Analisando as notas em geral

In [1]:
```
import pandas as pd

notas = pd.read_csv("dados/ratings.csv")
notas.head()
```

Out[1]:

	userId	movieId	rating	timestamp
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 [2]:
```
notas.shape
```

Out[2]:
```
(100836, 4)
```

In [3]:
```
notas.columns = ["usuarioId", "filmeId", "nota", "momento"]
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]:
```
notas['nota'].unique()
```

Out[4]:
```
array([4. , 5. , 3. , 2. , 1. , 4.5, 3.5, 2.5, 0.5, 1.5])
```

In [5]:
```
notas['nota'].value_counts()
```

Out[5]:
```
4.0    26818
3.0    20047
5.0    13211
3.5    13136
4.5     8551
2.0     7551
2.5     5550
1.0     2811
1.5     1791
0.5     1370
Name: nota, dtype: int64
```

In [6]:
```
print("Media",notas['nota'].mean())
print("Mediana",notas['nota'].median())
```

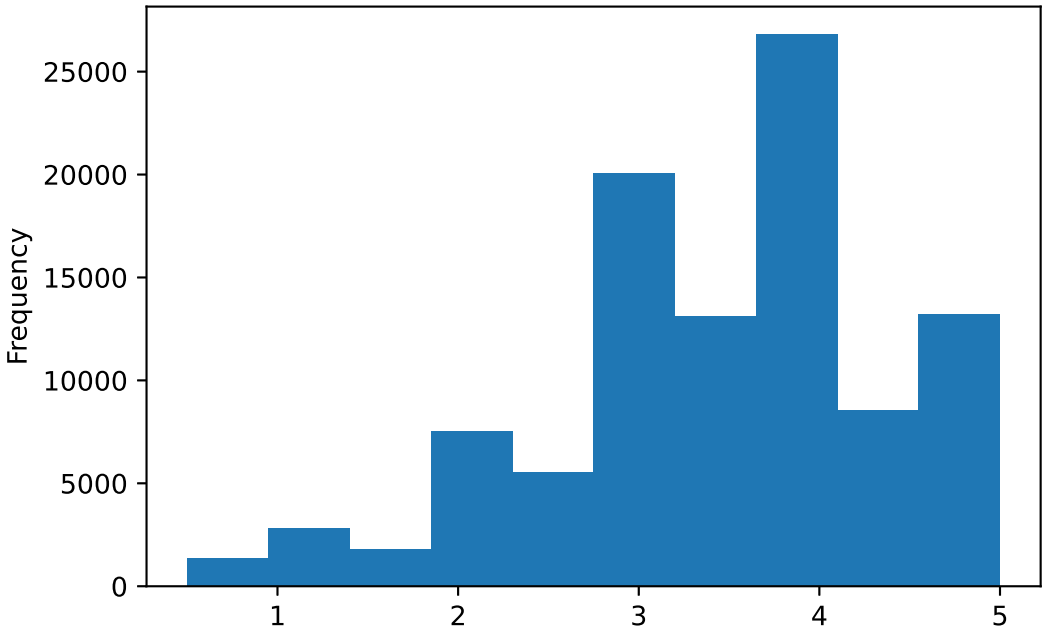
```
Media 3.501556983616962
Mediana 3.5
```

In [7]:
```
notas.nota.head()
```

Out[7]:
```
0    4.0
1    4.0
2    4.0
3    5.0
4    5.0
Name: nota, dtype: float64
```

In [8]:
```
notas.nota.plot(kind='hist')
```

Out[8]:
```
<AxesSubplot:ylabel='Frequency'>
```



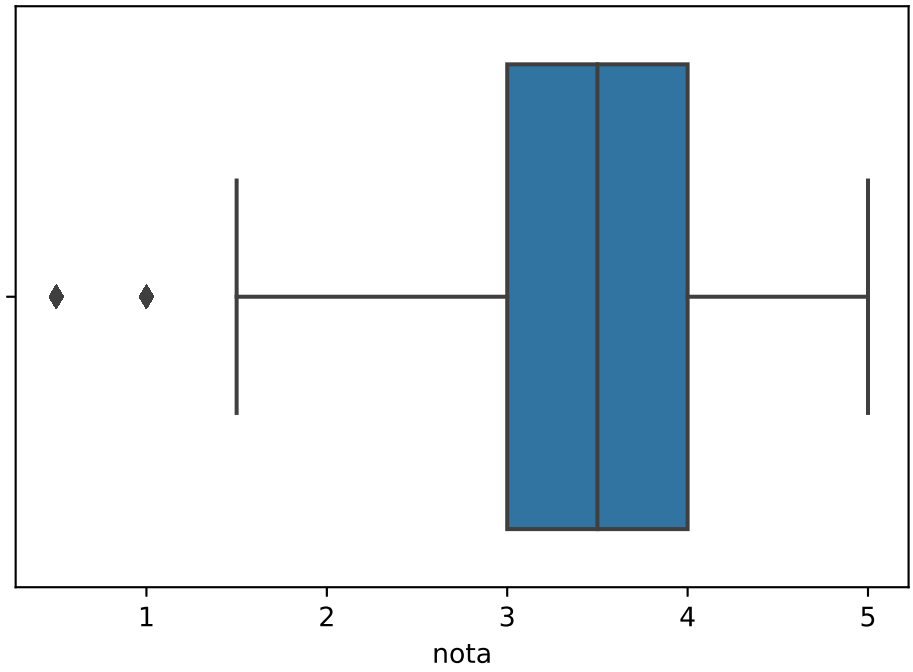
In [9]:
```
notas.nota.describe()
```

Out[9]:
```
count    100836.000000
mean          3.501557
std           1.042529
min           0.500000
25%           3.000000
50%           3.500000
75%           4.000000
max           5.000000
Name: nota, dtype: float64
```

In [10]:
```
import seaborn as sns

