

Questão 1. Modifique o Projeto 1 para executar a DAG 20 vezes, uma vez por minuto, a partir de uma hora específica. Envie a DAG modificada e o resultado da execução.

```
# Aluno: Carlos Eduardo Rocha Miranda
# Professor: Jesmmer.
from datetime import timedelta
from airflow import DAG
from airflow.operators.python import PythonVirtualenvOperator
from airflow.utils.dates import days_ago
from airflow.timetables.trigger import CronTriggerTimetable
# OBJETIVO DA DAG: Essa dag acessa uma api que retorna piadas aleatrias.

def callable_virtualenv_collect_joke():
    import requests # Biblioteca que permite fazer requisies HTTP.
    import json # Biblioteca que permite lidar com dados JSON usando Python.

    # Biblioteca que permite ler e escrever dados tabulares no formato csv.
    from csv import DictWriter

    # Faz a requisio na api pra pegar uma piada.
    resp = requests.get("https://official-joke-api.appspot.com/random_joke")
    # Se vier qualquer cdigo que no seja 200(que significa sucesso), alguma coisa de
    errado aconteceu.
    if resp.status_code != 200:
        # Se deu errado, informa o erro com o cdigo.
        return f'Failed with response code {resp.status_code}'
    else: # A requisio deu certo.
        # Pega o que veio em json e transforma num objeto em python.
        joke_dict = json.loads(resp.text)
        # Abre ou cria um arquivo chamado jokes.csv.
        with open('./resultado/jokes.csv', 'a+', newline='') as write_obj:
            # Define as colunas desse arquivo.
            field_names = ['id', 'type', 'setup', 'punchline']
            # Escreve no arquivo.
            dict_writer = DictWriter(write_obj, fieldnames=field_names)
            # Coloca a piada no arquivo jokes.csv.
            dict_writer.writerow(joke_dict)

default_args = {
    'owner': 'airflow',
    'depends_on_past': False,
    'email': ['airflow@example.com'],
    'email_on_failure': False,
    'email_on_retry': False,
    'retries': 1,
    'retry_delay': timedelta(minutes=1),
}
```

```

with DAG(
    'joke_collector',
    default_args=default_args,
    description="Uma DAG que coleta piadas da API do gerador de piadas aleatrios.",
    start_date=days_ago(1),
    tags=['automationwithairflow'],
    # Minuto(0 - 59) Hora(0 - 23) Dia do ms(1 - 31) Ms(1 - 12) Dia da Semana(0 -
    6)(Domingo at o Sbado)
    timetable=CronTriggerTimetable("0-20 15 * * *", timezone="UTC"),

) as dag:
    joke_collector_task1 = PythonVirtualenvOperator(
        task_id="joke_collector_task",
        python_callable=callable_virtualenv_collect_joke,
        requirements=['requests==2.25.1'],
        system_site_packages=True,
    )
    """
    Permite que uma funo rode dentro de um ambiente virtual que criado e
    desmantelado automaticamente.
    Para funcionar a funo deve ser definida com def e no ser parte de uma classe.
    Nela voc pode realizar imports.
    """

    joke_collector_task1

```

Questão 2. Envie os resultados da execução do Projeto 2.

Não consegui fazer o segundo projeto executar.