

Pacotes utilizados

Este bloco importa os pacotes essenciais utilizados ao longo do projeto:

- `pandas` : usado para carregar, transformar e analisar os dados.
- `numpy` : fornece funções matemáticas de alto desempenho.
- `matplotlib.pyplot` : cria gráficos como histogramas, scatter plots e outros.
- `seaborn` : biblioteca baseada no matplotlib, ideal para visualizações estatísticas aprimoradas.
- `math` : biblioteca para funções matemáticas
- `scipy.stats` : ermite realizar cálculos descritivos, testes de hipóteses, gerar distribuições probabilísticas e muito mais.

```
In [2]: import pandas as pd # Manipulação de dados tabulares
import numpy as np # Operações numéricas com arrays
import matplotlib.pyplot as plt # Geração de gráficos
import seaborn as sns # Visualização de dados com gráficos estatísticos
import math # Biblioteca padrão para funções matemáticas
from scipy.stats import t # Importa a distribuição t de Student para testes estatísticos (con
```

Importando as bases

Carregar as bases Microdados do enem dos anos de 2021, 2022 e 2023 em csv

```
In [4]: enem_2023 = pd.read_csv(r'C:\Users\rodri\Desktop\MICRODADOS_ENEM_2023.csv', sep=';', encoding='utf-8')
enem_2022 = pd.read_csv(r'C:\Users\rodri\Desktop\MICRODADOS_ENEM_2022.csv', sep=';', encoding='utf-8')
enem_2021 = pd.read_csv(r'C:\Users\rodri\Desktop\MICRODADOS_ENEM_2021.csv', sep=';', encoding='utf-8')
```

Selecionar colunas específicas

Este bloco seleciona as colunas 'TP_COR_RACA', 'NU_ANO', 'TP_ESCOLA', 'NO_MUNICIPIO_PROVA', 'NU_NOTA_MT', 'NU_NOTA_REDACAO', 'NU_NOTA_LC', 'NU_NOTA_CH', 'NU_NOTA_CN' que serão utilizadas no projeto

```
In [6]: enem_2023 = enem_2023[['TP_COR_RACA', 'NU_ANO', 'TP_ESCOLA', 'NO_MUNICIPIO_PROVA',
                                'NU_NOTA_MT', 'NU_NOTA_REDACAO', 'NU_NOTA_LC', 'NU_NOTA_CH', 'NU_NOTA_CN']]

enem_2022 = enem_2022[['TP_COR_RACA', 'NU_ANO', 'TP_ESCOLA', 'NO_MUNICIPIO_PROVA',
                        'NU_NOTA_MT', 'NU_NOTA_REDACAO', 'NU_NOTA_LC', 'NU_NOTA_CH', 'NU_NOTA_CN']]

enem_2021 = enem_2021[['TP_COR_RACA', 'NU_ANO', 'TP_ESCOLA', 'NO_MUNICIPIO_PROVA',
                        'NU_NOTA_MT', 'NU_NOTA_REDACAO', 'NU_NOTA_LC', 'NU_NOTA_CH', 'NU_NOTA_CN']]
```

Substituir os números da coluna TP_ESCOLA pelo respectivo nome e Remover linhas não necessárias

Este bloco faz o mapeamento correto da coluna TP_ESCOLA dos números pelo seu nome, para facilitar o entendimento do projeto. Além disso, remove as linhas onde o tipo de escola está como não respondida ou de quem estuda no exterior

```
In [8]: mapeamento = {
        1: 'Não respondeu',
        2: 'Escola pública',
```

```

3: 'Privada',
4: 'Exterior'
}

# Aplica o mapeamento e remove 'Não respondeu' com atribuição correta
enem_2023['TP_ESCOLA'] = enem_2023['TP_ESCOLA'].map(mapeamento)
enem_2023 = enem_2023[enem_2023['TP_ESCOLA'] != 'Não respondeu']
enem_2023 = enem_2023[enem_2023['TP_ESCOLA'] != 'Exterior']

enem_2022['TP_ESCOLA'] = enem_2022['TP_ESCOLA'].map(mapeamento)
enem_2022 = enem_2022[enem_2022['TP_ESCOLA'] != 'Não respondeu']
enem_2022 = enem_2022[enem_2022['TP_ESCOLA'] != 'Exterior']

enem_2021['TP_ESCOLA'] = enem_2021['TP_ESCOLA'].map(mapeamento)
enem_2021 = enem_2021[enem_2021['TP_ESCOLA'] != 'Não respondeu']
enem_2021 = enem_2021[enem_2021['TP_ESCOLA'] != 'Exterior']

```

Substituir os números da coluna TP_COR_RACA pelo respectivo nome e Remover linhas não necessárias

Este bloco faz o mapeamento correto da coluna TP_COR_RACA dos números pelo seu nome, para facilitar o entendimento do projeto. Além disso, remove as linhas onde o tipo de cor/raça está como não declarado

```

In [10]: mapeamento_raca = {
        0: 'não declarado',
        1: 'branca',
        2: 'preta',
        3: 'parda',
        4: 'amarela',
        5: 'indígena',
        6: 'não dispõe da informação'
    }

# aplica o mapeamento
enem_2023['TP_COR_RACA'] = enem_2023['TP_COR_RACA'].map(mapeamento_raca)
enem_2023 = enem_2023[enem_2023['TP_COR_RACA'] != 'não declarado']

enem_2022['TP_COR_RACA'] = enem_2022['TP_COR_RACA'].map(mapeamento_raca)
enem_2022 = enem_2022[enem_2022['TP_COR_RACA'] != 'não declarado']

enem_2021['TP_COR_RACA'] = enem_2021['TP_COR_RACA'].map(mapeamento_raca)
enem_2021 = enem_2021[enem_2021['TP_COR_RACA'] != 'não declarado']

```

Excluir linhas vazias

Este bloco exclui as linhas das tabelas onde são vazias para uma melhor análise do projeto

```

In [12]: enem_2023 = enem_2023.dropna()
        enem_2022 = enem_2022.dropna()
        enem_2021 = enem_2021.dropna()

```

Total de alunos que fizeram o ENEM

Este bloco calcula a quantidade de pessoas de escola particular e pública que fizeram o ENEM em 2021, 2022 e 2023 e depois mostra o total (particular + pública). Este cálculo irá servir para futuras análises.

```
In [14]: qtd_publica_2023 = (enem_2023['TP_ESCOLA'] == 'Escola pública').sum()
qtd_publica_2022 = (enem_2022['TP_ESCOLA'] == 'Escola pública').sum()
qtd_publica_2021 = (enem_2021['TP_ESCOLA'] == 'Escola pública').sum()

qtd_privada_2023 = (enem_2023['TP_ESCOLA'] == 'Privada').sum()
qtd_privada_2022 = (enem_2022['TP_ESCOLA'] == 'Privada').sum()
qtd_privada_2021 = (enem_2021['TP_ESCOLA'] == 'Privada').sum()

print("Total de alunos de escola pública 2023:", qtd_publica_2023)
print("Total de alunos de escola pública 2022:", qtd_publica_2022)
print("Total de alunos de escola pública 2021:", qtd_publica_2021)
print("-----")
print("Total de alunos de escola privada 2023:", qtd_privada_2023)
print("Total de alunos de escola privada 2022:", qtd_privada_2022)
print("Total de alunos de escola privada 2021:", qtd_privada_2021)

print("-----")

print("Total aluno enem 2023:", qtd_publica_2023 + qtd_privada_2023)
print("Total aluno enem 2022:", qtd_publica_2022 + qtd_privada_2022)
print("Total aluno enem 2021:", qtd_publica_2021 + qtd_privada_2021)
```

Total de alunos de escola pública 2023: 820141

Total de alunos de escola pública 2022: 746292

Total de alunos de escola pública 2021: 653335

Total de alunos de escola privada 2023: 218928

Total de alunos de escola privada 2022: 194062

Total de alunos de escola privada 2021: 176305

Total aluno enem 2023: 1039069

Total aluno enem 2022: 940354

Total aluno enem 2021: 829640

Cálculo das notas de corte

Este bloco calcula a média da nota de corte ampla concorrência da UFABC nos anos de 2023, 2022 e 2021

```
In [16]: #Disponível em: https://prograd.ufabc.edu.br/sisu/historico
nota_corte_2023 = 717.52
nota_corte_2022 = 720.24
nota_corte_2021 = 715.36

nota_corte_ufabc = (717.52 + 720.24 + 715.36) / 3
print(nota_corte_ufabc)
```

717.7066666666666

Cálculo Média BC&T

Este bloco tem uma função para calcular a média ponderada para entrar na UFABC no curso Bacharelado em Ciência e Tecnologia

```
In [18]: #Cálculo média BCT
def calcular_media_ponderada(row):
    return (
        row['NU_NOTA_LC'] * 1.0 +
        row['NU_NOTA_CH'] * 1.0 +
        row['NU_NOTA_CN'] * 1.5 +
        row['NU_NOTA_MT'] * 1.5 +
        row['NU_NOTA_REDACAO'] * 1.5
```

```

enem_2023['MEDIA_UFABC'] = enem_2023.apply(calcular_media_ponderada, axis=1)
enem_2022['MEDIA_UFABC'] = enem_2022.apply(calcular_media_ponderada, axis=1)
enem_2021['MEDIA_UFABC'] = enem_2021.apply(calcular_media_ponderada, axis=1)

```

Separação por tipo de escola

Este bloco está separando os dados do ENEM de cada ano (2021, 2022 e 2023) em dois grupos: -Alunos de escola pública -Alunos de escola privada

```

In [20]: #Separar por tipo de escola
publicos_2023 = enem_2023[enem_2023['TP_ESCOLA'] == 'Escola pública']
privados_2023 = enem_2023[enem_2023['TP_ESCOLA'] == 'Privada']

publicos_2022 = enem_2022[enem_2022['TP_ESCOLA'] == 'Escola pública']
privados_2022 = enem_2022[enem_2022['TP_ESCOLA'] == 'Privada']

publicos_2021 = enem_2021[enem_2021['TP_ESCOLA'] == 'Escola pública']
privados_2021 = enem_2021[enem_2021['TP_ESCOLA'] == 'Privada']

```

Análise de metas mínimas 2023, 2022 e 2021

Este bloco calcula a quantidade de pessoas de escolas públicas e privadas que obtiveram nota maior que 720 em Matemática, 650 em Linguagens, maior que a média de corte da UFABC e mostra o total de pessoas de escolas públicas e privadas em cada ano

```

In [22]: #Análise de metas mínimas 2023
privados_resultados_2023 = {
    "mat_720+": (privados_2023['NU_NOTA_MT'] >= 720).sum(),
    "ling_650+": (privados_2023['NU_NOTA_LC'] >= 650).sum(),
    "media_ufabc_nota_corte+": (privados_2023['MEDIA_UFABC'] >= nota_corte_ufabc).sum(),
    "total": len(privados_2023)
}
publicos_resultados_2023 = {
    "mat_720+": (publicos_2023['NU_NOTA_MT'] >= 720).sum(),
    "ling_650+": (publicos_2023['NU_NOTA_LC'] >= 650).sum(),
    "media_ufabc_nota_corte+": (publicos_2023['MEDIA_UFABC'] >= nota_corte_ufabc).sum(),
    "total": len(publicos_2023)
}

#Análise de metas mínimas 2022
privados_resultados_2022 = {
    "mat_720+": (privados_2022['NU_NOTA_MT'] >= 720).sum(),
    "ling_650+": (privados_2022['NU_NOTA_LC'] >= 650).sum(),
    "media_ufabc_nota_corte+": (privados_2022['MEDIA_UFABC'] >= nota_corte_ufabc).sum(),
    "total": len(privados_2022)
}
publicos_resultados_2022 = {
    "mat_720+": (publicos_2022['NU_NOTA_MT'] >= 720).sum(),
    "ling_650+": (publicos_2022['NU_NOTA_LC'] >= 650).sum(),
    "media_ufabc_nota_corte+": (publicos_2022['MEDIA_UFABC'] >= nota_corte_ufabc).sum(),
    "total": len(publicos_2022)
}

#Análise de metas mínimas 2021
privados_resultados_2021 = {
    "mat_720+": (privados_2021['NU_NOTA_MT'] >= 720).sum(),
    "ling_650+": (privados_2021['NU_NOTA_LC'] >= 650).sum(),
    "media_ufabc_nota_corte+": (privados_2021['MEDIA_UFABC'] >= nota_corte_ufabc).sum(),
    "total": len(privados_2021)
}

