



---

# APAHE AIRFLOW

---

Emanuel Alejandro Gutierrez Romero



16/10/23

COMPUTACION TOLERANTE A FALLAS  
Prof. Michel Emanuel López Franco

Introducción.

Comenzamos creando un archivo de nombre “airflow.cfg”, para después crear un archivo DAG.

```
from datetime import datetime
from airflow import DAG
from airflow.operators.dummy_operator import DummyOperator
from airflow.operators.python_operator import PythonOperator
```

Comenzamos importando las librerías que vamos a utilizar para la ejecución del programa.

```
# Definir las configuraciones del DAG
default_args = {
    'owner': 'Aldo',
    'start_date': datetime(2023, 10, 16),
    'retries': 1,
}
```

Comenzamos a definir las configuraciones de nuestro DAG, en este caso el dueño, la fecha en la que comienza y los retries.

```
# Crear un objeto DAG
dag = DAG('mi_primer_dag', default_args=default_args,
schedule_interval=None)

# Definir las tareas del DAG
start_task = DummyOperator(task_id='inicio', dag=dag)
```

Creamos nuestros objetos y definimos las tareas que va a ejecutar nuestro DAG.

```
def tarea_uno():
    print("Ejecutando tarea uno")

tarea_uno_task = PythonOperator(
    task_id='tarea_uno',
    python_callable=tarea_uno,
    dag=dag
)
```

Creamos las instancias de las clases, establecemos un id único para la tarea que queremos ejecutar, y el dag=dag establece en qué dag se va a incluir.

```
def tarea_dos():
    print("Ejecutando tarea dos")

tarea_dos_task = PythonOperator(
```

```
task_id='tarea_dos',  
python_callable=tarea_dos,  
dag=dag  
)
```

En esta parte básicamente hacemos lo mismo, pero ahora con la tarea dos, donde asignamos el identificador único y establecemos el dag.

```
end_task = DummyOperator(task_id='fin', dag=dag)  
  
# Definir las dependencias entre las tareas  
start_task >> tarea_uno_task >> tarea_dos_task >> end_task
```

Aquí creamos una tarea “Dummy”, básicamente las tareas dummy son tareas que no realizan ninguna operación únicamente se utilizan para marcar el inicio o el final de un flujo de trabajos o simplemente para agrupar tareas. Básicamente es como el orden en como se van a ejecutar las tareas, comenzando por la uno, por la segunda y finalmente terminando con la tarea.

Realmente no pude comprobar bien su funcionamiento ya que no instalé de manera correcta ciertas herramientas y otras se instalaron en versiones diferentes de python.