**Preparação, Orquestração e Fluxos de Dados  
(PUC MINAS)**

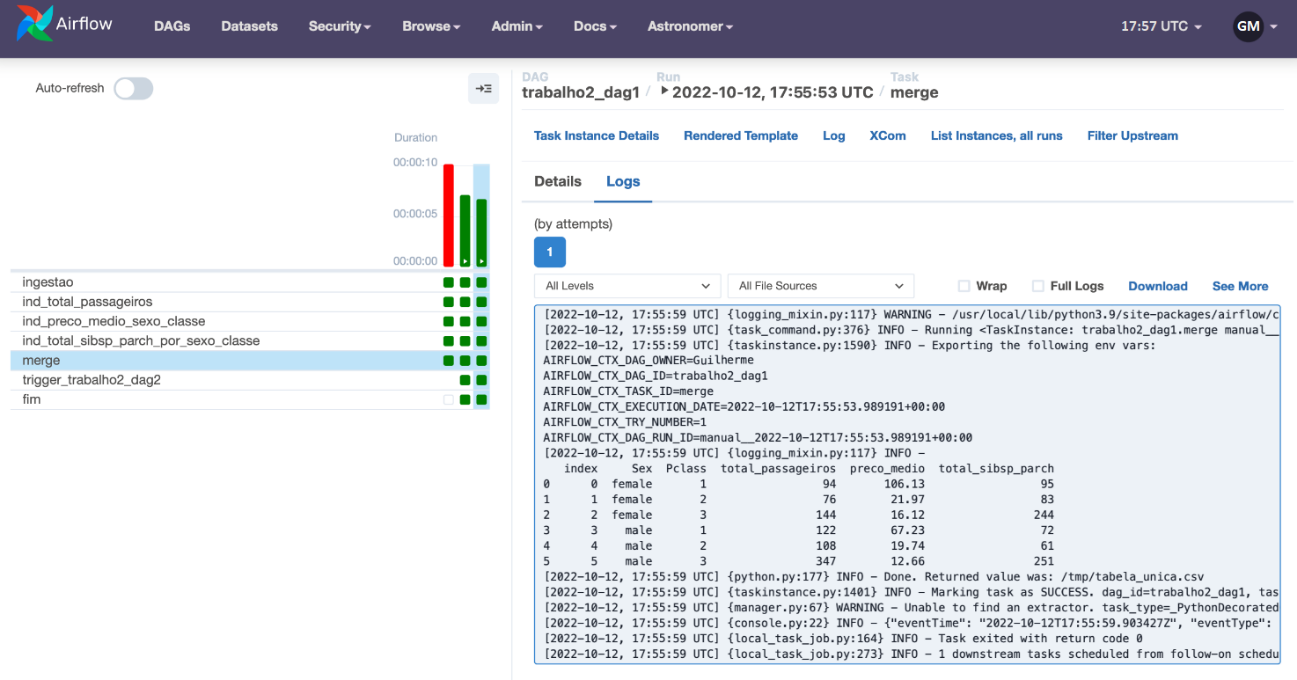
**Nome:** Guilherme Pereira Mendes

**1)** Código da DAG1

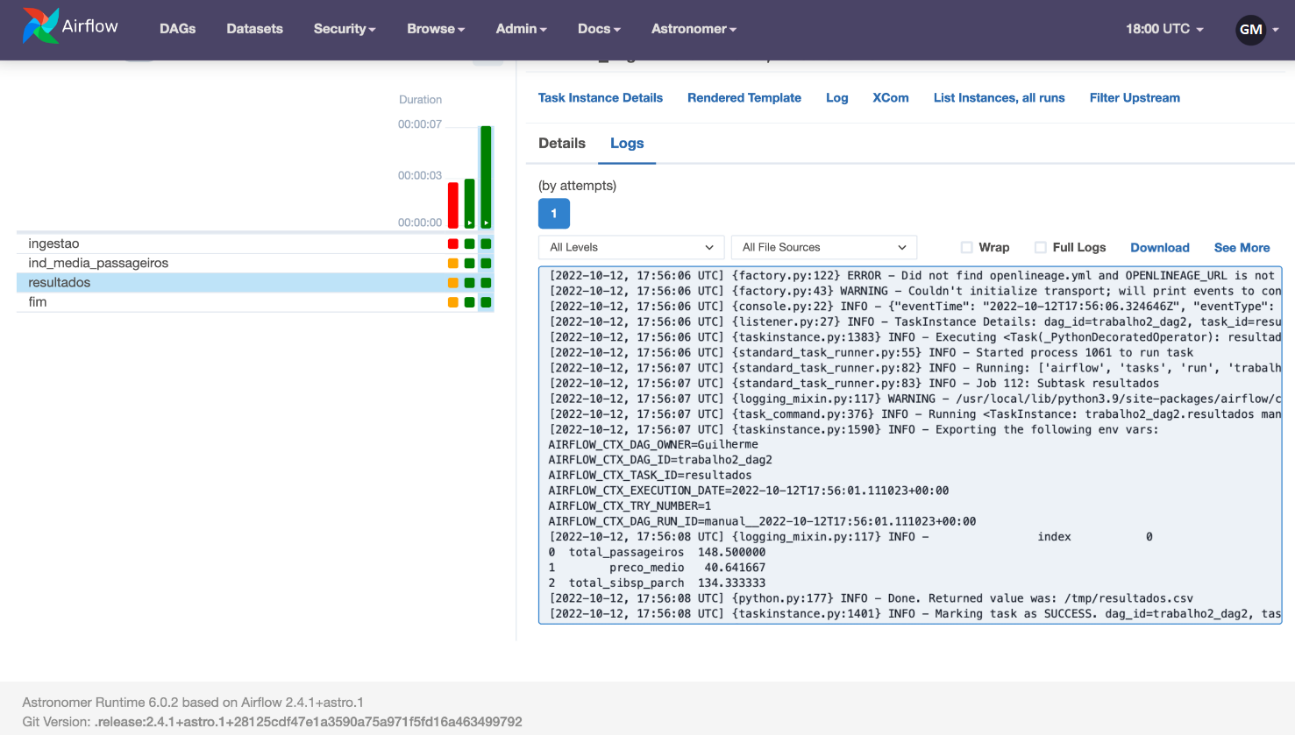
* import pandas as pd
* import functools as ft
* from airflow.decorators import dag, task
* from airflow.operators.dummy import DummyOperator
* from datetime import datetime, timedelta
* from airflow.operators.trigger\_dagrun import TriggerDagRunOperator
* URL = "https://raw.githubusercontent.com/neylsoncrepalde/titanic\_data\_with\_semicolon/main/titanic.csv"
* default\_args = {
* 'owner': 'Guilherme',
* 'dependes\_on\_past': False,
* 'start\_date': datetime(2002, 9, 29)
* }
* @dag(*default\_args*=default\_args, *schedule\_interval*='@once', *catchup*=False, *tags*=["Guilherme", "PUC", "dag1"])
* def trabalho2\_dag1():
* @task
* def ingestao():
* NOME\_DO\_ARQUIVO = "/tmp/titanic.csv"
* df = pd.read\_csv(URL, *sep*=';')
* df.to\_csv(NOME\_DO\_ARQUIVO, *index*=False, *header*=True, *sep*=";")
* return NOME\_DO\_ARQUIVO
* @task
* def ind\_total\_passageiros(*nome\_do\_arquivo*):
* NOME\_TABELA = "/tmp/passageito\_por\_sexo\_class.csv"
* df = pd.read\_csv(nome\_do\_arquivo, *sep*=';')
