Documentación técnica para el Desarrollo de Microservicio RESTful para la Administración de Teléfonos.

*Examen práctico*

*20-03-2025*

*Luigi Hernández García*

Contenido

[1. Requerimiento Examen práctico. 3](#_Toc193376663)

[2. Diagrama de Entendimiento para resolver el examen práctico. 4](#_Toc193376664)

[3. Diagrama de Arquitectura. 5](#_Toc193376665)

[4. Obtener Código Fuente. 5](#_Toc193376666)

[5. Entorno de desarrollo. 5](#_Toc193376667)

[5.1 Instalar Java. 6](#_Toc193376668)

[5.2 Instalar GIT 6](#_Toc193376669)

[5.3 Instalar Visual studio code. 7](#_Toc193376670)

[5.4 Instalar Docker. 7](#_Toc193376671)

[5.5 Instalar Postman 8](#_Toc193376672)

[5.6 Instalar Maven 8](#_Toc193376673)

[6. Como Descargar, Ejecutar y Probar el proyecto. 9](#_Toc193376674)

[6.1 Pasos para Iniciar Docker 9](#_Toc193376675)

[6.2 Descargar el proyecto de Github. 10](#_Toc193376676)

[6.3 Ejecutar el proyecto. 12](#_Toc193376677)

[6.4 Probamos el proyecto. 12](#_Toc193376678)

[7. Como obtener el header obligatorio DATA 14](#_Toc193376679)

[8. Servicios y documentación de la API tipos de respuesta. 16](#_Toc193376680)

[9. Validar y Probar los servicios en postman. 19](#_Toc193376681)

# Requerimiento Examen práctico.

El siguiente ejercicio pretende medir tu habilidad para desarrollar código usando java y Spring y las buenas prácticas que aplicas a tu trabajo diario, por lo que te sugerimos realizarlo como si fuera un entregable de tu día a día. Se considerará la simplicidad que puedas implementar pero que logres cumplir con los objetivos solicitados.

Diseñar un microservicio RESTFUL para que permita la **administración de Teléfonos**, la información del teléfono se define más adelante.

El microservicio debe permitir:

1. **Consultar** los datos de un teléfono
2. **Crear** los datos de un teléfono
3. **Eliminar** los datos de un teléfono
4. **Actualizar** los datos de un teléfono
5. Crear un método de **consulta por IMEI**

La base de datos puede ser Relacional o no SQL(mayores puntos, **NoSQL**).

El microservicio debe recibir un header obligatorio de nombre “DATA”. Como valor debe recibir una cadena hasheada con sha256, la cadena a hashear es “MIEXAMENPRUEBA”.

Dentro del servicio se tiene que validar que en todas las peticiones venga el header “DATA”, si no contiene el valor “MIEXAMENPRUEBA” se debe rechazar la petición.

**Agregar una Cache con redis en la consulta del cliente por IMEI**. Solo se tiene que ir a BD la primera vez o cuando se cambie un valor del teléfono consultado.

**Resetear el Cache** al hacer un update del registro

Creación de **Test unitarios Junit** (utilizar mock y Mockito)

Consulta de todos los registros de los clientes con paginación.

