

**UNIVERSIDAD PERUANA UNIÓN**  
**FACULTAD DE INGENIERÍA Y ARQUITECTURA**  
**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**



**TÍTULO:**

Examen Parcial

**CURSO:**

Lenguaje de programación II

**DOCENTE:**

David Mamani Pari

**PRESENTADO POR:**

Cayte Sejje Walter G.

Castillo Cuevaso Cherman

Escarcena Pancca Yohan

Huayta Mamani Jose Luis

Quispe Turpo Vladimir

**JULIACA – PERÚ**

**2024**

**Proyecto:** Crear un sistema para una microempresa que se dedica a la venta de prendas.

## **1. Descripción**

El proyecto elegido es un sistema para una tienda de ropa, cuyo objetivo es mejorar la gestión de inventario, ventas y compras dentro del negocio. El sistema permitirá a los empleados de la tienda realizar tareas clave como registrar nuevos productos, gestionar el stock, procesar ventas y generar reportes, todo de manera eficiente y automatizada.

### **Tecnologías usadas:**

- Java con Spring Boot
- Lombok
- MySQL con phpMyAdmin

## **2. Lista de Requerimientos:**

### **2.1 Requerimientos Funcionales:**

- El sistema debe permitir registrar un producto con su nombre, descripción, precio y stock.
- El sistema debe permitir registrar y modificar una categoría con un nombre.
- El sistema debe permitir registrar un proveedor con su nombre, dirección, teléfono y correo electrónico.
- El sistema debe permitir registrar un cliente con su nombre, dirección, teléfono y correo electrónico.
- El sistema debe permitir registrar una venta con la fecha de venta y el total.
- El sistema debe registrar los productos vendidos, con cantidad y precio de venta, dentro de una venta.
- El sistema debe permitir registrar una compra, incluyendo la fecha de compra y total.
- El sistema debe registrar los productos comprados, con cantidad y precio de compra.
- El sistema debe permitir registrar un empleado con nombre, puesto, salario, teléfono y correo electrónico.
- El sistema debe mostrar el inventario de productos, con la cantidad actual y la fecha de la última actualización.

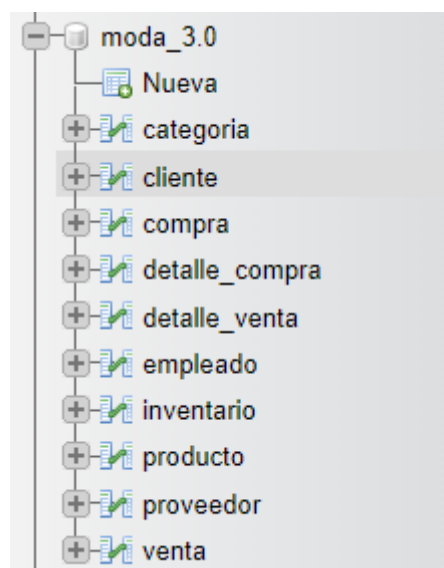
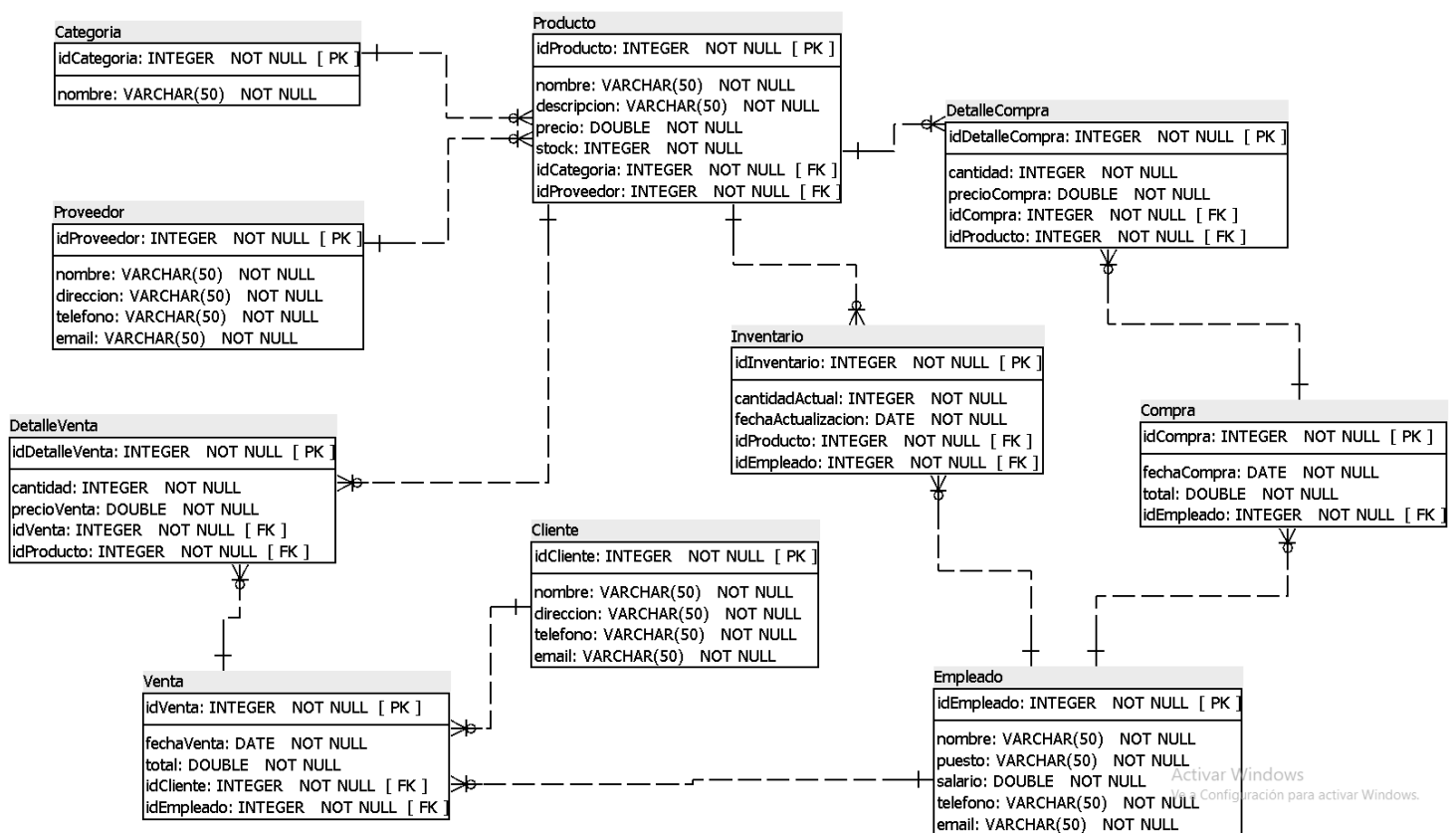
### **2.2 Requerimientos no Funcionales:**

- El sistema debe asegurar la autenticación de los usuarios mediante contraseñas cifradas.
- El sistema debe ser capaz de manejar un inventario de al menos 1000 productos sin degradar su rendimiento.

- El sistema debe ser compatible con los navegadores web más utilizados, ya que se espera que se acceda a través de una interfaz web.
- El código debe ser modular y seguir buenas prácticas de desarrollo para facilitar el mantenimiento y las futuras actualizaciones.
- El sistema debe proteger la información confidencial de los empleados.

