

Web Server

Johan Nicolas Cortes Torres
Escuela Colombiana de Ingenieria Julio Garivito
AREP

August 2020

Abstract

En el presente documento se encontrará información acerca de la elaboración de un laboratorio que se basaba en implementar un servidor web sin usar ningún tipo de framework como spark o Spring, solo se permite el uso de java y las librerías de manejo de la red que este mismo ofrece. Este servidor web debe retornar todos los archivos solicitados. Adicionalmente se debe crear la conexión a una base de datos que permita verificar el uso del servidor mediante un framework que debe permitir publicar servicios web "get".

1 Introduction

El laboratorio se divide en dos partes fundamentales. La primera parte trata de la implementación de un servidor web que soporte múltiples solicitudes seguidas pero que no sean concurrentes. El servidor debe retornar todos los archivos solicitados, desde páginas html hasta imágenes, y desplegarlo en Heroku. La segunda parte trata de crear un framework que permita simular a spark que permita publicar servicios web "get" y acceder a los recursos estáticos, adicionalmente a esto se debe conectar a una base de datos desde el servidor y desplegarla en heroku.

Un servidor web es un programa que utiliza HTTP (Hypertext Transfer Protocol) para mostrar los archivos como páginas web, en respuesta a las diferentes solicitudes que hace un usuario.

Spark es un simple y expresivo marco web de Java/Kotlin DSL construido para un desarrollo rápido. La intención de Sparks es proporcionar una alternativa para los desarrolladores de Java/Kotlin que quieran desarrollar sus aplicaciones web de la forma más expresiva posible y con un mínimo de complejidad.

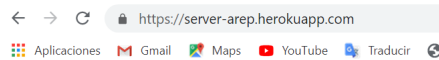
Heroku es una empresa que se especializa en ofrecer servicios de plataforma administrada, por sus siglas en inglés PaaS, en otras palabras ofrece servicios de servidores y redes administrados por Heroku en donde se pueden alojar aplicaciones de diferentes lenguajes de programación como Python, Java, PHP y más.

2 Arquitectura

La clase principal se llama App, esta clase crea un objeto tipo MongoDB y un objeto HttpServer al cual le asigna el objeto MongoDB para así crear el servidor y asignarle la conexión a la base de datos. Finalmente esta clase inicia el servidor.

2.1 Servidor Web

Para la implementación del servidor Web se creó una clase llamada HttpServer la cual simula un servidor web, esta clase permite obtener los archivos que se solicite el usuario, desde una página html hasta una imagen. La imagen siguiente muestra el servidor en su página inicial.



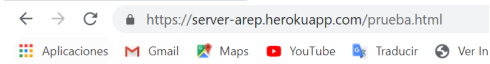
Mi propio mensaje

Esta clase cuenta con los siguientes métodos:

- start(): Método que inicia el servidor.
- manageMongo(): Método que consulta de la base de datos Mongo.
- manageImg(): Método que maneja la solicitud de una imagen.
- manageJSript(): Método que maneja la solicitud de un archivo JavaScript.
- manageHtml(): Método que maneja la solicitud de un archivo html.
- getPort(): Método que retorna el puerto sobre el que se va a ejecutar el servidor.

Los solicitudes que se permiten hacer en el servidor son:

- ”/prueba.html”



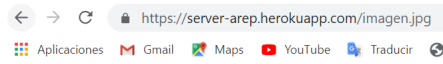
Este es un documento html

Autor: Johan Nicolas Cortes Torres

- ”/prueba2.js”



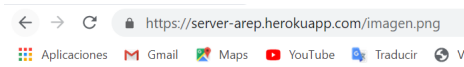
- ”/imagen.jpg”



Este es una imagen



- ”/imagen.png”



Este es una imagen



2.2 Conexion base de datos

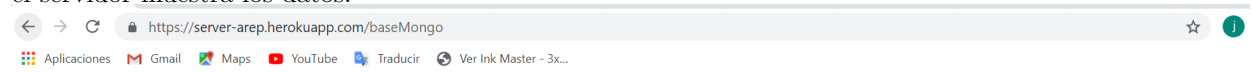
Para la implementación de a conexion se creo la clase MongoDB y una base de datos en mongodb.com. En esta clase se creo una conxion a la base de datos que se va a alojar en link anterior.

Los metodos que se implementaron para lograr una conexion exitosa, fueron:

- MongoDB(): Costructor de la clase, que permita crear la conexion a la base de datos.
- add(): Metodo que agrega documentos a la base de datos mongo.
- select(): Metodo que retorna los documentos que existen en a base de datos

Para la muestra de los datos que tiene la base de datos se le tiene que hacer una solicitud especifica al servidor.

La solicitud debe ser: `"/baseMongo"`. En la siguiente imagen se muestra como el servidor muestra los datos.



Este son datos de la base de datos en MongoDB

```
{"_id": {"$oid": "5f51a2d84deb72834ae700c4"}, "name": "prueba"} {"_id": {"$oid": "5f51aa7851b54e56c2bbbfa4"},  
"nombre": "sirve"} {"_id": {"$oid": "5f51ab3d63158163912f78ce"}, "nombre": "sirve1", "edad": 21, "Carrera":  
"ISIS"}
```

3 Conclusion

Concluyendo se realizó correctamente la implementación completa del laboratorio propuesto por la clase de AREP, se logró realizar un servidor web que retornara todos los archivos solicitados por el usuario, adicionalmente a esto se logro crear la conexion con una base de datos alojada en mongo.com, que retorna los datos alojados en esta. Todo esto se logro sin utilizar ningun framework, solo realizando implementaciones que simularan su actuar.

References

- [1] Laboratorio
<http://campusvirtual.escuelaing.edu.co/moodle/mod/assign/view.php?id=34731>
2020.
- [2] Repositorio
<https://github.com/jnicolasct/AREP-LAB3>
2020.
- [3] Heroku
<https://server-arep.herokuapp.com/>
2020.
- [4] Servidor Web
<https://www.osgroup.co/que-es-un-servidor-web/>
2020.