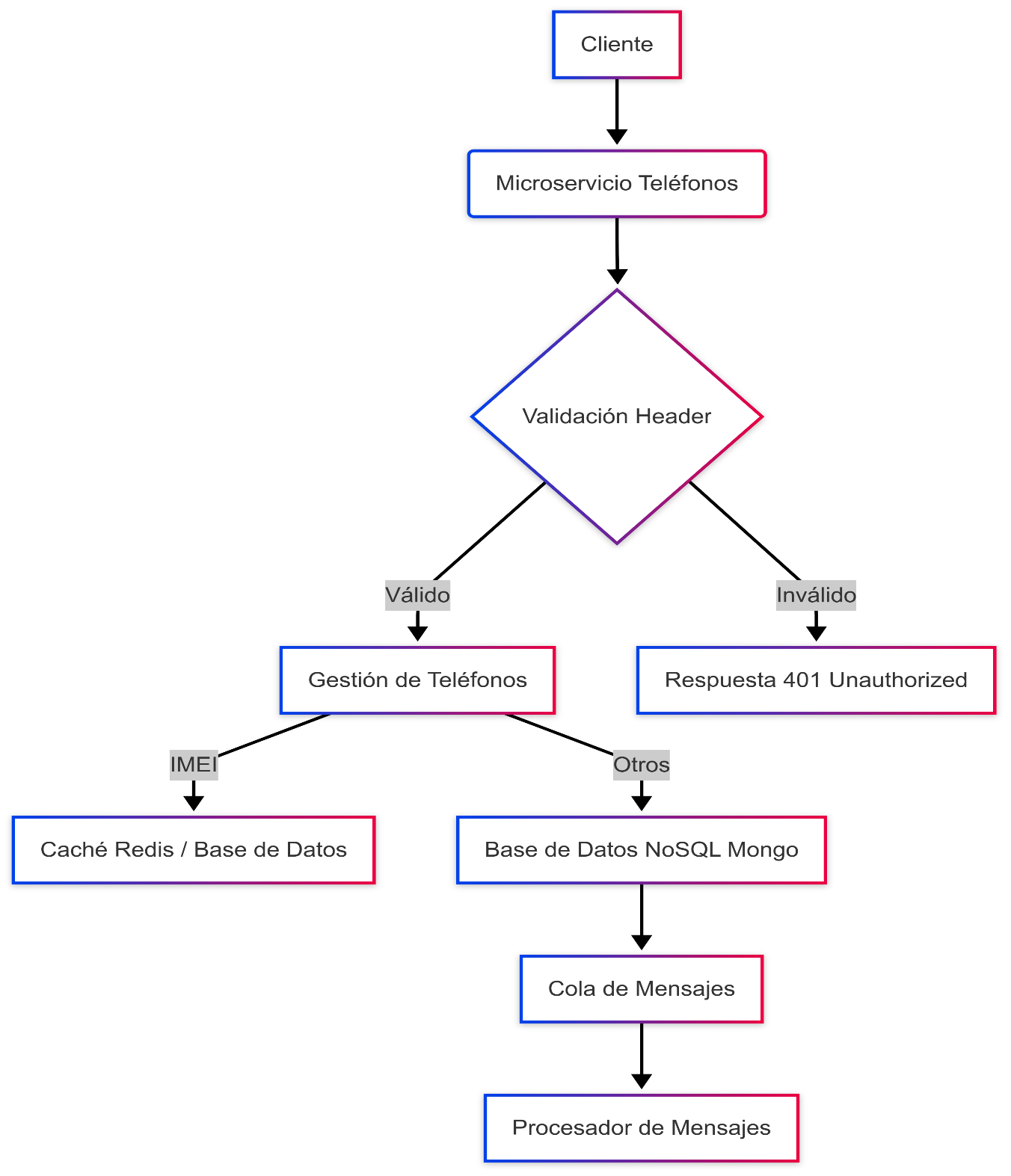
**Cada vez que se inserte o se actualice un registro en la base de datos de mongo se tiene que enviar un mensaje a una cola de mensajes**, indicando que se creó o modifico un objeto con su respectivo modificador, de preferencia RabbitMQ - **Kafka**

Implementar el consumo de los mensajes del rabbitMq o **Kafka** e insertarlos en la base de datos

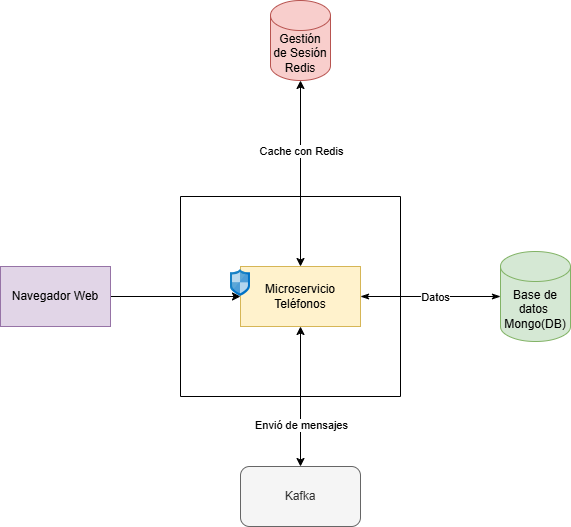
Los datos del teléfono son:

* Nombre (requerido, validar caracteres especiales)
* Marca (requerido, validar caracteres especiales)
* Modelo (requerido, validar caracteres especiales)
* Nombre corto(requerido, validar caracteres especiales)
* Fecha de creación (requerido, validar caracteres especiales)
* IMEI (requerido y de valor único )
* Numero celular (opcional)
* Email Soporte(opcional, opcional si trae valor validar formato de Email)
* ¿Tiene sistema IOS? (Campo booleano)

# Diagrama de Entendimiento para resolver el examen práctico.

****

# Diagrama de Arquitectura.

****

# Obtener Código Fuente.

**El proyecto se encuentra en GITHUB**\*\* Pre-condición tener instalado git[**(ver sección 5.2)**](#_5.2_Instalar_GIT)

git clone <https://github.com/luigihg/AdministrarTelefonos.git>

# Entorno de desarrollo.

Instalar el siguiente software.

## 5.1 Instalar Java.

1. **Descargar el JDK (Version 17 o superior):**

Descarga un JDK (Java Development Kit) de Oracle o Adoptium (OpenJDK). Adoptium es una opción común y gratuita.

Aquí tienes el enlace directo a las descargas de Adoptium OpenJDK 17:

<https://adoptium.net/releases.html?variant=jdk&jvmVariant=hotspot&version=17>

1. **Instalar el JDK:**

Ejecuta el instalador y sigue las instrucciones.

1. **Configurar Variables de Entorno (Windows/Linux):**

Configura JAVA\_HOME para que apunte a la instalación del JDK.

Agrega el directorio bin del JDK a la variable Path.

1. **Verificar la Instalación:**

Abre la línea de comandos y ejecuta java -version y javac -version

## 5.2 Instalar GIT

1. **Descargar Git:**

Descarga el instalador de la página oficial de Git.

La página principal para descargar Git es:

<https://git-scm.com/downloads>

1. **Instalar Git:**

Ejecuta el instalador y sigue las instrucciones.

1. **Verificar la Instalación:**

Abre la línea de comandos y escribe git --version.

1. **Configuración Inicial (Recomendado):**

Configura tu nombre de usuario y correo electrónico con los comandos git config --global user.name "Tu Nombre" y git config --global user.email "tu\_correo@ejemplo.com".

## 5.3 Instalar Visual studio code.

1. **Descargar VS Code:**

Descarga el instalador desde la página oficial de Visual Studio Code.

La página oficial para descargar Visual Studio Code es:

<https://code.visualstudio.com/>

1. **Instalar VS Code:**

Ejecuta el instalador y sigue las instrucciones.

1. **Iniciar VS Code:**

Abre la aplicación desde el menú de inicio, la carpeta de aplicaciones o la línea de comandos.

1. Instalar las siguientes extensiones: Extension Pack for Java, Maven for java, Spring Initializer.

## 5.4 Instalar Docker.

La página oficial para descargar Docker es:

* <https://www.docker.com/get-started>

En esta página, encontrarás enlaces para descargar Docker Desktop para diferentes sistemas operativos:

**Docker Desktop for Windows**

**Pasos Generales para la Instalación**

Los pasos exactos pueden variar ligeramente según tu sistema operativo, pero aquí tienes una descripción general:

**1. Requisitos del Sistema**

\* Asegúrate de que tu sistema operativo cumpla con los requisitos mínimos de Docker Desktop. Estos requisitos se detallan en la página de descarga y en la documentación de Docker.

\* Por lo general, necesitarás una versión reciente de Windows 10 u 11, macOS o una distribución de Linux compatible.

\* La virtualización de hardware debe estar habilitada en el BIOS/UEFI de tu computadora.

**2. Descargar Docker Desktop**

\* Ve a la página de descarga de Docker y descarga el instalador correspondiente a tu sistema operativo.

**3. Instalar Docker Desktop**

\* \*\*Windows/macOS:\*\*

\* Ejecuta el instalador descargado (`.exe` en Windows, `.dmg` en macOS) y sigue las instrucciones en pantalla.