* res = df.groupby(['Sex', 'Pclass']).agg({
* "PassengerId": "count"
* }).reset\_index()
* res.rename(*columns*={'PassengerId': 'total\_passageiros'}, *inplace*=True)
* print(res)
* res.to\_csv(NOME\_TABELA, *index*=False, *sep*=';')
* return NOME\_TABELA
* @task
* def ind\_preco\_medio\_sexo\_classe(*nome\_do\_arquivo*):
* PATH\_SAIDA = "/tmp/preco\_medio\_por\_sexo\_classe.csv"
* df = pd.read\_csv(nome\_do\_arquivo, *sep*=';')
* res = df.groupby(['Sex', 'Pclass']).agg({
* "Fare": "mean"
* }).reset\_index()
* res.rename(*columns*={'Fare': 'preco\_medio'}, *inplace*=True)
* res['preco\_medio'] = res['preco\_medio'].round(*decimals*=2)
* print(res)
* res.to\_csv(PATH\_SAIDA, *index*=False, *sep*=';')
* return PATH\_SAIDA
* @task
* def ind\_total\_sibsp\_parch\_por\_sexo\_classe(*nome\_do\_arquivo*):
* PATH\_SAIDA = "/tmp/familia\_por\_sexo\_classe.csv"
* df = pd.read\_csv(nome\_do\_arquivo, *sep*=';')
* soma = df['SibSp']+df['Parch']
* df['total'] = soma
* res = df.groupby(['Sex', 'Pclass']
* ).agg({
* "total": "sum"
* }).reset\_index()
* res.rename(*columns*={'total': 'total\_sibsp\_parch'}, *inplace*=True)
* print(res)
* res.to\_csv(PATH\_SAIDA, *index*=False, *sep*=';')
* return PATH\_SAIDA
* @task
