

# UNIVERSIDAD PANAMERICANA

Organizador de fotografías inteligente

Asignatura: Robótica industrial

Impartida por: Dr. Michel Emanuel Lopez Franco

Sebastián José Arrobo Moncayo, ID: 0198856.

Emilio Gerardo González Vázquez del Mercado, ID: 0197246.

25 de mayo de 2020

## Descripción del proyecto

El proyecto que desarrollamos se eligió buscando dar solución a la tediosa tarea de organizar las fotografías familiares, de amigos y personales. Lo que nos llevó a desarrollar una herramienta capaz de hacer reconocimiento facial para después clasificar la fotografía en tres diferentes carpetas: familia, amigos y novia.

Se eligieron estas tres categorías por motivos prácticos, pero realmente se pueden seleccionar las categorías que se quieran, siempre y cuando se organicen bien las fotografías de los rostros que se analizaran. De esto se hablará más a detalle adelante. Nuestro proyecto se realizó con código Python utilizando la librería de `face_recognition` y almacenando las fotografías en google Drive.

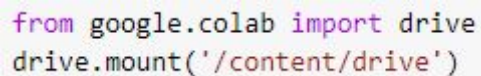
## Requerimientos preliminares

Para poder cumplir con nuestro objetivo tenemos que descargar y hacer uso de la librería `face_recognition` de Python. De la misma manera, como haremos uso de google Drive, lo importamos de las librerías de google Colab, donde desarrollamos nuestro proyecto.



```
!pip install face_recognition
```

*Imagen 1 - Instalación de librería face\_recognition*



```
from google.colab import drive  
drive.mount('/content/drive')
```

*Imagen 2 - Importando google Drive*

Teniendo ya instalados o importados los requerimientos preliminares solamente disponemos de ajustar la dirección de Drive en donde se encontrará nuestro espacio de trabajo.

## Desarrollo

Como se mencionó anteriormente es muy importante organizar nuestros archivos donde se encuentran los rostros que analizaremos, ya que nuestra lógica se basa en administrar y acomodar las imágenes dependiendo de los rostros que se identifiquen. Esto es, en la carpeta de Familia colocaremos únicamente rostros que estén relacionados con los miembros de

nuestra familia y así respectivamente para los casos de Amigos y Novia. Es por esto que en una carpeta independiente llamada Caras se suben en carpetas separadas los rostros de los integrantes de Familia, Amigos y Novia.

Una vez realizado esto le indicamos al programa qué rostros son respectivos a cada grupo a clasificar. Es decir, en la carpeta Caras nos dirigimos a la carpeta Familia y realizamos un `face_encoding` para cada rostro que se encuentre en esta carpeta, que no es más que identificar el rostro en cada imagen y codificarlo para procesarlo. Al realizar esto nosotros ya le indicamos al programa que rostros son de cada grupo clasificatorio añadiendo cara `face_encoding` a una lista dependiendo el grupo.

Es importante mencionar que para lo previamente descrito funcione de manera correcta es necesario subir a cada carpeta de Caras imágenes en donde solo se muestre de forma clara los rostros de las personas que pertenecen a este grupo. La imagen a continuación muestra un ejemplo de cómo deben de ser estas fotografías.



*Imagen 3 - Ejemplo de imagen de rostro a analizar*

Una vez ya realizados los `face_encoding` de cada rostro y al ser organizados en listas clasificatorias ya somos capaces de subir una fotografía cualquier y analizar los rostros presentes para redirigirla a una carpeta en específico. Para explicar nuestra lógica utilizaré la siguiente fotografía.



*Imagen 4 - Fotografía a analizar (muestra)*

En esta fotografía nosotros podemos observar que se encuentran dos rostros fácilmente identificables y podemos deducir que son nuestros amigos, por lo cual queremos que automáticamente se coloque este archivo en la carpeta de Amigos. Para lograr esto nosotros subimos el archivo y realizamos un `face_encoding` a la imagen, lo que da por resultado dos rostros ya codificados.

Al tener ya identificados estos dos rostros procedemos a comparar si estos rostros pertenecen a los rostros que previamente ya habíamos clasificado. Esto lo realizamos comparando de manera individual cada rostro en la nueva imagen con la lista de rostros codificados de cada grupo, si se encuentra alguna coincidencia aumenta en uno un contador indicador a qué grupo pertenece, y así para cada rostro identificado en la nueva imagen.

Una vez se hayan hecho todas las comparaciones nosotros creamos y organizamos un diccionario con la cantidad de coincidencias encontradas pertenecientes a cada grupo organizado de forma descendente. En nuestro algoritmo nosotros indicamos que el grupo con mayor coincidencias encontradas será el grupo al que se direccionará la nueva imagen, por lo que se analiza siempre la primera posición de nuestro diccionario ordenado de forma descendente y esta nos indica hacia dónde se irá nuestra imagen. Una vez hecho esto simplemente se cambia la dirección donde se encuentra guardada la imagen a la dirección de la carpeta a la que pertenece.

## Conclusiones

Con la librería de `face recognition` de Python nos dimos cuenta que es relativamente sencillo realizar análisis de imágenes para reconocer los rostros presentes y de ahí realizar lo que nosotros queramos. Con la herramienta desarrollada nos dimos cuenta de la alta capacidad que tiene este código y la facilidad con la que se manipulan las codificaciones de los rostros a los intereses que nosotros plasmemos en nuestro código.

# Bibliografía

<https://www.w3schools.com/python/default.asp>

<https://pypi.org/project/face-recognition/>