sns.boxplot(notas.nota)
```

Out[10]:
```
<AxesSubplot:xlabel='nota'>
```



Olhando os filmes

In [11]:
```
filmes = pd.read_csv("dados/movies.csv")
filmes.columns = ["filmeId", "titulo", "generos"]
filmes.head()
```

Out[11]:

	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

In [12]:
```
notas.head()
```

Out[12]:

	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

Analisando algumas notas especificas por filme

In [13]:
```
notas.query("filmeId==1").nota.mean()
```

Out[13]:
```
3.9209302325581397
```

In [14]:
```
notas.query("filmeId==2").nota.mean()
```

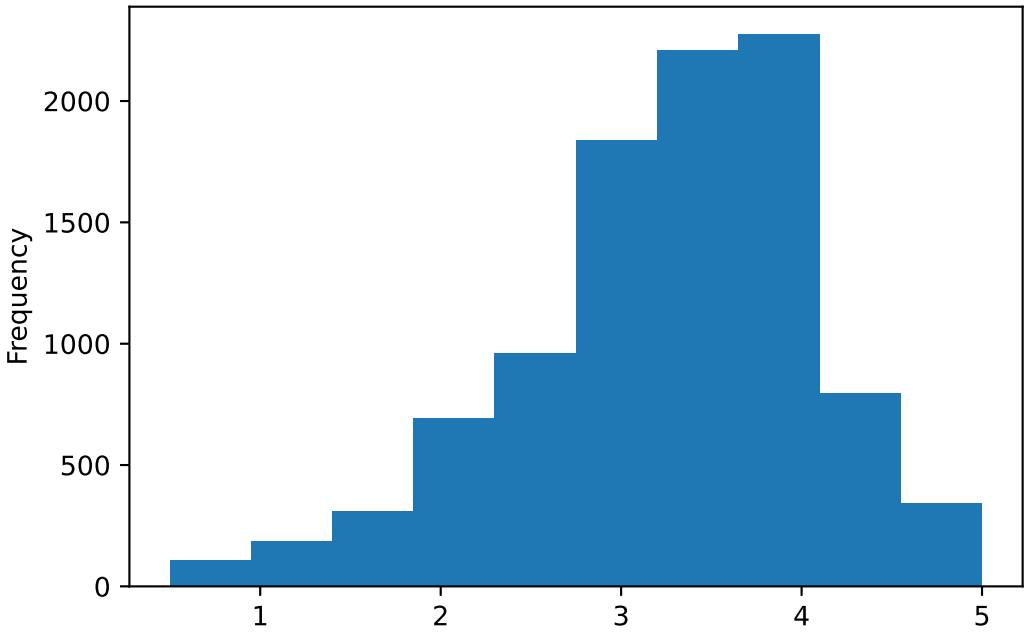
Out[14]:
```
3.4318181818181817
```

In [15]:
```
medias_por_filme = notas.groupby("filmeId").mean().nota
medias_por_filme.head()
```

Out[15]:
```
filmeId
1    3.920930
2    3.431818
3    3.259615
4    2.357143
5    3.071429
Name: nota, dtype: float64
```

In [16]:
```
medias_por_filme.plot(kind='hist')
```

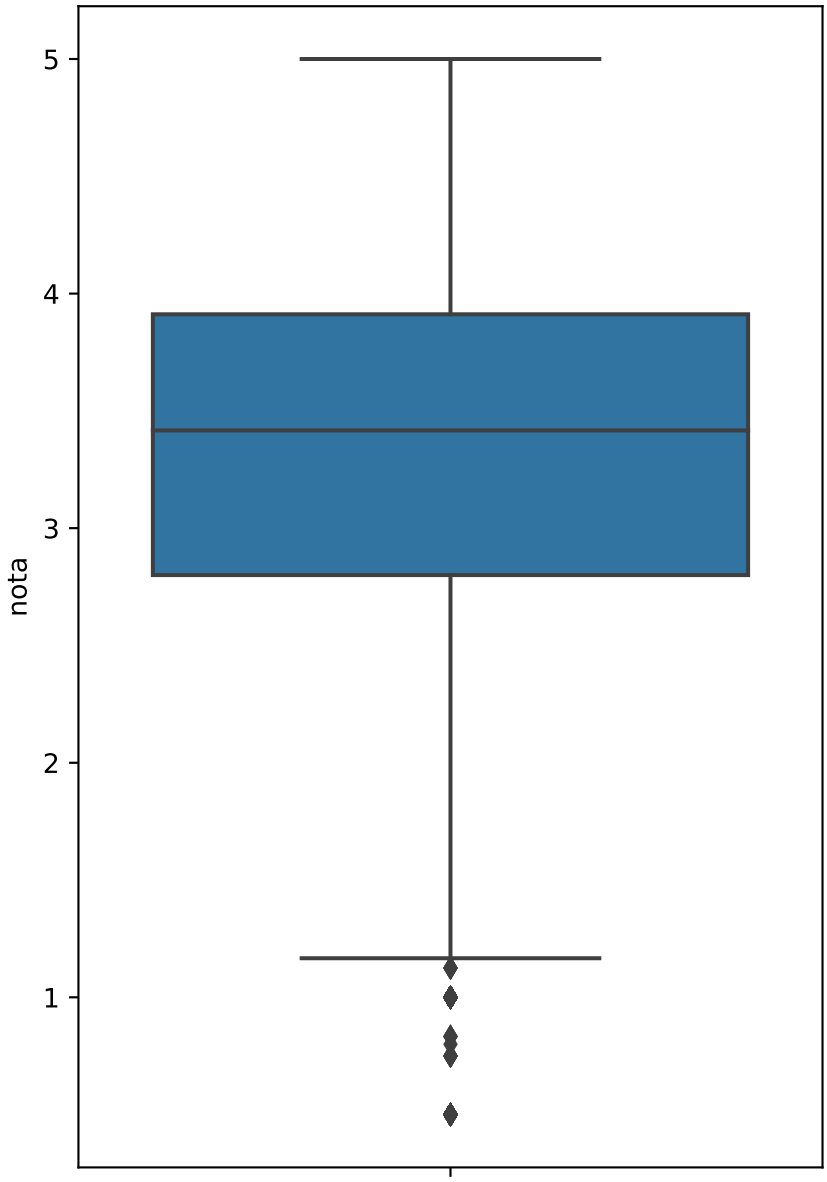
Out[16]:
```
<AxesSubplot:ylabel='Frequency'>
```



In [17]:
```
import matplotlib.pyplot as plt

