codigo-projeto-te2

November 28, 2023

1 Carolina dos Anjos Figueiredo - GU3015475

1.1 Projeto - Tópicos Especiais II

[34]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeRegressor, plot_tree
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor, export_text
from sklearn.metrics import mean_squared_error
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt
from sklearn.tree import plot_tree
```

1.2 Dataset Cursos IFSP

1.2.1 Problema:

- Analisar o desempenho dos valores do ENADE, CPC e CC dos cursos oferecidos nos campus do Instituto Federal de São Paulo (IFSP)

1.2.2 Extração

[ ]:
```
cursos = pd.read_csv('ifsp_cursos2023.csv') # dataset CURSOS IFSP 2023
```

1.2.3 Exploração do Dataset:

[ ]:
```
cursos.head() # mostra as primeiras 5 linhas
```

[ ]:

	CÓDIGO DA IES	NOME DA IES
0	1810	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNO…
1	1810	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNO…
2	1810	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNO…
3	1810	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNO…
4	1810	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNO…

	SITUACAO DA IES	CÓDIGO DO CURSO	CÓDIGO DA DENOMINAÇÃO
0	Ativa	1518897	5342
1	Ativa	1128375	36
2	Ativa	1400668	3911
3	Ativa	1263236	5187
4	Ativa	1341356	29

	MARCAÇÃO DA DENOMINAÇÃO
0	NaN
1	DENOMINAÇÃO UTILIZADA PELO SISTEMA/CATÁLOGO
2	DENOMINAÇÃO UTILIZADA PELO SISTEMA/CATÁLOGO
3	NaN
4	DENOMINAÇÃO UTILIZADA PELO SISTEMA/CATÁLOGO

	NOME DO CURSO
0	PEDAGOGIA EM EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
1	MATEMÁTICA
2	LETRAS - PORTUGUÊS E INGLÊS
3	FORMAÇÃO PEDAGÓGICA DE DOCENTES PARA A EDUCAÇÃ…
4	FÍSICA

	DATA DE CADASTRO DO CURSO	GRAU	CÓDIGO CINE RÓTULO	…	VALOR CC
0	27/02/2020	Licenciatura	0113P01	…	NaN
1	13/10/2010	Licenciatura	0114M01	…	4.0
2	23/05/2017	Licenciatura	0115L15	…	5.0
3	13/04/2016	Licenciatura	0113F01	…	4.0
4	12/11/2015	Licenciatura	0114F02	…	NaN

	ANO CC	CPC FAIXA	CPC CONTINUO	CPC ANO	VALOR ENADE	ENADE ANO
0	NaN	NaN	NaN	NaN	NaN	NaN
1	2016.0	4.0	333.0	2021.0	4.0	2021.0
2	2022.0	4.0	353.0	2021.0	4.0	2021.0
3	2018.0	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	NaN	NaN

	NOME COORDENADOR CURSO	SINALIZAÇÕES DA IES	SINALIZAÇÕES DE CURSO
0	Andreza Silva Areao	NaN	NaN
1	José Érick De Souza Lima	NaN	NaN
2	Patricia Horta	NaN	NaN
3	Flávio Biasutti Valadares	NaN	NaN

```
4                                NaN                 NaN                 NaN

[5 rows x 71 columns]
```

[ ]:
```
cursos.info() # Exibe informações sobre as variáveis e seus tipos
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 227 entries, 0 to 226
Data columns (total 71 columns):
 #   Column                                  Non-Null Count  Dtype
---  ------                                  --------------  -----
 0   CÓDIGO DA IES                           227 non-null    int64
 1   NOME DA IES                             227 non-null    object
 2   SITUACAO DA IES                         227 non-null    object
 3   CÓDIGO DO CURSO                         227 non-null    int64
 4   CÓDIGO DA DENOMINAÇÃO                   227 non-null    int64
 5   MARCAÇÃO DA DENOMINAÇÃO                 178 non-null    object
 6   NOME DO CURSO                           227 non-null    object
 7   DATA DE CADASTRO DO CURSO               227 non-null    object
 8   GRAU                                    227 non-null    object
 9   CÓDIGO CINE RÓTULO                      227 non-null    object
 10  CINE RÓTULO                             227 non-null    object
 11  CÓDIGO CINE ÁREA DETALHADA              227 non-null    int64
 12  CINE ÁREA DETALHADA                     227 non-null    object
 13  CÓDIGO CINE ÁREA ESPECÍFICA             227 non-null    int64
 14  CINE ÁREA ESPECÍFICA                    227 non-null    object
 15  CÓDIGO CINE ÁREA GERAL                  227 non-null    int64
 16  CINE ÁREA GERAL                         227 non-null    object
 17  MODALIDADE                              227 non-null    object
 18  SITUACAO DO CURSO                       227 non-null    object
 19  QT VAGAS AUTORIZADAS                    227 non-null    int64
 20  CARGA HORÁRIA                           227 non-null    int64
 21  CARGA HORÁRIA DISTÂNCIA                 203 non-null    float64
 22  CARGA HORÁRIA ESTÁGIO                   203 non-null    float64
 23  CARGA HORÁRIA ATIV. COMPLEMENTARES      203 non-null    float64
 24  CARGA HORÁRIA TCC                       203 non-null    float64
 25  CARGA HORÁRIA LIBRAS                    203 non-null    float64
 26  TIPO DE PERIODICIDADE                   227 non-null    object
 27  QUANTITATIVO PERIODICIDADE - INTEGRAL   37 non-null     float64
 28  QUANTIDADE DE VAGAS - INTEGRAL          37 non-null     float64
 29  QUANTITATIVO PERIODICIDADE - MATUTINO   48 non-null     float64
 30  QUANTIDADE DE VAGAS - MATUTINO          48 non-null     float64
 31  QUANTITATIVO PERIODICIDADE - VESPERTINO 8 non-null      float64
 32  QUANTIDADE DE VAGAS - VESPERTINO        8 non-null      float64
 33  QUANTITATIVO PERIODICIDADE - NOTURNO    113 non-null    float64
 34  QUANTIDADE DE VAGAS - NOTURNO           113 non-null    float64
 35  QUANTITATIVO PERIODICIDADE - NÃO SE APLICA  38 non-null     float64
```

```
 36  QUANTIDADE DE VAGAS - NÃO SE APLICA         38 non-null     float64
 37  CÓDIGO DO ENDEREÇO                          227 non-null    int64
 38  ENDERECO                                    227 non-null    object
 39  NUMERO ENDERECO                             223 non-null    object
 40  COMPLEMENTO                                 72 non-null     object
 41  BAIRRO                                      227 non-null    object
 42  MUNICIPIO                                   227 non-null    object
 43  UF                                          227 non-null    object
 44  TIPO DOC. AUTORIZACAO                       218 non-null    object
 45  DOCUMENTO DE AUTORIZACAO                    219 non-null    object
 46  DT CONSIDERADA AUTORIZACAO                  219 non-null    object
 47  DT. PUBLICACAO AUTORIZACAO                  219 non-null    object
 48  DT. CADASTRO AUTORIZACAO                    219 non-null    object
 49  TIPO DOC. RECONHECIMENTO                    111 non-null    object
 50  DOCUMENTO DE RECONHECIMENTO                 111 non-null    object
 51  DT CONSIDERADA RECONHECIMENTO               111 non-null    object
 52  DT. PUBLICACAO RECONHECIMENTO               111 non-null    object
 53  DT. CADASTRO RECONHECIMENTO                 111 non-null    object
 54  TIPO DOC. RENOVACAO                         72 non-null     object
 55  DOC. ULTIMA RENOVACAO                       72 non-null     object
 56  DT CONSIDERADA RENOVACAO                    72 non-null     object
 57  DT. PUBLICACAO RENOVACAO                    72 non-null     object
 58  DT. CADASTRO RENOVACAO                      72 non-null     object
 59  INICIO FUNCIONAMENTO                        221 non-null    object
 60  PROCESSOS EM TRAMITACAO                     92 non-null     float64
 61  VALOR CC                                    133 non-null    float64
 62  ANO CC                                      133 non-null    float64
 63  CPC FAIXA                                   87 non-null     float64
 64  CPC CONTINUO                                86 non-null     float64
 65  CPC ANO                                     87 non-null     float64
 66  VALOR ENADE                                 89 non-null     float64
 67  ENADE ANO                                   89 non-null     float64
 68  NOME COORDENADOR CURSO                      218 non-null    object
 69  SINALIZAÇÕES DA IES                         0 non-null      float64
 70  SINALIZAÇÕES DE CURSO                       0 non-null      float64
dtypes: float64(25), int64(9), object(37)
memory usage: 126.0+ KB
```

[ ]:
```
cursos.describe() # Exibe estatísticas descritivas para variáveis numéricas
```

[ ]:
```
       CÓDIGO DA IES  CÓDIGO DO CURSO  CÓDIGO DA DENOMINAÇÃO  \
count          227.0     2.270000e+02             227.000000
mean          1810.0     1.199882e+06            1441.295154
std              0.0     5.057366e+05            2092.866057
min           1810.0     4.823700e+04               1.000000
25%           1810.0     1.168204e+06              63.000000
50%           1810.0     1.313172e+06             192.000000
```

```
75%          1810.0     1.455788e+06            3007.000000
max          1810.0     5.001091e+06            5342.000000

       CÓDIGO CINE ÁREA DETALHADA  CÓDIGO CINE ÁREA ESPECÍFICA  \
count                  227.000000                   227.000000
mean                   422.814978                    41.911894
std                    291.142958                    29.098417
min                    113.000000                    11.000000
25%                    114.000000                    11.000000
50%                    413.000000                    41.000000
75%                    714.000000                    71.000000
max                   1015.000000                   101.000000

       CÓDIGO CINE ÁREA GERAL  QT VAGAS AUTORIZADAS  CARGA HORÁRIA  \
count              227.000000            227.000000     227.000000
mean                 4.079295            174.096916    2877.140969
std                  2.901459            348.763238     929.928995
min                  1.000000              0.000000       0.000000
25%                  1.000000             40.000000    2100.000000
50%                  4.000000             40.000000    3200.000000
75%                  7.000000             80.000000    3663.000000
max                 10.000000           1140.000000    4361.000000