### 3. Modelo de Base de Datos:

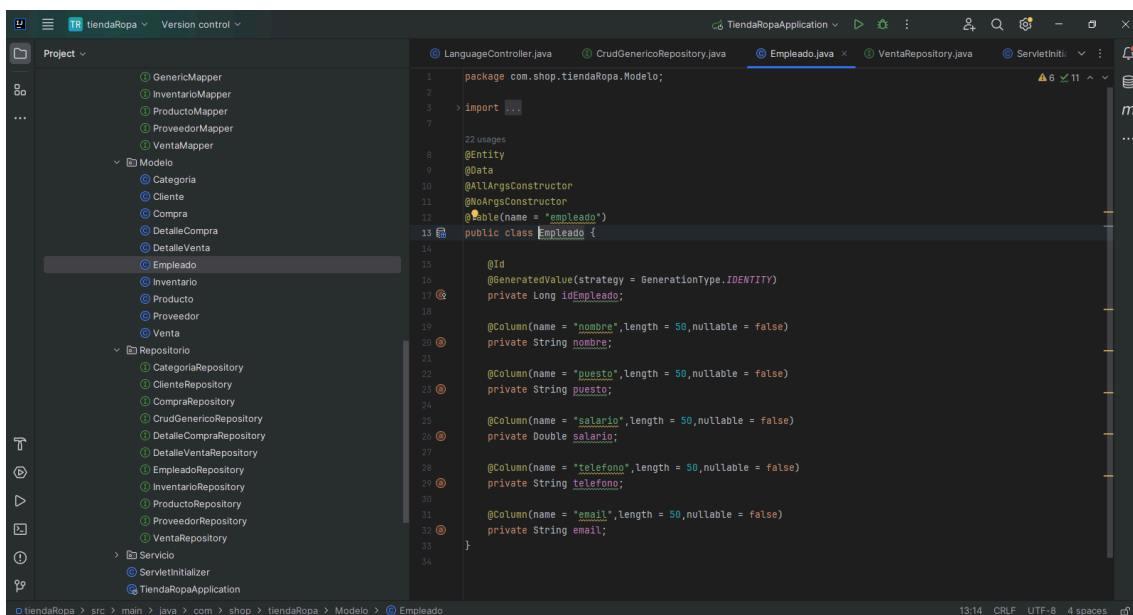
El modelo de base de datos debe incluir las siguientes tablas:



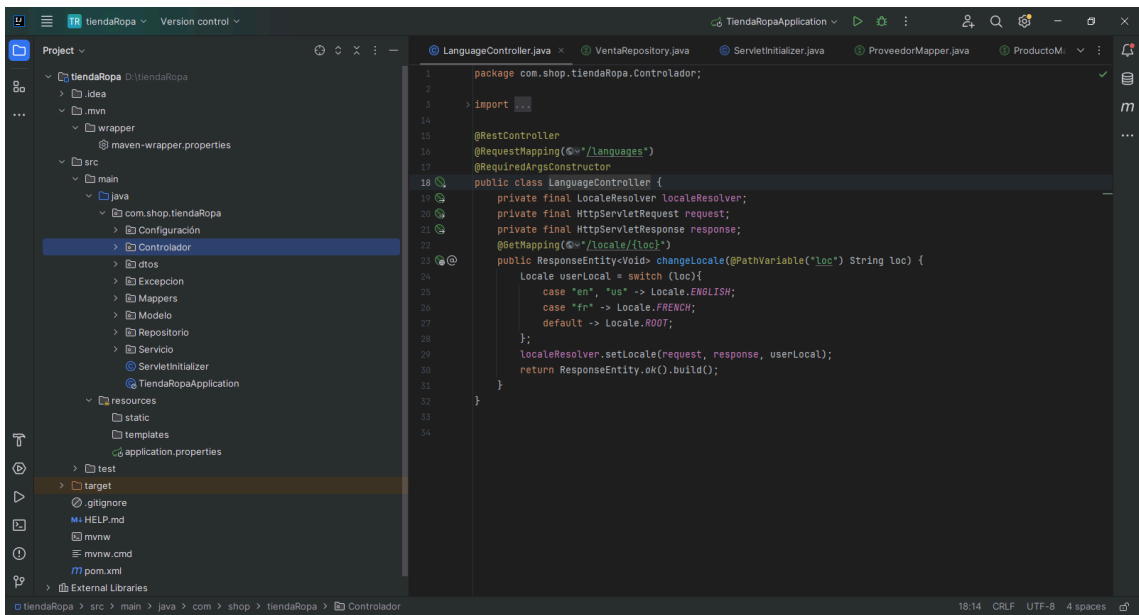
#### 4. Aplicación Backend en Java con Spring Boot:

La aplicación backend debe implementar las siguientes funcionalidades:

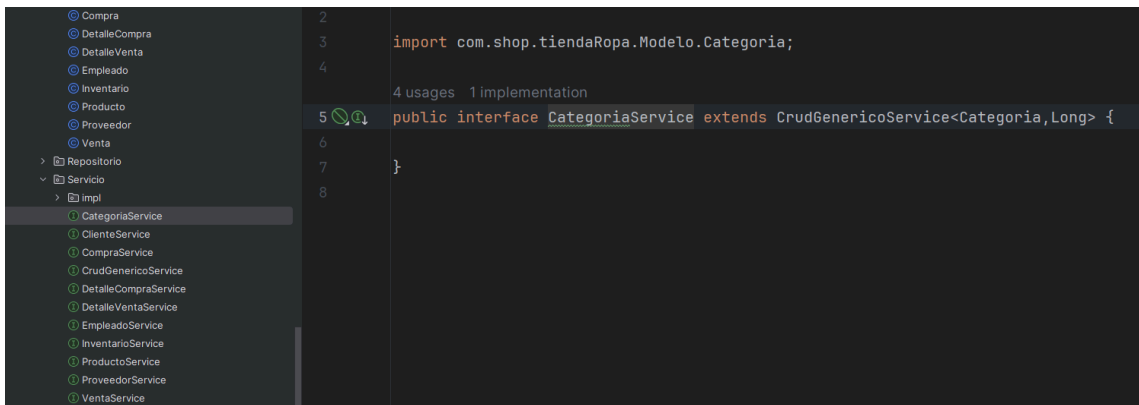
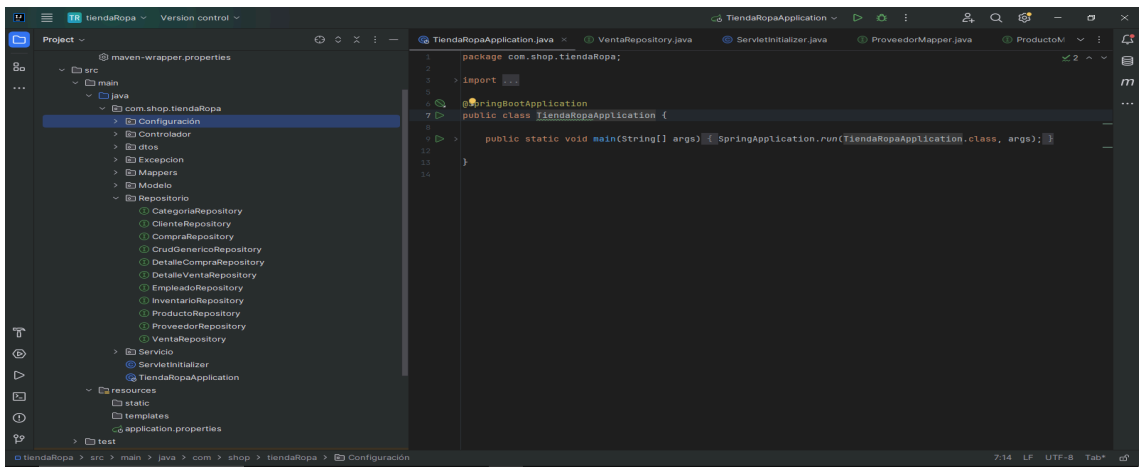
- API REST: Exponer una API REST para que la aplicación frontend pueda acceder a la información de la tienda.
- Gestión de Usuarios: Permitir el registro, login, actualización y eliminación de usuarios.
- Gestión de Productos: Permitir la creación, actualización y eliminación de productos.
- Gestión de Pedidos: Permitir la creación, actualización y eliminación de pedidos.
- Gestión de Pagos: Integrar con una pasarela de pago para procesar las transacciones.
- Gestión de Correo Electrónico: Enviar correos electrónicos a los usuarios con información sobre su pedido, etc.



