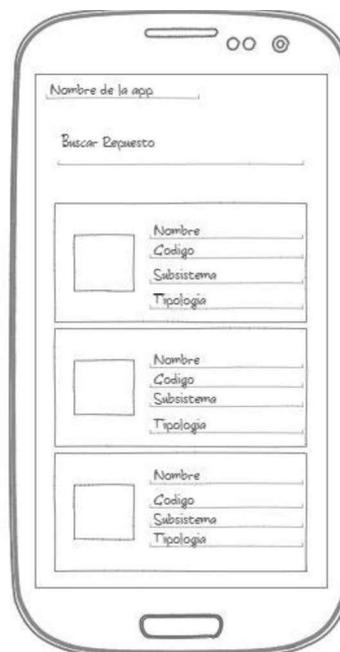
## 2.2 Diseño de interfaz

El diseño de la interfaz en pensado en un manejo fácil y eficaz, ya que como usualmente va hacer utilizado en momentos de trabajo bajo presión es necesario la simplicidad de su manejo, así como el tiempo de respuesta a las peticiones que se le hagan.

Es por eso que consta de 2 pantallas las cuales la primera o pantalla principal se podrá observar la base de datos completa, y la segunda pantalla mostrará la información mas especifica del repuesto seleccionado.

### 2.2.1 Pantalla Principal

Figura 2-3 Diseño interfaz pantalla principal



Fuente: Imagen fuente propia

Como se mencionaba anteriormente la pantalla principal tendrá la función de mostrar la base de datos en su totalidad, se podrá observar unos recuadros los cuales contienen la imagen del repuesto, el código con el cual se puede sacar de almacén, el subsistema al cual pertenece y la tipología del móvil a cuál aplica. Si desea el técnico buscar un repuesto de forma mucho más rápida, tendrá que pulsar encima de la frase “Buscar Repuesto” automáticamente se desplegara el teclado y el usuario podrá ingresar el nombre completo o solo parte de el, automáticamente la aplicación empezara a mostrar repuestos que coincidan con lo que está digitando, si el repuesto se encuentra en la base de datos lo mostrar en pantalla en caso de que no esté, la pantalla quedara completamente vacía. Si el repuesto se encuentra y el técnico desea información adicional solo tendrá que pulsar encima

del repuesto y se abrirá una segunda pantalla. Si desea buscar otro repuesto solo basta con borrar lo digitado e insertar el nuevo repuesto.

### 2.2.2 Pantalla numero 2

Al momento de elegir el repuesto en la pantalla principal y dar clic sobre él se abrirá la pantalla número 2, esta pantalla mostrará la misma información que la pantalla 1 pero solo del repuesto elegido, la imagen se podrá observar más grande, debajo de esta podrá observarse el nombre del repuesto, el código, el subsistema, la tipología y por ultimo una pequeña descripción del repuesto, o se podrá ver características importantes, como el torque al cual se debe ajustar, precauciones etc.

En caso de que el usuario ya no desee más información o desee buscar otro repuesto, solo deberá oprimir encima de la opción atrás y lo regresara a la pantalla principal.

Figura 2-4 Diseño interfaz segunda pantalla



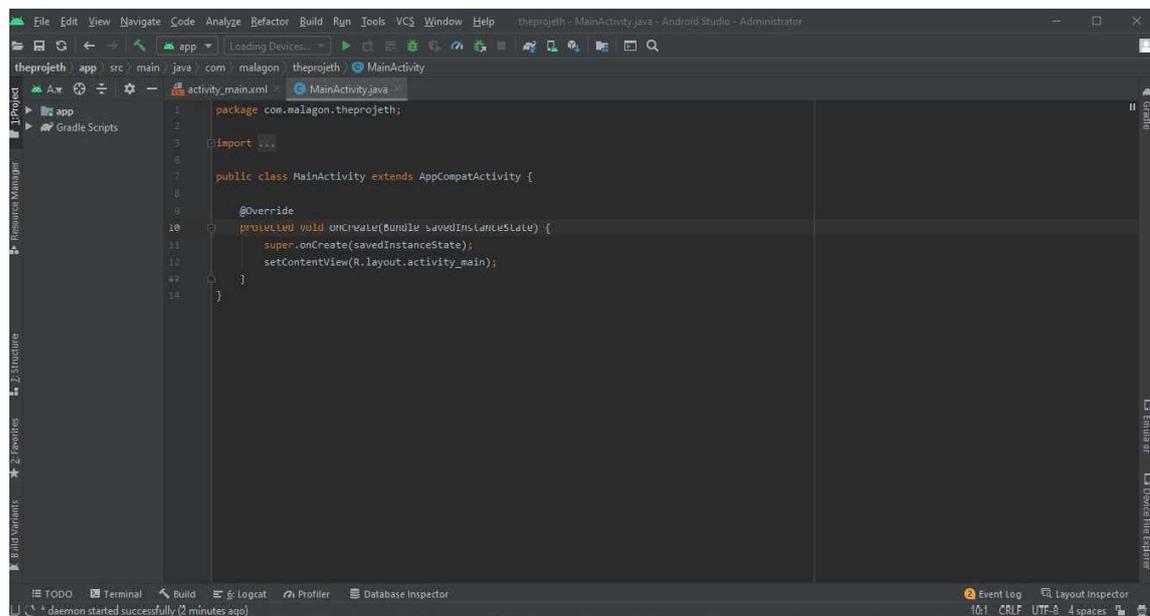
Fuente: Imagen propia

## 2.3 Código de programación en Android Studio

La base de las aplicaciones móviles o cualquier tipo de software es su código de programación, es aquí donde se digitan las instrucciones para que la aplicación funcione y cumpla las especificaciones deseadas.

Android studio es el entorno de desarrollo principal para diseño de aplicaciones móviles con sistema operativo Android, su lenguaje de programación es basado en java su interfaz en práctica lo que facilita llevar proyectos robustos de una forma ordena. Para desarrollar nuestro proyecto crearemos pequeñas paginas llamadas activitys cada una de estas cumplirá una función vital en nuestra aplicación. El apartado 2.3.1 se enfocará en cada una de esas activitys donde se conocerá la función principal de cada una.

Figura 2-5 Entorno de desarrollo Android studio



Fuente: Imagen propia

### 2.3.1 Activitys

Como se comentaba anteriormente para el desarrollo de esta aplicación, se optó por crear una serie de activitys las cuales cada una de ellas cumple una función importante en la ejecución del proyecto, a continuación de describirán cada una de ellas y se observara su respectivo código de programación.

- **Librerías**

Las librerías en los entornos de desarrollo son importantes ya que es un paquete de código predeterminado para realizar diferentes actividades en un código de programación. Por ejemplo, la librería Picasso la cual se utilizó para poder visualizar imágenes desde la web, otro claro ejemplo es la librería Retrofit2 cuya función principal es la conexión con servidores locales. Las librerías nos facilitan y nos ahorran líneas de código para ejecutar aplicaciones. A continuación, se mostrarán las librerías utilizadas para este proyecto.

Figura 2-6 Librerías del proyecto

```
dependencies {
    implementation 'androidx.appcompat:appcompat:1.2.0'
    implementation 'com.google.android.material:material:1.3.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
    implementation 'androidx.recyclerview:recyclerview:1.1.0'
    implementation 'androidx.cardview:cardview:1.0.0'
    testImplementation 'junit:junit:4.+'
    androidTestImplementation 'androidx.test.ext:junit:1.1.2'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'
    implementation 'com.squareup.retrofit2:retrofit:2.3.0'
    implementation 'com.google.code.gson:gson:2.8.2'
    implementation 'com.squareup.retrofit2:converter-gson:2.3.0'
    implementation 'com.squareup.picasso:picasso:2.71828'
```

Fuente: Imagen propia

- **Detalles de la Actividad**

La función principal de esta actividad es la de crear las variables principales de la aplicación, es aquí le damos la identidad a estas, las imágenes van hacer tipo ImageView, las opciones como nombre, rep, tipología, son creados tipo TextView lo que quiere decir que van hacer solo texto. A partir de esta actividad la programación se empezará a ejecutar. Ver anexo1.

- **Lista de Artículos**

En esta actividad se pretende configurar las listas de repuestos, es aquí donde el código de programación ordena a la aplicación mostrar la lista de los repuestos de

una forma fácil y sencilla, los repuestos serán mostrados en recuadros donde se podrá observar la imagen y las opciones del repuesto. Ver anexo 2.

- **Actividad Principal**

Cuando una aplicación o programa inicia, es necesario tener un punto de partida, La actividad principal o ActivityMain, es una actividad que crea el programa por defecto además que es donde inicia la aplicación. Es la más importante por partir de estos códigos la aplicación empieza a funcionar. Ver anexo 3.

- **Adaptador de Reciclador**

El Adaptador es la actividad encargada de realizar el filtro cuando el usuario se encuentra buscando un repuesto, además de adaptar la interfaz, pues cada vez que digita una letra, empieza a mostrar los repuestos que en su nombre contienen esas letras digitadas. Esta es la actividad principal pues es la actividad que agiliza el trabajo del usuario. Ver anexo 4.

- **Servicio de Actualización de Api**

Retrofit2 es la librería encargada que crear el cliente HTTP de tipo seguro para Android y Java, tiene la característica de leer datos tipo Json o Xml, luego analiza esos objetos y los muestra en la pantalla de nuestro teléfono celular, gracias a esta librería podemos tener seguridad nuestra base de datos ya que consume los datos de una Api-Rest y no directamente de la plataforma phpMyAdmin.

