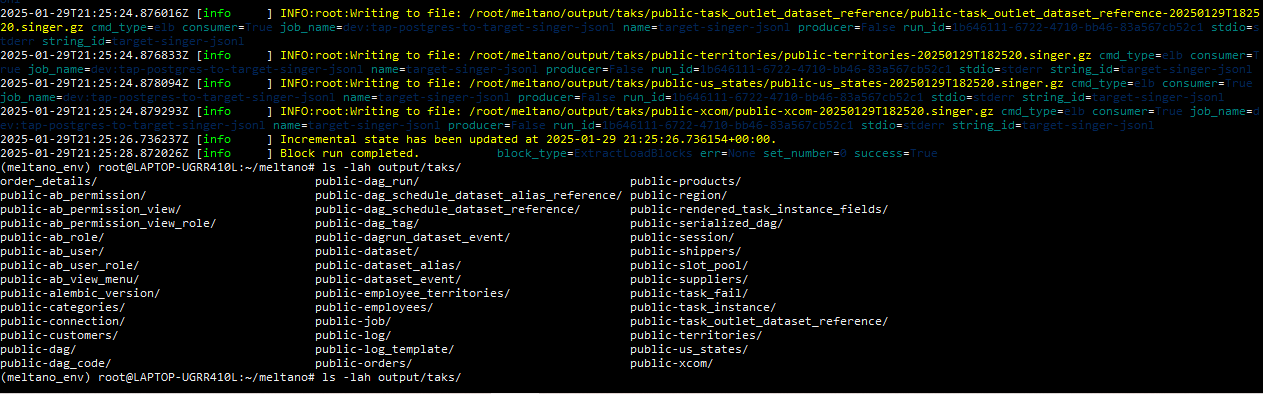
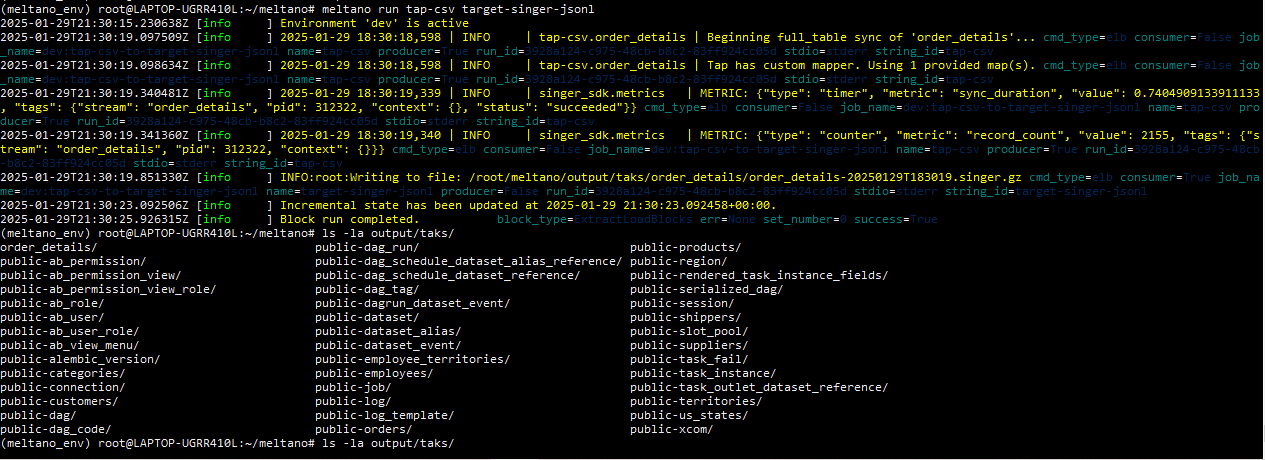
DESAFIO ENGENHARIA DE DADOS - **LH\_ED\_LUIZEDUARDO**

