



**CE-5508: Arquitectura orientada a
servicios emergentes**

Documentación de primer proyecto

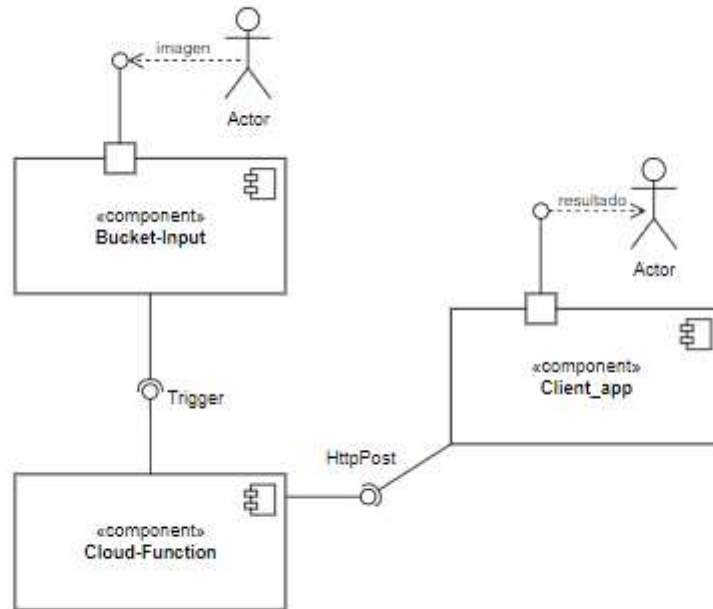
Estudiante:

Andrés Artavia López - 2017075875

Docente: Alejandra Bolaños

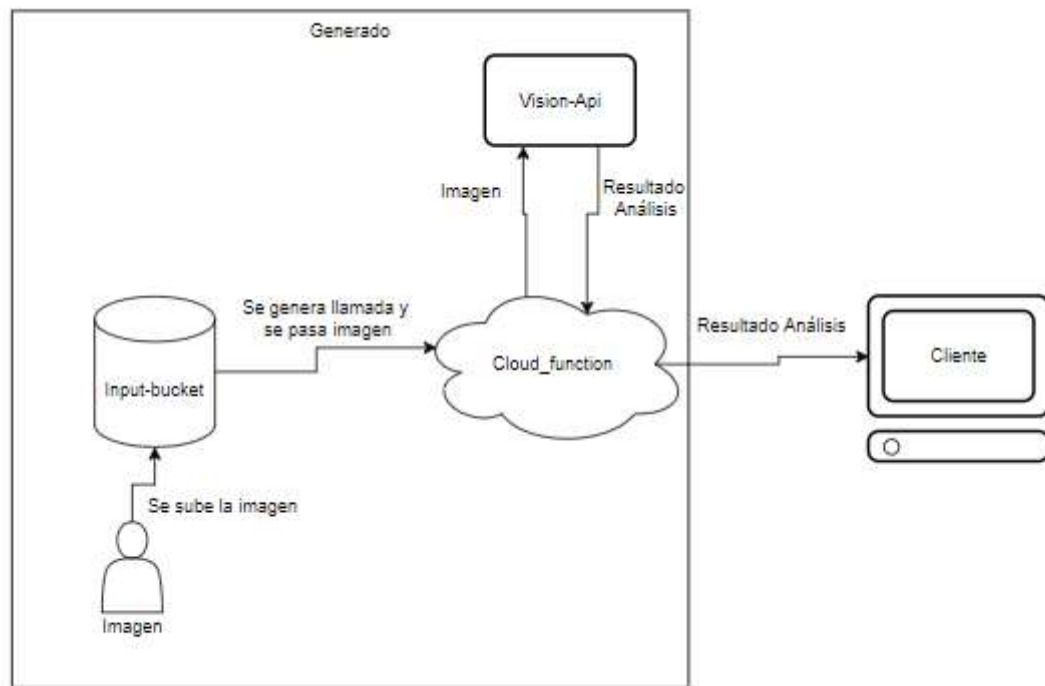
I Semestre 2022

1. Componentes y conexiones



Se tienen tres componentes principales, los principales son el *bucket* de entrada y el *cloud function*. El primero se encarga de servir de interfaz para que el usuario coloque la imagen a analizar, este se accede directamente desde la página de *Gcloud* y se sube la imagen como un repositorio de archivos cualquiera. Esta imagen dispara un evento que es dirigido hacia el otro componente, el *cloud function*. Este componente recibe el evento y este a su vez contiene el nombre del objeto que lo generó, en este caso, la imagen. La función toma el nombre de la imagen y le añade la dirección completa de acceso al recurso. Luego con esta dirección genera un *request* http al Vision-API con la dirección del recurso. Cuando el Api le contesta, genera otro *request* http que contiene los resultados del análisis y la envía al tercer componente. Este componente es un servidor http sencillo que recibe este mensaje e imprime los resultados en consola.

2. Arquitectura de la infraestructura



La infraestructura se compone de dos partes, la generada y la programada. La primera, es generada con Terraform y se compone de 3 componentes. Se tiene un *bucket* para manejar los datos de entrada y que funciona para generar eventos cuando se agrega un archivo. Luego se tiene un segundo *bucket* que contiene el código de la función y un objeto *cloud function* asociado, en el primero se sube el código y en el segundo, se ejecuta el código. El último componente generado es el Vision-API que tomará la imagen y la analizará.

En segundo lugar, se tiene el componente cliente, que no es generado, si no que es una aplicación de consola sencilla, que corre el servidor que escucha los resultados del análisis.

3. Decisiones sobre el proyecto

a. Elección de herramienta CD/DI

Para esto se tienen dos opciones principales, Jenkins y Cloud-Build. La primera tiene una alta gama de plugins para integrar con diferentes tecnologías como Github. También utiliza principalmente un archivo que usa lenguaje Groovy que permite un intermedio entre java y código en consola. En segundo lugar se tiene Cloud-Build de Google, que está integrado con las herramientas que ellos ofrecen. Lo que permite crear recursos, asignar usuario y permisos y demás. La facilidad de integración de Cloud-Build con el resto de herramientas hace que sea una buena opción, ya que el manejo de credenciales, usuarios y cuentas de servicio, es una tarea tediosa y con mucha prueba y error involucrados. El problema es que es un recurso de Google, por lo que no estará local y controlado completamente por el desarrollador. En cuanto a Jenkins la forma de hacer pipeline mediante un *script*, que permite invocar comandos de consola y muchas otras cosas, facilita mucho el desarrollar, probar y modificar cada etapa. Esto ya que es local y se tiene acceso directo al recurso. Por esto y por la disponibilidad de un computador para poder ejecutar el servidor con Jenkins, se utiliza esta herramienta para el CD/CI.