



ARQUITECTURA EMPRESARIAL  
LUIS DANIEL BENAVIDES NAVARRO

---

# INFORME LABORATORIO 3 - TALLER CLIENTES Y SERVICIOS

---

*Autores:*  
Guillermo Castro

Febrero 2021

## Tabla de Contenido

<b>1. Introducción</b>	<b>2</b>
<b>2. Objetivos</b>	<b>2</b>
<b>3. Estructura,Diseño y Aquitectura del Programa</b>	<b>3</b>
3.1. Estructura . . . . .	3
3.2. Diseño . . . . .	3
3.3. Arquitectura . . . . .	4
<b>4. Pruebas</b>	<b>5</b>
<b>5. Conclusiones</b>	<b>7</b>
<b>6. Referencias</b>	<b>8</b>

## 1. Introducción

La transformación digital en el mundo ha impulsado el crecimiento del uso de servicios cloud en las organizaciones, esto convierte las necesidades de computación sobre la nube en una oportunidad vital para publicar los servicios que se necesiten de manera rápida, práctica y segura. El objetivo de este informe es el de definir arquitecturas y presentar resultados acerca de una implementación sobre la nube de un servidor web que emplee las características funcionales de un framework que muchas empresas y desarrolladores están usando actualmente en sus puestas a producción conocido cuyo nombre es Spark. Se van a presentar los diseños arquitecturales del servidor y se va a explicar detalladamente el funcionamiento de cada uno.

## 2. Objetivos

- Retomar conocimientos aprendidos de asignaturas pasadas, poniendo en práctica todos los sistemas complejos.
- Crear un servidor web sencillo con sockets desde cero y entender el ciclo de vida de un socket.
- Entender e implementar las peticiones web y el ciclo de una aplicación web desplegada
- Entender los fundamentos de Maven, Git, Heroku, CircleCI vía línea de comandos; Además de Aprender e implementar la integración continua y darse cuenta de su importancia en entornos reales.
- Entender cómo se construye un servidor web simple desde cero e Identificar los componentes de una arquitectura basada en cliente-servidor.

### 3. Estructura,Diseño y Aquitectura del Programa

#### 3.1. Estructura

Esta estructura la usamos para ver como se esta conformando el proyecto y lo podemos unir al readme creado donde vemos mas a fondo la descripcion del proyecto.

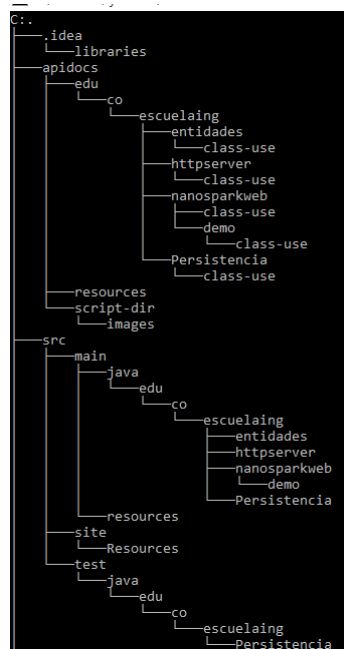


Figura 1: Estructura de árbol del proyecto creado

#### 3.2. Diseño

Aquí presento el diagrama del modelo de clases, se trata de una implementación básica de un servidor web como lo es Spark utilizando procesos de request y response implementado por mi persona [Ver figura 1]. este diseño está tomado del enunciado del taller