```

```

}
publicos_resultados_2021 = {
    "mat_720+": (publicos_2021['NU_NOTA_MT'] >= 720).sum(),
    "ling_650+": (publicos_2021['NU_NOTA_LC'] >= 650).sum(),
    "media_ufabc_nota_corte+": (publicos_2021['MEDIA_UFABC'] >= nota_corte_ufabc).sum(),
    "total": len(publicos_2021)
}

print("Resultados Escolas Privadas 2023:", privados_resultados_2023)
print("Resultados Escolas Públicas 2023:", publicos_resultados_2023)
print("-----")
print("Resultados Escolas Privadas 2022:", privados_resultados_2022)
print("Resultados Escolas Públicas 2022:", publicos_resultados_2022)
print("-----")
print("Resultados Escolas Privadas 2021:", privados_resultados_2021)
print("Resultados Escolas Públicas 2021:", publicos_resultados_2021)
print("-----")

```

Resultados Escolas Privadas 2023: {'mat_720+': 53952, 'ling_650+': 14035, 'media_ufabc_nota_corte+': 28151, 'total': 218928}

Resultados Escolas Públicas 2023: {'mat_720+': 26602, 'ling_650+': 6499, 'media_ufabc_nota_corte+': 7709, 'total': 820141}

Resultados Escolas Privadas 2022: {'mat_720+': 36635, 'ling_650+': 10628, 'media_ufabc_nota_corte+': 18390, 'total': 194062}

Resultados Escolas Públicas 2022: {'mat_720+': 22385, 'ling_650+': 5467, 'media_ufabc_nota_corte+': 6378, 'total': 746292}

Resultados Escolas Privadas 2021: {'mat_720+': 29884, 'ling_650+': 9403, 'media_ufabc_nota_corte+': 16859, 'total': 176305}

Resultados Escolas Públicas 2021: {'mat_720+': 16874, 'ling_650+': 4694, 'media_ufabc_nota_corte+': 6680, 'total': 653335}

Gráfico Distribuição Média Ponderada UFABC em 2023, 2022 e 2021

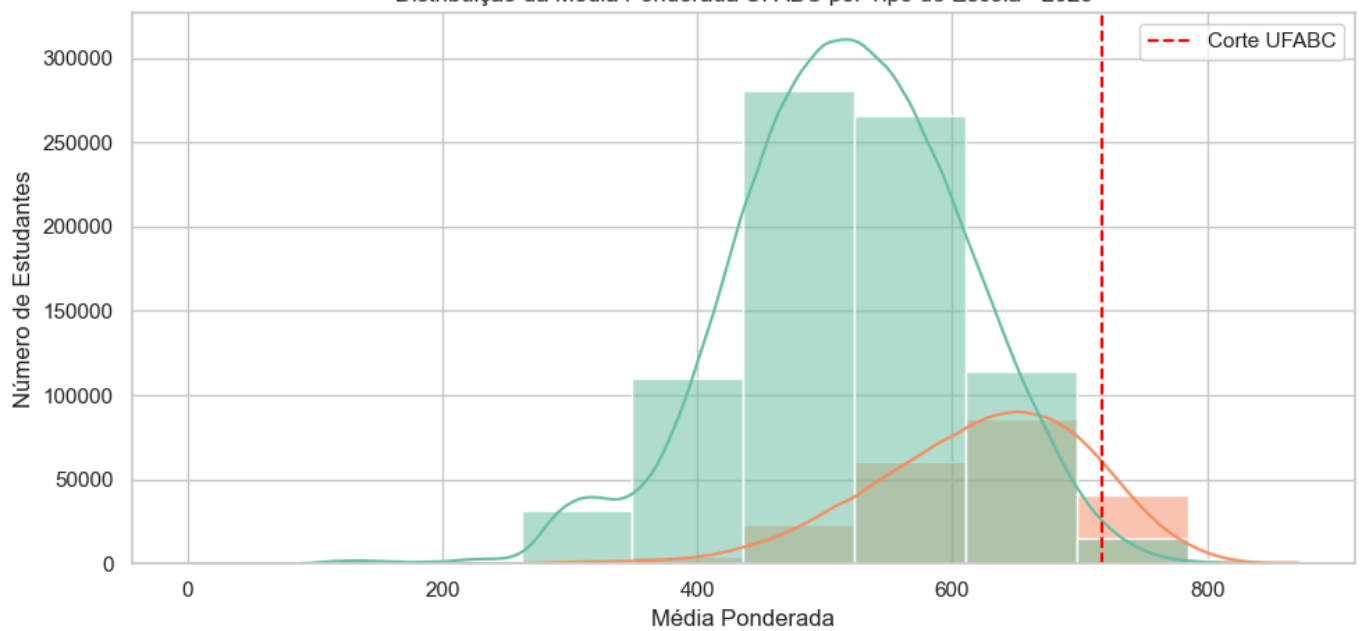
```

In [24]: sns.set(style="whitegrid")

plt.figure(figsize=(10, 5))
sns.histplot(data=enem_2023, x='MEDIA_UFABC', hue='TP_ESCOLA', kde=True, bins=10, palette='Set1')
plt.axvline(nota_corte_2023, color='red', linestyle='--', label='Corte UFABC')
plt.title('Distribuição da Média Ponderada UFABC por Tipo de Escola - 2023')
plt.xlabel('Média Ponderada')
plt.ylabel('Número de Estudantes')
plt.legend()
plt.tight_layout()
plt.show()

```

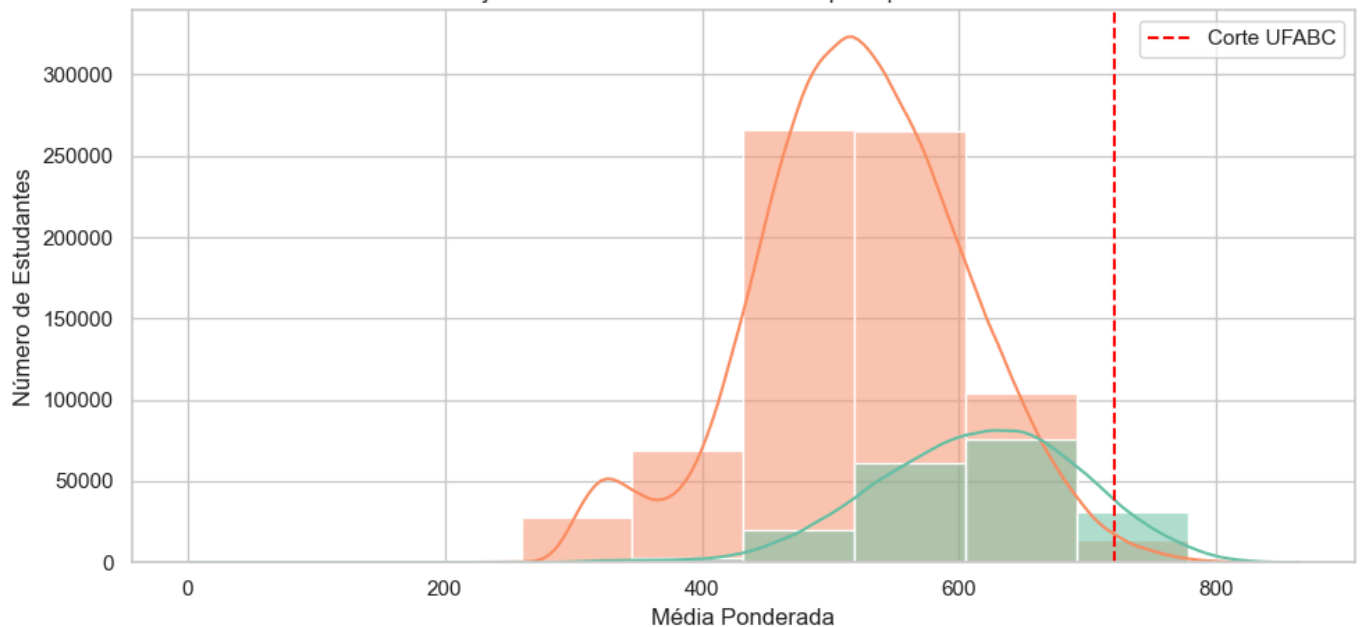
Distribuição da Média Ponderada UFABC por Tipo de Escola - 2023



```
In [25]: sns.set(style="whitegrid")

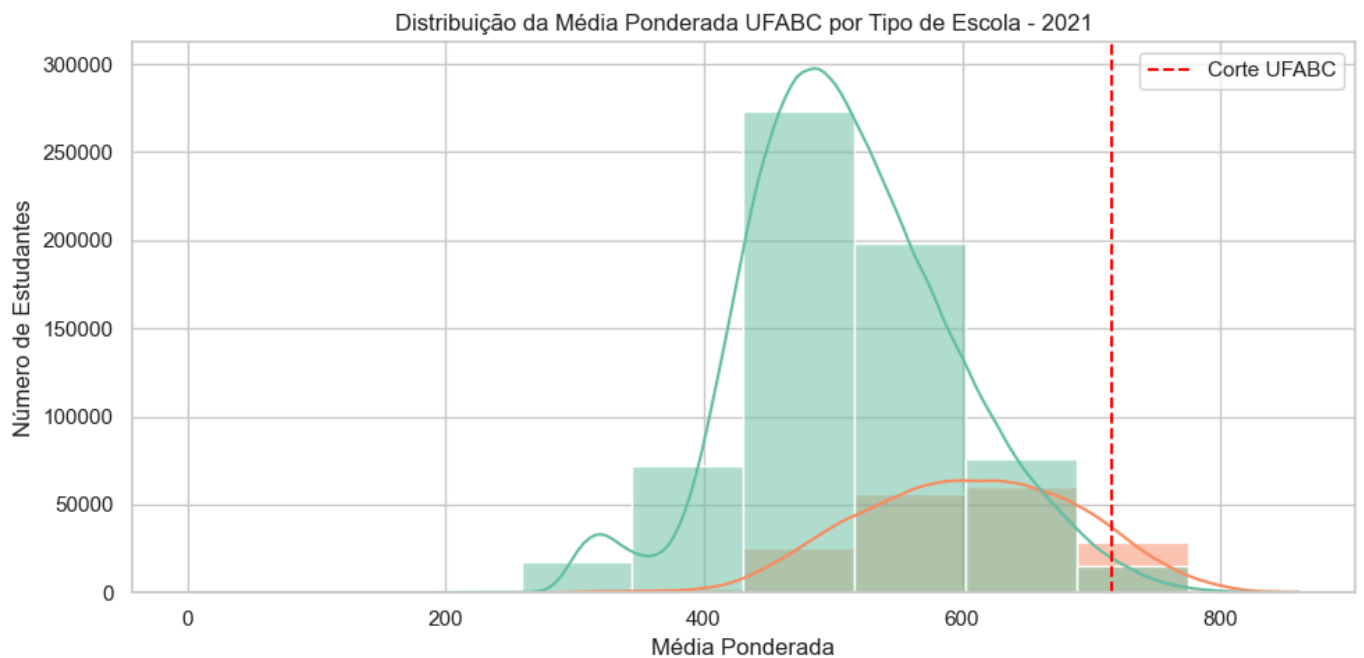
plt.figure(figsize=(10, 5))
sns.histplot(data=enem_2022, x='MEDIA_UFABC', hue='TP_ESCOLA', kde=True, bins=10, palette='Set1')
plt.axvline(nota_corte_2022, color='red', linestyle='--', label='Corte UFABC')
plt.title('Distribuição da Média Ponderada UFABC por Tipo de Escola - 2022')
plt.xlabel('Média Ponderada')
plt.ylabel('Número de Estudantes')
plt.legend()
plt.tight_layout()
plt.show()
```

