

**Diagrama de Despliegue**

**Proyecto: Servi OS**

**Ordenes de Servicio de Mantenimiento para Aires Acondicionado**

**Integrante :**

**Rodney Zapata Palacio**

**Presentado al instructor:**

**Gustavo Adolfo Rodriguez**

**Servicio Nacional de aprendizaje SENA**

**Centro de Comercio y Servicios (Regional Cauca)**

**Cauca - Popayán**

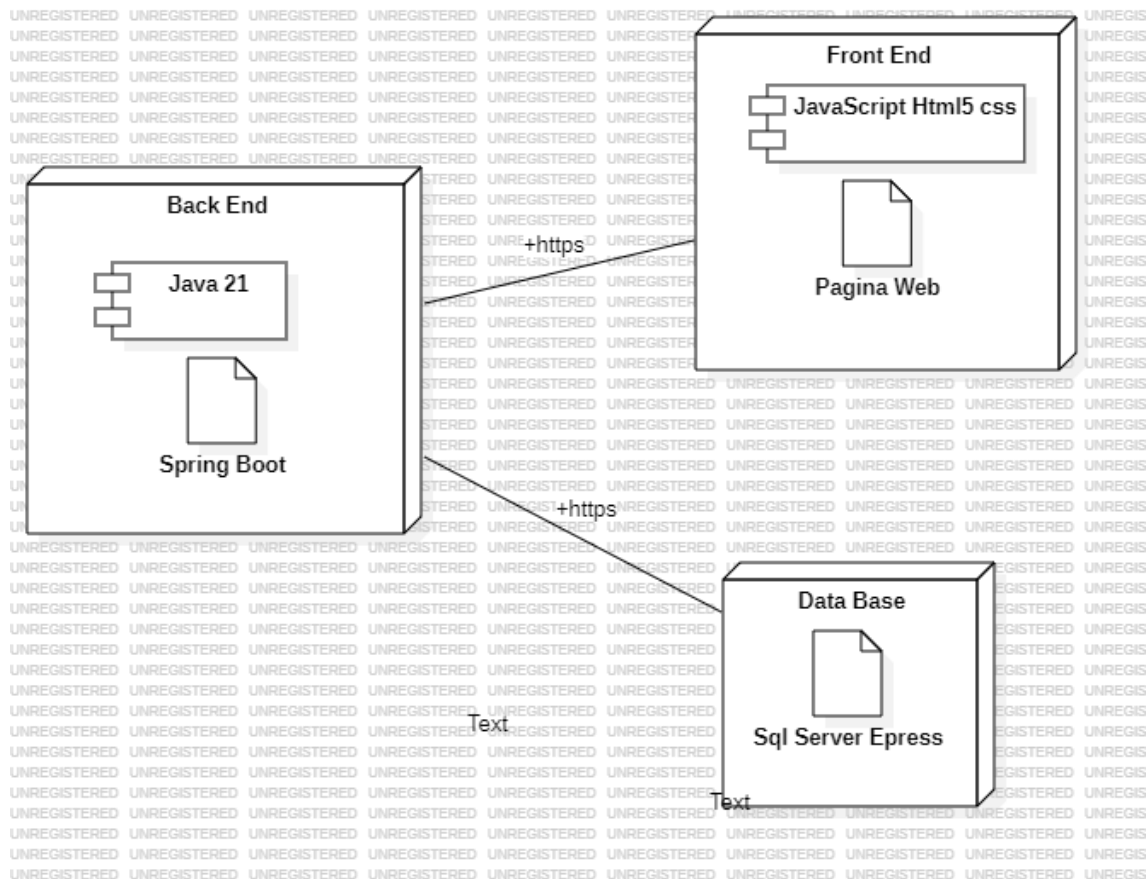
**Tecnólogo en Análisis y Desarrollo de Software**

**Ficha: 2675810**

**07 de septiembre del 2024**

**Propósito del desarrollo:** Servi OS se compone de un front Pagina web que consume un servicio web de una api rest realizada con Spring Boot. El cliente realiza las ordenes de servicio da mantenimiento para una empresa de aires acondicionado.

### Arquitectura General:



### Componentes principales:

#### 1. Cliente (Frontend):

- Dispositivo del usuario (PC, tableta, móvil): Navegador web (HTML5, CSS, JavaScript)
- El usuario interactúa con la aplicación a través del navegador.

