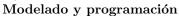


Proyecto 01 Semestre 2021-1

Prof. José Galaviz Casas Ayud. María Ximena Lezama





Kevin Ariel Merino Peña 1 Armando Abraham Aquino Chapa 2 9 de octubre de 2020

Pseudocódigo

CSVReader.py

Función 1: read_no_repeated_coordinates
Entrada: Nombre de un archivo (ruta) Salida : Lista de coordenadas no repetidas
 while Archivo(nombre dado) esté abierto do foreach renglones ← documento do if (latitud, longitud) no están en la lista then Agreagrlas

Función 2: read_csv_file

Entrada: Nombre de un archivo (ruta)
Salida: Lista de diccionarios con vuelos

1 try:

2 Abrir ruta

for $linea \leftarrow archivo$ do

 \downarrow linea \leftarrow lista

5 catch FileNotFoundError:

6 muesta Error, escribe una ruta válida

7 exit

8 catch FileExistsError:

muesta Error, archivo válido

10 exit

Función 3: read_headers

Entrada: Nombre de un archivo (ruta)

Salida : Lista de cabeceras

1 with:

 $\mathbf{2} \mid \text{lector} \leftarrow \mathbf{Leer} \ \mathbf{primera} \ \mathbf{linea} (\ \text{ruta})$

 $^{^1\}mathrm{Número}$ de cuenta 317031326

 $^{^2}$ Número de cuenta n

main.py

Función 4: validate_file Entrada: Nombre de un archivo (ruta) pasados como argumento al programa 1 if longiutd del argumento no es 2 then muestra: Error Debe indicar la ruta a un archivo csv 3 salir 4 5 if no coincide la extensión .csv then muestra: Error, sólo admito archivos csv salir $\mathbf{8}$ cabezera \leftarrow nombres de listas admitidas entrada_cabecera \leftarrow read_headers(argumento/1) if longitud(entrada_cabecera) no es iqual a longitud(cabezera) then muestra: ERROR El archivo csv debe tener los siguientes encabezados: origin, destination, origin_latitude, 11 origin_longitude, destination_latitude, destination_longitude salir **12** 13 foreach cabeza in entrada_cabezera do if cabeza no está en cabezera then 14 muestra: ERROR El archivo csv debe tener los siguientes encabezados: origin, destination, **15** origin_latitude, origin_longitude, destination_latitude, destination_longitude salir 16

Función 5: run

 $\bf Entrada:$ Nombre de un archivo (ruta)

- 1 validate_file(argumentos al correr el programa);
- 2 entradas ← read_csv_file(argumento al correr el programa)
- ${\tt 3}$ solicitudes_no_repetidas \leftarrow read_no_repeated_coordinates(argumentos al iniciar)
- 4 foreach $solicitud\ en\ solicitudes_no_repetidas\ do$
- $\mathsf{peticion} \leftarrow \mathtt{make_api_request_by_coordinates}(solicitud[0], solicitud[1])$
- 6 peticiones ← setdefault(solicitud, peticion)
- 7 foreach $entrada\ en\ entradas\ \mathbf{do}$
- muesta: Datos del clima;) con formato bonito

Weather.py

Función 6: make_api_request_by _coordinates

Entrada: latitud, lontigud

Salida: llamada a función parse_weather_info

1 if contador > 59 then

 $\mathbf{2} \mid \operatorname{contador} \leftarrow 0$

3 esperar 1 minuto para continuar

4 get(url + latitud y longitud dadas)

 $\mathbf{5}$ contador \leftarrow contador + 1

Función 7: formato_de_horas

Entrada: Número de fecha y hora (unix)

Salida : cadena de texto con hora en formato 12 hrs

1 convierte_fotante: numero dado

2 local_timezone ← get_localzone()()

3 local_time ← fromtimestamp(flotante, local_timezone)

4 regresar: local_time con formato de 12 horas, (CODIGO DEL TIEMPO)

Función 8: parse_weather_info

Entrada: respuesta en formato json

Salida : llamada a función parse_weather_info

1 try:

extraer información del archivo json con las llaves proporcionadas por la documentación de la API

з catch KeyError:

4 | regresar: Error, no se pudo consultar la información

5 regresar: El pronóstico del clima es: X , humedad: x

6 Temperatura actual: X°C, mínima: X°C, máxima: X°C Amanecer: X Puesta del sol: X