Análise exploratória das notas dos filmes

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import math
```

In [2]:
```
notas = pd.read_csv("dados/ratings.csv")
notas.columns = ["usuarioId", "filmeId", "nota", "momento"]
filmes = pd.read_csv("dados/movies.csv")
filmes.columns = ["filmeId", "titulo", "generos"]
```

In [3]:
```
notas.head()
```

Out[3]:

	usuarioId	filmeId	nota	momento
0	1	1	4.0	964982703
1	1	3	4.0	964981247
2	1	6	4.0	964982224
3	1	47	5.0	964983815
4	1	50	5.0	964982931

In [4]:
```
filmes.head()
```

Out[4]:

	filmeId	titulo	generos
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

Análise de notas por gênero

In [5]:
```
generos_de_filmes = filmes.generos
generos_de_filmes.drop_duplicates(inplace=True)
generos = set()
for genero in generos_de_filmes.values:
    for aux in genero.split("|"):
        generos.add(aux)
generos_de_filmes = pd.Series(list(generos))
generos_de_filmes
```

Out[5]:
```
0                Fantasy
1                Mystery
2                 Comedy
3              Film-Noir
4                 Action
5                 Sci-Fi
6                    War
7                Western
8               Children
9              Animation
```

```
10                 Crime
11    (no genres listed)
12             Adventure
13                Horror
14           Documentary
15              Thriller
16               Romance
17                  IMAX
18               Musical
19                 Drama
dtype: object
```

In [6]:
```
dados = {}
for genero in generos_de_filmes.values:
    dados[genero] = []
    filmes_selecionados = filmes[filmes["generos"].str.contains(genero, na=False)]
    for selecionado in filmes_selecionados.filmeId.values:
        dados[genero].extend(notas.query(f"filmeId == {selecionado}").nota.values)
```

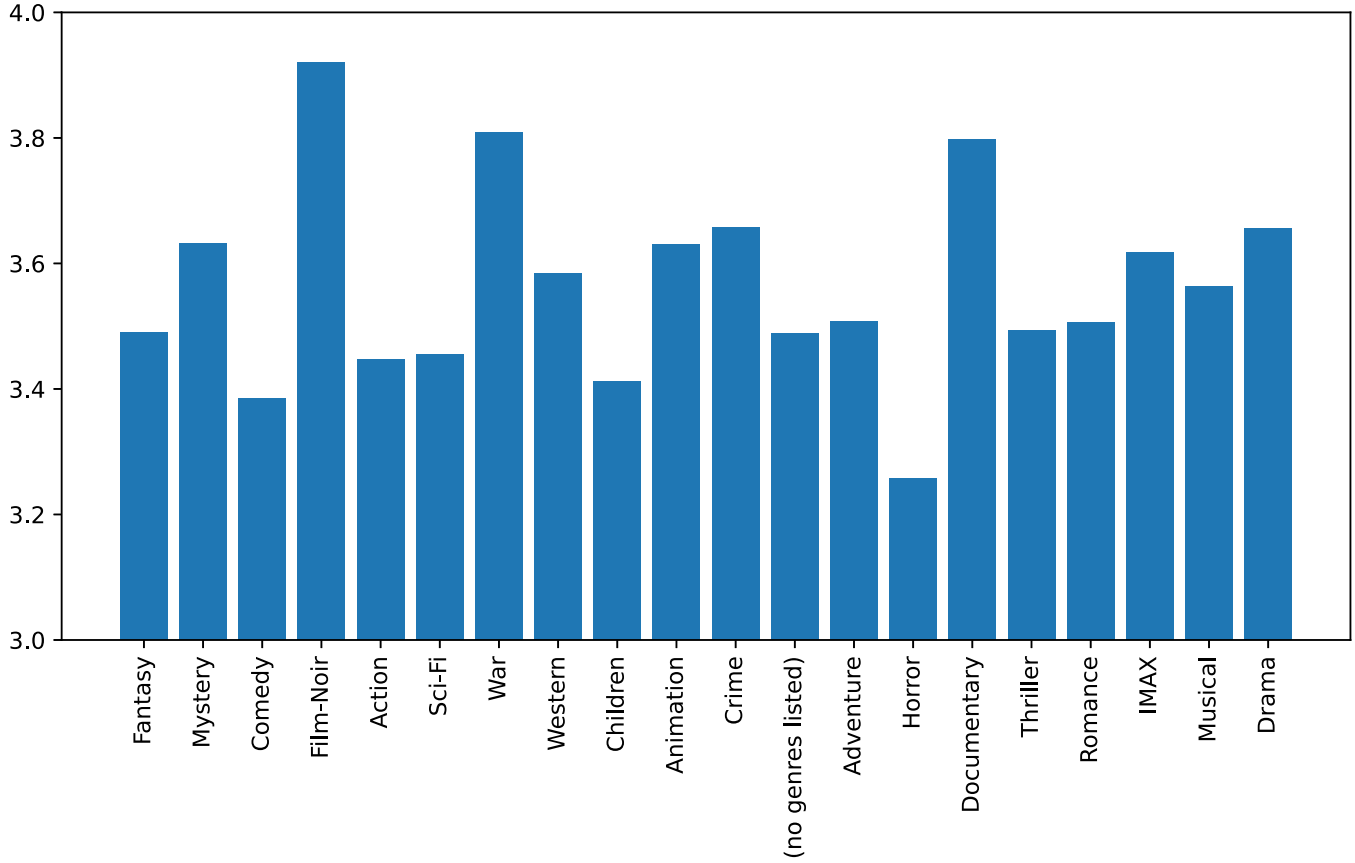
In [7]:
```
for key, value in dados.items():
    dados[key] = pd.Series(value)

medias = {key: value.mean() for key, value in dados.items()}
medias = pd.DataFrame.from_dict(medias, orient='index', columns=['media'])
medianas = {key: value.median() for key, value in dados.items()}
medianas = pd.DataFrame.from_dict(medianas, orient='index', columns=['mediana'])
desvio_padrao = {key: value.std() for key, value in dados.items()}
desvio_padrao = pd.DataFrame.from_dict(desvio_padrao, orient='index', columns=['desvio'])
