

Actividad obligatoria 2. Servidor RMI de constelaciones. UF3-1.

Miguel Lozano Cerrada 2DAM.

En esta actividad he implementado una estructura cliente-servidor utilizando la tecnología RMI (Remote Method Invocation).

Esta estructura tiene 3 agentes:

- **Interfaz:** donde se van a definir los métodos que luego implementará y compartirá la clase en el servidor.
- **El servidor:** donde va a estar la información, implementa la interfaz a través de una clase que será mostrada al cliente para hacer consultas.
- **El cliente:** que podrá usar la clase en el servidor para sacar la información deseada de forma remota.

En primer lugar, se han creado las clases modelo Planeta y Constelacion. Estas clases almacenan información sobre planetas y constelaciones, respectivamente. Ambas clases tienen como atributos: nombre y descripción. Es importante mencionar que estos objetos deben ser serializables para poder ser transmitidos a través de la red.

También se crearon las interfaces ConstelacionRMI y PlanetaRMI que, como ya hemos dicho, establecen los métodos que implementarán las clases ConstelacionRemoto y PlanetaRemoto. Estas interfaces actúan como contratos que definen qué servicios pueden ofrecer el servidor a los clientes.

Las clases ConstelacionRemoto y PlanetaRemoto implementan las interfaces ConstelacionRMI y PlanetaRMI, respectivamente. Estas clases simulan una base de datos generando listas con objetos Constelacion y Planeta respectivamente, sobre los que el cliente podrá realizar consultas.

Del lado del servidor, se crean instancias de las clases ConstelacionRemoto y PlanetaRemoto. Luego, se crea un objeto de la clase Registry con el puerto de escucha 4000 en el que registramos nuestros servicios "ConstelacionService" y "PlanetaService". Este registro permite que los clientes encuentren los servicios disponibles.

Finalmente, del lado del cliente, se ha creado la lógica para mostrar menús a través de la clase ClienteCLI. La clase ClienteRMI se encarga de la comunicación con el servidor. Se crea un Registry con el puerto establecido por el servidor y luego, al entrar a los diferentes submenús (Constelaciones y Planetas), se establece el registry.lookup específico para cada clase que se vaya a consultar.

Al tener la posibilidad de que el cliente pueda consultar diferentes objetos, se delega al ClienteRMI la responsabilidad de obtener el registro en el puerto establecido por el

servidor y luego, al entrar a los diferentes submenús, se realiza el `registry.lookup` específico para cada clase que se vaya a consultar.

Es importante mencionar que se han implementado mecanismos para manejar posibles excepciones que puedan ocurrir durante la comunicación RMI, como `RemoteException` y `NotBoundException`.