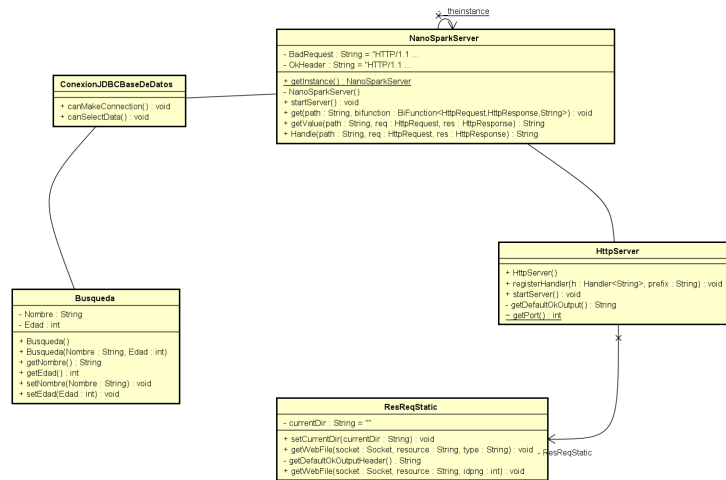


Figura 2: Diagrama de Clases creado en astah

### 3.3. 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 construirá 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.

## 4. Pruebas

Se realizaron dos tipos de pruebas mediante Junit en la cual se logro verificar el correcto funcionamiento de la aplicacion creada y en Heroku en la cual se verifico la web creada de manera correcta y funcional, mostrando la conexion a la base de datos creada con Postgres en Heroku.

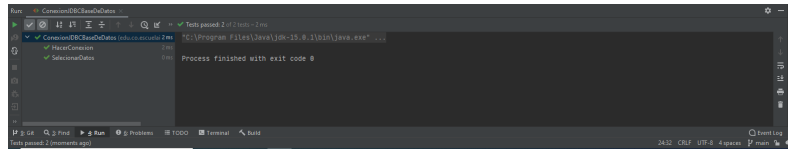


Figura 3: Pruebas

Se realizaron las pruebas con el despliegue en heroku llevado acabo y la aplicacion funcionando de manera optima

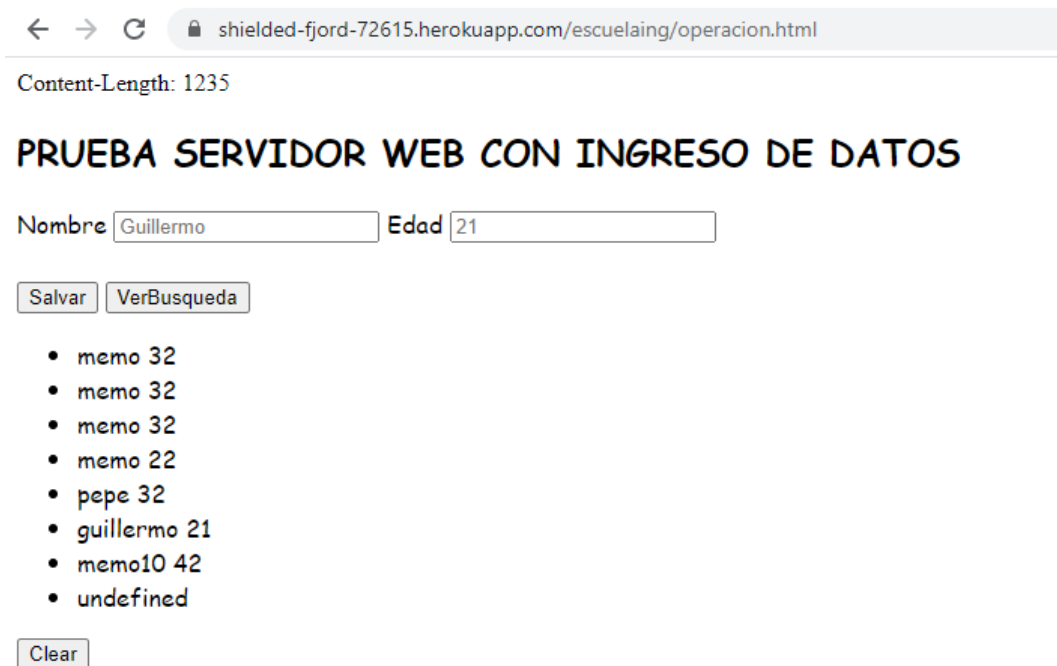


Figura 4: Prueba de despliegue en Heroku con ayuda del Spark creado

## INFORME LABORATORIO 3 - TALLER CLIENTES Y SERVICIOS

Guillermo Castro

y finalmente se hizo prueba del servidor web creado con el framework Spark por el puerto 4567

