

## 1. Tratamento de Nível de Senioridade (Limpeza Categórica)

Para começar, importei as bibliotecas `pandas` (para mexer na tabela) e `numpy` (para trabalhar com valores nulos).

- **Identificação e Verificação:**
  - Utilizei `print(dados['Nivel_Senioridade'].unique())` para ver todos os formatos diferentes de cargo.
- **Padronização (`.replace()`):**
  - Criei o dicionário `mapeamento_senioridade` e utilizei o comando `.replace()` para garantir que `Jr`, `JR`, `P`, `pleno`, `Senior` e `senior` virassem as categorias padrão: `'Júnior'`, `'Pleno'` e `'Sênior'`.
- **Tratamento de Nulos (`.fillna()`):**
  - Identifiquei os valores 'N/D' e mapeei para `np.nan`.
  - Calculei a moda (valor mais frequente) com `dados['Nivel_Senioridade'].mode()[0]`.
  - Utilizei `.fillna()` para preencher os valores nulos com a moda, garantindo que não ficasse buraco na coluna.

## 2. Tratamento de Avaliações (Numérico)

As notas precisavam ser numéricas para o cálculo, mas estavam formatadas com vírgulas e tinham nulos implícitos.

- **Preparação para Numérico:**
  - Utilizei o `.astype(str).str.replace(',', '.', '')` para trocar todas as vírgulas por pontos, o que permite a leitura como `float`.
  - O `pd.to_numeric()` finalizou a conversão para número.
- **Tratamento de Nulos (`.mean()` e `.fillna()`):**
  - Calculei a média (`.mean()`) de ambas as colunas.
  - Utilizei `.fillna()` para preencher os valores nulos com a respectiva média.
- **Padronização de Saída:**
  - Utilizei a função `.map(lambda x: f"{x:.1f}".replace('.', ','))` para retornar o formato com **uma casa decimal** e o separador **vírgula**.

### 3. Tratamento de Engajamento PIGs (Percentual)

Essa coluna estava em formato de texto e precisava virar um percentual limpo para cálculo.

- **Limpeza e Conversão:**
  - Removi o % com `.str.replace('%', '')`.
  - Troquei 'N/A', 'ND' e variações por `np.nan`.
  - Utilizei `pd.to_numeric()` para converter o texto para número.
- **Cálculo:**
  - Dividi a coluna por 100 (`dados['Engajamento_PIGs'] / 100`) para transformar 80 em 0.80.
- **Tratamento de Nulos:**
  - Calculei a média e utilizei `.fillna()` para preencher os buracos.
- **Padronização de Saída:**
  - Utilizei o `.map()` para formatar com **três casas decimais e vírgula**.

### 4. Enriquecimento: Criação do Score de Desempenho

Criei uma nova coluna consolidando as duas avaliações.

- **Cálculo da Média:**
  - Para o cálculo, precisei **voltar as colunas de nota para float temporariamente** usando `.str.replace(',', '.').astype(float)`.
  - Calculei a média e atribuí o resultado à coluna `Score_Desempenho`.
- **Padronização de Saída:**
  - Apliquei o `.map()` novamente para formatar o score com **uma casa decimal e vírgula**.
- **Verificação:**
  - Utilizei 

```
print(dados[['Avaliacao_Tecnica', 'Avaliacao_Comportamental', 'Score_Desempenho']].head())
```

 para conferir se o cálculo estava certo.

## 5. Enriquecimento: Classificação de Status

Criei a lógica para classificar os membros como "Em Destaque" ou "Padrão".

- **Lógica Booleana:**
  - Criei colunas auxiliares `_float` para o score e engajamento, permitindo a comparação numérica.
  - Criei a condição lógica: `(Score >= 7.0) E (Engajamento >= 0.8)`.
  - Utilizei o comando `np.where()` para aplicar a classificação: se a condição for verdadeira, atribui 'Em Destaque'; se for falsa, 'Padrão'.
- **Limpeza Final:**
  - Utilizei `dados.drop(..., axis=1, inplace=True)` para remover as colunas `_float` que usei só para o cálculo.

## 6. Exportação Final

- **Excel:** Usei o comando `.to_excel()` para salvar a base limpa e padronizada no formato Excel.
- **PDF (Bônus Visual):** Utilizei `matplotlib.pyplot` para gerar uma imagem da tabela e salvei em PDF com `plt.savefig()`, garantindo que todo o *dataset* fosse visualizado.