

## 1. Definición del problema

Dar a conocer el informe del clima de la ciudad de salida y de la ciudad de llegada de 3 mil tickets de avión.

## 2. Análisis del problema

### 1. ¿Qué es lo que queremos obtener?

Se nos solicita obtener el informe del clima de dos ciudades (de origen y de destino) de cada ticket en una lista de 3 mil tickets. El resultado será un archivo csv con los datos de *descripción*, *temperatura*, *sensación térmica*, *temperatura máxima* y *temperatura mínima*, *lluvia* y *nieve* solo cuando aplique.

### 2. ¿Cuáles son los datos que tenemos para obtenerlo?

Se nos proporciona un archivo csv con los datos de 3 mil tickets. Cada ticket se conforma de los datos de la ciudad de origen y de destino, los cuales son: código IATA, longitud y latitud. Los códigos son de tipo *string*, la longitud y latitud de tipo *float*.

### 3. ¿Son suficientes esos datos? Usando la API de OpenWeatherMap podemos obtener la información del clima en una ciudad usando su código IATA o bien usando su longitud y latitud. Entonces con los datos que tenemos es suficiente.

### 4. ¿Qué hace que el resultado obtenido resuelva el problema? ¿Cuál es la característica que hace de un resultado una solución?

Los datos regresados muestren el clima actual de las ciudades en los tickets proporcionados.

### 5. ¿Qué operaciones o construcciones se deben obtener para llegar a la solución?

- a) Leer el csv
- b) Hacer la petición a OpenWeatherMap por cada ciudad en cada ticket.
- c) Extraemos los datos de interés de la respuesta
- d) Construir una nueva tabla con los datos requeridos
- e) Escribir la tabla en un archivo csv de salida.

## 3. Selección de la mejor alternativa

Se usaron las siguientes bibliotecas: *pandas*, *python-dot* y *requests*.

La biblioteca *pandas* es una biblioteca para manejo de datos en forma de tablas, cuenta con dos estructuras básicas que son los *Series* y los *DataFrame*, el primero es una forma vectorial de almacenar datos, mientras que el segundo puede ser visto más como una forma matricial (o como tablas hablando de bases de datos). Cuenta con funciones para facilitar la lectura y escritura de sus estructuras a archivos de texto.

La biblioteca *python-dot* fue usada para cargar de forma automática las variables de entorno usadas en el proyecto.

La biblioteca *requests* funciona como cliente http, en este proyecto fue usado para hacer las peticiones a la api de OpenWeatherMap de la cuál se obtuvieron los datos de clima. Nos dio la ventaja de que las respuestas de la api, nos las convertía en diccionarios de python.

#### 4. Diagrama de flujo