plt.figure(figsize=(5,8))
sns.boxplot(y=medias_por_filme)
```

Out[17]:
```
<AxesSubplot:ylabel='nota'>
```

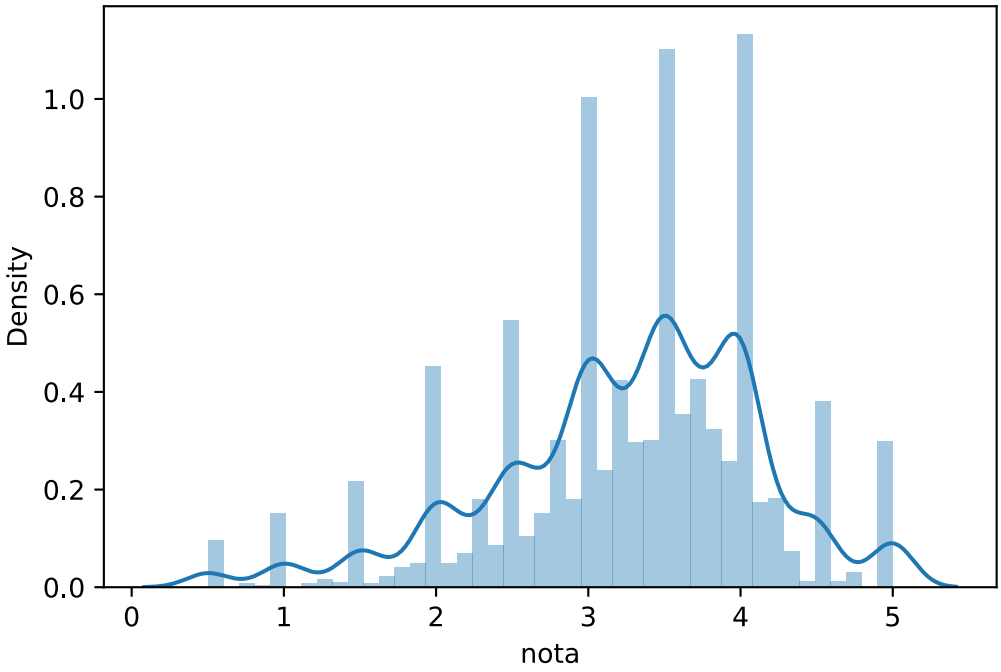


In [18]:
```
medias_por_filme.describe()
```

Out[18]:
```
count    9724.000000
mean        3.262448
std         0.869874
min         0.500000
25%         2.800000
50%         3.416667
75%         3.911765
max         5.000000
Name: nota, dtype: float64
```

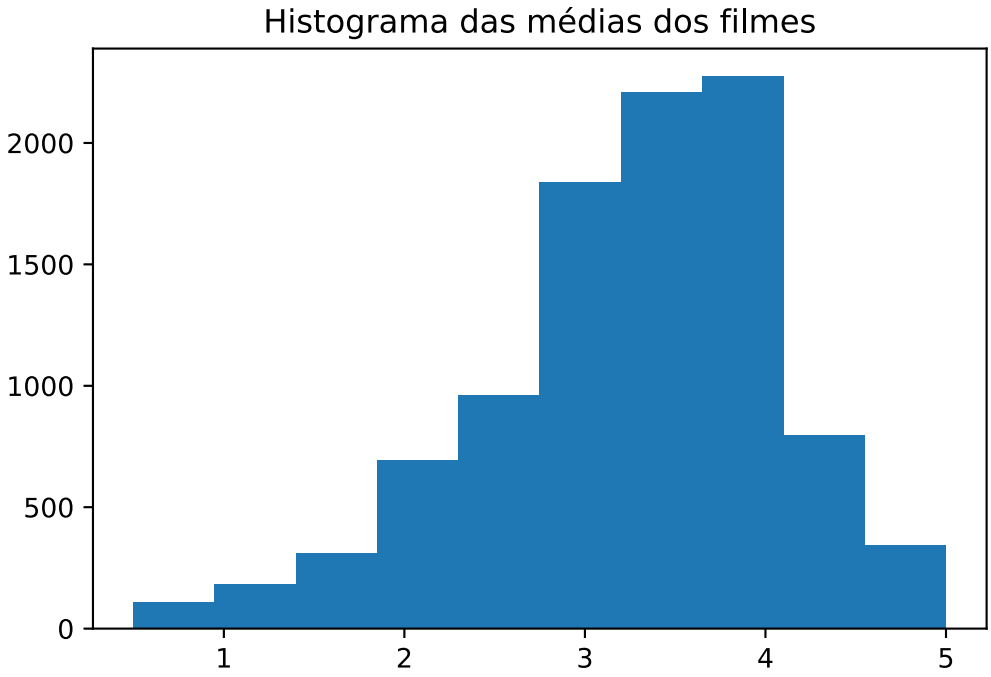
In [19]:
```
sns.distplot(medias_por_filme)
```

Out[19]:
```
<AxesSubplot:xlabel='nota', ylabel='Density'>
```



In [20]:
```
plt.hist(medias_por_filme)
plt.title("Histograma das médias dos filmes")
```

Out[20]:
```
Text(0.5, 1.0, 'Histograma das médias dos filmes')
```



In [21]:
```
tmdb = pd.read_csv("dados/tmdb_5000_movies.csv")
tmdb.head()
```

Out[21]:

	budget	genres	homepage	id	keywords	original_language	orig
0	237000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.avatarmovie.com/	19995	[{"id": 1463, "name": "culture clash"}, {"id":...	en	

	budget	genres	homepage	id	keywords	original_language	orig
1	300000000	[{"id": 12, "name": "Adventure"}, {"id": 14, "...	http://disney.go.com/disneypictures/pirates/	285	[{"id": 270, "name": "ocean"}, {"id": 726, "na...	en	Pira Cari W
2	245000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.sonypictures.com/movies/spectre/	206647	[{"id": 470, "name": "spy"}, {"id": 818, "name...	en	
3	250000000	[{"id": 28, "name": "Action"}, {"id": 80, "nam...	http://www.thedarkknightrises.com/	49026	[{"id": 849, "name": "dc comics"}, {"id": 853,...	en	Kn
4	260000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://movies.disney.com/john-carter	49529	[{"id": 818, "name": "based on novel"}, {"id":...	en	Jc

In [22]:
```
tmdb.original_language.unique() # categorica nominal
```

Out[22]:
```
array(['en', 'ja', 'fr', 'zh', 'es', 'de', 'hi', 'ru', 'ko', 'te', 'cn',
       'it', 'nl', 'ta', 'sv', 'th', 'da', 'xx', 'hu', 'cs', 'pt', 'is',
       'tr', 'nb', 'af', 'pl', 'he', 'ar', 'vi', 'ky', 'id', 'ro', 'fa',
       'no', 'sl', 'ps', 'el'], dtype=object)
```

In [23]:
```
tmdb["original_language"].value_counts().index
```

Out[23]:
```
Index(['en', 'fr', 'es', 'zh', 'de', 'hi', 'ja', 'it', 'cn', 'ru', 'ko', 'pt',
       'da', 'sv', 'nl', 'fa', 'he', 'th', 'ar', 'ta', 'id', 'cs', 'ro', 'pl',
       'is', 'el', 'no', 'tr', 'ky', 'ps', 'te', 'vi', 'hu', 'sl', 'xx', 'af',
       'nb'],
      dtype='object')
```

In [24]:
```
tmdb["original_language"].value_counts().values
```

Out[24]:
```
array([4505,   70,   32,   27,   27,   19,   16,   14,   12,   11,   11,
          9,    7,    5,    4,    4,    3,    3,    2,    2,    2,    2,
          2,    1,    1,    1,    1,    1,    1,    1,    1,    1,    1,
          1,    1,    1,    1], dtype=int64)
```

In [25]:
```
contagem_de_lingua = tmdb["original_language"].value_counts().to_frame().reset_index()
contagem_de_lingua.columns = ["original_language", "total"]
contagem_de_lingua.head()
```

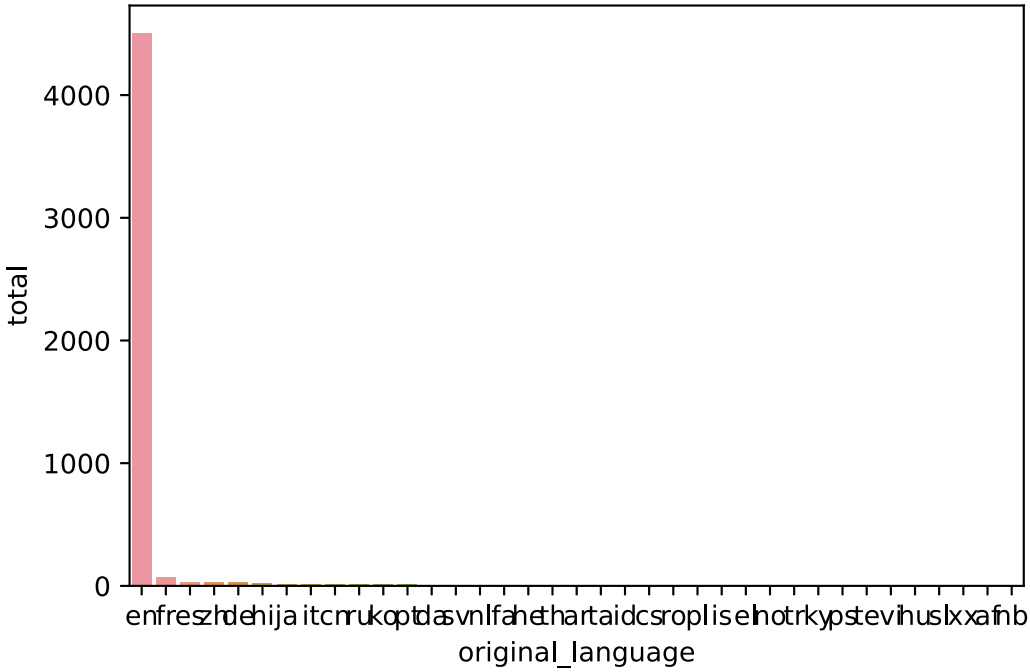
Out[25]:

	original_language	total
0	en	4505
1	fr	70
2	es	32

	original_language	total
3	zh	27
4	de	27

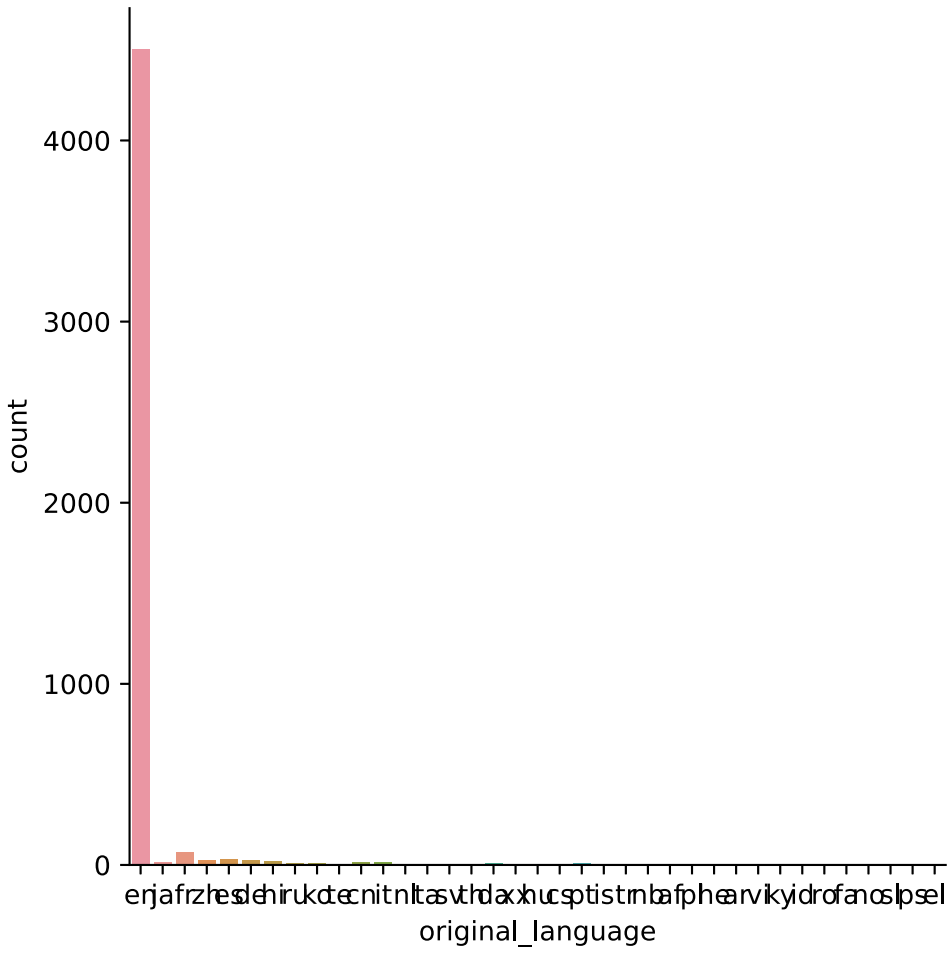
In [26]:
```
sns.barplot(x="original_language", y = "total", data = contagem_de_lingua)
```

Out[26]:
```
<AxesSubplot:xlabel='original_language', ylabel='total'>
```



In [27]:
```
sns.catplot(x = "original_language", kind="count", data = tmdb)
```

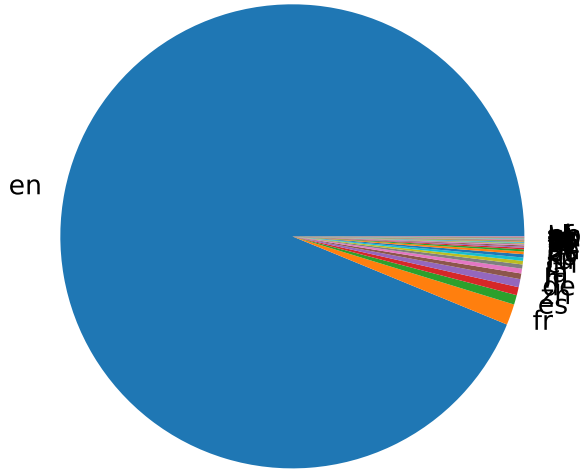
Out[27]:
```
<seaborn.axisgrid.FacetGrid at 0xd816160>
```



In [28]:
```
