

## DAGs dinâmicos: as duas maneiras!

# DAGs dinâmicos

Que assunto importante! 🤔

O conceito de "dinâmico" neste contexto é bastante simples. Sempre que você gerencia vários DAGs usando tarefas idênticas **em que apenas as entradas mudam**, pode fazer mais sentido gerar esses DAGs **dinamicamente**

Por que perder tempo escrevendo manualmente os mesmos DAGs repetidamente com variação mínima entre eles? 😊

Existem duas maneiras de gerar DAGs dinamicamente. Vamos começar analisando o primeiro:

## O método de arquivo único

Eu realmente não gosto deste e se você usá-lo, bem, leia o que vem a seguir, será muito útil para você.

O Método de Arquivo Único é baseado em um **único arquivo Python** responsável por gerar todos os DAGs com base em algumas entradas.

Por exemplo, uma lista de clientes, uma lista de tabelas, APIs e assim por diante.

Aqui está um exemplo de implementação desse método:

```

from airflow import DAG
from airflow.decorators import task
from datetime import datetime

partners = {
    'snowflake': {
        'schedule': '@daily',
        'path': '/data/snowflake'
    },
    'netflix': {
        'schedule': '@weekly',
        'path': '/data/netflix'
    },
}

def generate_dag(dag_id, schedule_interval, details, default_args):

    with DAG(dag_id, schedule_interval=schedule_interval, default_args=default_args) as dag:
        @task.python
        def process(path):
            print(f'Processing: {path}')

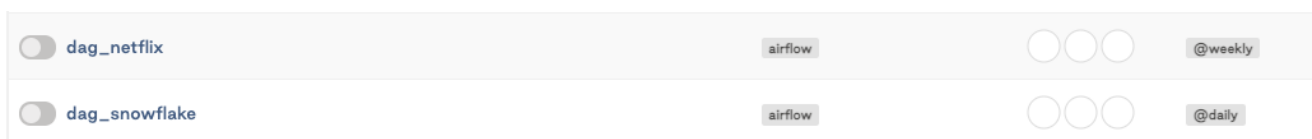
        process(details['path'])

    return dag

for partner, details in partners.items():
    dag_id = f'dag_{partner}'
    default_args = {
        'start_date': datetime(2021, 1, 1)
    }
    globals()[dag_id] = generate_dag(dag_id, details['schedule'], details, default_args)

```

Aqui, você vai gerar 2 DAGs diferentes de acordo com seus parceiros. Você vai acabar com:



Tudo certo! Mas, existem algumas desvantagens com este método.

- Os DAGs são gerados toda vez que o Agendador analisa a pasta DAG (se você tiver muitos DAGs para gerar, poderá ter problemas de desempenho)
- Não há como ver o código do DAG gerado na interface do usuário
- Se você alterar o número de DAGs gerados, acabará com DAGs visíveis na interface do usuário que não existem mais

Então, sim, é fácil de implementar, mas difícil de manter. É por isso que geralmente não recomendo esse método em produção para grandes implantações do Airflow.

Dito isto, vamos ver o segundo método, que eu prefiro muito mais! 😊

## O método de vários arquivos

Desta vez, em vez de ter um único arquivo Python responsável por gerar seus DAGs, você usará um script que criará um arquivo para cada DAG gerado. No final, você receberá um arquivo Python por DAG gerado.

Não vou fornecer o código completo aqui, mas deixe-me dar-lhe as diferentes etapas:

1. Crie um arquivo de modelo correspondente à sua estrutura de DAG com as diferentes tarefas. Nele, para as entradas, você coloca alguns placeholders que o script usará para substituir pelos valores reais.
2. Crie um script python encarregado de gerar os DAGs criando um arquivo e substituindo os espaços reservados em seu arquivo de modelo pelos valores reais.
3. Coloque o script, em outro lugar que não na pasta DAGs.
4. Acione esse script manualmente ou com seu pipeline de CI/CD.

Os prós deste método são numerosos!

- É escalável, pois os DAGs não são gerados toda vez que a pasta dags/ é analisada pelo Agendador
- Visibilidade total do código DAG (um DAG -> um arquivo)
- Controle total sobre a forma como os DAGs são gerados (script, menos propenso a erros ou DAGs "zumbi")

Obviamente, é preciso mais trabalho para configurar, mas se você tiver muitos DAGs para gerar, esse é realmente o caminho a seguir!



| PRODUTOS    | RECURSOS     | COMPANHIA |
|-------------|--------------|-----------|
| astro       | Documentos   | Sobre nós |
| Status      | Registro     | Carreiras |
| Apoiar      | Guias        | Contato   |
| Preços      | Webinars     | Parceiros |
| Por que o   | Academia     |           |
| fluxo de ar | Fórum        |           |
|             | Blogue       |           |
|             | Demonstração |           |
|             | semanal      |           |

Apache Airflow, Airflow e o logotipo Airflow são marcas registradas da Apache Software Foundation .