       CARGA HORÁRIA DISTÂNCIA  CARGA HORÁRIA ESTÁGIO  …  \
count               203.000000             203.000000  …
mean                 27.443350             268.965517  …
std                 118.038257             145.017629  …
min                   0.000000               0.000000  …
25%                   0.000000             160.000000  …
50%                   0.000000             300.000000  …
75%                   0.000000             400.000000  …
max                 800.000000             420.000000  …

       PROCESSOS EM TRAMITACAO    VALOR CC       ANO CC  CPC FAIXA  \
count             9.200000e+01  133.000000   133.000000  87.000000
mean              2.020502e+08    4.112782  2017.225564   3.735632
std               1.670738e+05    0.623457     3.797161   0.637214
min               2.017152e+08    2.000000  2005.000000   0.000000
25%               2.019263e+08    4.000000  2015.000000   3.000000
50%               2.020183e+08    4.000000  2017.000000   4.000000
75%               2.022077e+08    4.000000  2019.000000   4.000000
max               2.023062e+08    5.000000  2023.000000   5.000000

       CPC CONTINUO      CPC ANO  VALOR ENADE    ENADE ANO  \
count     86.000000    87.000000    89.000000    89.000000
mean     319.988372  2019.977011     3.752809  2019.842697
std       39.493408     2.246325     0.801715     2.392692
```

```
min      209.000000  2009.000000     0.000000  2009.000000
25%      294.250000  2019.000000     3.000000  2019.000000
50%      320.500000  2021.000000     4.000000  2021.000000
75%      343.000000  2021.000000     4.000000  2021.000000
max      419.000000  2021.000000     5.000000  2021.000000

       SINALIZAÇÕES DA IES  SINALIZAÇÕES DE CURSO
count                  0.0                    0.0
mean                   NaN                    NaN
std                    NaN                    NaN
min                    NaN                    NaN
25%                    NaN                    NaN
50%                    NaN                    NaN
75%                    NaN                    NaN
max                    NaN                    NaN

[8 rows x 34 columns]
```

[ ]:
```
# Verifica a presença de dados ausentes em cada coluna
print(cursos.isnull().sum())
```

```
CÓDIGO DA IES               0
NOME DA IES                 0
SITUACAO DA IES             0
CÓDIGO DO CURSO             0
CÓDIGO DA DENOMINAÇÃO       0
                           …
VALOR ENADE               138
ENADE ANO                 138
NOME COORDENADOR CURSO      9
SINALIZAÇÕES DA IES       227
SINALIZAÇÕES DE CURSO     227
Length: 71, dtype: int64
```

1.2.4 Pré-processamento de Dados:

- Quais colunas serão utilizadas.

[ ]:
```
# Definindo as colunas que vão ser utilizadas
df = cursos[['CÓDIGO DO CURSO', 'NOME DO CURSO', 'DATA DE CADASTRO DO CURSO',␣
 ↪'GRAU',
             'CINE RÓTULO', 'CÓDIGO CINE ÁREA ESPECÍFICA', 'CINE ÁREA␣
 ↪ESPECÍFICA',
             'CÓDIGO CINE ÁREA GERAL', 'CINE ÁREA GERAL', 'MODALIDADE',␣
 ↪'SITUACAO DO CURSO',
             'QT VAGAS AUTORIZADAS', 'MUNICIPIO', 'VALOR CC', 'ANO CC', 'CPC␣
 ↪FAIXA',
```

```
            'CPC CONTINUO', 'CPC ANO', 'VALOR ENADE', 'ENADE ANO']].copy()
```

[ ]:
```
# verificando as informações das colunas
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 78 entries, 1 to 219
Data columns (total 20 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   CÓDIGO DO CURSO               78 non-null     int64
 1   NOME DO CURSO                 78 non-null     object
 2   DATA DE CADASTRO DO CURSO     78 non-null     object
 3   GRAU                          78 non-null     object
 4   CINE RÓTULO                   78 non-null     object
 5   CÓDIGO CINE ÁREA ESPECÍFICA   78 non-null     int64
 6   CINE ÁREA ESPECÍFICA          78 non-null     object
 7   CÓDIGO CINE ÁREA GERAL        78 non-null     int64
 8   CINE ÁREA GERAL               78 non-null     object
 9   MODALIDADE                    78 non-null     object
 10  SITUACAO DO CURSO             78 non-null     object
 11  QT VAGAS AUTORIZADAS          78 non-null     int64
 12  MUNICIPIO                     78 non-null     object
 13  VALOR CC                      78 non-null     float64
 14  ANO CC                        78 non-null     float64
 15  CPC FAIXA                     78 non-null     float64
 16  CPC CONTINUO                  78 non-null     float64
 17  CPC ANO                       78 non-null     float64
 18  VALOR ENADE                   78 non-null     float64
 19  ENADE ANO                     78 non-null     float64
dtypes: float64(7), int64(4), object(9)
memory usage: 12.8+ KB
```

[ ]:
```
# Tirar valores nulos
df.dropna(inplace=True)
```

[ ]:
```
df = df[(df != 0).all(axis=1)] # Linhas diferentes de 0
```

[ ]:
```
df.shape # Verificar a quantidade de linhas e de colunas
```

[ ]:
```
(78, 20)
```

[ ]:
```
# Converter os valores float64 para int64
df['CPC CONTINUO'] = df['CPC CONTINUO'].astype(int)
df['VALOR CC'] = df['VALOR CC'].astype(int)
df['ANO CC'] = df['ANO CC'].astype(int)
df['CPC FAIXA'] = df['CPC FAIXA'].astype(int)
```

