

# Introducción a esquemas de nombres, redes, clientes y servicios con Java

Ingenieria de Sistemas.' AREP-2021-1. Arquitecturas Empresariales' Luis Daniel Benavides.  
ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO  
Lab 3 AREP: Fabian Mauricio Ramirez Pinto

1

**Abstract** / *In this laboratory practice 3, of the AREP subject, the development of a web server is shown, by means of the deployment with the Heroku tool. This web server allows the support of multiple requirements executed in a queue, but not concurrently. The web server has the ability to return all the requested requirements in file format. In addition, it has a website that was built with JavaScript and with the use of our web server. Thus allowing the operation of a basic framework and experimental use, similar to Spark.*

/

Contacto / [fabian.ramirez-p@mail.escuelaing.edu.co](mailto:fabian.ramirez-p@mail.escuelaing.edu.co)

## 1. Introducción

En este laboratorio 3 Introducción a esquemas de nombres, redes, clientes y servicios con Java, se procedió a desarrollar un framework web experimental en el cual tuviera la capacidad de ofrecer servicios similares a los de un framework Spark. El framework brinda la capacidad de ofrecer y resolver peticiones por medio del uso de funciones lambda y permitir el uso de recursos como paginas HTML, imágenes, archivos, archivos JavaScript, entre otros. El usuario hará uso de una base de datos PostgreSQL, en donde deberá adjuntar los datos correspondientes y leer datos de la misma. El uso de este framework es por razones de menor complejidad, debido a que el uso de Framework como Spark son mas para ambientes empresariales y no es conveniente para el uso de pequeños proyectos y de menor complejidad.

Aparte de lo ya descrito anteriormente, se procedió a realizar dos desafíos planteados en los cuales consistía coger practica y contexto para este laboratorio, este consistía en el primero en desplegar un servicio web que soportara múltiples tareas, en fila, mas no de forma recurrente, haciendo uso de los servicios de construcción de JavaScript. El segundo reto consistía en escribir un framework similar a Spark, pero mas básico, permitiendo realizar las funciones mencionadas en la parte de arriba, pero con la conexión a un servicio de base datos para realizar la prueba de solución.

## 2. Desarrollo

La implementación comienza con el uso de un Framework simple, similar al del framework de Spark, en donde se hace uso de funciones lambda, para que permita el uso de recursos estáticos como lo son Paginas HTML, imágenes y entre otro tipo de archivos. Después continuamos con la arquitectura de http, en donde se crea un http server, que será la encargada de procesar las peticiones y manejar el puerto por donde se van a realizar las

POR FAVOR ELEGIR TIPO DE PRESENTACIÓN

solicitudes "Java-net". Luego es necesario que creamos una clase referente al framework de Spark las cuales nos permitirán indicar cual es el recurso para utilizar y establecer el Path correspondiente. Todas las peticiones de httpServer se deben manejar de la misma manera.

Luego de realizar la implementación en HTTP, procedemos hacer uso de una base de datos como lo es PostgreSQL, la cual se encargará de realizar la conexión con la base de datos remota, a partir de la clase del código que le hayamos asignado en el código, la cual se encargará de realizar la petición de la inserción y consultar los datos, almacenando una lista de datos y sumándolos a los mismos. Luego se procede a realizar el correcto funcionamiento haciendo uso de framework y de la aplicación en general, permitiéndonos realizar las peticiones en el HTML, en el JavaScript y en una imagen de archivo.

El despliegue y construcción se hace con el uso de java 11, debido a la facilidad que brinda dicha herramienta al momento de utilizar el paquete de "java-net-http". Adicionalmente hacemos uso del manejo de dependencias de Maven para tener una integración completa con todos los componentes. Para finalizar la herramienta que nos permitió el despliegue es Heroku App, como servidor, debido a la buena integración que tiene con GitHub y Java.

## 3. Conceptos Basicos

### 3.1. URLs

Las listas enlazadas son un tipo de datos abstracto común utilizado para mantener colecciones de datos. Las listas enlazadas se implementan con punteros. Una lista enlazada suele tener dos componentes - cabeza de lista - nodo(s) de la lista

Algunas de las opciones de la estructura de la lista enlazada son - la cabeza de la lista puede apuntar al primer nodo, al último o a ambos - un nodo de la lista

Macro: BAAA62 con instrucciones de estilo

puede apuntar al siguiente nodo, al anterior o a ambos. Los punteros nulos se utilizan a menudo para indicar una lista vacía o el final de la lista. Las operaciones típicas en una lista enlazada son - añadir un nodo - eliminar un nodo - nodo siguiente - nodo anterior

### 3.2. TCP

El estado actual de desarrollo del protocolo TCP permite establecer una conexión entre dos puntos terminales en una red informática común que posibilite un intercambio mutuo de datos. En este proceso, cualquier pérdida de datos se detecta y resuelve, por lo que se considera un protocolo fiable. Dentro de la familia de protocolos de Internet, el TCP, junto con el UDP y el SCTP forma el grupo de los protocolos de transporte, que, según el modelo OSI, se ubican en la capa de transporte dentro de la arquitectura de red. Como el protocolo TCP se combina casi en todos los casos con el protocolo de Internet (IP) y esta conexión forma la base de la gran mayoría de redes locales y servicios de red, es común hablar del conjunto de protocolos TCP/IP, aunque en realidad se haga referencia a la familia de protocolos de Internet.