* def merge(*path1*, *path2*, *path3*):
* raw\_passageiros = path1
* raw\_tarifa = path2
* raw\_familiares = path3
* NOME\_TABELA = "/tmp/tabela\_unica.csv"
* passageiros = pd.read\_csv(raw\_passageiros, *sep*=";")
* tarifa = pd.read\_csv(raw\_tarifa, *sep*=";")
* familiares = pd.read\_csv(raw\_familiares, *sep*=";")
* df\_resultado = (
* passageiros
* .merge(tarifa, *how*="inner", *on*=['Sex', 'Pclass'])
* .merge(familiares, *how*="inner", *on*=['Sex', 'Pclass'])
* ).reset\_index()
* print("\n"+df\_resultado.to\_string())
* df\_resultado.rename(*columns*={'Sex': 'sexo'}, *inplace*=True)
* df\_resultado.rename(*columns*={'Pclass': 'classe'}, *inplace*=True)
* df\_resultado.to\_csv(NOME\_TABELA, *index*=False, *sep*=";")
* return NOME\_TABELA
* fim = DummyOperator(*task\_id*="fim")
* ing = ingestao()
* ind\_tp = ind\_total\_passageiros(ing)
* ind\_mp = ind\_preco\_medio\_sexo\_classe(ing)
* ind\_ts = ind\_total\_sibsp\_parch\_por\_sexo\_classe(ing)
* me = merge(ind\_tp, ind\_mp, ind\_ts)
* trigger\_dag2 = TriggerDagRunOperator(
* *task\_id*='trigger\_trabalho2\_dag2',
* *trigger\_dag\_id*='trabalho2\_dag2'
* )
* [ind\_tp, ind\_mp, ind\_ts] >> me >> trigger\_dag2 >> fim
* execucao = trabalho2\_dag1()