```
df['CPC ANO'] = df['CPC ANO'].astype(int)
df['VALOR ENADE'] = df['VALOR ENADE'].astype(int)
df['ENADE ANO'] = df['ENADE ANO'].astype(int)
```

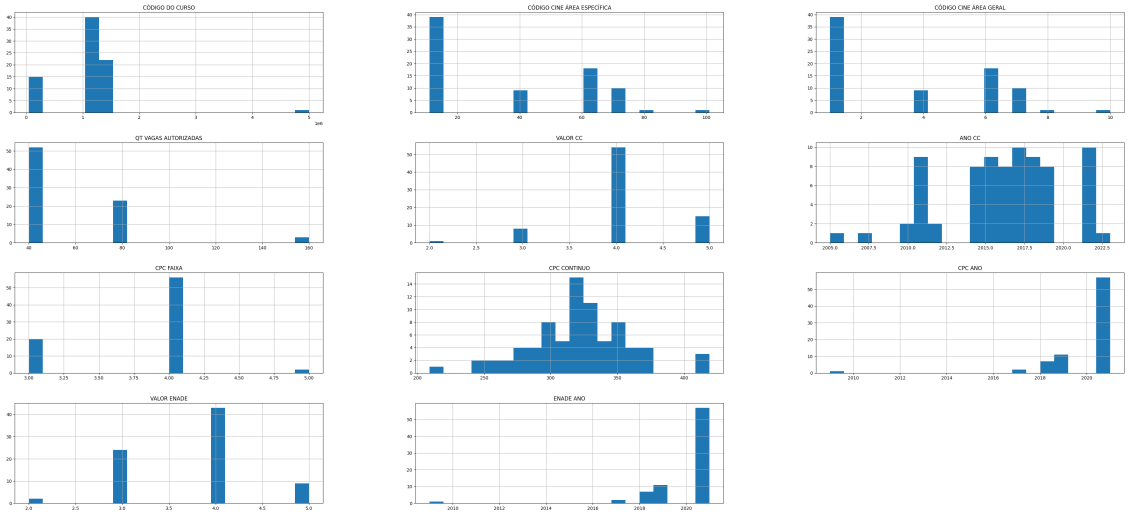
[ ]:
```
df.dtypes # Verificar se os dados foram transformados
```

[ ]:
```
CÓDIGO DO CURSO                 int64
NOME DO CURSO                  object
DATA DE CADASTRO DO CURSO      object
GRAU                           object
CINE RÓTULO                    object
CÓDIGO CINE ÁREA ESPECÍFICA     int64
CINE ÁREA ESPECÍFICA           object
CÓDIGO CINE ÁREA GERAL          int64
CINE ÁREA GERAL                object
MODALIDADE                     object
SITUACAO DO CURSO              object
QT VAGAS AUTORIZADAS            int64
MUNICIPIO                      object
VALOR CC                        int64
ANO CC                          int64
CPC FAIXA                       int64
CPC CONTINUO                    int64
CPC ANO                         int64
VALOR ENADE                     int64
ENADE ANO                       int64
ENADE                           int64
dtype: object
```

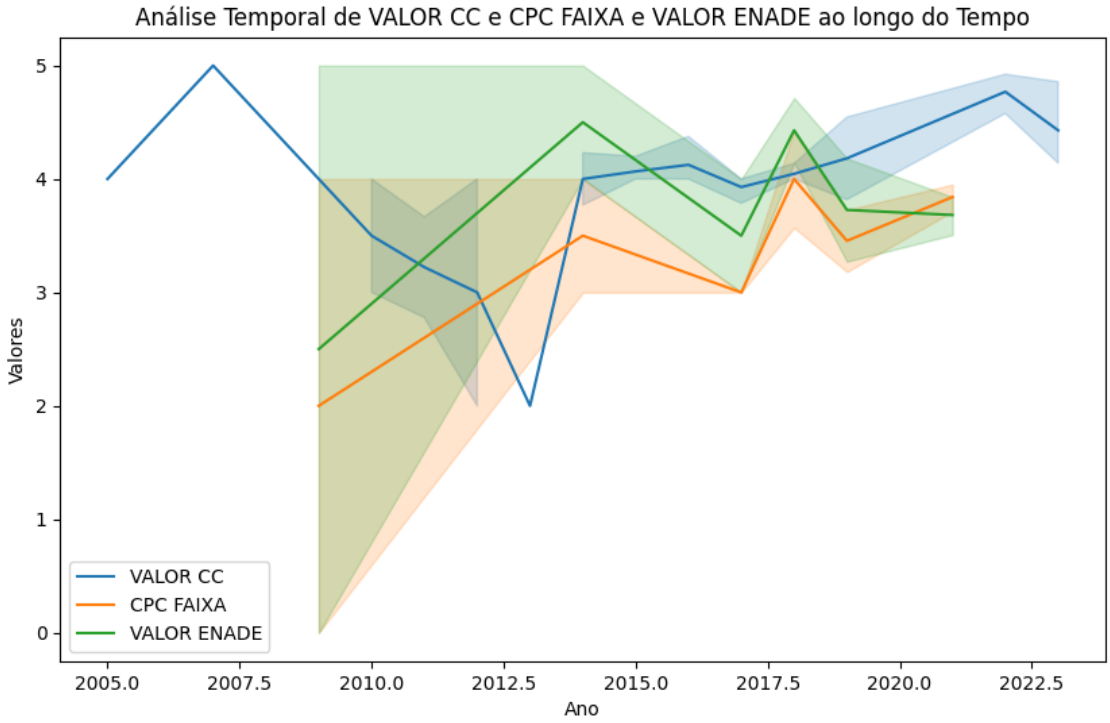
[ ]:
```
# Criando um novo csv com o Dataset tratado
df.to_csv('cursos.ifso_FINAL.csv', index=False)
```

1.3 Análise Exploratória de Dados (EDA):

[ ]:
```
# Exibe histogramas para variáveis numéricas
df.hist(bins=20, figsize=(45, 20))
plt.show(
```



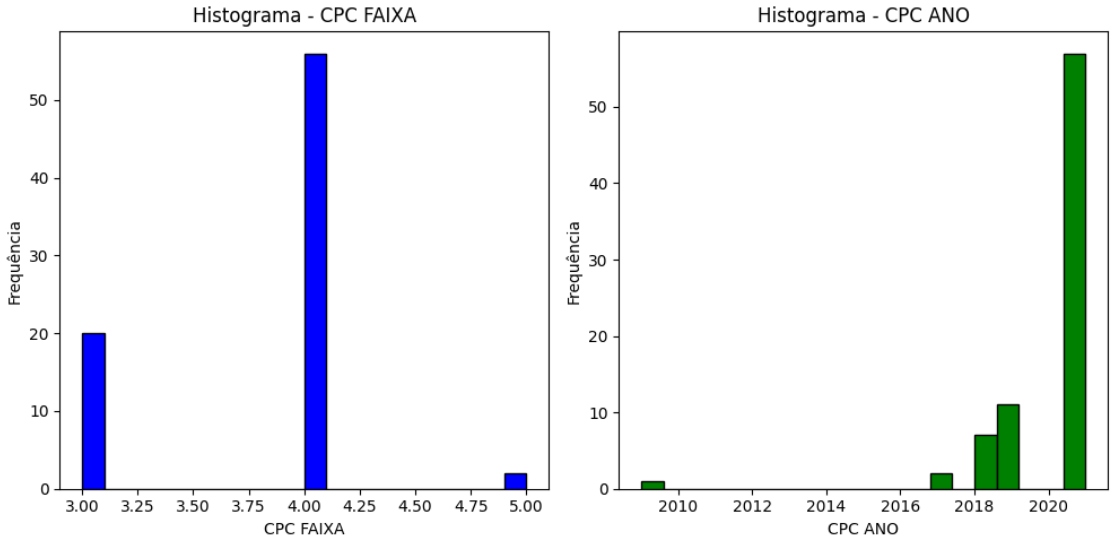
[15]:
```
plt.figure(figsize=(10, 6))
sns.lineplot(data=df, x='ANO CC', y='VALOR CC', label='VALOR CC')
sns.lineplot(data=df, x='CPC ANO', y='CPC FAIXA', label='CPC FAIXA')
sns.lineplot(data=df, x='ENADE ANO', y='VALOR ENADE', label='VALOR ENADE')
plt.xlabel('Ano')
plt.ylabel('Valores')
plt.title('Análise Temporal de VALOR CC e CPC FAIXA e VALOR ENADE ao longo do␣
↪Tempo')
plt.legend()
plt.show()
```



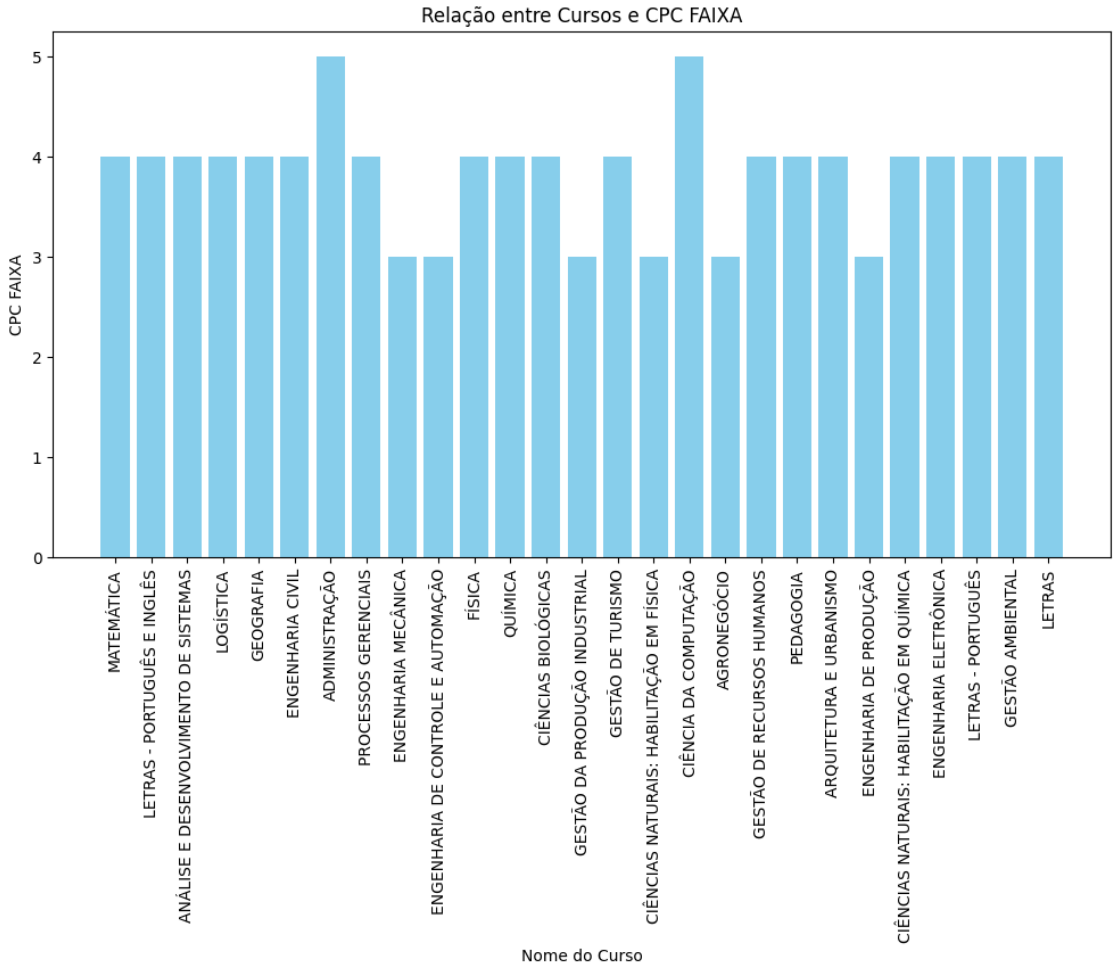
1.4 Visualização de Dados *CPC*

[ ]:
```
cpc_continuo = df['CPC FAIXA']
cpc_ano = df['CPC ANO']

plt.figure(figsize=(15, 5))

# Histograma para CPC CONTINUO
plt.subplot(1, 3, 1)
plt.hist(cpc_continuo, bins=20, color='blue', edgecolor='black')
plt.title('Histograma - CPC FAIXA')
plt.xlabel('CPC FAIXA')
plt.ylabel('Frequência')

# Histograma para CPC ANO
plt.subplot(1, 3, 2)
plt.hist(cpc_ano, bins=20, color='green', edgecolor='black')
plt.title('Histograma - CPC ANO')
plt.xlabel('CPC ANO')
plt.ylabel('Frequência')

# Ajusta o layout para evitar sobreposição
plt.tight_layout()
```

```
# Exibe
plt.show()
```



[ ]:
```
cursos = df['NOME DO CURSO']
cpc_continuo = df['CPC FAIXA']

