

Introducción AWS

First A. Juan Pablo Ospina Henao

I. INTRODUCCIÓN

En este artículo tiene como objetivo la construcción de un servidor Web concurrente (tipo apache) en Java y la elaboración de un cliente que permita realizar peticiones al servidor de forma concurrente. El servidor debe tener ciertos requisitos como ser capaz de entender páginas HTML e imágenes. Además de eso el servidor debe proveer un framework IoC para la construcción de aplicaciones a partir de POJOS. Para este caso se tendrá una aplicación básica de calculadora la cual podremos realizar actividades de suma, resta, multiplicación, división y potencia. Además de ver la funcionalidad de la aplicación pasaremos a explicar dos modelos importantes para el desarrollo de cualquier aplicación los cuales son arquitectura de software y arquitectura de despliegue. Estas herramientas nos permitirán ver de mejor forma la composición del software. Y por último se tendrá un espacio para ver el enlace a Heroku el cual explicaremos todas las funcionalidades del software. Además de ello se realizará el cliente de forma que llame repetidas veces al servidor siempre y cuando lo haga de manera concurrente.

II. DISEÑO

Para el diseño se tuvo en cuenta 4 componentes importantes para el desarrollo del proyecto entre ellos están:

- Server

La capa de servidor construye el medio para comunicarse dos computadores por medio de los sockets, donde el servidor espera cualquier petición de cualquier cliente.

- Framework

Esta capa es la encargada de poder revisar todos los métodos de la aplicación del cliente, en donde garantiza el funcionamiento de los métodos si se llaman al recurso.

- App

Es la aplicación del cliente en donde esta todas las funcionalidades que quiere realizar por medio del servidor

- Controlador

Por último, se encuentra el controlador, es el que inicializa el server y el framwork para que pueda empezar a escuchar y recibir request de peticiones que se le hagan.

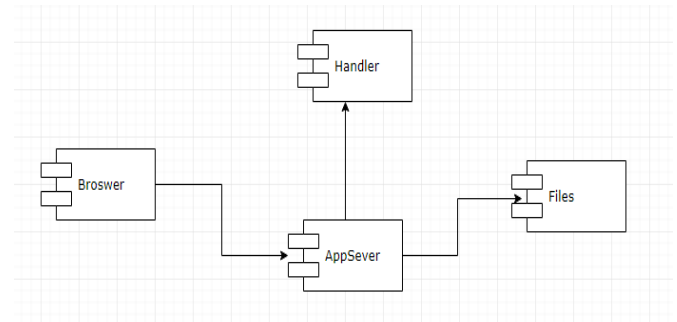


DIAGRAMA 1. Arquitectura de software

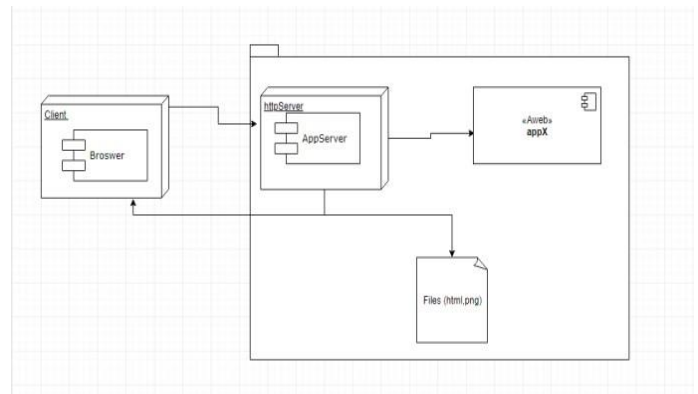
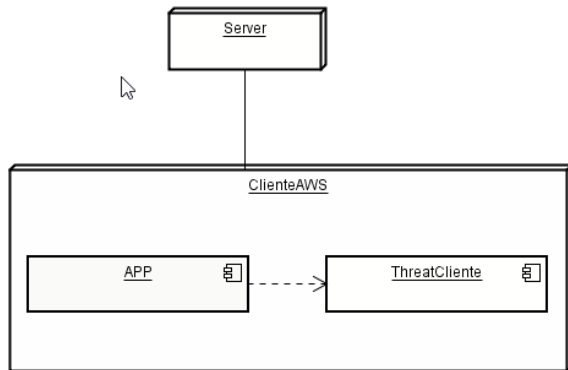


