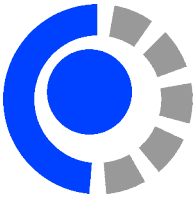
Universidad Nacional del Comahue

Facultad de Informática

Departamento de Ingeniería de Computadoras



Laboratorio de Programación Distribuida

Trabajo Obligatorio N° 2

Implementación de arquitectura de servicios utilizando RMI en Java

Amarante Carlos Adolfo, FAI-1922

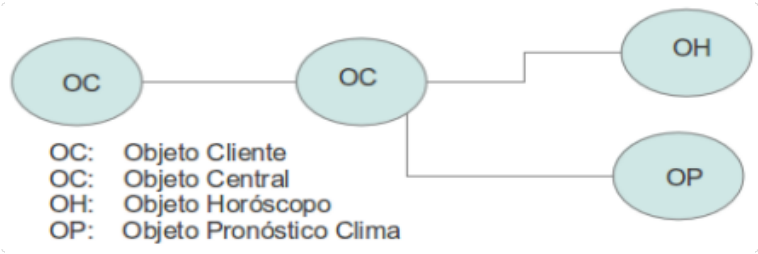
Mayo 2022

1 Introducción

1.1 Problema

Se pide desarrollar un programa en lenguaje java utilizando invocación de métodos remota - RMI.

El esquema general es el siguiente:



OC realiza una consulta a OC pidiendo una predicción del Horóscopo para un signo determinado y el pronóstico del clima dada una fecha. OC utilizando el signo consultará a OH y con la fecha a OP. Las predicciones de OH y OP son ramdom.
El trabajo puede realizarse en distintos niveles:

- Básico: Los servidores son simples y solo atienden una solicitud a la vez.
- Medio: Los servidores pueden responder múltiples consultas a la vez.
- Avanzado: Nivel Medio y las consultas repetidas obtienen el mismo resultado

El problema dado sugiere una arquitectura de servicios, donde un servidor central recibirá solicitudes de clientes, las redirigirá a otros servidores, de los cuales obtendrá su respuesta y las reenviara al cliente.

Se propone la creación de 3 interfaces:

- IBroadcastAPI para los servicios ofrecidos por el servidor de pronóstico del clima.
- IHoroscopeAPI para los servicios ofrecidos por el servidor de horóscopo.
- IMainServerAPI para los servicios ofrecido por el servidor central. Esta sera hija de IBroadcastAPI y IHoroscopeAPI

2 Desarrollo

2.1 Implementación

La implementación sera definida de la siguiente manera:

- BroadcastAPIImplementation sera la implementación para la interfaz IBroadcastAPI en el servidor correspondiente al servicio de pronostico del clima.
- HoroscopeAPIImplementation sera la implementación para la interfaz IHoroscopeAPI en el servidor correspondiente al servicio de horóscopo.
- MainServerAPIImplementation sera la implementación para la interfaz IHoroscopeAPI en el servidor central. Permitirá la comunicación con OH y OP.

El cliente sólo se comunicara con el servido central.

2.2 Uso del programa

La ejecución de los archivos debe darse en el siguiente orden:

1. BroadcastServer y HoroscopeServer
2. MainServer
3. Client

3 Conclusión

La arquitectura presentada permite responde a múltiples solicitudes de múltiples clientes. Por otro el uso de RMI permite simplificar el desarrollo de la comunicación entre cliente-servidor en comparación al uso de sockets.

Además, la elaboración de un servidor central que delegue la ejecución de distintas tareas a distintos servicios es sumamente sencilla, creando un nuevo tipo de objeto RMI que herede todas las interfaces de los objetos que hacen referencia a los objetos que brindan servicios.

Como contrapartida, su uso con otras tecnología que no estén desarrolladas en java es dificultoso dada la ausencia de recursos desarrollados.

En cuanto a que las consultas repetidas obtengan el mismo resultado en base a la misma entrada, surgió el problema de que se obtenía una fecha distinta al solicitar la fecha actual porque cambiaba la hora. Esto fue solucionado normalizando la fecha en los servidores de servicios.

De esta forma se cumplió con todos los requerimientos planteados.