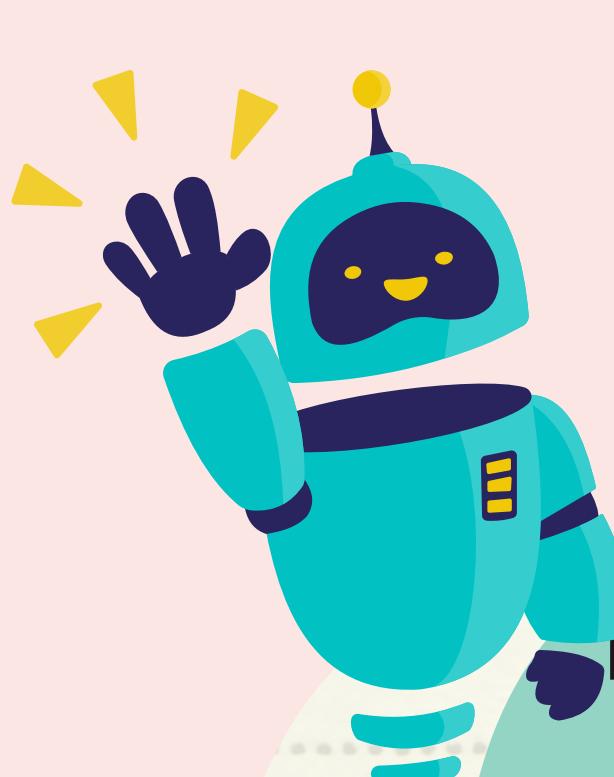


TESTING DE MICROSERVICIOS PERFUSMART

GRUPO 1: GONZALO NAVARRETE-CARLA PRADO-FERNANDO ZÁRATE



INTRODUCCIÓN

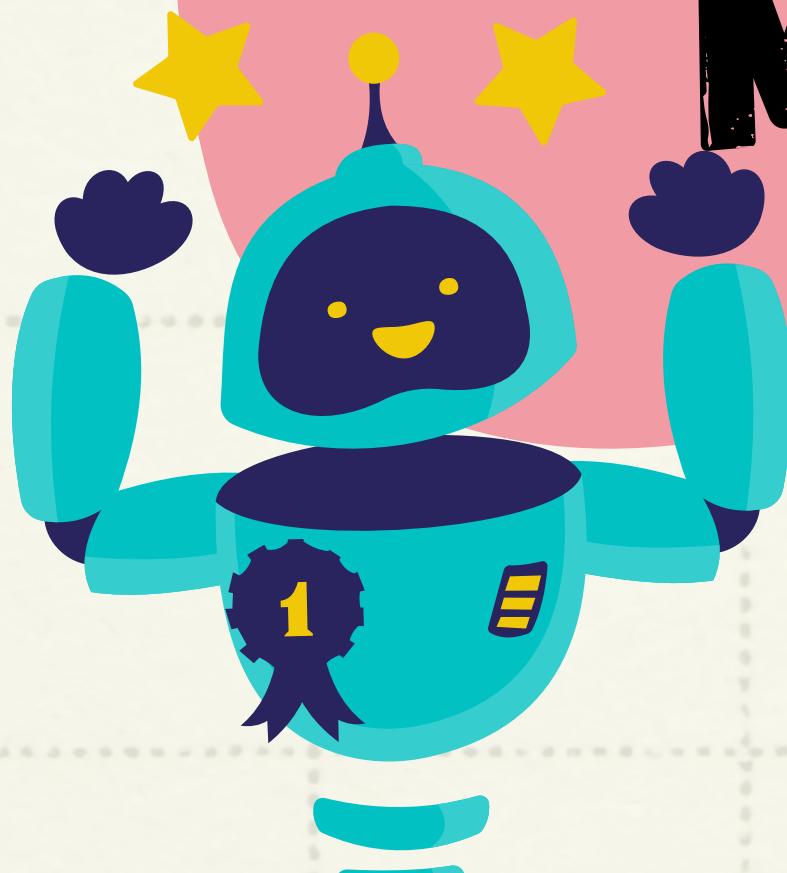
A medida que las aplicaciones se fragmentan en **servicios independientes**, se vuelve fundamental garantizar que **cada componente funcione correctamente de forma aislada**.

Esta presentación aborda el proceso de testing de microservicios en **Spring Boot** utilizando los framework **JUnit** y **Mockito** (Java). Además de su documentación y control de versiones.

El objetivo principal del **testing** es **validar el comportamiento interno de las clases** (como servicios y controladores) **sin depender de recursos externos** como bases de datos o servidores web reales.

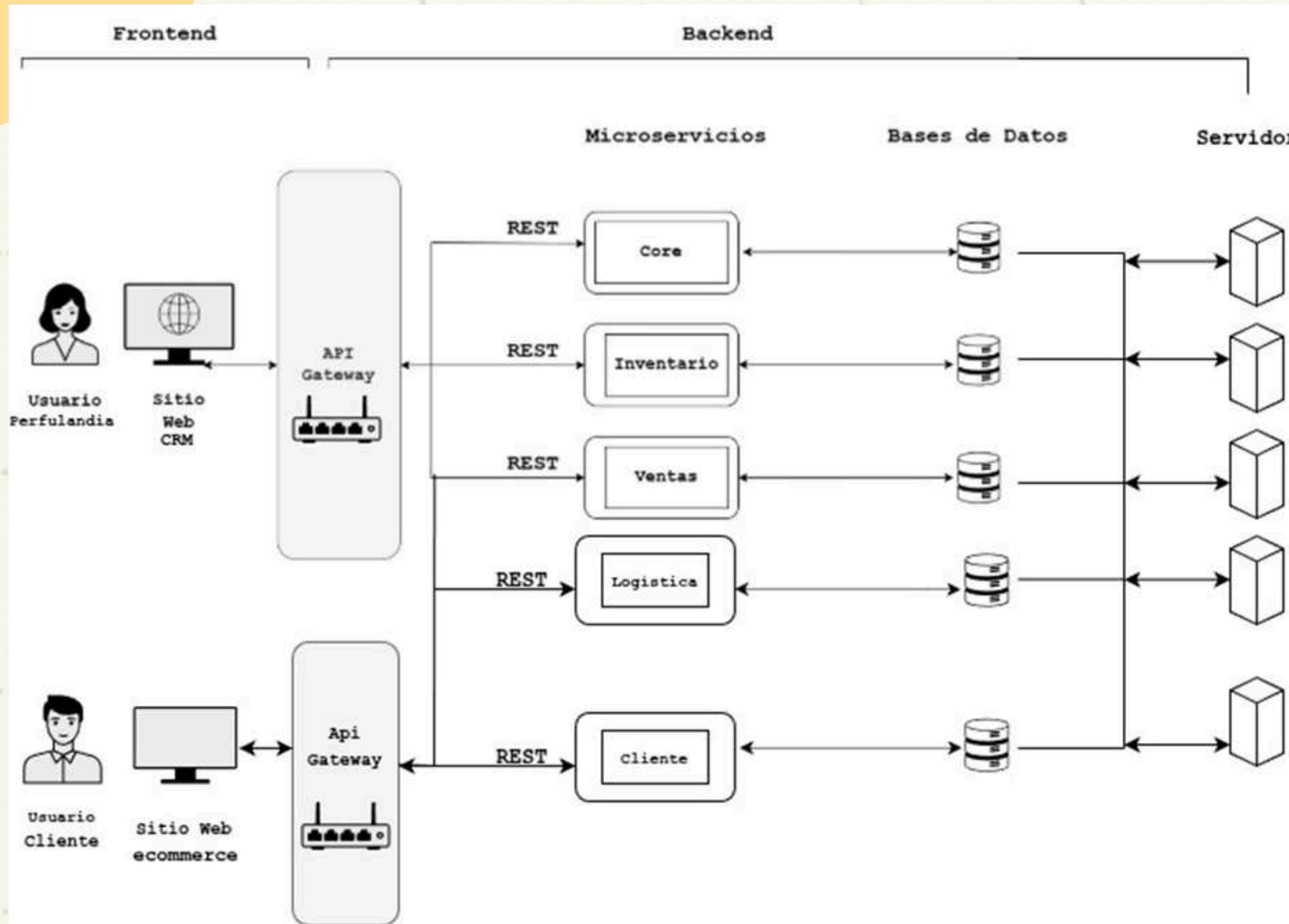


ARQUITECTURA DE MICROSERVICIOS



CORRECCIONES EXPERIENCIA 2

ARQUITECTURA DE MICROSERVICIOS (ENTORNO IDEAL)



Modularidad:
Cada microservicio tendría su propia base de datos en el **servidor** que le corresponde.

API gateway:
Punto centralizado donde se reciben y enrutan todas las peticiones a las APIs de un sistema.

Permite: Control de acceso, errores y tráfico, combinar datos de diferentes microservicios, monitoreo y registro de solicitudes.

CORRECCIONES EXPERIENCIA 2

DIAGRAMA DE COMPONENTES (ENTORNO DE DESARROLLO Y PRUEBAS)

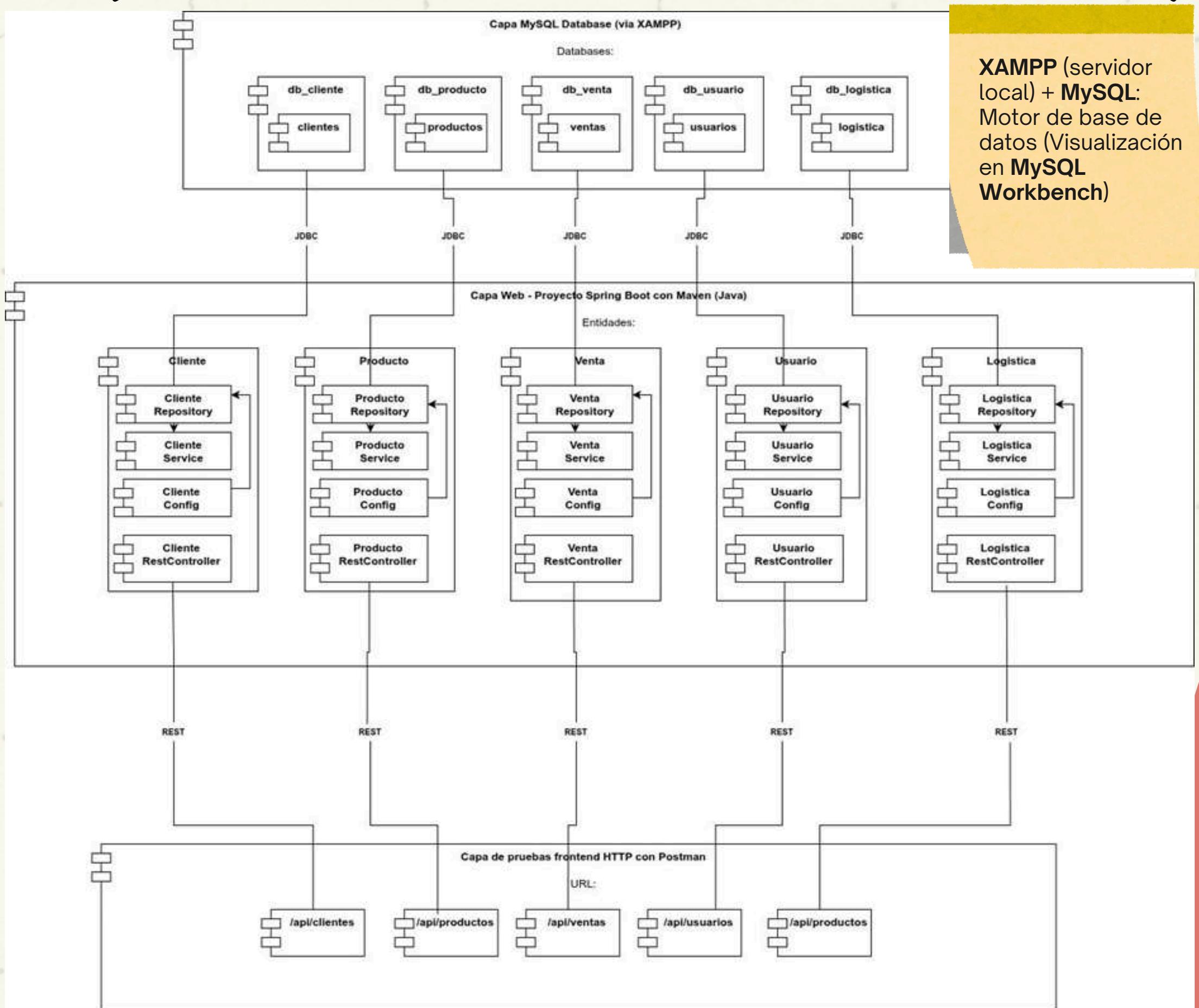
Se agregaron dos nuevas entidades.

Microservicios propuestos:

- 1.Core (users empleados)
- 2.Inventario (producto)
- 3.Ventas
- 4.Logística
- 5.Cliente (UI)

Backend de la aplicación (Proyecto Spring Boot)

Postman:
Simula el comportamiento de un cliente (pruebas manuales de rutas y endpoints)

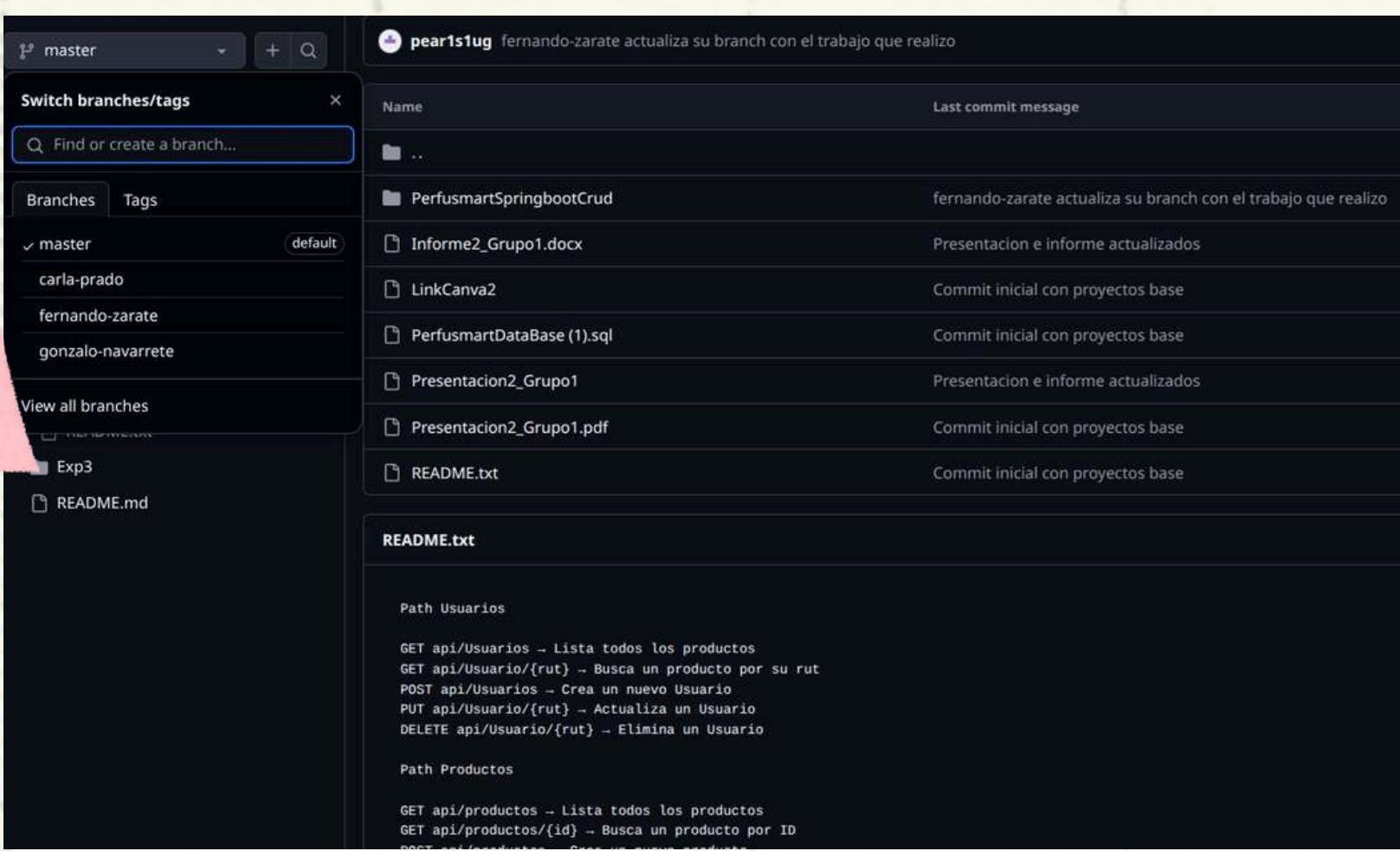


CORRECCIONES EXPERIENCIA 2

CONTROL DE VERSIONES

Estructura de la carpeta **Exp2** del repositorio

Historial de commits que muestra que cada integrante subió su trabajo por separado, en la branch que corresponde.



Commits

History for Exp3_Navarrete_Prado_Zarate / Exp2 on master

- o- Commits on Jun 23, 2025
 - fernando-zarate actualiza su branch con el trabajo que realizo

 pear1s1ug committed 10 hours ago
 - carla-prado añade su trabajo en exp2 desde su branch

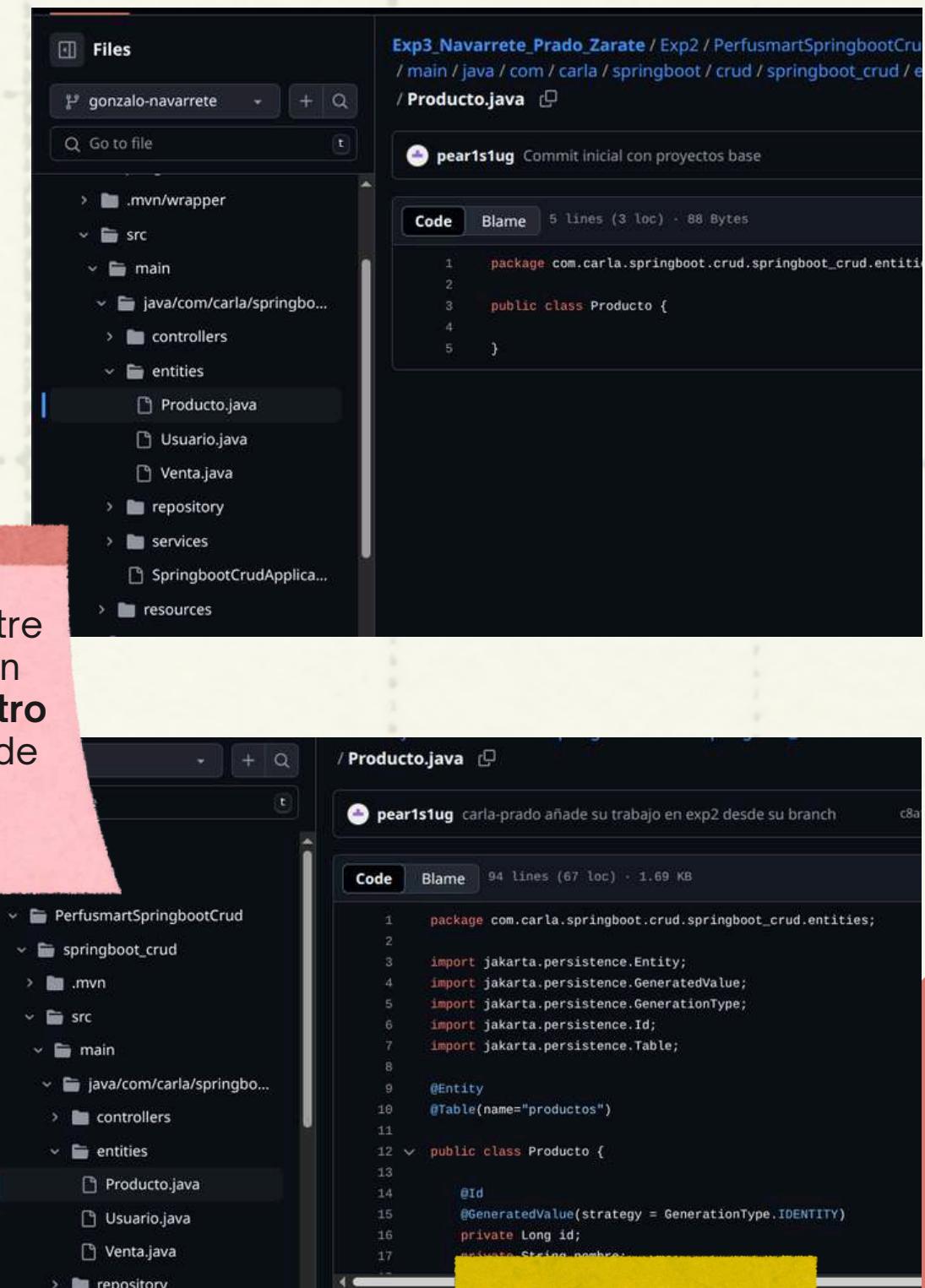
 pear1s1ug committed 10 hours ago
 - gonzalo-navarrete sube su trabajo desde su branch

 pear1s1ug committed 10 hours ago
 - Presentacion e informe actualizados

 pear1s1ug authored 10 hours ago
- o- Commits on Jun 22, 2025
 - Commit inicial con proyectos base

 pear1s1ug committed 19 hours ago
- o- End of commit history for this file

Diferencia entre un package en el **master** y otro en la **branch** de un integrante.



El proceso se describe en mayor detalle en Informe n°2.

CORRECCIONES EXPERIENCIA 2

MANEJO MÚLTIPLES BASES DE DATOS



Navigator

SCHEMAS

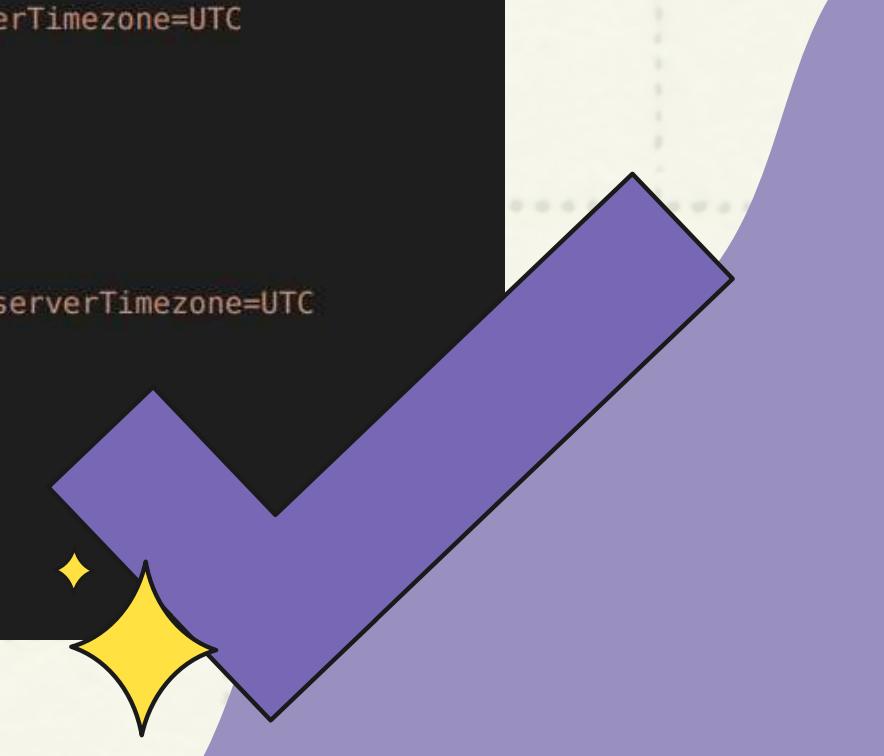
Filter objects

- db_cliente
 - Tables
 - Views
 - Stored Procedures
 - Functions
- db_logistica
 - Tables
 - Views
 - Stored Procedures
 - Functions
- db_producto
 - Tables
 - Views
 - Stored Procedures
 - Functions
- db_usuario
 - Tables
 - Views
 - Stored Procedures
 - Functions
- db_venta
 - Tables
 - Views
 - Stored Procedures
 - Functions

Bases de datos que contienen la tabla correspondiente a cada uno de los **cinco** microservicios.

```
application.properties 9+ X
springboot_crud > src > main > resources > application.properties
1 spring.application.name=springboot_crud
2 springdoc.swagger-ui.path=/swagger-ui.html
3
4 # DATASOURCE USUARIO
5 usuario.datasource.jdbc-url=jdbc:mysql://localhost:3306/db_usuario?serverTimezone=UTC
6 usuario.datasource.username=root
7 usuario.datasource.password=
8 usuario.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
9
10 # DATASOURCE PRODUCTO
11 producto.datasource.jdbc-url=jdbc:mysql://localhost:3306/db_producto?serverTimezone=UTC
12 producto.datasource.username=root
13 producto.datasource.password=
14 producto.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
15
16 # DATASOURCE VENTA
17 venta.datasource.jdbc-url=jdbc:mysql://localhost:3306/db_venta?serverTimezone=UTC
18 venta.datasource.username=root
19 venta.datasource.password=
20 venta.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
21
22 # DATASOURCE CLIENTE
23
24 cliente.datasource.jdbc-url=jdbc:mysql://localhost:3306/db_cliente?serverTimezone=UTC
25 cliente.datasource.username=root
26 cliente.datasource.password=
27 cliente.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
28
29
30 # DATASOURCE LOGISTICA
31 logistica.datasource.jdbc-url=jdbc:mysql://localhost:3306/db_logistica?serverTimezone=UTC
32 logistica.datasource.username=root
33 logistica.datasource.password=
34 logistica.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
35
36
37
38
```

application.properties
Contiene la configuración de la conexión a la base de datos correspondiente a la URL de cada microservicio.