plt.pie(contagem_de_lingua["total"], labels = contagem_de_lingua["original_language"])
```

Out[28]:
```
([<matplotlib.patches.Wedge at 0xc5d5580>,
  <matplotlib.patches.Wedge at 0xce49190>,
  <matplotlib.patches.Wedge at 0xce49f70>,
  <matplotlib.patches.Wedge at 0xcf07d30>,
  <matplotlib.patches.Wedge at 0xcea9970>,
  <matplotlib.patches.Wedge at 0xce37880>,
  <matplotlib.patches.Wedge at 0xc27fe50>,
  <matplotlib.patches.Wedge at 0xcf35e20>,
  <matplotlib.patches.Wedge at 0xcf9f2e0>,
  <matplotlib.patches.Wedge at 0xcf9fd90>,
  <matplotlib.patches.Wedge at 0xc5b2be0>,
  <matplotlib.patches.Wedge at 0xc273c10>,
  <matplotlib.patches.Wedge at 0xce4c100>,
  <matplotlib.patches.Wedge at 0xce23160>,
  <matplotlib.patches.Wedge at 0xce41c70>,
  <matplotlib.patches.Wedge at 0xce41910>,
  <matplotlib.patches.Wedge at 0xcfae6a0>,
  <matplotlib.patches.Wedge at 0xc2854f0>,
  <matplotlib.patches.Wedge at 0xcea4dc0>,
  <matplotlib.patches.Wedge at 0xcea4400>,
  <matplotlib.patches.Wedge at 0xcf54340>,
  <matplotlib.patches.Wedge at 0xcf8b0a0>,
  <matplotlib.patches.Wedge at 0xd756460>,
  <matplotlib.patches.Wedge at 0xcf85e20>,
  <matplotlib.patches.Wedge at 0xcf855e0>,
  <matplotlib.patches.Wedge at 0xd7cc130>,
  <matplotlib.patches.Wedge at 0xd736d30>,
  <matplotlib.patches.Wedge at 0xe55da30>,
  <matplotlib.patches.Wedge at 0xd7485e0>,
  <matplotlib.patches.Wedge at 0xce9cf40>,
  <matplotlib.patches.Wedge at 0xce9c0d0>,
  <matplotlib.patches.Wedge at 0xcfe5670>,
  <matplotlib.patches.Wedge at 0xcf9e190>,
  <matplotlib.patches.Wedge at 0xd728f70>,
  <matplotlib.patches.Wedge at 0xe54e7c0>,
  <matplotlib.patches.Wedge at 0xd8132e0>,
  <matplotlib.patches.Wedge at 0xd813be0>],
 [Text(-1.0791697536499925, 0.2130554923183512, 'en'),
  Text(1.0355355017029462, -0.3710339940124459, 'fr'),
  Text(1.0579676486019882, -0.3011718023181785, 'es'),
  Text(1.0687996606645356, -0.26012936274741094, 'zh'),
  Text(1.0773191105706255, -0.22222406260195313, 'de'),
  Text(1.0835167978583342, -0.18971386021801853, 'hi'),
  Text(1.0875756432724297, -0.16486121484618815, 'ja'),
  Text(1.0906010773146022, -0.14348968659882622, 'it'),
  Text(1.092883487371409, -0.12492270822755745, 'cn'),
  Text(1.0946390911069936, -0.10846778425161549, 'ru'),
  Text(1.0960865535188649, -0.09270527058984593, 'ko'),
  Text(1.0972054830031333, -0.07835896928789601, 'pt'),
  Text(1.097965443340663, -0.06687215586282344, 'da'),
  Text(1.0984565010300316, -0.05825217030171998, 'sv'),
  Text(1.0987803851616647, -0.0517847968421653, 'nl'),
  Text(1.0990363161210686, -0.04603450713357274, 'fa'),
  Text(1.0992355702663055, -0.04100196411527794, 'he'),
  Text(1.0993880184234357, -0.03668766750546649, 'th'),
  Text(1.0995021239019234, -0.033091985965784415, 'ar'),
  Text(1.099584941078101, -0.03021518416739545, 'ta'),
  Text(1.0996602312343366, -0.027338175536150495, 'id'),
  Text(1.099727993855245, -0.024460979766119193, 'cs'),
  Text(1.0997882284769684, -0.02158361655264929, 'ro'),
  Text(1.0998284639438185, -0.01942549610642471, 'pl'),
  Text(1.0998529348820232, -0.01798670707495573, 'is'),
  Text(1.0998755236058106, -0.01654788726224571, 'el'),
  Text(1.0998962300765243, -0.01510903913059323, 'no'),
  Text(1.0999150542587282, -0.013670165142345335, 'tr'),
  Text(1.0999319961202083, -0.012231267759896247, 'ky'),
  Text(1.0999470556319713, -0.01079234944567632, 'ps'),
  Text(1.099960232768245, -0.00935341266215563, 'te'),
  Text(1.0999715275064792, -0.007914459871831963, 'vi'),
  Text(1.0999809398273452, -0.006475493537234394, 'hu'),
```

```
Text(1.0999884697147349, -0.005036516120911278, 'sl'),
Text(1.0999941171557621, -0.0035975300854338356, 'xx'),
Text(1.0999978821407626, -0.0021585378933851127, 'af'),
Text(1.0999997646632929, -0.0007195420073586872, 'nb')])
```



In [29]:

```
total_por_lingua = tmdb["original_language"].value_counts()
total_geral = total_por_lingua.sum()
total_de_ingles = total_por_lingua.loc["en"]
total_do_resto = total_geral - total_de_ingles
print(total_de_ingles, total_do_resto)
```

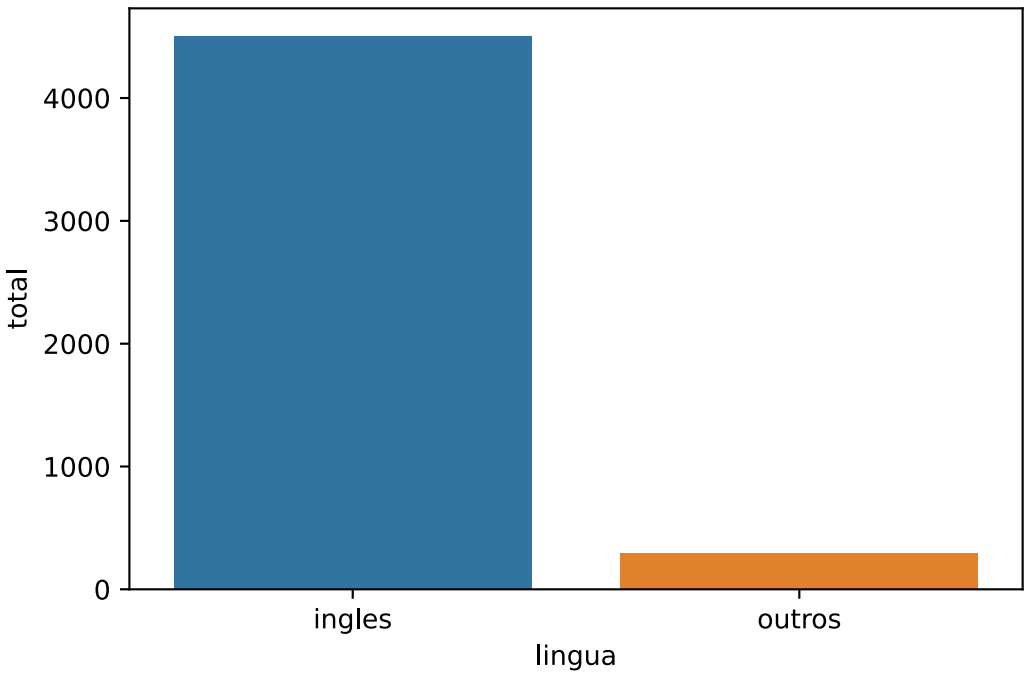
```
4505 298
```

In [30]:

