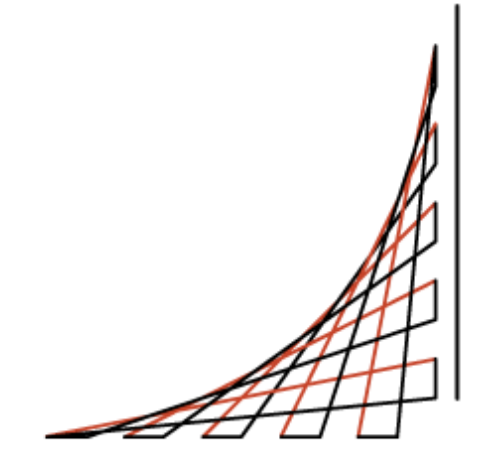
Taller 3 Arem

Fernando Barrera Barrera

Luis Daniel Benavides Navarro

Arquitecturas Empresariales



ESCUELA COLOMBIANA DE INGENIERÍA
JULIO GARAVITO

Índice

1. Introduccion **2**

2. Conceptos Basicos **2**

3. Diseño **2**

4. Pre-Requisitos **3**

5. Intalacion **4**

6. Ejecucion **4**

7. Pruebas de EndPoints **5**
7.1. Recursos Estaticos 5
7.2. Recursos Dinamicos 6

8. Registro en Base de Datos **7**

9. Conclusion **8**

10. Bibliografía **9**

1. Introduccion

Este taller fue hecho con el objetivo de comprender la arquitectura de un framework de servidor web como spark por lo cual en el laboratorio se realizo una implementacion de un servidor web usando sockets y configurandolo para que reciba peticiones y devuelva recursos estaticos como archivos html,archivos txt,archivos js e imegenes en formato PNG Y JPG ,Porl ultimo se implementaron funciones lambda para la asignacion de endpoints para los recursos dianmicos,tal como en el framework spark y se logro realizar la conectada al endpoint a una base de datos Mongo DB.

2. Conceptos Basicos

- **Maven:** Herramienta de sofware dedicada a la estructutracion y construccion de proyectos java. [5]
- **Git:** sofware de control de versiones de proyectos [1]
- **Java :** es un lenguaje de programacion orientado a objetos que se desarrollo en los años 90 [4]
- **Heroku :** plataforma de despliegue de sofware que funciona de forma similar a un servicio de almacenamiento en la nube [2]
- **Funcion Lambda :** Es una funcion que funciona como una subrutina para construir una respuesta de una fucnion de orden superio [3]
- **Mongo DB :** Una base de datos no relacional que usa documentos JSON para el almacenmiento datos [6]

3. Diseño

Acontinuacion prodra observar el diagrama de clases de la aplicacion .

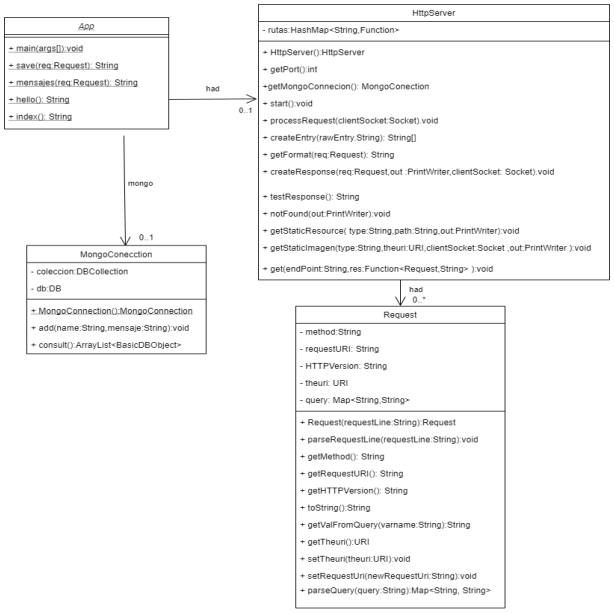


Figura 1:

Las clase principal es la clase App que se encarga de iniciar el servidor http y a su vez la conneccion a la base de datos,la clase mongoConnecction es la clase encargada de ser el puente entre la app y la base

datos,ya que basicamente se conecta a la base datos mediante un clinte con la URI de la base de datos y es el que se encarga de realizar las consultas e inserciones a la base de datos

```
 * @throws UnknownHostException
 */
public MongoConnection() throws UnknownHostException {
    MongoClientURI uri = new MongoClientURI("mongodb+srv://admin:protocolo15@basemongocluster.otant.mongodb.net/AREM?retryWr:
    MongoClient mongoClient = new MongoClient(uri);
    db= mongoClient.getDB("AREM");
    coleccion= db.getCollection("LAB");

}

/**
 * Este metodo recibe el nombre y el mensaje a insertar en la base de datos
 *
 * @param name
 * @param mensaje
 */
public void add(String name,String mensaje) {
    BasicDBObject objeto= new BasicDBObject();
    objeto.put("nombre",name);
    objeto.put("mensaje",mensaje);
    coleccion.insert(objeto);

}
/**
 * Este metodo retorna un ArrayList con los mensajes registrados en la base de datos
 *
 * @return ArrayList<BasicDBObject>
 */
public ArrayList<BasicDBObject> consult() {
    ArrayList<BasicDBObject> registros = new ArrayList<BasicDBObject>();
    DBCursor mensajes = coleccion.find();
    while (mensajes.hasNext()){
```

Figura 2:

La asignacion de EndPoint se hace desde la clase principal donde a cada enpoint se le asigna una respectiva funcion por medio de una funcion lambda como se vera acontinuacion

```
 */
public static void main(String[] args) throws IOException {
    HttpServer server=new HttpServer();
    mongo=server.getMongoConnecion();
    server.get("/hello", (req) -> Hello());
    server.get("/home", (req) -> index());
    server.get("/registro", (req) -> mensajes(req));
    server.get("/save", (req) -> save(req));
    server.start();
}
/**
 * Este metodo es el encargado de realizar inserciones a la base datos con los datos recicbidos en la peticion y redir
 *
 * @param req
 * @return String
 */
private static String save(Request req) {
     System.out.println("post          "+req.getValFromQuery("name"));
     mongo.add(req.getValFromQuery("name"),req.getValFromQuery("message"));
     return  "HTTP/1.1 200 OK\r\n"
         + "Content-Type: text/"+"html"+"\r\n"
         + "\r\n"
         +"<!DOCTYPE html>"
       + "<html>"
       +"<head>"
       +"<meta http-equiv='Content-Type' content='text/html; charset=utf-8'>"
       +"<meta http-equiv='refresh' content='2;URL=https://arem-taller3.herokuapp.com/Apps/registro'>"
       +"</head>"
       + "<body>"
       + "<h1>Message Saved Successfully</h1>"
       + "</body>"
       + "</html>";

}
```

Figura 3:

4. Pre-Requisitos

- Git

- Java
- Maven

5. Intalacion

Para iniciar la instalacion del programa primero clone el repositorio donde se encuentra alojado el programa desde la consola de comandos

Figura 4:

Despues se procede a entrar al directorio donde se encuentra el programa y se procede a empaquetarlo con el comando mvn package:

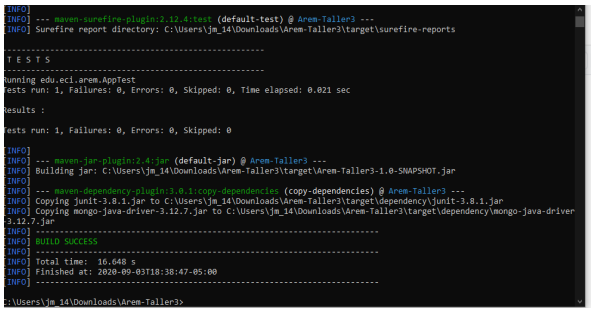


Figura 5:

6. Ejecucion

Para ejecutar el programa basta con acceder a linea de comandos windows y digitar el siguiente comando que encedera el servidor para iniciar el servidor web

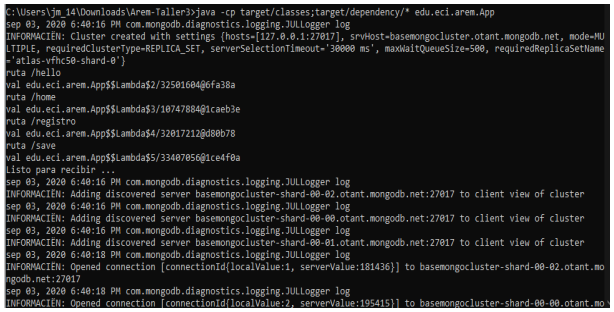


Figura 6:

Ahora solo basta con abrir el buscador y digitar localhost:36000 y se vera la pagina de inicia del servidor como se vera acontinuacion:

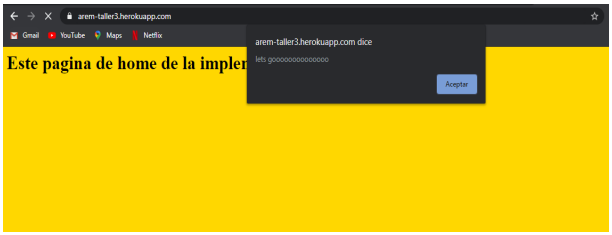


Figura 7:

7. Pruebas de EndPoints

Este aplicacion web fue diseñada para que pueda retornar recusos estaticos y dinamicos

7.1. Recursos Estaticos

Son los recursos como html,txt,js e imagenes ne formato PNG y JPG que se encuentran en la carpeta resource del proyecto y para acceder a ellos solo basta con digitar el nombre del recurso despues de la ruta del servidor con sus respectivo formato como se vera acontinuacion

- / o /index html

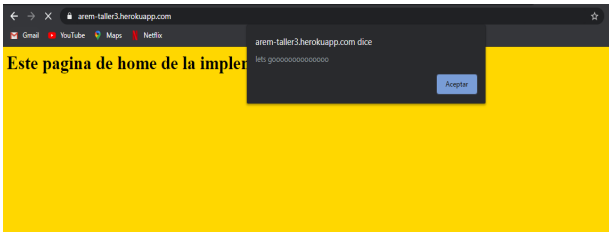


Figura 8:

- /bienvenida.txt

Figura 9:

- /perro.JPG

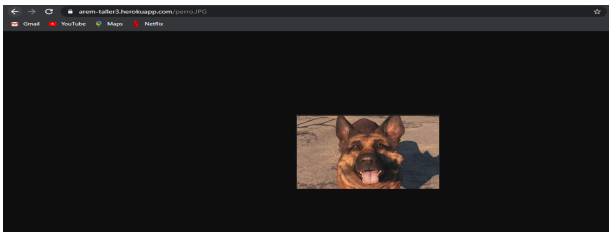


Figura 10:

- /nave.JPG

Figura 11:

- /eclipse.PNG

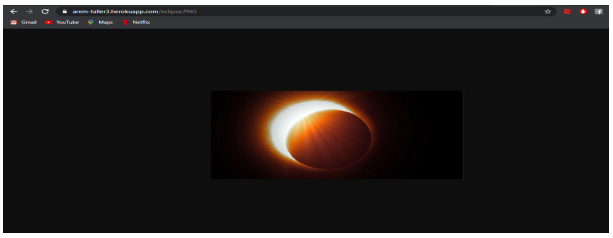


Figura 12:

7.2. Recursos Dinamicos

Son los endPoints que estan asignados a funciones apartir de funciones lambda y su path comienza con /Apps como se vera acontinuacion:

- /hello

Figura 13:

