



**UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO**



FACULTAD DE CIENCIAS

Proyecto 1: Web Service

Modelado y Programación



INTEGRANTES DEL EQUIPO:

- **Alvarado Camacho Andrea**
- **Almanza Torres Jose Luis**
- **Mondragón Segoviano Alfonso**
- **Velázquez Rosas Abner Elias**

Web Service

1. Definición del problema

En este proyecto nos piden hacer un programa para el aeropuerto de la Ciudad de México, el cual requiere de un informe del clima de la ciudad de salida y la ciudad de llegada para 3 mil tickets que salen el mismo día que se corre este algoritmo. Para esto se nos proporciona un documento csv con la información necesaria para poder realizar el proyecto, por lo que, la persona encargada de utilizar el programa no debe ingresar nada, sólo debe visualizar el resultado.

2. Análisis del problema

Para poder consultar el clima de cada localidad se necesita conocer la forma en que se debe hacer las llamadas al API que se va elegir, por lo que podría utilizarse el nombre de cada ciudad para estas llamadas, sin embargo, en el archivo que nos brindan, este nombre viene abreviado. Así que, la mejor manera va a ser utilizar las coordenadas de cada ciudad para hacer las consultas, esto sería más que suficiente. Además, al realizar el algoritmo de este proyecto es esencial considerar que el web service que utilicemos se limita a proporcionar cierta cantidad de información por minuto, por lo que es fundamental que las ciudades que están repetidas en la base de datos no vuelvan a realizar una petición. También hay que tener en cuenta que el clima de cada ciudad debe abarcar los rubros de temperatura, temperatura mínima, temperatura máxima, humedad y sensación térmica. En conclusión, se necesita leer el archivo csv, realizar las peticiones del clima al web service, procesar la información devuelta y devolver los datos al usuario de la manera más amigable y eficiente posible.

