



ESCUELA
COLOMBIANA
DE INGENIERÍA
JULIO GARAVITO

IMPLEMENTACIÓN DE UN WEVIDOR WEB

LUIS DANIEL BENAVIDES NAVARRO

Arquitecturas Empresariales

AUTOR:

JOHANN SEBASTIAN PÁEZ CAMPOS

SEPTIEMBRE 03 DEL 2020

Tabla de Contenidos

1. Introducción	2
2. Definiciones	2
3. Diseño y Arquitectura	2
4. Pruebas	3
5. Conclusiones	5
6. Referencias	5

1. Introducción

En este escrito se mostrará y se detallará cómo se implementó un servidor web desde cero utilizando el lenguaje de programación JAVA. Además de esto el aplicativo se conectará a una base de datos de Mongo Atlas, para poder consultar todos los datos y poder mostrarlos en el sitio web implementado, el cuál puede hacer peticiones get y post, para consultar todos los datos (en este caso animales) y añadir otros.

2. Definiciones

Socket: Un socket es conocido como un tipo de software que actúa como un punto final que funciona estableciendo un enlace de comunicación de red bidireccional entre el extremo del servidor y el programa receptor del cliente. [1]

Servidor Web: El servidor web (también llamado webserver en inglés) es el software que se encarga de despachar el contenido de un sitio web al usuario. [2]

Get: El método GET solicita un recurso del servidor indicado en el campo URI. Si la URI apunta a una base de datos de producción de recursos como un servlet, los datos serán devueltos dentro del mensaje de respuesta. [3]

Post: El método HTTP POST envía datos al servidor. El tipo del cuerpo de la solicitud es indicada por la cabecera Content-Type. [4]

3. Diseño y Arquitectura

En el aplicativo tiene una arquitectura muy sencilla en cuanto a la parte web. Se basa en un servidor web, en este caso es nuestro socket principal y el hilo que se estará ejecutando todo el tiempo, este servidor en cada solicitud http enviada, revisará las respectivas cabeceras y construira todo el mensaje para poder responder el mensaje correcto al usuario.

Esta solicitud o request, tendrá el tipo de petición, el path o ruta de lo que el usuario quiere que el servidor web responda y demás componentes necesarios para tratar los datos como el cuerpo del mensaje para añadir un nuevo dato.

Luego de haber tratado toda la solicitud y de haber guardado el contenido, se consulta la información de la base de datos de Mongo Atlas, o se inserta un nuevo dato en caso de la petición post, después se procede a responder con un encabezado http, mostrando el contenido solicitado por el usuario o un mensaje de error en caso de que no exista esa solicitud mediante las funciones incorporadas del socket en JAVA.

Después de cada respuesta, el servidor web sigue escuchando para aceptar más solicitudes.

En la figura 1 esta el diagrama de componentes que muestra gráficamente la arquitectura descrita anteriormente.

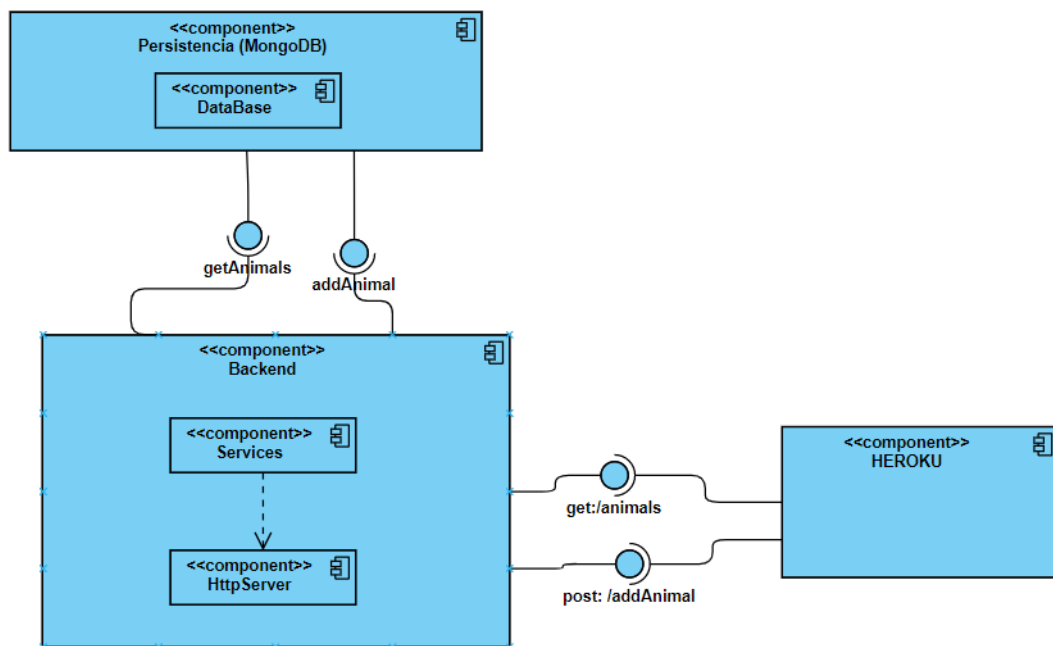


Figura 1: Representación de la arquitectura descrita anteriormente.

4. Pruebas

Verificando que el aplicativo web funciona correctamente, procedemos a mostrar algunos archivos principales como el `index.html` para probar que la solicitud `get` funciona correctamente al consultar en la base de datos y añadir un nuevo dato para probar que la petición `post` añade correctamente los datos.

Bienvenido!
Agrega Animales y Consúltalos Abajo

ANIMAL:

NOMBRE:

EDAD:

Añadir Animal

Animal	Nombre	Edad
Perro	Tony	18
Gato	Fufu	1
Pez	Doroti	2

Figura 2: Podemos observar que el aplicativo genera una tabla con los datos de los animales que se almacenan en la base de datos.

ANIMAL:

NOMBRE:

EDAD:

Añadir Animal

Animal	Nombre	Edad
Perro	Tony	18
Gato	Fufu	1
Pez	Doroti	2
Caracol	Gary	8

Figura 3: Podemos observar que el aplicativo realiza la petición post correctamente al agregar un nuevo animal, en este caso Gary el caracol.

5. Conclusiones

- Con las herramientas que nos ofrece el lenguaje de programación Java, es posible implementar un servidor web desde cero, aceptando solicitudes http.
- Es necesario tener conocimientos sobre los encabezados http y como se envían ya que estos son la base de como podemos tratar la información recibida y de como debemos enviarla para que el navegador pueda entender.
- La implementación del servidor web cumple con su propósito de recibir peticiones get y post, así como de renderizar archivos estáticos como JavaScript, Css, Html e imágenes.

6. Referencias

- [1] *Socket*, <https://www.speedcheck.org/es/wiki/socket/#fn1>, Accessed on 2020-09-03.
- [2] *Servidor Web*, <https://blog.infranetworking.com/servidor-web/>, Accessed on 2020-09-03.
- [3] *HTTP Request / Response*, <https://sites.google.com/site/conceptoprogramacion/request-response>, Accessed on 2020-09-03.
- [4] *POST*, <https://developer.mozilla.org/es/docs/Web/HTTP/Methods/POST>, Accessed on 2020-09-03.