**ARTÍCULO PROYECTO TERCIO II**

**ANDRÉS FELIPE PARDO MESA**

**JUAN DAVID RAMIREZ MENDOZA**

**DAVID FELIPE VACA SANTA**

**ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO**

**INGENIERÍA DE SISTEMAS**

**ARQUITECTURAS EMPRESARIALES**

**BOGOTÁ D.C.**

**ABRIL DE 2018**

**ÍNDICE**

1. INTRODUCCIÓN
2. DESCRIPCIÓN DEL PROYECTO
3. DESCRIPCIÓN DE ARQUITECTURA
4. ENLACES

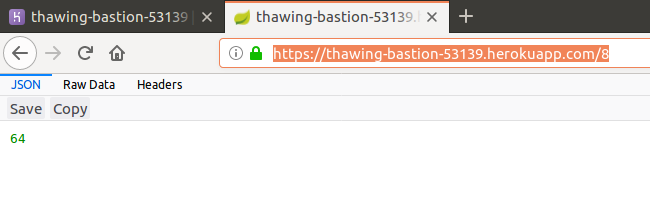
**INTRODUCCIÓN**

Este artículo busca explicar de forma detallada el proyecto y la arquitectura realizadas para el segundo tercio del curso Arquitecturas Empresariales.

En este artículo se puede evidenciar cómo se aplicaron los conceptos aprendidos en el tercio en la elaboración del proyecto y la interpretación y solución que le dimos a este para llegar a obtener un resultado final.

**DESCRIPCIÓN DEL PROYECTO**

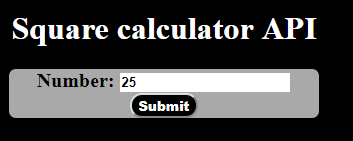
La primera parte del proyecto consiste en una aplicación REST que al recibir una petición GET con un número, devuelve el cuadrado de este, la aplicación fue desplegada en heroku y se puede utilizar usando la siguiente URL: <https://thawing-bastion-53139.herokuapp.com/numeroquedeseaconsultar>.



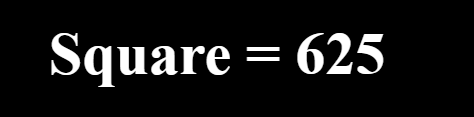
La segunda parte consiste en reutilizar un servidor de peticiones concurrentes realizado posteriormente y convertirlo en un servidor de aplicaciones, para esto se utilizó el principio de inversión de control. Para extender este servidor creamos una clase bean que es inyectada al thread que recibe la petición, luego en este bean se inyecta con etiquetas spring el api que será utilizada para realizar la operación con el número dado por el usuario.

La tercera parte consiste en una aplicación web donde el usuario a través de un formulario inserta un número, luego de que el bean recibe el número utiliza el api desplegado en heroku definido anteriormente para obtener el resultado que el usuario espera.

