

# Exercício em Dupla: Análise Colaborativa de Acidentes de Trânsito no Brasil

Hub de Inteligência Artificial - Turma 7

14/11/2024

## 1 Introdução

Neste exercício, vocês irão trabalhar em duplas para analisar dados de acidentes de trânsito no Brasil dos anos 2021 a 2024. Além de praticar o processamento de dados e visualização, também irão utilizar o Git para colaborar ao longo do desenvolvimento do projeto.

## 2 Objetivos

- Colaborar em duplas utilizando o Git para controle de versão.
- Baixar, preparar e validar os dados dos arquivos CSV.
- Realizar análises exploratórias e engenharia de atributos nos dados.
- Desenvolver uma aplicação interativa com o Streamlit, incluindo visualizações gráficas.
- Praticar o fluxo de trabalho colaborativo em um ambiente compartilhado.

## 3 Instruções

### 3.1 Configuração Inicial

- Criem um repositório **privado** no GitHub e adicionem seu parceiro de equipe como colaborador.
- Certifiquem-se de que ambos os membros têm acesso completo ao repositório.
- Adicionem os seguintes colaboradores para avaliação:

```
https://github.com/leticiacechinel
https://github.com/LeleoSanches
https://github.com/pedarias
https://github.com/juicynir
```

## 3.2 Configuração do Ambiente

- Criem um ambiente virtual para o projeto:

```
conda create -n nome_ambiente python=3.x
ou
python -m venv venv
source venv/bin/activate
```

- Instalem o Python 3.x e as bibliotecas necessárias:

- pandas
- numpy
- streamlit
- plotly
- matplotlib
- seaborn
- Outras que julgarem necessárias.

- Criem o arquivo `requirements.txt` com as dependências ou exporte em `yaml`:

```
conda env export > environment.yml

pip freeze > requirements.txt
```

- Façam commit do `requirements.txt` ou `environment.yml`:

```
git add requirements.txt
git commit -m "Adiciona arquivo de dependências"
git push
```

### 3.3 Download e Preparação dos Dados

- Baixem os arquivos `2021.csv`, `2022.csv`, `2023.csv` e `2024.csv`.
- Carreguem os arquivos CSV usando `pandas.read_csv()`.
- Concatem os `DataFrames` para criar um único conjunto de dados abrangendo todos os anos.
- Salvem o conjunto de dados consolidado.

### 3.4 Validação e Limpeza dos Dados

- Verifiquem a existência de valores ausentes (nulos) e tratem-nos:
  - Para colunas numéricas, considerem substituir valores nulos pela média ou mediana.
  - Para colunas categóricas, considerem substituir pela moda.
- Assegurem que as colunas de data estejam no formato `datetime`.
- Identifiquem e removam duplicatas ou registros incoerentes.

### 3.5 Engenharia de Atributos

- Criem novas colunas relevantes para a análise:
  - Extração de informações de data: dia da semana, mês, ano, período do dia (`PERIODO_DIA`).
  - Cálculo do total de feridos (`TOTAL_FERIDOS`) combinando feridos leves e graves.
  - Outras variáveis que considerarem relevantes.
- Explore a criação de categorias ou faixas para variáveis contínuas. Por exemplo, podemos extrair o horário dos acidentes, que seria uma coluna com valores contínuos de 0 a 23 e então classificar em categorias como `MADRUGADA`, `MANHÃ`, `TARDE` E `NOITE`.

### 3.6 Planejamento e Distribuição de Tarefas

- Dividam as responsabilidades:
  - **Pessoa A:** Focará na preparação, limpeza e engenharia dos dados.
  - **Pessoa B:** Desenvolverá a interface e as visualizações no Streamlit.
- Criem branches separadas:

```
git checkout -b dados_preparacao          # Pessoa A
git checkout -b desenvolvimento_interface # Pessoa B
```

### 3.7 Desenvolvimento Individual

**Para Pessoa A (Dados):**

- Realizem todas as etapas de preparação e limpeza dos dados.
- Garantam que o conjunto de dados final esteja pronto para as análises.
- Façam commits frequentes com mensagens claras:

```
git add dados.py
git commit -m "Limpeza e preparação dos dados concluída"
git push origin dados_preparacao
```

**Para Pessoa B (Interface):**

- Configurem o `app.py` para a aplicação Streamlit.
- Desenvolvam a interface, incluindo seções para cada tipo de análise.
- Implementem elementos interativos como filtros e seletores.
- Façam commits frequentes com mensagens claras:

```
git add app.py
git commit -m "Interface básica criada"
git push origin desenvolvimento_interface
```

### 3.8 Integração e Colaboração

- Reúnam-se para integrar as alterações:

```
git checkout main
git pull
git merge dados_preparacao
git merge desenvolvimento_interface
```

- Resolvam conflitos que possam surgir.
- Testem a aplicação conjunta para garantir que os dados são corretamente incorporados à interface.

### 3.9 Análises Exploratórias

- Realizem análises para compreender a distribuição dos dados:
  - Histogramas e boxplots das principais variáveis.
  - Matriz de correlação entre variáveis numéricas.
  - Etc.
- Documentem insights relevantes obtidos durante a exploração.

### 3.10 Desenvolvimento das Visualizações

- Implementem as visualizações solicitadas:
  - **Gráfico de barras:** Top 5 causas mais comuns dos acidentes. E top 5 tipos mais comuns de acidentes.
  - **Mapa interativo:** Distribuição geográfica dos acidentes.
  - **Gráfico de dispersão:** Relação entre número de vítimas e condições do tempo. Por exemplo, ocorrem mais mortes em acidentes quando está chovendo, céu claro, neblina, etc.
  - **Gráficos adicionais:** Que considerem pertinentes.
- Utilizem bibliotecas como `plotly` para gráficos interativos.
- Adicionem filtros e controles na interface para que o usuário possa interagir com os dados. Filtrar os dados por ano por exemplo.

### 3.11 Refinamento da Aplicação

- Melhorem o layout e a usabilidade da aplicação.
- Adicionem explicações e legendas que facilitem o entendimento das visualizações.

### 3.12 Documentação e Finalização

- Atualizem o `README.md` com:
  - Descrição do projeto.
  - Instruções de instalação e execução.
  - Principais resultados e insights obtidos.
- Verifiquem se todas as dependências estão listadas em `requirements.txt`.
- Realizem os commits finais:

```
git add .
git commit -m "Documentação atualizada e refinamentos finais"
git push
```

## 4 Entrega

- Certifiquem-se de que ambos contribuíram efetivamente para o projeto.
- O Projeto deve conter os códigos, o ambiente virtual e read.md.
- Dúvidas:

leonardo.sanches@sistemafiep.org.br  
pedro.oliveira@sistemafiep.org.br  
leticia.cechinel@sistemafiep.org.br

- Prazo Final: 29/11/2024 - 12:00h;

**\*\*Obs\*\*:** Não se prenam as análises que estão neste arquivo. Sejam criativos! Boa sorte e bom trabalho!