Figura 2-7 Código para leer formato json

```

1 package com.malagon.theproject;
2
3 import java.util.List;
4
5 import retrofit2.Call;
6 import retrofit2.http.GET;
7
8 public interface RetrofitApiService {
9
10     @GET("api.php")
11     Call<List<ItemList>> getItemsDB();
12
13 }
14

```

Fuente: Imagen propia

- **Retrofit Cliente**

Para poder consumir datos de la Api Rest, es necesario conectarse con el servidor local, para ello es necesario diseñar el código para acceso del cliente, o conectividad de la aplicación. Esta conexión se realiza por medio del protocolo de internet IPv4 ES es por esta razón que gracias a Xampp, habilitamos los puertos para facilitar esta conexión. El código es bastante corto y sencillo, lo único que necesitamos en la dirección IP del servidor para el caso de este proyecto es el del computador portátil. Por seguridad se eliminaron los últimos números de la dirección IP del computador que se utilizó para este proyecto.

Figura 2-8 Código para conectar con la api-rest

```

1 package com.malagon.theproject;
2
3 import retrofit2.Retrofit;
4 import retrofit2.converter.gson.GsonConverterFactory;
5
6 public class RetrofitClient {
7
8     private static Retrofit retrofit = null;
9     public static final String URL_BASE = "http://192.168.0.█:█/";
10
11     public static RetrofitApiService getApiservice() {
12         if (retrofit == null) {
13             retrofit = new Retrofit.Builder()
14                 .baseUrl(URL_BASE)
15                 .addConverterFactory(GsonConverterFactory.create())
16                 .build();
17         }
18         return retrofit.create(RetrofitApiService.class);
19     }
20 }
21

```

Fuente: Imagen propia

### 2.3.2 Código Interfaz Final

Android Studio tiene la capacidad de desarrollar la interfaz de forma manual y o diseñarla a partir de código de programación la forma de crearla depende de la destreza del programador. Para este caso se crearán líneas de código para la interfaz, en estas líneas irán el tipo de letra a usar, el tamaño, color, que tan grande queremos la imagen, y una línea de código que nos creara el recuadro donde ira insertada toda la información.

Es importante que cada variable debe tener sus propias características, esto quiere decir que para la imagen tenemos que digitar unas líneas de código, donde se especificara el tamaño la ubicación en el recuadro etc, para el nombre, el rep, la tipología, el subsistema igualmente cada una debe tener líneas de código donde nos especifique el tamaño, si va en negrilla, tipo de letra, ubicación etc. Ver anexo 5.

## 2.4 Diseño base de datos

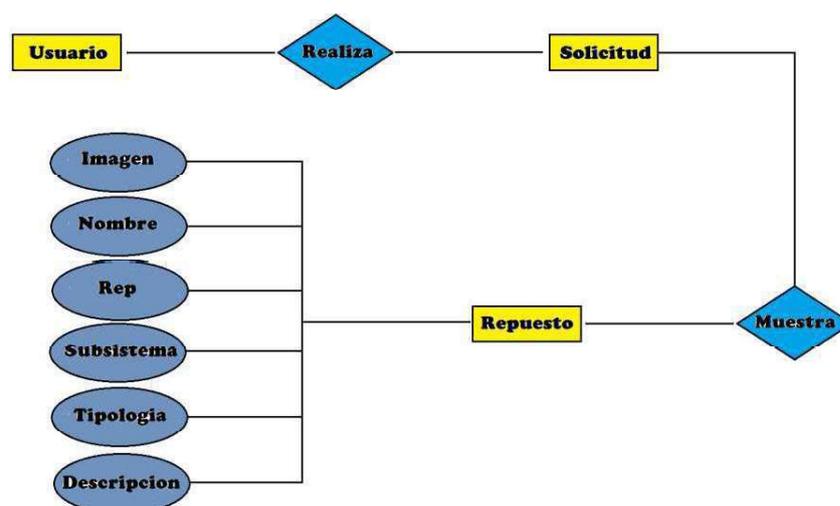
Una de las formas de diseñar bases de datos es teniendo el objetivo claro, la función que va a desarrollar esta en la empresa o en el proyecto donde se vaya a ejecutar. Es importante ser lo más explícito y claro posible ya que tener bastantes datos una base puede llegar hacer complicado.

Se cree que cuando una base de datos se encuentra bien diseñada, esta se explica sola, esto quiere decir que es fácil de entender, la idea es almacenar datos concretos y que vayan hacer de gran utilidad para los usuarios.

En este proyecto la base de datos es de bastante importancia ya que es la que va a contener toda la información que los usuarios vayan a solicitar, el objetivo de este apartado es la de diseñar y crear la base de datos para el almacenaje de la información de los repuestos.

### 2.4.1 Modelo entidad relación

Figura 2-9 Grafica identidad relación



Fuente: Imagen propia

Como se observa en la anterior imagen se muestra la entidad relación de la base de datos del proyecto. Como se puede ver, el usuario realiza la solicitud de un repuesto y en ella va a encontrar las características específicas de este, las cuales son Rep, Nombre, Subsistema, Tipología y descripción. La base de datos no tiene una cantidad específica de repuestos, ya que longitud depende de la cantidad de repuestos que se le encuentran a un automóvil.

### 2.4.2 Esquema de la base de datos

Tabla 2  
Esquema base de datos

Nombre	Rep	Subsistema	Tipología	descripcion	Imágenes

Fuente: Tabla propia

El esquema de la base de datos, se diseñó pensando en la facilidad de búsqueda de los repuestos con la aplicación móvil, la tabla está diseñada con 6 (seis) columnas donde cada una de ellas tiene una característica especial del repuesto. La columna donde se identifica con el Rep se diseñó como llave, esto significa que no se puede repetir un código dos veces, ya que va hacer el código único digital. Se tuvo en cuenta esta observación ya que puede existir un repuesto que sea suministrado por don proveedores diferentes el usuario podrá elegir retirar de almacén el repuesto de la marca que desee.

### 2.4.3 Base de datos en phpMyAdmin

El entorno de desarrollo seleccionado para este proyecto fue phpMyAdmin gracias a su fácil manejo e interfaz sencilla de comprender, es aquí donde se diseñará la

base de datos final, esta base de datos estará cargada en los servidores de phpMyAdmin su ingreso estará protegido por un usuario y contraseña además podrá ser modificado en cualquier momento e ilimitadamente, se podrán realizar actividades como eliminar, agregar o modificar repuestos.

Figura 2-10 Base de datos PhpMyAdmin

	rep	nombre	subsistema	tipologia	descripcion	imgResource
<input type="checkbox"/>	ELE11369	Baterías 4D	Electricidad	Todos	La batería de un coche es un acumulador que se enc...	bateria_4d.jpg
<input type="checkbox"/>	ELE12875	Cable # 16	ELECTRICIDAD	Todos	Es un tipo de cable muy delgado que se emplea en a...	cable_calibre_16.jpg
<input type="checkbox"/>	ELE13875	Réle 24v	Electricidad	Todos	Es un interruptor cuyo control corre por cuenta de...	rele_24v.JPG
<input type="checkbox"/>	ELE14698	Arranque Volvo	Electricidad	Volvo	funciona como receptor y consumidor de la corrient...	motor_arranque_volvo

Fuente: Imagen propia

Como la base de datos va a ser manipulada desde la web, para ingresar a esta se debe digitar la url <http://localhost/phpmyadmin/index.php?route=/server/databases>, seguido de seleccionar la base de datos.

La estructura de la base de datos se configuró de la siguiente manera:

Figura 2-11 Estructura base de datos PhpMyAdmin

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/> 1	rep	varchar(10)	utf8_spanish_ci		No	Ninguna			Cambiar  Eliminar  Más
<input type="checkbox"/> 2	nombre	varchar(30)	utf8_spanish_ci		No	Ninguna			Cambiar  Eliminar  Más
<input type="checkbox"/> 3	subsistema	varchar(30)	utf8_spanish_ci		No	Ninguna			Cambiar  Eliminar  Más
<input type="checkbox"/> 4	tipologia	varchar(30)	utf8_spanish_ci		No	Ninguna			Cambiar  Eliminar  Más
<input type="checkbox"/> 5	descripcion	varchar(250)	utf8_spanish_ci		No	Ninguna			Cambiar  Eliminar  Más
<input type="checkbox"/> 6	imgResource	text	utf8_spanish_ci		No	Ninguna			Cambiar  Eliminar  Más

Fuente: Imagen propia

Como se describió anteriormente la llave o el código digital único será el rep, es por eso que se ve el icono de la llave, todas las columnas de la base de datos se configuró para aceptar variable tipo varchar esto quiere decir que acepta tanto números tanto como letras, en el caso de descripción se creó un varchar (250) esto indica que puede recibir en la casilla hasta 250 caracteres. El resto de columnas solo reciben 30.

Como se puede observar en la imagen la columna de `imgResource` es donde se almacena la imagen, pero solo se lee el nombre, no se evidencia imágenes, eso es porque cuando se instala la aplicación Xampp esta crea una carpeta en la siguiente dirección.

Figura 2-12 Ruta de acceso imágenes para la aplicación



Fuente: Imagen propia

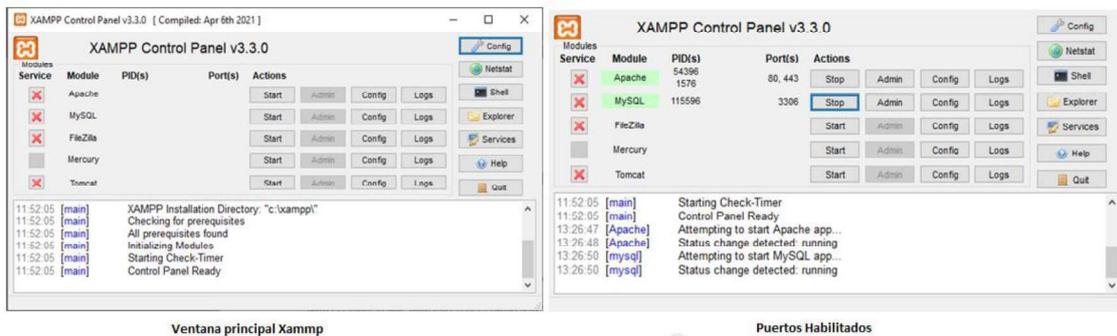
En esta carpeta se guardarán las imágenes que se van a mostrar en la aplicación, estas deben ser guardadas en formato `jpg`. Ya que, si se guarda en otro tipo de formato, la aplicación no reconocerá la imagen y no se mostrará en pantalla. En la base de datos se digitará el nombre tal cual como se guardó en esta carpeta. Es importante que termine con las letras `.jpg`.

