

## Instruções de uso para o Projeto

Candidato: João Gabriel Soares da Motta

Email: [joao.motta.dev@gmail.com](mailto:joao.motta.dev@gmail.com)

Celular/Whatsapp: +55 (21) 98619-4413

Github do projeto: <https://github.com/joao-motta-dev/UBC.git>

Data: 17/07/2025

### 1. Considerações iniciais

- a. Todos os arquivos necessários para avaliação estão no Github do projeto.
- b. Necessário ter o Docker, pode ser o Personal, na máquina e o VS Code com extensão para Python.
- c. O próprio script faz o download do arquivo alunos.csv, alvo da análise.
- d. Estou à plena disposição para apresentação e esclarecimentos sobre o projeto.

### 2. Preparação do Ambiente

- a. Como citado no item 1.b serão necessários alguns downloads primeiro
  - i. Docker Personal: <https://www.docker.com/products/personal/>
  - ii. VS Code: <https://code.visualstudio.com/download>
  - iii. Recomendo utilizar a extensão Python da Microsoft no VS Code.
- b. Recomendo baixar os arquivos do Github e utilizar a estrutura de pastas que está no link. O caminho pode ser de sua preferência.
- c. Salve os arquivos dentro de uma pasta chamada *airflow*.

### 3. Docker

- a. Abra o Docker, e no terminal direcione para a pasta que com os arquivos na estrutura recomendada, o comando para isso é o `cd`. Por exemplo: `cd airflow`.
- b. Abra o arquivo `wait-for-services.sh` no VS Code e verifique se no canto inferior direito da tela temos a sigla “CRLF” ou “LF”. Para esse arquivo é imprescindível que esteja em “LF”.
- c. Com o terminal apontando para a pasta com os arquivos, executar o comando `docker-compose build` no terminal para criar o container.
- d. Após a conclusão do comando acima, executar `docker-compose up` para subir o container.
- e. Aguarde um minuto para todas as instancias subirem corretamente. Durante esse tempo monitore para ver se alguma das instancias cai durante o processo, sendo que somente a “init” pode cair sem gerar prejuízos para o processo nesse caso.

## Instruções de uso para o Projeto

- f. Fique atento ao “webservice”, “scheduler” e “worker”, pois sem eles estarem funcionando corretamente o processo irá falhar em algum ponto.

### 4. Airflow

- a. Abra o endereço <http://localhost:8080/> para acesso ao Airflow
- b. Provavelmente na tela inicial irá aparecer a DAG (projeto do Airflow) Projeto\_Joao\_Motta. Clique nela.
- c. Caso você comece por outra tela é só clicar em “DAGs” na parte superior da tela.
- d. Uma vez dentro da DAG, procure o botão de “play” no canto superior direito da tela e clique nele.
- e. Recomendo utilizar a visualização “graph” para acompanhar o fluxo.
- f. O Airflow não muda os status “automaticamente” então é necessário atualizar a página para vermos a mudança de status. Faça isso a cada 15 segundos em média.
- g. Verifique se todas as etapas obtiveram “success”.
- h. Com esses status para todas as etapas, ocorreram duas coisas:
  - i. Na pasta “dados” da estrutura de pastas foram gerados dois arquivos. Um alunos.csv que é o download feito originalmente, e arquivo\_saida.xlsx com todas as alterações e transformações.
  - ii. Os dados foram enviados ao Solr.

### 5. Solr

- a. Acesse o link <http://localhost:8983/solr>
- b. No lado esquerdo da tela procure uma caixa de seleção logo abaixo de “Suggestions” e selecione “alunos”.
- c. Uma vez selecionado “alunos”, procure embaixo da caixa a opção query e clique nela.
- d. Clique no botão azul “query” e veja se os dados aparecem ao lado. Caso positivo o processo correu perfeitamente.

### 6. Finalização

- a. Feche todas as telas de navegador abertas
- b. No terminal do Docker utilizar o comando `docker-compose down` para fechar o container.