**2)** Código da DAG2

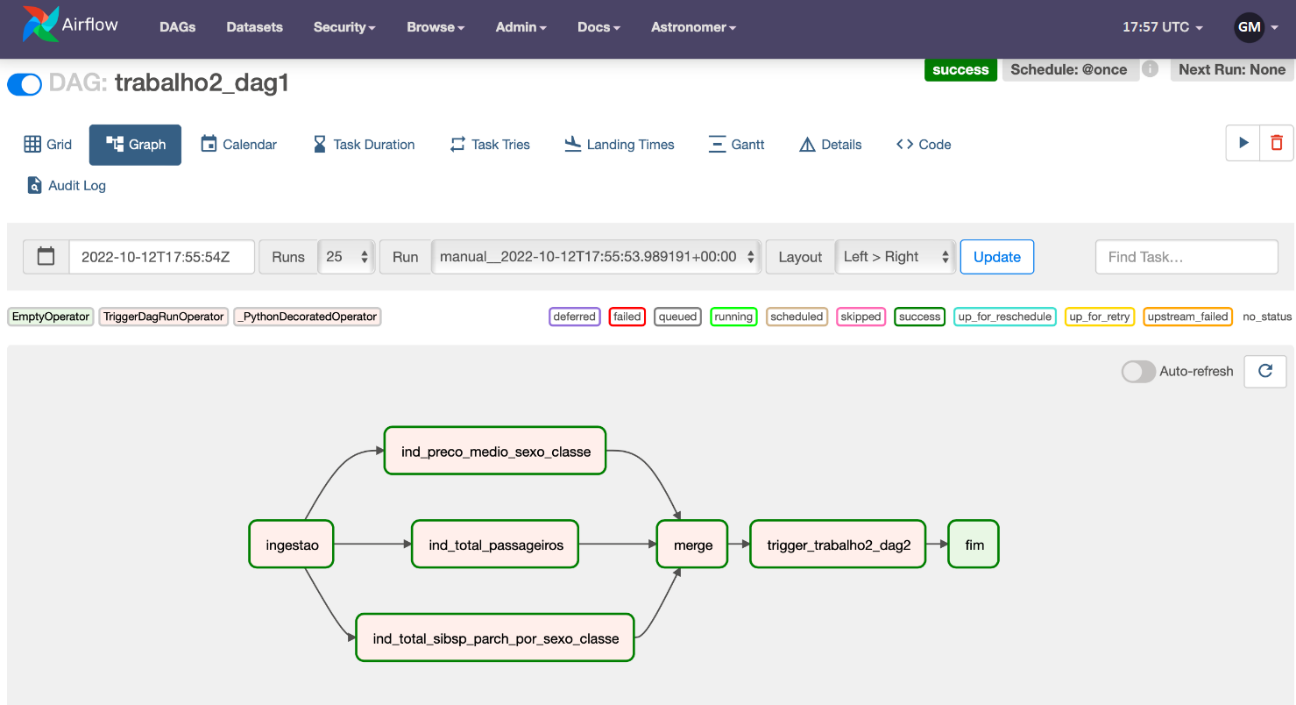
* import pandas as pd
* import functools as ft
* from airflow.decorators import dag, task
* from airflow.operators.dummy import DummyOperator
* from datetime import datetime, timedelta
* default\_args = {
* 'owner': 'Guilherme',
* 'dependes\_on\_past': False,
* 'start\_date': datetime(2002, 9, 29)
* }
* @dag(*default\_args*=default\_args, *schedule\_interval*='@once', *catchup*=False, *tags*=["Guilherme", "PUC", "dag2"])
* def trabalho2\_dag2():
* @task
* def ingestao():
* NOME\_DO\_ARQUIVO = "/tmp/tabela\_unica.csv"
* df = pd.read\_csv(NOME\_DO\_ARQUIVO, *sep*=';')
* df.to\_csv(NOME\_DO\_ARQUIVO, *index*=False, *header*=True, *sep*=";")
* print(df)
* return NOME\_DO\_ARQUIVO
* @task
* def ind\_media\_passageiros(*nome\_do\_arquivo*):
* NOME\_DO\_ARQUIVO = "/tmp/tabela\_media\_total.csv"
* df = pd.read\_csv(*nome\_do\_arquivo*, *sep*=';')
* res = df.agg({
* "total\_passageiros": "mean",
* "preco\_medio": "mean",
* "total\_sibsp\_parch": "mean"
* }).reset\_index()
* res.rename(
* *columns*={'total\_passageiros': 'media\_passageiros'}, *inplace*=True)
* res.rename(*columns*={'preco\_medio': 'media\_preco'}, *inplace*=True)
* res.rename(
* *columns*={'total\_sibsp\_parch': 'media\_sibsp\_parch'}, *inplace*=True)
* print(res)
* res.to\_csv(NOME\_DO\_ARQUIVO, *index*=False, *sep*=';')
* return NOME\_DO\_ARQUIVO
* @task
* def resultados(*path1*):
* PATH\_SAIDA = "/tmp/resultados.csv"
* df\_final = pd.read\_csv(*path1*, *sep*=';')
* print(df\_final)
* df\_final.to\_csv(PATH\_SAIDA, *index*=False, *sep*=';')
* return PATH\_SAIDA
* fim = DummyOperator(*task\_id*="fim")
* ing = ingestao()
* ind = ind\_media\_passageiros(ing)
* p = resultados(ind)
* ind >> p >> fim
* execucao = trabalho2\_dag2()

**3)** Imagem (PNG, JPG) com o print da tabela 1 no log  
 

**4)** Imagem (PNG, JPG) com o print da tabela 2 no log



**5)** Imagem com print do grafo da DAG1



**6)** Imagem com print do grafo da DAG2