- /home

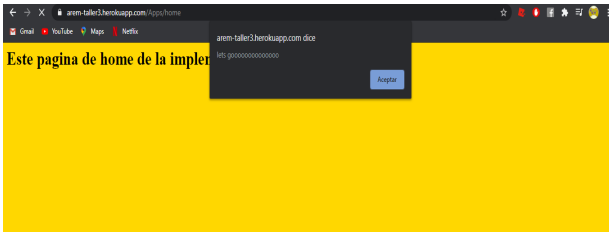


Figura 14:

- /registro

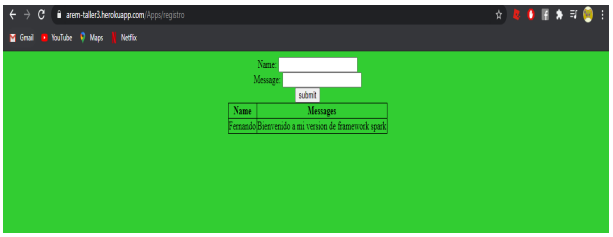


Figura 15:

8. Registro en Base de Datos

Desde el endpoint /registro como se vera acontinuacion se mostrara una tabla con los mensajes registrados en la base de datos Mongo BD hasta el momento y hay 2 text label para que un usurio pueda registrar su nombre y un nuevo mensaje

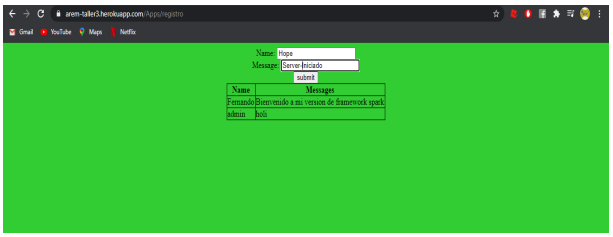


Figura 16:

Depues de diligenciar los campo de nombre y mensaje y oprimir el boton de submit saldra la siguinet pagina de registro exitoso

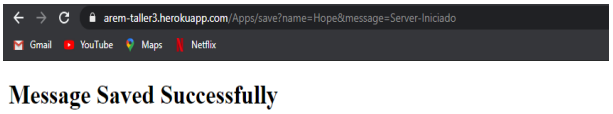


Figura 17:

Despues de 2 segundos esta pagina se redireccionara automaticamente a la pagina de registro donde en la tabla ya no aparecera el nuevo registro

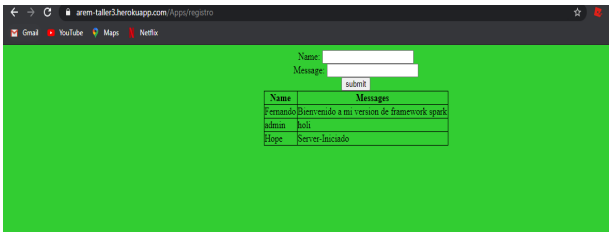


Figura 18:

9. Conclusion

Este taller fue util para comprender la arquietectura interna de los frameworks de servidores web como spark o spring y nos ayudo a visualizar como a punta de sockets ,drivers de conneccion a base de datos,requests y funciones lambda se puede implementar un servidor web con funcionalidades parecedidas a las de spark y spring

10. Bibliografía

[1] Codigo Facilito. *Git*. URL: https://codigofacilito.com/articulos/que-es-git. (entered: 16-08-2015).

[2] Heroku. *Heroku*. URL: https://es.wikipedia.org/wiki/Heroku. (entered: 20-12-2019).

[3] Funcion Lambda. *Funcion Lambda*. URL: https://www.tokioschool.com/noticias/expresiones-lambda-uso-programacion-aplicaciones/. (entered: 2020).

[4] Wikipedia. *Java*. URL: https://es.wikipedia.org/wiki/Java_(https://es.wikipedia.org/wiki/Java). (entered: 23-06-2020).

[5] Wikipedia. *Maven*. URL: https://es.wikipedia.org/wiki/Maven. (entered: 31-03-2020).

[6] wikipedia. *MongoDB*. URL: https://es.wikipedia.org/wiki/MongoDB. (entered: 27-08-2020).