\* Es posible que se te pida que reinicies tu computadora durante el proceso de instalación.

\* \*\*Linux:\*\*

\* El proceso de instalación en Linux varía según la distribución.

\* Docker proporciona instrucciones detalladas para diferentes distribuciones como Ubuntu, Debian, Fedora y CentOS.

\* Generalmente, la instalación implica usar la línea de comandos para agregar el repositorio de Docker, instalar los paquetes necesarios y configurar el servicio de Docker.

**4. Iniciar Docker Desktop**

\* Una vez instalado, inicia la aplicación Docker Desktop.

\* Es posible que se te pida que aceptes los términos de servicio.

\* Docker Desktop se ejecutará en segundo plano y mostrará un icono en la barra de tareas (Windows) o en la barra de menú (macOS).

**5. Verificar la Instalación**

\* Abre una terminal o línea de comandos.

\* Ejecuta el siguiente comando para verificar si Docker se ha instalado correctamente:

```bash docker --version ```

\* Deberías ver la versión de Docker instalada.

\* También puedes ejecutar:

```bash docker run hello-world ```

\* Este comando descarga una imagen de prueba y ejecuta un contenedor "hello-world". Si todo está configurado correctamente, verás un mensaje de bienvenida.

## 5.5 Instalar Postman

1. **Descargar Postman:**

Descarga el instalador desde la página oficial de Postman.

La página oficial para descargar Postman es:

<https://www.postman.com/downloads/>

1. **Instalar Postman:**

Ejecuta el instalador y sigue las instrucciones.

1. **Iniciar Postman:**

Abre la aplicación Postman.

## 5.6 Instalar Maven

1. **Requisitos Previos:**

Instala el JDK (Java Development Kit) [Ver Sección 5.1](#_5.1_Instalar_Java.).

1. **Descargar Maven:**

Descarga el archivo binario .zip o .tar.gz desde la página oficial de Maven.

La página oficial para descargar Apache Maven es:

<https://maven.apache.org/download.cgi>

1. **Instalar Maven:**

Extrae el archivo descargado en el directorio de instalación.

1. **Configurar Variables de Entorno:**

Configura JAVA\_HOME (apuntando al JDK).

Configura M2\_HOME o MAVEN\_HOME (apuntando a la instalación de Maven).

Agrega el directorio bin de Maven a la variable Path.

1. **Verificar la Instalación:**

Abre la línea de comandos y ejecuta mvn -v.

# Como Descargar, Ejecutar y Probar el proyecto.

## 6.1 Pasos para Iniciar Docker

1. **Buscar Docker Desktop:**
   * Puedes buscar "Docker Desktop" en el menú Inicio de Windows.
2. **Iniciar Docker Desktop:**
   * Haz clic en la aplicación "Docker Desktop" para iniciar

Interfaz de usuario gráfica, Tabla

El contenido generado por IA puede ser incorrecto.

1. **Verificar el Estado de Docker:**
   * Una vez que Docker Desktop se esté ejecutando, verás el icono de Docker en la bandeja del sistema (la barra de tareas, generalmente en la esquina inferior derecha de la pantalla).
   * Puedes hacer clic en el icono de Docker para ver el estado de Docker y acceder a las opciones de configuración.
2. **Abrir una Terminal:**
   * Abre una terminal de Windows (PowerShell o Símbolo del sistema).
3. **Verificar la Instalación de Docker (Opcional):**
   * Puedes verificar si Docker se ha iniciado correctamente ejecutando el siguiente comando en la terminal:

Bash

docker --version

Texto

El contenido generado por IA puede ser incorrecto.

## 6.2 Descargar el proyecto de Github.

1. En la consola de powershell o cmd ejecutamos el siguiente comando:  
>git clone <https://github.com/luigihg/AdministrarTelefonos.git>

Texto

El contenido generado por IA puede ser incorrecto.

>cd AdministrarTelefonos

AdministrarTelefonos> code .

Verificar las siguientes extensiones estén instaladas:  
Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Ir al menú Terminal> ejecutar el siguiente comando>

**Elimina y Recrea la Red:**

* A veces, la red Docker Compose puede entrar en un estado inconsistente. Eliminarla y recrearla puede solucionar el problema.