Distribuição da Média Ponderada UFABC por Tipo de Escola - 2022



```
In [26]: sns.set(style="whitegrid")

plt.figure(figsize=(10, 5))
sns.histplot(data=enem_2021, x='MEDIA_UFABC', hue='TP_ESCOLA', kde=True, bins=10, palette='Set1')
plt.axvline(nota_corte_2021, color='red', linestyle='--', label='Corte UFABC')
plt.title('Distribuição da Média Ponderada UFABC por Tipo de Escola - 2021')
plt.xlabel('Média Ponderada')
plt.ylabel('Número de Estudantes')
plt.legend()
plt.tight_layout()
plt.show()
```



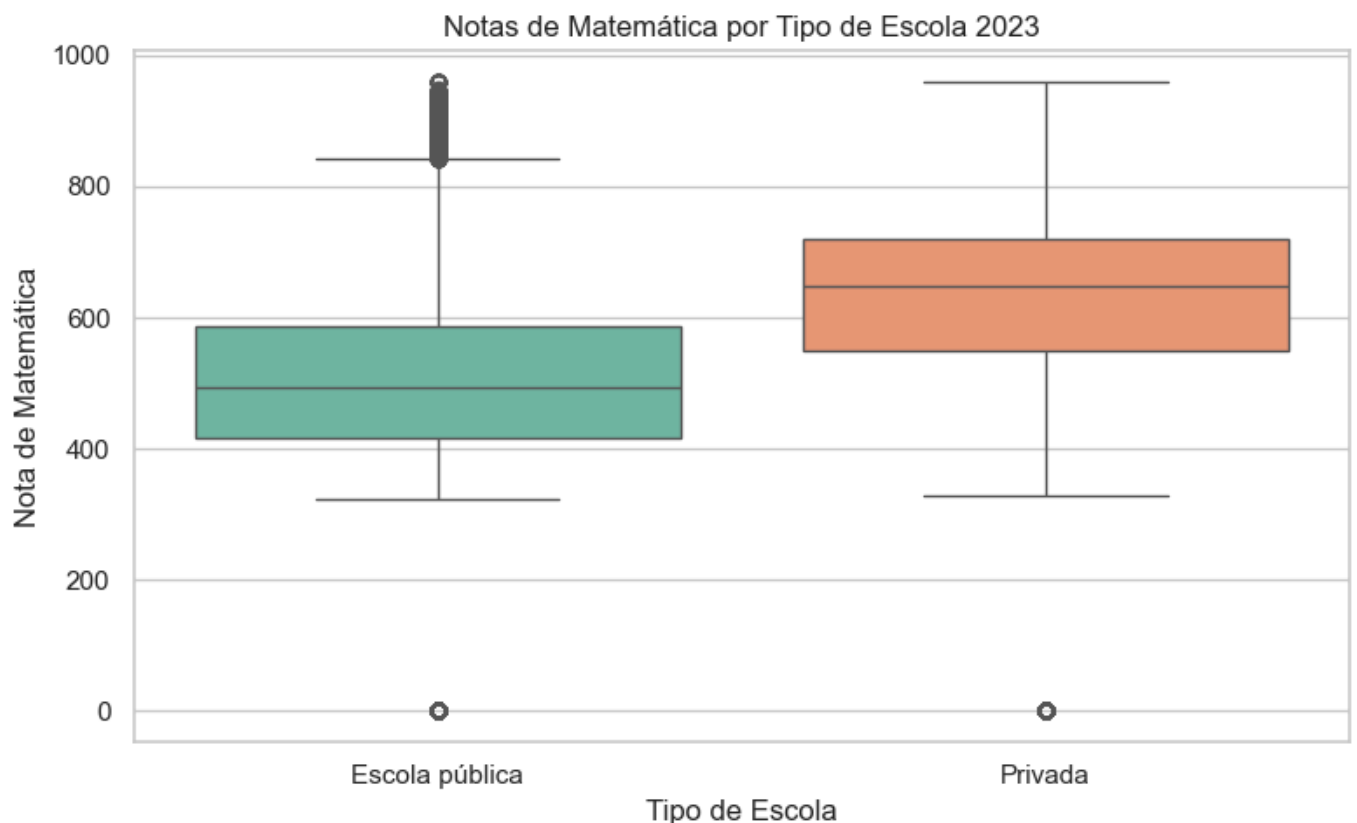
Boxplot das Notas de Matemática em 2023, 2022 e 2021

```
In [28]: plt.figure(figsize=(8, 5))
sns.boxplot(data=enem_2023, x='TP_ESCOLA', y='NU_NOTA_MT', palette='Set2')
plt.title('Notas de Matemática por Tipo de Escola 2023')
plt.ylabel('Nota de Matemática')
plt.xlabel('Tipo de Escola')
plt.tight_layout()
plt.show()
```

C:\Users\rodri\AppData\Local\Temp\ipykernel_6636\3440075739.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.boxplot(data=enem_2023, x='TP_ESCOLA', y='NU_NOTA_MT', palette='Set2')
```

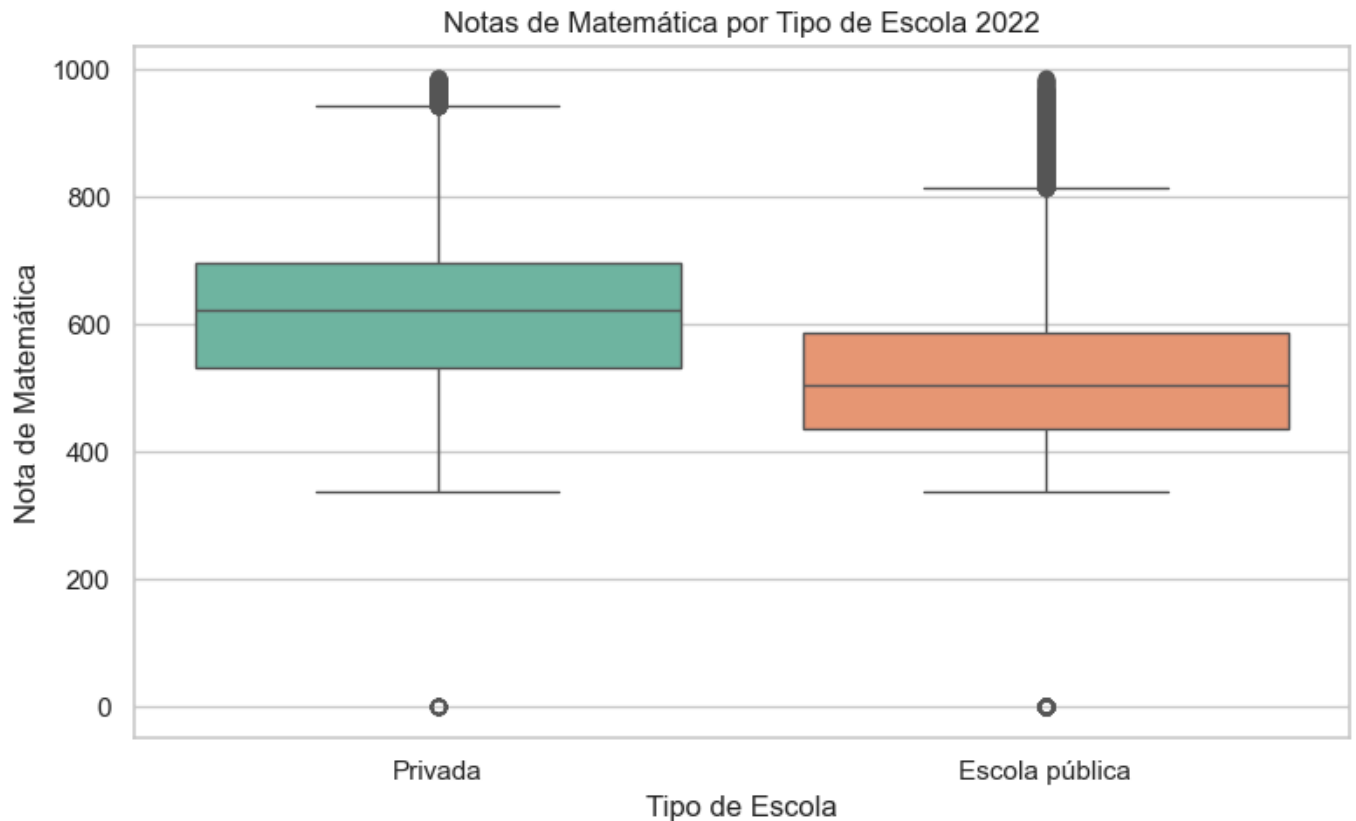


```
In [29]: plt.figure(figsize=(8, 5))
sns.boxplot(data=enem_2022, x='TP_ESCOLA', y='NU_NOTA_MT', palette='Set2')
plt.title('Notas de Matemática por Tipo de Escola 2022')
plt.ylabel('Nota de Matemática')
plt.xlabel('Tipo de Escola')
plt.tight_layout()
plt.show()
```

C:\Users\rodri\AppData\Local\Temp\ipykernel_6636\1300207972.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.boxplot(data=enem_2022, x='TP_ESCOLA', y='NU_NOTA_MT', palette='Set2')
```

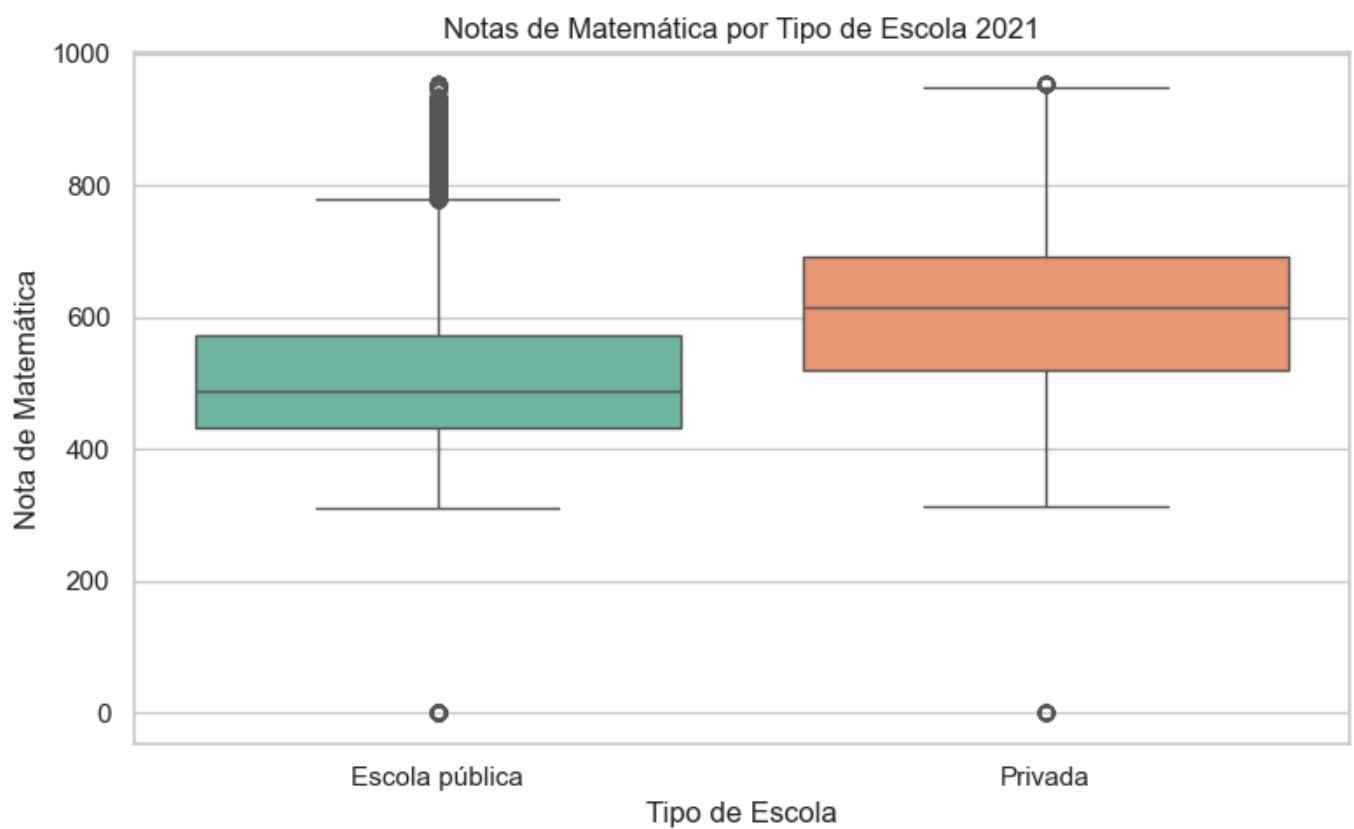


```
In [30]: plt.figure(figsize=(8, 5))
sns.boxplot(data=enem_2021, x='TP_ESCOLA', y='NU_NOTA_MT', palette='Set2')
plt.title('Notas de Matemática por Tipo de Escola 2021')
plt.ylabel('Nota de Matemática')
plt.xlabel('Tipo de Escola')
plt.tight_layout()
plt.show()
```

