**Documentação do Tratamento de Dados**

Este documento detalha as etapas de tratamento de dados aplicadas a um DataFrame.

1. Carregamento do DataFrame:

O DataFrame é carregado a partir de um arquivo CSV chamado 'dados\_nao\_tratados.csv'.

2. Tratamento da Coluna ativo

Foram encontrados valores inconsistentes na coluna ativo (ex: “sim”, “Sim”, “Yes”, “nao”, “FALSE”, etc.).

A solução foi mapear todos os valores para booleanos (True ou False) e None quando inválido.  
 df["ativo"] = df["ativo"].str.strip().map({

'sim': True,

'Não':False,

'Sim': True,

'TRUE': True,

'TRUE': True, 'e': None,

'FALSE': False,

'nao': False,

'Yes': True

})

3. Tratamento da Coluna nota

A coluna nota apresentava valores inválidos, como "dez" e "e". Foi aplicado o seguinte tratamento: - Substituir "dez" por 10. - Converter para numérico, transformando inválidos em NaN.

df.replace('dez', 10, inplace=True)

df['nota'] = pd.to\_numeric(df['nota'], errors='coerce')

4. Tratamento da Coluna data\_inscricao

Algumas datas estavam em formatos inválidos ou impossíveis (ex: 2023/13/01, 2022-08-XX, 2023-02-30).

A conversão foi feita para datetime, com erros transformados em NaT.

df['data\_inscricao'] = pd.to\_datetime(df['data\_inscricao'], errors='coerce')

5. Tratamento da Coluna idade

Na coluna idade, havia valores como "vinte e três" e "?". Foi feito: - Substituição de "vinte e três" por 23. - Conversão para numérico, inválidos viraram NaN.

df.replace('vinte e três', 23, inplace=True)

df["idade"] = pd.to\_numeric(df["idade"], errors="coerce")

Resultado Final

Após o tratamento:

- ativo foi padronizado para booleano (True/False).

- nota contém apenas valores numéricos.

- data\_inscricao está no formato datetime com inválidos como NaT.

- idade contém apenas valores numéricos (int ou NaN).