plt.figure(figsize=(12, 6))
plt.bar(cursos, cpc_continuo, color='skyblue')
plt.xticks(rotation=90)  # Rotaciona os rótulos do eixo x para melhor␣
↪legibilidade
plt.title('Relação entre Cursos e CPC FAIXA')
plt.xlabel('Nome do Curso')
plt.ylabel('CPC FAIXA')
plt.show()
```



[ ]:
```
print(df[['CPC FAIXA', 'CPC ANO']].info())
```

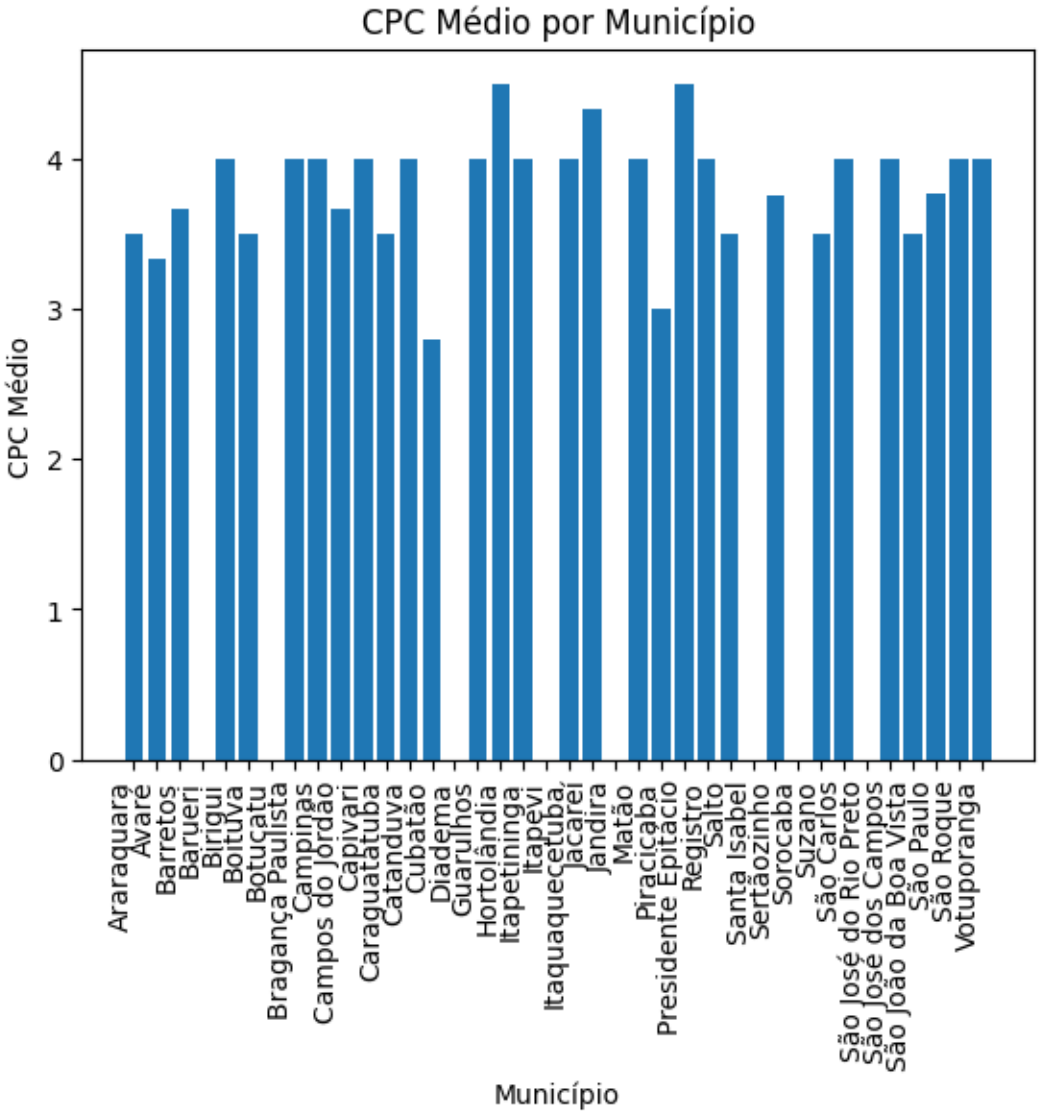
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 78 entries, 1 to 219
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   CPC FAIXA   78 non-null     int64
 1   CPC ANO     78 non-null     int64
dtypes: int64(2)
memory usage: 3.9 KB
None
```

[ ]:
```
contagem = df['CPC FAIXA'].value_counts()
print(contagem)
```

```
4    56
```

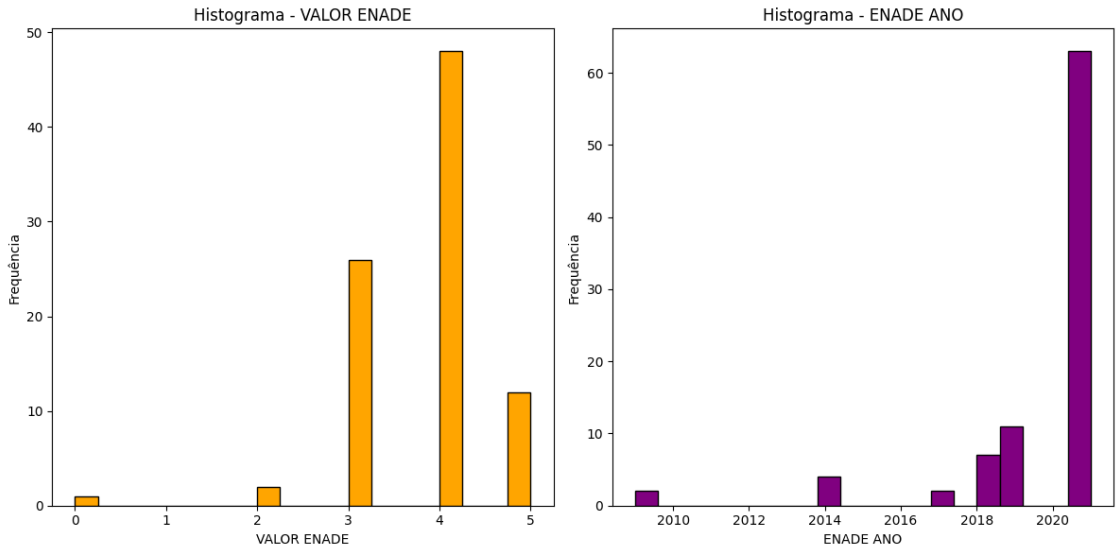
```
3    20
5     2
Name: CPC FAIXA, dtype: int64
```

[16]:
```
# Vai agrupar por município e calcular a média do CPC
dados_agrupados = df.groupby('MUNICIPIO')['CPC FAIXA'].mean().reset_index()
#Plotagem
plt.bar(dados_agrupados['MUNICIPIO'], dados_agrupados['CPC FAIXA'])
plt.xlabel('Município')
plt.ylabel('CPC Médio')
plt.title('CPC Médio por Município')
plt.xticks(rotation=90, ha='right')
plt.show()
```



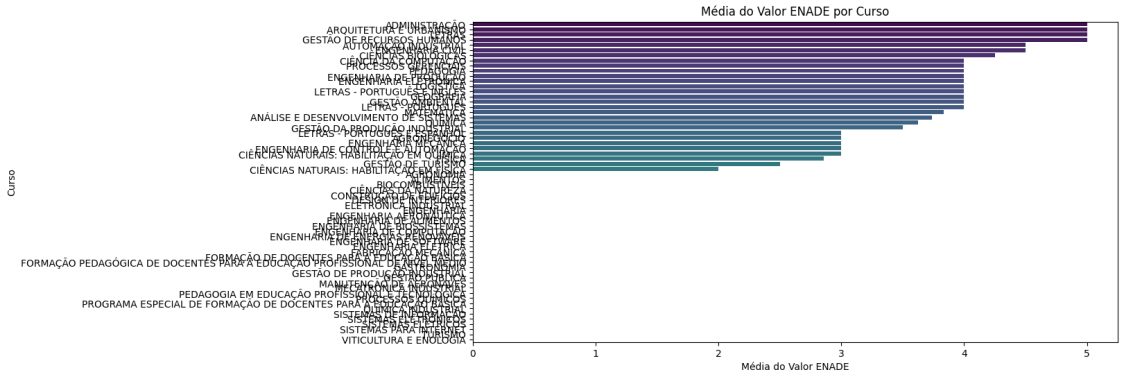
1.5 Visualização de Dados ENADE

[17]:
```
valor_enade = df['VALOR ENADE']
enade_ano = df['ENADE ANO']

plt.figure(figsize=(12, 6))

# Histograma para VALOR ENADE
plt.subplot(1, 2, 1)
plt.hist(valor_enade, bins=20, color='orange', edgecolor='black')
plt.title('Histograma - VALOR ENADE')
plt.xlabel('VALOR ENADE')
plt.ylabel('Frequência')

# Histograma para ENADE ANO
plt.subplot(1, 2, 2)
plt.hist(enade_ano, bins=20, color='purple', edgecolor='black')
plt.title('Histograma - ENADE ANO')
plt.xlabel('ENADE ANO')
plt.ylabel('Frequência')

plt.tight_layout()

plt.show()
```



[24]:
```
# Escolha as variáveis para a análise
cursos = df['NOME DO CURSO']
valor_enade = df['VALOR ENADE']
```

```
# Crie um DataFrame com as duas variáveis
df_relacao_cursos_enade = pd.DataFrame({'Curso': cursos, 'Valor ENADE':␣
 ↪valor_enade})

# Calcule a média do Valor ENADE para cada curso
media_enade_por_curso = df_relacao_cursos_enade.groupby('Curso')['Valor ENADE'].
 ↪mean().reset_index()

# Ordene os cursos pela média do Valor ENADE
media_enade_por_curso = media_enade_por_curso.sort_values(by='Valor ENADE',␣
 ↪ascending=False)

# Crie um gráfico de barra
plt.figure(figsize=(12, 6))
sns.barplot(x='Valor ENADE', y='Curso', data=media_enade_por_curso,␣
 ↪palette='viridis')
plt.title('Média do Valor ENADE por Curso')
plt.xlabel('Média do Valor ENADE')
plt.ylabel('Curso')
plt.show()
```



[ ]:
```
print(df[['VALOR ENADE', 'ENADE ANO']].info())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 78 entries, 1 to 219
Data columns (total 2 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   VALOR ENADE  78 non-null     int64
 1   ENADE ANO    78 non-null     int64
dtypes: int64(2)
memory usage: 3.9 KB
None
```

[ ]:
```
# Vai agrupar os dados por município e calcular a média do VALOR ENADE para␣
  ↪cada município
resultado_por_municipio = df.groupby('MUNICIPIO')['VALOR ENADE'].mean()

print(resultado_por_municipio)
```