```

In [8]:
```
generos_ = list(medias.index)
medias_ = [v[0] for v in medias.values]
medianas_ = [v[0] for v in medianas.values]
desvio_padrao_ = [v[0] for v in desvio_padrao.values]
```

Médias das notas por gênero

In [9]:
```
plt.figure(figsize=(10, 5))
plt.xticks(rotation=90)
low = min(medias_)
high = max(medias_)
plt.ylim([math.ceil(low-0.5*(high-low)), math.ceil(high-0.5*(high-low))])
plt.bar(generos_, medias_)
```

Out[9]:
```
<BarContainer object of 20 artists>
```

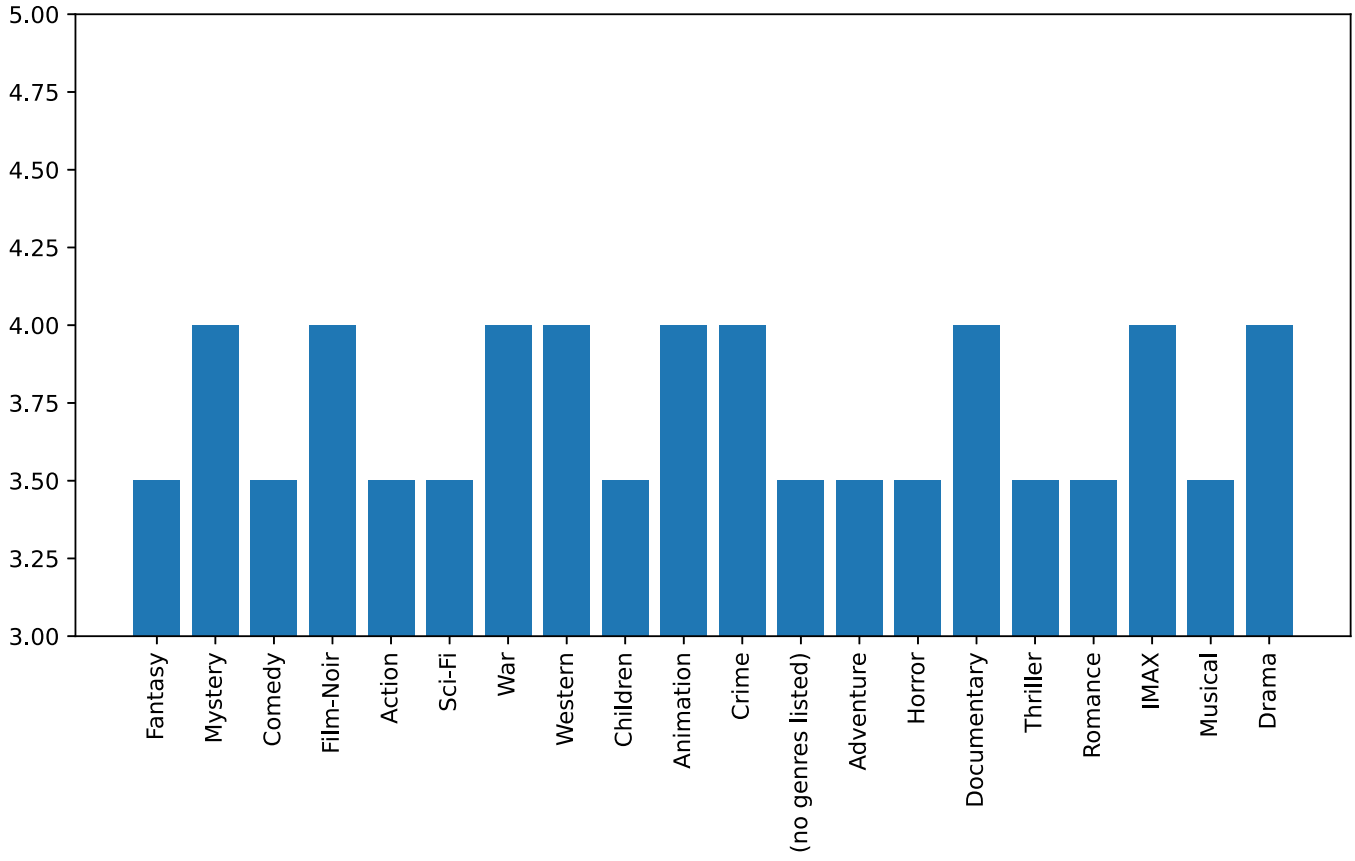


Mediana das notas por gênero

In [10]:

```
plt.figure(figsize=(10, 5))
plt.xticks(rotation=90)
low = min(medianas_)
high = max(medianas_)
plt.ylim(3, 5)
plt.bar(generos_, medianas_)
```

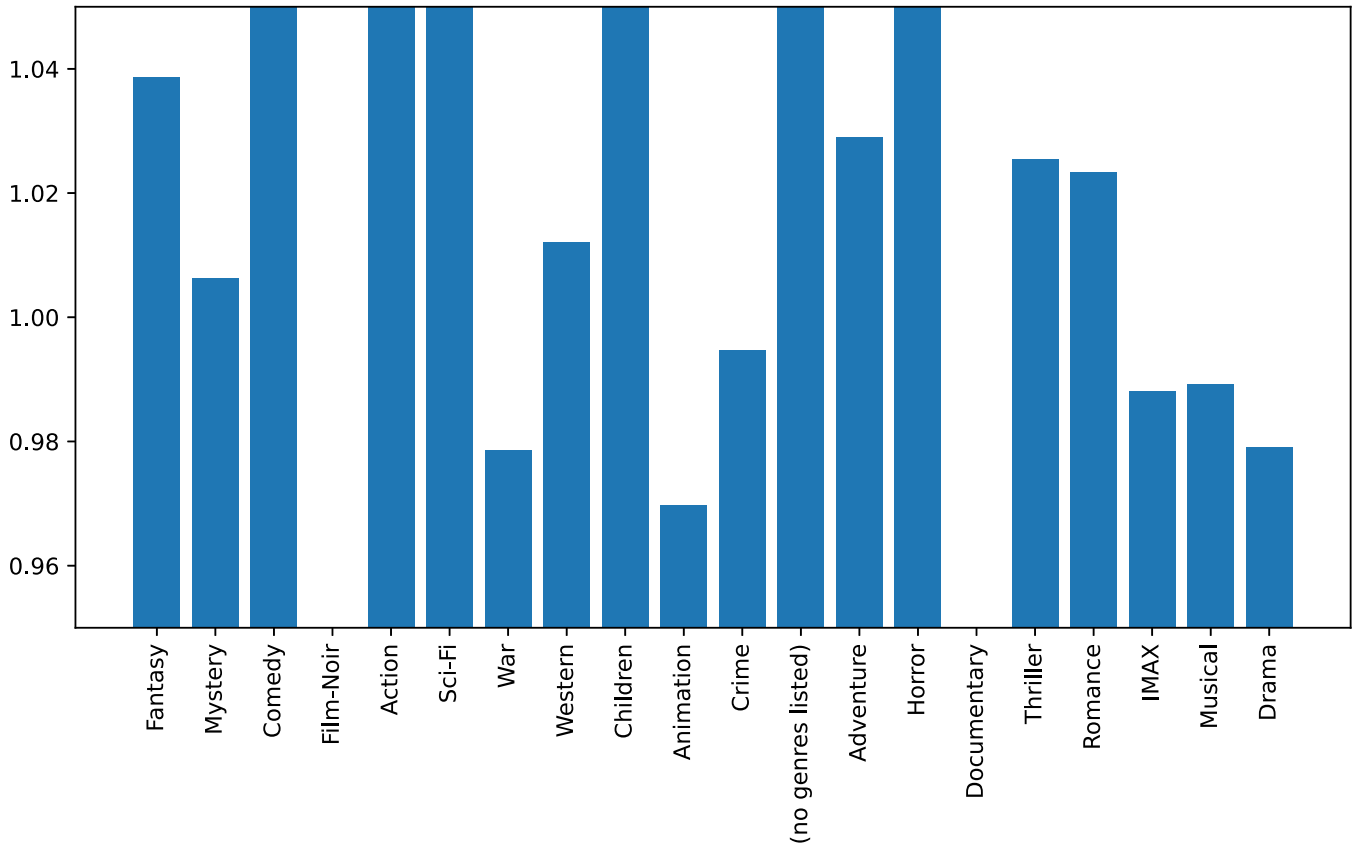
Out[10]: <BarContainer object of 20 artists>



Desvio padrão das notas por gênero

In [11]:
```
plt.figure(figsize=(10, 5))
plt.xticks(rotation=90)
low = min(desvio_padrao_)
high = max(desvio_padrao_)
plt.ylim([math.ceil(low*(high-low)), math.ceil(high*(high-low))])
plt.bar(generos_, desvio_padrao_)
```

Out[11]: <BarContainer object of 20 artists>



In [ ]: