

Introducción a Apache Airflow - Día 1

Change Status

pending 40 min

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
--------	----------	----------	------

Activities and Learning

Task 1: ¿Qué es Apache Airflow? (10 minutos)

¿Por qué necesitamos orquestar pipelines?

Los pipelines complejos tienen múltiples pasos que deben ejecutarse en orden específico, con dependencias entre ellos.

Problemas sin orquestación:

- Tareas manuales propensas a error
- Dificultad para re-ejecutar pasos específicos
- Falta de visibilidad del estado del pipeline
- Problemas de coordinación en equipos

Solución: Apache Airflow

- Orquestación:** Coordina ejecución de tareas
- Programación:** Ejecuta en horarios específicos
- Monitoreo:** Dashboard para ver estado
- Reintentos:** Automatiza recuperación de fallos

Task 2: Arquitectura Básica de Airflow (10 minutos)

Componentes principales:

- Web Server:** Interfaz web para monitoreo
- Scheduler:** Programa y ejecuta DAGs
- Workers:** Ejecutan las tareas individuales
- Metadata Database:** Almacena estado y configuración

Flujo de trabajo típico:

Usuario escribe DAG → Scheduler detecta → Worker ejecuta tareas →
Usuario monitorea en Web UI → Resultados se almacenan

Ventajas de la arquitectura:

- Escalable:** Múltiples workers pueden procesar tareas en paralelo
- Fault-tolerant:** Si un worker falla, otros pueden continuar
- Observable:** Web UI muestra estado detallado de todas las tareas

Task 3: Conceptos Básicos de DAGs (10 minutos)

¿Qué es un DAG?

DAG = Directed Acyclic Graph (Grafo Dirigido Acíclico)

Características importantes:

- Directed:** Las tareas tienen dirección (A → B significa A antes que B)
- Acyclic:** No hay ciclos (no puede haber A → B → A)
- Graph:** Estructura de nodos conectados

Elementos de un DAG:

```
from airflow import DAG
from airflow.operators.dummy import DummyOperator
from datetime import datetime

# Definición básica de DAG
dag = DAG(
```




- Dashboard
- Career Path
- Forms
- Profile
- Support

```
'mi_primer_dag',
description='DAG de ejemplo',
schedule_interval='@daily',
start_date=datetime(2024, 1, 1),
catchup=False
)

# Tareas (nodos del grafo)
inicio = DummyOperator(
    task_id='inicio',
    dag=dag
)

fin = DummyOperator(
    task_id='fin',
    dag=dag
)

# Dependencias (flechas del grafo)
inicio >> fin
```