*Realizei o projeto na minha máquina local, onde instalei o banco de dados Postgres no Windows, e todo o restante fiz no ambiente Linux utilizando a ferramenta WSL do Windows. Para o MELTANO, eu criei um ambiente virtual, localizado em* ***“/mnt/c/Users/luizi/meltano\_env/”****. Alguns plugins não funcionaram com a versão do python 3.12, portanto precisei criar outro ambiente virtual, localizado em* ***“~/.meltano-venv/”*** *para utilizar o python 3.11. Foi preciso instalar também o Meltano (versão 3.6.0), o Apache Airflow (versão 2.10.4) para criação desse projeto.*

Este é um documento que explica o passo a passo para extrair os dados de duas fontes (arquivo CSV e o banco de dados Northwind). Na primeira parte da execução do pipeline, a ferramenta MELTANO executa o comando: **“meltano run tap-postgres target-singer-jsonl”** para realizar a extração dos dados de todas as tabelas do banco de dados, e armazena no diretório local (para esse desafio eu utilizei o caminho: **“/root/meltano/output/taks”**). Note que é gravado um arquivo para cada tabela do banco de dados.  
 Para executar manualmente o comando CLI, é necessário acessar o diretório onde está o projeto (neste caso, está no diretório: **/root/meltano**).  
  


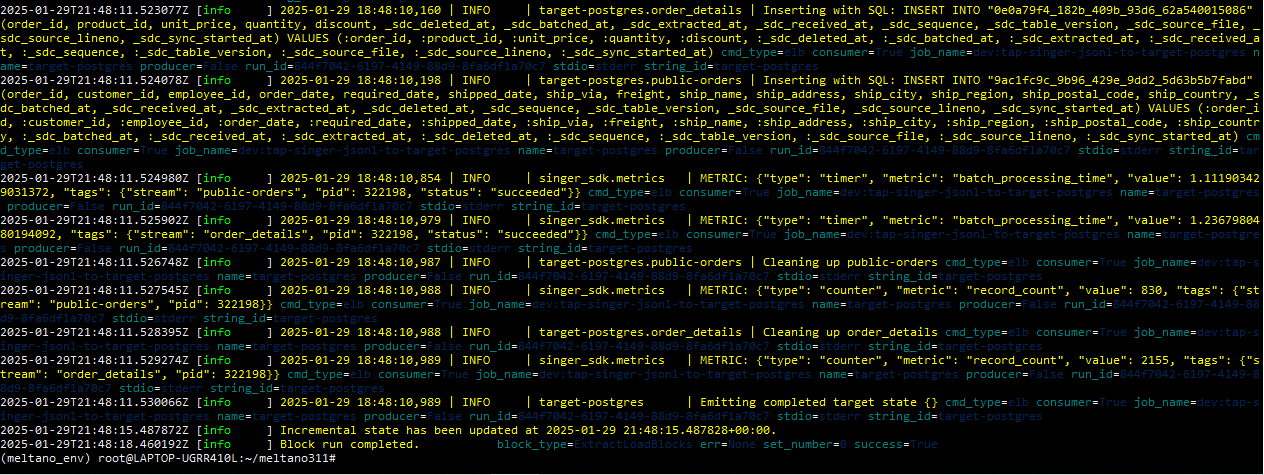
Após gravar os dados de todas as tabelas, o MELTANO executa outro comando: **“meltano run tap-csv target-singer-jsonl”** para realizar a extração dos dados da tabela **order\_details** disponível em arquivo CSV, e armazena no mesmo diretório local (**“/root/meltano/output/taks”**). Para executar o comando manualmente, é necessário acessar o diretório onde está o projeto (neste caso, está no diretório: **/root/meltano**).