## 2.5 Conectividad entre aplicación y la base de datos

### 2.5.1 Iniciar Xampp

Para poder realizar la conexión entre la base de datos y la aplicación se debe iniciar el programa Xampp, esta interfaz es la encargada de abrir los puertos para la conexión con los servidores de phpMyadmin y la aplicación. En la imagen se podrá observar la pantalla de inicio y cuando se habilitan los puertos, para el caso de este proyecto el puerto 80 será el encargado de conectarse con la aplicación móvil, y el puerto 3306 será la encargada de realizar la conexión con la base de datos en phpMyadmin.

Figura 2-13 Habilitación de puertos



Fuente: Imagen propia

Después de habilitar los puertos, lo único que se debe hacer es escribir la url antes mencionada para ingresar a la plataforma principal de phpMyadmin.

### 2.5.2 *Api-Rest*

La Api-Rest fue el método más seguro para el consumo de la base de datos, ya que solo recibirá la información de la de la plataforma phpMyadmin, y se conectará directamente con la aplicación, lo que impedirá la conexión directa con la base de datos.

Para poder desarrollar la api-rest fue necesario crear un código de programación en php donde se publicará la base de datos en formato json, así la aplicación podrá descargar los datos al móvil y podrá realizar las búsquedas que sean necesarias.

Figura 2-14 Código para publicación de Api-Rest

```

1  <?php
2  $con = mysqli_connect("localhost","wjmalacon53","90081975948","repuestos");
3  $response = array();
4  if ($con){
5      $sql = "select * from repuestos_almacen";
6      $result = mysqli_query($con,$sql);
7      if ($result) {
8          header("content-Type: JSON");
9          $i=0;
10         while($row = mysqli_fetch_assoc($result)){
11             $response[$i]['rep'] = $row['rep'];
12             $response[$i]['nombre'] = $row['nombre'];
13             $response[$i]['subsistema'] = $row['subsistema'];
14             $response[$i]['tipologia'] = $row['tipologia'];
15             $response[$i]['descripcion'] = $row['descripcion'];
16             $response[$i]['imgResource'] = $row['imgResource'];
17             $i++;
18         }
19         echo json_encode($response,JSON_PRETTY_PRINT);
20     }
21 }
22 else{
23     echo "fallo conexion base de datos";
24 }
25 ?>

```

Fuente: Imagen propia

El código se ejecutará una única vez ya que se abrirá la base de datos en la web, y se guardará la url que será donde se podrá observar la api-rest, para poder observar la api se deberá ingresar a la siguiente dirección de internet.

Figura 2-15 Url Api-Rest



The image shows a browser address bar with the URL `localhost/api.php`. To the left of the URL is a circular icon containing an information symbol (an 'i'). The address bar has a light blue background.

Fuente: Imagen propia

### 3. Resultados

#### 3.1 Aplicación

Como resultado final se obtuvo un archivo apk listo para estar disponible en la Play store de google y poder ser distribuido de manera masiva, para la instalación de la aplicación móvil para métodos de ensayo se debe conceder un permiso el cual deja instalar directamente sin afectar el funcionamiento de la aplicación.

El usuario podrá buscar el repuesto que desea instalar en el vehículo, esta aplicación mostrara información importante como el código en el cual se encuentra almacenado, el subsistema al cual hace parte, y una breve explicación de la función principal de estos repuestos.

Esta aplicación ahorra un importante tiempo para la salida de los repuestos de almacén, ya que no tendrán que volver a buscar el código en grandes listas que se encuentran generalmente en el almacén el tiempo de búsqueda es casi de inmediato, además de identificar el repuesto fácilmente por medio de imágenes que tiene la aplicación.

##### 3.1.1 Logo aplicación

A la hora de diseñar un programa ya sea para computadoras o móviles inclusive hasta crear grandes empresas, es fundamental el logo el nombre o el símbolo que los va a identificar y reconocer en cualquier parte del mundo donde sean vistos.

El logo de esta aplicación fue diseñado y pensado en forma de abreviar su nombre (Búsqueda de repuestos para vehículos), ya que es una aplicación diseñada para obtener el título de ingeniero electrónico de la universidad Antonio Nariño se decidió utilizar los mismos colores que representan esta institución, además de por derecho agregar el logo de la universidad antes mencionada,

Como resultado final se obtuvo la siguiente imagen.

Figura 3-1 Logo Aplicación



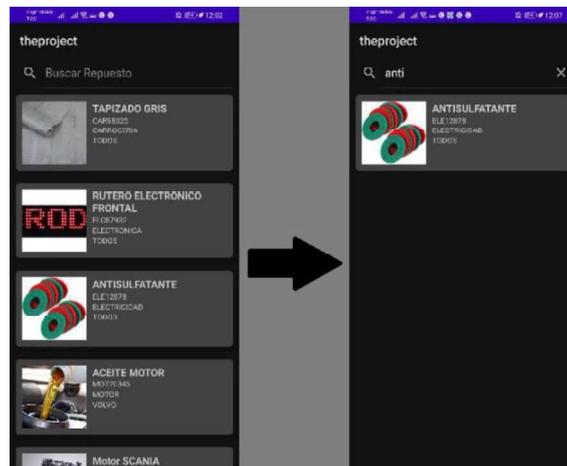
Fuente: Imagen propia

### **3.1.1 Interfaz final**

La aplicación móvil desarrollada en este proyecto consta de 2 pantallas, la primera que es la que se muestra cuando se inicia la aplicación. Esta pantalla mostrará toda la base de datos que tenemos disponible, esta será mostrada en forma de lista y recuadros, cada recuadro muestra la imagen, el nombre, rep, subsistema y tipología del móvil al que se puede instalar el repuesto. Se podrá buscar un repuesto simplemente bajando la lista y observando el repuesto, si desea realizar una búsqueda más rápida solo basta con darle clic en el recuadro que se encuentra en la parte superior de la pantalla, se inserta el nombre y la aplicación automáticamente empezará a realizar el filtrado y mostrar repuestos los cuales coincidan con el nombre que se esta introduciendo.

A continuación, se mostrará la interfaz final la cual aparecerá cuando se inicia la aplicación.

Figura 3-2 Interfaz final aplicación



Fuente: Imagen propia

Para ingresar a la segunda pantalla solo se debe dar clic en el repuesto que deseamos ver, esta segunda pantalla muestra la información de la primera pantalla más una leve descripción del repuesto. Acá se puede encontrar características más concretas del repuesto, como para que funciona, como se debe instalar o cuidados para su instalación.

Figura 3-3 Interfaz final segunda pantalla



Fuente: Imagen propia

### **3.1.2 Pruebas aplicación**

Las pruebas de la aplicación se realizaron en diferentes dispositivos móviles con sistema operativo Android, los celulares a los cuales se instaló fueron.

Huawei P8

Huawei Y9 Prime

Samsung A3

Motorola G9

Xiaomi Note 8

En las pruebas se pudo observar que el tiempo de descarga de la base de datos al móvil depende de la velocidad de internet que tenga el dispositivo. Cuando la velocidad o los datos de internet del celular son bastante lentos, la aplicación se demora 5 segundos aproximadamente descargando la información y mostrándola en pantalla.

Cuando la velocidad de internet era la adecuada la descarga era casi de manera inmediata mostrando la información antes de los 2 segundos, en el caso de la búsqueda de los repuestos, todos los móviles funcionaron correctamente la búsqueda era inmediata y ningún móvil tuvo errores. En cuanto a la segunda pantalla todas funcionaron correctamente.

Anexo se encontrará el manual de usuario para el manejo de la aplicación donde se especificará por medio de imágenes el método adecuado de manipular la aplicación. Adicional el manual de manejo de la base de datos donde se explica la manera la cual el cliente puede ingresar, eliminar y o modificar repuestos en la plataforma PhpMyAdmin y la publicación de la Api-Rest para el consumo de datos por medio de la aplicación.

### **3.2 Encuesta de satisfacción**

Se tuvo la oportunidad de realizar pruebas de la aplicación en una empresa en la ciudad de Bogotá, esta empresa está dedicada al servicio de transporte de personas, actualmente cuenta con aproximadamente 1200 automotores para realizar sus actividades, la funcionalidad de la aplicación se hizo en una de las flotillas que tiene con aproximadamente 200 vehículos padrones.

Se buscaron 10 personas las cuales se desempeñan en diferentes áreas pero que en algún momento de sus actividades laborales necesiten de un repuesto de forma rápida y tengan la oportunidad de utilizar esta aplicación.

Luego de las pruebas se les compartió una pequeña encuesta virtual para ver el nivel de satisfacción que tuvieron al utilizar la aplicación.

La encuesta consta de 7 preguntas las cuales se compartirán a continuación.

1. Nombre completo
2. Cargo
3. ¿Tuvo inconvenientes a la hora de instalar la aplicación?
4. ¿La aplicación proporciono la información que necesitaba?
5. ¿La interfaz de la aplicación la considera adecuada?
6. ¿El tiempo de respuesta a las solicitudes es el adecuado?
7. ¿Qué le cambiara o agregaría a la aplicación?

