COMUNICACION ENTRE MICROSERVICIOS SINCRONICA SPRING BOOT

Jesús Estenllos Loaiza Serrano 2313021

Desarrollo de Software III

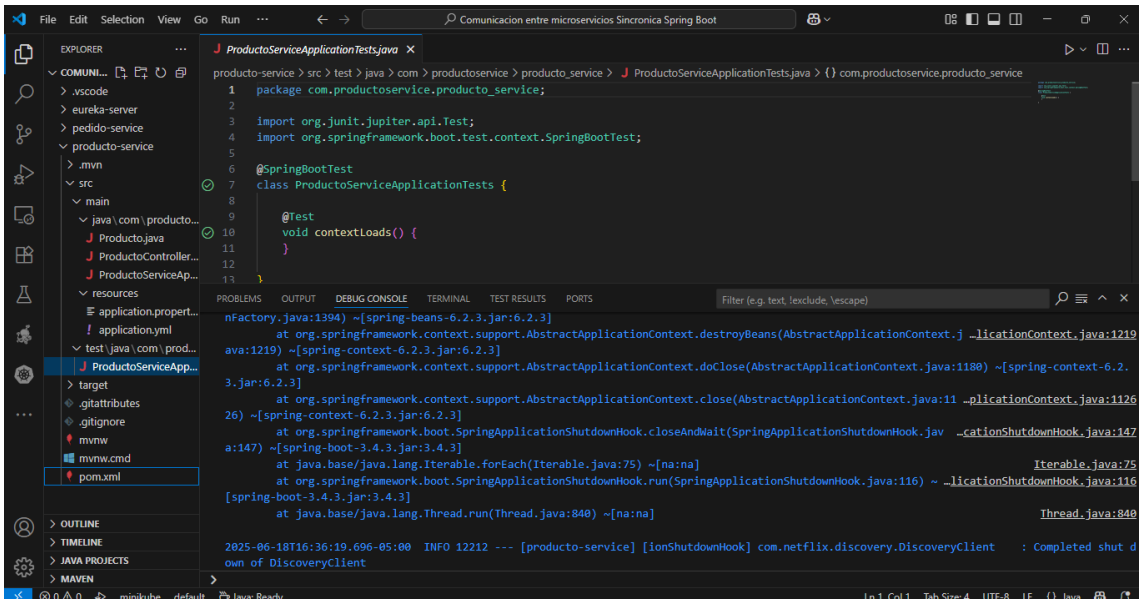
Docente: Ing. Álvaro Salazar

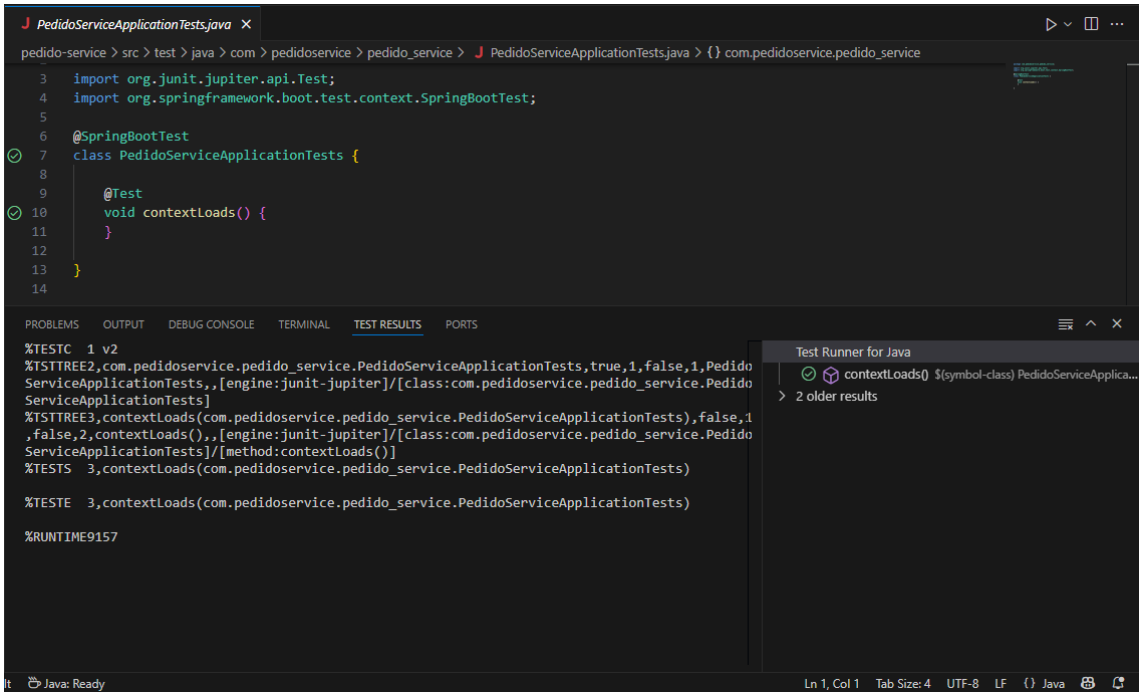
Grupo 50

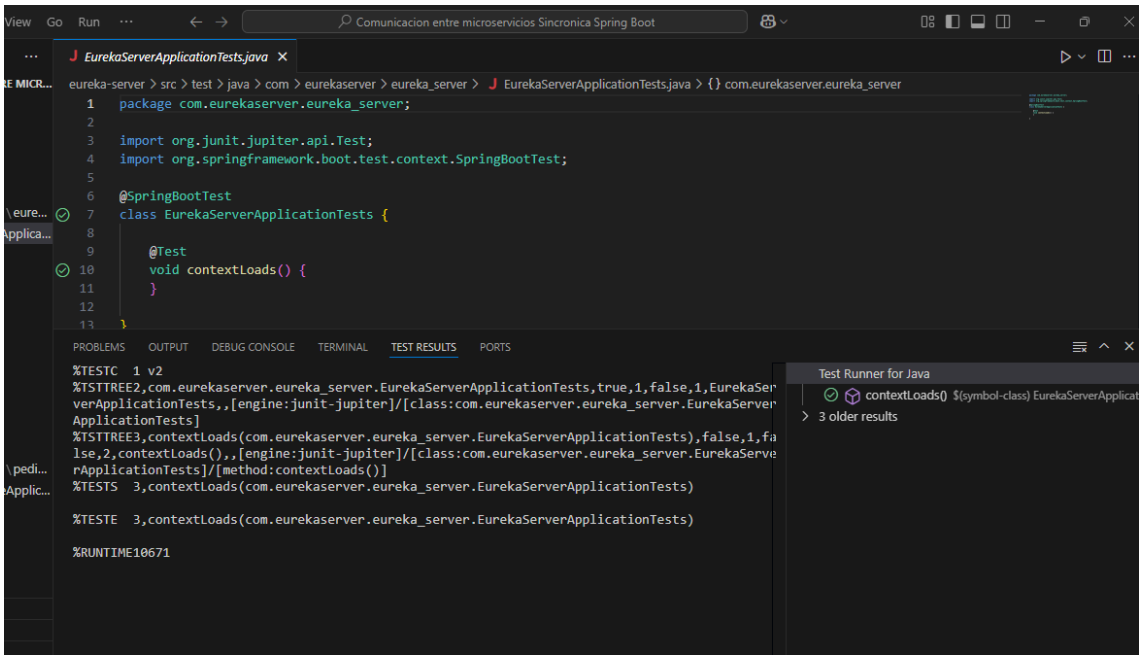
Sede Tuluá

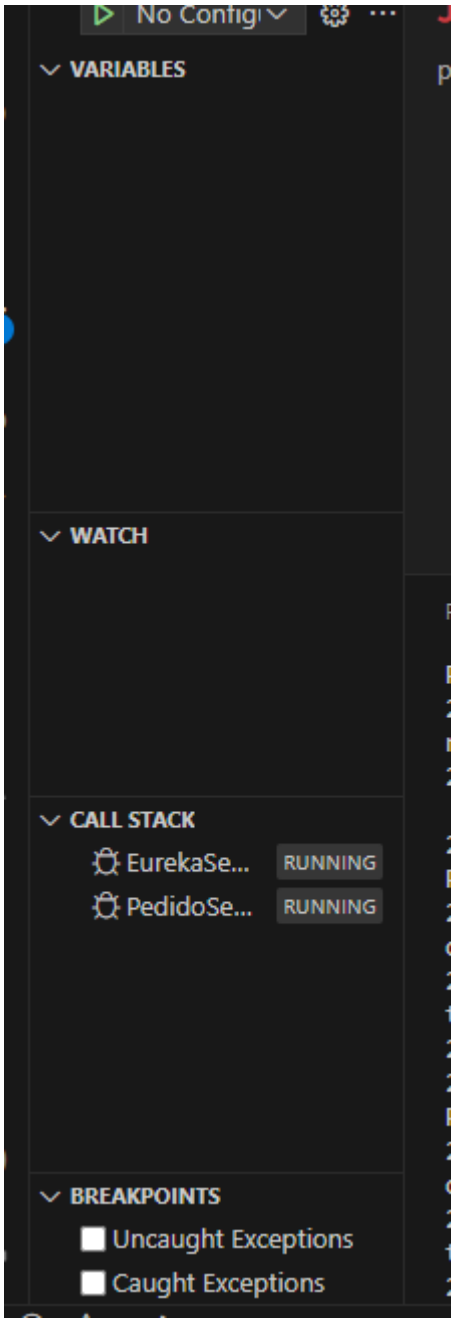
17/06/2025

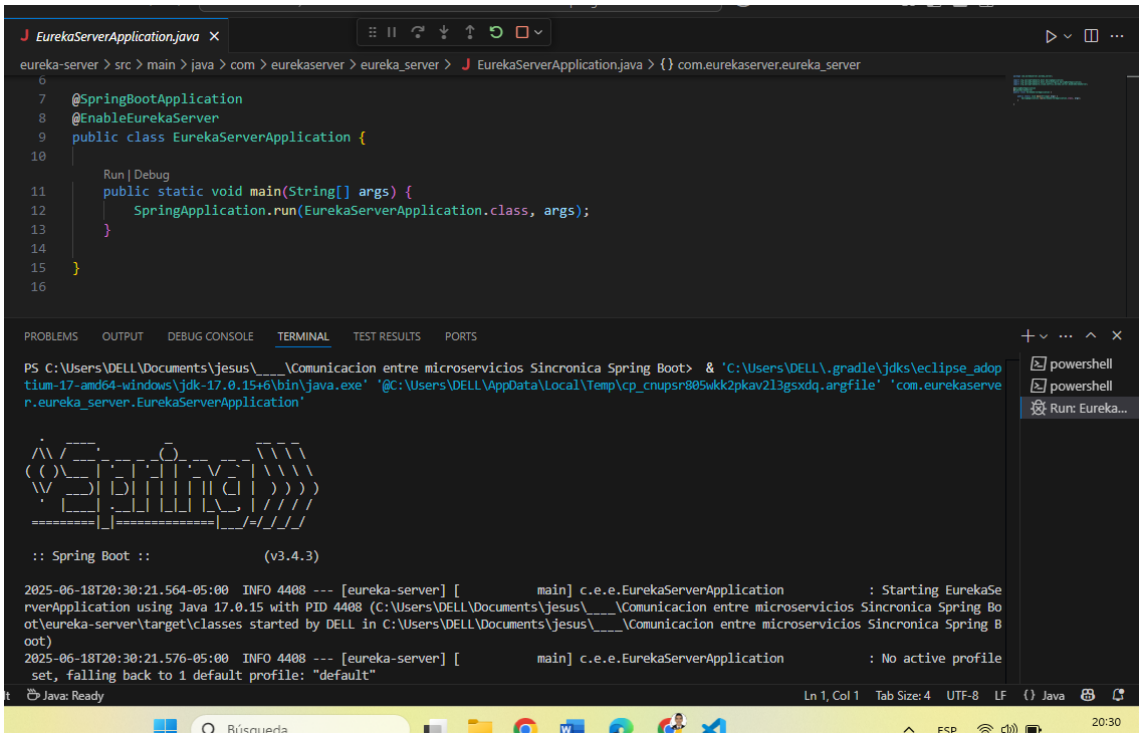
Test servicios_producto service test



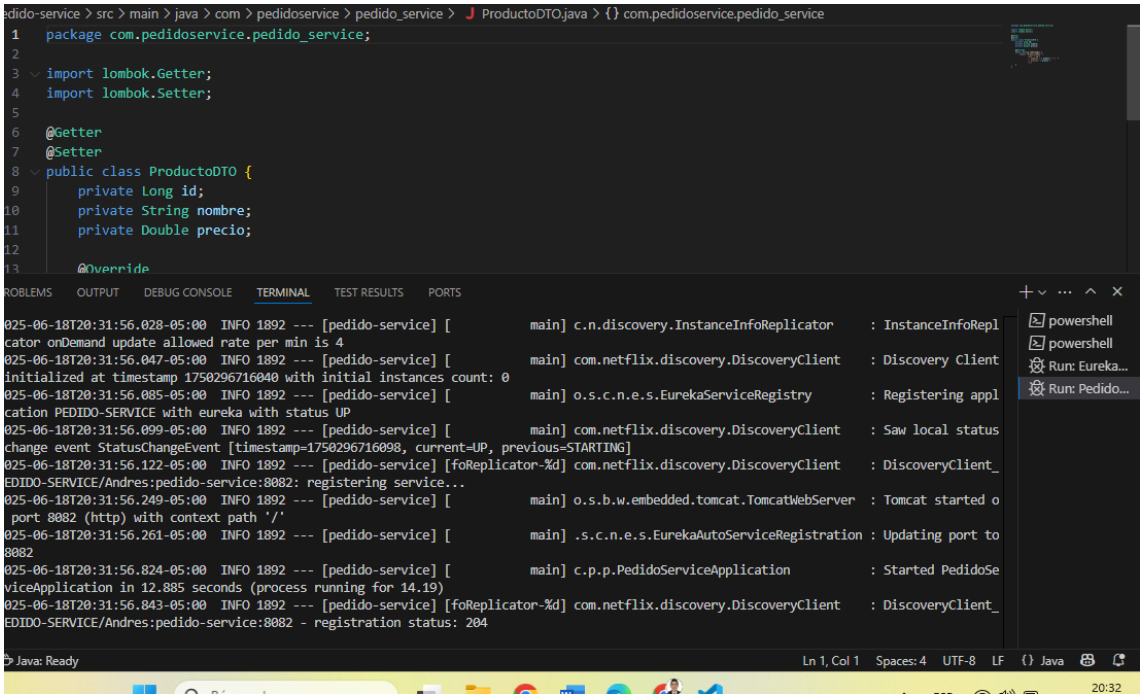



View Go Run
