

### 1. Definición del problema

Se tiene una lista de vuelos proporcionada por un aeropuerto; necesitamos escribir un programa para mostrar información acerca del clima de los lugares de origen y destino en tiempo real.

### 2. Análisis del problema

Recibiremos un archivo csv con la información ordenada por columnas; además del nombre del lugar del vuelo, también se nos brindan dos coordenadas que representan la altitud y longitud de la localización.

Necesitamos guardar los datos de entrada que nos proporciona el archivo csv en una estructura de datos eficiente para poder hacer consultas a futuro en la misma.

La información requerida para dar la salida solicitada se obtendrá de un medio exterior, lo cual inmediatamente supone que contaremos con conexión a internet.

Obtener la información necesaria involucra interactuar con otros sistemas en línea; ello implica orientar el programa para que intercambie datos con un Web Service.

Finalmente, los datos que se obtengan de los servicios web, deben ser llevados a un formato para su salida en el programa.

### 3. Selección de la mejor alternativa.

Una breve búsqueda en internet ha sugerido que el mejor Web Service para llevar a cabo nuestra tarea, es Open Weather Map.

Mientras que la estructura que nos brindará eficiencia para guardar y consultar posteriormente datos, serán los diccionarios.

Se usará Python para escribir el código correspondiente, ya que el lenguaje cuenta con varios módulos útiles para lo que nos atañe.

### 4. Pseudocódigo

Abrir archivo csv.

Desechar el primer renglón.

Mientras haya renglones, tomar la primera columna, usarla como llave para el diccionario y tomar como valor la tupla que contenga la latitud y longitud del lugar (tercera y cuarta columna). Hacer lo mismo para la segunda, quinta y sexta columna.

Preparar la API key.

Hacer una petición al Web Service proporcionando la API key, latitud y la longitud de la localización en cuestión.

Con la información recibida del Open Weather Map, se extrae la humedad, nubosidad, temperatura, tiempo atmosférico, y se genera una cadena.

Si la cadena contiene algún número que represente alguna falla en la consulta, se cierra el programa. De lo contrario, se despliega la información.

*Explicación*

El pseudocódigo trata de mostrar la apertura de la base de datos proporcionada por el aeropuerto donde se encuentran los vuelos, después la distribución de estos en un diccionario que nos permitirá hacer consultas a este conjunto de información de una manera muy eficiente. Después se intentan ilustrar las peticiones al Web Service dando datos recabados del archivo csv; finalmente se le da un formato legible al producto considerando los datos más importantes y desplegándolos en pantalla si no presentan ningún mensaje

de error; se cierra el sistema en caso contrario.

*Nota:* la primera línea se desecha porque contiene el encabezado de los datos.

## 5. **Mantenimiento y costo**

Considero que el programa necesitará en un futuro ser más general y poder procesar varios archivos, contemplar otros casos de error como la falta de conexión a internet, y usar multihilos para lidiar de una manera más eficiente con los varios conjuntos de datos. Tomando en cuenta el tiempo que me llevé escribiendo el programa, así como el usado para investigar ciertos aspectos técnicos que se hacían presentes en la solución del problema; cobraría 1500 pesos por la primera entrega y por los futuros mantenimientos pediría 600 pesos si estos llegasen a ser triviales o si requirieran poca inversión de tiempo.