CORRECCIONES EXPERIENCIA 2

MANEJO MÚLTIPLES BASES DE DATOS

Clase Config
requerida para
configurar
manualmente el
acceso a la base
de datos (cuando
son múltiples)

```
J ClienteConfig.java X

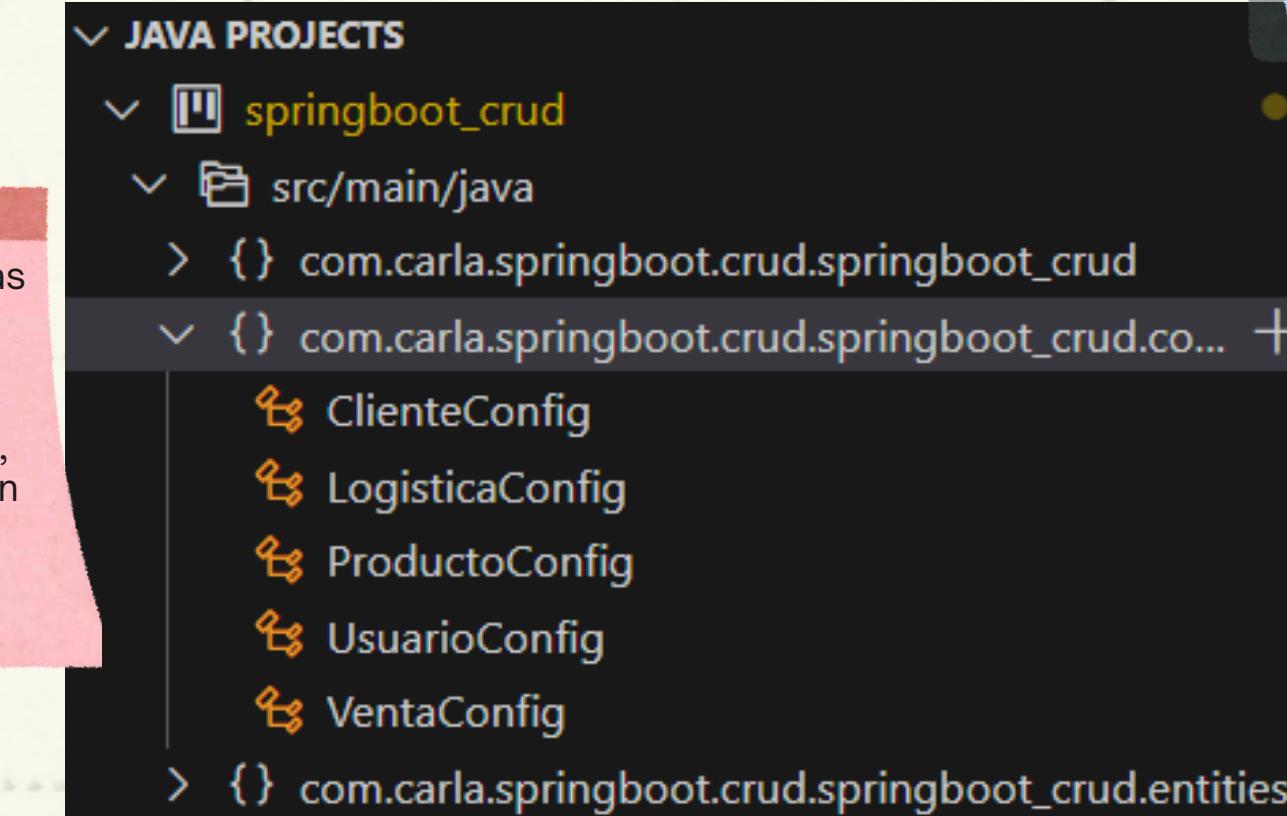
springboot_crud > src > main > java > com > carla > springboot > crud > springboot_crud > config > J ClienteConfig.java > Language Support for Java(T

1 package com.carla.springboot.crud.springboot_crud.config;
2
3 import javax.sql.DataSource;
4
5 import org.springframework.beans.factory.annotation.Qualifier;
6 import org.springframework.boot.context.properties.ConfigurationProperties;
7 import org.springframework.boot.jdbc.DataSourceBuilder;
8 import org.springframework.context.annotation.*;
9 import org.springframework.data.jpa.repository.config.EnableJpaRepositories;
10 import org.springframework.orm.jpa.*;
11 import org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter;
12 import org.springframework.transaction.PlatformTransactionManager;
13 import org.springframework.transaction.annotation.EnableTransactionManagement;
14
15 import jakarta.persistence.EntityManagerFactory;
16 @Configuration
17 @EnableTransactionManagement
18 @EnableJpaRepositories(
19     basePackages = "com.carla.springboot.crud.springboot_crud.repository.cliente",
20     entityManagerFactoryRef = "clienteEntityManager",
21     transactionManagerRef = "clienteTransactionManager"
22 )
23
24 public class ClienteConfig {
25
26     @Bean(name = "clienteDataSource")
27     @ConfigurationProperties(prefix = "cliente.datasource")
28     public DataSource dataSource() {
29         return DataSourceBuilder.create().build();
30     }
31
32     @Bean(name = "clienteEntityManager")
33     public LocalContainerEntityManagerFactoryBean entityManagerFactory(
34         @Qualifier("clienteDataSource") DataSource dataSource) {
35         LocalContainerEntityManagerFactoryBean em = new LocalContainerEntityManagerFactoryBean();
36         em.setDataSource(dataSource);
37         em.setPackagesToScan("com.carla.springboot.crud.springboot_crud.entities");
38         em.setJpaVendorAdapter(new HibernateJpaVendorAdapter());
39         em.setPersistenceUnitName("cliente");
40         return em;
41     }
42
43     @Bean(name = "clienteTransactionManager")
44     public PlatformTransactionManager transactionManager(
45         @Qualifier("clienteEntityManager") EntityManagerFactory emf) {
46         return new JpaTransactionManager(emf);
47     }
48
49 }
50 }
```

Se importan librerías relacionadas a JPA (Java Persistence API), jakarta.persistence, y otras que manejan acceso a base de datos y servicios web.

Data Source:
origen/configuración de base de datos.

EntityManagerFactory:
separar manejo de entidades



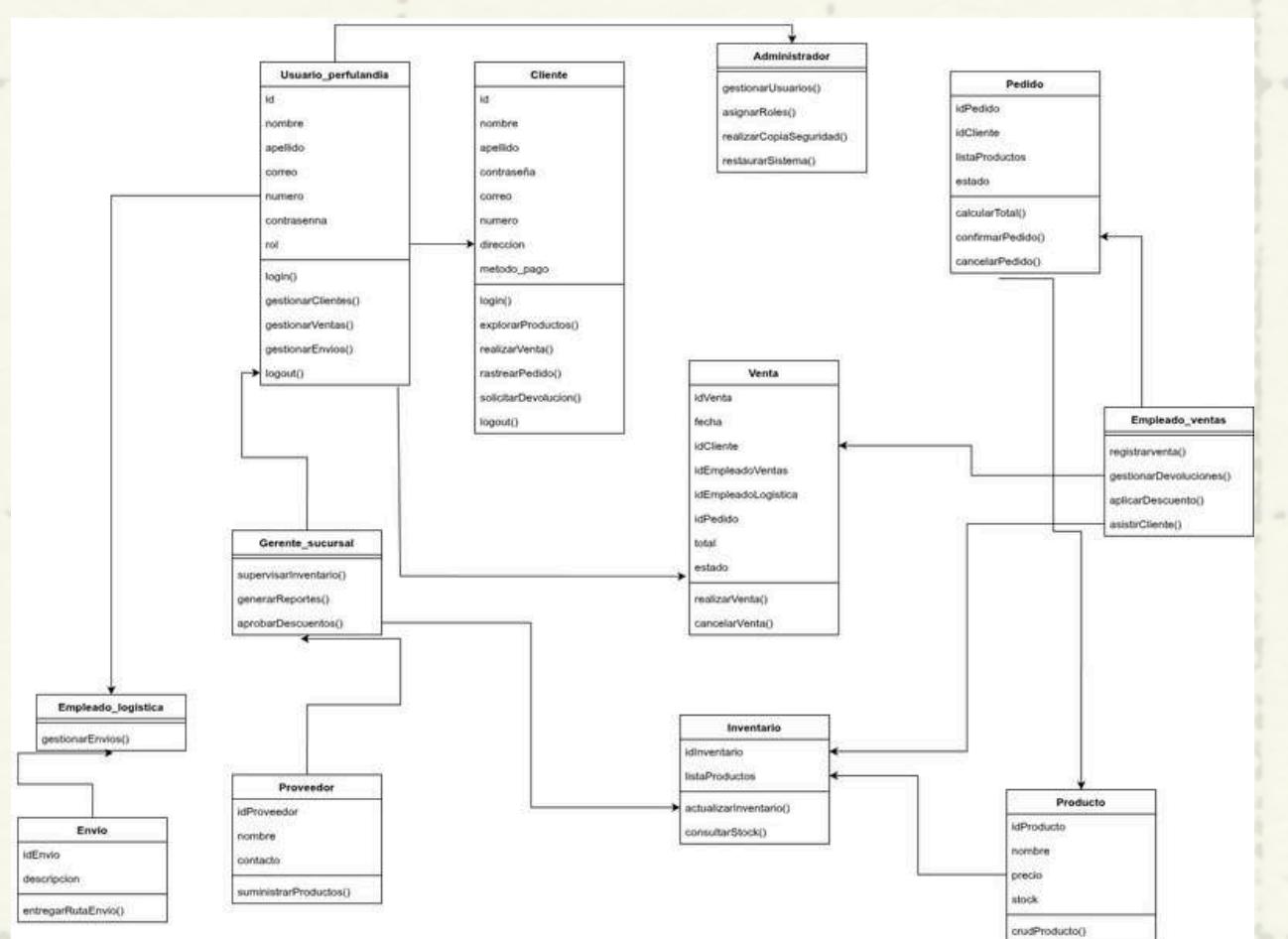
Cada microservicio tiene su propio archivo Config.

```
41     }
42
43     @Bean(name = "clienteTransactionManager")
44     public PlatformTransactionManager transactionManager(
45         @Qualifier("clienteEntityManager") EntityManagerFactory emf) {
46         return new JpaTransactionManager(emf);
47     }
48
49 }
50 }
```

TransactionManager:
separa gestión de transacciones

CORRECCIONES EXPERIENCIA 1

Diagrama de clases
Con estándares UML adecuados



Plan de migración
Método Strangler Fig (reemplazo gradual)

4. Planificación de la Migración

Plan de migración método Strangler Fig para Perfulandia SPA:

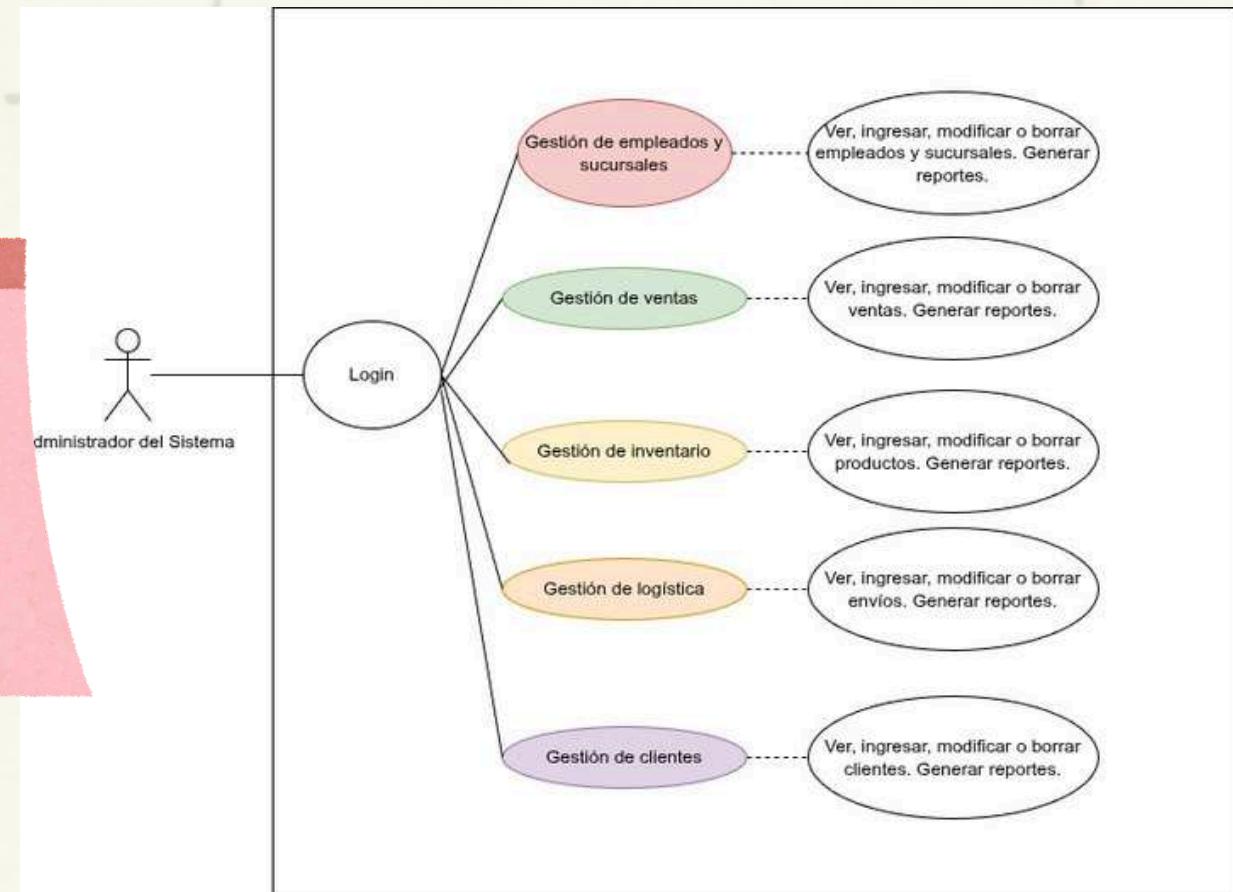
Fase 1: Preparación y Monitoreo Inicial

Duración estimada: 2-3 semanas

Acciones:

- Configurar infraestructura base (Kubernetes, Docker, RabbitMQ).
- Instalar y configurar Prometheus + Grafana.
- Auditar el sistema monolítico para identificar puntos de entrada (controladores REST, endpoints clave).
- Establecer un API Gateway que canalice todas las peticiones (p. ej., Zuul o Spring Cloud Gateway).

Diagramas de casos de uso
Contemplan Login



Todos los cambios registrados con commit en GitHub.

PLAN DE PRUEBAS



PLAN DE PRUEBAS

FRAMEWORKS UTILIZADOS

JUNIT 5:



Framework principal para **escribir y ejecutar pruebas**. Permite crearlas y automatizarlas de forma clara, por métodos separados.

- Su estructura:
 - **Preparación** (@BeforeEach)
 - **Ejecución** (Llamar al método a testear. Ej: int resultado = calculadora.sumar(2, 3);)
 - **Verificación** (assertions()).
- Pueden agruparse en **clases de prueba**.
- Se integra fácilmente con IDEs populares de Java (Eclipse, VSCode).
- Se utiliza en **pruebas unitarias y de integración**.

MOCKITO:

Framework que sirve para crear **objetos falsos** (simulados) llamados **Mocks** y decirles cómo deben comportarse.

- Permite comprobar que cada parte funcione por sí sola (métodos y clases).
- Se utiliza principalmente para **pruebas unitarias**.

PLAN DE PRUEBAS

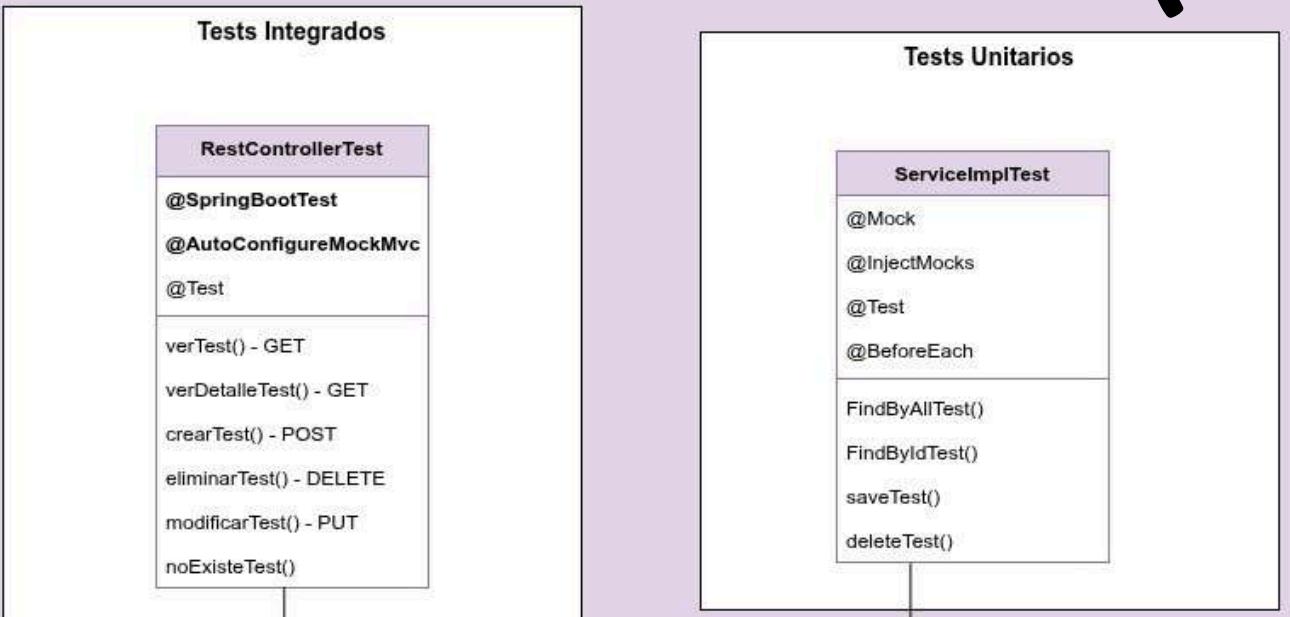
El entorno de pruebas está aislado del resto, y sirve como “mock” o simulación de alguna clase en particular.

Importancia de testear clase RestController

- Son el punto de entrada a la aplicación
- Responsables de recibir las solicitudes HTTP
- Se relaciona con el servicio y, a través de este, al repositorio (Integración).
- Un error puede hacer que la API no funcione aunque el servicio esté bien.

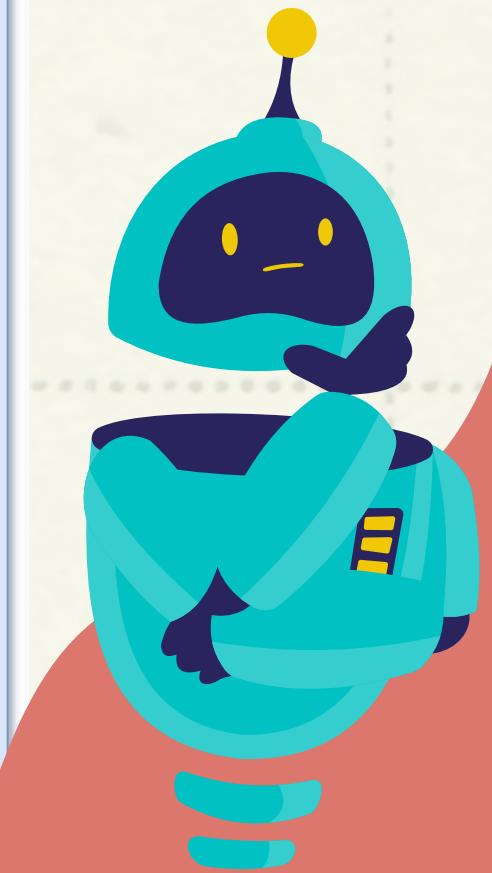
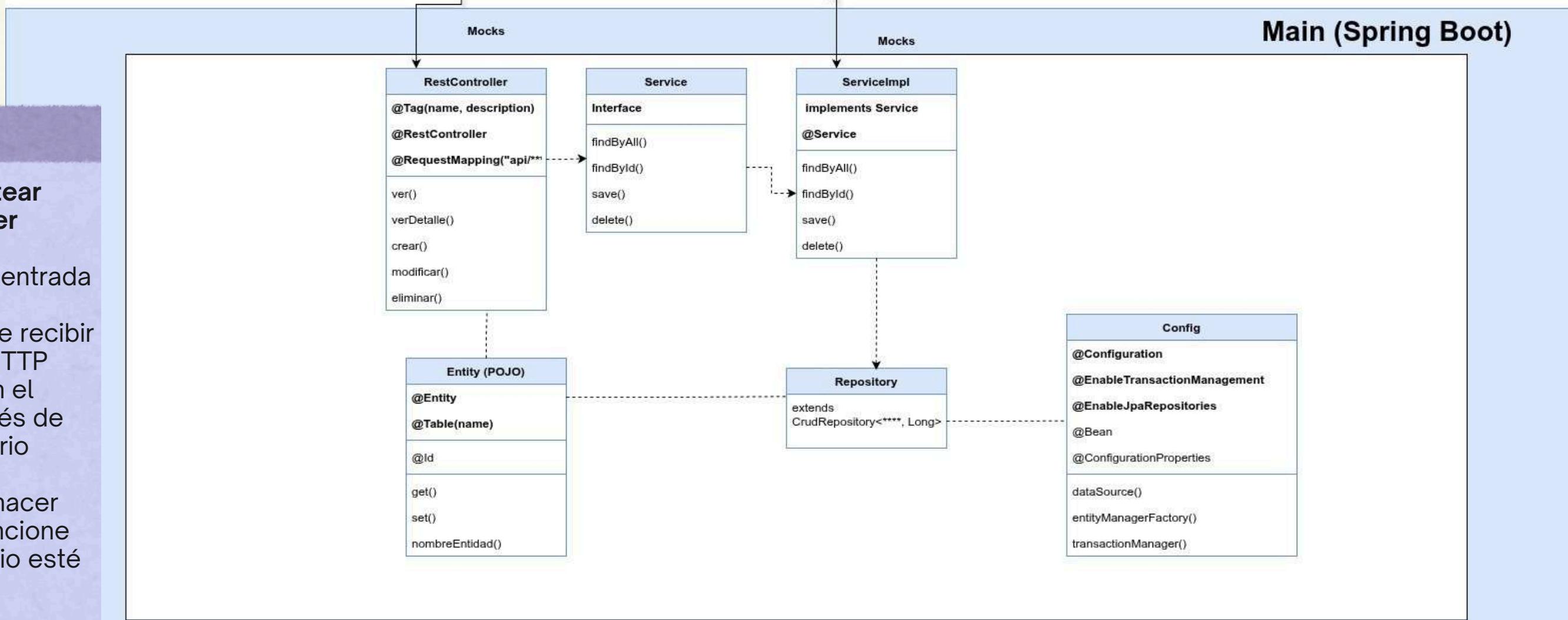
Tests (JUNIT + Mockito)

DIAGRAMA DE CLASES (GENERAL)



Importancia de testear clase Servicelmpl

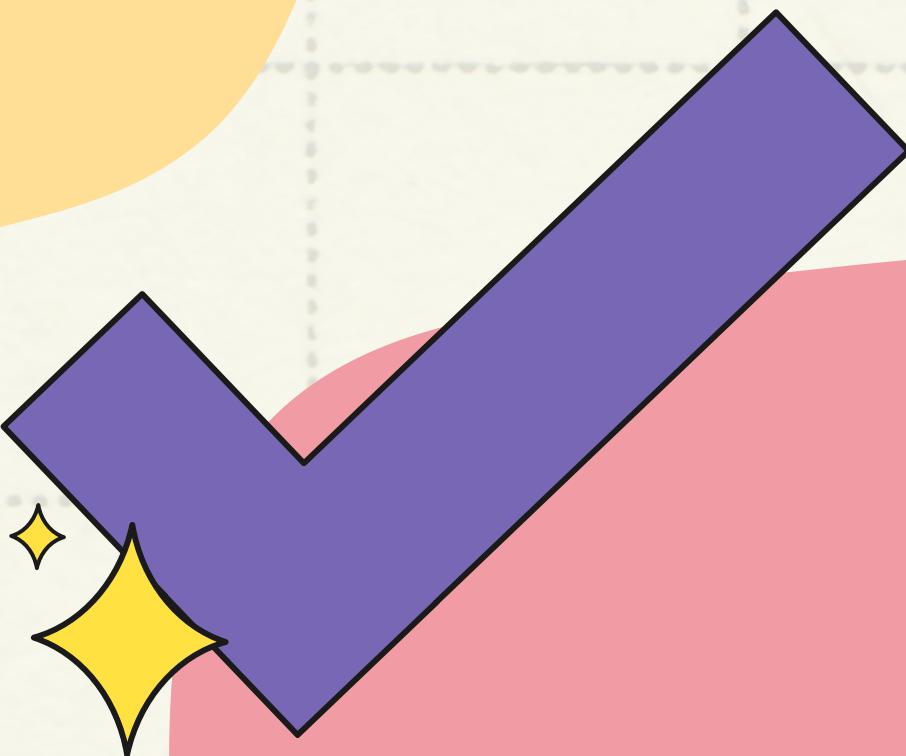
- Interactúa con otras, como repository.
- Al tener la lógica de negocio, puede causar errores en cascada si es que falla.
- Al no requerir base de datos para testear, puede hacerse rápido con pruebas unitarias.



