UNIVERSIDAD DE LAS FUERZAS ARMADAS "ESPE"

APLICACIONES DISTRIBUIDAS

RMI (Remote Method Invocation) en Java

Nombre: Soria Giovanna, Caraguay Richard

NRC: 3877

Fecha: 6/01/2024

1.INTRODUCCIÓN

El propósito de este proyecto es desarrollar una aplicación distribuida utilizando Java RMI (Remote Method Invocation). RMI permite la invocación de métodos de objetos remotos, facilitando la comunicación entre un servidor y varios clientes de forma transparente y eficiente.

La aplicación consiste en:

1. Un servidor que implementa un servicio remoto básico.
2. Múltiples clientes que interactúan con el servicio remoto.

2.DESARROLLO

Arquitectura del Proyecto

1. Componentes Principales

El proyecto está organizado en las siguientes clases principales:

- IHelloService: Interfaz remota que define los métodos disponibles para los clientes.
- HelloServiceImpl: Implementación del servicio remoto.
- ServerRMI: Clase principal del servidor que registra el servicio en el registro RMI.
- ClientRMI: Clase cliente que busca el servicio remoto y lo utiliza.

2. Estructura del Proyecto

La estructura del proyecto sigue una organización modular:

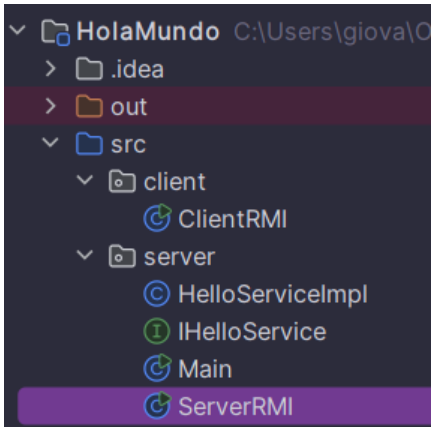

Figura1.Estructura del proyecto

Servidor

- Se creó la interfaz IHelloService que extiende java.rmi.Remote e incluye métodos declarados con la excepción RemoteException.
- La clase HelloServiceImpl implementa la interfaz remota y extiende UnicastRemoteObject.
- La clase ServerRMI configura y registra el servicio remoto en el registro RMI.