```
dados = {
    'lingua' : ['ingles','outros'],
    'total' : [total_de_ingles, total_do_resto]
}
dados = pd.DataFrame(dados)
sns.barplot(x="lingua", y="total", data = dados)
```

Out[30]:

```
<AxesSubplot:xlabel='lingua', ylabel='total'>
```



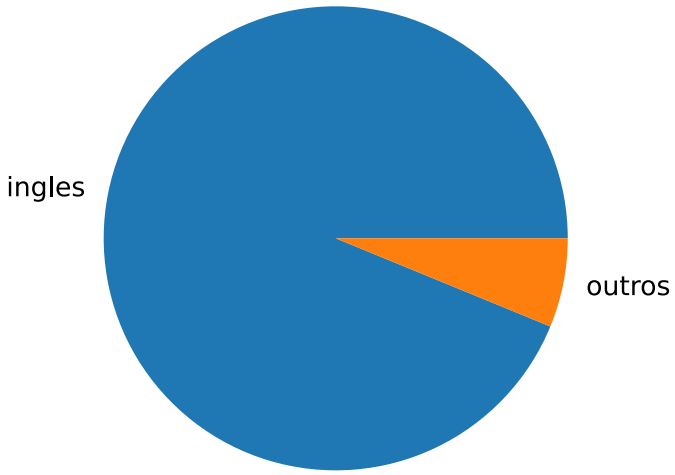
In [31]:

```
plt.pie(dados["total"], labels = dados["lingua"])
```

Out[31]:

```
([<matplotlib.patches.Wedge at 0xda755b0>,
```

```
 <matplotlib.patches.Wedge at 0xda75a90>],
[Text(-1.0791697536499925, 0.2130554923183512, 'ingles'),
 Text(1.0791697511565306, -0.2130555049482467, 'outros')])
```



In [32]:
```
total_por_lingua_de_outros_filmes = tmdb.query("original_language != 'en'").original_language.v
total_por_lingua_de_outros_filmes
```

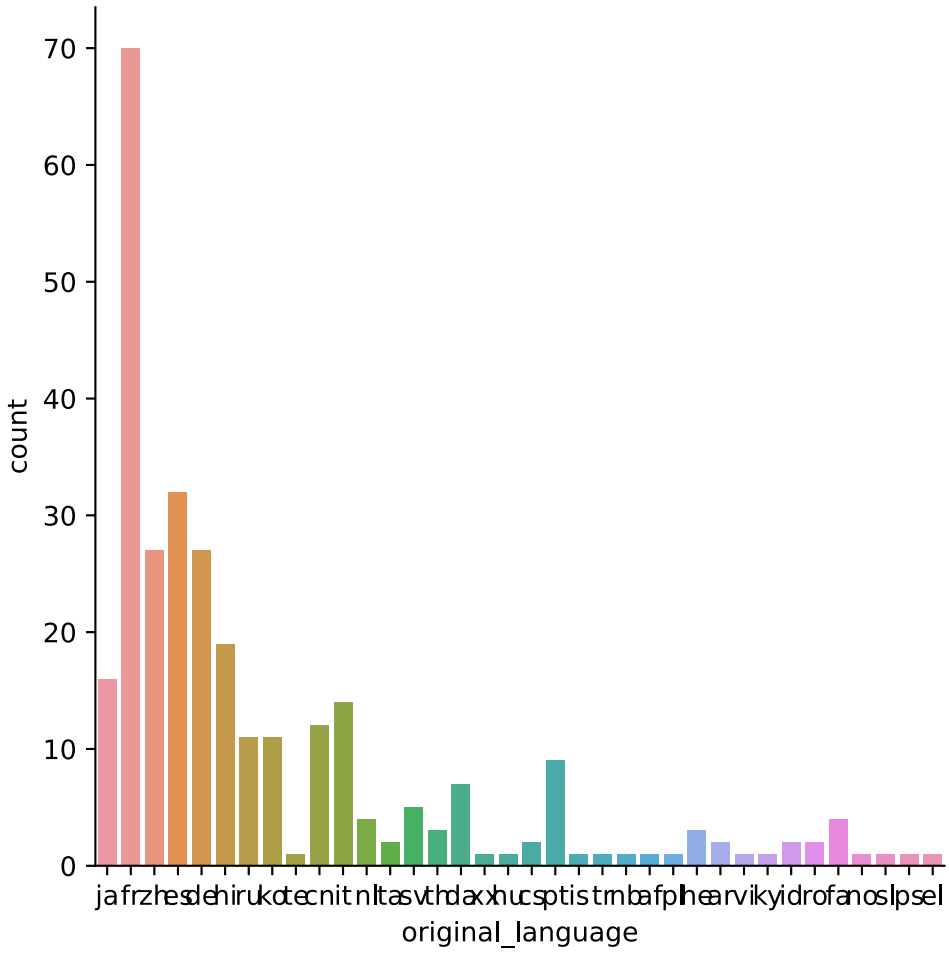
Out[32]:
```
fr    70
es    32
zh    27
de    27
hi    19
ja    16
it    14
cn    12
ko    11
ru    11
pt     9
da     7
sv     5
fa     4
nl     4
th     3
he     3
ta     2
cs     2
ar     2
id     2
ro     2
is     1
nb     1
hu     1
ps     1
tr     1
el     1
te     1
af     1
vi     1
pl     1
no     1
ky     1
xx     1
sl     1
Name: original_language, dtype: int64
```

In [33]:
```
filmes_sem_lingua_original_em_ingles = tmdb.query("original_language != 'en'")

sns.catplot(x = "original_language", kind="count",
            data = filmes_sem_lingua_original_em_ingles)
```

```
<seaborn.axisgrid.FacetGrid at 0xd7c5940>
```

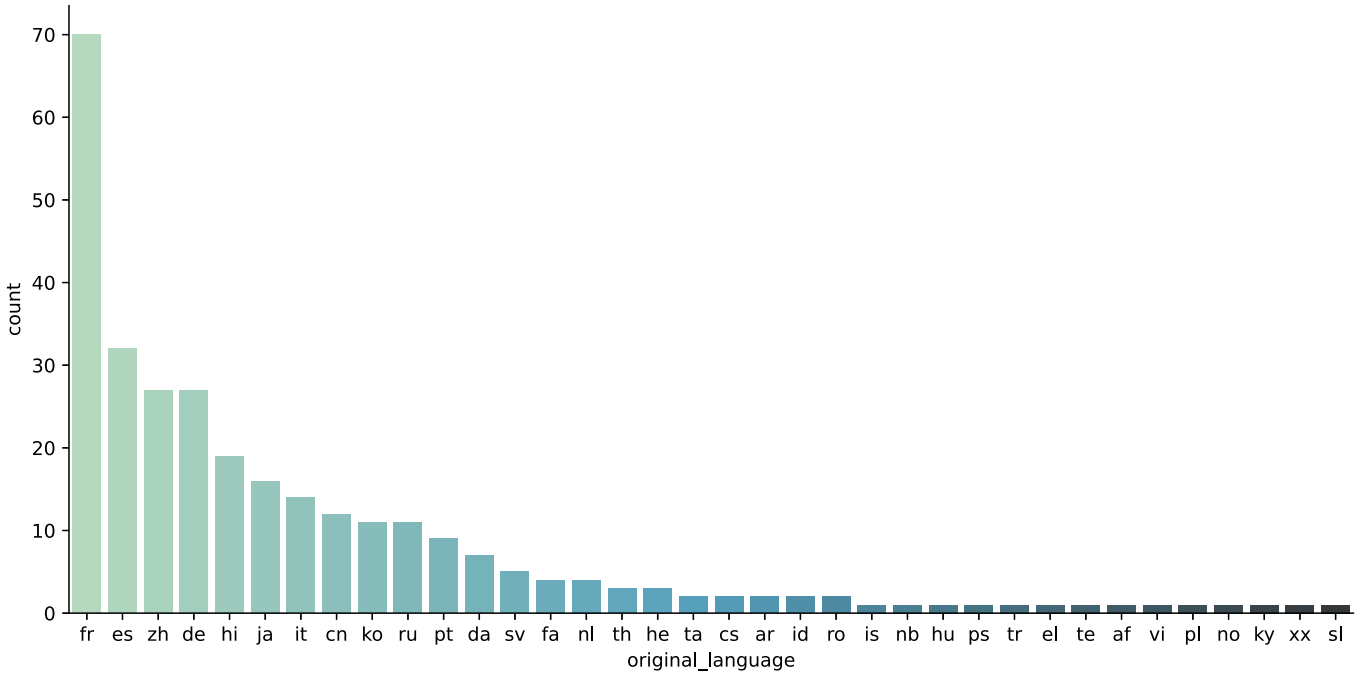
Out[33]:



In [34]:

```
sns.catplot(x = "original_language", kind="count",
            data = filmes_sem_lingua_original_em_ingles,
            aspect = 2,
            palette="GnBu_d",
            order = total_por_lingua_de_outros_filmes.index)
```

Out[34]: <seaborn.axisgrid.FacetGrid at 0xdb3b790>



Revisando o papel da média, mediana, medidas de tendência central, dispersão, desvio padrão, box

plot, histograma

In [35]:
```
filmes.head(2)
```

Out[35]:

	filmeId	titulo	generos
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy

In [36]:
```
notas_do_toy_story = notas.query("filmeId==1")
notas_do_jumanji = notas.query("filmeId==2")
print(len(notas_do_toy_story), len(notas_do_jumanji))
```