C:\Users\rodri\AppData\Local\Temp\ipykernel_6636\2201177412.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.boxplot(data=enem_2021, x='TP_ESCOLA', y='NU_NOTA_MT', palette='Set2')
```

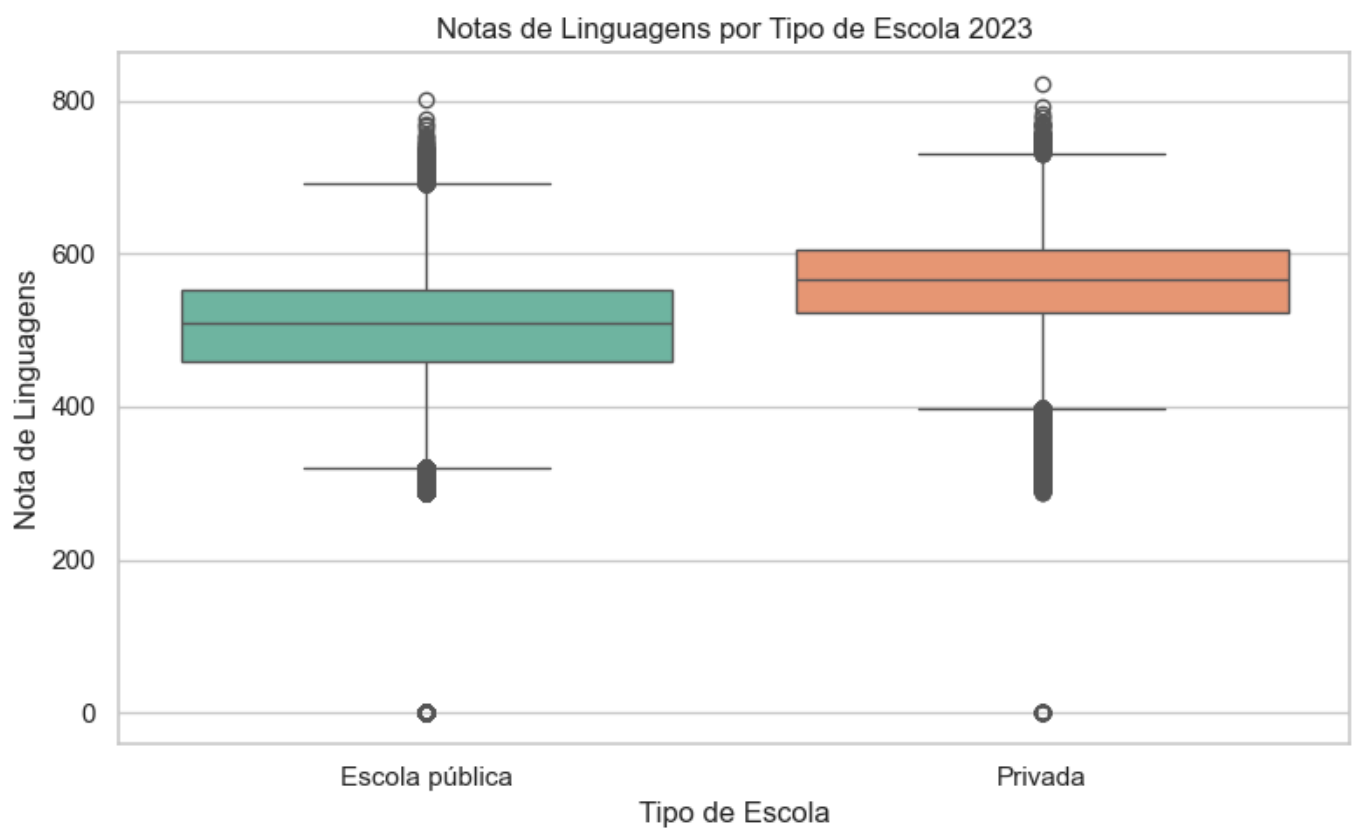
Boxplot das Notas de Linguagens em 2023, 2022 e 2021

```
In [32]: plt.figure(figsize=(8, 5))
sns.boxplot(data=enem_2023, x='TP_ESCOLA', y='NU_NOTA_LC', palette='Set2')
plt.title('Notas de Linguagens por Tipo de Escola 2023')
plt.ylabel('Nota de Linguagens')
plt.xlabel('Tipo de Escola')
plt.tight_layout()
plt.show()
```

C:\Users\rodri\AppData\Local\Temp\ipykernel_6636\2829749429.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.boxplot(data=enem_2023, x='TP_ESCOLA', y='NU_NOTA_LC', palette='Set2')
```

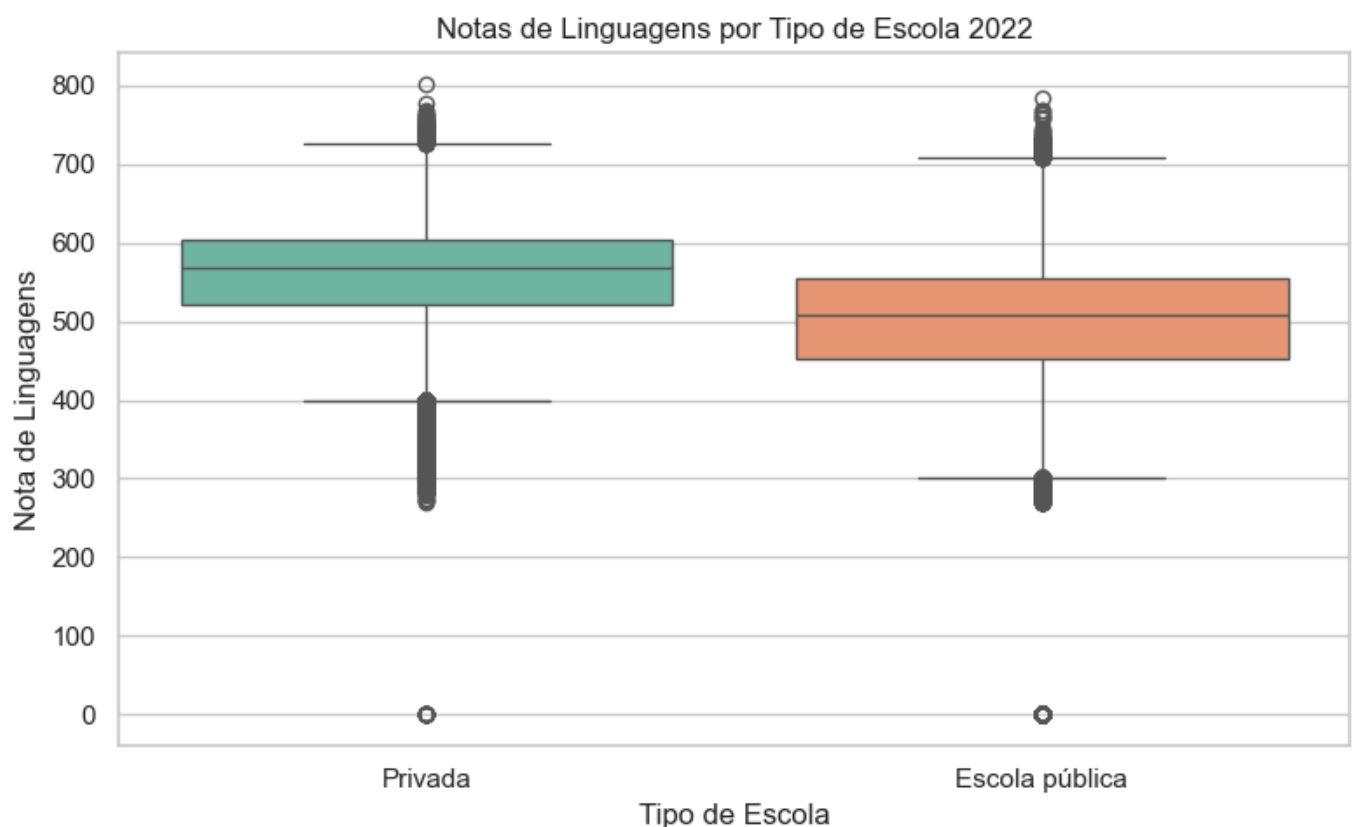


```
In [33]: plt.figure(figsize=(8, 5))
sns.boxplot(data=enem_2022, x='TP_ESCOLA', y='NU_NOTA_LC', palette='Set2')
plt.title('Notas de Linguagens por Tipo de Escola 2022')
plt.ylabel('Nota de Linguagens')
plt.xlabel('Tipo de Escola')
plt.tight_layout()
plt.show()
```

C:\Users\rodri\AppData\Local\Temp\ipykernel_6636\4189663982.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.boxplot(data=enem_2022, x='TP_ESCOLA', y='NU_NOTA_LC', palette='Set2')
```



```
In [34]: plt.figure(figsize=(8, 5))
sns.boxplot(data=enem_2021, x='TP_ESCOLA', y='NU_NOTA_LC', palette='Set2')
plt.title('Notas de Linguagens por Tipo de Escola 2021')
plt.ylabel('Nota de Linguagens')
plt.xlabel('Tipo de Escola')
plt.tight_layout()
plt.show()
```

C:\Users\rodri\AppData\Local\Temp\ipykernel_6636\3096516871.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.boxplot(data=enem_2021, x='TP_ESCOLA', y='NU_NOTA_LC', palette='Set2')
```

