UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Facultad de Ciencias

Proyecto 1

Modelado y Programación

Autores:
Aguirre Muñoz Leonardo
318017686
wraithpuma@ciencias.unam.mx
Valencia Cruz Jonathan Josué

12 de octubre de 2021

1. Definición del problema

Dada una lista de 3000 vuelos con ciudad de origen y ciudad de destino, se solicita que se realice un programa que de un informe del clima en tiempo real de la ciudad de salida y la ciudad de llegada para cada uno de los vuelos. El programa no debe ser interactivo

Los datos del clima requeridos en el informe son los siguientes:

a) Descripción del clima

b) Temperaturas mínima y máxima

c) Humedad

d) Sensación térmica

e) Presión

2. Análisis del problema

Se nos da un archivo csv que almacena la siguiente información:

a) Ciudad de origen

b) Ciudad de destino

c) Longitud y latitud de origen

d) Longitud y latitud de destino

A partir de esta información se solicita que se entregue la información anteriormente dicha, esta información se va extraer de alguna API dedicada al clima y tiempo atmosférico, en este caso se hará uso de la API OpenWeatherMap, siendo que para su funcionamiento, requeriremos de algunos datos los cuales son:

a) Una Key personalizada por parte de OpenWeatherMap (Esta se solicita desde la página oficial de OpenWeatherMap, creando una cuenta de usuario)

b) Información de la ciudad o lugar a realizar la petición del informe del clima.

c) Lenguaje a solicitar la descripción del clima (En este caso español)

d) Unidades de medida para las diferentes variables del clima tales como temperatura y humedad (En este caso haremos uso del Sistema de Unidades Internacional)

Una vez identificado y obtenido la información requerida para hacer uso de la API, procedemos a realizar las llamadas a la API almacenando la información obtenida en algún lugar y comenzamos a vincular el informe del clima obtenido para su respectiva ciudad de origen y ciudad de destino. Esta información pasa a ser mostrada en pantalla o en algún otro medio en donde se pueda visualizar el contenido.

La cuestión es la siguiente, dado que nuestro número de solicitudes a la API está restringida a un cierto límite, debemos hacer uso de un método que nos ayude a evitar realizar solicitudes repetidas, es decir, podemos hacer uso de alguna memoria caché o por defecto filtrar la lista de ciudades de modo que no haya ciudades repetidas.

Otro problema a solucionar son los errores que pueden presentarse al momento de realizar una solicitud a la API, estos errores son diversos pero algunos de ellos son:

a) Error 404, este error surge cuando la información solicitada no se encuentra y esto puede deberse a diversos motivos como lo es tener problemas de conectividad.

b) Ciudad no encontrada, este es parte del error 404 pero quisimos mencionarlo debido a que es un problema a considerar en primera instancia.

c) Tiempo de solicitud excedida, este error puede surgir por ciertos motivos como lo es problemas de conectividad o se realizan varias peticiones de manera simultanea.

Para solucionarlo, se opta por un manejo de errores implementado en el programa llamados excepciones que, en caso de encontrar algún error, pasa a la siguiente solicitud.

3. Selección de la mejor alternativa

Se utilizó el lenguaje de programación en R ya que simplifica y aligera el trabajo de lectura, almacenamiento y modificación de extensas bases de datos además de implementar bibliotecas que ayudan en la lectura y escritura de documentos csv, siendo que el problema requiere de su uso.

Otra virtud que posee el lenguaje R es su fácil uso, lenguaje práctico y sencillo, haciendo que el proceso de aprendizaje de este lenguaje no sea un obstáculo al momento de querer realizar el proyecto.

De igual manera se usó la API OpenWeatherMap pues además de proporcionarnos la información requerida por el problema resulta fácil de usar y no requiere de un gran esfuerzo para comprender el funcionamiento de la misma, a su vez puedes modificar la información obtenida a partir de la misma como lo podría ser pasar de kelvin a grados centígrados o grados farenheit e inclusive cambiar el idioma.