Comunicacion entre microservicios Sincronica Spring Boot
EurekaServerApplicationTests.java
eureka-server > src > test > java > com > eurekaserver > eureka_server > EurekaServerApplicationTests.java > { } com.eurekaserver.eureka_server
1 package com.eurekaserver.eureka_server;
3 import org.junit.jupiter.api.Test;
4 import org.springframework.boot.test.context.SpringBootTest;
6 @SpringBootTest
7 class EurekaServerApplicationTests {
9 @Test
10 void contextLoads() {
11 }
13 }
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL TEST RESULTS PORTS
%TESTC 1 v2
%TSTTREE2,com.eurekaserver.eureka_server.EurekaServerApplicationTests,true,1,false,1,EurekaSer
verApplicationTests,,[engine:junit-jupiter]/[class:com.eurekaserver.eureka_server.EurekaServer
ApplicationTests]
%TSTTREE3,contextLoads(com.eurekaserver.eureka_server.EurekaServerApplicationTests),false,1,fa
lse,2,contextLoads(),,[engine:junit-jupiter]/[class:com.eurekaserver.eureka_server.EurekaServe
rApplicationTests]/[method:contextLoads()]
%TESTS 3,contextLoads(com.eurekaserver.eureka_server.EurekaServerApplicationTests)
%TESTE 3,contextLoads(com.eurekaserver.eureka_server.EurekaServerApplicationTests)
%RUNTIME10671
Test Runner for Java
contextLoads() ${symbol-class} EurekaServerApplicat
3 older results


No Config
VARIABLES
WATCH
CALL STACK
EurekaSe...
RUNNING
PedidoSe...
RUNNING
BREAKPOINTS
Uncaught Exceptions
Caught Exceptions



Ç



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