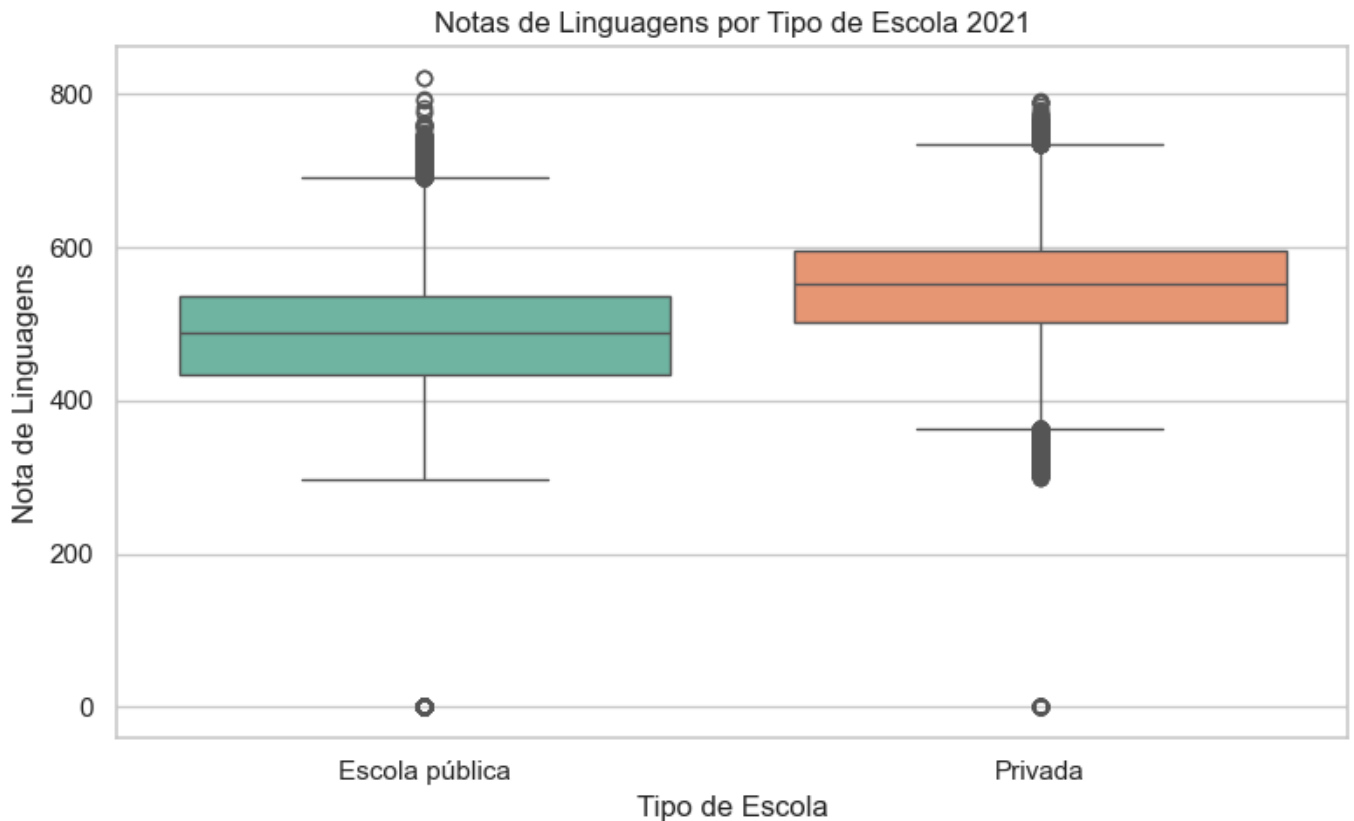


Gráfico de Barras – Proporção de Alunos com Média UFABC $\geq$ Média Corte em 2023, 2022 e 2021

```
In [36]: #2023
counts = pd.DataFrame({
    "Acima da media_corte": [
        (publicos_2023['MEDIA_UFABC'] >= nota_corte_2023).sum(),
        (privados_2023['MEDIA_UFABC'] >= nota_corte_2023).sum()
    ],
    "Total": [len(publicos_2023), len(privados_2023)]
}, index=["Escola Pública", "Privada"])

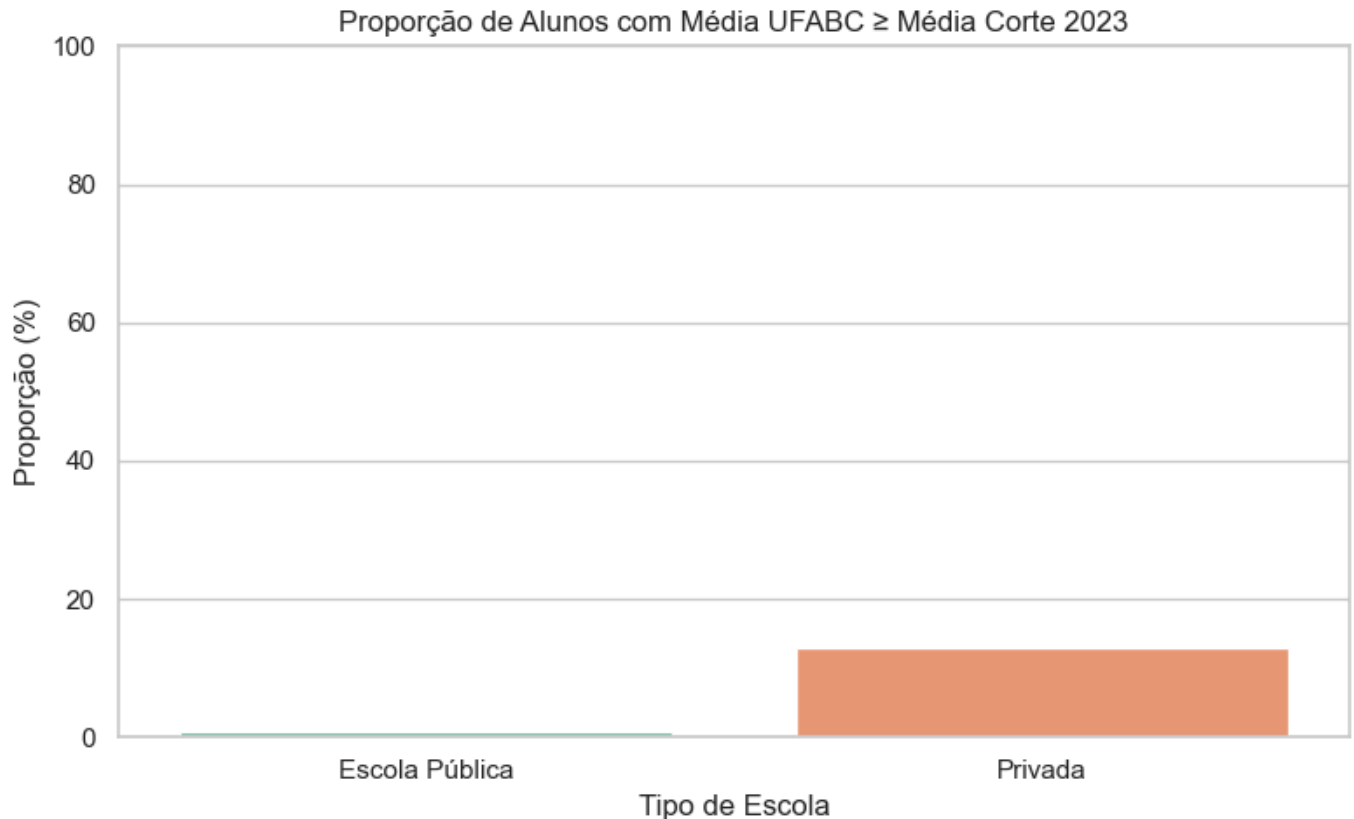
counts["Proporção (%)"] = (counts["Acima da media_corte"] / counts["Total"]) * 100

plt.figure(figsize=(8, 5))
sns.barplot(x=counts.index, y=counts["Proporção (%)"], palette='Set2')
plt.title('Proporção de Alunos com Média UFABC  $\geq$  Média Corte 2023')
plt.ylabel('Proporção (%)')
plt.xlabel('Tipo de Escola')
plt.ylim(0, 100)
plt.tight_layout()
plt.show()
```

C:\Users\rodri\AppData\Local\Temp\ipykernel_6636\2865050533.py:13: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x=counts.index, y=counts["Proporção (%)"], palette='Set2')
```



In [37]: #2022

```
counts = pd.DataFrame({
    "Acima da media_corte": [
        (publicos_2022['MEDIA_UFABC'] >= nota_corte_2022).sum(),
        (privados_2022['MEDIA_UFABC'] >= nota_corte_2022).sum()
    ],
    "Total": [len(publicos_2022), len(privados_2022)]
}, index=["Escola Pública", "Privada"])

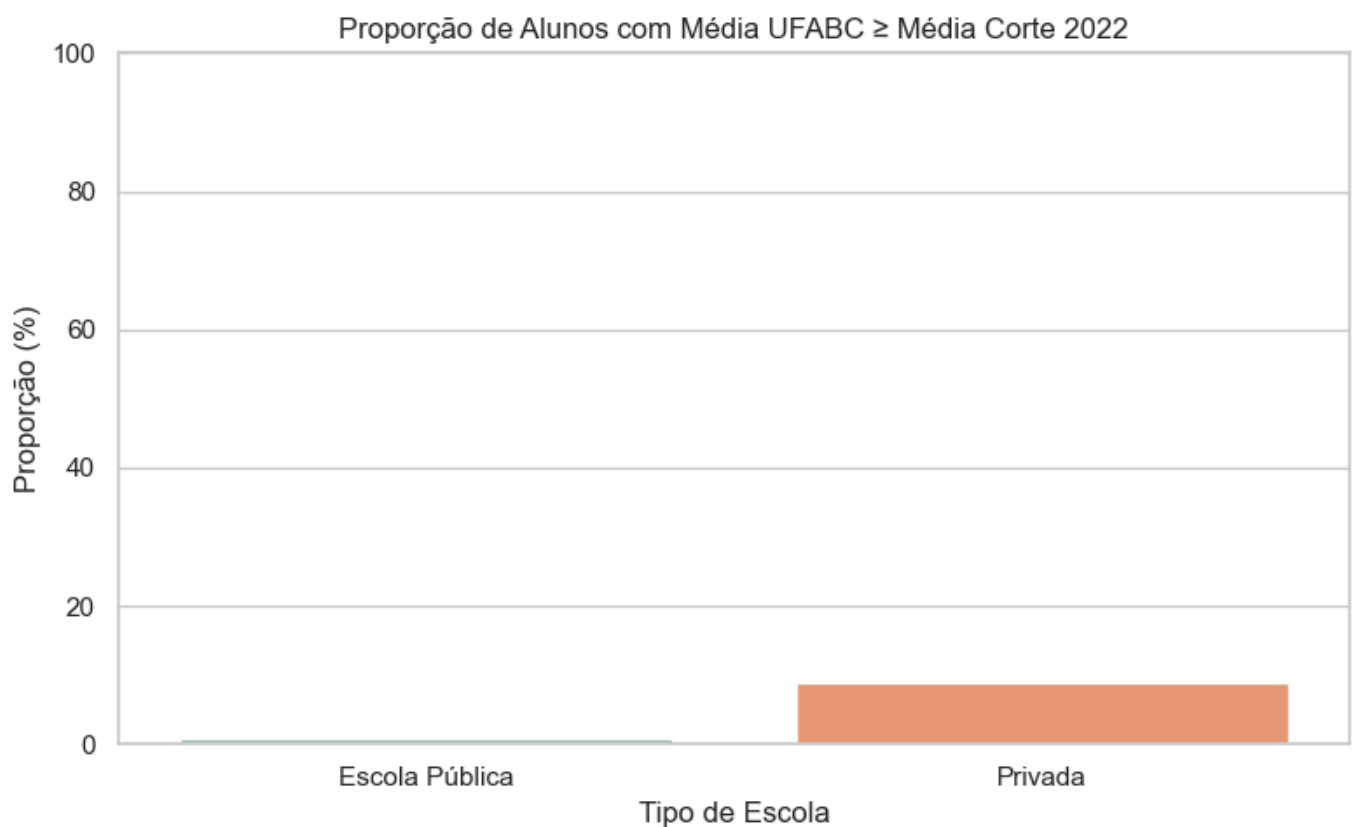
counts["Proporção (%)"] = (counts["Acima da media_corte"] / counts["Total"]) * 100

plt.figure(figsize=(8, 5))
sns.barplot(x=counts.index, y=counts["Proporção (%)"], palette='Set2')
plt.title('Proporção de Alunos com Média UFABC  $\geq$  Média Corte 2022')
plt.ylabel('Proporção (%)')
plt.xlabel('Tipo de Escola')
plt.ylim(0, 100)
plt.tight_layout()
plt.show()
```

C:\Users\rodri\AppData\Local\Temp\ipykernel_6636\2089402765.py:14: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x=counts.index, y=counts["Proporção (%)"], palette='Set2')
```