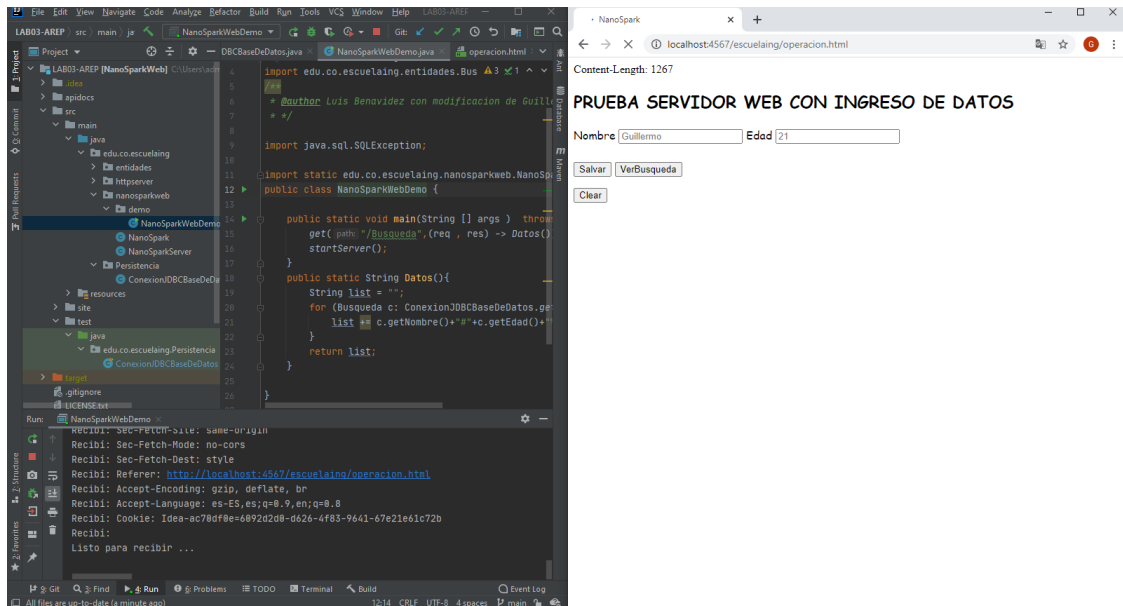


Figura 5: Prueba de Spark Web

## 5. Conclusiones

- Con las herramientas que nos ofrece el lenguaje de programación Java, es posible implementar un servidor web desde cero, aceptando solicitudes latp.
- Se realizó la implementación de los métodos solicitados para realizar el servidor web de manera óptima.
- Es necesario tener conocimientos sobre los encabezados http y cómo se envían ya que estos son la base de cómo podemos tratar la información recibida y de cómo debemos enviarla para que el navegador pueda entender.
- Se montó todo en Maven para una mejor gestión del Software y finalmente logramos mostrar nuestro Servidor en Heroku,
- 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.
- Esta actividad permitió conocer la arquitectura básica del framework de spark, su funcionamiento, comportamiento y su metodología. Además del comportamiento de los encabezados HTTP, estructura, envío y recepción.



## 6. Referencias

- Java printwriter. (2020-09-02) de <http://tutorials.jenkov.com/java-io/printwriter>
- What is a web server? ( 2020-09-02.) de <https://www.nginx.com/resources/glossary/web-server>
- Socket,(2018-09-03) de <https://www.speedcheck.org/es/wiki/socket/fn1>
- Lab3 AREP (20/02/2021) de Alan Marin Mendez.