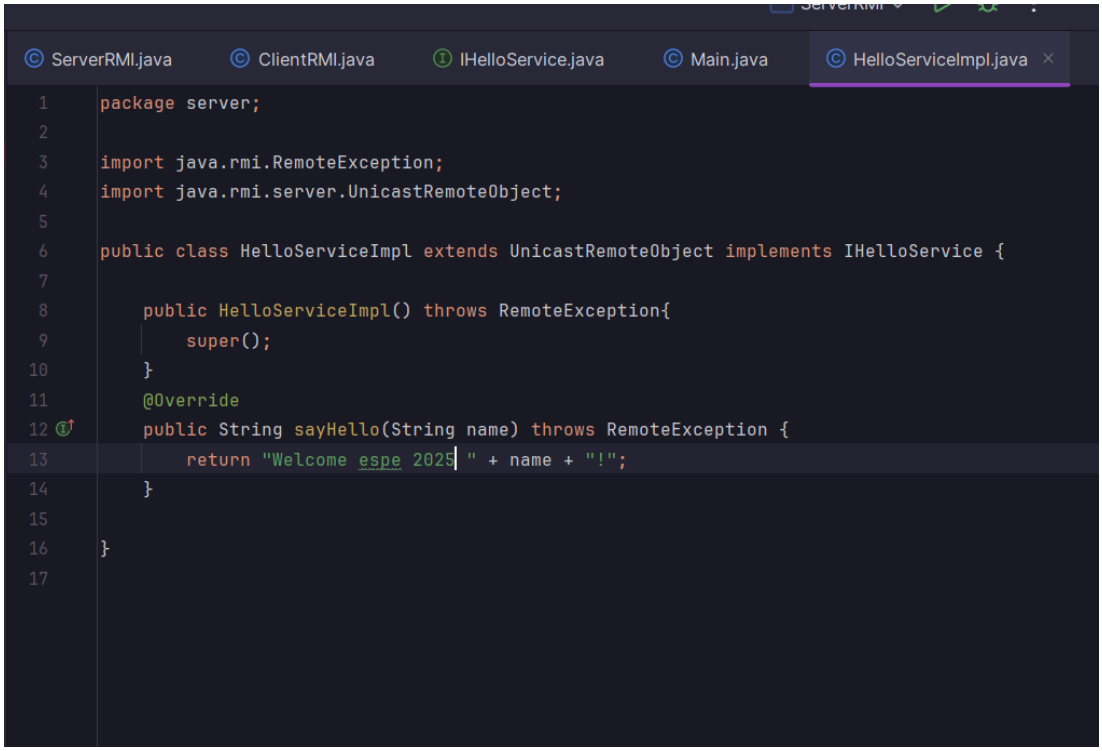

Figura 2. Funcionamiento del server en ejecución

Cliente

- La clase ClientRMI se conecta al servidor y utiliza el servicio remoto.
- Al ejecutar el cliente, se realiza la conexión al servidor y se obtiene la respuesta del método remoto, mostrando resultados como:

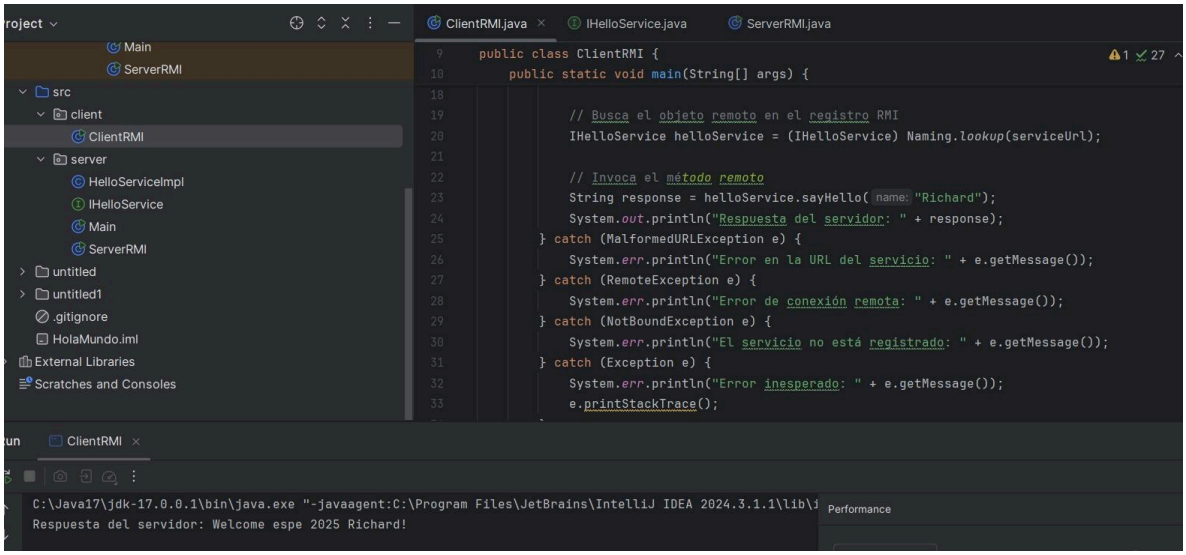

Figura 3.Fucionamiento del cliente

3.CONCLUSIÓN

La aplicación desarrollada demuestra el uso eficiente de Java RMI para la implementación de un sistema distribuido. Este enfoque simplifica la comunicación entre procesos, abstractando la complejidad de las conexiones remotas. En aplicaciones más avanzadas, se pueden incluir características adicionales como autenticación, seguridad y balanceo de carga.

4.ENLACE AL REPOSITORIO DE GITHUB

https://github.com/ricchhard/RMI_DEMO_Caraguay_Soria