In [38]: #2021

```
counts = pd.DataFrame({
    "Acima da media_corte": [
        (publicos_2021['MEDIA_UFABC'] >= nota_corte_2021).sum(),
        (privados_2021['MEDIA_UFABC'] >= nota_corte_2021).sum()
    ],
    "Total": [len(publicos_2021), len(privados_2021)]
}, index=["Escola Pública", "Privada"])

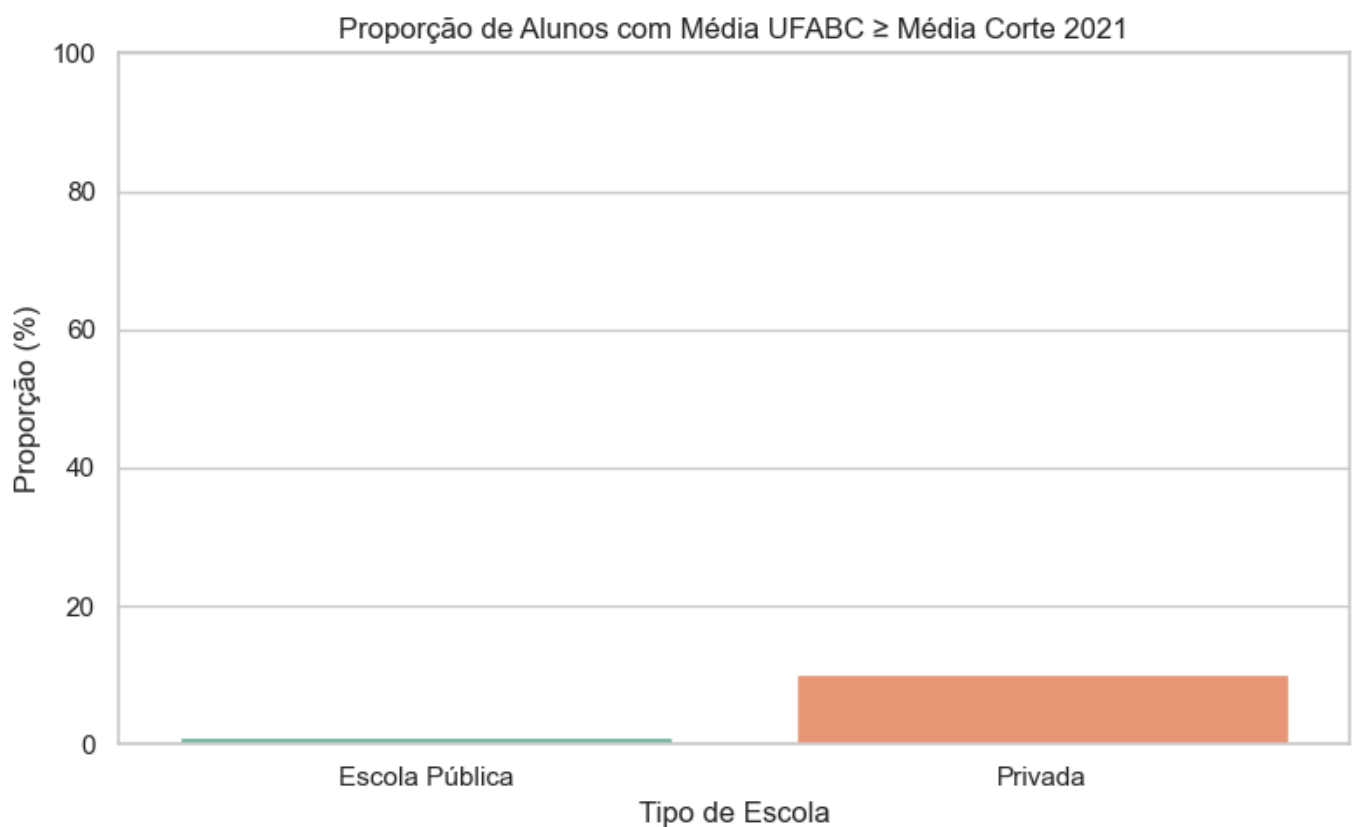
counts["Proporção (%)"] = (counts["Acima da media_corte"] / counts["Total"]) * 100

plt.figure(figsize=(8, 5))
sns.barplot(x=counts.index, y=counts["Proporção (%)"], palette='Set2')
plt.title('Proporção de Alunos com Média UFABC  $\geq$  Média Corte 2021')
plt.ylabel('Proporção (%)')
plt.xlabel('Tipo de Escola')
plt.ylim(0, 100)
plt.tight_layout()
plt.show()
```

C:\Users\rodri\AppData\Local\Temp\ipykernel_6636\2915826068.py:14: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x=counts.index, y=counts["Proporção (%)"], palette='Set2')
```



Cálculo Estatístico

Este bloco realiza o cálculo da média geral das notas dos alunos participantes do ENEM, separando-os por tipo de escola (pública ou privada). A média é calculada com base nas cinco provas principais: Redação, Matemática, Ciências da Natureza, Ciências Humanas e Linguagens. Em seguida, são computadas:

- A média geral das médias dos alunos por tipo de escola
- A variância da média geral, para avaliar a dispersão dos dados

Esses indicadores estatísticos permitem comparar o desempenho médio e a consistência das notas entre os dois grupos educacionais ao longo dos anos analisados.

```
In [40]: #2023
# Criar a média das 5 notas para cada aluno
enem_2023['MEDIA_GERAL'] = enem_2023[
    ['NU_NOTA_REDACAO', 'NU_NOTA_MT', 'NU_NOTA_CH', 'NU_NOTA_LC', 'NU_NOTA_CN']
].mean(axis=1)

# Filtrar por tipo de escola
publica_2023 = enem_2023[enem_2023['TP_ESCOLA'] == 'Escola pública']
privada_2023 = enem_2023[enem_2023['TP_ESCOLA'] == 'Privada']

# Calcular a média da média geral
media_publica_2023 = publica_2023['MEDIA_GERAL'].mean()
media_privada_2023 = privada_2023['MEDIA_GERAL'].mean()

#Calcular a Variância
var_publica_2023 = publica_2023['MEDIA_GERAL'].var()
var_privada_2023 = privada_2023['MEDIA_GERAL'].var()

print(f"Média geral dos alunos de escola pública 2023: {media_publica_2023:.2f}")
print(f"Média geral dos alunos de escola privada 2023: {media_privada_2023:.2f}")

print(f"Variância dos alunos de escola pública 2023: {var_publica_2023:.2f}")
print(f"Varância dos alunos de escola privada 2023: {var_privada_2023:.2f}")
```

Média geral dos alunos de escola pública 2023: 515.97
Média geral dos alunos de escola privada 2023: 616.17
Variância dos alunos de escola pública 2023: 7650.14
Varância dos alunos de escola privada 2023: 6752.81

```
In [41]: #2022
# Criar a média das 5 notas para cada aluno
enem_2022['MEDIA_GERAL'] = enem_2022[
    ['NU_NOTA_REDACAO', 'NU_NOTA_MT', 'NU_NOTA_CH', 'NU_NOTA_LC', 'NU_NOTA_CN']
].mean(axis=1)

# Filtrar por tipo de escola
publica_2022 = enem_2022[enem_2022['TP_ESCOLA'] == 'Escola pública']
privada_2022 = enem_2022[enem_2022['TP_ESCOLA'] == 'Privada']

# Calcular a média da média geral
media_publica_2022 = publica_2022['MEDIA_GERAL'].mean()
media_privada_2022 = privada_2022['MEDIA_GERAL'].mean()

#Calcular a Variância
var_publica_2022 = publica_2022['MEDIA_GERAL'].var()
var_privada_2022 = privada_2022['MEDIA_GERAL'].var()

print(f"Média geral dos alunos de escola pública 2022: {media_publica_2022:.2f}")
print(f"Média geral dos alunos de escola privada 2022: {media_privada_2022:.2f}")

print(f"Variância dos alunos de escola pública 2022: {var_publica_2022:.2f}")
print(f"Varância dos alunos de escola privada 2022: {var_privada_2022:.2f}")
```

Média geral dos alunos de escola pública 2022: 519.27
Média geral dos alunos de escola privada 2022: 608.47
Variância dos alunos de escola pública 2022: 6510.06
Varância dos alunos de escola privada 2022: 5938.25

```
In [42]: #2021
# Criar a média das 5 notas para cada aluno
enem_2021['MEDIA_GERAL'] = enem_2021[
    ['NU_NOTA_REDACAO', 'NU_NOTA_MT', 'NU_NOTA_CH', 'NU_NOTA_LC', 'NU_NOTA_CN']
].mean(axis=1)

# Filtrar por tipo de escola
publica_2021 = enem_2021[enem_2021['TP_ESCOLA'] == 'Escola pública']
privada_2021 = enem_2021[enem_2021['TP_ESCOLA'] == 'Privada']

# Calcular a média da média geral
media_publica_2021 = publica_2021['MEDIA_GERAL'].mean()
media_privada_2021 = privada_2021['MEDIA_GERAL'].mean()

#Calcular a Variância
var_publica_2021 = publica_2021['MEDIA_GERAL'].var()
var_privada_2021 = privada_2021['MEDIA_GERAL'].var()

print(f"Média geral dos alunos de escola pública 2021: {media_publica_2021:.2f}")
print(f"Média geral dos alunos de escola privada 2021: {media_privada_2021:.2f}")

print(f"Variância dos alunos de escola pública 2021: {var_publica_2021:.2f}")
print(f"Varância dos alunos de escola privada 2021: {var_privada_2021:.2f}")
```

Média geral dos alunos de escola pública 2021: 509.37
Média geral dos alunos de escola privada 2021: 598.91
Variância dos alunos de escola pública 2021: 6358.43
Varância dos alunos de escola privada 2021: 6857.71

```
In [43]: # Função da variância combinada
def variancia_combinada(var1, var2, n1, n2):
    return ((n1 - 1) * var1 + (n2 - 1) * var2) / (n1 + n2 - 2)
```

```

# Função do t
def estatistica_t(media1, media2, var_pooled, n1, n2):
    return (media1 - media2) / math.sqrt(var_pooled * (1/n1 + 1/n2))

# Função do p-valor bilateral
def calcular_p_valor(t_stat, n1, n2):
    gl = n1 + n2 - 2 # graus de liberdade
    return 2 * t.cdf(t_stat, gl) if t_stat < 0 else 2 * (1 - t.cdf(t_stat, gl))

# ----- 2023 -----
n1_2023 = len(publica_2023)
n2_2023 = len(privada_2023)
var_pool_2023 = variancia_combinada(var_publica_2023, var_privada_2023, n1_2023, n2_2023)
t_2023 = estatistica_t(media_publica_2023, media_privada_2023, var_pool_2023, n1_2023, n2_2023)
p_2023 = calcular_p_valor(t_2023, n1_2023, n2_2023)
print(f"Variância combinada 2023: {var_pool_2023:.2f}")
print(f"Estatística t para 2023: {t_2023:.2f}")
print(f"Valor-p para 2023: {p_2023:.10f}")

# ----- 2022 -----
n1_2022 = len(publica_2022)
n2_2022 = len(privada_2022)
var_pool_2022 = variancia_combinada(var_publica_2022, var_privada_2022, n1_2022, n2_2022)
t_2022 = estatistica_t(media_publica_2022, media_privada_2022, var_pool_2022, n1_2022, n2_2022)
p_2022 = calcular_p_valor(t_2022, n1_2022, n2_2022)
print(f"Variância combinada 2022: {var_pool_2022:.2f}")
print(f"Estatística t para 2022: {t_2022:.2f}")
print(f"Valor-p para 2022: {p_2022:.10f}")

# ----- 2021 -----
n1_2021 = len(publica_2021)
n2_2021 = len(privada_2021)
var_pool_2021 = variancia_combinada(var_publica_2021, var_privada_2021, n1_2021, n2_2021)
t_2021 = estatistica_t(media_publica_2021, media_privada_2021, var_pool_2021, n1_2021, n2_2021)
p_2021 = calcular_p_valor(t_2021, n1_2021, n2_2021)
print(f"Variância combinada 2021: {var_pool_2021:.2f}")
print(f"Estatística t para 2021: {t_2021:.2f}")
print(f"Valor-p para 2021: {p_2021:.10f}")

```

