

	<p>Instituto Federal de Educação Ciência e Tecnologia – Sudeste de Minas Gerais</p> <p>Disciplina: Visualização de Dados <span style="float: right;">Data: 22/07/2024</span></p> <p>Aluno: _____ <span style="float: right;">Valor: 4,0 pontos</span></p> <p>Data da Entrega: 30/07/2024 <span style="float: right;">Grupo: 02 alunos</span> <span style="float: right;">Entrega: SIGAA</span></p>		
--	--	--	--

O objetivo deste trabalho é construir uma aplicação para um projeto de Machine Learning de ponta a ponta. Ou seja, desde a concepção do projeto, passando pela análise exploratória de dados, até o deploy da aplicação em uma plataforma em nuvem.

Para isso algumas ferramentas são sugeridas para serem utilizadas no projeto:

### Infraestrutura do Projeto: Tecnologias utilizadas



## 1. Preparação do ambiente no Sistema Operacional Linux

### 1.1 Faça download do Anaconda no site:

Realizar o download do arquivo .sh

<https://www.anaconda.com/products/individual#Downloads>

No terminal Linux realizar a instalação do Anaconda a partir do arquivo .sh  
<https://docs.anaconda.com/anaconda/install/linux/>

- For Python 3.7, enter the following:

```
# Include the bash command regardless of whether or not you are using the Bash shell
bash ~/Downloads/Anaconda3-2020.05-Linux-x86_64.sh
# Replace ~/Downloads with your actual path
# Replace the .sh file name with the name of the file you downloaded
```

Após instalar o Anaconda, criar o atalho na área de trabalho.

<https://www.joelotz.com/blog/2021/creating-a-launch-shortcut-for-anaconda-on-ubuntu.html>

<https://linux.how2shout.com/create-anaconda-navigator-desktop-shortcut-ubuntu-20-04-18-04/>

## 1.2 Faça download do Visual Studio Code (baixar o pacote .debian)

Construção do APP para deploy

<https://code.visualstudio.com/download>

## 1.3 Instalação e Configuração do SQLite e SQLiteStudio/DBBrowser for SQLite

BD para fazer consultas SQL na base de dados

<https://www.sqlite.org/download.html>

### 1.3.1 Primeiro instale o SQLite via terminal

<https://cloudinfrastructureservices.co.uk/how-to-install-db-browser-for-sqlite-in-ubuntu-server-20-04/>

### 1.3.2 Instalação do SQLiteStudio

<https://sqlitestudio.pl/>

Baixar o arquivo com extensão .run :

<https://github.com/pawelsalawa/sqlitestudio/releases/download/3.4.4/SQLiteStudio-3.4.4-linux-x64-installer.run>

Ir nas propriedades do arquivo e marcar o arquivo como um programa  
pesquisar : install file .run ubuntu

Duplo click no arquivo para iniciar a instalação.

### **1.3.3. Instalação do DB Browser for SQLite**

O DB Browser for SQLite pode ser instalado através do repositório do Linux ou através do link abaixo.

<https://github.com/pawelsalawa/sqlitestudio/releases/download/3.4.4/sqlitestudio-3.4.4.tar.xz>

## **1.4 Instalação e Configuração do Apache Airflow**

<https://airflow.apache.org/>

Utilizado para fazer os processos de orquestração, automação, ETL

Realizar a instalação do Apache Airflow no prompt do Anaconda.

```
pip install apache-airflow
```

Navegue até o diretório airflow e crie o diretório dags.

Criar o arquivo: data\_pipeline.py e depois copiá-lo para dentro do diretório dags

## **1.5 - Git e GitHub**

publicar o App no repositório

Criar uma conta no github: <https://github.com/>

## **1.6 Streamlit**

Tecnologia utilizada para criar data Apps

<https://streamlit.io/>

Realizar a instalação do streamlit no prompt do Anaconda

Ir no diretório do anaconda, dentro da pasta bin do anaconda instalar o streamlit

```
./pip3 install streamlit
```

```
pip install streamlit
```

Realizar o cadastro na plataforma streamlit para permitir publicar o data app na web.

<https://share.streamlit.io/>

Logar com a conta do github para conectar o github com o streamlit.

O streamlit vai buscar seus fontes no github

Construindo a aplicação de ponta a ponta e realizando o deploy:

Selecione uma base de dados para o seu projeto ou crie uma a partir do zero.

Fonte para bases de dados:

<https://codingdatatoday.co/blog/os-10-melhores-sites-para-encontrar-datasets/#:~:text=O%20DataHub%20%C3%A9%20uma%20plataforma,compartilhar%20e%20colaborar%20em%20dados.>

- 1) (1,0 pontos) Crie uma aplicação (CRUD) para manipular esta base de dados, que deverá ser armazenada em um banco de dados (SQLite, MySQL, MongoDB, MariaDB, etc...). Sua aplicação pode ser desenvolvida em: Delphi, WEB (php, Javascript, Flask, Django, etc...) ou um script python para alimentar a base de dados em produção de forma automatizada.
- 2) (1,0 pontos) Utilize uma ferramenta para orquestração, manipulação dos dados e agendamento, ou seja, acessar os dados em produção e realizar uma cópia destes dados no formato .csv, .json, .xml, etc...  
Sugestão de frameworks para orquestração e manipulação de dados: Apache Airflow. Outras soluções disponíveis no mercado: <https://hevodata.com/learn/airflow-alternatives/>

- a. Implemente um script Python para realizar a ETL dos dados. Com a utilização do framework de manipulação e orquestração dos dados gere uma base de dados a ser consumida por um algoritmo de Machine Learning.
- 3) (1,0 pontos) O algoritmo de Machine Learning deverá ser construído com base nos dados existentes, esse algoritmo deverá ser atualizado à medida que novas informações são manipuladas na base de dados em produção.
- a. Implemente um script Python para realizar a construção do modelo de Machine Learning. Este script deverá ficar sendo executado no framework de orquestração (Ex: Apache Airflow).
- 4) (1,0 pontos) Implemente um DATA APP para realizar o deploy do seu modelo de Machine Learning desenvolvido. Sugestão de framework para deploy de modelos de Machine Learning (<https://www.cienciaedados.com/gradio-vs-streamlit-vs-dash-vs-flask-deploy-de-modelos-de-machine-learning-em-python/>)
- a. Hospede sua aplicação (Data APP e modelo de Machine Learning) no GitHub ou em outra plataforma de hospedagem. Como o Machine Learning deve ser atualizado à medida que novos dados são inseridos na base de dados, construa um script em python para realizar a atualização desse modelo em nuvem. Sugestão de biblioteca Python: GitPython (<https://towardsdatascience.com/updating-your-github-repository-using-python-a5c283a011d7>)

Considerações Finais:

- a) O grupo deverá ter no máximo 2 integrantes, podendo ser desenvolvido de forma individual se assim desejar.
- b) Procure ser auxiliado por uma IA Gerativa (ChatGPT, Gemini, Copilot, etc..) para a construção da aplicação.