Bash

docker-compose down

docker network prune -f # Elimina todas las redes no utilizadas

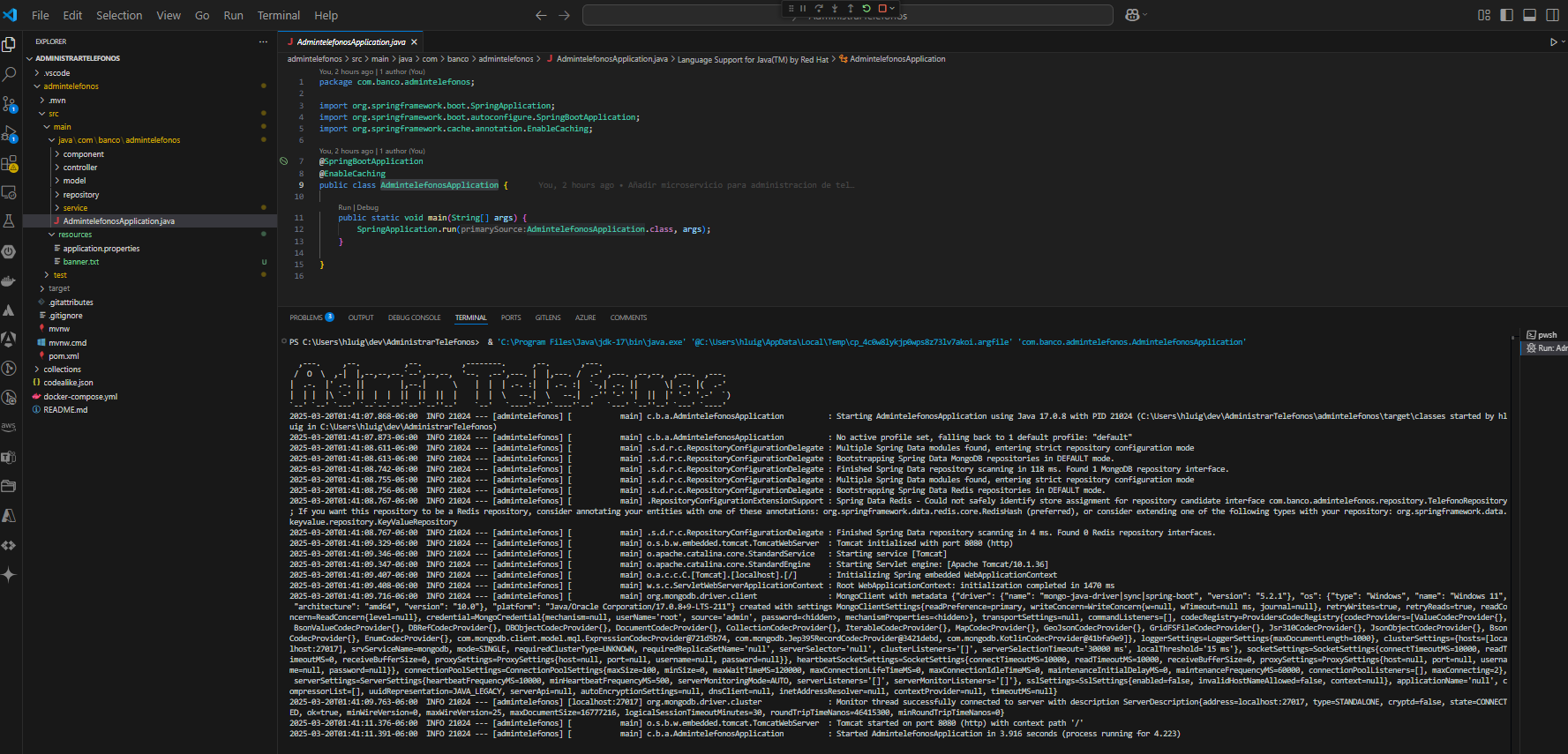
**docker-compose up --build**

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

## 6.3 Ejecutar el proyecto.

Ejecutamos Run Java seleccionamos la clase AdmintelefonosApplication.java botón derecho Run Java.