FRAMEWORKS Y HERRAMIENTAS

- JUnit 5
- Mockito
- MockMvc
- ObjectMapper
- AutoConfigureMockMvc

PRUEBAS UNITARIAS



```
public class UsuarioServiceImplTest {  
    @InjectMocks  
    private UsuarioServiceImpl service;  
    @Mock  
    private UsuarioRepository repository;  
  
    List<Usuario> list=new ArrayList<Usuario>();  
    @BeforeEach  
    public void init(){  
        MockitoAnnotations.openMocks(this);  
  
        this.chargeUsuario();  
    }  
  
    @Test  
    public void UsuarioFindByAllTest(){  
        when(repository.findAll()).thenReturn(list);  
        List<Usuario>response=service.findByAll();  
  
        assertEquals(3, response.size());  
  
        verify(repository,times(1)).findAll();  
    }  
  
    @Test  
    public void UsuarioFindByIdTest(){  
  
        Usuario Usuario=new Usuario("21861087-0", "Gonzalo", "Navarrete", "passw0rd", "Empleado", "gonzalonavarrete@gmail.com", "+569 83022268");  
        when(repository.findById("21861087-0")).thenReturn(Optional.of(Usuario));  
  
        Optional<Usuario> response = service.findById("21861087-0");  
  
        assertTrue(response.isPresent());  
        assertEquals("Gonzalo", response.get().getNombre());  
        verify(repository, times(1)).findById("21861087-0");  
    }  
}
```

PRUEBAS UNITARIAS

```
@Test
public void modificarUsuarioTest() {
    Usuario UsuarioOriginal = new Usuario(rut:"21861087-0", nombre:"Nombre antiguo", apellido:"Apellido antiguo", contrasenna:"Pass antigua", rol:"Rol antiguo", correo:"Correo antiguo", ..."Numero antiguo");
    Usuario UsuarioModificado = new Usuario(rut:"21861087-0", nombre:"Nombre Modificado", apellido:"Apellido Modificado", contrasenna:"Pass Modificado", rol:"Rol Modificado", correo:"Correo Modificado", ..."Numero Modificado");

    when(repository.findById(id:"21861087-0")).thenReturn(Optional.of(UsuarioOriginal));
    when(repository.save(any(type:Usuario.class))).thenReturn(UsuarioModificado);

    Optional<Usuario> UsuarioBD = service.findById(id:"21861087-0");
    Usuario actualizado = null;
    if (UsuarioBD.isPresent()) {
        Usuario u = UsuarioBD.get();
        u.setNombre(nombre:"Nombre Modificado");
        u.setApellido(apellido:"Apellido Modificado");
        u.setcontrasenna(contrasenna:"Pass Modificado");
        u.setRol(rol:"Rol Modificado");
        u.setCorreo(correo:"Correo Modificado");

        actualizado = service.save(u);
    }

    assertNotNull(actualizado);
    assertEquals(expected:"Nombre Modificado", actualizado.getNombre());
    assertEquals(expected:"Apellido Modificado", actualizado.getApellido());
    assertEquals(expected:"Rol Modificado", actualizado.getRol());
    verify(repository).findById(id:"21861087-0");
    verify(repository).save(any(type:Usuario.class));
}

@Test
public void eliminarUsuarioTest() {

    Usuario UsuarioAEliminar = new Usuario(rut:"21861087-0", nombre:"Nombre a Eliminar", apellido:"Apellido a Eliminar", contrasenna:"Pass a Eliminar", rol:"Rol a Eliminar", correo:"Correo a Eliminar", ..."Numero a Eliminar");

    when(repository.findById(id:"21861087-0")).thenReturn(Optional.of(UsuarioAEliminar));

    Optional<Usuario> response = service.delete(UsuarioAEliminar);

    assertTrue(response.isPresent());
    assertEquals(expected:"Nombre a Eliminar", response.get().getNombre());

    verify(repository, times(wantedNumberOfInvocations:1)).findById(id:"21861087-0");
    verify(repository, times(wantedNumberOfInvocations:1)).delete(UsuarioAEliminar);
}
```

PRUEBAS UNITARIAS

```
public void chargeUsuario(){
    Usuario usuario1 =new Usuario(rut:"Rut 1", nombre:"Nombre 1", apellido:"Apellido 1", contrasenna:"Pass 1", rol:"Rol 1", correo:"Correo 1", ..."Numero 1");
    Usuario usuario2 =new Usuario(rut:"Rut 2", nombre:"Nombre 2", apellido:"Apellido 2", contrasenna:"Pass 2", rol:"Rol 2", correo:"Correo 2", ..."Numero 2");
    Usuario usuario3 =new Usuario(rut:"Rut 3", nombre:"Nombre 3", apellido:"Apellido 3", contrasenna:"Pass 3", rol:"Rol 3", correo:"Correo 3", ..."Numero 3");

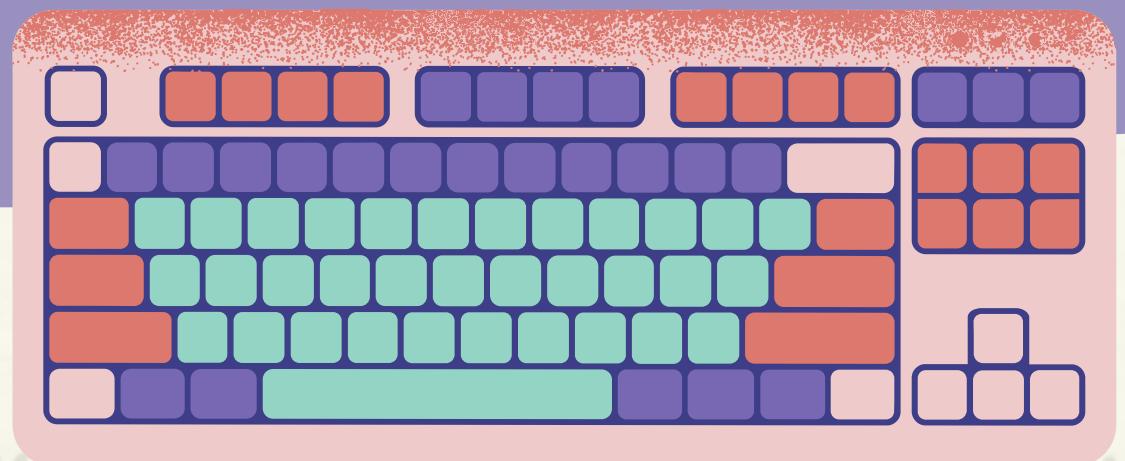
    list.add(usuario1);
    list.add(usuario2);
    list.add(usuario3);
}
```

PRUEBAS UNITARIAS

EJECUCIÓN

✓	⌚ { } com.carla.springboot.crud.springboot_crud.services 1.5s
✓	⌚ ⚡ ClienteServiceImplTest 48ms
✓	⌚ 📦 clienteFindByAllTest() 2.0ms
✓	⌚ 📦 clienteFindByIdTest() 4.0ms
✓	⌚ 📦 crearClienteTest() 35ms
✓	⌚ 📦 modificarClienteTest() 4.0ms
✓	⌚ 📦 eliminarClienteTest() 3.0ms
✓	⌚ ⚡ LogisticaServiceImplTest 49ms
✓	⌚ 📦 logisticaFindByAllTest() 3.0ms
✓	⌚ 📦 logisticaFindByIdTest() 4.0ms
✓	⌚ 📦 crearLogisticaTest() 3.0ms
✓	⌚ 📦 modificarLogisticaTest() 4.0ms
✓	⌚ 📦 eliminarLogisticaTest() 35ms
✓	⌚ ⚡ ProductoServiceImplTest 1.3s
✓	⌚ 📦 productoFindByAllTest() 5.0ms
✓	⌚ 📦 productoFindByIdTest() 1.2s
✓	⌚ 📦 crearProductoTest() 6.0ms
✓	⌚ 📦 modificarProductoTest() 8.0ms
✓	⌚ 📦 eliminarProductoTest() 5.0ms
✓	⌚ ⚡ UsuarioServiceImplTest 56ms
✓	⌚ 📦 UsuarioFindByAllTest() 8.0ms
✓	⌚ 📦 UsuarioFindByIdTest() 0.0ms
✓	⌚ 📦 crearUsuarioTest() 38ms
✓	⌚ 📦 modificarUsuarioTest() 5.0ms
✓	⌚ 📦 eliminarUsuarioTest() 5.0ms
✓	⌚ ⚡ VentaServiceImplTest 52ms
✓	⌚ 📦 VentaFindByAllTest() 3.0ms
✓	⌚ 📦 VentaFindByIdTest() 10ms
✓	⌚ 📦 crearVentaTest() 3.0ms
✓	⌚ 📦 modificarVentaTest() 0.0ms
✓	⌚ 📦 eliminarVentaTest() 36ms

PRUEBAS DE INTEGRACIÓN



PRUEBAS DE INTEGRACION

```
@SpringBootTest
@AutoConfigureMockMvc
public class ProductoRestControllerTest {

    @Autowired
    private MockMvc mockMvc;
    @Autowired
    private ObjectMapper objectMapper;
    @MockitoBean
    private ProductoServiceImpl productoServiceImpl;
    private List<Producto> productosLista;

    @Test
    public void verProductosTest() throws Exception{
        when(productoServiceImpl.findAll()).thenReturn(productosLista);
        mockMvc.perform(get(uriTemplate:"/api/productos")
            .contentType(MediaType.APPLICATION_JSON))
            .andExpect(status().isOk());
    }

    @Test
    public void verDetalleTest(){
        Producto unProducto=new Producto(id:1L, nombre:"La Vie Est Belle Eau de Parfum", precio:45000,marca:"Lancôme ",tipo:"De mujer",cantidad:50 );
        try{
            when(productoServiceImpl.findById(id:1L)).thenReturn(Optional.of(unProducto));
            mockMvc.perform(get(uriTemplate:"/api/productos/1")
                .contentType(MediaType.APPLICATION_JSON))
                .andExpect(status().isOk());
        }
        catch(Exception ex){
            fail("El testing lanza un error "+ ex.getMessage());
        }
    }

    @Test
    public void productoNoExisteTest() throws Exception{
        when(productoServiceImpl.findById(id:10L)).thenReturn(Optional.empty());
        mockMvc.perform(get(uriTemplate:"/api/productos/10")
            .contentType(MediaType.APPLICATION_JSON))
            .andExpect(status().isNotFound());
    }
}
```

PRUEBAS DE INTEGRACION

```
@Test
public void crearProductoTest() throws Exception{
    Producto unProducto=new Producto(id:null, nombre:"La Vie Est Belle Eau de Parfum 100ml",precio:100000,marca:"Lancôme ",tipo:"De mujer",cantidad:50);
    Producto otroProducto=new Producto(id:4L, nombre:"La Vie Est Belle Eau de Parfum 100ml", precio:100000,marca:"Lancôme ",tipo:"De mujer",cantidad:50);
    when(productoServiceImpl.save(any(type:Producto.class))).thenReturn(otroProducto);
    mockMvc.perform(post(uriTemplate:"/api/productos")
        .contentType(MediaType.APPLICATION_JSON)
        .content(objectMapper.writeValueAsString(unProducto)))
        .andExpect(status().isCreated());
}

@Test
public void eliminarProductoTest() throws Exception {
    Producto productoAEliminar = new Producto();
    productoAEliminar.setId(id:2L);

    when(productoServiceImpl.delete(any(type:Producto.class))).thenReturn(Optional.of(productoAEliminar));

    mockMvc.perform(delete(uriTemplate:"/api/productos/2")
        .contentType(MediaType.APPLICATION_JSON))
        .andExpect(status().isOk());
}

@Test
public void eliminarProductoNoExisteTest() throws Exception {
    when(productoServiceImpl.findById(id:99L)).thenReturn(Optional.empty());

    mockMvc.perform(delete(uriTemplate:"/api/productos/99"))
        .andExpect(status().isNotFound());
}
```

PRUEBAS DE INTEGRACION

```
@Test
public void modificarProductoExistenteTest() throws Exception {
    Long id = 1L;

    Producto productoExistente = new Producto(id, nombre:"Perfume Viejo", precio:40000, marca:"MarcaX", tipo:"De mujer", cantidad:10);
    Producto productoActualizado = new Producto(id, nombre:"Perfume Nuevo", precio:45000, marca:"MarcaY", tipo:"De mujer", cantidad:20);

    when(productoServiceImpl.findById(id)).thenReturn(Optional.of(productoExistente));
    when(productoServiceImpl.save(any(type:Producto.class))).thenReturn(productoActualizado);

    mockMvc.perform(put(uriTemplate:"/api/productos/{id}", id)
        .contentType(MediaType.APPLICATION_JSON)
        .content(objectMapper.writeValueAsString(productoActualizado)))
        .andExpect(status().isOk());
}

@Test
public void modificarProductoNoExisteTest() throws Exception {
    Long id = 999L;
    Producto producto = new Producto(id, nombre:"Desconocido", precio:12345, marca:"N/A", tipo:"N/A", cantidad:0);

    // Simular que no se encontró ningún producto con ese ID
    when(productoServiceImpl.findById(id)).thenReturn(Optional.empty());

    mockMvc.perform(put(uriTemplate:"/api/productos/{id}", id)
        .contentType(MediaType.APPLICATION_JSON)
        .content(objectMapper.writeValueAsString(producto)))
        .andExpect(status().isNotFound());
}
```

PRUEBAS DE INTEGRACION

EJECUCIÓN

The screenshot shows a test execution interface with a tree view of test cases and their execution times. The total execution time is 11.1s.

