

Arquitectura Empresarial Laboratorio-03

Johann Steven Bogotá Vélez

Estudiante, Escuela Colombiana de Ingeniería Julio Garavito

Ingeniería de Sistemas

Email: johann.bogota@mail.escuelaing.edu.co

Bogotá, Colombia

February 12, 2021

Abstract

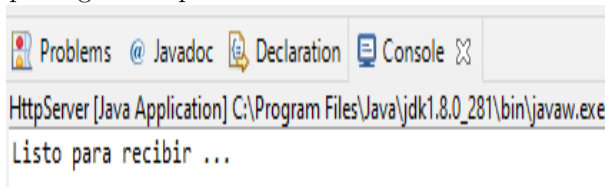
Se va a diseñar y construir un servidor web capaz de recibir múltiples solicitudes enviadas por los usuarios, de manera no concurrente; y de estar forma funcione como un servidor estático que retorna todos los archivos que les solicita el cliente.

1 Introducción

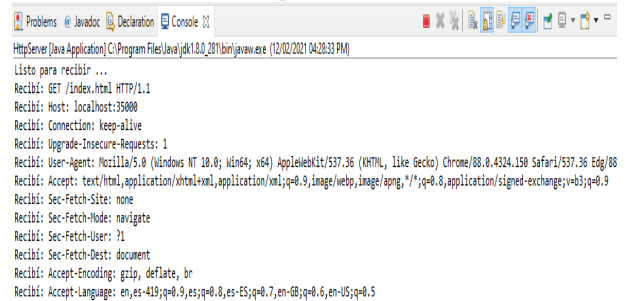
En este artículo se va a mostrar el proceso para diseñar un servidor web básico en el lenguaje de programación java. No vamos a utilizar frameworks web como Spark(Laboratorio 2) o Spring, ya que se quiere mostrar librerías que hacen uso del manejo de la red por donde enviamos nuestros archivos.

2 Diseño

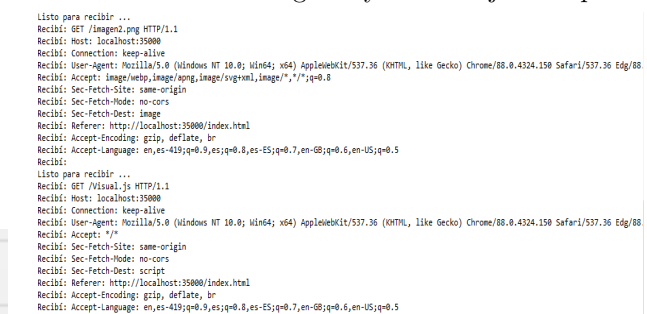
Para hacer la conexión entre cliente y servidor se usaron sockets, que crea una conexión y así por medio estos podemos enviar y recibir paquetes. Cuando generamos la conexión el servidor se queda esperando a que llegue una petición.



Cuando por medio de nuestro navegador hacemos una petición a un recurso, este le llega al servidor por medio de la petición GET, en este caso pide el recurso index.html.



Del mismo modo después de pedir este recurso, pide los demás recurso que se necesita para cargar la página, en este ejemplo podemos ver como pide otros recursos como imágenes y archivos javaScript.



Ya teniendo los encabezados de estas peticiones, el programa se encargara de validar y hacer que estos recursos sean enviados por la misma conexión a su

cliente, y así de esta forma tener un servidor web estático donde los recursos se encuentren en disco.

Mi sitio web

Mi sitio web basado en Spark

Creado por Johann Bogotá

Mostrar contenido 1 | Mostrar contenido 2

Una vez terminado nuestra implementación, se desplego la aplicación en heroku, y así tener un servicio en la red mostrando la funcionalidad del desarrollo de este programa.

3 Conclusión

Como conclusión, se puede decir que gracias a algunas librerías que nos ofrece java, y al manejo de conexiones y peticiones http, podemos generar un programa que funciones como un servidor web, sin necesidad de usar otros frameworks y ser capaces de crear nuestras propias implementaciones.

References

- [1] DataOutputStream.
<https://docs.oracle.com/javase/7/docs/api/java/io/DataOutputStream.html>
- [2] ByteArrayOutputStream.
<https://docs.oracle.com/javase/7/docs/api/java/io/ByteArrayOutputStream.html>