- **Nombre completo – Cargo**

Tabla 3 Personal que realizo la encuesta

Técnico	Cargo
Yilmar Barona	Mecánico B
Luis Muñoz	Auxiliar Administrativo
Felipe Bohórquez	Mecánico
Luis Duran	Eléctrico B
Andrés Bernal	Supervisor
Víctor Gómez	Eléctrico C
Oscar Aguirre	Jefe de Taller
Nelson Sánchez	Lubricador
Javier Cocca	Almacenista

Fuente: Imagen propia

Para la ejecución de la encuesta se tomó un 33% de los trabajadores que intervienen en el mantenimiento de esta flota. Se eligió personal de diferentes cargos esto con el fin de tener un mayor rango de alcance de la funcionalidad de la aplicación. Esta encuesta fue realizada por personal de mantenimiento y administrativo que en algún momento de sus actividades pueden requerir de esta aplicación para agilizar sus trabajos.

- **¿Tuvo inconvenientes a la hora de instalar la aplicación?**

Figura 3-4 Grafica pregunta 3



Fuente: Imagen propia

A la pregunta 3 se observó que en general no hubo inconvenientes a la hora de la instalación de la aplicación, solo una advertencia que emite el móvil por temas de seguridad ya que la aplicación no fue descargada directamente de la plataforma oficial de Android Studio (Play Store). Esta advertencia se muestra cuando se va a instalar una app que fue descargada de sitios extraoficiales, para este caso la apk (instalador) se generó desde una computadora portátil y no fue cargada a la Play Store.

- **¿La aplicación proporciono la información que necesitaba?**

Figura 3-5 Grafica pregunta 4

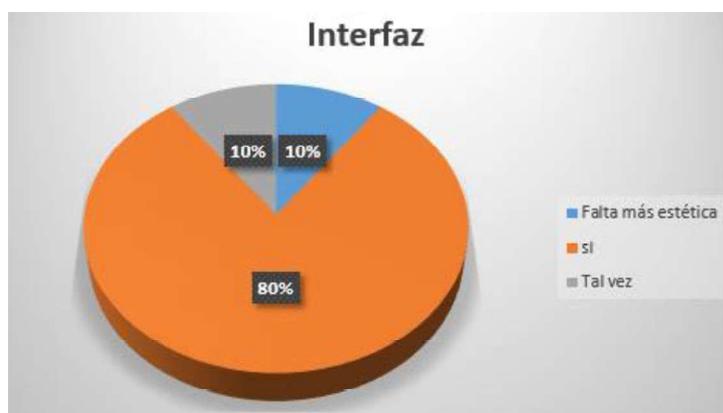


Fuente: Imagen propia

A la pregunta la 4 'de la encuesta se evidencio una satisfacción del 100% ya que para las personas que realizaron la encuesta en el momento solo requerían del código del repuesto para poder realizar la solicitud en almacén y así retirar el repuesto para su instalación.

- **¿La interfaz de la aplicación la considera adecuada?**

Figura 3-6 Grafica pregunta 5



Fuente: Imagen propia

A la hora de diseñar aplicaciones móviles es importante tener en cuenta la interfaz, a qué tipo de usuario se enfocará la aplicación, visualmente como como lo haría, y sobre todo que la funcionalidad sea lo más didáctica posible. Esta interfaz está diseñada para momentos de estrés donde se pueda obtener una información de forma rápida y sencilla.

La mayoría de los encuestados quedaron satisfechos con la interfaz ya que revelaron que la aplicación se manejaba de forma muy sencilla y práctica lo que les ayudó a minimizar los tiempos de retiro de repuestos de almacén, sobre todo cuando la solicitud requería más de 5 de estos.

Como la aplicación es una prueba de concepto se buscó que fue sencilla posible pero la falta de estética se tendrá en cuenta para realizar las mejoras necesarias para poder ser aplicada a grandes empresas.

- **¿El tiempo de respuesta a las solicitudes es el adecuado?**

Figura 3-7 Gráfica pregunta 6



Fuente: Imagen propia

Para el tiempo de descarga de la base de datos al dispositivo móvil por medio de la aplicación, los encuestados en un rango general estuvieron satisfechos por el tiempo de descarga, el 20% consideraron que tal vez fue buena pero que podría mejorar, y un 10% consideraron que la descarga de la información fue muy lenta. Esto es debido y como ha comentado anteriormente, depende de la velocidad del

internet que tenga el dispositivo. La única forma de mejorar la descarga es teniendo buen acceso a internet.

- **¿Qué le cambiara o agregaría a la aplicación?**

Figura 3-8 Grafica pregunta 7

Que le cambiara o agregaría a la aplicación	cantidad
Acceso a inpac	1
Agregaría acceso al inpac	1
Agregaría más información del repuesto	1
Cambiaría el logo	1
Crearía un filtro por patio	1
Nada	2
No, la aplicación es fácil de manejar y muy util	1
Poder filtrar por subsistema	1
Que se pueda filtrar por cualquier opción	1
<b>Total general</b>	<b>10</b>

Fuente: Imagen propia

Al realizar la encuesta a una población en específica, las recomendaciones de modificaciones para esta fueron encaminadas a ser utilizadas en este patio.

Las grandes concesionarias como lo son IZUZU, VOLVO, SCANIA cuentan con plataformas bastante robustas donde los clientes o dueños de flotas pueden acceder a ella para buscar información específica de alguna parte de los móviles. Para el caso de VOLVO cuenta con IMPAC una plataforma que ayuda a buscar información más detallada de sus vehículos es por eso que una de las observaciones fue el tener acceso a ella, de este modo el personal técnico tendría mucha más información del repuesto que estén indagando.

Otra de las observaciones es la de poder realizar búsqueda no solo por el nombre si no por el subsistema al cual pertenezca el repuesto, o el de filtrar la tipología del vehículo esta observación se realiza ya que en ocasiones los técnicos no recuerdan el nombre con el cual se encuentra el repuesto en la base de datos. Así ellos tendrán diferentes maneras de buscar el repuesto.

Para el caso en particular de esta empresa una modificación que se podría hacer es la de caracterizar por patio ya que cada patio maneja una tipología diferente de móviles, esto implicaría que se tendrían que diseñar una base de datos para cada patio, así el técnico descargaría la información del patio que desee.

Esta encuesta ayudo a entender que a pesar de que la aplicación cumple con las necesidades básicas para buscar repuestos, se pueden realizar mejoras que brinden una mejor relación entre el usuario y la aplicación, también se pudo concluir que la app tiene grandes posibilidades de ser comercializada, así como ser aplicada en varios campos donde se manejen bases de datos y se requiera información de forma rápida.

### 3.3 Base de datos

Para lograr desarrollar este proyecto era necesario crear una base de datos la cual contaría con la información necesaria para mostrarla en la aplicación, esta base de datos está compuesta por 40 repuestos los cuales están clasificados por subsistemas (Electricidad, Carrocería, Motor, Refrigeración, Neumática, Llantas, Dirección, Suspensión). Ver anexo

#### 3.3.1 Base de datos phpMyadmin

Luego de haber creado la base de datos en un archivo Excel, se cargó toda la información en la base de datos de phpMyadmin es aquí donde se modificará toda la información de algún repuesto que se vaya a anexas o se tenga que eliminar. Es desde este punto que la Api-rest obtendrá toda la información necesaria para publicar la información y así poder descargarse de los dispositivos móviles. A continuación, se observará la plataforma principal con toda la información cargada.

Figura 3-9 Base de datos plataforma PhpMyAdmin

Mostrando filas 0 - 3 (total de 4.1 a consulta tardó 0.0011 segundos) [rep: FI F11369... FI F14698...]

```
SELECT * FROM `repuestos_almacen` ORDER BY `rep` ASC
```

Perfilando [ Editar en línea ] [ Editar ] [ Explicar SQL ] [ Crear código PHP ] [ Actualizar ]

Mostrar todo | Número de filas: 50 | Filtrar filas: Buscar en esta tabla | Sort by key: PRIMARY (ASC)

rep	nombre	subsistema	tipologia	descripción	imgResource
ELE11369	Baterías 4D	Electricidad	Todos	La batería de un coche es un acumulador que se enc...	bateria_4d.jpg
ELE12876	Cable #. 16	ELECTRICIDAD	Todos	Es un tipo de cable muy delgado que se emplea en a...	cable_cable_16.jpg
ELE13875	Rele 24v	Electricidad	Todos	Es un interruptor cuyo control corre por cuenta de ...	rele_24v.JPG
ELE14698	Arranque Volvo	Electricidad	Volvo	funciona como receptor y consumidor de la corrient...	motor_arranque_volvo

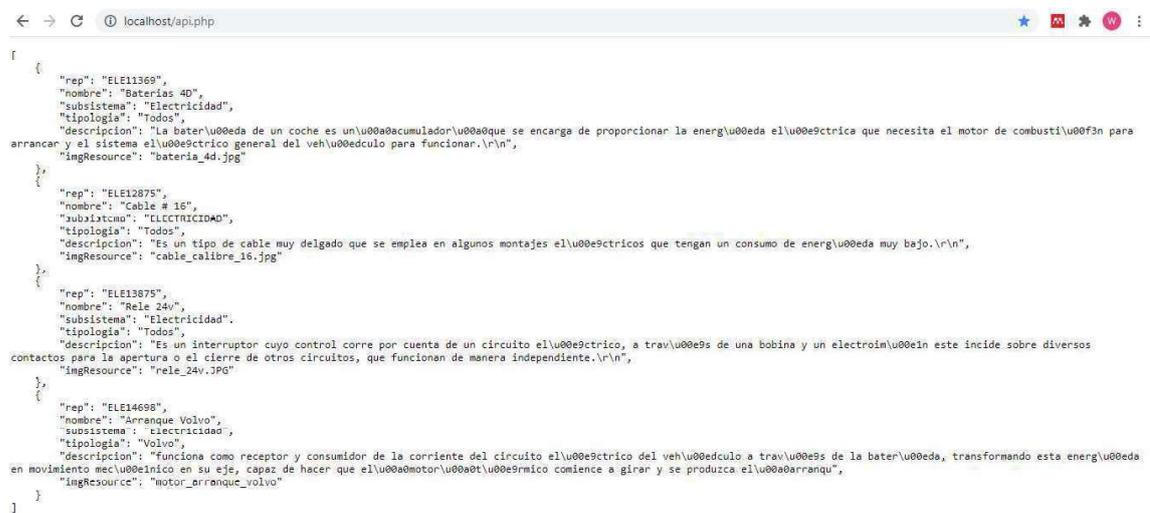
Operaciones sobre los resultados de la consulta

Imprimir | Copiar al portapapeles | Exportar | Mostrar gráfico | Crear vista

### 3.3.2 Api-rest

Por ultimo podemos observar la Api-rest publicada en la red lista para el consumo de la información por medio de los celulares móviles. Es importante tener presente que si la información es modificada en la plataforma de la base de datos se deberá actualizar la api-rest para que se vean reflejados los cambios, así cuando se abra la aplicación móvil tendrá toda la información actualizada. A continuación, se observará la api-rest publicada en la red.

