

# Prototipo de Spark

Gómez Camacho, José Luis

Arquitecturas Empresariales

Ingeniería de sistemas, escuela colombiana de ingeniería julio garavito,  
Bogotá, Colombia

4 de Septiembre de 2020



**Resumen:** En este documento encontraremos una implementación experimental de Spark la cual se desarrolla con el fin de saber a grandes rasgos la estructura que tiene un framework como Spark. Veremos la estructura implantada para este experimento.

## 1. Introducción

En este momento la infraestructura en la nube ha tomado gran impulso dado que permite alcanzar recursos computacionales exageradamente potentes a precios accesibles y permitiendo que el consumidor se evite los esfuerzos de instalación, mantenimiento y compra de estos equipos. Consecuencia de esto, muchas compañías pueden diseñar sus aplicaciones web y llevarlas a un entorno de producción en considerable menos tiempo, sin embargo muchos de los frameworks usados presentemente tienden a ser pesados y complejos de utilizar por lo que requieren de excesivo tiempo y esfuerzo de desarrollo, como solución a esto se están desarrollando micro-frameworks que permitan facilitar y acelerar la creación de estas aplicaciones un ejemplo de esto es Spark. El objetivo de este experimento es recrear Spark la cual deberá resolver peticiones get con función lambda y permitirá acceder a recursos estáticos como páginas, JavaScript y CSSs. Para probar esta funcionalidad se conectará con una base de datos y por ultimo será desplegada en Heroku. Todo esto con el objetivo de conocer la estructura que utilizan estos micro-Frameworks lo dividiremos en dos partes en como leer y proyectar una imagen y luego como poder insertar datos y proyectar información

## 2. Metodología

### 2.1. Diseño

Ante el primer problema que nos enfrentamos es como abrir un puerto para enviar y recibir las peticiones para eso hacemos uso de un socket el cual es el que se encargara de unir la información del servidor a nuestra aplicación. Luego de

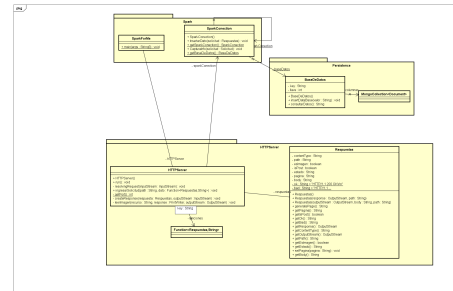


Figura 1: Diagrama 1

esto tendremos que analizar las peticiones que se realicen del servidor para esto se utiliza un manejo de cadenas para identificar el tipo de peticiones que se desean. Estas son bastante largas pero lo único que nos interesará será la petición que nos que se identifica por el verbo GET o POST. Después de esto encontraremos una ruta del recurso deseado. Si es un HTML o una imagen la intentaremos abrir y para eso utilizaremos si no se encuentra el recurso se retornará un error 404 informando que no podemos encontrar el recurso. Si encontramos el archivo lo enviaremos a proyectar para que se vea desde el browser. Esto deberá tener un encabezado indicando que tipo de texto se va a proyectar. Esta solución se encuentra en la clase HTTPServer junto con la clase Respues-

tas. Una vez atendido estos problemas pasaremos a la conexión a la base de datos para esto utilizaremos MongoDB la cual es una herramienta que nos permite crear bases de datos no relacionales de forma gratuita. Para la conexión necesitaremos una clase que represente el base de datos a la cual se le realizarán peticiones de inserción y consulta y por supuesto de su creación. Por último, necesitaremos una clase que nos una la cual será SparkConexion la cual se encargará de transformar la petición que nos da el servidor en el formato específico para inserción en la base de datos. Para dar por terminado el experimento diseñaremos una página HTML que nos permita insertar datos y otra que nos permita consultarlos. Las cuales son mostradas a continuación.



The image shows a web form with a title "Insercion en una base de datos". Below the title is a label "Ingresar datos" above a text input field. At the bottom of the form is a green button with the text "Ingresar".

**Figura 2:** Aplicación desplegada

## 2.2. Construcción

Para nosotros el lenguaje que mejor se acomoda a este problema es Java con una estructura Maven y pruebas en JUnit y publicado en GitHub. Para el front utilizaremos HTML ya que es la mejor herramienta para este tipo de aplicativos sencillos y JavaScript para tener una comunicación fácil y efectiva con la capa lógica.

## 3. Conclusiones

Durante el experimento nos pudimos dar cuenta del gran trabajo que deben hacer los frameworks para facilitar a los programadores la creación de estos tipos de aplicaciones ya que algunos con unas simples etiquetas ya son capaces de proyectar imágenes, páginas, imágenes y distintos recursos lo cuales si se tuvieran que hacer sin la ayuda de los frameworks aumentaría el tiempo de trabajo ya que surgen muchos problemas y vacíos que no se contemplan cuando se realiza sin ellos.

Por último pudimos observar que realizar aplicaciones web sin la ayuda de estos frameworks sería posible pero se tendrían que destinar recur-

sos innecesarios para la compañía para crear su propia forma de proyectar sus aplicaciones.

## 4. Bibliografía

- masadelante. (s. f.). ¿Que es un Socket? - Definición de Socket. Recuperado 3 de septiembre de 2020, de <https://www.masadelante.com/faqs/socket>
- C. (2017, 16 octubre). path. Microsoft Docs. <https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/path>
- Spark. (s. f.). Spark Framework: An expressive web framework for Kotlin and Java. sparkjava. Recuperado 20 de agosto de 2020, de <http://sparkjava.com/>