



**Instituto Tecnológico de Costa Rica**

**Área Académica de Ingeniería en Computadores**

**Programa de Licenciatura en Ingeniería en Computadores**

**soa41d: Arquitectura Orientada a Servicios Aplicada a Sistemas  
Emergentes - GR 1**

**Proyecto 1 - SENTIMENT ANALYSIS INFRASTRUCTURE**

**Realizado por:**

**Daniel Sing - 2017124999**

**Adrián López - 2016108981**

**Profesor:**

**Alejandra Bolaños Murillo**

**Septiembre 17, 2022**

**II Semestre 2022**

# Documentación de Componentes y Conectores

## CI/CD Pipeline

Esta funcionalidad del proyecto se encarga de ejecutar de manera automática las funciones descritas en el enunciado (Code check, ejecución de pruebas unitarias, plan de Terraform y ejecución de Terraform apply). En otras palabras, este componente permite seguir una serie de pasos que logran validar y desplegar el proyecto de manera automática.

## Cloud Build

Esta herramienta lee de un archivo (cloudbuild.yaml) para ejecutar los pasos necesarios, descritos anteriormente. Además, se logra tener un historial de los eventos que ejecutan el pipeline en este caso el *push* en el repositorio de GitHub.

## Cloud Function

Con ayuda de *Terraform* se realiza un despliegue de la función contenida en el repositorio, al finalizar el proceso de validación del Pipeline este por último corre el comando *Apply* el cual directamente interactúa con esta herramienta para poder ejecutar el código por medio de este API de manera remota.

## Vision API

API de Google que cuenta con una inteligencia artificial de reconocimiento de patrones en las expresiones del rostro humano para la detección de sentimientos. Esta se accede por medio del código de python importando una biblioteca, la imagen se carga desde el *bucket* de Cloud Storage, se procede a analizarla y devuelve un resultado de los sentimientos encontrados.

## Diagrama de la arquitectura de la infraestructura

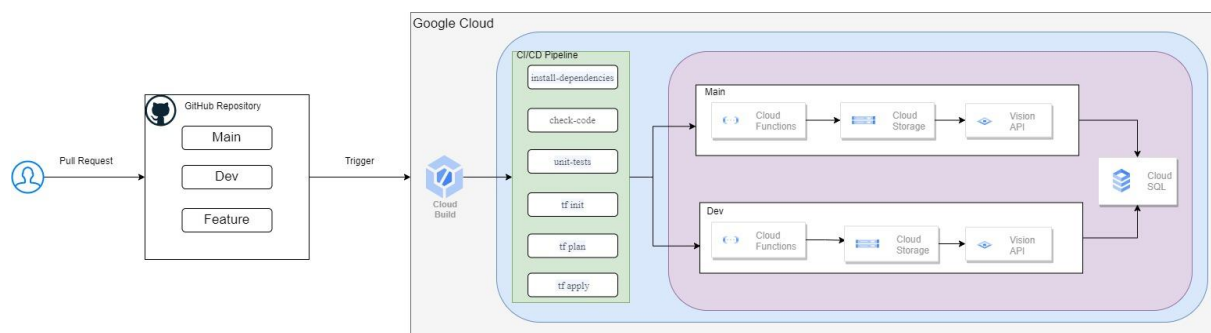


Figura 1. Diagrama de Arquitectura

# Set de tecnologías

## Google Cloud

Se trata de una nube pública que ofrece servicios que fueron utilizados en el presente proyecto como Cloud Build y Cloud Function, a sus clientes de manera que estos solamente pagan por el consumo del procesamiento utilizado correspondiendo a un proveedor de nube pública. Esta herramienta ofrece muchas más funcionalidades y su uso es muy sencillo. En la siguiente imagen se muestra el proyecto en cuestión que fue configurado según las necesidades descritas en el enunciado.

## Terraform

Es una forma de crear y definir la arquitectura de Cloud por medio de comandos, este tipo de infraestructura es *Infrastructure as Code (IaC)*. Se utiliza Terraform para hacer el deployment de la función de python en el Cloud Function y los buckets necesarios para resolver el problema descrito. Además forma parte del pipeline (init, plan y apply) por lo que se automatiza el proceso.

## GitHub

GitHub se trata de un controlador de versiones de código, en este se creó el repositorio que además de albergar el código fuente también nos ayuda a ejecutar el pipeline por medio de su característica de *Push* a una rama.

## Python

Lenguaje de programación utilizado en el proyecto para la generación del script que soluciona el problema de la empresa.

## Pylint

Herramienta de verificación de código de Python. Analiza la sintaxis del código y despliega un resultado para que el programador tenga una calificación del código que escribió, se utiliza más que todo para mantener buenas prácticas de programación.

## Pruebas unitarias

Las pruebas unitarias corresponden a pruebas realizadas al código a los diferentes módulos de software lo que permite detectar errores en los datos, lógica o en los algoritmos ya que se puede comprobar que la salida sea la deseada. En el caso del proyecto, se utiliza pytest ya que ofrece diversas herramientas para realizar dichas pruebas en el entorno de python. Si las pruebas pasan se continúa con el despliegue de la función.

# Documentación de decisiones

Primero se pretendía realizar una implementación del pipeline con Jenkins, sin embargo, después de una investigación y varias recomendaciones de la profesora del curso, se optó por utilizar Cloud Build para la ejecución del pipeline.

Para la base de datos se seleccionó MySQL debido a que se puede integrar una instancia dentro de Google Cloud lo que permite generar tablas que puedan almacenar fácilmente los datos obtenidos. Además se utiliza GitHub como controlador de versiones debido a que permite el manejo de los controles de versiones y almacena el historial de los cambios realizados.

Se realizó el programa en el lenguaje de programación python debido a la facilidad de comunicación con el API de Vision y las bibliotecas de pytest y pylint para pruebas unitarias y chequeo de código respectivamente.