```

Variância combinada 2023: 7461.08
Estatística t para 2023: -482.22
Valor-p para 2023: 0.0000000000
Variância combinada 2022: 6392.06
Estatística t para 2022: -437.81
Valor-p para 2022: 0.0000000000
Variância combinada 2021: 6464.53
Estatística t para 2021: -414.93
Valor-p para 2021: 0.0000000000

```

Junção das tabelas de 2023, 2022 e 2021 em uma só

Este bloco faz a junção das tabelas dos três anos em uma só e cria uma nova coluna com a média ponderada para ingressar em BC&T na UFABC

```

In [45]: # Aplicar a média ponderada de BC&T da UFABC para cada ano
for df in [enem_2021, enem_2022, enem_2023]:
    df['MEDIA_UFABC'] = df.apply(calcular_media_ponderada, axis=1)

# Adicionar coluna de ano, se ainda não tiver
enem_2021['ANO'] = 2021
enem_2022['ANO'] = 2022
enem_2023['ANO'] = 2023

```

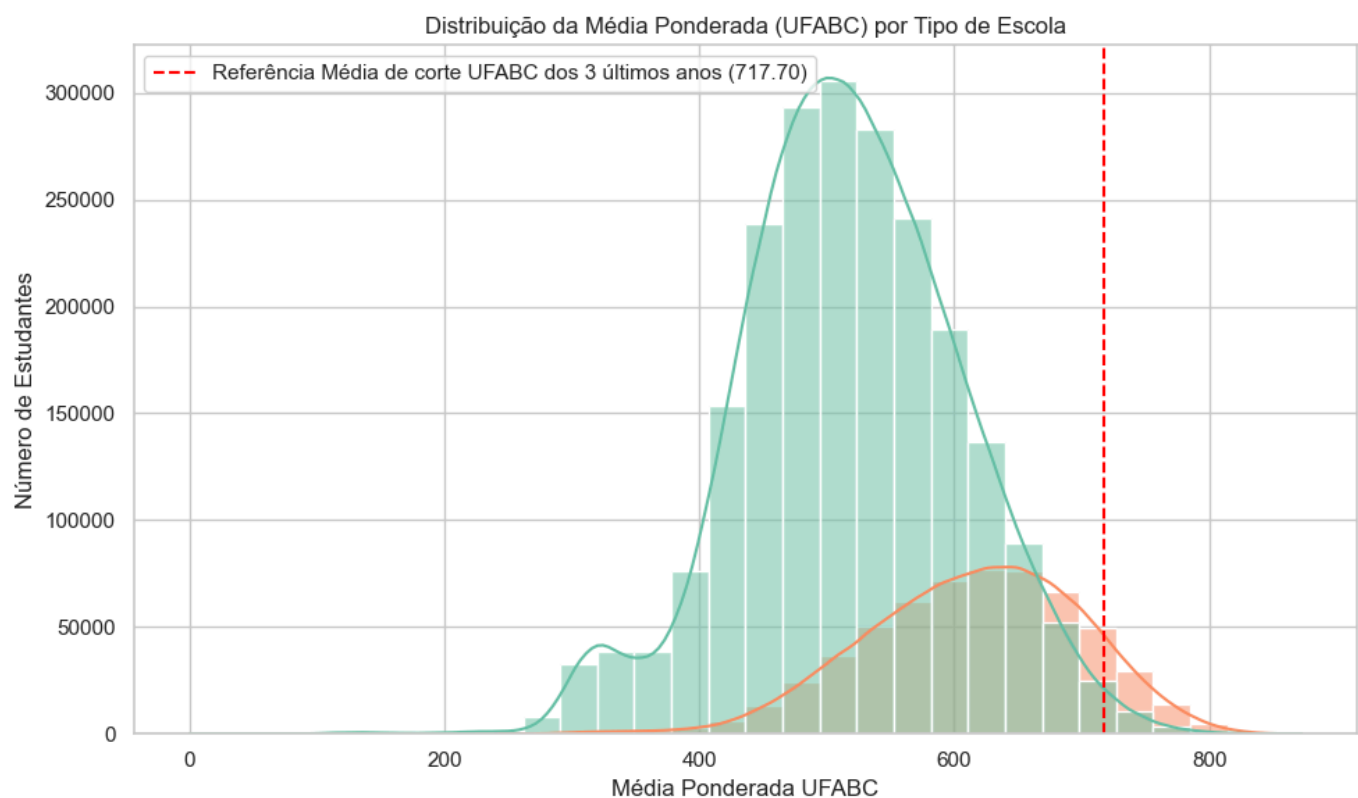


```
# Juntar todos os anos em um único DataFrame
```

```
enem_total = pd.concat([enem_2021, enem_2022, enem_2023], ignore_index=True)
```

Distribuição da Média Ponderada (UFABC) dos 3 anos juntos por Tipo de Escola

```
In [48]: plt.figure(figsize=(10, 6))
sns.histplot(data=enem_total, x='MEDIA_UFABC', hue='TP_ESCOLA', kde=True, bins=30, palette='S
plt.axvline(nota_corte_ufabc, color='red', linestyle='--', label='Referência Média de corte U
plt.title('Distribuição da Média Ponderada (UFABC) por Tipo de Escola')
plt.xlabel('Média Ponderada UFABC')
plt.ylabel('Número de Estudantes')
plt.legend()
plt.tight_layout()
plt.show()
```



Quantidade de pessoas que obtiveram notas maiores que as notas de corte por tipo de escola

Este bloco mostra a quantidade de pessoas por tipo de escola que tiraram nota maior que a de corte da UFABC, em Linguagens (nota hipotética 650) e Matemática (nota hipotética 720)

```
In [54]: qtd_publica = ((enem_total['TP_ESCOLA'] == 'Escola pública') & (enem_total['MEDIA_UFABC'] >=
qtd_privada = ((enem_total['TP_ESCOLA'] == 'Privada') & (enem_total['MEDIA_UFABC'] >= nota_co

print(f'MEDIA_UFABC ≥ {nota_corte_ufabc} – Pública: {qtd_publica} | Privada: {qtd_privada}')
```

```
nota_corte_mt = 720
nota_corte_lc = 650

# Contagem para Matemática
qtd_publica_mt = ((enem_total['TP_ESCOLA'] == 'Escola pública') & (enem_total['NU_NOTA_MT'] >=
qtd_privada_mt = ((enem_total['TP_ESCOLA'] == 'Privada') & (enem_total['NU_NOTA_MT'] >= nota_

# Contagem para Linguagens
qtd_publica_lc = ((enem_total['TP_ESCOLA'] == 'Escola pública') & (enem_total['NU_NOTA_LC'] >=
```

```

qtd_privada_lc = ((enem_total['TP_ESCOLA'] == 'Privada') & (enem_total['NU_NOTA_LC'] >= nota_
# Exibir os resultados
print(f'Matemática (≥ {nota_corte_mt}) – Pública: {qtd_publica_mt} | Privada: {qtd_privada_mt}
print(f'Linguagens (≥ {nota_corte_lc}) – Pública: {qtd_publica_lc} | Privada: {qtd_privada_lc}

MEDIA_UFABC ≥ 717.70666666666666 – Pública: 20767 | Privada: 63400
Matemática (≥ 720) – Pública: 65861 | Privada: 120471
Linguagens (≥ 650) – Pública: 16660 | Privada: 34066

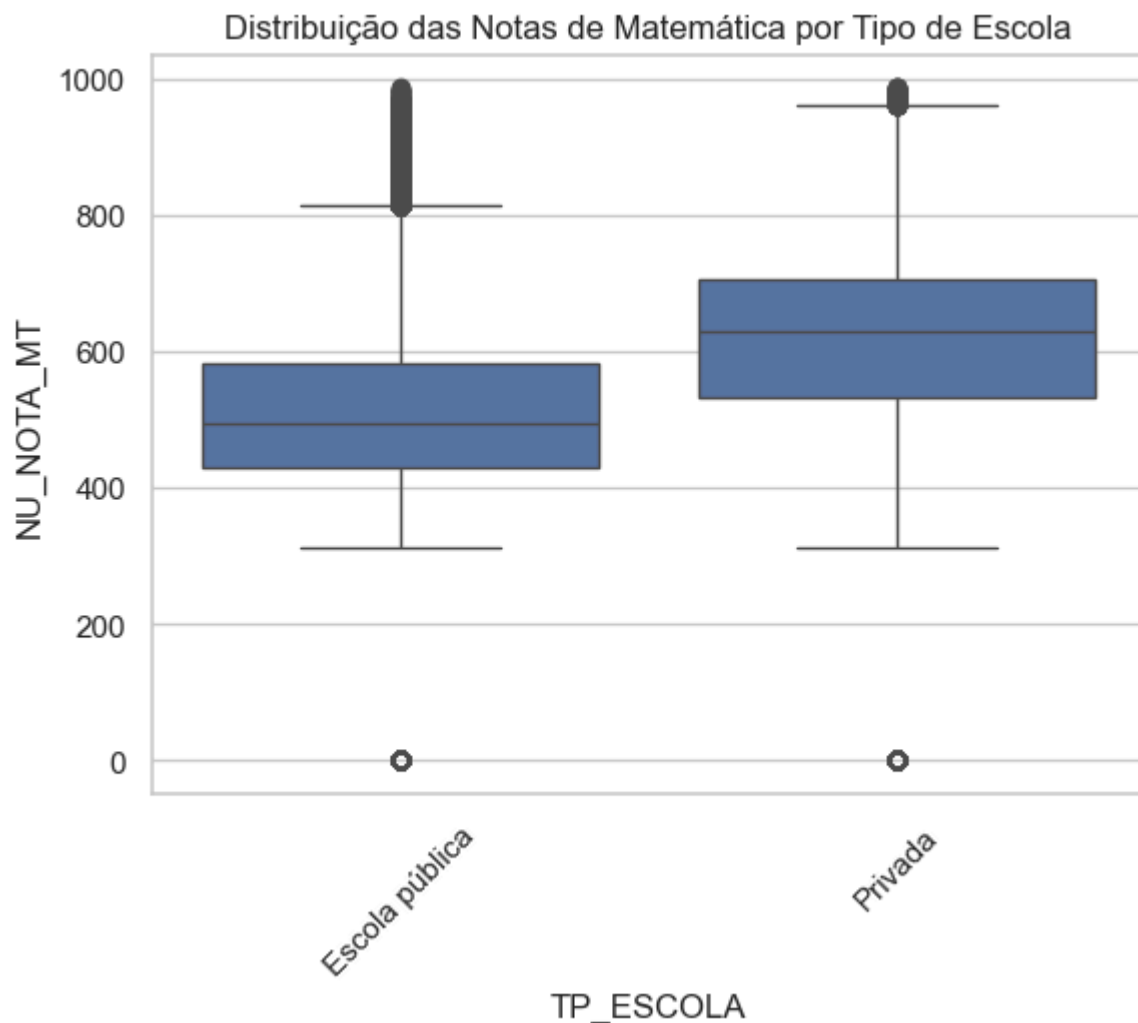
```

Distribuição das Notas de cada matéria por Tipo de Escola

```

In [58]: #Matemática
sns.boxplot(data=enem_total, x='TP_ESCOLA', y='NU_NOTA_MT')
plt.title("Distribuição das Notas de Matemática por Tipo de Escola")
plt.xticks(rotation=45)
plt.show()

