**Psico-engine**

**Introducción**

El presente documento presenta la arquitectura de psico-engine y como está desarrollado, presenta las herramientas y utilidades necesarias para su funcionamiento, también presenta el paso a paso que se debe seguir para su despliegue en un ambiente de desarrollo.

**Google Cloud Platform - API Vision**

Vision API proporciona una interfaz RESTful que facilita la tarea de tener que desarrollar algoritmos de procesamiento de imágenes. Vision API soporta formatos de archivo de imagen como JPEG, BMP, RAW, PNG. Para un análisis lo más preciso posible, la resolución de imagen mínima recomendada es VGA (640 × 480 píxeles), y el tamaño de los ficheros no debe superar los 4 MB.

Mediante peticiones a la API para que analice una imagen concreta, podemos extraer diferentes tipos de características explicadas a continuación.

La característica Face Detection tiene como objetivo detectar caras humanas en imágenes. Se debe tener en cuenta que no es lo mismo detección de caras y reconocimiento de caras. El reconocimiento de caras intenta hacer coincidir información biométrica asociada a una cara detectada con información biométrica de caras almacenadas y etiquetadas.

Esta característica funciona principalmente con la detección de caras vistas frontalmente. Además, es importante la distancia en píxeles entre pupilas para obtener una detección precisa. En ese sentido, la API ofrece buenos resultados cuando las distancia en píxeles entre pupilas es de un mínimo de 32 pixeles. Además, información como las coordenadas de la posición de la cara o caras detectadas, puntos de referencia (ojos, pupilas, nariz, boca, labios y otros), orientación de la cara (ángulos roll, pan y tilt) y emociones (sorpresa, enojo, alegría y otros) son algunas de las características que podemos obtener.

Landmark detection gira en torno a la identificación de estructuras populares ya sean naturales o construidas dentro de una imagen. La API nos devolverá la longitud y la latitud del punto de referencia identificado. Logo detection intenta identificar logotipos de productos dentro de una imagen. Así, la API de Google devuelve el logotipo de la marca del producto identificado y sus coordenadas dentro de la imagen.

Label detection agrega semántica a las imágenes de un conjunto de categorías predefinidas de objetos. La API devuelve una estructura JSON con todas las etiquetas encontradas y su tanto por ciento de acierto. Safe search detection intenta detectar contenido inapropiado en imágenes usando un modelo de red neuronal “deep learning” previamente entrenada.

Text detection extrae texto de una imagen mediante reconocimiento óptico de caracteres (OCR, del inglés Optical Character Recognition) siendo compatible con varios idiomas. Los algoritmos de OCR combinados con reglas semánticas son utilizados para el reconocimiento de matrículas, y de este modo, la API de Google Vision devuelve una cadena de texto y sus coordenadas.

Document text detection aplica OCR(Reconocimiento optico de caracteres) a imágenes con mucho texto, como documentos.

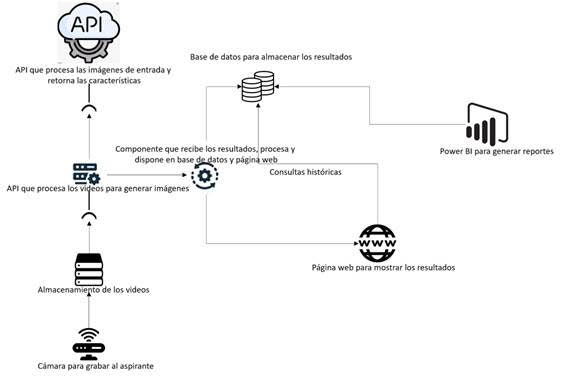
Image properties calcula un conjunto de propiedades sobre la imagen como, por ejemplo, sus colores predominantes.

Web detection detecta entidades temáticas en la imagen (como noticias, eventos o famosos) y busca imágenes similares en Internet mediante la tecnología de la Búsqueda de imágenes de Google.

Crop hints determina los vértices sugeridos de una región de recorte de la imagen.

Object localization detecta y extrae varios objetos de una imagen.

**Psico-engine - Arquitectura**

****

Psico-engine es una API Rest creada para realizar análisis de sentimiento consumiendo un servicio de Google Cloud Platform denominada API Vision, presentada anteriormente.

Para el correcto funcionamiento de Psico-engine se requiere la generación de una api key en la consola de Google Cloud Platform, para ello se deben seguir los pasos que se describen en el [enlace](https://cloud.google.com/vision/docs/before-you-begin?hl=es-419), una vez obtenida la api key se debe activar la facturación, esto permitirá el consumo de la api.