- springboot_crud 2.6s
 - com.carla.springboot.crud.springboot_crud.restcontrollers 1.2s
 - ClienteRestControllerTest 349ms
 - verClientesTest() 15ms
 - verDetalleTest() 12ms
 - clienteNoExisteTest() 16ms
 - crearClienteTest() 258ms
 - eliminarClienteTest() 12ms
 - eliminarClienteNoExisteTest() 8.0ms
 - modificarClienteExistenteTest() 9.0ms
 - modificarClienteNoExisteTest() 19ms
 - LogisticaRestControllerTest 363ms
 - verLogisticaTest() 11ms
 - verDatosTest() 258ms
 - productoNoExisteTest() 9.0ms
 - crearDatosTest() 8.0ms
 - eliminarDatosTest() 10ms
 - eliminarDatosNoExisteTest() 8.0ms
 - modificarDatosExistenteTest() 9.0ms
 - modificarDatosNoExisteTest() 50ms
 - ProductoRestControllerTest 70ms
 - verProductosTest() 14ms
 - verDetalleTest() 7.0ms
 - productoNoExisteTest() 8.0ms
 - crearProductoTest() 11ms
 - eliminarProductoTest() 7.0ms
 - eliminarProductoNoExisteTest() 8.0ms
 - modificarProductoExistenteTest() 7.0ms
 - modificarProductoNoExisteTest() 8.0ms
 - UsuarioRestControllerTest 273ms
 - verusuariosTest() 10ms
 - verDetalleTest() 10ms
 - UsuarioNoExisteTest() 9.0ms
 - crearUsuarioTest() 18ms
 - eliminarUsuarioTest() 7.0ms
 - eliminarUsuarioNoExisteTest() 17ms
 - modificarUsuarioExistenteTest() 9.0ms
 - modificarUsuarioNoExisteTest() 193ms

VentasRestControllerTest 114ms

 - verventasTest() 24ms
 - verDetalleTest() 11ms
 - VentaNoExisteTest() 8.0ms
 - crearVentaTest() 10ms
 - eliminarVentaTest() 11ms
 - eliminarVentaNoExisteTest() 9.0ms
 - modificarVentaExistenteTest() 14ms
 - modificarVentaNoExisteTest() 27ms



DOCUMENTACIÓN

DOCUMENTACIÓN DEPENDENCIAS UTILIZADAS

Proporciona una interfaz de usuario basada en Swagger (facilita construir APIs) para documentar y probar APIs REST. Se pueden visualizar los endpoints, parámetros y respuestas.



```
<dependency>
    <groupId>org.springdoc</groupId>
    <artifactId>springdoc-openapi-starter-webmvc-ui</artifactId>
    <version>2.8.7</version>
</dependency>
<dependency>
    <groupId>org.springdoc</groupId>
    <artifactId>springdoc-openapi-starter-webflux-ui</artifactId>
    <version>2.8.7</version>
</dependency>
```

Funcionalidad casi idéntica a la anterior, pero considera aplicaciones que usan Spring WebFlux, un sistema para manejar solicitudes http (optimizado para servidores modernos)

Estas dependencias de **Springdoc OpenAPI** son utilizadas para generar documentación automática de APIs REST en aplicaciones Spring.

DOCUMENTACIÓN

VISTA ENDPOINTS – CLASES REST CONTROLLER

Esta es una interfaz de documentación de una API generada por Springdoc OpenAPI.

clientes Usuarios sitio web ecommerce

GET	/api/clientes/{id}	Obtener cliente por ID
PUT	/api/clientes/{id}	Modificar cliente por ID
DELETE	/api/clientes/{id}	Eliminar cliente por ID
GET	/api/clientes	Obtener lista de clientes
POST	/api/clientes	Crear nuevo cliente

Logistica Proceso de logística por cada Logistica

GET	/api/logistica/{idEnvio}	Obtener Logistica por ID
PUT	/api/logistica/{idEnvio}	Modificar Logistica por ID
DELETE	/api/logistica/{idEnvio}	Eliminar datos por ID
GET	/api/logistica	Obtener lista de Logisticas
POST	/api/logistica	Crear nuevo Logistica

Ventas Operaciones relacionadas con ventas

GET	/api/ventas/{idVenta}	Obtener Venta por id
PUT	/api/ventas/{idVenta}	Modificar Venta por ID
DELETE	/api/ventas/{idVenta}	Eliminar Venta por ID
GET	/api/ventas	Obtener lista de Ventas
POST	/api/ventas	Ingresar nueva Venta

Esta documentación detalla los métodos de las clases RestController

Estos endpoints consisten de operaciones CRUD. Cada uno está asociado a un método HTTP.

DOCUMENTACIÓN

VISTA ENDPOINTS – CLASES REST CONTROLLER

Productos Operaciones relacionadas con productos

GET /api/productos/{id} Obtener producto por ID

PUT /api/productos/{id} Modificar producto por ID

DELETE /api/productos/{id} Eliminar producto por ID

GET /api/productos Obtener lista de productos

POST /api/productos Crear nuevo producto

Usuarios Operaciones relacionadas con Usuarios

GET /api/usuarios/{rut} Obtener usuario por RUT

PUT /api/usuarios/{rut} Modificar Usuario por ID

DELETE /api/usuarios/{rut} Eliminar usuario por ID

GET /api/usuarios Obtener lista de Usuarios

POST /api/usuarios Ingresar nuevo Usuario

Schemas

Venta > Expand all object

Producto > Expand all object

Usuario > Expand all object

Logistica > Expand all object

Cliente > Expand all object

Documentar los métodos de los RestControllers facilita el entendimiento por otros desarrolladores, facilitando la integración. Además, siempre es útil tener registros para monitorear cambios y evolución del código

DOCUMENTACIÓN

1. OBTENER USUARIO POR RUT

Parámetros requeridos
----->

Resuestas
----->

Usuarios Operaciones relacionadas con Usuarios

GET /api/usuarios/{rut} Obtener usuario por RUT

Obtiene el detalle de un Rut específico

Parameters

Name Description

rut * required string (path) rut

Try it out

Responses

Code	Description	Links
200	Usuario encontrado	No links
	Media type application/json	
	Controls Accept header	
	Example Value Schema	
	{ "id": 0, "nombre": "string", "precio": 0, "marca": "string", "tipo": "string", "cantidad": 0 }	
404	Usuario no encontrado	No links
	Media type */*	
	Example Value Schema	
	0	

Se muestra, como ejemplo, la documentación de los **endpoints asociados** a la entidad **Usuario**. Las demás tienen la misma estructura.

Códigos de estado:

- 200: (**Éxito**): Devuelve los detalles del usuario en formato JSON
- 404: (**No encontrado**): Se retorna si el usuario con el RUT especificado no existe.

DOCUMENTACIÓN

2. MODIFICAR DATOS DE UN USUARIO (RUT)

Incluye un botón "Try it out" para probar el endpoint ingresando el cuerpo JSON y visualizando la respuesta esperada.

PUT /api/usuarios/{rut} Modificar Usuario por ID

Actualiza un usuario existente con los nuevos datos proporcionados.

Parameters

Name	Description
rut * required	rut

Request body required

application/json

Example Value | Schema

```
{
  "rut": "string",
  "nombre": "string",
  "apellido": "string",
  "getcontraseña": "string",
  "rol": "string",
  "correo": "string",
  "numero": "string"
}
```

Responses

Code	Description	Links
200	Usuario modificado correctamente	No links
404	Usuario no encontrado	No links

Media type: application/json

Controls Accept header:

Example Value | Schema

```
{
  "id": 0,
  "nombre": "string",
  "precio": 0,
  "marca": "string",
  "tipo": "string",
  "cantidad": 0
}
```

Otros códigos de estado:

- 201 (Created): Funcionó un **POST**, se crea recurso nuevo.
- 400 (Bad Request): solicitud tiene **errores**
- 500 (Internal Server Error):

DOCUMENTACIÓN

3. ELIMINAR UN USUARIO (RUT)

DELETE /api/usuarios/{rut} Eliminar usuario por ID

Elimina un usuario específico si existe en la base de datos

Parameters

Name Description

rut * required string (path)

Responses

Code	Description	Links
200	Usuario eliminado correctamente	No links
	Media type	
	application/json	
	Controls Accept header.	
	Example Value Schema	
	<pre>{ "id": 0, "nombre": "string", "precio": 0, "marca": "string", "tipo": "string", "cantidad": 0 }</pre>	
404	Usuario no encontrado	No links

DOCUMENTACIÓN

4. OBTENER LISTA DE TODOS LOS USUARIOS

No requiere parámetros específicos

GET /api/usuarios Obtener lista de Usuarios

Devuelve todos los Usuarios Registrados

Parameters

No parameters.

Responses

Code	Description	Links
200	Lista de Usuarios retornada correctamente	No links

Media type

application/json

Controls Accept header

Example Value | Schema

```
{  
  "id": 0,  
  "nombre": "string",  
  "precio": 0,  
  "marca": "string",  
  "tipo": "string",  
  "cantidad": 0  
}
```

DOCUMENTACIÓN

5. INGRESAR UN NUEVO USUARIO

POST /api/usuarios Ingresar nuevo Usuario

ingresa un usuario con los datos proporcionados

Parameters

No parameters

Request body **required**

application/json

Example Value | Schema

```
{
  "rut": "string",
  "nombre": "string",
  "apellido": "string",
  "getcontrasenna": "string",
  "rol": "string",
  "correo": "string",
  "numero": "string"
}
```

Responses

Code	Description	Links
201	Usuario ingresado correctamente	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "id": 0,
  "nombre": "string",
  "precio": 0,
  "marca": "string",
  "tipo": "string",
  "cantidad": 0
}
```

Requiere un **cuerpo** en formato **JSON** con campos como **rut**, **nombre**, **apellido**, **contraseña**, etc.



CONTROL DE VERSIONES CON GIT Y GITHUB



CONTROL DE VERSIONES

VISTA DEL REPOSITORIO

Estructura
final
branch
master

The screenshot shows a GitHub repository named "Exp3_Navarrete_Prado_Zarate". The repository is public. The master branch is selected. There are 4 branches and 0 tags. A search bar for "Go to file" is present. The repository contains three main folders: Exp1, Exp2, and Exp3. The Exp1 folder has a commit from "pear1s1ug" titled "Commit inicial con proyectos base" made 17 hours ago. The Exp3 folder has two commits from "fernando-zarate": one titled "actualiza su branch con el trabajo que realizo" made 8 hours ago, and another titled "Update README.txt" made 9 hours ago.

The screenshot shows a "Switch branches/tags" dialog box. It includes a search bar for "Find or create a branch...". Below it, there are tabs for "Branches" and "Tags", with "Branches" currently selected. It lists several branches: "master" (marked with a checkmark and labeled "default"), "carla-prado", "fernando-zarate", and "gonzalo-navarrete". At the bottom of the dialog, there is a link "View all branches".

Branches de
cada
integrante

CONTROL DE VERSIONES

VISTA DEL REPOSITORIO

La carpeta de la Experiencia 3 contiene todos los entregables.



La branch **master** contiene un proyecto con todas las **clases creadas, pero sin código**, y el archivo pom y application.properties ya terminados

Luego, cada integrante sólo modifica las clases que le corresponden.

Name	Last commit message
...	
PerfusmartFinal	Merge branch 'gonzalo-navarrete'
DatabasePerfusmartFinal.sql	Update DatabasePerfusmartFinal.sql
Informe3_Grupo1.docx	Reemplazar documento plantilla informe
LinkCanva3	Commit inicial con proyectos base
Presentacion3_Grupo1.pdf	Commit inicial con proyectos base
README.txt	Update README.txt

README.txt

Path Usuarios