Figura 3-10 Api-Rest



```
[
  {
    "rep": "ELE11369",
    "nombre": "Baterías 40",
    "subsistema": "Electricidad",
    "tipologia": "Todos",
    "descripcion": "La bater\u00eda de un coche es un\u00a0acumulador\u00a0que se encarga de proporcionar la energ\u00eda el\u00e9ctrica que necesita el motor de combusti\u00f3n para arrancar y el sistema el\u00e9ctrico general del veh\u00e9culo para funcionar.\r\n",
    "imgResource": "bateria_4d.jpg"
  },
  {
    "rep": "ELE12875",
    "nombre": "Cable # 16",
    "subsistema": "ELECTRICIDAD",
    "tipologia": "Todos",
    "descripcion": "Es un tipo de cable muy delgado que se emplea en algunos montajes el\u00e9ctricos que tengan un consumo de energ\u00eda muy bajo.\r\n",
    "imgResource": "cable_calibre_16.jpg"
  },
  {
    "rep": "ELE13875",
    "nombre": "Rele 24v",
    "subsistema": "Electricidad",
    "tipologia": "Todos",
    "descripcion": "Es un interruptor cuyo control corre por cuenta de un circuito el\u00e9ctrico, a trav\u00e9s de una bobina y un electroim\u00e1n este incide sobre diversos contactos para la apertura o el cierre de otros circuitos, que funcionan de manera independiente.\r\n",
    "imgResource": "rele_24v.JPG"
  },
  {
    "rep": "ELE14698",
    "nombre": "Arranque Volvo",
    "subsistema": "Electricidad",
    "tipologia": "Volvo",
    "descripcion": "Funciona como receptor y consumidor de la corriente del circuito el\u00e9ctrico del veh\u00e9culo a trav\u00e9s de la bater\u00eda, transformando esta energ\u00eda en movimiento mec\u00e1nico en su eje, capaz de hacer que el\u00a0motor\u00a0\u00e9rmino comience a girar y se produzca el\u00a0arranque",
    "imgResource": "motor_arranque_volvo"
  }
]
```

Fuente: Imagen propia

## Conclusiones

Al concluir este proceso se pudo obtener una aplicación móvil para sistema operativo android en el cual cuenta con una gran integridad ya que los datos que se manejan en las bases de datos no pueden ser modificados por los usuarios que hagan uso de esta , esto gracias a la Api-Rest que se diseño para separar la base de datos de la aplicación, la seguridad que nos brinda es por medio de protocolo HTTP esto quiere decir que los usuarios no van a tener contacto directo con la plataforma que contiene la base de datos asi evitando la manipulacion y o modificacion de datos.

Al ser aplicado a una poblacion la cual diariamente requieren sacar repuestos de grandes almacenes la usabilidad de esta aplicación es bastante alta lo que genera una gran posibilidad de escalabilidad a nivel comercial, siempre y cuando se realicen las modificaciones adecuadas.

Gracias a que no es tan robusta la portabilidad de la misma es minima apenas con 70mb aproximadamente, lo que hace que su portabilidad no genere procesos lentos en los equipos en los cuales vaya hacer instalados, para su mantenimiento es importante tener acceso a la plataforma phpMyAdmin ya que es desde aquí en donde se modificaran agregaran y o eleminaa informacion.

Anexo a este trabajo se encuentra la documentacion necesaria para el manejo de la aplicación y el ingreso a la plataforma donde se encuentra la base de datos y como manejar la plataforma de manera segura y sencilla sin correr riesgos de eliminar la informacion.

## Anexos

### Anexo 1. Detail Activity

```

package com.malagon.theproject;
import android.os.Bundle;
import android.widget.ImageView;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;
public class DetailActivity extends AppCompatActivity {
    private ImageView imgItemDetail;
    private TextView tvNombreDetail;
    private TextView tvRepDetail;
    private TextView tvSubsistemaDetail;
    private TextView tvTipologiaDetail;
    private TextView tvDescripcionDetail;
    private ItemList itemDetail;
    @Override
    protected void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detail);
        setTitle(getClass().getSimpleName());
        initView();
        initValues();
    }
    private void initView(){
        imgItemDetail = findViewById(R.id.imgItemDetail);
        tvNombreDetail = findViewById(R.id.tvNombreDetail);
        tvRepDetail = findViewById(R.id.tvRepDetail);
        tvSubsistemaDetail = findViewById(R.id.tvSubsistemaDetail);
        tvTipologiaDetail = findViewById(R.id.tvTipologiaDetail);
        tvDescripcionDetail = findViewById(R.id.tvDescripcionDetail);
    }
    private void initValues(){
        itemDetail = (ItemList) getIntent().getExtras().getSerializable("itemDetail");
        //imgItemDetail.setImageResource(itemDetail.getImgResource());
        tvNombreDetail.setText(itemDetail.getNombre());
        tvRepDetail.setText(itemDetail.getRep());
        tvSubsistemaDetail.setText(itemDetail.getSubsistema());
        tvTipologiaDetail.setText(itemDetail.getTipologia());
        tvDescripcionDetail.setText(itemDetail.getDescripcion());
    }
}

```

## B Anexo 2 Item List

```

package com.malagon.theproject;
import android.content.Intent;
import android.os.Bundle;
import android.widget.Toast;
import java.util.List;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.SearchView;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class MainActivity extends AppCompatActivity implements
RecyclerView.Adapter.OnItemClickListener, SearchView.OnQueryTextListener {
    private RecyclerView rvLista;
    private SearchView svSearch;
    private RecyclerView.Adapter adapter;
    private List<ItemList> items;

    private RetrofitApiService retrofitApiService;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        initView();
        initValues();
        initListener();
    }

    private void initView(){

        rvLista = findViewById(R.id.rvLista);
        svSearch = findViewById(R.id.svSearch);
    }

    private void initValues(){
        retrofitApiService = RetrofitClient.getApiservice();

```

```

LinearLayoutManager manager = new LinearLayoutManager(this);
rvLista.setLayoutManager(manager);

getItemsSQL();
}

private void initListener(){
    svSearch.setOnQueryTextListener(this);
}

private void getItemsSQL(){
    retrofitApiService.getItemsDB().enqueue(new Callback<List<ItemList>>() {
        @Override
        public void onResponse(Call<List<ItemList>> call,
Response<List<ItemList>> response) {
            items = response.body();
            adapter = new RecyclerViewAdapter(items, MainActivity.this);
            rvLista.setAdapter(adapter);
        }

        @Override
        public void onFailure(Call<List<ItemList>> call, Throwable t) {
            Toast.makeText(MainActivity.this, "Error "+t.getMessage(),
Toast.LENGTH_SHORT).show();
        }
    });
}

@Override
public void onItemClick(ItemList item) {
    Intent intent = new Intent(this, DetailActivity.class);
    intent.putExtra("itemDetail", item);
    startActivity(intent);
}

@Override
public boolean onQueryTextSubmit(String query) {
    return false;
}

@Override
public boolean onQueryTextChange(String newText) {
    adapter.filter(newText);
    return false;
}
}

```

### C Anexo 3 Activity Main

```

package com.malagon.theproject;
import android.content.Intent;
import android.os.Bundle;
import android.widget.Toast;
import java.util.List;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.SearchView;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class MainActivity extends AppCompatActivity implements
RecyclerView.Adapter.OnItemClickListener, SearchView.OnQueryTextListener {
    private RecyclerView rvLista;
    private SearchView svSearch;
    private RecyclerView.Adapter adapter;
    private List<ItemList> items;
    private RetrofitApiService retrofitApiService;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        initView();
        initValues();
        initListener();
    }

    private void initView(){

        rvLista = findViewById(R.id.rvLista);
        svSearch = findViewById(R.id.svSearch);
    }

    private void initValues(){
        retrofitApiService = RetrofitClient.getApiservice();

        LinearLayoutManager manager = new LinearLayoutManager(this);

