



---

# WORKFLOW MANAGER

---

Emanuel Alejandro Gutierrez Romero



2 DE OCTUBRE DE 2023

COMPUTACION TOLERANTE A FALLAS

Prof. Michel Emanuel López Franco

## Introducción.

En este caso seguiremos el tutorial propuesto en el video para crear un Workflow manager. El código es en Python y el entorno es VS Code.

## Desarrollo.

```
import requests
import json
from collections import namedtuple
from contextlib import closing
import sqlite3

from prefect import task, Flow
```

En este caso estas son las librerías que llevarán a buen puerto el código.

```
@task
def get_complaint_data():
    r = requests.get("https://www.consumerfinance.gov/data-research/consumer-complaints/search/api/v1/", params={'size':10})
    response_json = json.loads(r.text)
    return response_json['hits']['hits']

## transform
@task
def parse_complaint_data(raw):
    complaints = []
    Complaint = namedtuple('Complaint', ['data_received', 'state', 'product', 'company', 'complaint_what_happened'])
    for row in raw:
        source = row.get('_source')
        this_complaint = Complaint(
            data_received=source.get('date_recieved'),
            state=source.get('state'),
            product=source.get('product'),
            company=source.get('company'),
            complaint_what_happened=source.get('complaint_what_happened')
        )
        complaints.append(this_complaint)
    return complaints

## load
@task
def store_complaints(parsed):
```

```

create_script = 'CREATE TABLE IF NOT EXISTS complaint (timestamp TEXT,
state TEXT, product TEXT, company TEXT, complaint_what_happened TEXT)'
insert_cmd = "INSERT INTO complaint VALUES (?, ?, ?, ?, ?)"

with closing(sqlite3.connect("cfpbcomplaints.db")) as conn:
    with closing(conn.cursor()) as cursor:
        cursor.executescript(create_script)
        cursor.executemany(insert_cmd, parsed)
        conn.commit()

```

Estas son las funciones tanto para cargar información como para guardar la información en el archivo Json.

```

with Flow("my etl flow") as f:
    raw = get_complaint_data()
    parsed = parse_complaint_data(raw)
    store_complaints(parsed)

f.run()

```

Y por último esta es la función que permite generar los "Flow" Y por ultimo se le da run a todo.

## Conclusión

Un Workflow Manager (gestor de flujos de trabajo, en español) es una herramienta o software que se utiliza para diseñar, controlar y supervisar los flujos de trabajo en una organización o en un proceso específico. Los flujos de trabajo son secuencias de tareas interconectadas que deben completarse en un orden específico para lograr un objetivo o resultado deseado.

Un Workflow Manager es esencialmente una plataforma que permite a las organizaciones automatizar, optimizar y supervisar sus procesos empresariales y tareas. Algunas de las características y capacidades comunes de un Workflow Manager incluyen:

1. Diseño de flujos de trabajo: Permite a los usuarios diseñar gráficamente flujos de trabajo, definir las tareas, las reglas de negocio y las condiciones de paso de una tarea a otra.
2. Automatización: Facilita la automatización de tareas y procesos, lo que ahorra tiempo y reduce errores al eliminar la necesidad de realizar tareas manualmente.

3. Gestión de tareas: Ayuda en la asignación y seguimiento de tareas a los miembros del equipo, lo que garantiza una distribución eficiente de la carga de trabajo.

4. Control de procesos: Proporciona visibilidad y control sobre el estado de los flujos de trabajo, lo que permite a los administradores supervisar y tomar medidas cuando sea necesario.

5. Integración: Puede integrarse con otras aplicaciones y sistemas empresariales, lo que facilita la transferencia de datos y la colaboración entre diferentes herramientas.

6. Notificaciones y alertas: Envía notificaciones y alertas a los usuarios cuando se requiere su acción o cuando se producen cambios en el flujo de trabajo.

7. Informes y análisis: Proporciona informes y análisis de los flujos de trabajo para evaluar la eficiencia y la calidad de los procesos.

Los Workflow Managers son utilizados en una variedad de industrias y sectores para optimizar procesos, mejorar la productividad, reducir errores y garantizar la consistencia en la ejecución de tareas. Pueden adaptarse a las necesidades específicas de una organización y pueden ser una parte fundamental de la transformación digital de una empresa.