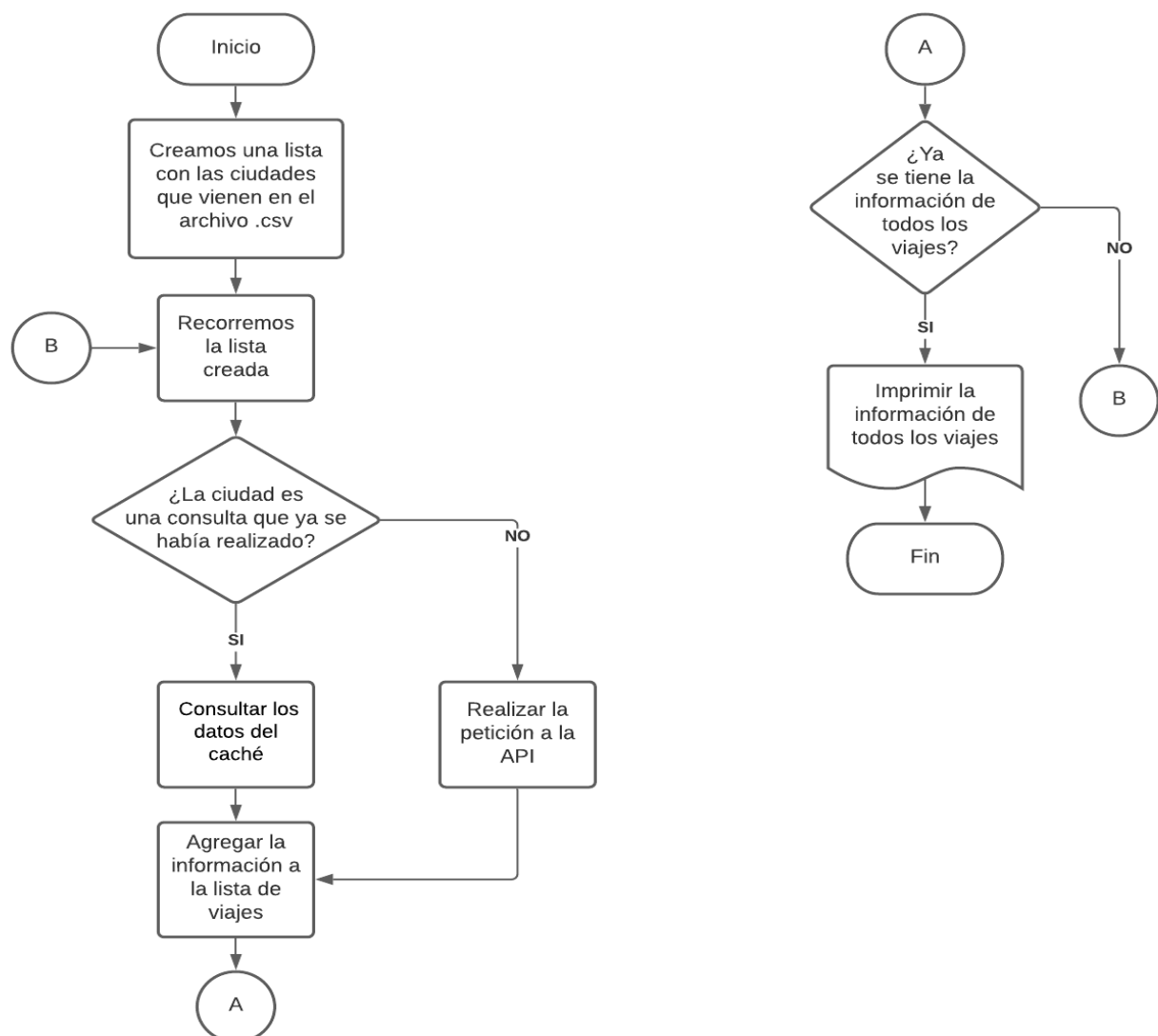
3. Selección de la mejor alternativa

Este proyecto será hecho con Java debido a que es un lenguaje que nos da una gran facilidad de reutilizar y mejorar algoritmos ya creados, esto gracias a que es un lenguaje orientado a objetos. Además funciona en cualquier sistema operativo ya que es multiplataforma. Para realizar las consultas del clima utilizaremos el web service llamado: OpenWeatherMap, el cual nos provee de una API que es eficiente y nos proporciona la información que necesitamos. La estructura que se emplea en

nuestro proyecto es a través de clases, las cuales nos facilitan el manejo de la información y nos ayuda a mantener un código más limpio.

La mejor forma de solucionar el problema de que las ciudades se repetirán en la base de datos, provocando así que se hicieran llamadas al API innecesarias y el programa tardará más tiempo en este proceso. Se decidió implementar un caché, el cual gracias a una lista en la que se almacenan objetos con la información del clima de cada ciudad, se procede a revisar antes de que se realice una llamada, provocando que si esa ciudad ya había pasado anteriormente, se regrese la información del lugar y no se hagan más peticiones, hasta que aparezca una nueva ciudad. Además, se decidió que la mejor manera de mostrar el clima de la ciudad de origen y de llegada sería a través de una interfaz gráfica, que permite una correcta interpretación de los datos del usuario que utilicé el programa.

4. Diagrama de flujo



Para poder realizar este algoritmo, primero se crea una lista la cual está compuesta por objetos que tienen diversos atributos como el nombre de la ciudad y sus coordenadas. Posteriormente, se hace un recorrido por esta lista para ir consultando el clima de cada ciudad a través de llamadas al API. Pero, algo muy importante es que en cada petición, esa ciudad se agrega a una nueva lista llamada viajes, así que cada llamada al API pasa por un filtro, el cual es el caché, que nos dice si esa ciudad ya había pasado con anterioridad. Esto, gracias a que se compara la primera lista con el contenido de la segunda y se sabe si esa ciudad ya había pasado con anterioridad y así devolver los resultados correspondientes. Posteriormente, cuando se tiene toda la información de los viajes, se regresa esta información al usuario que esté utilizando el programa.

5. Mantenimiento que va a requerir en un futuro y precio del proyecto

El mantenimiento que va a necesitar este programa es el ir actualizando cómo es que este lee y procesa la información de la base de datos, debido que puede existir la posibilidad de que en un futuro los datos del archivo csv estén diferentes, ocasionando que se necesite una actualización inmediata del programa. Otro aspecto importante es el de ampliar la cantidad de llamadas al API, si es necesario consultar una cantidad mayor del clima de las ciudades en un tiempo determinado, por lo que sería necesario implementar un plan diferente en OpenWeatherMap. Además, el mantenimiento puede incluir nuevas funciones como el de que la interfaz sea accesible y amigable a personas que hablan un idioma diferente, que se agregue nuevas herramientas para facilitar la búsqueda de ciudades, entre otras. Por este proyecto cobraría entre \$2500 y \$3500 pesos mexicanos. Y dependiendo del mantenimiento entre \$200 y \$500.