



Área Académica de Ingeniería en Computadores

Arquitectura orientada a servicios aplicada a  
sistemas emergentes

Proyecto 1

Anderson Taylor Cordero - 2014045517

19 de abril, 2022

## 1. Componentes y sus responsabilidades

- a. Código fuente: Es donde se define la organización del proyecto, así como sus carpetas, dependencias y demás. Este es cargado a una rama de Github mediante los comandos:

```
git add --all
git commit -m "<indicación del cambio>"
git push origin <rama adecuada>
```

- b. Github: Manejador de código utilizado que contiene el código fuente, su control de versiones y que se conecta a Google Cloud Platform. Aquí se toma un repositorio asignado por la profesora, al cual se le hace *fork* y se trabaja en ramas separadas de la principal para no afectar a toda la organización.
- c. Google cloud platform (GCP): Plataforma de google que permite el uso de herramientas y APIs virtuales. Para acceder a esta se hace una cuenta y se habilitan las opciones de facturación, así como las APIs que se utilizan en el proyecto.
- d. Github branches: Separación lógica del código, en este caso se tiene la rama dev y prod además de la principal. Esta va guardando los cambios que posteriormente se añaden a la rama principal.
- e. Cloud build: Herramienta de GCP con la cual se puede configurar el pipeline entre github y GCP para automatizar el código y sus ejecuciones.
- f. Virtual private cloud: Sección virtual y privada que brinda mayor seguridad a las funciones creadas. Aquí es donde se asegura el código creado y las funciones que realiza en un ambiente virtual.
- g. Cloud bucket: Componente de GCP que guarda los archivos que se usarán en el programa.
- h. Cloud function: Herramienta que permite la revisión del código y llamadas a la Vision API.
- i. Vision API: Es un API de google que mediante inteligencia artificial permite detectar objetos, rostros, etc.

## 2. Conectores entre cada componente

- Código fuente y Github: Son conectados por medio de ramas de trabajo o *branches* por medio de un comando "push" a la rama respectiva.
- Github y Google Cloud Platform: Mediante la configuración adecuada de GCP, las ramas de Github son conectadas a una instancia de Cloud Build (usado como pipeline) donde a través de terraform init, plan y apply los datos son revisados y se cargan en esta instancia. Acá se crea una red privada donde se almacenan los datos de forma segura con su respectivo compute engine. Acá también se define el Cloud Bucket que será utilizado para almacenar la imagen de entrada en la Vision API.
- Código fuente y Cloud Function: Dentro de la configuración se incluyen los archivos correspondientes para conectar el código con la Vision API y Cloud Function, por medio de terraform init, plan y apply. Se lee desde el Cloud Bucket las imágenes ingresadas y se imprime en consola lo que se detecta en ellas, en este caso los sentimientos apreciables.

## 3. Diagrama de la arquitectura de la infraestructura

A continuación se muestra el diagrama utilizado a grandes rasgos en la Figura 1:

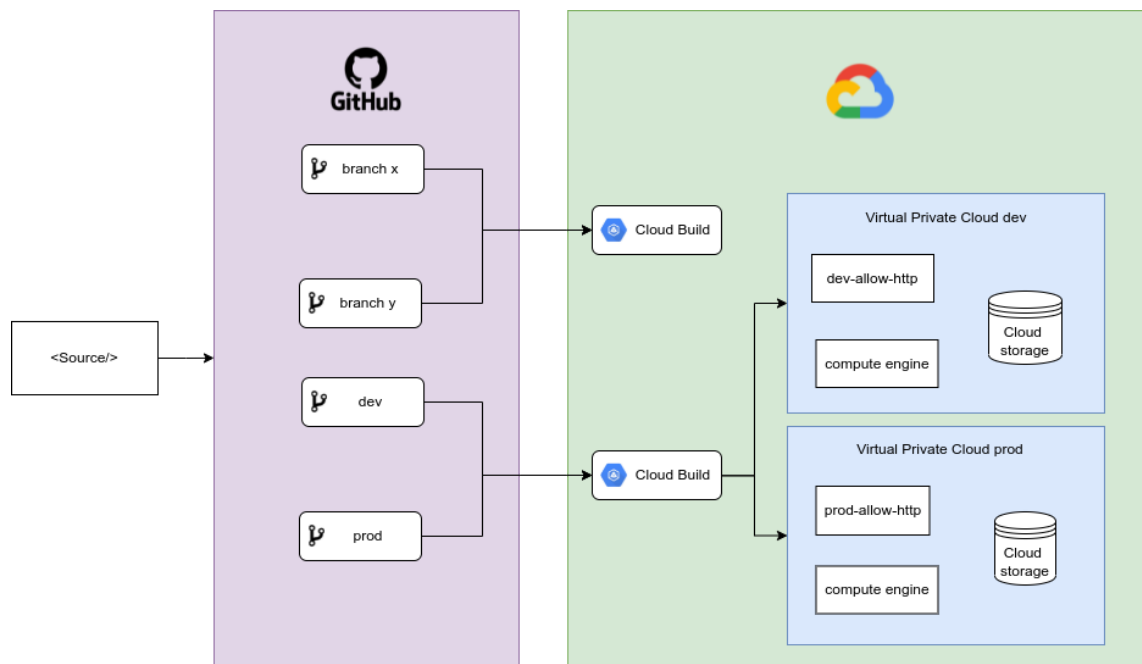


Figura 1. Diagrama de arquitectura

**4. Documente las decisiones de tecnologías específicas, productos, lenguajes y cualquier otra cosa relevante para cada componente.**

a. Tecnologías utilizadas:

- i. Github: Como manejador de versiones para las diferentes etapas del proyecto. Se define un branch strategy para manejar las versiones.
- ii. Cloud build: Como pipeline entre el código fuente, github y Google Cloud Platform.
- iii. Cloud function: Elemento serverless donde se ejecuta el código fuente.
- iv. Google Cloud Bucket: Elemento para almacenar la imagen utilizada en el procesamiento de la Google API Vision.
- v. Google API Vision: API utilizada para dar solución al problema planteado. Contiene algoritmos que permiten identificar sentimientos en el rostro de una imagen.
- vi. Terraform: Software de IaC con el cual se configura la estructura del proyecto y enlaces entre los componentes.
- vii. Draw.io: Herramienta para generar el diagrama de arquitectura de forma sencilla.

b. Productos: Se podría considerar las ramas de github como un producto ya que son el resultado de las actualizaciones que se hacen al código. Además de la respuesta del cloud function respecto al Vision API cuando se ejecuta el programa. Esto último también se aprecia de forma local.

c. Lenguajes: Se utiliza Python como el lenguaje que permite la interacción con el API Vision de google cloud. Además de su facilidad de uso, Python es un lenguaje poderoso que permite una gran variedad de funciones y enlaces a servicios, productos y bibliotecas que hicieron posible la implementación requerida. El formato JSON no es un lenguaje como tal pero es el utilizado para el intercambio de datos entre GCP y el código fuente.