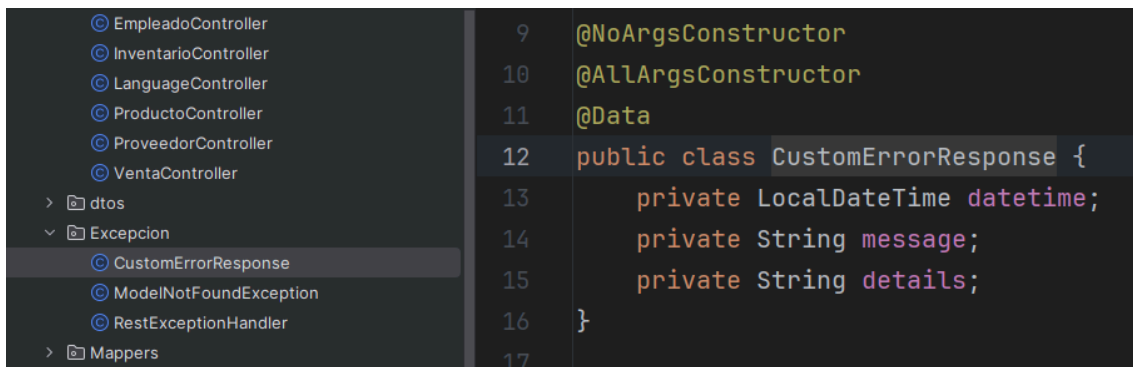
#### 5. Aplicar Conceptos de Inyección de Dependencia a Nivel de las Distintas Capas del Proyecto Backend:



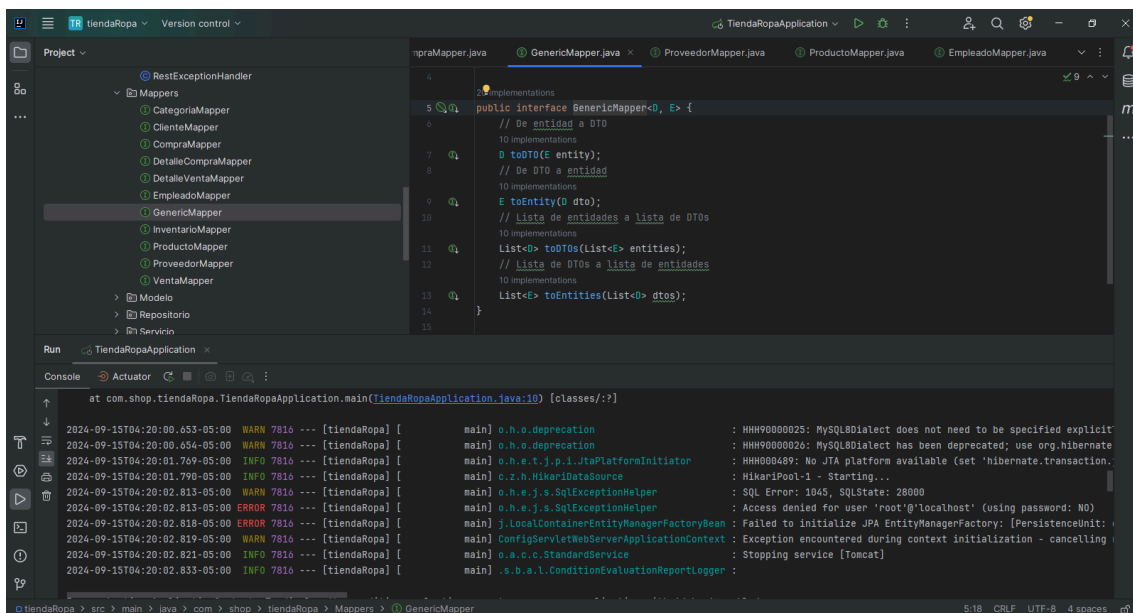
## 6. Aplicar los Conceptos de Genéricos a Nivel de Capa de Repository y Service según CRUD Básico:



