

# COMUNICACION ENTRE MICROSERVICIOS SINCRONICA SPRING BOOT



Jesús Estenllos Loaiza Serrano 2313021

Desarrollo de Software III

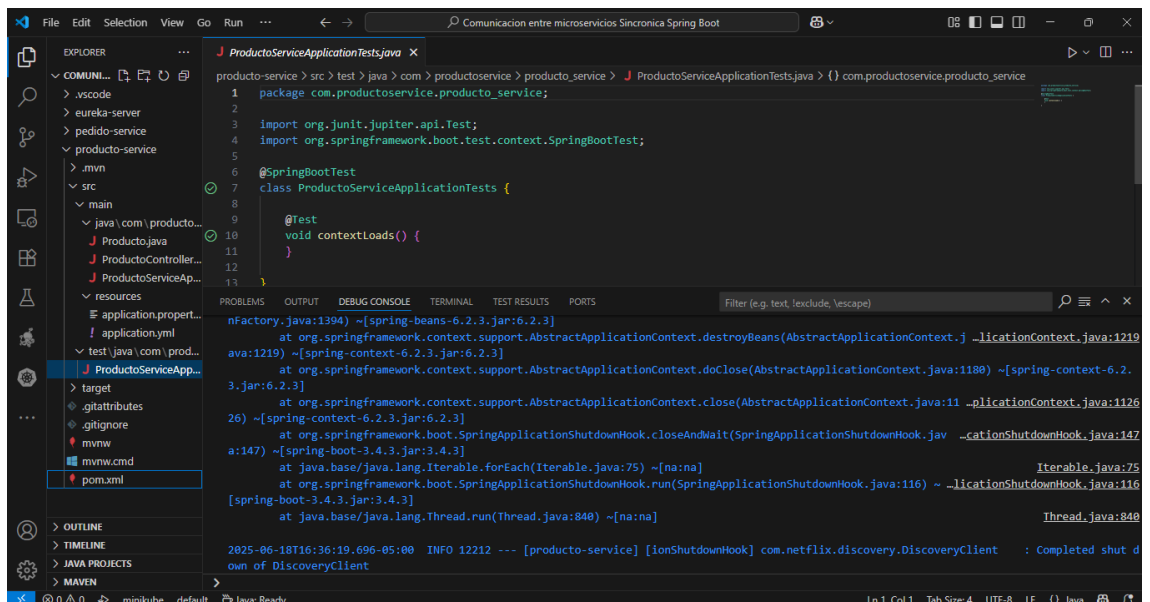
Docente: Ing. Álvaro Salazar

Grupo 50

Sede Tuluá

17/06/2025

## Test servicios\_producto service test

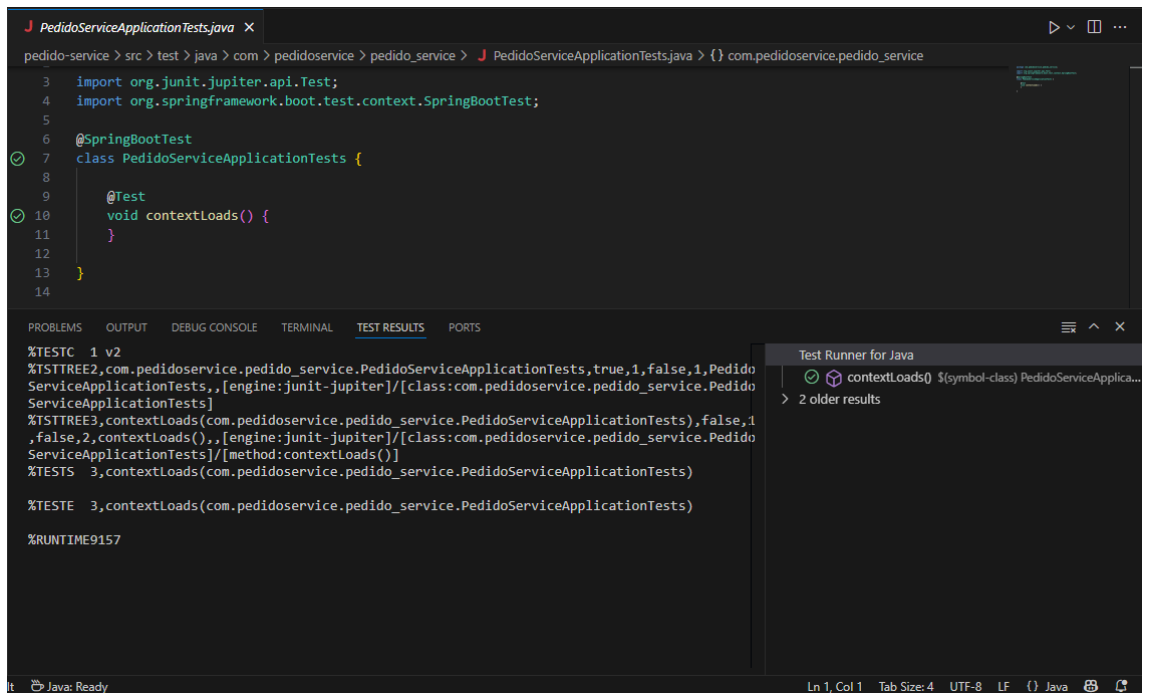


The screenshot shows an IDE with the file explorer on the left displaying the project structure. The main editor shows the source code of `ProductoServiceApplicationTests.java`. The code is as follows:

```
1 package com.productoservice.producto_service;
2
3 import org.junit.jupiter.api.Test;
4 import org.springframework.boot.test.context.SpringBootTest;
5
6 @SpringBootTest
7 class ProductoServiceApplicationTests {
8
9     @Test
10     void contextLoads() {
11     }
12 }
13
```

The terminal at the bottom shows the output of the test execution, including the classpath and the successful completion of the test.

```
2025-06-18T16:36:19.696-05:00 INFO 12212 --- [producto-service] [ionShutdownHook] com.netflix.discovery.DiscoveryClient : Completed shut d
own of DiscoveryClient
```

