

ESCUELA COLOMBIANA DE INGENIERIA JULIO GARAVITO

# Laboratorio 5 Arquitectura Empresarial

por

David Mateo González Grimaldos

en el programa de  
Ingeniería de Sistemas

Marzo 2020

# Taller de introducción a AWS y modularización por virtualización

## Resumen

En el presente laboratorio logramos desarrollar un servidor Web concurrente que es capaz de soportar una serie de archivos (.png , .html o .js ) o recursos de manera mas eficiente gracias a los objetos **Thread**, este sera desplegado en *Heroku* y consumido por una maquina virtual creada mediante *EC2* de AWS Educate la cual simulara un cliente, para esto emplearemos el protocolo de transferencia de archivos *SFTP* y el protocolo de acceso remoto a servidores *SSH*.

## Introducción

En el presente laboratorio se desplegara un servidor web recurrente capaz de soportar múltiples búsquedas al mismo tiempo, esto mediados por el estilo arquitectónico *Cliente Servidor*, y el ambiente de despliegue continuo *Heroku*, para el cual se cargaran varios archivos con un formato *png* (imágenes) o *html* simulando el ambiente de una pagina web, de igual manera se creara una maquina virtual con AWS Educate, la cual resolverá las peticiones simulando las funcionalidades de un *Cliente*.

AWS Educate es una iniciativa global de Amazon para proveer a los estudiantes de recursos integrales y capacitarles en tecnología de nube. Con la vista puesta en las experiencias de formación aplicada en el mundo real, AWS Educate ofrece a los jóvenes el acceso a contenidos adaptados a sus conocimientos, diseñados para introducirles en tecnologías informáticas de nube que potencian la innovación en ámbitos profesionales como inteligencia artificial, reconocimiento facial y de voz, juegos y avances médicos, entre otras cosas.

Java posibilita la programación concurrente a través de threads. Los threads son procesos ligeros, con línea de flujo de control propia pero que comparte el espacio de direcciones del programa. Los threads hacen posible la ejecución concurrente de código.

## Contenido

El desarrollo arquitectónico del servidor web requiere ser capaces de garantizar la correcta lectura de los formatos que brindan los archivos *.jpg*, *.png* y los formatos de las paginas web *html* o *.txt*, ubicados en los recursos de nuestro src, una vez capturemos estos datos mediados por la implementación de java de la clase *BufferedReader* seremos capaces de transmitir la información a nuestro servidor.

De igual manera crearemos una serie de recursos o métodos que puedan leer el contenido de diferentes paginas web, contenido que podrá ser visualizado en la maquina virtual desplegada por AWS educate, para esto se creara una llave y se hará una copia del *.jar* o archivo que contenga las clases de nuestro proyecto, dentro de las cuales encontraremos el servidor concurrente y el *clienteConcurrente* que obtendrá como parametros la URL de la pagina web desplegada en Heroku y el número de hilos que se ejecutaran.

```
[ec2-user@ip-172-31-39-119 ~]$ java edu/eci/arep/laboratorio5/client/clienteConcurrente https://lab5arep.herokuapp.com/campus.html 5
Contenido cliente #1
<!DOCTYPE html> <html><head><meta charset="UTF-8"><title>CampusEci</title></head><body><h1>Bienvenido al Campus de la escuela Colombiana de ingenieria</h1></span></body></html>
Contenido cliente #2
<!DOCTYPE html> <html><head><meta charset="UTF-8"><title>CampusEci</title></head><body><h1>Bienvenido al Campus de la escuela Colombiana de ingenieria</h1></span></body></html>
Contenido cliente #3
<!DOCTYPE html> <html><head><meta charset="UTF-8"><title>CampusEci</title></head><body><h1>Bienvenido al Campus de la escuela Colombiana de ingenieria</h1></span></body></html>
Contenido cliente #4
<!DOCTYPE html> <html><head><meta charset="UTF-8"><title>CampusEci</title></head><body><h1>Bienvenido al Campus de la escuela Colombiana de ingenieria</h1></span></body></html>
Contenido cliente #5
<!DOCTYPE html> <html><head><meta charset="UTF-8"><title>CampusEci</title></head><body><h1>Bienvenido al Campus de la escuela Colombiana de ingenieria</h1></span></body></html>
Tiempo total: 0.863021835
[ec2-user@ip-172-31-39-119 ~]$
```

## Conclusiones

- Mediante el presente laboratorio se aprendió sobre la utilidad de AWS como metodo para desplegar contenido en la nube.

- Se comprenden los beneficios que otorga la virtualización como método de abstracción de recursos, emulando en esta ocasión a un cliente.
- Se concluye la utilidad de desplegar nuestras aplicaciones web en el ámbito de Heroku , aunque se reconocen su limitaciones en cuanto al tamaño y flujo de datos.

## Bibliografía

- Amazon Web Services, Inc. (2020). AWS Educate. [online] Available at: <https://aws.amazon.com/and-older/> [Accessed 5 Mar. 2020].
- Ctr.unican.es. (2020). [online] Available at: <https://www.ctr.unican.es/asignaturas/procodis3ii/d> [Accessed 5 Mar. 2020].
- Amazon Web Services, Inc. (2020). AWS — Elastic compute cloud (EC2) de capacidad modificable en la nube. [online] Available at: <https://aws.amazon.com/es/ec2/> [Accessed 5 Mar. 2020].