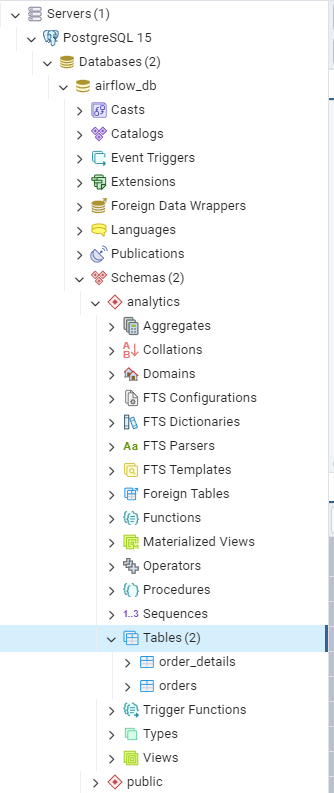


***OBS:*** *Não consegui fazer a configuração para salvar em pastas separadas por data de cada dia de execução.*

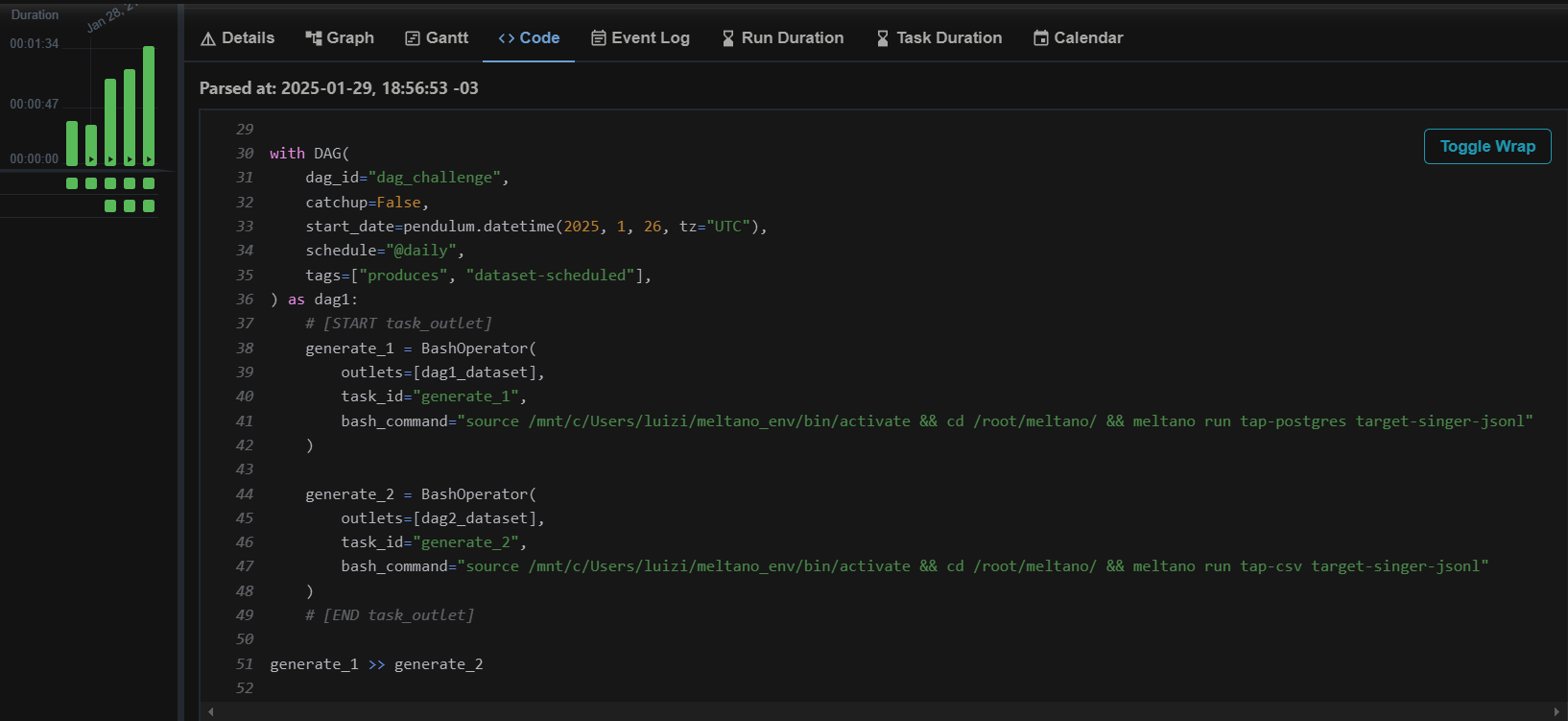
Após os arquivos serem gerados, o MELTANO passa para a segunda etapa do pipeline, onde ele carrega os dados dos arquivos gerados, e escreve no banco de dados.

Nessa etapa o MELTANO executa o comando “**meltano run tap-singer-jsonl target-postgres”** para ler os dados dos arquivos das tabelas **orders** e **orders\_details**, e escrever no Banco de dados Postgres em um novo database (aqui foi utilizado o nome **airflow\_db**), no schema **analytics**.

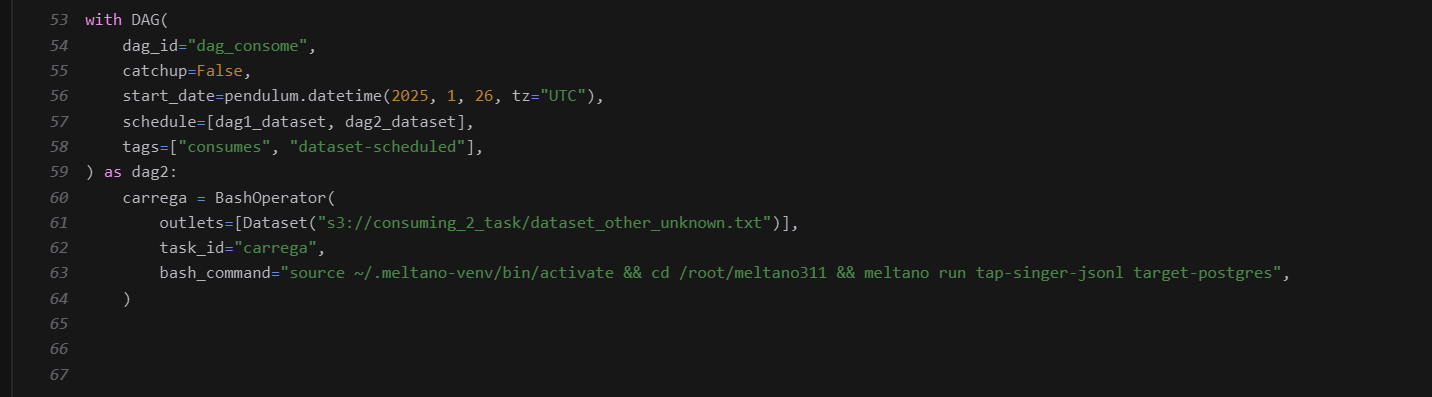




No Apache Airflow, criei duas DAGS que realizam a automação dos processos e etapas do MELTANO. A primeira DAG chamei de **“dag\_challenge”**, que está configurada para executar o comando de extração dos dados das tabela do banco Postgres e salvar os arquivos em diretório local, e posteriormente executar o comando de extração dos dados do arquivo .CSV e salvar também em diretório loca.



A segunda DAG se chama **“dag\_consome”**, que está configurada para executar o comando de leitura dos arquivos gerados na primeira etapa e posteriormente grava as informações no banco de dados nas tabelas **orders** e **order\_details**, no database **airflow\_db**, schema **analytics**. Essa DAG só será chamada se as duas *tasks* “generate\_1” e “generate\_2” forem executadas. Na linha ***schedule=[dag1\_dataset, dag2\_dataset]*** mostra que ela precisa dos dois datasets atualizados.



Por fim, após a execução do pipeline, é possível realizar a consulta no banco de dados para trazer as informações que mostram os pedidos e seus detalhes.

