



UNIVERSIDAD DE GUADALAJARA
CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS
DEPARTAMENTO DE CIENCIAS COMPUTACIONALES

Workflow managers

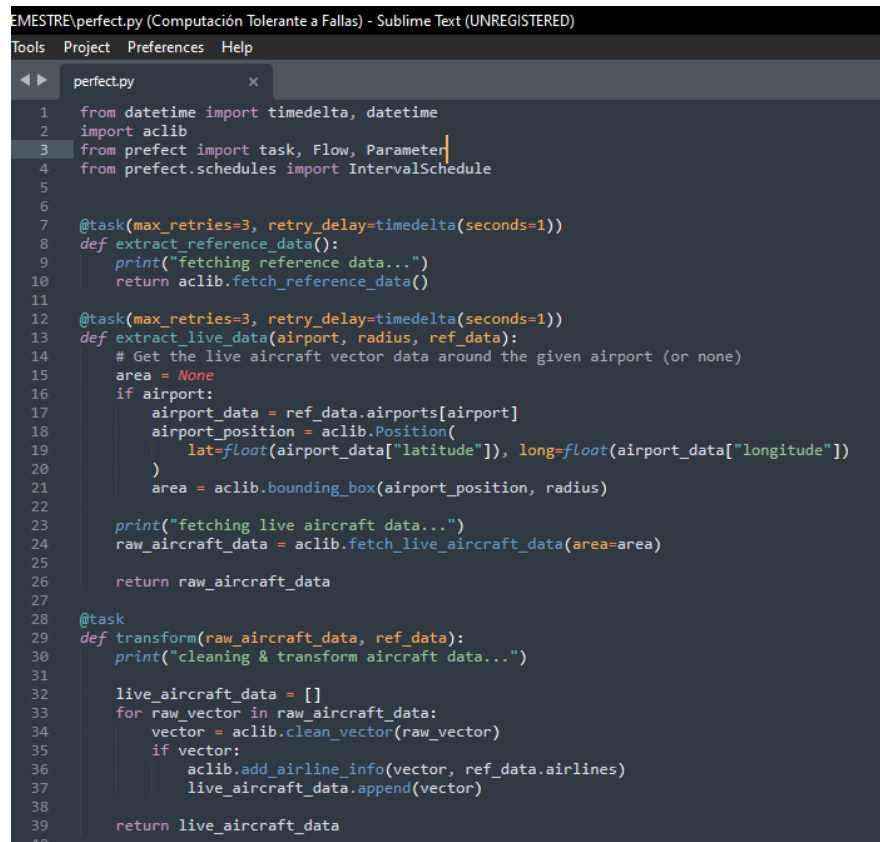
Oscar Evanilson Gutiérrez Pérez

Carrera: Ingeniería en Computación

Código de estudiante: 219748308

7 de marzo de 2022

En este programa, utilicé el lenguaje de Python para seguir un tutorial de internet que encontré en el que pude utilizar la librería prefect, es un ejemplo sencillo donde se va revisando información en vivo y va indicando el estado de cada una de estas que encuentra.



```
EMESTRE\perfect.py (Computación Tolerante a Fallas) - Sublime Text (UNREGISTERED)
Tools Project Preferences Help
perfect.py
1 from datetime import timedelta, datetime
2 import aclib
3 from prefect import task, Flow, Parameter
4 from prefect.schedules import IntervalSchedule
5
6
7 @task(max_retries=3, retry_delay=timedelta(seconds=1))
8 def extract_reference_data():
9     print("fetching reference data...")
10    return aclib.fetch_reference_data()
11
12 @task(max_retries=3, retry_delay=timedelta(seconds=1))
13 def extract_live_data(airport, radius, ref_data):
14     # Get the live aircraft vector data around the given airport (or none)
15     area = None
16     if airport:
17         airport_data = ref_data.airports[airport]
18         airport_position = aclib.Position(
19             lat=float(airport_data["latitude"]), long=float(airport_data["longitude"])
20         )
21         area = aclib.bounding_box(airport_position, radius)
22
23     print("fetching live aircraft data...")
24     raw_aircraft_data = aclib.fetch_live_aircraft_data(area=area)
25
26     return raw_aircraft_data
27
28 @task
29 def transform(raw_aircraft_data, ref_data):
30     print("cleaning & transform aircraft data...")
31
32     live_aircraft_data = []
33     for raw_vector in raw_aircraft_data:
34         vector = aclib.clean_vector(raw_vector)
35         if vector:
36             aclib.add_airline_info(vector, ref_data.airlines)
37             live_aircraft_data.append(vector)
38
39     return live_aircraft_data
40
```

En esta primera parte del ejemplo vemos las librerías que se importan para que funcione de manera correcta, primero tenemos la datetime que con esa va a poder marcar la hora de la información en vivo que va recabando, después tenemos una que se llama aclib y por ultimo tenemos la de prefect, esta es la librería que nos interesa en esta práctica.

Vemos que los modulos que se importan de prefect son task, Flow y parameter, estos son los que se van a utilizar en todo el programa.

```

40
41 @task
42 def load_reference_data(ref_data):
43     print("saving reference data...")
44     db = aclib.Database()
45     db.update_reference_data(ref_data)
46
47 @task
48 def load_live_data(live_aircraft_data):
49     print("saving live aircraft data...")
50     db = aclib.Database()
51     db.add_live_aircraft_data(live_aircraft_data)
52
53 def main():
54     schedule = IntervalSchedule(
55         start_date=datetime.utcnow() + timedelta(seconds=1),
56         interval=timedelta(minutes=1),
57     )
58
59     with Flow("etl", schedule=schedule) as flow:
60         airport = Parameter("airport", default="IAD")
61         radius = Parameter("radius", default=200)
62
63         reference_data = extract_reference_data()
64         live_data = extract_live_data(airport, radius, reference_data)
65
66         transformed_live_data = transform(live_data, reference_data)
67
68         load_reference_data(reference_data)
69         load_live_data(transformed_live_data)
70
71     flow.run()
72
73
74 if __name__ == "__main__":
75     main()

```

En esta parte del programa podemos ver como se implementan estos módulos mencionados a continuación, cada una de las task se ponen en una definición de una función y vemos que el módulo Flow se declara y al final del programa se corre para que inicie.

Link de supositorio en GitHub para revisar el código:

<https://github.com/oscarevanilson/Computacion-Tolerante-a-Fallas>

Conclusiones

Este programa me resultó demasiado complicado, aunque solamente haya tenido que seguir un tutorial de internet, la verdad es que se me hace muy complicado y confuso el funcionamiento de esta librería de prefect, además de que en este ejemplo son entendí muy bien que es lo que el programa revisaba y que información obtenía realmente.

Creo que esta librería es muy reciente y no hay demasiada información sobre ella en internet y por esto me costó trabajo entender bien el funcionamiento de esta. Por esta misma razón creo que esta herramienta es muy poco usada por lo que pude ver en los ejemplos que busqué en internet, espero que en un futuro pueda obtener más conocimiento de esta y que pueda utilizarla de mejor manera en tareas o programas más complejos.