DIAGRAMA 2. Arquitectura despliegue

Está compuesto por los diferentes recursos, los cuales se pueden encontrar en el índice de la aplicación, de aquí se generan los diferentes vínculos a las demás páginas web, cabe recalcar que al necesitar un servidor de alta concurrencia, cuando se genera el socket para su comunicación a partir de ahí se puede acceder de manera independiente a los recursos de la aplicación que se encuentra en el servidor

III. ESTRUCTURA DEL CLIENTE



El cliente está compuesto de dos clases las cuales una se encarga de realizar la creación de diferentes threads dependiendo de la cantidad especificada y la otra es la encargada de ejecutar el código específico del cliente.

IV. DESCRIPCION APLICACIÓN WEB DESARROLLADA EN LA PLATAFORMA

Para este proyecto como bien se había dicho debe cumplir ciertos requerimientos los cuales son:

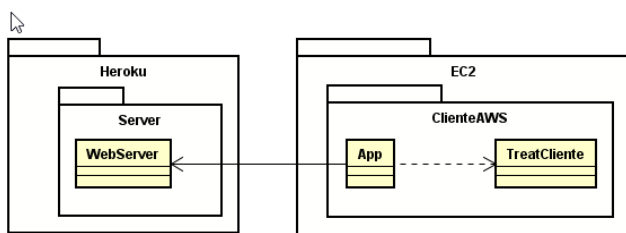
- Servidor web
- Poder ser capaz de entregar imágenes tipo png y páginas HTML
- Tiene que tener un framework IoC a partir de pojos
- Debe estar desplegado en Heroku

V. DESCRIPCION DEL CLIENTE

El cliente tiene como objetivo leer el código HTML de cualquier url que se le ingrese como parámetro, como se quiere ver la funcionalidad de un servidor concurrete se realiza median threads

VI. DESCRIPCIÓN AWS

Con el fin de ver el objetivo primordial de servicio en la nube es necesario desplegar la aplicación de cliente en una máquina virtual ec2 de Amazon.



Como se puede observar en la anterior figura el cliente se encuentra alojado en una máquina virtual de Amazon Web Services, donde se harán múltiples solicitudes en paralelo al aplicativo desplegado en heroku.

VII. EXPERIMENTOS

Para el primer experimento se realizan 5 petición a un servidor no concurrente, como es de esperarse solamente puede atender la siguiente hasta que el servidor termine con una de estas, como lo podemos ver en la siguiente imagen:

Por lo que es necesario implementar concurrencia para que el servidor pueda replicar a cada una de ellas ya que no sale el error 503 de dsponibilidad.

- Cliente concurrente vs Servidor concurrente

Este experimento la idea es realizar diferentes peticiones al servidor por medio de la maquina virtual de Linux en aws en donde al ejecutar para este caso 20 solicitudes se atienden casi inmediatamente.

```

$ curl -i -X GET -H "Host: 172.31.33.88" http://172.31.33.88:8080/
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 12
Date: Mon, 10 Sep 2018 18:18:18 GMT
Server: Apache/2.4.18 (Ubuntu)

```

VIII. REFERENCIAS

- LuisDaniel Benavides Navarro, “03Meta-Reflection-Annotations”
 Diapositiva. Ciudad de Publicación, Ciudad: Colombia.
- LuisDaniel Benavides Navarro, “ClientesServiciosEIntegracion” Diapositiva.
 Ciudad de Publicación, Ciudad: Colombia.
- LuisDaniel Benavides Navarro, “TallerClientesYServicios” Diapositiva. Ciudad de Publicación, Ciudad: Colombia.
- Jorge V. (2011). Sockets en Java (cliente y servidor): Codigoprogramacion. Recuperado de <http://codigoprogramacion.com/cursos/java/103-sockets-en-java-con-cliente-y-servidor.html#W5W3YM5KjIU>