

Cliente Servidor

Valentina Siabatto

Febrero 2020

1 Resumen

Este artículo muestra la implementación y arquitectura de un servidor web desarrollado en java. El cual soporta múltiples solicitudes seguidas y retorna los archivos solicitados, sin usar ningún framework.

2 Introducción

El objetivo de este laboratorio es crear un servidor web sin usar ningún framework tales como spark o spring. El servidor web debe soportar múltiples solicitudes seguidas del cliente y atenderlas adecuadamente, retornando los archivos que se estén solicitando. Está implementado en java, utiliza el paquete java.net el cual permite realizar conexiones y transacciones a través de la red, junto con el paquete java.io, el encargado de gestionar las operaciones de entrada/salida.

3 Contenido

Usando los elementos de Java se realizó un servidor que respondiera las peticiones de un cliente. Este fue implementado a través de sockets. Los sockets permiten establecer un enlace entre dos máquinas independientes. La librería Java.net provee dos clases para implementar la conexión desde el lado del Cliente (Socket) y desde el lado del servidor (ServerSocket). El servidor que se ejecuta está vinculado a un número de puerto específico, el servidor espera escuchando el socket para que el cliente haga una solicitud de conexión.[1]

La clase Cliente, establecerá la conexión con el servidor usando un socket y el puerto específico, una vez establecida la conexión entre el cliente y el servidor, el cliente es capaz de hacer peticiones al servidor.

Al llegar estas peticiones al servidor, el encabezado http se desglosa por partes para poder reconocer el tipo de solicitud que necesita. Primero, se verifica que sean métodos "GET" ya que este servidor solo implementa este método, de no ser así se retorna una página explicando que el método no está soportado. Del mismo modo, se reconoce el recurso que hace referencia la petición, si es una imagen (png o jpg) o un html, y si existe. Si el recurso existe se retorna la

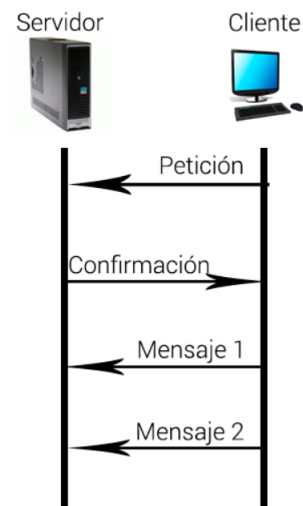


Figure 1: Cliente Servidor

respuesta al cliente con el recurso que solicitó. De esta forma el servidor recibe las peticiones y las responde.

4 Conclusiones

El servidor fue capaz de resolver las peticiones enviadas por el cliente. La implementación del servidor fue capaz de resolver las solicitudes gracias a las librerías de java y a la información del encabezado http. Sin embargo, la facilidad para resolver estas peticiones es mucho menor comparada con los frameworks que puede implementar java, tales como spark o spring, que permiten desarrollar de manera más sencilla aplicaciones web.

References

- [1] Escuela Colombiana de Ingeniería. Introduccion a esquemas de nombres, redes, clientes y servicios con java.