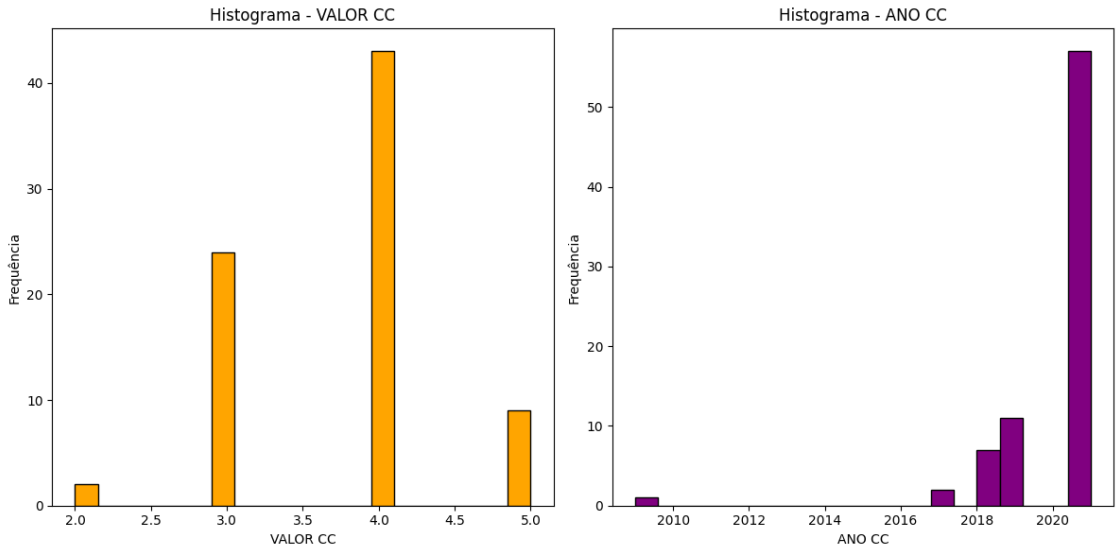
```
MUNICIPIO
Araraquara               3.500000
Avaré                    3.500000
Barretos                 3.500000
Birigui                  3.500000
Boituva                  3.500000
Bragança Paulista        4.000000
Campinas                 4.000000
Campos do Jordão         3.666667
Capivari                 4.000000
Caraguatatuba            3.750000
Catanduva                3.500000
Cubatão                  3.666667
Guarulhos                4.000000
Hortolândia              4.000000
Itapetininga             3.500000
Itaquaquecetuba          4.000000
Jacareí                  4.500000
Matão                    3.000000
Piracicaba               2.666667
Presidente Epitácio      4.000000
Registro                 3.000000
Salto                    4.000000
Sertãozinho              3.666667
Suzano                   4.000000
São Carlos               4.000000
São José dos Campos      4.000000
São João da Boa Vista    3.000000
São Paulo                4.066667
São Roque                4.333333
Votuporanga              3.666667
Name: VALOR ENADE, dtype: float64
```

[ ]:
```
contagem = df['VALOR ENADE'].value_counts()
print(contagem)
```

```
4    43
3    24
5     9
2     2
Name: VALOR ENADE, dtype: int64
```

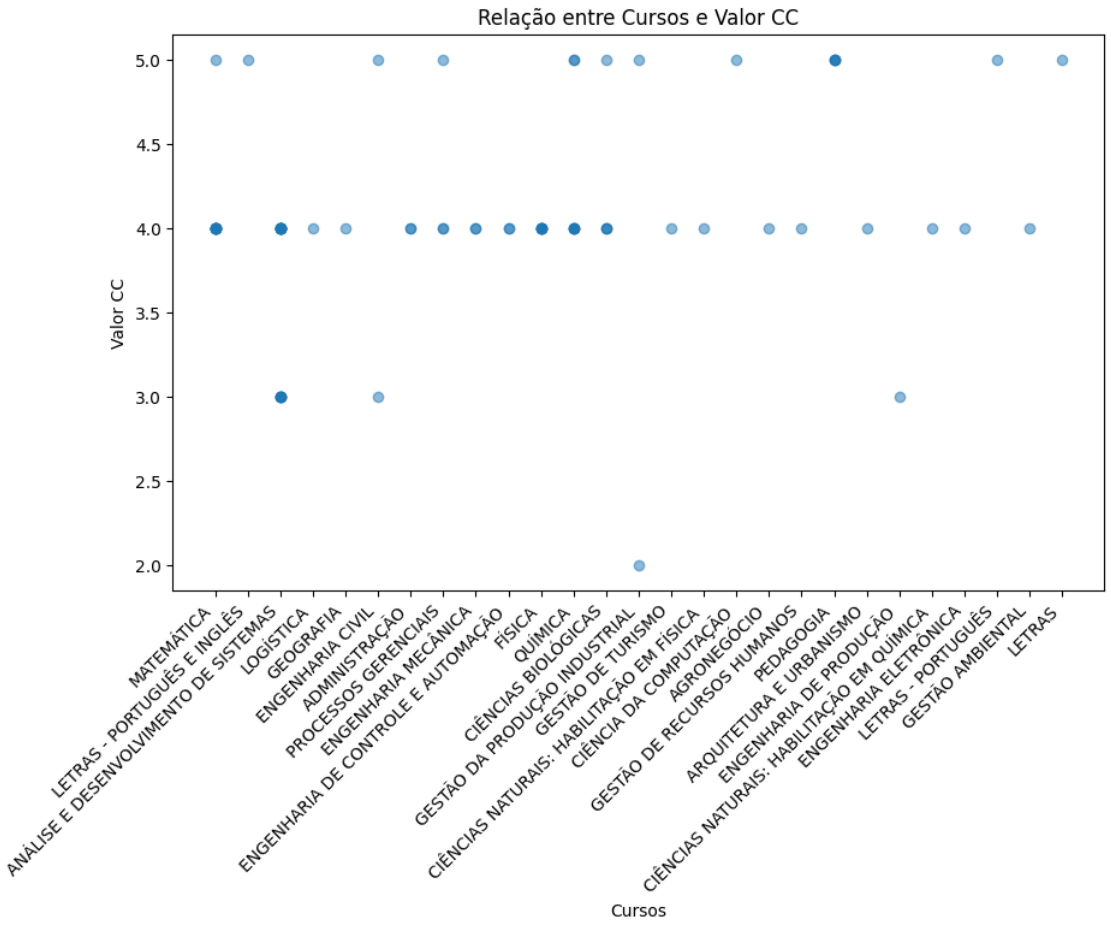
1.6 Visualização de Dados CC

[ ]:
```
valor_CC = df['VALOR CC']
CC_ano = df['ANO CC']

plt.figure(figsize=(12, 6))

# Histograma para VALOR ENADE
plt.subplot(1, 2, 1)
plt.hist(valor_enade, bins=20, color='orange', edgecolor='black')
plt.title('Histograma - VALOR CC')
plt.xlabel('VALOR CC')
plt.ylabel('Frequência')

# Histograma para ENADE ANO
plt.subplot(1, 2, 2)
plt.hist(enade_ano, bins=20, color='purple', edgecolor='black')
plt.title('Histograma - ANO CC')
plt.xlabel('ANO CC')
plt.ylabel('Frequência')

plt.tight_layout()

plt.show()
```



[ ]:
```
cursos = df['NOME DO CURSO']
valor_cc = df['VALOR CC']

plt.figure(figsize=(10, 6))
```

```
plt.scatter(cursos, valor_cc, alpha=0.5)
plt.title('Relação entre Cursos e Valor CC')
plt.xlabel('Cursos')
plt.ylabel('Valor CC')
plt.xticks(rotation=45, ha='right')  # Ajuste a rotação do rótulo do eixo x
plt.show()
```



[ ]:
```
contagem = df['VALOR CC'].value_counts()
print(contagem)
```

```
4    54
5    15
3     8
2     1
Name: VALOR CC, dtype: int64
```

[ ]:
```
cc_por_municipio = df.groupby('MUNICIPIO')['VALOR CC'].mean().reset_index()
```

```
print(cc_por_municipio)
```

```
                MUNICIPIO  VALOR CC
0             Araraquara  4.000000
1                  Avaré  4.000000
2               Barretos  4.000000
3                Birigui  4.000000
4                Boituva  4.500000
5      Bragança Paulista  3.500000
6               Campinas  4.000000
7       Campos do Jordão  4.333333
8               Capivari  4.000000
9          Caraguatatuba  4.000000
10             Catanduva  4.000000
11               Cubatão  4.000000
12             Guarulhos  3.500000
13           Hortolândia  4.000000
14          Itapetininga  4.000000
15       Itaquaquecetuba  5.000000
16               Jacareí  4.500000
17                 Matão  4.000000
18           Piracicaba  4.000000
19   Presidente Epitácio  5.000000
20              Registro  4.000000
21                 Salto  2.500000
22           Sertãozinho  4.000000
23                Suzano  4.500000
24            São Carlos  3.500000
25   São José dos Campos  4.500000
26  São João da Boa Vista  4.250000
27             São Paulo  4.000000
28             São Roque  4.333333
29           Votuporanga  4.333333
```

1.7 Visualização das 3 Variaveis

[ ]:
```
estatisticas_por_grau = df.groupby('GRAU').agg({
    'VALOR CC': ['mean', 'median', 'std'],
    'VALOR ENADE': ['mean', 'median', 'std'],
    'CPC FAIXA': ['mean', 'median', 'std']
}).reset_index()

print(estatisticas_por_grau)
```

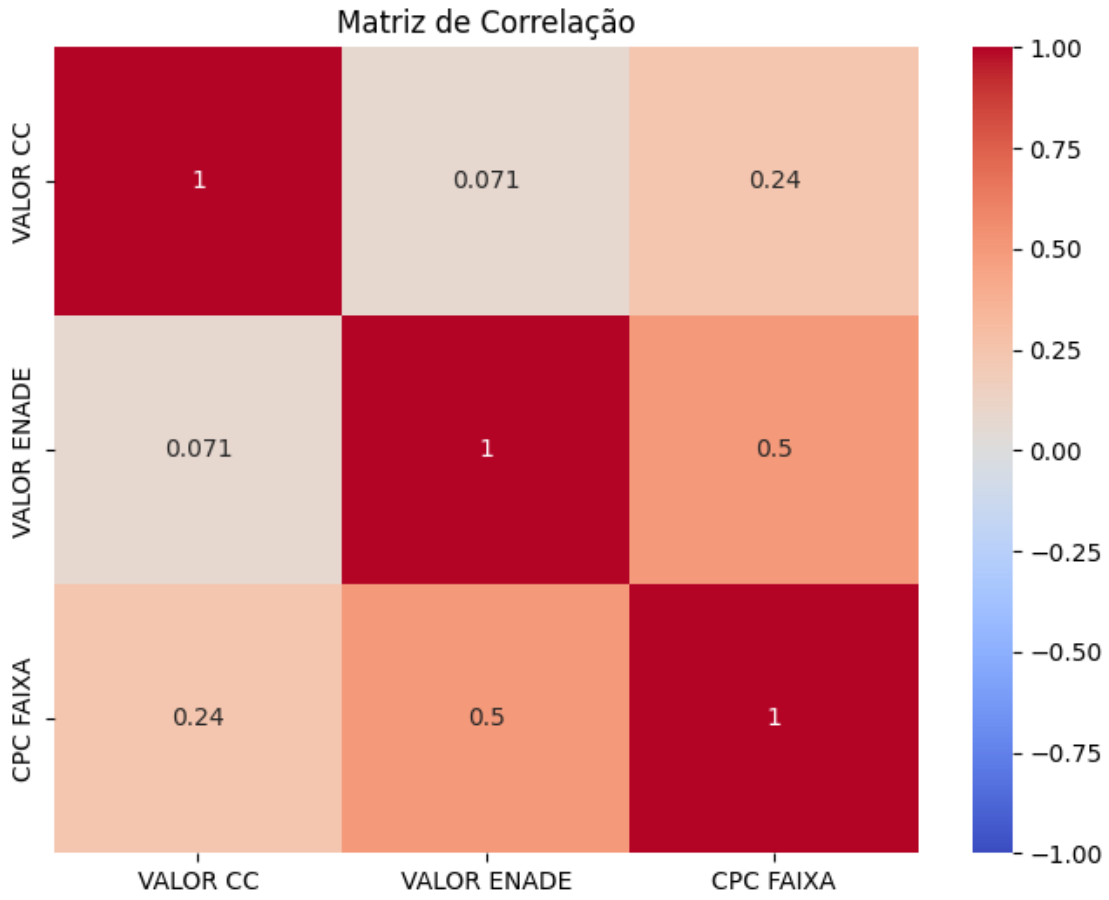
```
          GRAU  VALOR CC                   VALOR ENADE                   \
                    mean median       std        mean median       std
0   Bacharelado  4.000000    4.0  0.603023    4.000000    4.0  0.852803
```

```
1  Licenciatura  4.282051    4.0  0.455881    3.615385    4.0  0.673380
2   Tecnológico  3.777778    4.0  0.640513    3.851852    4.0  0.601518

  CPC FAIXA
       mean median       std
0  3.750000    4.0  0.753778
1  3.871795    4.0  0.338688
2  3.629630    4.0  0.492103
```

