

Learning Objectives

- 1 Entender conceptos básicos de orquestación de workflows
- 2 Aprender la arquitectura de Apache Airflow
- 3 Comprender DAGs y dependencias entre tareas
- 4 Conocer instalación y configuración básica

Theory

Practice

Evidence

Quiz

Practical exercise to apply the concepts learned.

Ejercicio: Crear tu primer DAG funcional

Instalación básica de Airflow:

```
# Crear entorno virtual
python -m venv airflow_env
source airflow_env/bin/activate

# Instalar Airflow
pip install apache-airflow

# Inicializar base de datos
airflow db init

# Crear usuario admin
airflow users create \
  --username admin \
  --firstname Admin \
  --lastname User \
  --role Admin \
  --email admin@example.com
```

Crear primer DAG:

```
# dags/mi_primer_dag.py
from airflow import DAG
from airflow.operators.bash import BashOperator
from airflow.operators.python import PythonOperator
from datetime import datetime, timedelta

def saludar():
    print("¡Hola desde Airflow!")
    return "Saludo completado"

# Definir DAG
dag = DAG(
    'saludo_diario',
    description='DAG que saluda cada día',
    schedule_interval=timedelta(days=1), # Ejecutar diariamente
    start_date=datetime(2024, 1, 1),
    catchup=False, # No ejecutar ejecuciones pasadas
    tags=['ejemplo', 'saludo']
)

# Tarea 1: Comando bash
tarea_bash = BashOperator(
    task_id='tarea_bash',
    bash_command='echo "Ejecutando tarea bash a las $(date)"',
    dag=dag
)

# Tarea 2: Función Python
tarea_python = PythonOperator(
    task_id='tarea_python',
    python_callable=saludar,
    dag=dag
)

# Tarea 3: Esperar (simular procesamiento)
tarea_esperar = BashOperator(
    task_id='tarea_esperar',
    bash_command='sleep 5',
    dag=dag
)
```

```
)  
  
# Definir orden de ejecución  
tarea_bash >> tarea_python >> tarea_esperar
```

Ejecutar y monitorear:

```
# Iniciar scheduler (en terminal separado)  
airflow scheduler  
  
# Iniciar webserver  
airflow webserver --port 8080  
  
# Ejecutar DAG manualmente  
airflow dags unpause saludo_diario  
airflow dags trigger saludo_diario
```

Ver resultados:

```
# Ver Logs de ejecución  
# Visitar http://localhost:8080 en navegador  
# Ir a DAGs → saludo_diario → Graph View para ver flujo  
# Ir a Tree View para ver historial de ejecuciones
```

Verificación: ¿Qué diferencia hay entre un DAG y un simple script de Python? ¿Por qué es importante que los DAGs no tengan ciclos?

Requerimientos:

Python 3.7+
Apache Airflow instalado
Conocimiento básico de grafos y workflows