**Text

Description automatically generated with medium confidence**

**Arquitectura Orientada a Servicios aplicada a**

**sistemas emergentes**

**CE-5508**

**Grupo 01**

**Proyecto 1**

**Image Analyzer Infrastructure**

**Documentación**

**Prof. Alejandra Bolaños Murillo**

**Alumno:**

**Mario Araya Chacón - 2018319178**

**Fabián Ramírez Arrieta - 2018099536**

**Mariana Vargas Ramírez - 2018086985**

**II Semestre 2022**

# Índice

# 

[Índice](#_q0551kn12qlv) **2**

[Componentes y conectores](#_4cqw5f69kzg) **3**

[Diagrama de arquitectura](#_fty8ei4qdua3) **4**

[Set de tecnologías](#_5zf74rocmp6h) **5**

[GitHub](#_fltqk55jewkg) 5

[Python](#_6t5skkqk5qba) 5

[Google Cloud](#_lsnibxkjd68q) 5

[Jenkins](#_m4wli24ki0uv) 5

[GCP Vision API](#_d5eyh71f4ya6) 5

[GCP Cloud Firestore](#_wjpdm6dmxi5l) 5

[Bibliografía](#_ap8o76kgzp7r) **7**

# 

# Componentes y conectores

# Diagrama de arquitectura

# Set de tecnologías

A continuación se procederá a enumerar cada una de las tecnologías utilizadas y cuál es su función dentro de la solución implementada, además del porqué se decidió esta alternativa sobre las otras opciones que hay en el mercado.

## GitHub

Utilizado como controlador de versiones para albergar el código. Es la opción más popular del mercado y con la que el equipo tiene mayor familiaridad, por estos motivos se optó por utilizar este producto. Además de poseer una muy buena integración con las herramientas que se verán a continuación.

## Python

Se eligió Python como lenguaje de alto nivel para implementar la lógica del proyecto. Es uno de los lenguajes más populares a nivel mundial con una comunidad de desarrolladores sumamente extensa. Esto hace que su utilización sea más sencilla debido a la vasta documentación disponible. Además de la simpleza en su sintaxis.

## Google Cloud

Herramienta para hospedar los distintos componentes de la aplicación y poder almacenarlos en la nube. Se eligió esta herramienta ya que al pertenecer a uno de los gigantes de la industria tal y como es Google hace que sea un producto sumamente pulido y de alta fiabilidad. Además que la documentación proveída por el fabricante es de gran calidad, lo que permite una guía de calidad durante el desarrollo de la aplicación.

## Jenkins

Se decidió por usar Jenkins por encima de Cloud Build debido a su mayor popularidad. Se hizo más sencillo implementar el Pipeline con esta herramienta ya que al ser la herramienta líder en su sector, se encontraron gran cantidad de ayudas en la web. Jenkins posee muchos más usuarios en comparación a Cloud Build. [1]

## GCP Vision API

Se eligió utilizar esta API ya que, similar a la herramienta anterior, al pertenecer al gigante de Google es una excelente alternativa para lograr el objetivo de la aplicación, la cual es reconocer las emociones de personas en las imágenes que se le provean. Esta inteligencia artificial, la cual funciona por algoritmos de Machine learning, entre mejores sean los datos, mejores serán los resultados obtenidos y al ser esta API propiedad de Google se puede dar por un hecho que el conjunto de datos con el cual alimentan a la inteligencia artificial está sumamente depurado.

## GCP Cloud Firestore

Finalmente, como herramienta para almacenar información, especialmente las imágenes procesadas y el análisis posterior se eligió Cloud Firestore. Nuevamente, esta herramienta pertenece al kit de herramientas creadas por Google, por lo cuál luego de haber seleccionado varias herramientas de esta compañía, el hecho de seguir dentro del mismo ecosistema hará que el desarrollo sea más uniforme.

# Bibliografía

[1] Google Cloud Build vs Jenkins. (2022). Online. Available on: <https://stackshare.io/stackups/google-cloud-build-vs-jenkins>