



# Universidad Nacional Autónoma de México

FACULTAD DE CIENCIAS

Modelado y Programación

Proyecto 01

## **Alumnos:**

Main Cerezo Asahel Said

Reyes López Eduardo Alfonso

# Explicación del proyecto

## 1. Definición del problema

Se nos solicita entregar el informe del clima de la ciudad de origen y de llegada para tres mil tickets que salen el mismo día en el que corremos nuestro algoritmo

### a) Entender el problema:

- ¿Qué queremos obtener?

La información del clima para las ciudades de origen y llegada de los tres mil tickets que se nos proporcionan

- ¿Cuáles son los datos que tenemos para obtenerlo?

Contamos con un archivo csv en el cual se encuentran:

- Código IATA de cada ciudad de origen y de destino
  - Latitud y longitud de cada ciudad de origen y de destino
- ¿Qué hace que el resultado obtenido resuelva el problema?, ¿cuál es la característica que hace de un resultado, una solución?

El resultado esperado por el programa es la información del clima de cada ciudad de origen y destino en el momento en el que se corre el algoritmo, que es justamente lo que se nos solicita.

- ¿Qué operaciones o construcciones se deben obtener para llegar a la solución?

a) Comunicarnos con la base de datos (csv)

- b)* Extraer los datos
- c)* Consultar el caché. Si existe, mostrarlo, si no:
- d)* Crear la petición
- e)* Hacer el request
- f)* Procesar la salida
- g)* Guardar el caché. Mostrar o recuperar la información

**b) Requisitos funcionales:**

La información de cada ticket debe ser almacenada en una estructura de datos Para cada ciudad se debe hacer un request para consultar la información actual del clima. Se tiene que hacer un caché para evitar hacer requests innecesarias. La información del clima se guardará en un diccionario para posteriormente poder mostrarla al usuario.

Entradas:

- Tipo: matriz
- Formato: csv
- Tamaño: 3000
- Cantidad: 1

Salidas:

- Tipo: diccionario
- Tamaño: 3000

**c) Requisitos no funcionales**

- Eficiencia
- Tolerancia a fallas
- Amigable con el usuario
- Seguridad
- Escalabilidad

**d) Arsenal**

- Paradigma de programación: Orientado a objetos
- Herramientas: WebService - OpenWeather. Para consultar el clima
- Lenguaje: Utilizaremos Python debido a las diversas bibliotecas con las que cuenta para poder procesar los datos
- Bibliotecas: request, csv, tkinter

## **2. Análisis del problema**

Queremos obtener la información del clima de las ciudades que se nos solicitan en cada uno de los tickets, para ello tenemos que ocupar algún web service que proporcione información climática. Ejemplos de lo anterior son OpenWeather y Yahoo Weather.

Primero tenemos que hacer una lectura del archivo csv proporcionado, el cual contiene todos los tickets de los cuales queremos conocer información climática de su ciudad de origen y destino. La información del csv la guardamos en una estructura de datos para que sea más fácil de manipular posteriormente

Una vez leído el archivo csv surge un problema con el web service elegido, y es que la versión gratuita de estos servicios tiene un número limitado de llamadas a la API. Este número es menor a la cantidad de tickets a los que queremos obtener el clima de la ciudad de origen y destino. Para solucionar esto podemos usar un caché que guarde la información de las ciudades de las cuales ya se conoce el clima.

Finalmente tenemos que mostrar de alguna manera la información del clima de todas las ciudades solicitadas, ya sea en la terminal o usando una interfaz gráfica.

### **3. Selección de la mejor alternativa**

Utilizamos Python dado que queremos tomar un paradigma orientado a objetos y además porque el lenguaje cuenta con diversas bibliotecas que hacen que sea más fácil leer archivos y hacer las llamadas a la API.

Para leer el archivo csv que guarda la información de todos los tickets optamos por usar la biblioteca csv, que cuenta con una función `reader()`, que regresa un objeto que permite iterar por las líneas del archivo. Posteriormente separamos cada línea en palabras y con ellas creamos un objeto de tipo `Ticket` el cual guarda la información de toda una línea del archivo csv. Después de leer el archivo, nos quedamos con una lista de `Tickets`.

Para el web service optamos por usar `OpenWeather` por la completa documentación que tiene.

Decidimos crear dos métodos para el clima dentro de la clase `Ticket`, uno que regresa el clima de la ciudad de origen y otro que regresa el clima de la ciudad de llegada. En ambos se hace el request con la respectiva latitud y longitud de la ciudad y se usa el método `json()` para extraer los datos, dichos datos se usan para crear un objeto de tipo `WeatherInfo`, que contiene toda la información climática que necesitamos mostrar. Finalmente se regresa dicho objeto.

Para el caché decidimos crear un diccionario en el que las llaves son los códigos IATA de las

ciudades y los valores son objetos de tipo WeatherInfo. Antes de hacer un request a OpenWeather, revisamos si nosotros ya habíamos consultado el clima de esa ciudad previamente, si es que sí, entonces regresamos el valor asociado a esa llave en el diccionario, si es que no, entonces hacemos el request a OpenWeather y posteriormente actualizamos el diccionario con la nueva entrada.

Para pedir el ticket del que queremos conocer la información del clima y mostrar la información, decidimos hacer una interfaz gráfica ocupando la biblioteca tkinter.

## **4. Pseudocódigo**

## **5. Mantenimiento y requerimientos a futuro**