## 7. Gestión de Errores; Gestionar Importe de Data; Perfiles de Desarrollo - Producción:



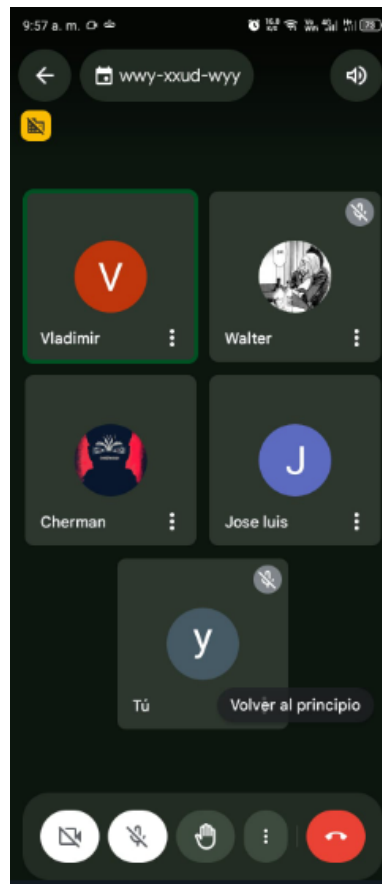
## 8. Manejo de DTO y Mappers; Idiomas y Validaciones de Inputs de RestControllers:



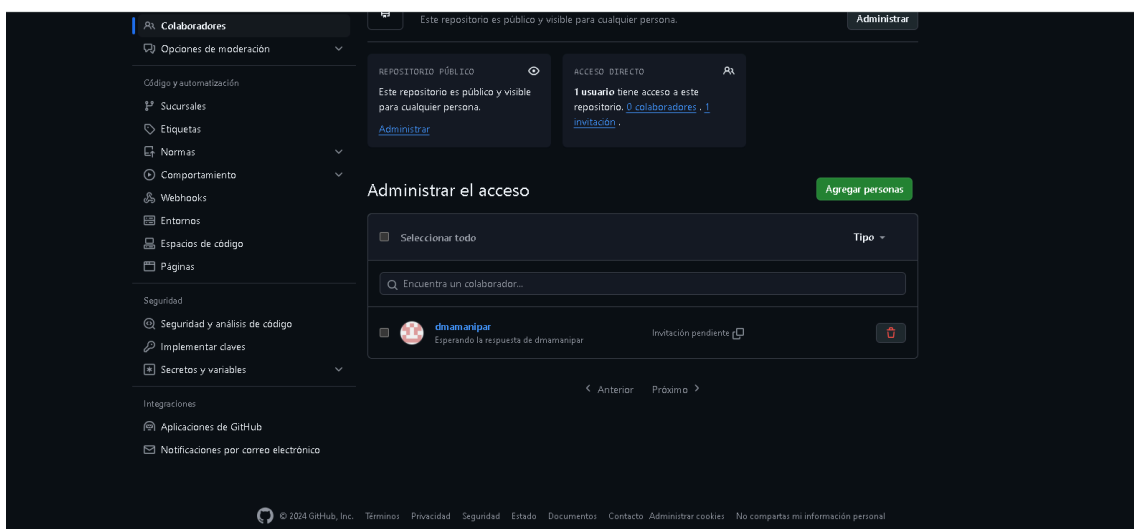
## 1.-Subir el proyecto completo a un repositorio de Github

