# Proyecto 01

Carlos Cabrera Ramírez Sergio Medina Guzmán

Marzo 2022

## Descripción del trabajo.

Uso del programa:
Sistema Operativo: Linux - Mac OS / OS X - Windows Versión de Python: Python 3.6+ Se debe asegurar que pip3 esté actualizado.
Instalar pytest, requests y python-dotenv:
$ pip3 install pytest
$ pip3 install python-dotenv - Para usar variables de entorno definidas en .env
$ pip3 install requests
Situarse en el directorio src/app y agregar la llave API de OpenWeather al archivo .env. Con esto y los módulos instalados ejecutar el comando:
$ python3 weather_app.py

## Esbozo del proceso de solución

Al pensar y analizar el proyecto, nos preguntamos qué es lo que queremos obtener. El pdf del proyecto nos indica que debemos obtener el clima en ciudades de origen y destino para tickets de vuelos entre aeropuertos. Los datos que tenemos para obtener dicha información son los códigos IATA de los aeropuertos junto con las latitudes y longitudes de cada uno de ellos. Dicha información se encuentra en el archivo dataset1.csv. Observamos que con dicha información y con acceso a OpenWeather, es realmente toda la información necesaria. Por medio de la API de clima actual que provee OpenWeather podemos tener acceso al clima en tiempo real pasando como argumentos a la llamada la longitud y latitud de la ciudad por consultar. Se procede entonces a hacer pruebas de que la petición funciona correctamente, por medio de asserts en Python.

Tenemos que la entrada es un archivo csv que contiene:

1. Latitud del aeropuerto de la ciudad de origen.
2. Latitud del aeropuerto de la ciudad de destino.
3. Longitud del aeropuerto de la ciudad de origen.
4. Longitud del aeropuerto de la ciudad de destino.
5. Código IATA de los aeropuertos de las ciudades de origen y destino.

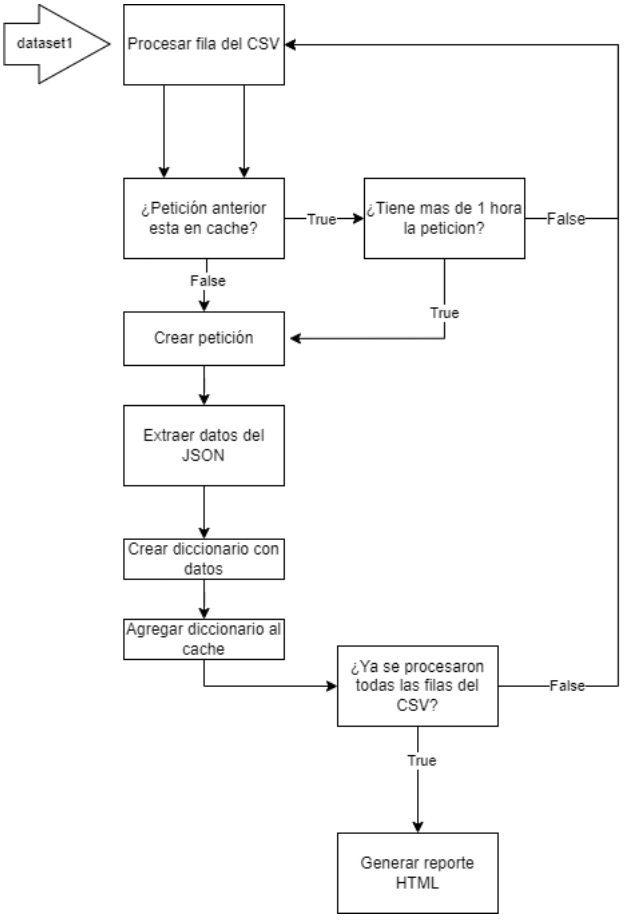
Además, como salida de cada petición tendremos un String que contiene:

1. Código IATA de los aeropuertos.
2. Descripción del clima.
3. Temperatura.
4. Temperatura máxima.
5. Temperatura mínima.

Así, como salida del programa se genera un reporte en formato HTML dentro del cual se tendrá una tabla que contenga la información que proporciona la salida de cada petición para cada aeropuerto.
Cada línea en el csv contendrá dos peticiones. Antes de hacer cualquiera de las peticiones al API de OpenWeather, se revisa si la petición ya está en el cache. Si no se encuentra en el caché, se hacen las peticiones al API y entonces se maneja el json que otorga el API como respuesta.

Pruebas. Mediante pytest:
1. Con test_cache.py se simula información falsa para verificar el funcionamiento del cache.
2. Con test_report.py se simula el generar una fila del reporte.
3. Con test_request.py se verifica que la petición al API sea correcta.

## Diagrama de flujo




## Pensando a futuro

En un futuro podría ser necesaria la consulta de tickets que constantemente se van emitiendo, entonces podría tener que trabajarse en una entrada que sea un flujo constante de datos y no sólo una hoja con datos fijos. Igualmente la implementación de una interfaz gráfica que permita buscar información específica de un ticket o permitir al usuario mediante un mapa seleccionar origen y destino de de los vuelos y que despliegue entonces sobre la selección del mapa la información que originalmente se mandaba al reporte HTML. Por este proyecto pensaríamos cobrar entre $20000 y $25000 pesos, y por los mantenimientos o implementaciones posteriores deade $10000.