Logotipo, nombre de la empresa

Descripción generada automáticamente

Instituto Tecnológico de Costa Rica

Arquitectura Orientada A Servicios Aplicada A Sistemas Emergentes

**Primer Semestre Año 2022**

**Profesor:**

Alejandra Bolaños Murillo

**Estudiantes:**

Adrián González Jiménez – 2019173563

Componentes y sus responsabilidades:

1. **Cloud-Bucket:** Este componente va a estar encargado de servir para almacenamiento y de accionar un trigger en caso de que se suba una imagen, dicho trigger llama a la cloud function que evoca el proceso de análisis.
2. **Componente Serverless de Cloud function:** Este componente corresponde al script de Python que invoca a la REST de Vision y opera su resultado, trabaja realizando la consulta al bucket .
3. **RestApi de Visión:** Esta api será la encargada de recibir una imagen y retornar el análisis solicitado, en este caso, corresponde al análisis de detección de rostros, que contiene información sobre los sentimientos de la imagen.
4. **Repositorio de GitHub:** Este componente se encargará de respaldar versiones del código, pero también de ejecutar acciones que permitan facilitar el CI Y CD, tales acciones son : la ejecución de un lint, ejecución de pruebas unitarias, ejecución de plan de terraform y aprobación manual (estos dos últimos en caso de un push request)

Diagrama del sistema  
Diagrama

Descripción generada automáticamente

Conectores entre cada componente

1. Bucket-CloudFunction: Cada vez que subamos una imagen ejecutaremos un trigger que se encargará de llamar a la CloudFuntion.
2. CloudFunction-Vision: Cada vez que se llame al proceso la cloud function le enviará dicha fotografía a dicha api, que nos retorna un array de información a cerca de las caras en la foto.
3. Github-CloudFunction: Cada vez que se ejecute un push request, el pipeline en GitHub se va a encargar de realizar una serie de pasos, entre ellos el deploy a Cloud.

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

**Python:** En este caso, es un requerimiento solicitado, pero también hay que destacar que para las acciones que vamos a realizar, tiene bastante soporte y una comunidad bastante extensa, lo que nos permite establecer que Python es bastante confiable para el objetivo de desarrollar una arquitectura escalable y mantenible.

**Google-Cloud:** Se va a utilizar este proveedor de servicios, ya que tiene un soporte y comunidad bastante extenso, Además que tiene una documentación bastante clara y que de hecho nos guía en una posible implementación de Cloud Vision.

Para este proyecto vamos a utilizar la estrategia de Git Flow, primeramente, porque es bastante sencillo de entender y segundo que debido a su estructuración nos permite trabajar de manera descentralizada entre equipos, pero manteniendo un grado esencial de centralización a la hora de trabajar en el proyecto.