[ ]:
```
correlacao_por_grau = df.groupby('GRAU')[['VALOR CC', 'VALOR ENADE', 'CPC␣
↪FAIXA']].corr().reset_index()

print(correlacao_por_grau)
```

```
           GRAU      level_1  VALOR CC  VALOR ENADE  CPC FAIXA
0   Bacharelado     VALOR CC  1.000000     0.176777   0.400000
1   Bacharelado  VALOR ENADE  0.176777     1.000000   0.707107
2   Bacharelado    CPC FAIXA  0.400000     0.707107   1.000000
3  Licenciatura     VALOR CC  1.000000     0.448405   0.069923
4  Licenciatura  VALOR ENADE  0.448405     1.000000   0.470422
5  Licenciatura    CPC FAIXA  0.069923     0.470422   1.000000
6   Tecnológico     VALOR CC  1.000000    -0.188563   0.094907
7   Tecnológico  VALOR ENADE -0.188563     1.000000   0.587109
8   Tecnológico    CPC FAIXA  0.094907     0.587109   1.000000
```

[ ]:
```
variaveis = ['VALOR CC', 'VALOR ENADE', 'CPC FAIXA']

# Calcula a matriz de correlação
matriz_correlacao = df[variaveis].corr()

plt.figure(figsize=(8, 6))
sns.heatmap(matriz_correlacao, annot=True, cmap='coolwarm', vmin=-1, vmax=1)
plt.title('Matriz de Correlação')
plt.show()
```



2 MACHINE LEARNING - DESEMPENHO

2.1 REGRESSÃO LINEAR E ARVORE DE DECISAO ENADE

[27]:
```
dados_interesse_enade = df[['VALOR ENADE', 'ENADE ANO']]

# Remover linhas com valores nulos, se necessário
dados_interesse_enade = dados_interesse_enade.dropna()
```

[28]:
```
# Dividir os dados em conjuntos de treino e teste
X = dados_interesse_enade[['ENADE ANO']]
y = dados_interesse_enade['VALOR ENADE']
X_train_enade, X_test_enade, y_train_enade, y_test_enade = train_test_split(X,␣
 ↪y, test_size=0.2, random_state=42)
```

[29]:
```
# Criar o modelo de regressão linear
model_enade = LinearRegression()
```

[30]:
```
# Treinar o modelo
model_enade.fit(X_train_enade, y_train_enade)
```

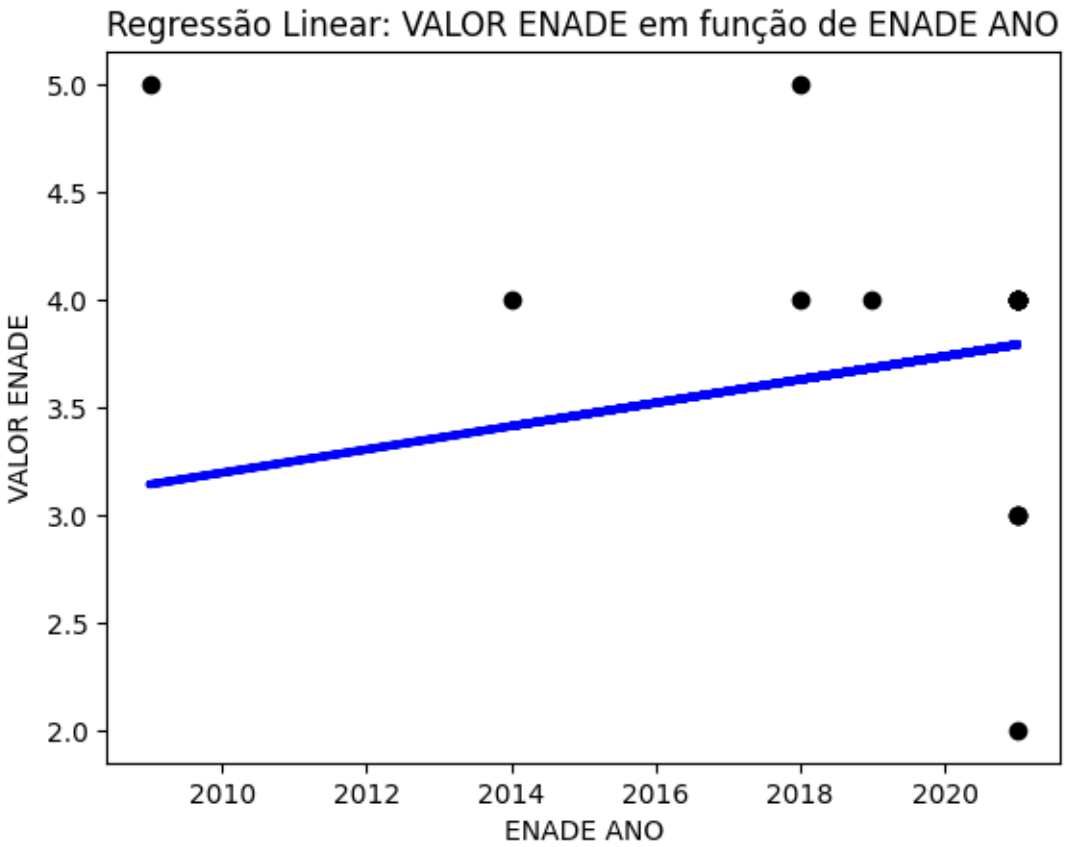
[30]:
```
LinearRegression()
```

[31]:
```
# Fazer previsões no conjunto de teste
y_pred_enade = model_enade.predict(X_test_enade)
```

[32]:
```
# Avaliar o desempenho do modelo
mse_enade = mean_squared_error(y_test_enade, y_pred_enade)
print(f'Mean Squared Error para VALOR ENADE: {mse_enade}')
```

```
Mean Squared Error para VALOR ENADE: 0.6331347027615535
```

[33]:
```
# Visualizar a regressão
plt.scatter(X_test_enade, y_test_enade, color='black')
plt.plot(X_test_enade, y_pred_enade, color='blue', linewidth=3)
plt.xlabel('ENADE ANO')
plt.ylabel('VALOR ENADE')
plt.title('Regressão Linear: VALOR ENADE em função de ENADE ANO')
plt.show()
```



[35]:
```
# Cria o modelo de árvore de decisão
model_enade = DecisionTreeRegressor()
```

[36]:
```
# Treina o modelo
model_enade.fit(X_train_enade, y_train_enade)
```

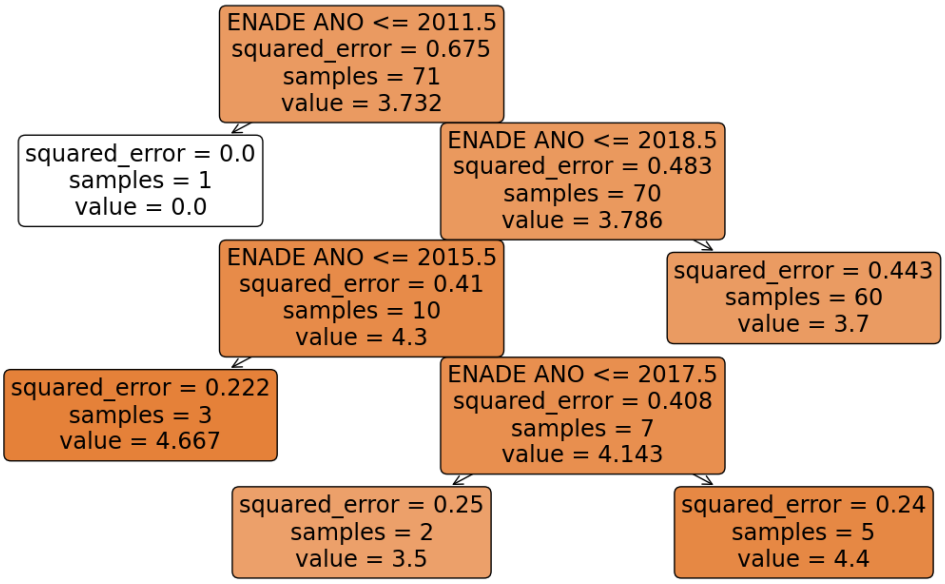
[36]:
```
DecisionTreeRegressor()
```

[37]:
```
# Faz previsões no conjunto de teste
y_pred_enade = model_enade.predict(X_test_enade)
```