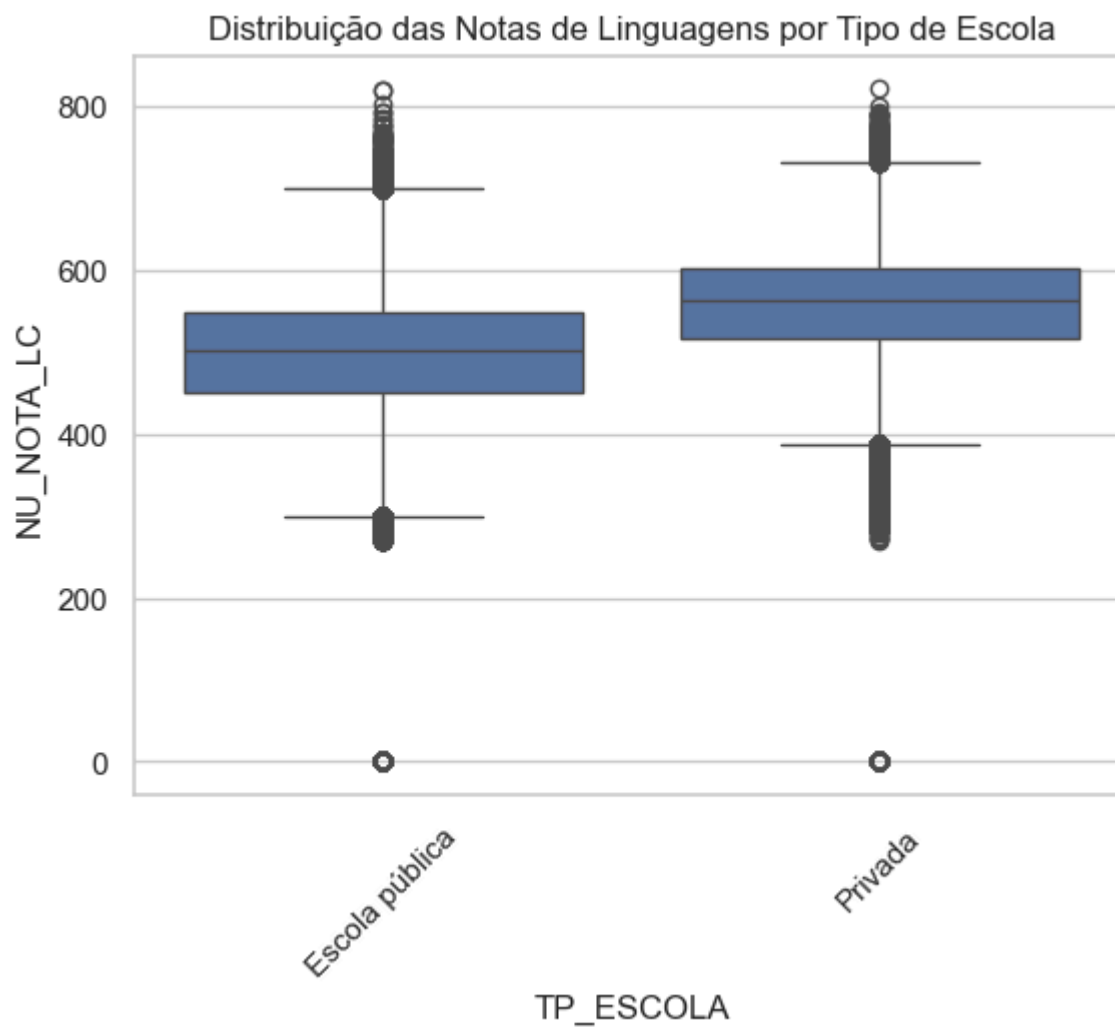
```



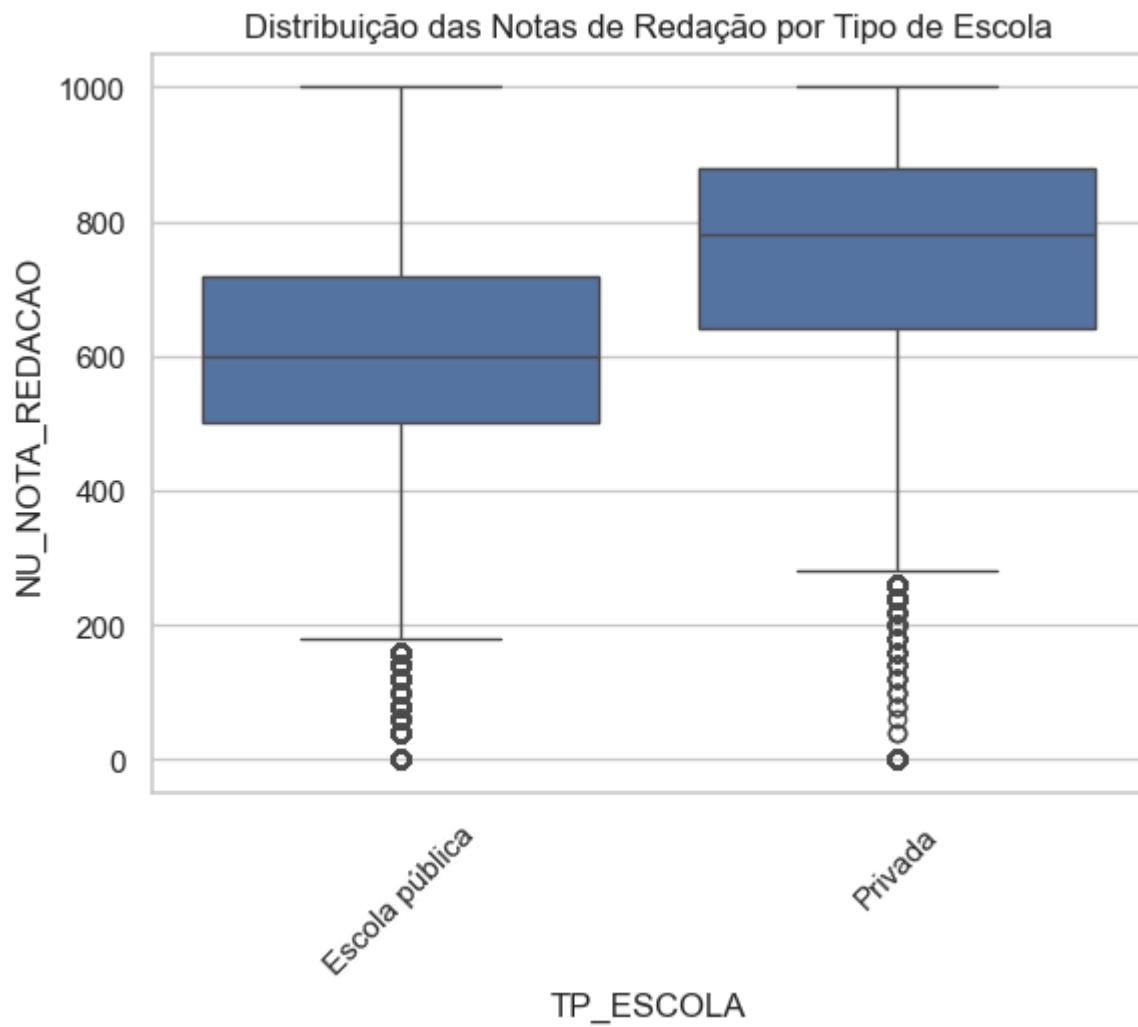
```

In [60]: #Linguagens
sns.boxplot(data=enem_total, x='TP_ESCOLA', y='NU_NOTA_LC')
plt.title("Distribuição das Notas de Linguagens por Tipo de Escola")
plt.xticks(rotation=45)
plt.show()

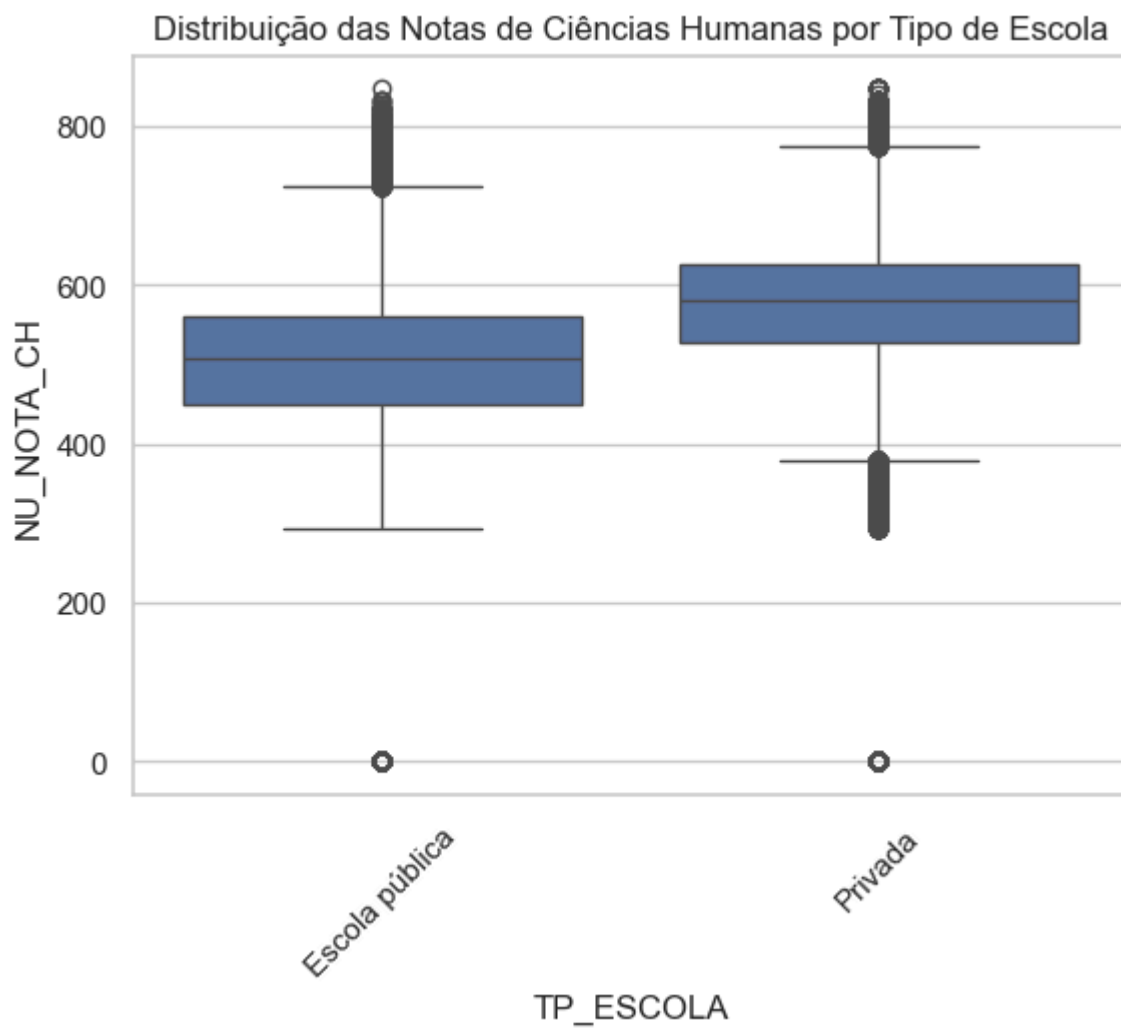
```



```
In [62]: #Redação
sns.boxplot(data=enem_total, x='TP_ESCOLA', y='NU_NOTA_REDACAO')
plt.title("Distribuição das Notas de Redação por Tipo de Escola")
plt.xticks(rotation=45)
plt.show()
```



```
In [64]: #Ciências Humanas
sns.boxplot(data=enem_total, x='TP_ESCOLA', y='NU_NOTA_CH')
plt.title("Distribuição das Notas de Ciências Humanas por Tipo de Escola")
plt.xticks(rotation=45)
plt.show()
```



```
In [66]: #Ciências da Natureza
sns.boxplot(data=enem_total, x='TP_ESCOLA', y='NU_NOTA_CN')
plt.title("Distribuição das Notas de Ciências da Natureza por Tipo de Escola")
plt.xticks(rotation=45)
plt.show()
```

Distribuição das Notas de Ciências da Natureza por Tipo de Escola