#### 2. Servidor web (Backend):

- Servidor de aplicaciones: Spring Boot con Java 21
- Maneja las solicitudes HTTP del cliente, ejecuta la lógica del negocio y expone una API REST.

#### 3. Base de datos:

- Servidor de base de datos: SQL Server (o SQL Server Express)
- Almacena datos relacionados con las órdenes de servicio, clientes, equipos, actividades, etc.**

#### 4. Red:

- Internet: Permite la comunicación entre el cliente y el servidor.
- Red interna: Puede estar entre el servidor de aplicaciones y el servidor de base de datos si están en la misma infraestructura.

#### Detalles de cada componente:

1. **Cliente (Navegador web):** Los usuarios interactúan con la aplicación a través de una interfaz web, que está construida con HTML5 para la estructura, CSS para el estilo y JavaScript para la lógica en el lado del cliente. Las peticiones HTTP (GET, POST, PUT, DELETE) se envían a la API REST del servidor backend.
2. **Servidor de aplicaciones (Backend):** La aplicación backend está implementada en Java 21 utilizando Spring Boot para la construcción de la API REST. El servidor de aplicaciones maneja las solicitudes del cliente, realiza la lógica del negocio (como el procesamiento de órdenes de servicio) y responde con datos en formato JSON.
3. **Base de datos (SQL Server):** El servidor de base de datos SQL Server almacena toda la información de la aplicación, como las órdenes de servicio, clientes, técnicos, equipos, entre otros. Spring Boot utiliza JPA/Hibernate para realizar operaciones CRUD en la base de datos.

#### Condiciones en las que se realiza la entrega:

1. Se despliegan los backend en el servidor wildfy por la consola 10.122.7.10:9990.
2. El front se despliega en la ruta welcont-context del servidor wildfly 10.122.7.10
3. Se crea las tablas y se deja el ambiente apuntando a la base de datos 172.16.16.15:1521/test.

#### Importación del proyecto en desarrollo:

1. Descargar la suit de spring boot.
2. Instalar java 1.8.
3. Instalar el jdk disponible en Oracle.
4. Abrir la suit de spring boot – Ingresar a la pestaña “File” y escoger “Import...” – abrir la carpeta “Maven” y escoger la opción “Existing Maven Projects” – Colocar la ubicación del código fuente y escoger finalizar.