<https://github.com/layder93122/moda3.0>

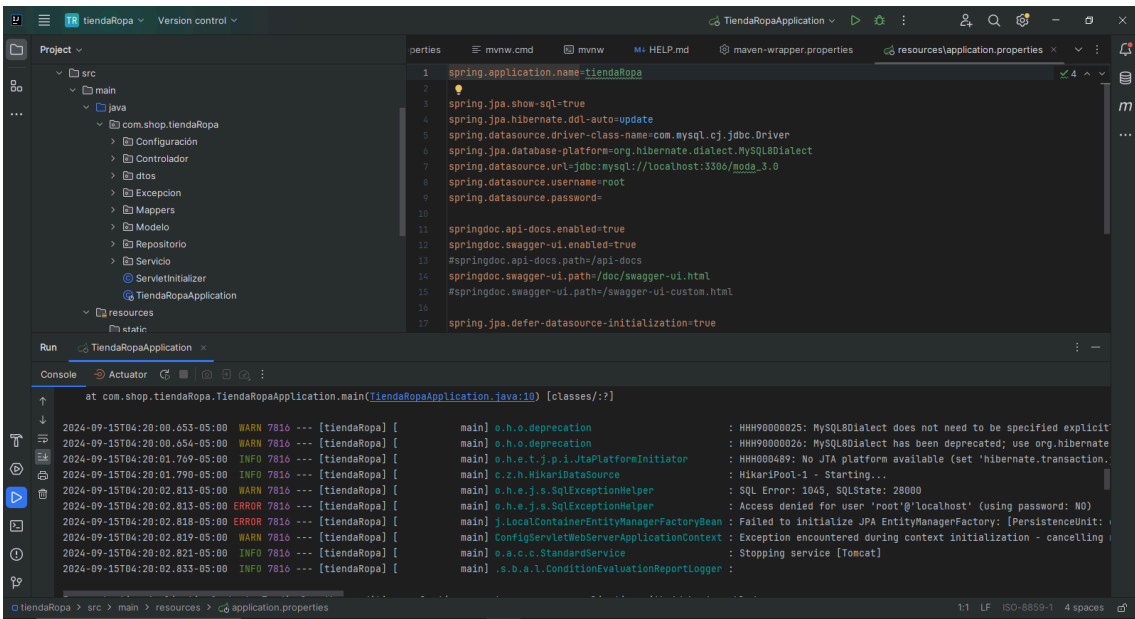
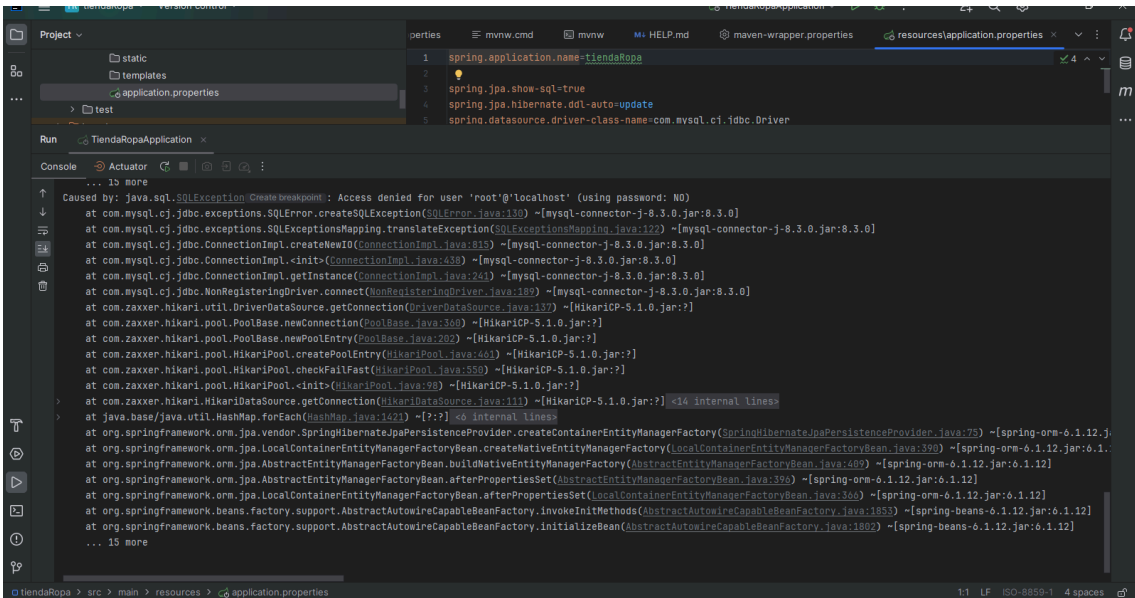
## 2.- Demostrar que todos integrantes del proyecto colaboraron en el desarrollo



## 3.- Asignar como colaborador al usuario dmamanipar



## 4.-Subir capturas de pantalla del funcionamiento de Proyecto Backend



## Guardar

METHOD

POST

SCHEME // HOST [ "?" PORT ] [ PATH [ "?" QUERY ] ]

http://localhost:8080/cliente/guardar

QUERY PARAMETERS

HEADERS

☒

Content-Type

:

application/json

x

+

Add header

?

Add authorization

Form

BODY

1

{

2

"id": "1",

3

"nombre": "Saray",

4

"direccion": "Av. Triunfo",

5

"telefono": "978647563",

6

"email": "saray@gmail.com"

7

}

Response

200

HEADERS

pretty

BODY

Content-Type:

application/json

Transfer-Encoding:

chunked

Date:

Sun, 15 Sep 2024 17:09:05 GMT

Keep-Alive:

timeout=60

Connection:

keep-alive

{

idCliente: 1,

nombre: "Saray",

direccion: "Av. Triunfo",

telefono: "978647563",

}



# Eliminar

METHOD

DELETE

SCHEME // HOST [ ":" PORT ] [ PATH [ "?" QUERY ] ]

http://localhost:8080/cliente/eliminar/1

Send

length: 40 byte(s)

QUERY PARAMETERS

HEADERS

Form

BODY

XHR does not allow payloads for DELETE request.

Response

Cache Detected - Elapsed Time: 28ms

200

# Obtener por id

METHOD

GET

SCHEME // HOST [ ":" PORT ] [ PATH [ "?" QUERY ] ]

http://localhost:8080/cliente/list/3

Send

length: 0 byte(s)

QUERY PARAMETERS

HEADERS

Form

BODY

XHR does not allow payloads for GET request.

Response

Cache Detected - Elapsed Time: 28ms

200

HEADERS

pretty

Content-Type:

application/json

Transfer-Encoding:

chunked

Date:

Sun, 15 Sep 2024 17:15:05 GMT

Keep-Alive:

timeout=60

Connection:

keep-alive

BODY

```
{
  nombre: "Santana",
  direccion: "Sda. Puno",
  telefono: "978647563",
  email: "santana@gmail.com"
}
```

COMPLETE REQUEST HEADERS

# Listar

METHOD

GET

SCHEME // HOST [ ":" PORT ] [ PATH [ "?" QUERY ] ]

http://localhost:8080/cliente/list

Send

length: 34 byte(s)

QUERY PARAMETERS

HEADERS

Form

BODY

XHR does not allow payloads for GET request.

Response

Cache Detected - Elapsed Time: 88ms

200

HEADERS

pretty

Content-Type:

application/json

Transfer-Encoding:

chunked

Date:

Sun, 15 Sep 2024 17:15:51 GMT

Keep-Alive:

timeout=60

Connection:

keep-alive

BODY

```
[
  {
    nombre: "Saray",
    direccion: "Av. Triunfo",
    telefono: "978647563",
    email: "saray@gmail.com"
  },
  {
    nombre: "Santana",
    direccion: "Sda. Puno",
    telefono: "978647563",
    email: "santana@gmail.com"
  }
]
```

COMPLETE REQUEST HEADERS

Top

Bottom

Collapse

Open

2Request

Copy

Download