```
215 110
```

In [47]:
```
print(f"Nota média do Toy Story {notas_do_toy_story.nota.mean():.2f}")
print(f"Nota média do Jumanji {notas_do_jumanji.nota.mean():.2f}")
```

```
Nota média do Toy Story 3.92
Nota média do Jumanji 3.43
```

In [38]:
```
print(notas_do_toy_story.nota.std(), notas_do_jumanji.nota.std())
```

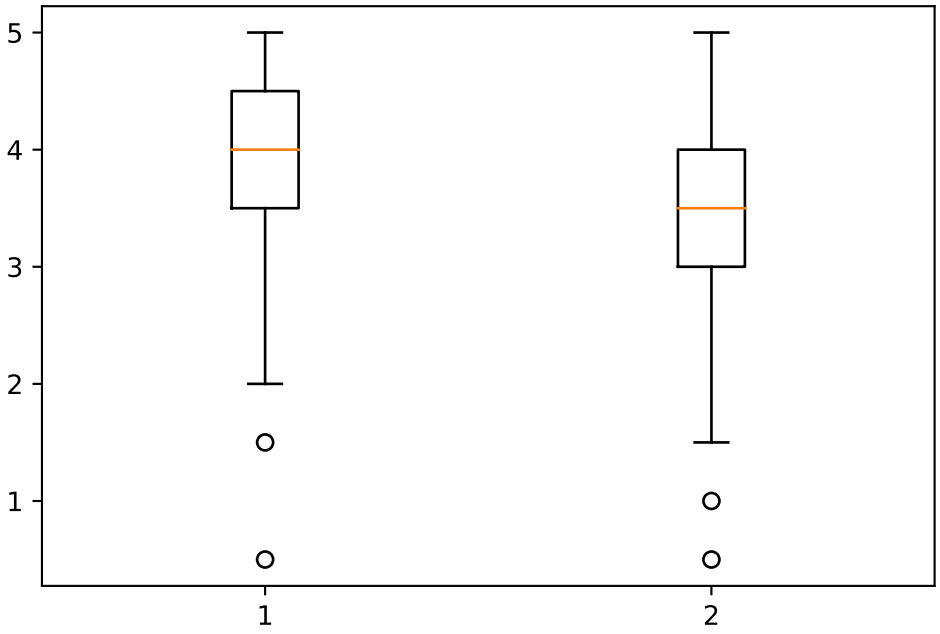
```
0.8348591407114047 0.8817134921476455
```

In [49]:
```
print(f"Nota mediana do Toy Story {notas_do_toy_story.nota.median():.2f}")
print(f"Nota mediana do Jumanji {notas_do_jumanji.nota.median():.2f}")
```

```
Nota mediana do Toy Story 4.00
Nota mediana do Jumanji 3.50
```

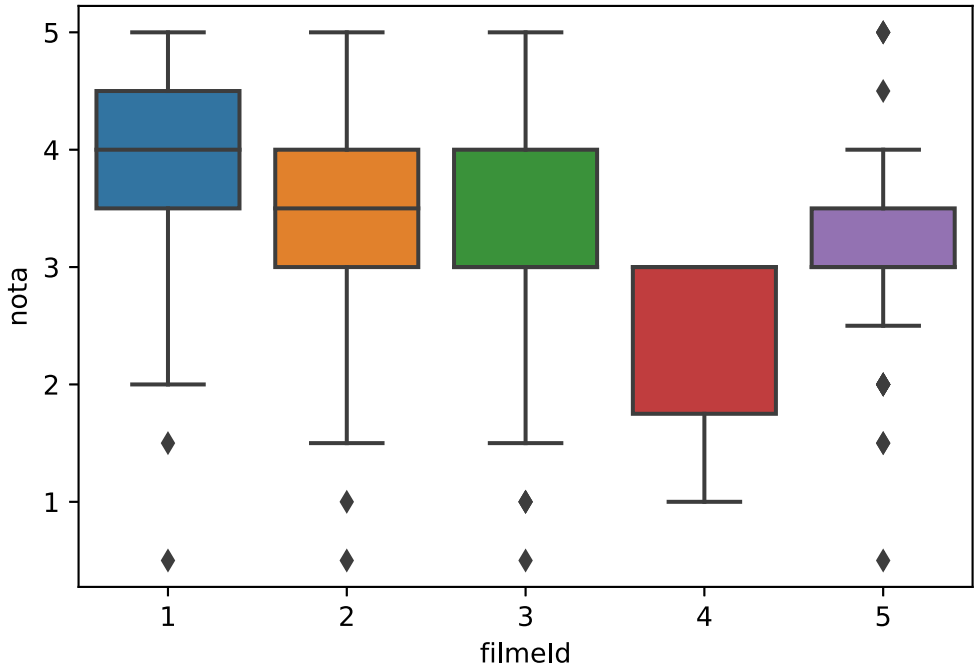
In [51]:
```
plt.boxplot([notas_do_toy_story.nota, notas_do_jumanji.nota])
```

Out[51]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0xdf2c8b0>,
  <matplotlib.lines.Line2D at 0xdf2c250>,
  <matplotlib.lines.Line2D at 0xdff12b0>,
  <matplotlib.lines.Line2D at 0xdff1610>],
 'caps': [<matplotlib.lines.Line2D at 0xdbeffa0>,
  <matplotlib.lines.Line2D at 0xdbefd00>,
  <matplotlib.lines.Line2D at 0xdff1970>,
  <matplotlib.lines.Line2D at 0xdff1cd0>],
 'boxes': [<matplotlib.lines.Line2D at 0xdcbab80>,
  <matplotlib.lines.Line2D at 0xdf3df10>],
 'medians': [<matplotlib.lines.Line2D at 0xdb6d6a0>,
  <matplotlib.lines.Line2D at 0xdffd070>],
 'fliers': [<matplotlib.lines.Line2D at 0xdf3dc10>,
  <matplotlib.lines.Line2D at 0xdffd3d0>],
 'means': []}
```



In [45]:
```
sns.boxplot(x = "filmeId", y = "nota", data = notas.query("filmeId in [1,2,3,4,5]"))
```

Out[45]: <AxesSubplot:xlabel='filmeId', ylabel='nota'>



In [0]:
```

```