**Nota:** Para este proceso se requiere tener acceso a internet para descargar las librerías de Maven.

#### Parámetros requeridos:

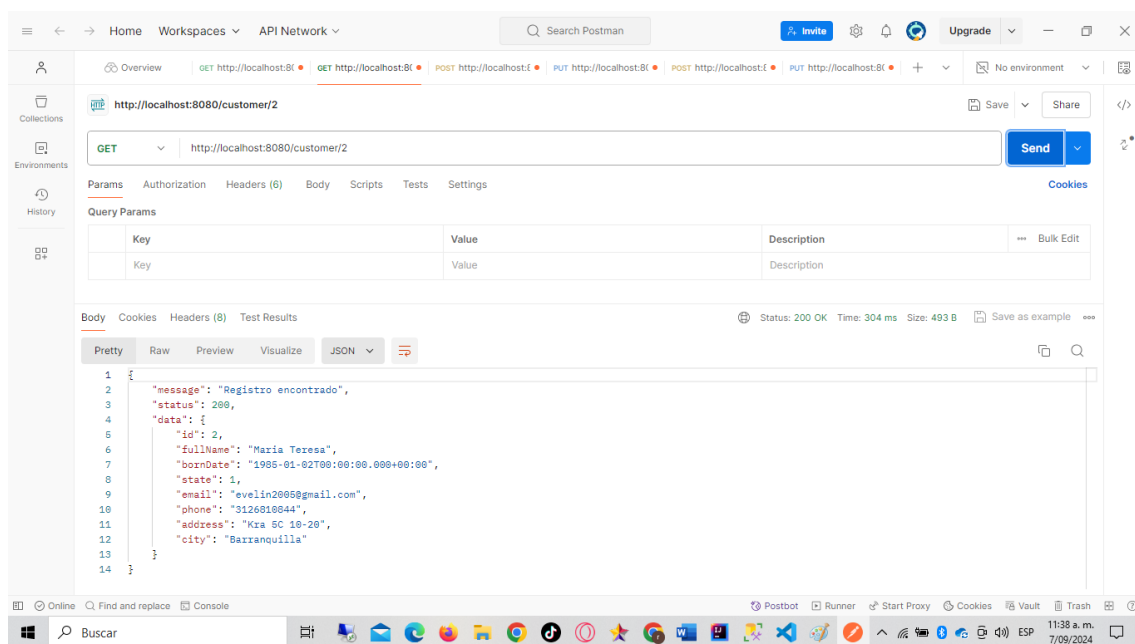
#CONFIGURACION DE LA BASE DE DATOS

spring:  
datasource:  
url:  
jdbc:sqlserver://localhost:61061;encrypt=false;trustServerCertificate=true  
;databaseName=servi\_ot  
username: **usuario de la base de datos**  
password: **contraseña de la base de datos**  
driver-class-name: com.microsoft.sqlserver.jdbc.SQLServerDriver

jpa:  
hibernate:  
ddl-auto: update  
show-sql: true  
database-platform: org.hibernate.dialect.SQLServerDialect

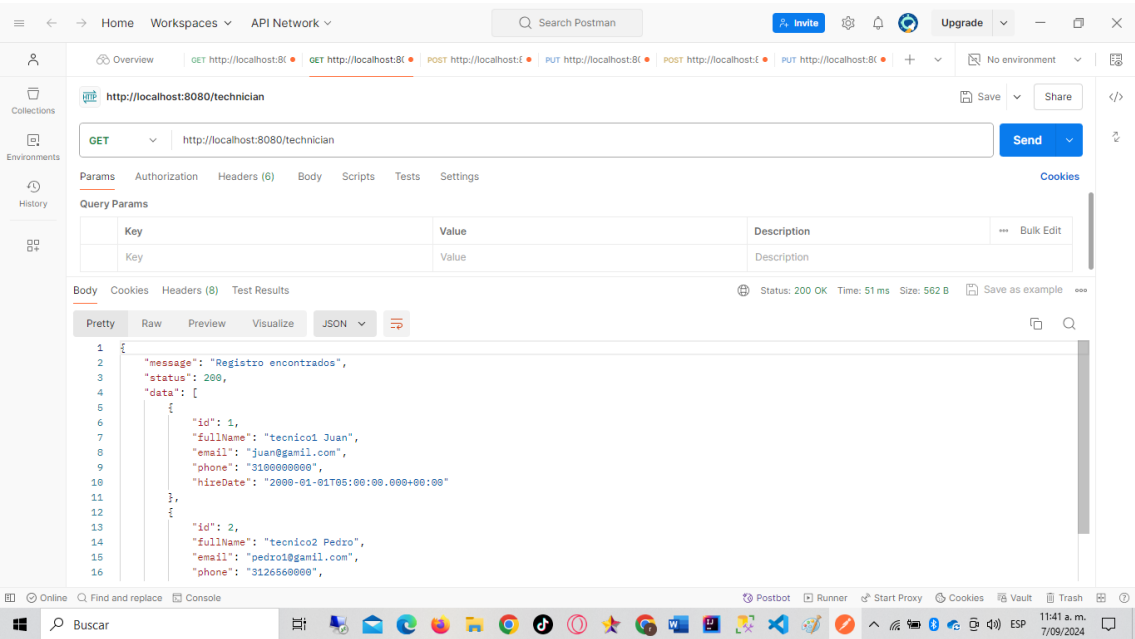
## Consumo del servicio Clientes:

<http://localhost:8080/customer/2>



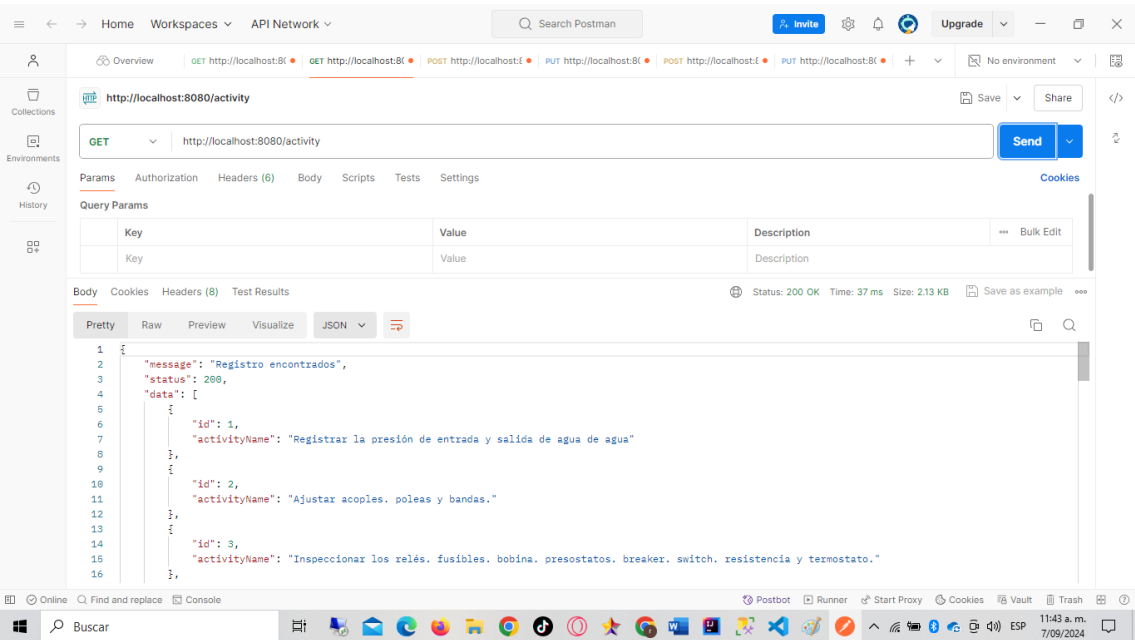
# Consumo del servicio Tecnicos:

<http://localhost:8080/technician>

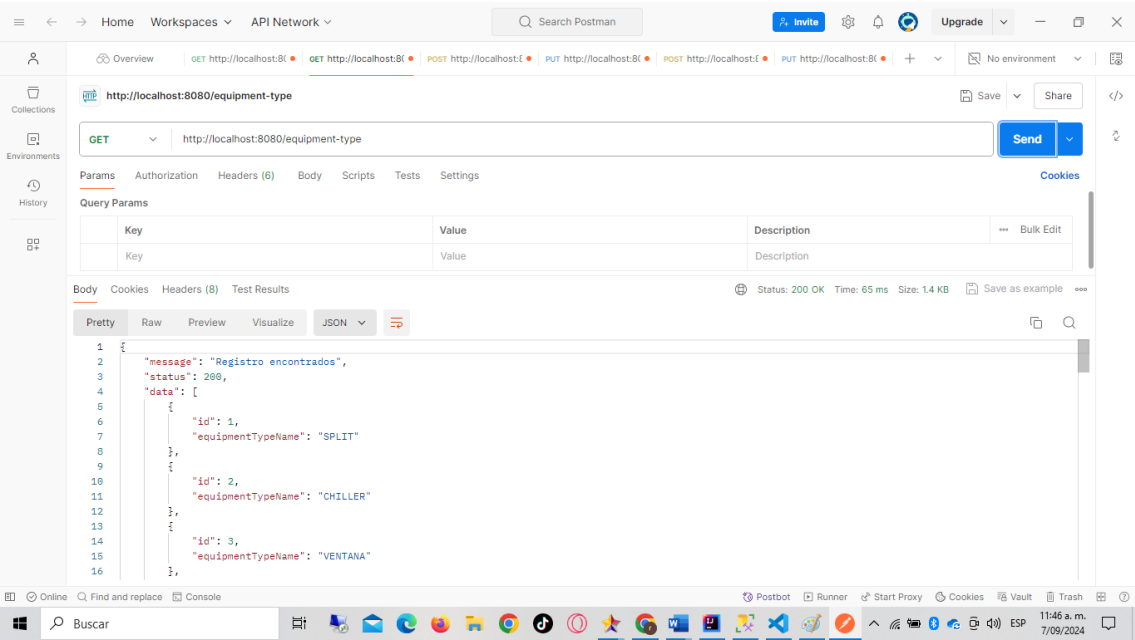


# Consumo del servicio Actividades:

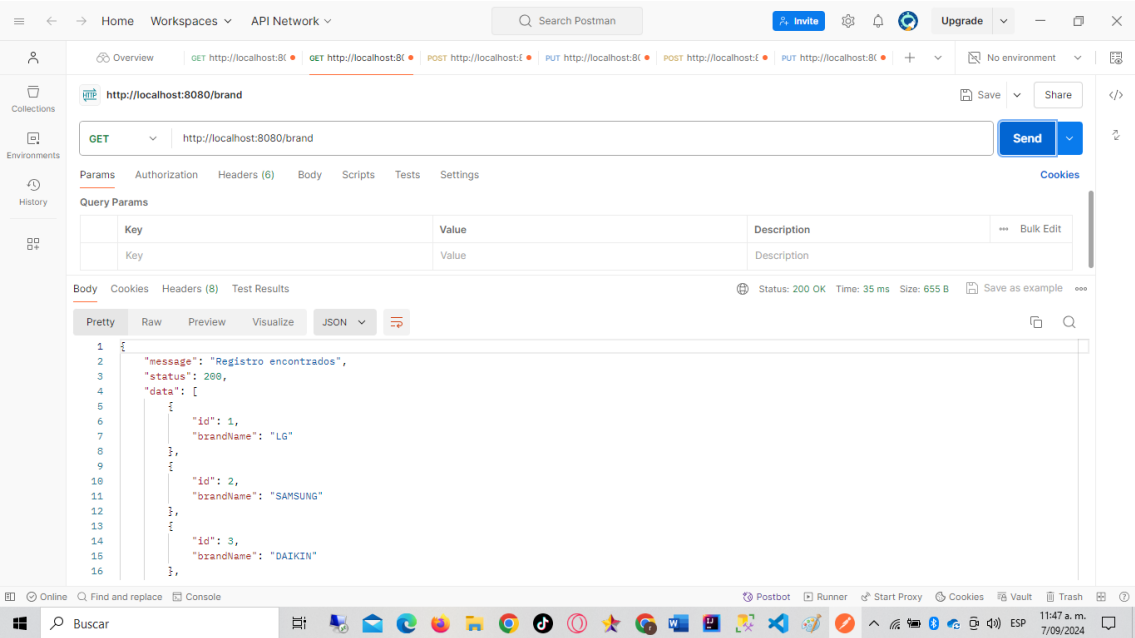
<http://localhost:8080/activity>



Consumo del servicio Tipos de aires acondicionado:  
<http://localhost:8080/equipment-type>



Consumo del servicio Marcas de equipos:  
<http://localhost:8080/brand>



## Consumo del servicio Refrigerantes para los equipos:

<http://localhost:8080/refrigerant>

The screenshot shows the Postman interface with a GET request to `http://localhost:8080/refrigerant`. The response is a JSON object with the following structure:

```
1 {
2   "message": "Registro encontrados",
3   "status": 200,
4   "data": [
5     {
6       "id": 5,
7       "refrigerantName": "GAS REFRIGERANTE R-1234yf"
8     },
9     {
10      "id": 4,
11      "refrigerantName": "GAS REFRIGERANTE R-134a"
12    },
13    {
14      "id": 1,
15      "refrigerantName": "GAS REFRIGERANTE R-22"
16    }
17  ]
18 }
```

## Consumo del servicio Equipos de aires acondicionado :

<http://localhost:8080/equipment/1>

The screenshot shows the Postman interface with a GET request to `http://localhost:8080/equipment/1`. The response is a JSON object with the following structure:

```
1 {
2   "message": "Registro encontrado",
3   "status": 200,
4   "data": {
5     "id": 1,
6     "serialNumber": "1005",
7     "installationDate": "2023-01-01T05:00:00.000+00:00",
8     "lastMaintenanceDate": null,
9     "customerid": 2,
10    "customerFullName": "Maria Teresa",
11    "customerPhone": "3126810044",
12    "equipmentTypeId": 1,
13    "equipmentTypeName": "SPLIT",
14    "brandId": 1,
15    "brandName": "LG",
16    "refrigerantId": 1,
17  }
18 }
```

## Consumo del Ordenes de servicios :

<http://localhost:8080/service-order>

The screenshot shows the Postman interface with a GET request to `http://localhost:8080/service-order`. The response status is 200 OK, with a time of 114 ms and a size of 542 B. The response body is displayed in JSON format, showing a service order with the following details:

```
1 {
2   "id": 1,
3   "customerId": 2,
4   "equipmentId": 1,
5   "date": "2024-01-01T05:00:00.000+00:00",
6   "customerFullName": "Maria Teresa",
7   "serialNumber": "1005",
8   "equipmentTypeName": "SPLIT",
9   "brandName": "LG",
10  "refrigerantName": "GAS REFRIGERANTE R-22",
11  "details": [
12    {
13      "id": 1,
14      "activityId": 1,
15      "description": "descripcion 1"
16    }
17  ]
18 }
```

The bottom of the image shows the Windows taskbar with the search bar and various application icons. The system clock indicates 11:52 a.m. on 7/09/2024.