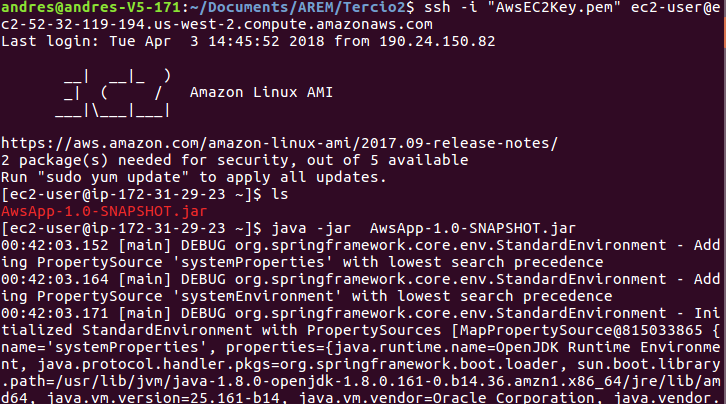


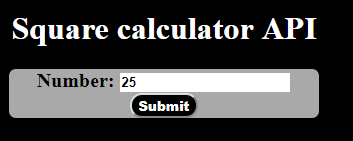


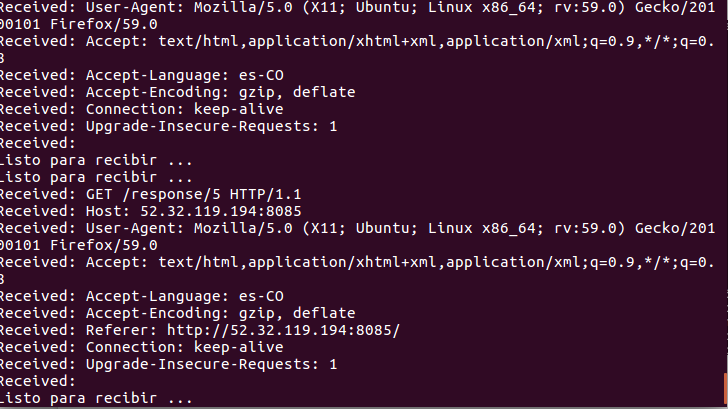


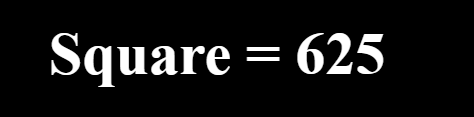


Por último se creó una máquina virtual en AWS y allí se desplegó el servidor realizado utilizando el archivo compilado del servidor con extensión .jar, para acceder al servidor desplegado en AWS utilizamos la siguiente URL: <http://52.32.119.194:8085>

****

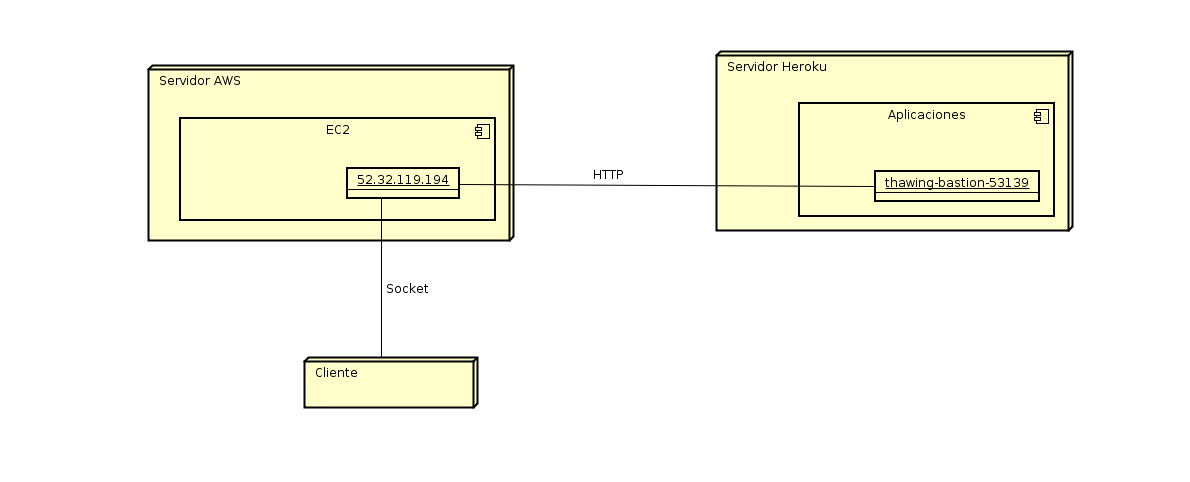


****



**DESCRIPCIÓN DE ARQUITECTURA**

En la arquitectura base tenemos como punto de inicio un cliente el cual realiza una solicitud a un servidor, pero no solo se trata de una simple solicitud, esta opera de la siguiente manera: El cliente se conecta por medio de un Socket (Conducto que permite el intercambio de información entre dos puntos por medio del protocolo TCP) a una Máquina virtual, la cual está desarrollada y alojada en el EC2 (Elastic Compute Cloud) de los servidores de AWS (Amazon Web Services), esta máquina virtual corre el servidor web de aplicaciones que se conecta por medio del protocolo HTTP, realiza una solicitud a la aplicación encargada del procesamiento, esta aplicación se aloja en un servidor web llamado Heroku, el cual nos permite manejar aplicaciones "en la nube" de manera que se tenga en cuenta el flujo y manejo de las solicitudes de acuerdo a nuestro diseño y desarrollo, esta aplicación nos retorna la información procesada por el mismo canal de comunicación y el Servidor AWS se encarga de hacer llegar al cliente dicha información por medio del socket inicial.



**ENLACES**

**Repositorio GITHUB**

<https://github.com/AREMProyectos/RepositorioProyecto2>

**API Rest Square Heroku** <https://thawing-bastion-53139.herokuapp.com/numeroquedeseaconsultar>

**Servidor WEB AWS**

<http://52.32.119.194:8085>