También se hizo uso del formato JSON para la solicitud de información hacia la API pues resultaba fácil de cambiar el formato a csv por la biblioteca de r "jsonlite", siendo que, tanto el formato JSON como el formato csv, hacen del trabajo de lectura, escritura y manipulación de contenido algo más sencillo.

4. Diagrama de flujo o pseudocódigo (pueden hacer cualquiera de las dos o las dos) y explícalo.

```
#Se realiza una lectura del archivo y almacenarla en una variable llamada archivo
archivo <- read.csv("rutadelarchivo.csv")
filtro <- columnasrequeridas(origen,destino)
filtro1 <- comparar(origen,destino) #Si hay vuelos que aparecen mas de una vez
filtro2 <- convertiraLista(origen,destino)
listavuelos <- hacerListaUnica(t1$origin,t1$destination)
listabuena <- pasar una tabla(listavuelos)
vuelossinrep <- removerciudadesrepetidas(listabuena)

vuelossinrep <- covertirAcsv(vuelossinrep,"NombreArchivo.csv")

aero <- buscar("Aeropuertos.csv") #Obtener las ciudades que tiene esas abreviaciones
aero <- convertirAListaaero)

a <- no.de.elementos.lista(aero[[1]])
b <- no.de.elementos.lista(vuelossinrep[[1]])


ciclo(numero del 1 hasta a){
ciclo(numero2 del 1 hasta b){
si(?'aero.Abreviatura[numero2] es el mismo que vuelossinrep.listavuelos[numero])?{
 enEspacio[numero] agregar el valor de vuelo[numero]
 enEspacio[[numero]] ciudadnombre[numero2]
}
}
}
guardararchivo(enEspacio,"Nombredelarchivo.csv")

#Para guardar los datos del clima de las ciudades
tabla <-agregarcolumnas(tabla,Descripcion="",Temp.min="",Temp.max="",Sens.Termica="",
Humedad="",Presion="",.after="Ciudad")

////////////////////////////////////////////////////////////////////////
obtenerDatosClima <- obtenerDatosClima(Ciudad)
{
si (Ciudad == 'cuidado')
    {
        datos <- 'Ten cuidado'
    }
```

```
    si no evaluar (Ciudad == 'explota')
    {
        datos <- 'Error Garrafal'
        Parar("")
    }
    en caso contrario
    {
        datos <- llamadaAPI(Ciudad)
    }
    regresar (datos)
}

for(i in 1:largo){
Ciudad <- direccion(url para realizar la peticion)
intentar <- intentar({
DatosClima <- obtenerDatosClima(Ciudad)

},
cuidado=obtenerDatosClima()
{"Mostrar la advertencia"
},
error=identificarerror
{
Mostrar el error y detalles del mismo
},
seguir ejecutando con normalidad
{
})
Clima_filtro <- (Descripcion del clima, Presion, Temperatura minima,
Temperatura Maxima, Humedad y la Sensacion Termica)

tabla.Descripcion[i] <- Clima_filtro$Descripcion del clima
tabla.Temp.min[i] <- Clima_filtro.Temperatura Minima
tabla.Temp.max[i] <- Clima_filtro.Temperatura Maxima
tabla.Sens.Termica[i] <- Clima_filtro.SensacionTermica
tabla.Humedad[i] <- Clima_filtro.Humedad
tabla.Presion[i] <- Clima_filtro.Presion
}

#filtro son las columnas origen y destino obtenidas en una funcion
ejecutada con anterioridad
filtro <- agregarColumnasEn(filtro, Descripcion="",Temp.min="",Temp.max="",
Sens.Termica="",Humedad="",Presion="",despuesDeColumna="origen")

filtro <- agregarColumnasEn(filtro, Descripcion.Des="",Temp.min.Des="",Temp.max.Des="",
Sens.Termica.Des="",Humedad.Des="",Presion.Des="",despuesDeColumna=Destino)

longitud <- numerodeRenglones(tabla)
longitud2 <- numerodeRenglones(filtro)

ciclo(en i desde 1:longitud){
ciclo(en j desde 1:longitud2){
si(?'nuevo.Ciudad[i] tiene lo mismo en filtro.Origen[j]?){}
filtro.Descripcion[j] <- nuevo.Descripcion[i]
filtro.Temp.min[j] <- nuevo.Temp.min[i]
filtro.Temp.max[j] <- nuevo.Temp.max[i]
filtro.Sens.Termica[j] <- nuevo.Sens.Termica[i]
filtro.Humedad[j] <- nuevo.Humedad[i]
filtro.Presion[j] <- nuevo.Presion[i]
}si no revisar, si(?'nuevo.Ciudad[i] tiene lo mismo que filtro.Destino[j])?{
filtro.Descripcion.Des[j] agregar nuevo$Descripcion[i]
```

```
filtro.Temp.min.Des[j] agregar nuevo$Temp.min[i]
filtro.Temp.max.Des[j] agregar nuevo$Temp.max[i]
filtro.Sens.Termica.Des[j] agregar nuevo$Sens.Termica[i]
filtro.Humedad.Des[j] agregar nuevo$Humedad[i]
filtro.Presion.Des[j] agregar nuevo$Presion[i]
}
}
}
escribir.csv(filtro , "ClimaTodosLosVuelos.csv")
Por ultimo mostrar en pantalla los datos del clima de los vuelos
```