### 3.3. UDP

El protocolo de datagramas de usuario, abreviado como UDP, es un protocolo que permite la transmisión sin conexión de datagramas en redes basadas en IP. Para obtener los servicios deseados en los hosts de destino, se basa en los puertos que están listados como uno de los campos principales en la cabecera UDP. Como muchos otros protocolos de red, UDP pertenece a la familia de protocolos de Internet, por lo que debe clasificarse en el nivel de transporte y, en consecuencia, se encuentra en una capa intermedia entre la capa de red y la capa de aplicación.

### 3.4. Sockets

Los sockets son los puntos finales del enlace de comunicación entre dos programas ejecutándose en la red. Cada socket está vinculado a un puerto específico, así la capa que implementa el protocolo TCP puede saber a qué aplicación enviar los mensajes. En general un servidor es un proceso que se ejecuta y tiene un socket, vinculado a un puerto, que está esperando solicitudes de clientes externos. Los sockets son una abstracción de más bajo nivel que las URLs y sirven para implementar protocolos de comunicación cliente-servidor. El protocolo cliente servidor consiste en un programa cliente que hace solicitudes a un programa servidor que atiende dichas solicitudes. Java provee dos clases para manejar la comunicación por medio de sockets: Socket y ServerSocket. Ambas clases se encuentran en el paquete java.net. NOTA: Una idea clara para entender los sockets es imaginar que son los 4 enchufes donde se conectan las aplicaciones para comunicarse

### 3.5. Datagramas

Un datagrama es un mensaje independiente autocontenido que es enviado a través de la red, y cuya llegada, tiempo de llegada y contenido no son garantizados. Estos datagramas son útiles para implementar servicios cuyos mensajes no tienen un contenido del cual dependen procesos fundamentales. Por ejemplo usted quiere que la comunicación entre un avión y la torre de control sea inmediata y garantizada, sin embargo, si tiene una página que muestra el estado del tiempo en la playa, no le importa si el último mensaje es de hace 1 hora y de pronto no es tan exacto.

### 3.6. Stubs y Skeletons

La Spark Web App es un servicio de computación que permite tener el acceso de un usuario para poder aplicar servicios. La web app permite la conexión entre la interfaz y la calculadora. Haciendo uso de las dos operaciones que usa la calculadora, esta permitirá calcular los valores correspondientes mediante una Linked List. Además, esto cambia la lectura requerida a la hora de realizar la petición el usuario, permitiendo finalmente la respuesta que el usuario hizo, mediante el formato HTML.

### 3.7. Base de Datos Firebase

Todos los datos de Firebase Realtime Database se almacenan como objetos JSON. La base de datos puede conceptualizarse como un árbol JSON alojado en la nube. A diferencia de una base de datos de SQL, no hay tablas ni registros. Cuando le agregas datos al árbol JSON, estos se convierten en un nodo de la estructura JSON existente con una clave asociada. Puedes entregar tus propias claves, como ID de usuario o nombres semánticos, o también puedes obtenerlas mediante el método push().

## 4. Resultados

Los resultados arrojan el uso correcto del framework simple, similar al de Spark, en donde se puede ver desplegado el servicio web, a partir del servidor web usado y las herramientas descritas anteriormente. Mostrando el uso de funcionalidades que tiene el mismo laboratorio y que permiten evidenciar el adecuado uso de este.

## 5. Conclusiones

Se aprendió de otro tipo de framework, a diferencia del de Spark usado en el laboratorio 2, también se aprendió hacer uso de funciones de Http para generar peticiones y responder requerimientos, acorde a lo solicitado. Se aprendió a construir servidores con Http, en donde podemos evidenciar parcialmente el funcionamiento interno de framework web, en donde se puede implementar desde cero y ofreciendo la velocidad y procesamiento de sentencias acorde a las necesarias. Se aprecia la dificultad que se puede evidenciar a la hora de implementar un framework desde cero, pues framework como Spark

facilitan mucho las implementaciones y las conexiones con los servidores web.

## 6. Referencias

[1] Benavides(2021). Introducci' on a esquemas de nombres, redes, clientes y servicios con Java. Arquitectura empresarial, Bogotá, Colombia.

[2] Digital Guide IONOS.Transport Control Protocol [online] Available at:

<https://www.ionos.es/digitalguide/servidores/know-how/que-es-tcp-transport-control-protocol/> 2021].

[3] Digital Guide IONOS. User Datagram Protocol [online] Available at:

<https://www.ionos.es/digitalguide/servidores/know-how/udp-user-datagram-protocol/> 2021].

[4] FireBase Realtime Database. [online] Available at:

<https://firebase.google.com/docs/database/web/structuredata?hl=es/> 2021].