## 6.4 Probamos el proyecto.

1. Abrir postman

2. Importamos la collection que se encuentra en la raíz de la carpeta con el nombre AdministrarTelefonos\collections\AdministradorTelefonos.postman\_collection.json

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

1. Abrimos la collection CrearDatosTelefono.

Ejecutamos para crear un nuevo teléfono.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Damos clic a ejecutar la collection consultarTelefonos

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

# 7. Como obtener el header obligatorio DATA

**1. Obtener SHA-256 MIEXAMENPRUEBA**

**Ir a la página.**

[**https://emn178.github.io/online-tools/sha256.html**](https://emn178.github.io/online-tools/sha256.html)

**Seleccionar OutputEncoding : Base 64**

**Input colocar: MIEXAMENPRUEBA  
Verificar la salida: bXIHHGDUTBaGk7VX3Uj+7dfHY5N6nwK684IKIYvucD8=**

**2. Configurar Postman**

* **Abre Postman.**
* **Crea una nueva solicitud.**
* **Ingresa la URL del endpoint que deseas probar.** Por ejemplo, http://localhost:8080/telefonos.
* **Selecciona el método HTTP apropiado.** Por ejemplo, GET, POST, PUT, DELETE.

**3. Prueba con el Header "DATA" Correcto**

* **Ve a la pestaña "Headers".**
* **Agrega un nuevo header con el nombre "DATA".**
* **Pega el hash SHA-256 que obtuviste en el paso 1 como el valor del header.**
* **Envía la solicitud.**

**Resultado esperado:**

* La solicitud debe ser procesada correctamente por tu controlador.
* Debes recibir la respuesta esperada del endpoint.
* El código de estado de la respuesta debe ser 200 (OK) o el código de estado apropiado para el endpoint.

**4. Prueba con el Header "DATA" Incorrecto**

* **Ve a la pestaña "Headers".**
* **Cambia el valor del header "DATA" a un hash incorrecto.** (Por ejemplo, un hash aleatorio o el hash de otra cadena).
* **Envía la solicitud.**

**Resultado esperado:**

* La solicitud debe ser rechazada por el filtro.
* Debes recibir una respuesta con código de estado 401 (Unauthorized).
* El cuerpo de la respuesta debe contener el mensaje "Unauthorized: Invalid DATA header".

**5. Prueba sin el Header "DATA"**

* **Ve a la pestaña "Headers".**
* **Elimina el header "DATA".**
* **Envía la solicitud.**

**Resultado esperado:**

* La solicitud debe ser rechazada por el filtro.
* Debes recibir una respuesta con código de estado 401 (Unauthorized).
* El cuerpo de la respuesta debe contener el mensaje "Unauthorized: Invalid DATA header".

# 8. Servicios y documentación de la API tipos de respuesta.

**1. Descripción de los Servicios de la API de Teléfonos.**

1. **Consultar** los datos de un teléfono
2. **Crear** los datos de un teléfono
3. **Eliminar** los datos de un teléfono
4. **Actualizar** los datos de un teléfono
5. Crear un método de **consulta por IMEI**
6. Consulta de todos los registros de los clientes con paginación

|  |
| --- |
| **2. Servicios y Documentación de la API** |
| Aquí se detalla cada endpoint, sus métodos, payloads, respuestas y posibles errores. |

|  |  |  |
| --- | --- | --- |
| **Descripción** | **Método** | **Endpoint** |
| Lista teléfonos | GET | /telefonos |
| Obtiene un teléfono por su ID | GET | /telefonos/{id} |
| Obtiene un teléfono por su IMEI | GET | /telefonos/imei/{imei} |
| Crea un nuevo teléfono | POST | /telefonos |
| Actualiza un teléfono existente | PUT | /telefonos/{id} |
| Elimina un teléfono por su ID | DELETE | /telefonos/{id} |

**a) Obtener Teléfonos (Listar)**

**Payload de Entrada:**

Ninguno (puede aceptar parámetros de paginación en la URL, como page y size).

**Salida de Respuesta (Éxito - 200 OK):**

**Tipos de Errores:**

Ninguno específico del endpoint (errores generales del servidor pueden ocurrir).