```
GET api/Usuarios → Lista todos los productos  
GET api/Usuario/{rut} → Busca un producto por su rut  
POST api/Usuarios → Crea un nuevo Usuario  
PUT api/Usuario/{rut} → Actualiza un Usuario  
DELETE api/Usuario/{rut} → Elimina un Usuario
```

CONTROL DE VERSIONES

COMMIT INICIAL DEL PROYECTO “BASE”

```
carrie@carrie-OMEN-by-HP-Laptop-16-c0xxx:/Desktop/Expl_Navarrete_Prado_Zarate $ git init
hint: Using 'master' as the name for the initial branch
hint: is subject to change. To configure the initial branch name to use in all
hints of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/carrie/Desktop/Expl_Navarrete_Prado_Zarate/.git/
carrie@carrie-OMEN-by-HP-Laptop-16-c0xxx:/Desktop/Expl_Navarrete_Prado_Zarate $ git remote add origin https://github.com/pearlslug/Expl_Navarrete_Prado_Zarate
carrie@carrie-OMEN-by-HP-Laptop-16-c0xxx:/Desktop/Expl_Navarrete_Prado_Zarate $ git add .
```

```
carrie@carrie-OMEN-by-HP-Laptop-16-c0xxx:/Desktop/Expl_Navarrete_Prado_Zarate $ git config --global user.name "pearlslug"
carrie@carrie-OMEN-by-HP-Laptop-16-c0xxx:/Desktop/Expl_Navarrete_Prado_Zarate $ git config --global user.email "ca.predom@duocucl.cl"
carrie@carrie-OMEN-by-HP-Laptop-16-c0xxx:/Desktop/Expl_Navarrete_Prado_Zarate $ git commit -m "Commit inicial con proyectos base"
[master (root-commit) 4cd0d1a] Commit inicial con proyectos base
 88 files changed, 343582 insertions(+)
 create mode 100644 Expl1/Informe1_Grupol.docx
 create mode 100644 Expl1/LinkCanva
 create mode 100644 Expl1/Presentacion1_Grupol.pdf
 create mode 100644 Exp2/Informe2_Grupol.docx
 create mode 100644 Exp2/LinkCanva2
 create mode 100644 Exp2/PerfumartDataBase_(1).sql
 create mode 100644 Exp2/PerfumartSpringbootCrud/.gitignore
 create mode 100644 Exp2/PerfumartSpringbootCrud/springboot_crud/.gitattributes
 create mode 100644 Exp2/PerfumartSpringbootCrud/springboot_crud/.gitignore
 create mode 100644 Exp2/PerfumartSpringbootCrud/springboot_crud/.mvn/wrapper/maven-wrapper.properties
 create mode 100644 Exp2/PerfumartSpringbootCrud/springboot_crud/mvnw
 create mode 100644 Exp2/PerfumartSpringbootCrud/springboot_crud/mvnw.cmd
 create mode 100644 Exp2/PerfumartSpringbootCrud/springboot_crud/pom.xml
 create mode 100644 Exp2/PerfumartSpringbootCrud/springboot_crud/src/main/java/com/caria/springboot/crud/springbootCrudApplication.java
 create mode 100644 Exp2/PerfumartSpringbootCrud/springboot_crud/src/main/java/com/caria/springboot/crud/springbootCrud/controllers/ProductoController.java
 create mode 100644 Exp2/PerfumartSpringbootCrud/springboot_crud/src/main/java/com/caria/springboot/crud/springboot_crud/controllers/UsuarioController.java
 create mode 100644 Exp2/PerfumartSpringbootCrud/springboot_crud/src/main/java/com/caria/springboot/crud/springboot_crud/controllers/VentaController.java
 create mode 100644 Exp2/PerfumartSpringbootCrud/springboot_crud/src/main/java/com/caria/springboot/crud/springboot_crud/entities/Producto.java
 create mode 100644 Exp2/PerfumartSpringbootCrud/springboot_crud/src/main/java/com/caria/springboot/crud/springboot_crud/entities/Usuario.java
 create mode 100644 Exp2/PerfumartSpringbootCrud/springboot_crud/src/main/java/com/caria/springboot/crud/springboot_crud/entities/Venta.java
 create mode 100644 Exp2/PerfumartSpringbootCrud/springboot_crud/src/main/java/com/caria/springboot/crud/springboot_crud/repository/ProductoRepository.java
 create mode 100644 Exp2/PerfumartSpringbootCrud/springboot_crud/src/main/java/com/caria/springboot/crud/springboot_crud/repository/UsuarioRepository.java
```

CONTROL DE VERSIONES

PUSH INICIAL DEL PROYECTO “BASE”

```
create mode 100644 Exp3/PerfumartFinal/springboot_crud/src/test/java/com/carla/springboot/crud/springboot_crud/restcontrollers/UsuarioRestController.java
create mode 100644 Exp3/PerfumartFinal/springboot_crud/src/test/java/com/carla/springboot/crud/springboot_crud/restcontrollers/VentasRestController.java
create mode 100644 Exp3/PerfumartFinal/springboot_crud/src/test/java/com/carla/springboot/crud/springboot_crud/services/ClienteServiceImplTest.java
create mode 100644 Exp3/PerfumartFinal/springboot_crud/src/test/java/com/carla/springboot/crud/springboot_crud/services/LogisticaServiceImplTest.java
create mode 100644 Exp3/PerfumartFinal/springboot_crud/src/test/java/com/carla/springboot/crud/springboot_crud/services/ProductoServiceImplTest.java
create mode 100644 Exp3/PerfumartFinal/springboot_crud/src/test/java/com/carla/springboot/crud/springboot_crud/services/UsuarioServiceImplTest.java
create mode 100644 Exp3/PerfumartFinal/springboot_crud/src/test/java/com/carla/springboot/crud/springboot_crud/services/VentaServiceImplTest.java
create mode 100644 Exp3/Presentacion3_Grupol.pdf
create mode 100644 Exp3/README.txt
carrie@carrie-OMEN-by-HP-Laptop-16-c0xxx:/Desktop/Exp3_Navarrete_Prado_Zarate$ git branch -M master
```

```
carrie@carrie-OMEN-by-HP-Laptop-16-c0xxx:/Desktop/Exp3_Navarrete_Prado_Zarate$ git push -u origin master
Username for 'https://github.com': pearlslug
Password for 'https://pearlslug@github.com':
Enumerating objects: 139, done.
Counting objects: 100% (139/139), done.
Delta compression using up to 16 threads
Compressing objects: 100% (113/113), done.
Writing objects: 100% (139/139), 79.77 MiB | 6.36 MiB/s, done.
Total 139 (delta 25), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (25/25), done.
remote: warning: See https://gh.io/lfs for more information.
remote: warning: File Exp3/Presentacion3_Grupol.pdf is 56.17 MB; this is larger than GitHub's recommended maximum file size of 50.00 MB
remote: warning: GH001: Large files detected. You may want to try Git Large File Storage - https://git-lfs.github.com.
To https://github.com/pearlslug/Exp3_Navarrete_Prado_Zarate
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

origin: Es el nombre del repositorio remoto al que se envían los cambios (por defecto, el remoto principal).

Este push (envío de datos a repositorio), incluye una relación de seguimiento (los cambios locales se implementan en el repositorio remoto)

CONTROL DE VERSIONES

CREACIÓN BRANCHES DE CADA INTEGRANTE

```
carrie@carrie-OMEN-by-HP-Laptop-16-c0xxx:~/Desktop/Exp3_Navarrete_Prado_Zarate $ git checkout -b gonzalo-navarrete
Switched to a new branch 'gonzalo-navarrete'
carrie@carrie-OMEN-by-HP-Laptop-16-c0xxx:~/Desktop/Exp3_Navarrete_Prado_Zarate $ git push -u origin gonzalo-navarrete
Username for 'https://github.com': pearlslug
Password for 'https://pearlslug@github.com':
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'gonzalo-navarrete' on GitHub by visiting:
remote:     https://github.com/pearlslug/Exp3_Navarrete_Prado_Zarate/pull/new/gonzalo-navarrete
remote:
To https://github.com/pearlslug/Exp3_Navarrete_Prado_Zarate
 * [new branch]      gonzalo-navarrete -> gonzalo-navarrete
branch 'gonzalo-navarrete' set up to track 'origin/gonzalo-navarrete'.
```

Checkout: se posiciona en la branch y la crea de no existir

```
carrie@carrie-OMEN-by-HP-Laptop-16-c0xxx:~/Desktop/Exp3_Navarrete_Prado_Zarate $ git push -u origin 'carla-prado'
Username for 'https://github.com': pearlslug
Password for 'https://pearlslug@github.com':
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'carla-prado' on GitHub by visiting:
remote:     https://github.com/pearlslug/Exp3_Navarrete_Prado_Zarate/pull/new/carla-prado
remote:
To https://github.com/pearlslug/Exp3_Navarrete_Prado_Zarate
 * [new branch]      carla-prado -> carla-prado
branch 'carla-prado' set up to track 'origin/carla-prado'.
carrie@carrie-OMEN-by-HP-Laptop-16-c0xxx:~/Desktop/Exp3_Navarrete_Prado_Zarate $ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
carrie@carrie-OMEN-by-HP-Laptop-16-c0xxx:~/Desktop/Exp3_Navarrete_Prado_Zarate $ git checkout -b fernando-zarate
Switched to a new branch 'fernando-zarate'
carrie@carrie-OMEN-by-HP-Laptop-16-c0xxx:~/Desktop/Exp3_Navarrete_Prado_Zarate $ git push -u origin fernando-zarate
Username for 'https://github.com': pearlslug
Password for 'https://pearlslug@github.com':
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'fernando-zarate' on GitHub by visiting:
remote:     https://github.com/pearlslug/Exp3_Navarrete_Prado_Zarate/pull/new/fernando-zarate
remote:
To https://github.com/pearlslug/Exp3_Navarrete_Prado_Zarate
 * [new branch]      fernando-zarate -> fernando-zarate
```