```

```

        rvLista.setLayoutManager(manager);

        getItemsSQL();
    }

    private void initListener(){
        svSearch.setOnQueryTextListener(this);
    }

    private void getItemsSQL(){
        retrofitApiService.getItemsDB().enqueue(new Callback<List<ItemList>>() {
            @Override
            public void onResponse(Call<List<ItemList>> call,
Response<List<ItemList>> response) {
                items = response.body();
                adapter = new RecyclerViewAdapter(items, MainActivity.this);
                rvLista.setAdapter(adapter);
            }

            @Override
            public void onFailure(Call<List<ItemList>> call, Throwable t) {
                Toast.makeText(MainActivity.this, "Error "+t.getMessage(),
Toast.LENGTH_SHORT).show();
            }
        });
    }

    @Override
    public void onItemClick(ItemList item) {
        Intent intent = new Intent(this, DetailActivity.class);
        intent.putExtra("itemDetail", item);
        startActivity(intent);
    }

    @Override
    public boolean onQueryTextSubmit(String query) {
        return false;
    }

    @Override
    public boolean onQueryTextChange(String newText) {
        adapter.filter(newText);
        return false;
    }
}

```

## D Anexo 4 Recycler Adapter

```

package com.malagon.theproject;

import android.os.Build;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import com.squareup.picasso.Picasso;

import java.util.ArrayList;
import java.util.List;
import java.util.stream.Collectors;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

public class RecyclerViewAdapter extends
RecyclerView.Adapter<RecyclerViewAdapter.RecyclerViewHolder> {
    private List<ItemList> items;
    private List<ItemList> originalItems;
    private RecyclerViewItemClickListener itemClick;

    private String doaming_image = "http://192.168.0.9:80/drawable/";

    public RecyclerViewAdapter(List<ItemList> items, RecyclerViewItemClickListener itemClick) {
        this.items = items;
        this.itemClick = itemClick;
        this.originalItems = new ArrayList<>();
        originalItems.addAll(items);
    }

    @NonNull
    @Override
    public RecyclerViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int
viewType) {
        View view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_list_view,parent,fals
e);

```

```

    return new RecyclerViewHolder(view);
}

@Override
public void onBindViewHolder(@NonNull final RecyclerViewHolder holder, final int
position) {
    final ItemList item = items.get(position);
    //holder.imgItem.setImageResource(item.getImgResource());
    Picasso.get()
        .load(doaming_image+item.getImgResource())
        .into(holder.imgItem);
    holder.tvNombre.setText(item.getNombre());
    holder.tvRep.setText(item.getRep());
    holder.tvSubsistema.setText(item.getSubsistema());
    holder.tvTipologia.setText(item.getTipologia());
    holder.itemView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            itemClick.itemClick(item);
        }
    });
}

@Override
public int getItemCount() {

    return items.size();
}

public void filter(String strSearch){
    if(strSearch.length() == 0){
        items.clear();
        items.addAll(originalItems);
    }
    else {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.N) {
            items.clear();
            List<ItemList> collet = originalItems.stream()
                .filter(i -> i.getNombre().toLowerCase().contains(strSearch))
                .collect(Collectors.toList());
            items.addAll(collet);
        }
        else {
            items.clear();
            for (ItemList i : originalItems){

```

```

        if (i.getNombre().toLowerCase().contains(strSearch)){
            items.add(i);
        }
    }
}
notifyDataSetChanged();
}

```

```

public class RecyclerViewHolder extends RecyclerView.ViewHolder{
    private ImageView imgItem;
    private TextView tvNombre;
    private TextView tvRep;
    private TextView tvSubsistema;
    private TextView tvTipologia;

```

```

    public RecyclerViewHolder(@NonNull View itemView_1) {
        super(itemView_1);
        imgItem = itemView.findViewById(R.id.imgItem);
        tvNombre = itemView.findViewById(R.id.tvNombre);
        tvRep = itemView.findViewById(R.id.tvRep);
        tvSubsistema = itemView.findViewById(R.id.tvSubsistema);
        tvTipologia = itemView.findViewById(R.id.tvTipologia);
    }
}

```

```

public interface RecyclerViewItemClickListener {
    void onItemClick(ItemList item);
}
}

```

## E Anexo 5 Código de interfaz final

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="10dp"
    tools:context=".DetailActivity">

    <ImageView
        android:id="@+id/imgItemDetail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:contentDescription="TODO"
        app:layout_constraintTop_toTopOf="parent"
        tools:layout_editor_absoluteX="10dp" />

    <TextView
        android:id="@+id/tvNombreDetail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="30sp"
        android:textStyle="bold"
        app:layout_constraintLeft_toLeftOf="parent"
```

```
app:layout_constraintRight_toRightOf="parent"  
app:layout_constraintTop_toBottomOf="@id/imgItemDetail" />
```

```
<TextView  
    android:id="@+id/tvRepDetail"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:textSize="20sp"  
    app:layout_constraintLeft_toLeftOf="parent"  
    app:layout_constraintRight_toRightOf="parent"  
    app:layout_constraintTop_toBottomOf="@id/tvNombreDetail" />
```

```
<TextView  
    android:id="@+id/tvSubsistemaDetail"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:textSize="20sp"  
    app:layout_constraintLeft_toLeftOf="parent"  
    app:layout_constraintRight_toRightOf="parent"  
    app:layout_constraintTop_toBottomOf="@id/tvRepDetail" />
```

```
<TextView  
    android:id="@+id/tvTipologiaDetail"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:textSize="20sp"  
    app:layout_constraintLeft_toLeftOf="parent"
```

```
app:layout_constraintRight_toRightOf="parent"  
app:layout_constraintTop_toBottomOf="@id/tvSubsistemaDetail" />
```

```
<TextView
```

```
    android:id="@+id/tvDescripcionDetail"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:textSize="20sp"  
    app:layout_constraintLeft_toLeftOf="parent"  
    app:layout_constraintRight_toRightOf="parent"  
    app:layout_constraintTop_toBottomOf="@id/tvTipologiaDetail" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

## E Anexo 5 Manual manejo aplicación movil

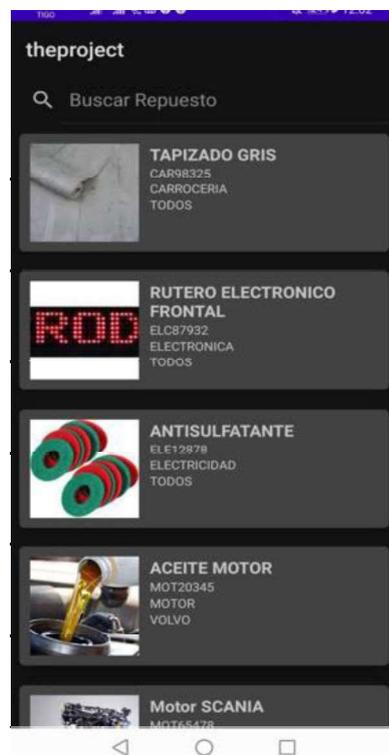
La aplicación búsqueda de repuestos para vehículos es una aplicación de manejo fácil y dinámico ya que la misión de esta es poder brindar información al usuario de forma rápida y sencilla.

- **Buscar repuestos**

Para realizar una búsqueda en la aplicación lo primero que se debe realizar es abrir la app, esta automáticamente descargara la base de datos en el móvil, el tiempo de descarga dependerá de la velocidad del internet con la que cuente el usuario.

la pantalla principal de la aplicación se vera de la siguiente manera.

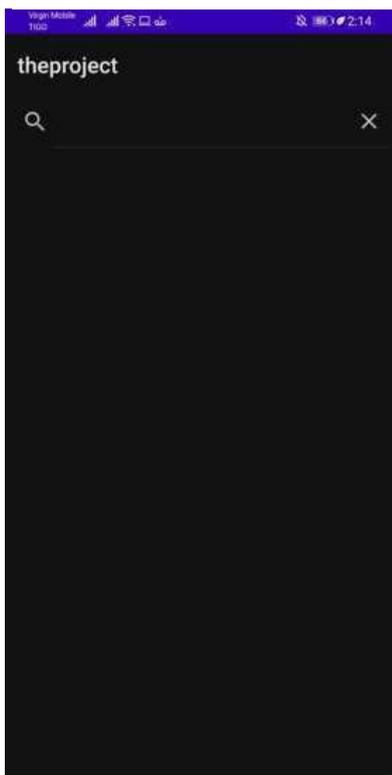
Figura E- 1 Pantalla principal App



Fuente: Imagen propia

Si el usuario no tiene acceso a internet o la red esta demaciado lenta la aplicación no mostrara ninguna informacion. Esto quiere decir que solo se vera el icono de busqueda de repuesto como se puede observar a continuacion.

Figura E- 2 Error de descarga

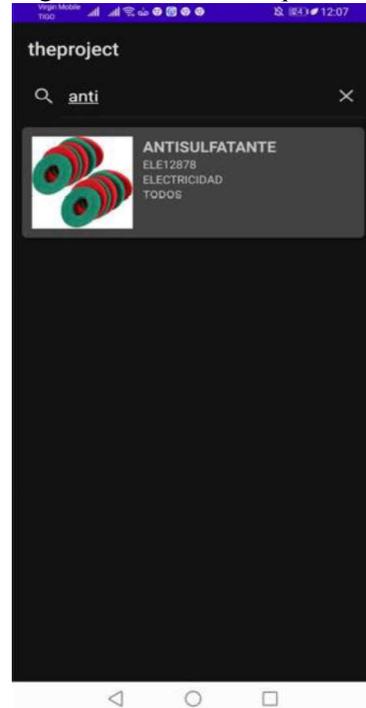


Fuente: Imagen propia

Esta misma imagen se mostrara cuando el usuario ingrese el nombre de un repuesto que no se encuentre en la base de datos esta es la forma en la cual el usuario se da por enterado que el repuesto no existe en la base de datos.

Para realizar la busqueda solo tiene que empezar a digitar el nombre en el recuadro de busqueda, la aplicación se encuentra diseñada para que empiece a filtrar desde que se ingrese el primer carácter, estos quiere decir que si el usuario digita la w, la aplicación mostrara todos los repuestos que en su nombre contengan esta letra, a medida que va ingresando mas caracteres el filtro sera mas exacto. Si el repuesto se encuentra en la base de datos se mostrara la imagen el nombre el codigo con el cual puede ser solicitado, a que subsistema depende y para que tipo de movil aplica.