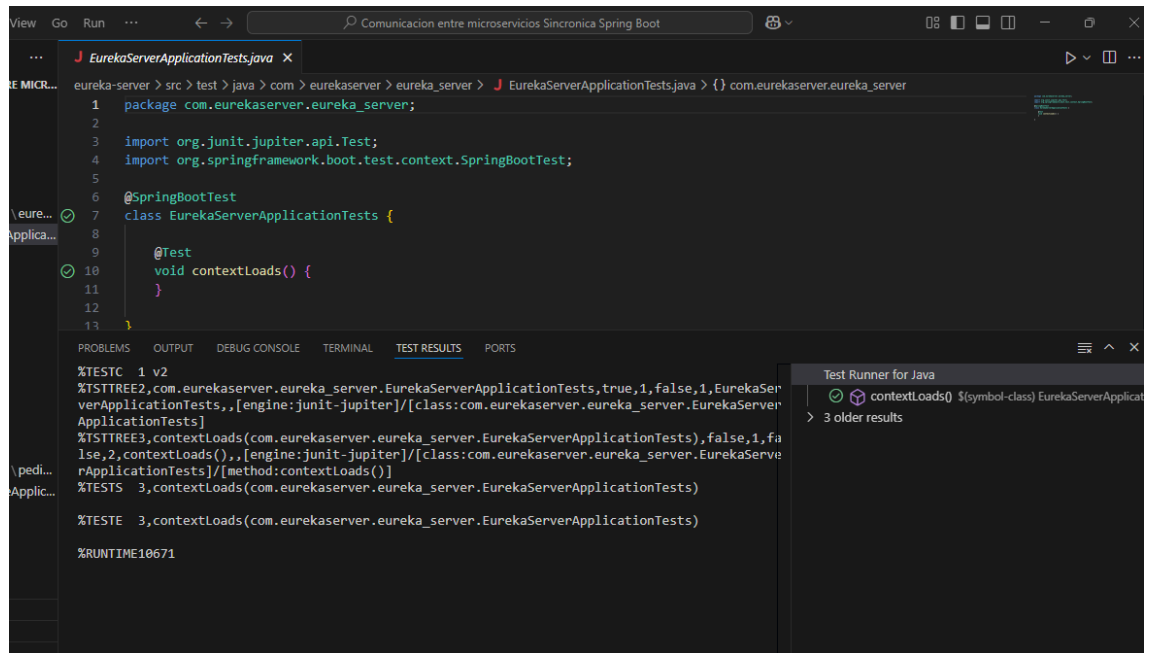


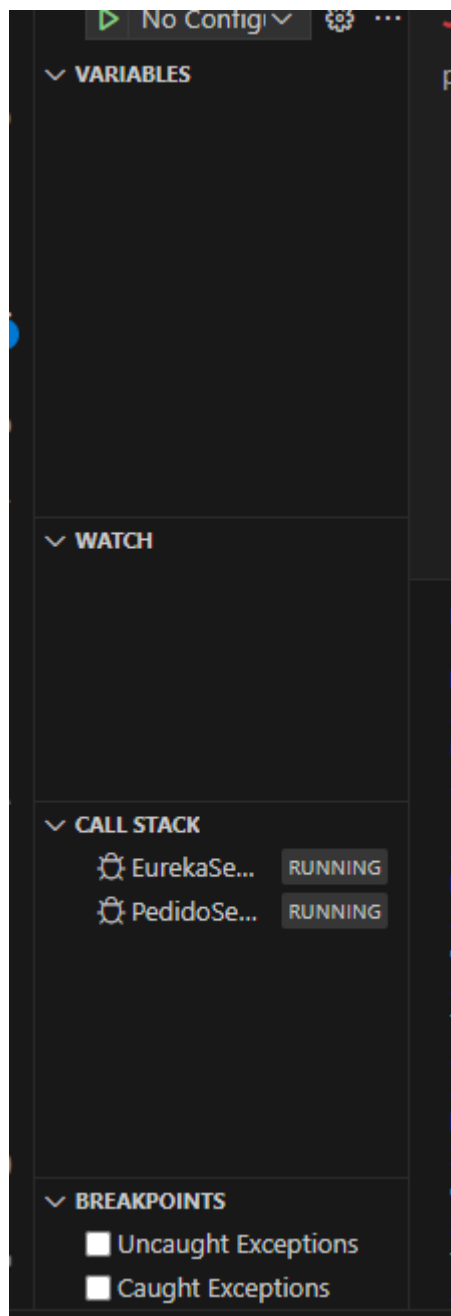
The screenshot shows an IDE with the file explorer on the left displaying the project structure. The main editor shows the source code of `PedidoServiceApplicationTests.java`. The code is as follows:

```
3 import org.junit.jupiter.api.Test;
4 import org.springframework.boot.test.context.SpringBootTest;
5
6 @SpringBootTest
7 class PedidoServiceApplicationTests {
8
9     @Test
10     void contextLoads() {
11     }
12 }
13
14
```

The Test Results window at the bottom shows the test execution results, including the test name, the test method, and the test status.

```
%TESTC 1 v2
%TSTTREE2,com.pedidoservice.pedido_service.PedidoServiceApplicationTests,true,1,false,1,Pedido
ServiceApplicationTests,,[engine:junit-jupiter]/[class:com.pedidoservice.pedido_service.Pedido
ServiceApplicationTests]
%TSTTREE3,contextLoads(com.pedidoservice.pedido_service.PedidoServiceApplicationTests),false,1
,false,2,contextLoads(),,[engine:junit-jupiter]/[class:com.pedidoservice.pedido_service.Pedido
ServiceApplicationTests]/[method:contextLoads()]
%TESTS 3,contextLoads(com.pedidoservice.pedido_service.PedidoServiceApplicationTests)
%TESTE 3,contextLoads(com.pedidoservice.pedido_service.PedidoServiceApplicationTests)
%RUNTIME9157
```