[38]:
```
# Avalia o desempenho do modelo
mse_enade = mean_squared_error(y_test_enade, y_pred_enade)
print(f'Mean Squared Error para VALOR ENADE: {mse_enade}')
```

```
Mean Squared Error para VALOR ENADE: 1.7346913580246914
```

[39]:
```
# Visualiza a árvore de decisão (opcional)
plt.figure(figsize=(15, 8))
plot_tree(model_enade, feature_names=X_train_enade.columns, filled=True,␣
↪rounded=True)
plt.show()
```

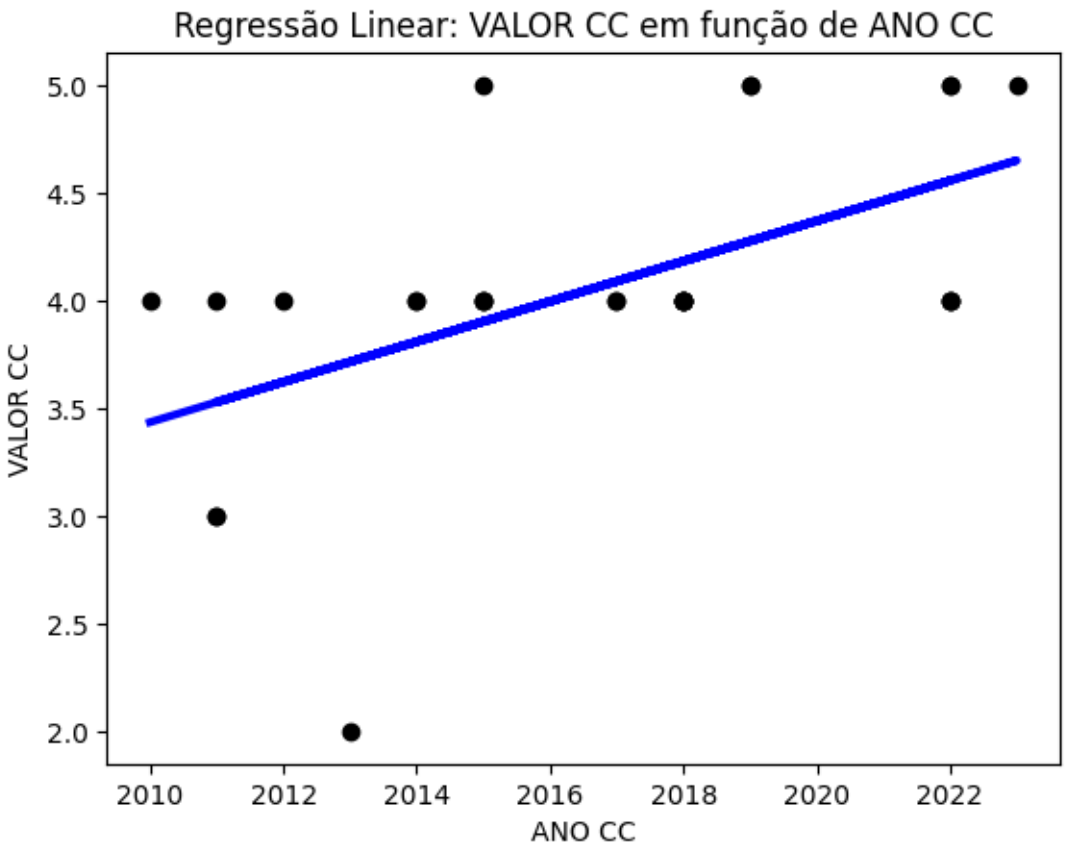


[41]:
```
# Visualiza a árvore de decisão
tree_rules = export_text(model_enade, feature_names=['ENADE ANO'])
print(tree_rules)
```

```
|--- ENADE ANO <= 2011.50
|   |--- value: [0.00]
|--- ENADE ANO >  2011.50
|   |--- ENADE ANO <= 2018.50
|   |   |--- ENADE ANO <= 2015.50
|   |   |   |--- value: [4.67]
|   |   |--- ENADE ANO >  2015.50
|   |   |   |--- ENADE ANO <= 2017.50
|   |   |   |   |--- value: [3.50]
|   |   |   |--- ENADE ANO >  2017.50
|   |   |   |   |--- value: [4.40]
|   |--- ENADE ANO >  2018.50
|   |   |--- value: [3.70]
```

2.2 REGRESSÃO LINEAR E ARVORE DE DECISÃO CC

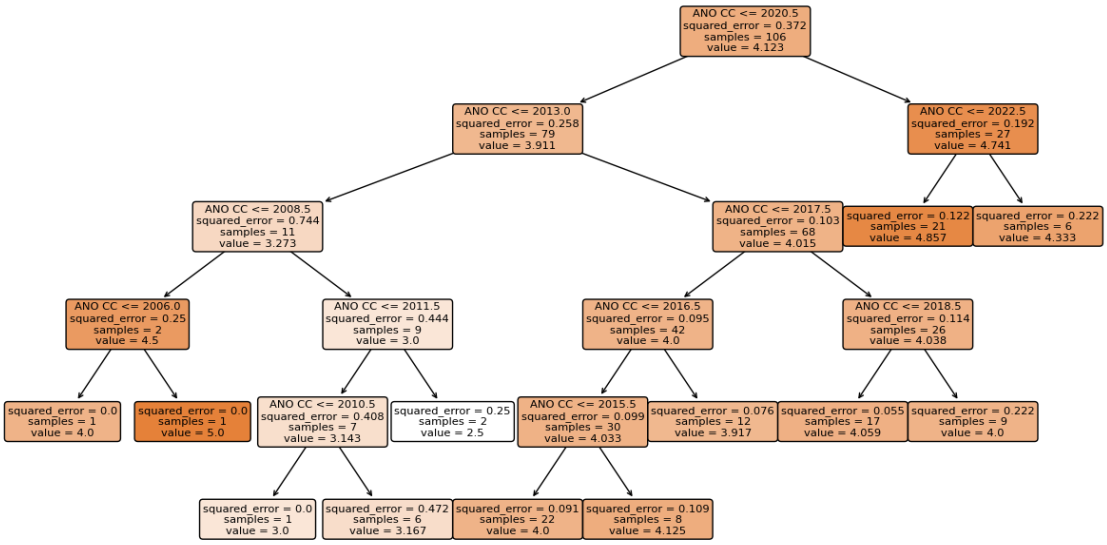
[70]:
```
dados_interesse_cc = df[['ANO CC', 'VALOR CC']]

dados_interesse_cc = dados_interesse_cc.dropna()

X = dados_interesse_cc[['ANO CC']]
y = dados_interesse_cc['VALOR CC']
X_train_cc, X_test_cc, y_train_cc, y_test_cc = train_test_split(X, y,␣
 ↪test_size=0.2, random_state=42)

model_cc = LinearRegression()

model_cc.fit(X_train_cc, y_train_cc)

y_pred_cc = model_cc.predict(X_test_cc)

plt.scatter(X_test_cc, y_test_cc, color='black')
plt.plot(X_test_cc, y_pred_cc, color='blue', linewidth=3)
plt.xlabel('ANO CC')
plt.ylabel('VALOR CC')
plt.title('Regressão Linear: VALOR CC em função de ANO CC')
plt.show()
```



[71]:
```
dados_interesse_cc = df[['ANO CC', 'VALOR CC']]

dados_interesse_cc = dados_interesse_cc.dropna()

X = dados_interesse_cc[['ANO CC']]
y = dados_interesse_cc['VALOR CC']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

model_cc = DecisionTreeRegressor()

model_cc.fit(X_train, y_train)

y_pred = model_cc.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
print(f'Mean Squared Error para VALOR CC: {mse}')

plt.figure(figsize=(15, 8))
plot_tree(model_cc, feature_names=X.columns, filled=True, rounded=True)
```

```
plt.show()
```

```
Mean Squared Error para VALOR CC: 0.3692789752066403
```



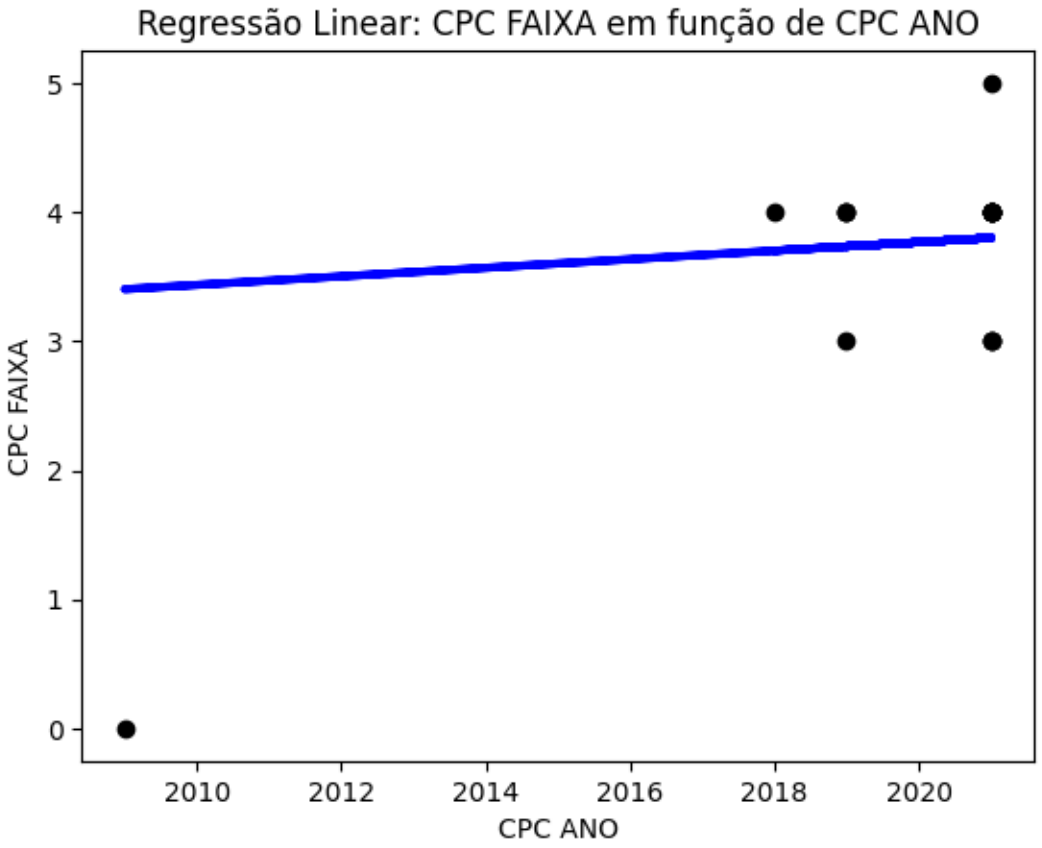
[ ]:
```
tree_rules = export_text(model_cc, feature_names=['VALOR CC'])
print(tree_rules)
```

```
|--- VALOR CC <= 2020.00
|   |--- VALOR CC <= 2013.00
|   |   |--- value: [4.00]
|   |--- VALOR CC >  2013.00
|   |   |--- VALOR CC <= 2017.50
|   |   |   |--- value: [3.00]
|   |   |--- VALOR CC >  2017.50
|   |   |   |--- VALOR CC <= 2018.50
|   |   |   |   |--- value: [3.75]
|   |   |   |--- VALOR CC >  2018.50
|   |   |   |   |--- value: [3.56]
|--- VALOR CC >  2020.00
|   |--- value: [3.83]
```

