

# Proyecto 01 Web Service

## Universidad Nacional Autónoma de México

### FACULTAD DE CIENCIAS

### Modelado y Programación

Integrantes:

Andrea Aguirre González

Hannia Laura

Mónica Miranda Mijangos

12 de octubre de 2021

Día a día miles de personas van y vienen de aeropuertos a aeropuertos, cambiando luego drásticamente de zona horaria y clima. El aeropuerto de la Ciudad de México te contrata para una tarea, la cual es entregar el 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 el algoritmo. No es interactivo, solo nos interesa el clima.

## 1. Desarrollo

### 1.1. Definición del Problema

- ¿Qué es lo que queremos obtener? El informe del clima de la ciudad de salida y la ciudad de llegada para 3 mil tickets que salen el mismo día.
- ¿Cuáles son los datos que tenemos para obtenerlo? Una base de datos con 3 mil tickets proporcionada por el aeropuerto.
- ¿Son suficientes? Sí
- ¿Qué hace que el resultado obtenido resuelva el problema? ¿Cuál es la característica que hace de un resultado, una solución? El personal de aviación puede estar al tanto de los cambios drásticos del clima de las ciudades de las que partirá y de las ciudades que visitará.
- ¿Qué operaciones o construcciones se deben obtener para llegar a la solución? Leer la base de datos y hacer una solicitud a los aeropuertos acerca del clima en la ciudad para después entregar dicho informe para 3000 tickets.
- Entrada / Salida

- Tipo: String
- Formato: CSV / Texto en terminal
- Tamaño:
- Cantidad: 3000 tickets / 6000 informes
- Fuente / Diseño: Recolección de datos del Aeropuerto / Terminal?

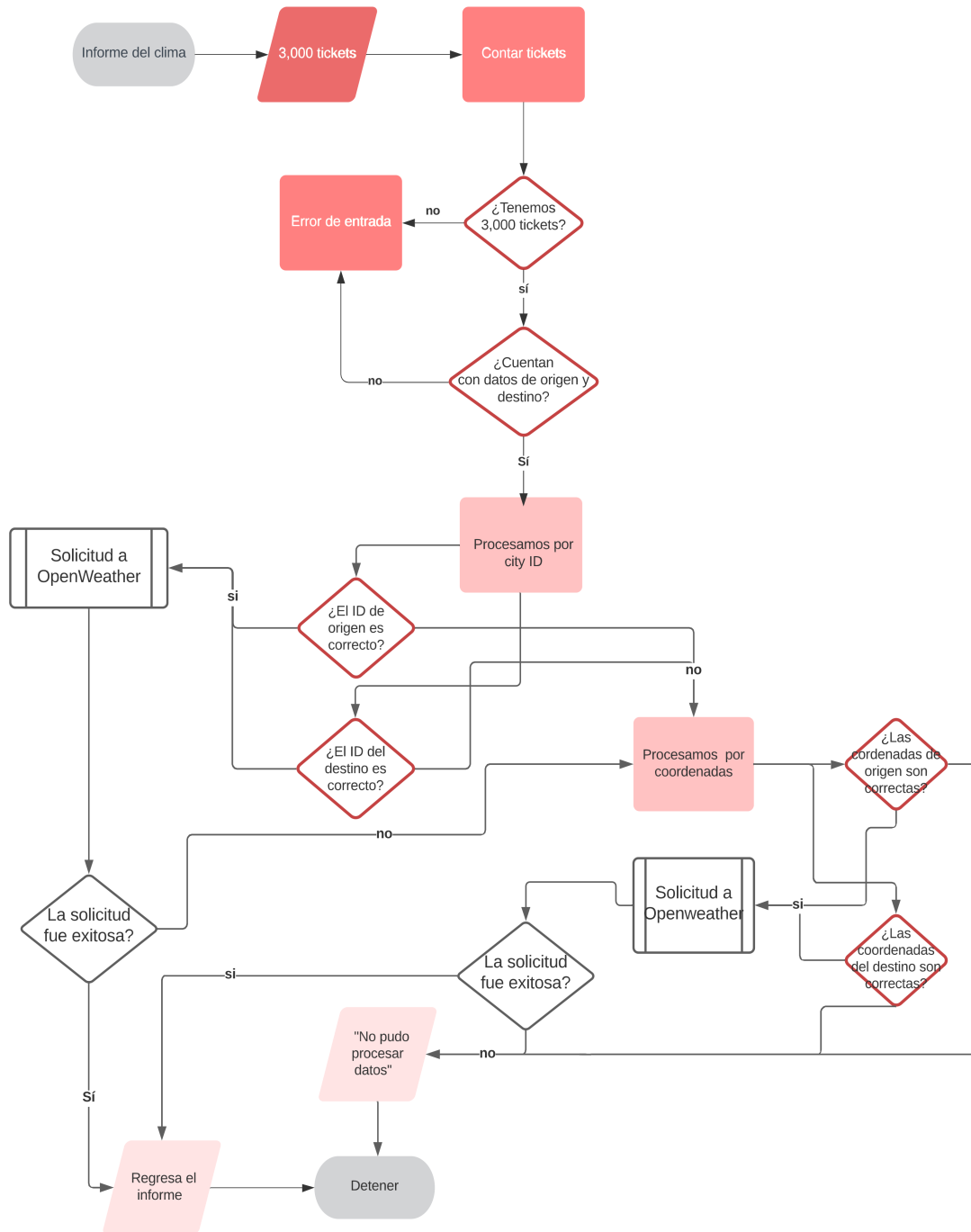
## 1.2. Análisis del Problema

- Requisitos funcionales:
  - ¿Qué debe entregar como salida dado cierto tipo de entrada? Dado un ticket con datos de origen y de destino debe regresar el reporte del clima para la ciudad de origen y la ciudad de destino.
  - ¿Qué se debe llevar a cabo? Al registrar un ticket, se debe mandar una solicitud a la API de clima elegida, la cual debe regresar el reporte del clima de la ciudad de origen y de destino.
- Requisitos no funcionales:
  - Eficiencia
  - Tolerancia a fallas
  - Amigabilidad
  - Escalabilidad
  - Seguridad
  - Interoperabilidad

## 1.3. Selección de la mejor alternativa

- Paradigma de programación: Orientado a Objetos
- Lenguaje: Python
- Herramientas para pre o post procesar datos: OpenWeather
- Frameworks o bibliotecas: request, json, csv, time...
  - ¿Qué tan útiles son? OpenWeather nos permite obtener la información del clima de cada ciudad. Request nos permite realizar enviar solicitudes HTTP fácilmente. CSV nos permite leer nuestra base de datos para manipular los datos.
  - ¿Qué se necesita para usarlas? Para OpenWeather se necesita una llave para realizar las solicitudes.
  - ¿Están bien documentadas? Sí
  - ¿Han sido probadas?
  - ¿Vale la pena usarlas? Definitivamente, nos ahorra hacer una petición a cada aeropuerto o ciudad para conocer el estado del clima. Así como de enviar manualmente una solicitud a http.
  - ¿Son seguras? Sí

## 1.4. Diagrama de flujo



## 1.5. Mantenimiento

- ¿Cuánto cobrarías por el proyecto y futuro mantenimiento? Para poder poner un precio al proyecto podemos desglosar una lista de tareas a través de las cuales se estimara el tiempo que se le dedicara a cada una en horas, hay que tener en cuenta que a cada una de las tareas se le deben agregar los costos que se requerirán para que el sistema funcione, como lo sería:
  - Equipo específico para el proyecto.
  - Servidores en la nube.
  - Dominios
  - Licencias
  - Bases de datos
- Una buena forma de determinar el precio por hora del proyecto es tomando en cuenta la experiencia que se tiene y en todo caso lo que se desea ganar mensualmente. Tomando esto en cuenta, de acuerdo a nuestra experiencia y horas trabajadas, el precio por hora sería de 125 MXN lo que nos da un total de 20,000 MXN.
- Tomando en cuenta de que el programa no es interactivo y se desea que se mantenga de esa manera, su mantenimiento es relativamente fácil. Las cosas a considerar serían el número de tickets que obtiene, ya sea un número mayor o menor al que fue establecido en este caso en particular, al igual que la expansión de la base de datos en caso de que se agregué un destino nuevo a la aerolínea.

Si es que se quisiera implementar algún tipo de acción donde el usuario tenga que interactuar con el programa, ya sea la opción de escoger un ticket específico dentro de la base de datos entonces ahí se tomarían en consideración las distintas formas que puede llevarse esto a cabo: creando una lista con los tickets dados, hacer que el usuario escriba el city ID o las coordenadas geográficas.

También hay que considerar el funcionamiento de el web service. En este caso en particular se utilizó la versión gratis de OpenWeather, este tiene restricciones y si es que se quisiera ampliar el paquete de servicios de este se tendría que considerar pagar por el servicio deseado o buscar un servicio web alternativo.