The screenshot shows an IDE with the file `EurekaServerApplication.java` open. The code is a Spring Boot application with the following structure:

```
6
7 @SpringBootApplication
8 @EnableEurekaServer
9 public class EurekaServerApplication {
10
11     Run | Debug
12     public static void main(String[] args) {
13         SpringApplication.run(EurekaServerApplication.class, args);
14     }
15 }
16
```

The terminal output shows the command to run the application and the resulting logs:

```
PS C:\Users\DELL\Documents\jesus\...Comunicacion entre microservicios Sincronica Spring Boot> & 'C:\Users\DELL\gradle\j...eclipse_adoptium-17-amd64-windows\jdk-17.0.15+6\bin\java.exe' '@C:\Users\DELL\AppData\Local\Temp\cp_cnupsr885wkk2pkav2l3gsxdq.argfile' 'com.eureka-server.eureka_server.EurekaServerApplication'
```

The logs indicate that the application is starting successfully and that there are no active profiles.

Ç

The screenshot shows an IDE with the file `ProductoDTO.java` open. The code is a DTO class with the following structure:

```
1 package com.pedidoservice.pedido_service;
2
3 import lombok.Getter;
4 import lombok.Setter;
5
6 @Getter
7 @Setter
8 public class ProductoDTO {
9     private Long id;
10    private String nombre;
11    private Double precio;
12
13    @Override

```

The terminal output shows the command to run the application and the resulting logs:

```
025-06-18T20:31:56.028-05:00 INFO 1892 --- [pedido-service] [main] c.n.discovery.InstanceInfoReplicator : InstanceInfoReplicator onDemand update allowed rate per min is 4
025-06-18T20:31:56.047-05:00 INFO 1892 --- [pedido-service] [main] com.netflix.discovery.DiscoveryClient : Discovery Client initialized at timestamp 1750296716040 with initial instances count: 0
025-06-18T20:31:56.085-05:00 INFO 1892 --- [pedido-service] [main] o.s.c.n.e.s.EurekaServiceRegistry : Registering application PEDIDO-SERVICE with eureka with status UP
025-06-18T20:31:56.099-05:00 INFO 1892 --- [pedido-service] [main] com.netflix.discovery.DiscoveryClient : Saw local status change event StatusChangeEvent [timestamp=1750296716098, current=UP, previous=STARTING]
025-06-18T20:31:56.122-05:00 INFO 1892 --- [pedido-service] [foReplicator-%d] com.netflix.discovery.DiscoveryClient : DiscoveryClient_EDIDO-SERVICE/Andres:pedido-service:8082: registering service...
025-06-18T20:31:56.249-05:00 INFO 1892 --- [pedido-service] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8082 (http) with context path '/'
025-06-18T20:31:56.261-05:00 INFO 1892 --- [pedido-service] [main] .s.c.n.e.s.EurekaAutoServiceRegistration : Updating port to 8082
025-06-18T20:31:56.824-05:00 INFO 1892 --- [pedido-service] [main] c.p.p.PedidoServiceApplication : Started PedidoServiceApplication in 12.885 seconds (process running for 14.19)
025-06-18T20:31:56.843-05:00 INFO 1892 --- [pedido-service] [foReplicator-%d] com.netflix.discovery.DiscoveryClient : DiscoveryClient_EDIDO-SERVICE/Andres:pedido-service:8082 - registration status: 204
```

Se verifica el estado de microservicios usando la consola Eureka, asegurando que estén activos. Se hacen solicitudes HTTP a través de PostMan para listar productos y crear pedidos. Se muestran resultados de estas pruebas para asegurar que el sistema funcione correctamente.

Somprender y poner en práctica los conceptos fundamentales del desarrollo de microservicios, incluyendo la configuración, despliegue y monitoreo de servicios en un entorno distribuido. Se evidenció la importancia de herramientas como Eureka para el registro y gestión de servicios, así como de Postman para realizar pruebas y verificar el funcionamiento de las solicitudes

HTTP. Además, se fortalecieron habilidades en la verificación del estado de los microservicios y en la interacción entre ellos, lo que proporciona una base sólida para el desarrollo de sistemas escalables y modularizado