Una vez se activa la facturación, se debe generar el archivo .JSON que contiene los secretos del cliente, este también se puede obtener de la consola de Google, el archivo contiene las variables de entorno para el consumo de la API.

**Implementación**

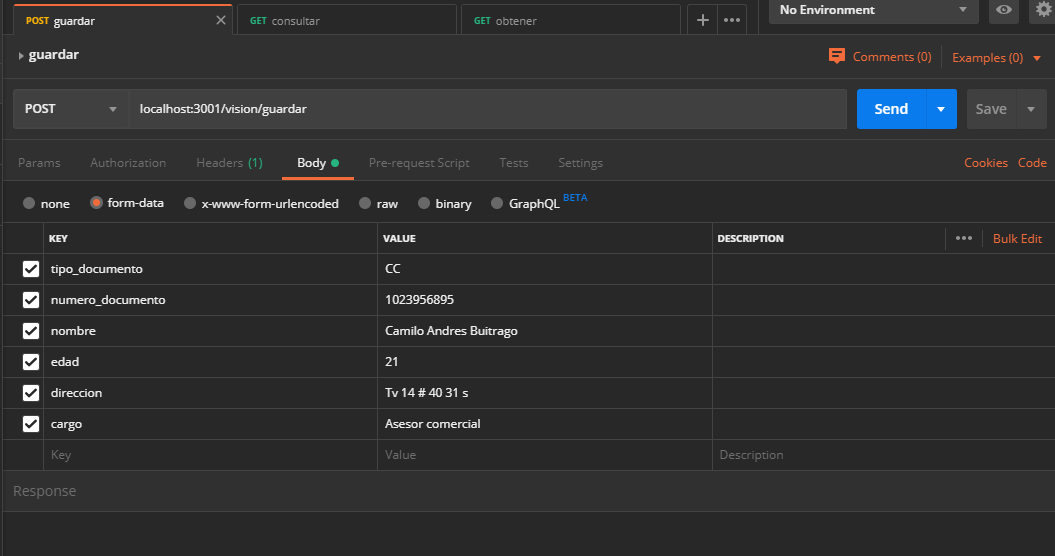
Las tecnologías utilizadas para el desarrollo de psico-engine son:

* Python3.6
* MySQL5.7
* Google Cloud Platform - API Vision

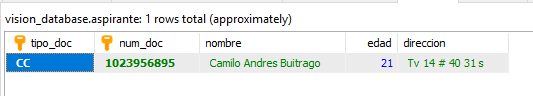
Psico-engine es una una interfaz RESTful compuesta de dos endpoint, el endpoint /vision/guardar dispone dos métodos, el método POST es un formulario que solicita los datos básicos de un aspirante, datos como:

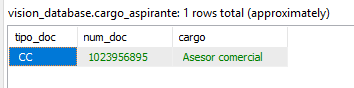
* Nombre completo
* Tipo de documento
* Número de documento
* Edad
* Dirección
* Cargo

Estos datos deben ser enviados en el body del request así:

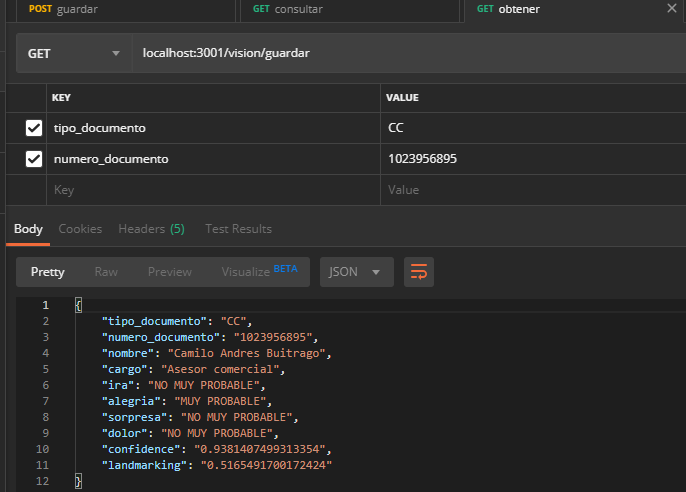


Una vez ingresados estos datos, son almacenados en la base de datos así:

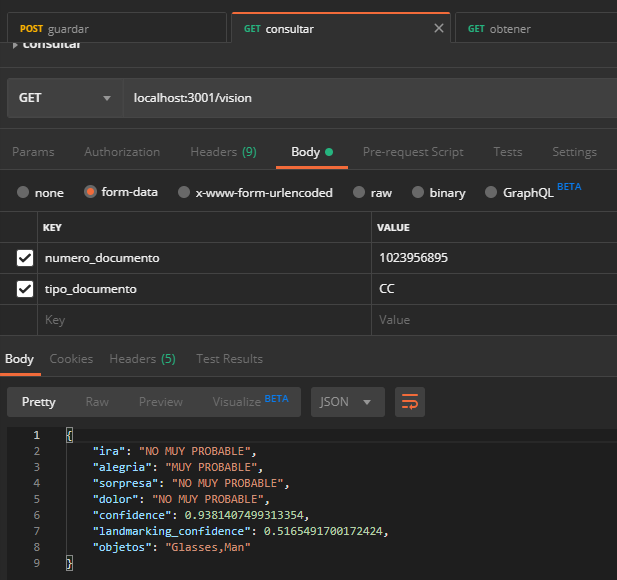




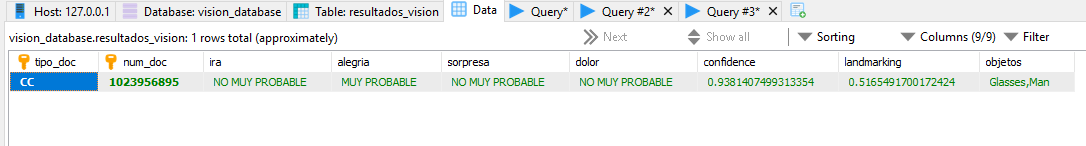
El método GET permite consultar los resultados de la API relacionados con el aspirante, los parámetros para ejecutar el método son: tipo de documento y número de documento, los resultados se visualizan así:



El endpoint /vision dispone el método GET que consume la API de Google, para ello es necesario enviar los parámetros tipo de documento, número de documento en el body del request, los resultados se verán así:



Los resultados también se almacenan en base de datos para posteriores consultas:



Es importante resaltar que una vez almacenados los datos con el método POST se debe ejecutar el método GET del siguiente endpoint ya que este es el que consume la API de Google.

También es necesario aclarar que se debe disponer mínimo una foto para ser analizada por la API, el objetivo principal de la herramienta es implementar un frontend que habilite la cámara del aspirante y mientras se lleva a cabo la entrevista tome fotos entre determinados intervalos de tiempo, sin embargo por temas de tiempo no se pudo implementar un frontend.

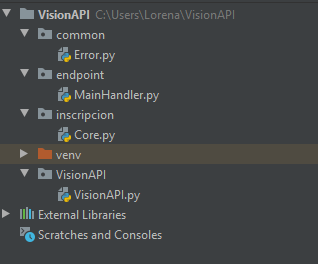
**Despliegue en un ambiente de desarrollo**

Para el correcto funcionamiento de psico-engine se requieren las siguientes librerias de python:

* Tornado Web Server
* MySQL Connector
* Google CloudI
* Time
* io
* os

Se debe contar con el archivo json generado de la consola de Google que contiene los secretos del cliente, este archivo contiene las variables de entorno para el consumo de la api de google. Se debe contar con la api key de Google.

La arquitectura del proyecto es la siguiente:



En common, deben quedar aquellos componentes que son comunes para todas las funcionalidades del proyecto, por el momento solo contiene los códigos de error.

En inscripcion, se definió la clase Core que contiene toda la lógica de negocio.

En VisionAPI, se definió el componente que consume la API de Google.

En endpoint, deben hospedarse todos los endpoint que se deseen implementar.

Anexo a este documento se envía el código fuente junto con el dump de la base de datos.

**Conclusiones**

Implementar una solución de inteligencia artificial puede ser sencillo por todas las facilidades que se dan en las nubes disponibles.

En caso de querer entrenar un modelo de forma autónoma, también es muy viable por todas las utilidades opensource disponibles, anexo al documento se envía un archivo .py que contiene un modelo de análisis de sentimiento implementación in-house, para ejecutar el modelo y entrenarlo se requiere una m´quina con muy buenas características, el proceso de entrenamiento puede tardar bastante tiempo pero el resultado será muy positivo.

Esto no es un documento oficial, solo quiere presentar algunas de los servicios creados para servicios de inteligencia artificial.