5. Mantenimiento y proyecto a largo plazo

A lo largo de la creación de este proyecto se tuvieron algunas complicaciones y se tiene en mente algunas cuestiones que podrían hacerse a futuro como parte de dar mantenimiento al programa, algunos de estos planteamientos son:

a) Una de las cosas a mejorar a corto plazo es la parte de implementación de una interfaz gráfica o de algún medio de visualización más agradable para el usuario, esta interfaz ayudaría para hacer del programa más amigable con el usuario.

b) Otra cuestión a remarcar es la implementación de la opción de actualizar el programa para así actualizar el estado del clima en las diferentes ciudades de origen y destino de los vuelos, la razón es simple y es que en una situación en la vida real un programa como este requiere de tener información lo más actualizada posible debido a que es de utilidad para diversos aeropuertos del mundo y pues, si el programa lo implementara sería un gran acierto.

c) El tercer posible mantenimiento que podría requerir el programa sería la parte de verificar actualizaciones, ya sean por parte del lenguaje de programación, de la API o incluso del mismo sistema operativo pues en el caso de la API podría suponer un problema mayor para el programa pues si se modificase algún apartado para su funcionamiento, haría de nuestro programa obsoleto y es por eso que sería un importante punto a considerar.

d) El cuarto es la actualización del código para atender complicaciones que puedan surgir por parte de los usuarios pues suele suceder que cuando se crea un programa, no se tiene en consideración algunos aspectos que pueden hacer de errores para algunos usuarios, algunos de estos errores pueden ser bugs, situaciones no planeadas por parte del programa, errores de compatibilidad con algún equipo por parte de algún usuario, entre otros.

e) El quinto y último punto a mencionar pero no menos importante es el mejoramiento y optimización de código de manera general pues al ser un proyecto creado a partir de programadores con poca experiencia y en formación da lugar a que el programa sea poco eficiente en algunos aspectos y que se podrían mejorar con la implementación de algunas metodologías.

Una vez dicho lo anterior, sobre la cuestión de cuánto se esperaría cobrar por la realización de este programa y por futuros mantenimientos, una situación a plantearse consiste en la labor de un programador, que presentan varios retos, siendo principalmente el planteamiento y desarrollo en la solución de un problema un aspecto clave a considerar al momento de estimar algún honorario además de quien solicita el servicio del programador, horario de trabajo, fecha de entrega, factores externos como lo es consultar el servicio de demás lugares, el tipo de problema a resolver; esto formula la "pregunta definitiva"de responder pero diremos que, en este caso, se cobraría por el programa un monto de alrededor de 1000 a 4000 pesos aproximadamente pues se trata de un proyecto menor y que no requiere de una gran planificación de por medio.