**b) Obtener Teléfono por ID**

**Payload de Entrada:**

Ninguno

**Salida de Respuesta (Éxito - 200 OK):**

**Tipos de Errores:**

404 Not Found: Si el teléfono con el ID especificado no existe.

**c) Obtener Teléfono por IMEI**

**Payload de Entrada:**

Ninguno

**Salida de Respuesta (Éxito - 200 OK):**

**Tipos de Errores:**

404 Not Found: Si el teléfono con el IMEI especificado no existe**.**

**d) Crear Teléfono**

**Payload de Entrada:**

{

    "nombre": "Ejemplo Telefono",

    "marca": "Marca Ejemplo",

    "modelo": "Modelo Ejemplo",

    "nombreCorto": "Ejemplo \*",

    "fechaCreacion": "2024-03-15T10:30:00",

    "imei": "112233445566778",

    "numeroCelular": "5642877166",

    "emailSoporte": "soporte gmail.com",

    "tieneSistemaIOS": true

}

**Salida de Respuesta (Éxito - 201 Created):**

**Tipos de Errores:**

**400 Bad Request:** Si los datos de entrada no son válidos (por ejemplo, campos faltantes, formato incorrecto, IMEI ya existe).

* + **ErrorResponse:**

**Códigos de error posibles:**

**1001:** El nombre es requerido.

**1002:** El nombre no debe tener caracteres especiales.

**1003:** La Marca es requerida.

**1004:** La Marca no debe tener caracteres especiales.

**1005:** El Modelo es requerido.

**1006:** El Modelo no debe tener caracteres especiales.

**1007:** El Nombre corto es requerido.

**1008:** El Nombre corto no debe tener caracteres especiales.

**1009:** La Fecha de creación es requerida.

**1011:** La Fecha de creación tiene un formato invalido el formato valido es YYYY-MM-DDTHH:mm:ss

**1012:** El IMEI ingresado ya existe.

**1013:** El IMEI es requerido.

**1014:** El formato del correo electrónico es inválido.

**e) Actualizar Teléfono**

**Payload de Entrada:**

{

    "nombre": "Ejemplo Telefono",

    "marca": "Marca Ejemplo",

    "modelo": "Modelo Ejemplo",

    "nombreCorto": "Ejemplo",

    "fechaCreacion": "2025-03-15T10:30:00",

    "imei": "11223344556678",

    "numeroCelular": "5642877166",

    "emailSoporte": "soporte@gmail.com",

    "tieneSistemaIOS": true

}

**Salida de Respuesta (Éxito - 200 OK):**

**Tipos de Errores:**

* **400 Bad Request: Si los datos de entrada no son válidos.** 
  + **ErrorResponse:**

**1001:** El nombre es requerido.

**1002:** El nombre no debe tener caracteres especiales.

**1003:** La Marca es requerida.

**1004:** La Marca no debe tener caracteres especiales.

**1005:** El Modelo es requerido.

**1006:** El Modelo no debe tener caracteres especiales.

**1007:** El Nombre corto es requerido.

**1008:** El Nombre corto no debe tener caracteres especiales.

**1009:** La Fecha de creación es requerida.

**1011:** La Fecha de creación tiene un formato invalido el formato valido es YYYY-MM-DDTHH:mm:ss

**1012:** El IMEI ingresado ya existe.

**1013:** El IMEI es requerido.

**1014:** El formato del correo electrónico es inválido.

**f) Eliminar Teléfono**

**Payload de Entrada:**

Ninguno

**Salida de Respuesta (Éxito - 204 No Content):**

Sin cuerpo de respuesta

**Tipos de Errores:**

**404 Not Found: Si el teléfono con el ID especificado no existe.**

# 9. Validar y Probar los servicios en postman.

1. **Consultar** los datos de un teléfono

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

1. **Crear** los datos de un teléfono

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

1. **Eliminar** los datos de un teléfono

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

1. **Actualizar** los datos de un teléfono

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

1. Crear un método de **consulta por IMEI**

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

1. Consulta de todos los registros de los clientes con **paginación**.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

1. Validaciones campos.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto. Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto. Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto. Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.