Figura E- 3 Buscar repuesto



Fuente: Imagen propia

Si el usuario desea ver mas informacion del repuesto solo basta con dar clic sobre la imagen del repuesto y se abriara una nueva ventana donde se mostrara la informacion de la pagina principal mas una descripcion o caracteristica breve del repuesto, esta informacion puede ser util para repuestos los cuales tengan alguna configuracion especial a la hora de ser instalado en el movil.

Si se desea buscar un nuevo repuesto se da clic al boton de atrás, en la casilla de busqueda se elimina la informacion y se digita el nuevo nombre del repuesto la aplicación empezara a realizar un nuevo filtro y a mostrar el nuevo repuesto que se esta buscando.

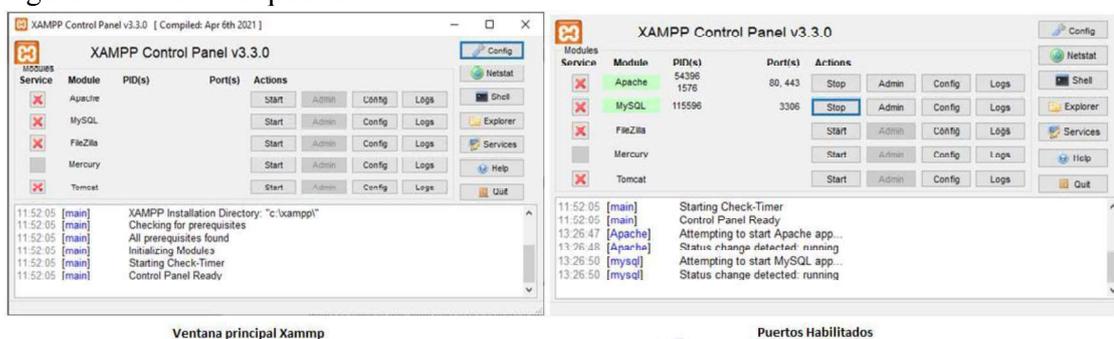
## F Anexo 5 Manual manejo base de datos

La base de datos para este proyecto se encuentra diseñada y ejecutada en los servidores de PhpMyadmin, la cual tiene la intención de manejar y administrar bases de datos de MySQL a través de un servidor en la red.

Para poder acceder a estos servidores es esencial habilitar algunos puertos del computador para esto se requiere de Xampp este software de uso lo cual la función principal es la de habilitar estos puertos para facilitar la conexión con los servidores y la aplicación.

Para iniciar Xampp se abre la aplicación en el computador el cual va hacer el servidor local se mostrara la pantalla principal de Xampp se habilitan los puertos de Apache y MySQL, solo dando clic sobre los botones Start que se encuentran frente a cada uno.

Figura F- 1 Habilitar puertos



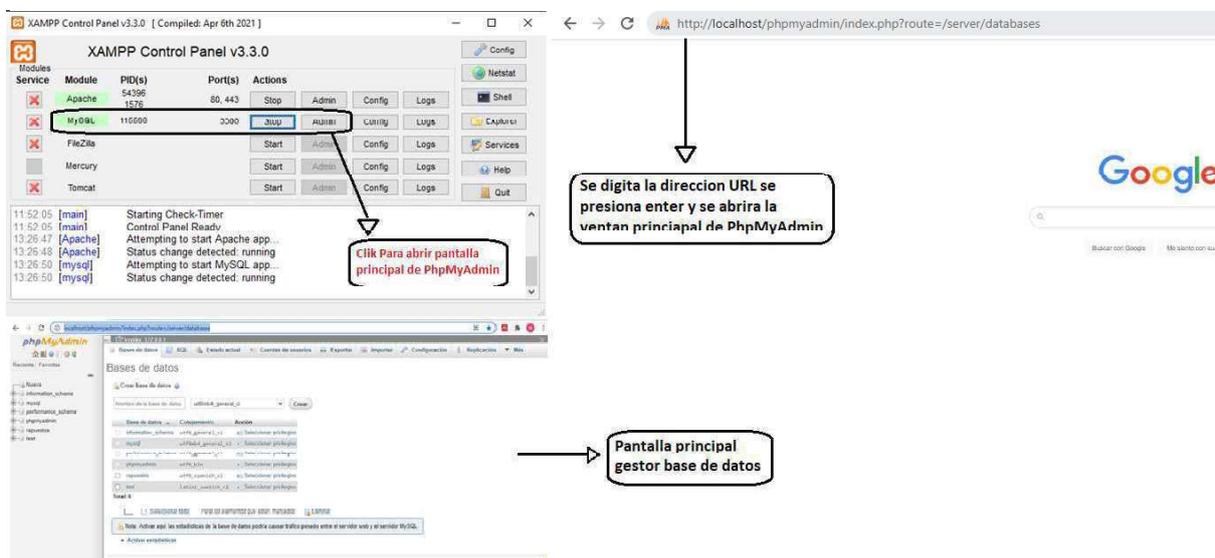
Fuente: Imagen propia

PhpMyAdmin es uno de los gestores de bases de datos más utilizados en la actualidad, se eligió para este proyecto ya que su manejo es fácil y práctico, además de su posible conexión para el diseño de la API – Rest. Existen dos

formas de acceder a la plataforma de PhpMyAdmin, una es dando clic en el icono de Admin frente al módulo de MySQL en la ventana principal de la plataforma Xampp seguido se abrirá una ventana de internet en donde se mostrará la pantalla principal del gestor de base de datos. el otro método es digitando el siguiente enlace en el buscador de internet donde se abrirá igualmente la pantalla principal.

<http://localhost/phpmyadmin/index.php?route=/server/databases>.

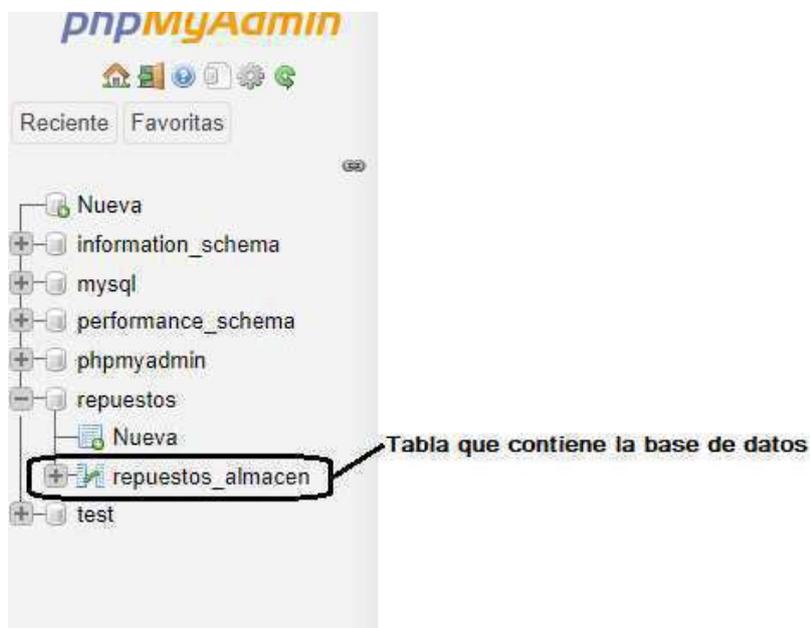
Figura F- 2 Acceso a plataforma PhpMyAdmin



Fuente: Propia

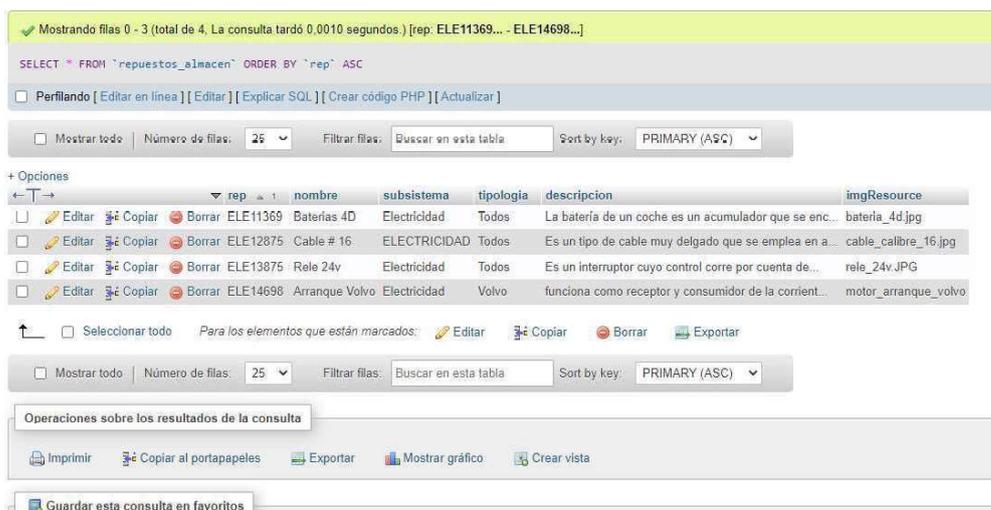
Luego de acceder a la pantalla principal de PhpMyadmin se selecciona en la parte izquierda la base de datos llamada repuestos se desplegara una lista de tablas que contiene la base de datos, para este proyecto solo se encontrara una tabla llamada repuestos\_almacen se selecciona y se mostrara la base de datos.

