LABORATORIO 4 Arquitecturas de Servidores de Aplicaciones, Meta protocolos de objetos, Patrón IoC, Reflexión

Jonatan Esteban Gonzalez Rodriguez*
Escuela Colombiana de Ingenieria Julio Garavito
(Dated: Febrero 23 del 2020)

Se desarrollará un servidor web tipo apache capaz de visualizar páginas web, imagenes en los diferentes formatos y realizar conexión a bases de datos.

Keywords: Apache, servidor web, base de datos, paginas html, Heroku

I. INTRODUCCIÓN

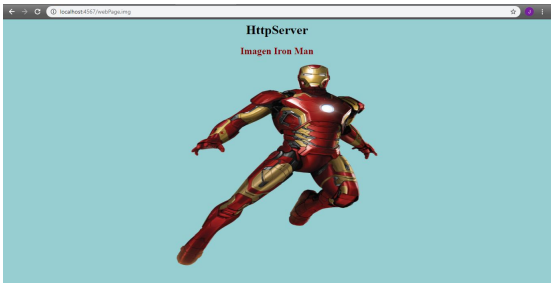
El servidor web desarrollado de *tipo Apache* en java facilita la visita de un sitio web de manera concurrente, ya que con solo ingresar la URL, el servidor web envía los archivos solicitados actuando como un repartidor virtual.
Se puede apreciar el uso de conexión a bases de datos, uso de un servidor, la interacción con servicios y usuarios de Java, utilizando los principales paquetes de java.net y sus clases como URL, URLConnection, Socket, ServerSocket, DatagramPacket, DatagramSocket y MulticastSocket.
Se realizara el despliegue del servidor en Heroku donde se podrá evidenciar que el servido web es capaz de entregar una pagina web, una imagen y realizar la conexión con la base de datos creada y la obtención de los datos que en ella se almacenan.

II. ESTRUCTURA

Para que el servidor entregue la página con conexión a base de datos:
Se cuenta con un método donde se realiza la conexión a la base de datos antes creada mediante Heroku, en la base de datos se creó una tabla *"Usuario"* con atributos id, nombre, apellido y correo.
Además se cuenta con un método que consulta los datos almacenados en la base de datos y los retorna en un *ArrayList<Usuario>*.
Por último se cuenta con un método donde se construye el documento *HTML* y contiene la información traída desde la base de datos y retornada por el método *getUsuarios*
Para que el servidor entregue la página HTML o una imagen:
En la clase *WebServices* se crean los métodos donde se generan el HTML y son requeridos por la clase *Server* donde cada método se encarga de entregar el servicio requerido por el usuario.
Se agrega la clase BetterSpringBoot donde se genera el controlador y ayuda a generar la concurrencia.

III. VALIDACIÓN

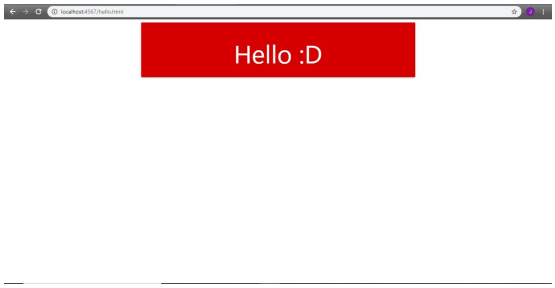
En el servidor podemos encontrar los diferentes servicios que presta como
Imagen



Conexion Base de datos

Pagina HTML

* Correo: jonatan.gonzalez@mail.escuelaing.edu.co



Para comprobar que el servidor web es capaz de entregar cada servicio dirigase a
`https://evening-scrubland-29292.herokuapp.com/webPage.img`
`http://evening-scrubland-29292.herokuapp.com/users.bd`
`http://evening-scrubland-29292.herokuapp.com/hello.html`
Donde encontrará el servidor web desplegado en Heroku.

IV. CONCLUSIONES

- Desplegar las aplicaciones en heroku solo es recomendable para aplicaciones pequeñas ya que cuenta con restricción de tamaño y flujo de datos.
- Conocer la utilidad de algunas clases de los paquetes de Java.net Y aplicarlos de forma satisfactoria.
- La construcción del servidor debe garantizar la lectura de los diferentes formatos de los recursos ubicados dentro del servidor.
- El uso de POJOS y anotaciones permiten estructurar mejor el uso y administracion de un servidor web a nivel de código.
- La respuesta a un cliente se puede ver afectada por la latencia entre el servidor, la ubicación del cliente y las características hardware del cliente.

V. BIBLIOGRAFÍA

1. Marc Lankhorst et al. Enterprise Architecture at work. Springer, 2009
2. https://docs.oracle.com/javase/tutorial/networking/index.html