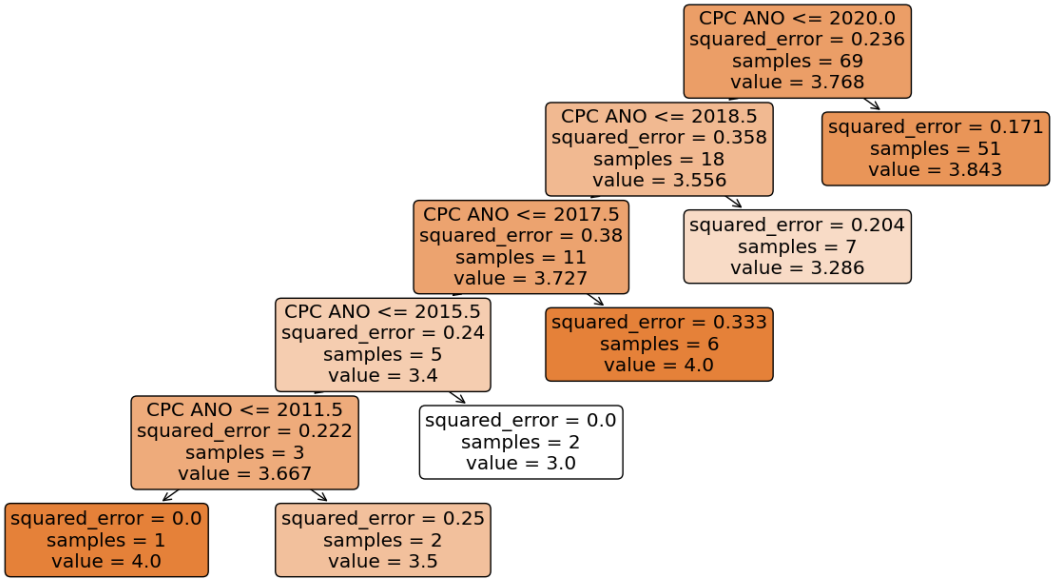
2.3 REGRESSÃO LINEAR E ARVORE DE DESEMPENHO CPC

[51]:
```
plt.scatter(X_test_cpc, y_test_cpc, color='black')
plt.plot(X_test_cpc, y_pred_cpc, color='blue', linewidth=3)
plt.xlabel('CPC ANO')
plt.ylabel('CPC FAIXA')
```

```
plt.title('Regressão Linear: CPC FAIXA em função de CPC ANO')
plt.show()
```



[56]:
```
plt.figure(figsize=(15, 8))
plot_tree(model_cpc, feature_names=X_train_cpc.columns, filled=True,␣
↪rounded=True)
plt.show()
```



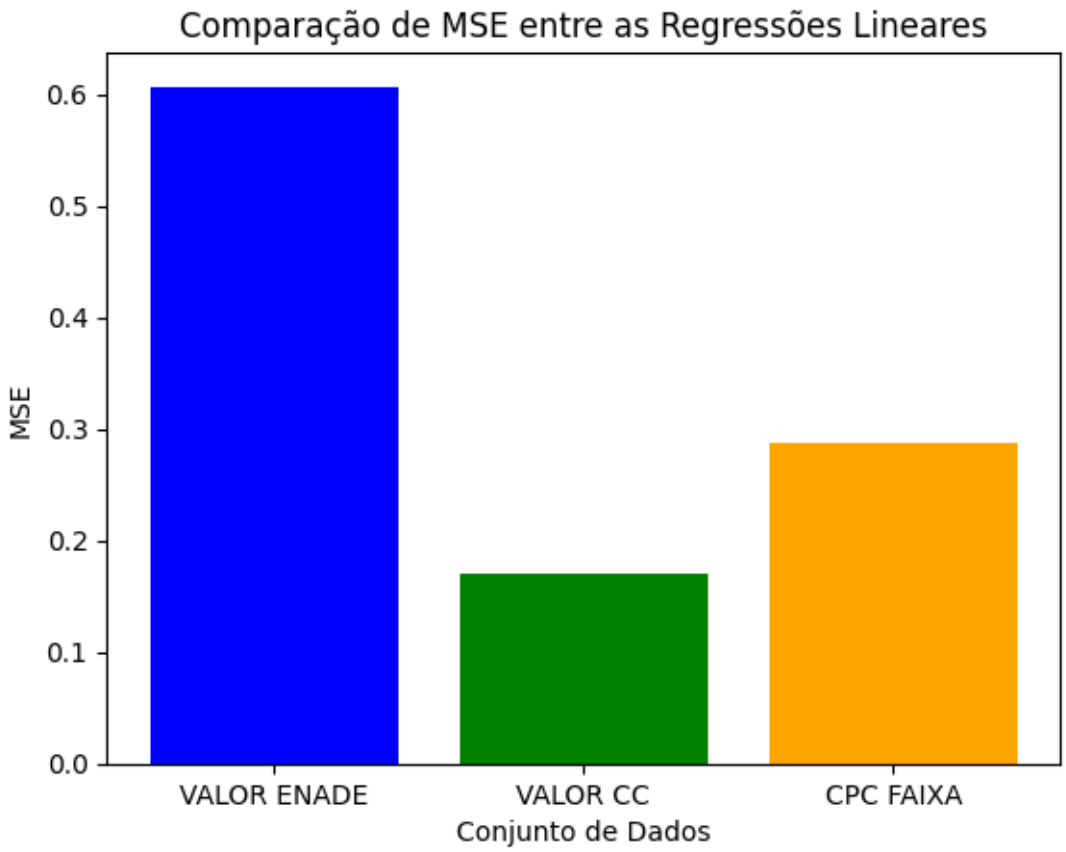
[57]:
```
tree_rules = export_text(model_cpc, feature_names=['CPC ANO'])
print("Regras da Árvore de Decisão:")
print(tree_rules)
```

```
Regras da Árvore de Decisão:
|--- CPC ANO <= 2020.00
|   |--- CPC ANO <= 2018.50
|   |   |--- CPC ANO <= 2017.50
|   |   |   |--- CPC ANO <= 2015.50
|   |   |   |   |--- CPC ANO <= 2011.50
|   |   |   |   |   |--- value: [4.00]
|   |   |   |   |--- CPC ANO >  2011.50
|   |   |   |   |   |--- value: [3.50]
|   |   |   |--- CPC ANO >  2015.50
|   |   |   |   |--- value: [3.00]
|   |   |--- CPC ANO >  2017.50
|   |   |   |--- value: [4.00]
|   |--- CPC ANO >  2018.50
|   |   |--- value: [3.29]
|--- CPC ANO >  2020.00
|   |--- value: [3.84]
```

2.4 COMPARAÇÃO ENTRE AS 3 VARIAVEIS - REGRESSÃO LINEAR E A ARVORE DE DECISÃO

[ ]:
```
# Exibe os valores de MSE
print(f'MSE para VALOR ENADE: {mse_enade}')
print(f'MSE para VALOR CC: {mse_cc}')
print(f'MSE para CPC FAIXA: {mse_cpc}')

# Compara os valores de MSE
melhor_modelo = min(mse_enade, mse_cc, mse_cpc)
print(f'O melhor modelo tem MSE: {melhor_modelo}')

# Exibe gráfico comparativo
plt.bar(['VALOR ENADE', 'VALOR CC', 'CPC FAIXA'], [mse_enade, mse_cc, mse_cpc],␣
 ↪color=['blue', 'green', 'orange'])
plt.xlabel('Conjunto de Dados')
plt.ylabel('MSE')
plt.title('Comparação de MSE entre as Regressões Lineares')
plt.show()
```

```
MSE para VALOR ENADE: 0.606803705218063
MSE para VALOR CC: 0.16987955619779882
MSE para CPC FAIXA: 0.28798931013154977
O melhor modelo tem MSE: 0.16987955619779882
```



[ ]:
```
# Exibe os valores de MSE
print(f'MSE para VALOR ENADE: {mse_enade}')
print(f'MSE para VALOR CC: {mse_cc}')
print(f'MSE para CPC FAIXA: {mse_cpc}')

# Compara os valores de MSE
melhor_modelo_arvore = min(mse_enade, mse_cc, mse_cpc)
print(f'A árvore com menor MSE é: {melhor_modelo_arvore}')

# Visualiza gráfico comparativo
plt.bar(['VALOR ENADE', 'VALOR CC', 'CPC FAIXA'], [mse_enade, mse_cc, mse_cpc],␣
  ↪color=['blue', 'green', 'orange'])
plt.xlabel('Conjunto de Dados')
plt.ylabel('MSE')
plt.title('Comparação de MSE entre as Árvores de Decisão')
plt.show()
```

```
MSE para VALOR ENADE: 0.5540097019475367
MSE para VALOR CC: 0.3077810551303855
MSE para CPC FAIXA: 0.24636468689467664
```

```
A árvore com menor MSE é: 0.24636468689467664
```