Figura F- 3 Selección tabla base de datos



Fuente: Imagen propia

Figura F- 4 Base de datos

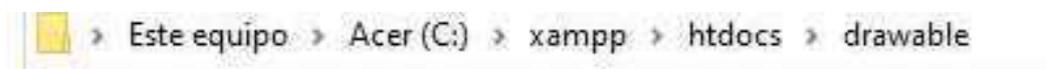


Fuente: Imagen propia

- **Agregar repuestos**

Para agregar un repuesto nuevo en la parte superior se da clic sobre la opción “INSERTAR” el servidor solicitará la información que se desea ingresar, es importante tener en cuenta que solo deja agregar imágenes tipo Jpg. Esta imagen deberá ser guardada en una carpeta con el nombre de drawable en el disco C. y en la base de datos se digita el nombre con el cual fue guardada.

Figura F- 5 Ruta acceso carpeta imágenes



Fuente: Imagen propia

Después de haber ingresa toda la información necesaria se da clic en la opción guardar y el repuesto será ingresado con éxito.

Figura F- 6 Agregar información

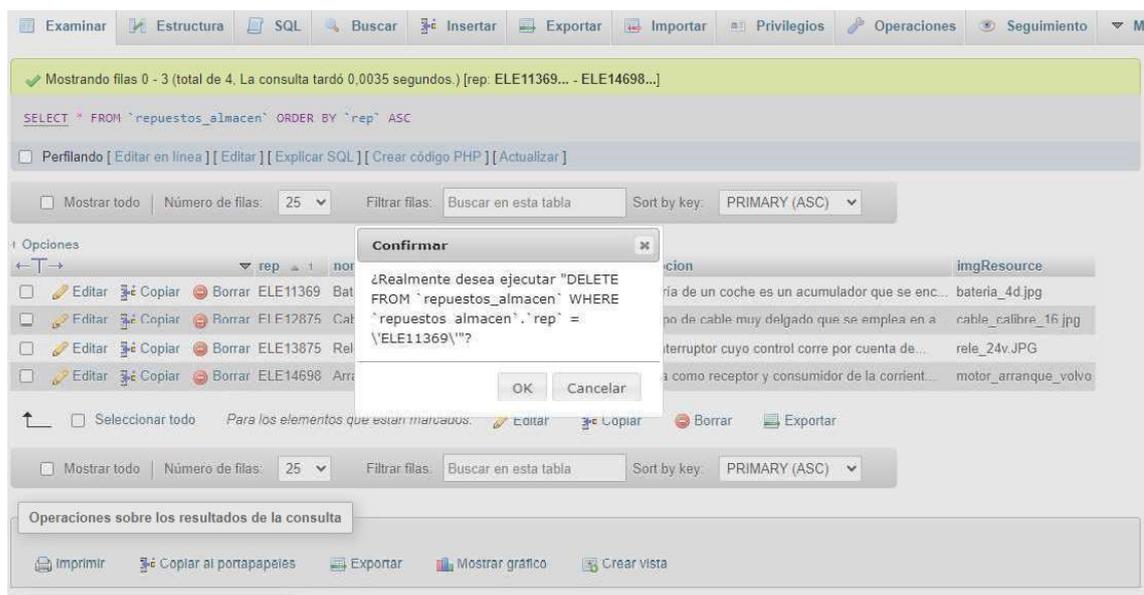
Columna	Tipo	Función	Nulo	Valor
rep	varchar(10)			
nombre	varchar(30)			
subistema	varchar(30)			
tipologia	varchar(30)			
descripcion	varchar(250)			
imgResource	text			

Fuente: Imagen propia

- **Eliminar Repuesto**

Para eliminar un repuesto de la base de datos se ubica en la página principal se busca el repuesto y se da clic en la opción eliminar frente al repuesto se abrirá una ventana de confirmación da clic en ok y el repuesto se eliminará.

Figura F- 7 Eliminar repuestos



Fuente: Imagen propia

- **Api-Rest**

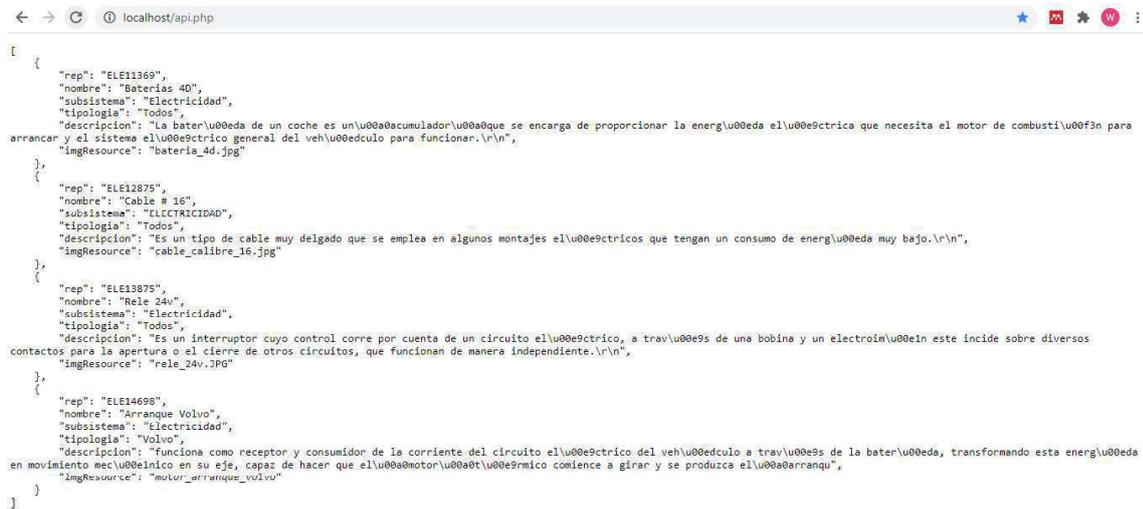
Para habilitar la api rest y empezar ser consumida por la aplicación móvil se tiene que abrir una url la cual da acceso directo a la api-rest, cuando un repuesto se ingresa o se elimina de la base de datos, se refresca la página web y se actualizará la api-rest.

Figura F- 8 Enlace Api-Rest



Fuente: Imagen propia

Figura F- 9 Api-Rest



```

[
  {
    "rep": "ELE11369",
    "nombre": "Baterias 40",
    "subsistema": "Electricidad",
    "tipologia": "Todos",
    "descripcion": "La bater\u00eda de un coche es un\u00acumulador\u00ac que se encarga de proporcionar la energ\u00eda el\u00e9ctrica que necesita el motor de combusti\u00f3n para arrancar y el sistema el\u00e9ctrico general del veh\u00edculo para funcionar.\r\n",
    "imgResource": "bateria_4d.jpg"
  },
  {
    "rep": "ELE12875",
    "nombre": "Cable # 16",
    "subsistema": "ELECTRICIDAD",
    "tipologia": "Todos",
    "descripcion": "Es un tipo de cable muy delgado que se emplea en algunos montajes el\u00e9ctricos que tengan un consumo de energ\u00eda muy bajo.\r\n",
    "imgResource": "cable_calibre_16.jpg"
  },
  {
    "rep": "ELE13875",
    "nombre": "Rele 24v",
    "subsistema": "Electricidad",
    "tipologia": "Todos",
    "descripcion": "Es un interruptor cuyo control corre por cuenta de un circuito el\u00e9ctrico, a trav\u00e9s de una bobina y un electroim\u00e1n este incide sobre diversos contactos para la apertura o el cierre de otros circuitos, que funcionan de manera independiente.\r\n",
    "imgResource": "rele_24v.jpg"
  },
  {
    "rep": "ELE14698",
    "nombre": "Arranque Volvo",
    "subsistema": "electricidad",
    "tipologia": "Volvo",
    "descripcion": "funciona como receptor y consumidor de la corriente del circuito el\u00e9ctrico del veh\u00edculo a trav\u00e9s de la bater\u00eda, transformando esta energ\u00eda en movimiento mec\u00e1nico en su eje, capaz de hacer que el\u00acmotor\u00ac comience a girar y se produzca el\u00acarranque",
    "imgResource": "motor_arranque_volvo"
  }
]

```

Fuente: Imagen propia

## Referencias Bibliográficas

- Xabier Barandiaran, “Cómo y por qué usar licencias copyleft,” 2003.
- C. Galeano, “Herramientas de software con licencia pública general para el modelado por elementos finitos,” *Dyna*, vol. 74, no. 153, pp. 313–324, 2007.
- M. Vidal, “Informe sobre licencias libres,” pp. 1–16, 2008.
- “RUA: Objetos de aprendizaje y licencias en Internet.”  
<http://rua.ua.es/dspace/handle/10045/27517> (accessed Nov. 09, 2020).
- E. Mifsuf, “Introducción a Apache,” p. 19, 2017, [Online]. Available:  
<http://descargas.pntic.mec.es/mentor/visitas/Apache.pdf>.
- E. Maida and J. Pacienza, “Metodologías de desarrollo de software,” *Bibl. Digit. la Univ. Católica Argentina*, p. 116, 2015, [Online]. Available:  
<http://bibliotecadigital.uca.edu.ar/repositorio/tesis/metodologias-desarrollo-software.pdf>.
- “Visual Studio Code: características principales - Damián De Luca.”  
<https://damiandeluca.com.ar/visual-studio-code-caracteristicas-principales> (accessed Oct. 28, 2021).
- D. D. Dvorski, “Installing, configuring, and developing with Xampp,” *D. Dvorski Dalibor*, no. March, pp. 1–10, 2007.
- “phpMyAdmin.” <https://www.phpmyadmin.net/> (accessed Oct. 28, 2021).
- “Introducción a Android Studio | Desarrolladores de Android.”  
<https://developer.android.com/studio/intro?hl=es-419> (accessed Oct. 28, 2021).
- H. M. Deitel, “Java: cómo programar,” 2016.
- E. Ruiz Lizama, *Clases de datos*, vol. 3, no. 1. 2014.