CONTROL DE VERSIONES

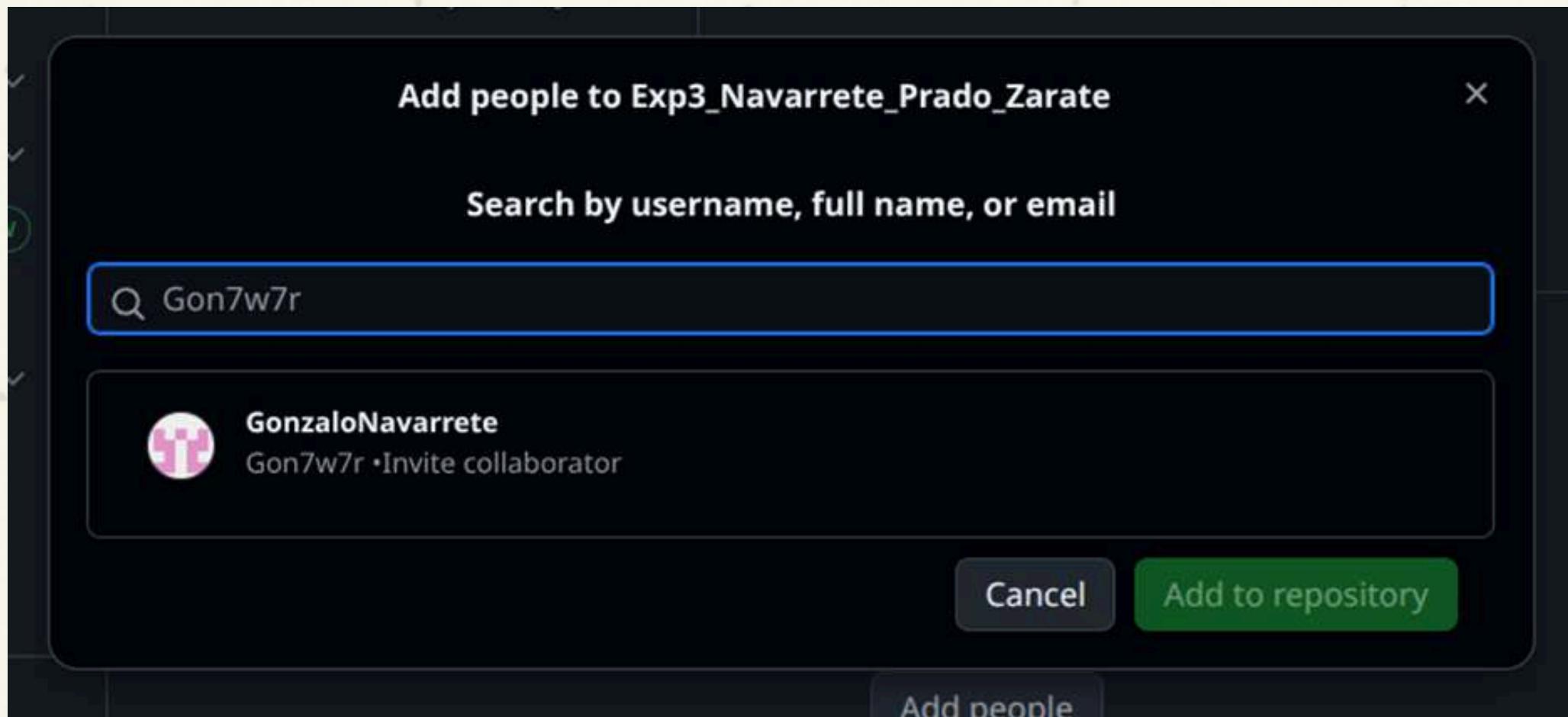
PUSH DEL TRABAJO INDIVIDUAL DESDE UNA BRANCH

```
carrie@carrie-OMEN-by-HP-Laptop-16-c0xxx:~/Desktop/Exp3_Navarrete_Prado_Zarate$ cp -r ~/Desktop/Branches/carla-prado/PerfusmartFinal/* \
~/Desktop/Exp3_Navarrete_Prado_Zarate/Exp3/PerfusmartFinal/
carrie@carrie-OMEN-by-HP-Laptop-16-c0xxx:~/Desktop/Exp3_Navarrete_Prado_Zarate$ git add Exp3/PerfusmartFinal/
carrie@carrie-OMEN-by-HP-Laptop-16-c0xxx:~/Desktop/Exp3_Navarrete_Prado_Zarate$ git commit -m "Commit: Se agrego trabajo Carla Prado correspondiente a Exp3"
[carla-prado b60f411] Commit: Se agrego trabajo Carla Prado correspondiente a Exp3
 8 files changed, 573 insertions(+), 8 deletions(-)
carrie@carrie-OMEN-by-HP-Laptop-16-c0xxx:~/Desktop/Exp3_Navarrete_Prado_Zarate$ git push
Username for 'https://github.com': pearlslug
Password for 'https://pearlslug@github.com':
Enumerating objects: 64, done.
Counting objects: 100% (64/64), done.
Delta compression using up to 16 threads
Compressing objects: 100% (25/25), done.
Writing objects: 100% (36/36), 8.14 KiB | 2.71 MiB/s, done.
Total 36 (delta 9), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (9/9), completed with 7 local objects.
To https://github.com/pearlslug/Exp3_Navarrete_Prado_Zarate
carrie@carrie-OMEN-By-HP-Laptop-16-c0xxx:~/Desktop/Exp3_Navarrete_Prado_Zarate$
```

```
carrie@carrie-OMEN-by-HP-Laptop-16-c0xxx:~/Desktop/Exp3_Navarrete_Prado_Zarate$ git checkout fernando-zarate
Switched to branch 'fernando-zarate'
Your branch is up to date with 'origin/fernando-zarate'.
carrie@carrie-OMEN-by-HP-Laptop-16-c0xxx:~/Desktop/Exp3_Navarrete_Prado_Zarate$ cp -r ~/Desktop/Branches/fernando-zarate/PerfusmartFinal/* \
~/Desktop/Exp3_Navarrete_Prado_Zarate/Exp3/PerfusmartFinal/
carrie@carrie-OMEN-by-HP-Laptop-16-c0xxx:~/Desktop/Exp3_Navarrete_Prado_Zarate$ git add Exp3/PerfusmartFinal/
carrie@carrie-OMEN-by-HP-Laptop-16-c0xxx:~/Desktop/Exp3_Navarrete_Prado_Zarate$ git commit -m "Commit: Se agrega trabajo de Fernando Zarate a Exp3"
[fernando-zarate 830f9b5] Commit: Se agrega trabajo de Fernando Zarate a Exp3
 8 files changed, 675 insertions(+), 10 deletions(-)
carrie@carrie-OMEN-by-HP-Laptop-16-c0xxx:~/Desktop/Exp3_Navarrete_Prado_Zarate$ git push
Username for 'https://github.com': pearlslug
Password for 'https://pearlslug@github.com':
Enumerating objects: 64, done.
Counting objects: 100% (64/64), done.
Delta compression using up to 16 threads
Compressing objects: 100% (25/25), done.
Writing objects: 100% (36/36), 8.29 KiB | 2.76 MiB/s, done.
Total 36 (delta 9), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (9/9), completed with 8 local objects.
To https://github.com/pearlslug/Exp3_Navarrete_Prado_Zarate
 4cdcdada..830f9b5  fernando-zarate -> fernando-zarate
carrie@carrie-OMEN-by-HP-Laptop-16-c0xxx:~/Desktop/Exp3_Navarrete_Prado_Zarate$
```

CONTROL DE VERSIONES

DAR PERMISOS DE COLABORADOR Y REALIZAR PUSH DESDE OTRO PC



```
MINGW64:/c/Users/Chalo/Desktop/Exp3_Navarrete_Prado_Zarate
Chalo@DESKTOP-2106TE3 MINGW64 ~/Desktop
$ git clone https://github.com/pearlslug/Exp3_Navarrete_Prado_Zarate
Cloning into 'Exp3_Navarrete_Prado_Zarate'...
remote: Enumerating objects: 211, done.
remote: Counting objects: 100% (72/72), done.
remote: Compressing objects: 100% (48/48), done.
remote: Total 211 (delta 15), reused 50 (delta 2), pack-reused 139 (from 1)
Receiving objects: 100% (211/211), 79.78 MiB | 18.75 MiB/s, done.
Resolving deltas: 100% (40/40), done.

Chalo@DESKTOP-2106TE3 MINGW64 ~/Desktop
$ cd Exp3_Apellido1_Apellido2_Apellido3
bash: cd: Exp3_Apellido1_Apellido2_Apellido3: No such file or directory

Chalo@DESKTOP-2106TE3 MINGW64 ~/Desktop
$ cd Exp3_Navarrete_Prado_Zarate

Chalo@DESKTOP-2106TE3 MINGW64 ~/Desktop/Exp3_Navarrete_Prado_Zarate (master)
$ git checkout gonzalo-navarrete
branch 'gonzalo-navarrete' set up to track 'origin/gonzalo-navarrete'.
Switched to a new branch 'gonzalo-navarrete'
```

CONTROL DE VERSIONES

DAR PERMISOS DE COLABORADOR Y REALIZAR PUSH DESDE OTRO PC

```
Chalo@DESKTOP-2106TE3 MINGW64 ~/Desktop/Exp3_Navarrete_Prado_Zarate (gonzalo-navarrete)
$ cp -r ~/Desktop/Github/PerfusmartFinal/* ~/Desktop/Exp3_Navarrete_Prado_Zarate/Exp3/PerfusmartFinal/
Chalo@DESKTOP-2106TE3 MINGW64 ~/Desktop/Exp3_Navarrete_Prado_Zarate (gonzalo-navarrete)
$ git add Exp3/PerfusmartFinal/
warning: in the working copy of 'Exp3/PerfusmartFinal/springboot_crud/.gitattributes', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Exp3/PerfusmartFinal/springboot_crud/.gitignore', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Exp3/PerfusmartFinal/springboot_crud/.mvn/wrapper/maven-wrapper.properties', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Exp3/PerfusmartFinal/springboot_crud/pom.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Exp3/PerfusmartFinal/springboot_crud/src/main/java/com/carla/springboot/crud/springboot_crud/SpringbootCrudApplication.java', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Exp3/PerfusmartFinal/springboot_crud/src/main/java/com/carla/springboot/crud/springboot_crud/entities/Producto.java', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Exp3/PerfusmartFinal/springboot_crud/src/main/java/com/carla/springboot/crud/springboot_crud/entities/Usuario.java', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Exp3/PerfusmartFinal/springboot_crud/src/main/java/com/carla/springboot/crud/springboot_crud/entities/Venta.java', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Exp3/PerfusmartFinal/springboot_crud/src/main/java/com/carla/springboot/crud/springboot_crud/repository/ProductoRepository.java', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Exp3/PerfusmartFinal/springboot_crud/src/main/java/com/carla/springboot/crud/springboot_crud/repository/usuario/UsuarioRepository.java', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Exp3/PerfusmartFinal/springboot_crud/src/main/java/com/carla/springboot/crud/springboot_crud/repository/venta/VentaRepository.java', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Exp3/PerfusmartFinal/springboot_crud/src/main/java/com/carla/springboot/crud/springboot_crud/restcontrollers/ProductoRestController.java', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Exp3/PerfusmartFinal/springboot_crud/src/main/java/com/carla/springboot/crud/springboot_crud/restcontrollers/UsuarioRestController.java', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Exp3/PerfusmartFinal/springboot_crud/src/main/java/com/carla/springboot/crud/springboot_crud/restcontrollers/VentaRestController.java', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Exp3/PerfusmartFinal/springboot_crud/src/main/java/com/carla/springboot/crud/springboot_crud/services/ProductoService.java', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Exp3/PerfusmartFinal/springboot_crud/src/main/java/com/carla/springboot/crud/springboot_crud/services/ProductoServiceImpl.java', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Exp3/PerfusmartFinal/springboot_crud/src/main/java/com/carla/springboot/crud/springboot_crud/services/UsuarioService.java', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Exp3/PerfusmartFinal/springboot_crud/src/main/java/com/carla/springboot/crud/springboot_crud/services/UsuarioServiceImpl.java', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Exp3/PerfusmartFinal/springboot_crud/src/main/java/com/carla/springboot/crud/springboot_crud/services/VentaService.java', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Exp3/PerfusmartFinal/springboot_crud/src/main/java/com/carla/springboot/crud/springboot_crud/services/VentaServiceImpl.java', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Exp3/PerfusmartFinal/springboot_crud/src/main/resources/application.properties', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Exp3/PerfusmartFinal/springboot_crud/src/test/java/com/carla/springboot/crud/springboot_crud/SpringbootCrudApplicationTests.java', LF will be replaced by CRLF the next time Git touches it
```

```
Chalo@DESKTOP-2106TE3 MINGW64 ~/Desktop/Exp3_Navarrete_Prado_Zarate (gonzalo-navarrete)
$ git config --global user.email "gon.navarrete@duocuc.cl"

Chalo@DESKTOP-2106TE3 MINGW64 ~/Desktop/Exp3_Navarrete_Prado_Zarate (gonzalo-navarrete)
$ git config --global user.name "Gon7w7r"

Chalo@DESKTOP-2106TE3 MINGW64 ~/Desktop/Exp3_Navarrete_Prado_Zarate (gonzalo-navarrete)
$ git commit -m "gon-navarrete: implementó su trabajo completo en exp3"
[gonzalo-navarrete 778c3c8] gon-navarrete: implementó su trabajo completo en exp3
 24 files changed, 1765 insertions(+), 40 deletions(-)

Chalo@DESKTOP-2106TE3 MINGW64 ~/Desktop/Exp3_Navarrete_Prado_Zarate (gonzalo-navarrete)
$ git push
Enumerating objects: 90, done.
Counting objects: 100% (90/90), done.
Delta compression using up to 12 threads
Compressing objects: 100% (42/42), done.
Writing objects: 100% (54/54), 14.87 KiB | 1.24 MiB/s, done.
Total 54 (delta 19), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (19/19), completed with 6 local objects.
To https://github.com/pearis1ug/Exp3_Navarrete_Prado_Zarate
 4cdcdda..778c3c8 gonzalo-navarrete -> gonzalo-navarrete
```

CONCLUSIÓN

En resumen, esta experiencia no solo reforzó aspectos **técnicos** como el **fortalecimiento del código y el testeo**, sino también habilidades como el **trabajo colaborativo**, la **organización de versiones** y la **entrega profesional de proyectos**.

Se aplican los **conocimientos** adquiridos en torno a **microservicios, pruebas y buenas prácticas** de documentación.

Se comprende la importancia de **aislar responsabilidades** y asegurar que cada **componente funcione** tanto de forma **individual como en conjunto**.





¡GRACIAS!

