In [51]:
```
#Resolvendo o problema de import do matplotlib intalando através do sys
#import sys
#!{sys.executable} -m pip install --user matplotlib
# Fonte: http://jakevdp.github.io/blog/2017/12/05/installing-python-packages-from-ju
```

Analisando as notas em geral

In [2]:
```
import pandas as pd
import numpy as np
import os
import matplotlib
import matplotlib.pyplot as plt
import random

# pip install seaborn
import seaborn as sns

#Lê o caminho atual: os.path.join(current_path,'ml-latest-small',"rating.csv" )
current_path = os.getcwd()
notas = pd.read_csv(os.path.join(current_path,'ml-latest-small','ratings.csv'), sep=

notas.head()
```

```
C:\Users\alexsandro.ignacio\AppData\Local\Programs\Python\Python37\lib\site-packages
\ipykernel_launcher.py:13: ParserWarning: Falling back to the 'python' engine becaus
e the 'c' engine does not support sep=None with delim_whitespace=False; you can avoi
d this warning by specifying engine='python'.
  del sys.path[0]
```

Out[2]:

	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 [3]:
```
notas.shape
```

Out[3]: (100836, 4)

In [4]:
```
notas.columns = ["usuarioiD", "filmeID", "nota", "momento"]
notas.head()
```

Out[4]:

	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 [5]:
```
notas['nota'].unique()
```

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

In [6]:
```
notas.head()
```

Out[6]:

	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 [7]:
```
notas['nota'].value_counts()
```

Out[7]:
```
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 [8]:
```
print("Média",notas['nota'].mean())
print("Mediana",notas['nota'].median())
```

```
Média 3.501556983616962
Mediana 3.5
```

In [9]:
```
notas.nota
```

Out[9]:
```
0         4.0
1         4.0
2         4.0
3         5.0
4         5.0
         ...
100831    4.0
100832    5.0
100833    5.0
100834    5.0
100835    3.0
Name: nota, Length: 100836, dtype: float64
```

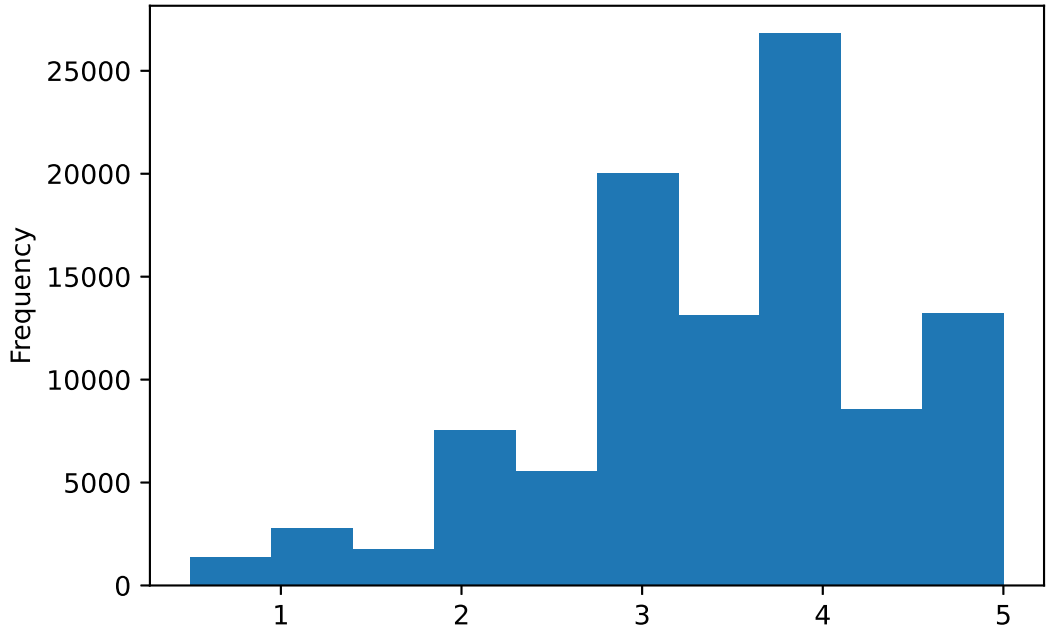
In [10]:
```
notas.nota.head()
```

Out[10]:
```
0    4.0
1    4.0
2    4.0
3    5.0
```

```
4    5.0
Name: nota, dtype: float64
```

In [11]:
```
notas.nota.plot(kind='hist')
```

Out[11]: <AxesSubplot:ylabel='Frequency'>



In [12]:
```
notas.nota.describe()
```

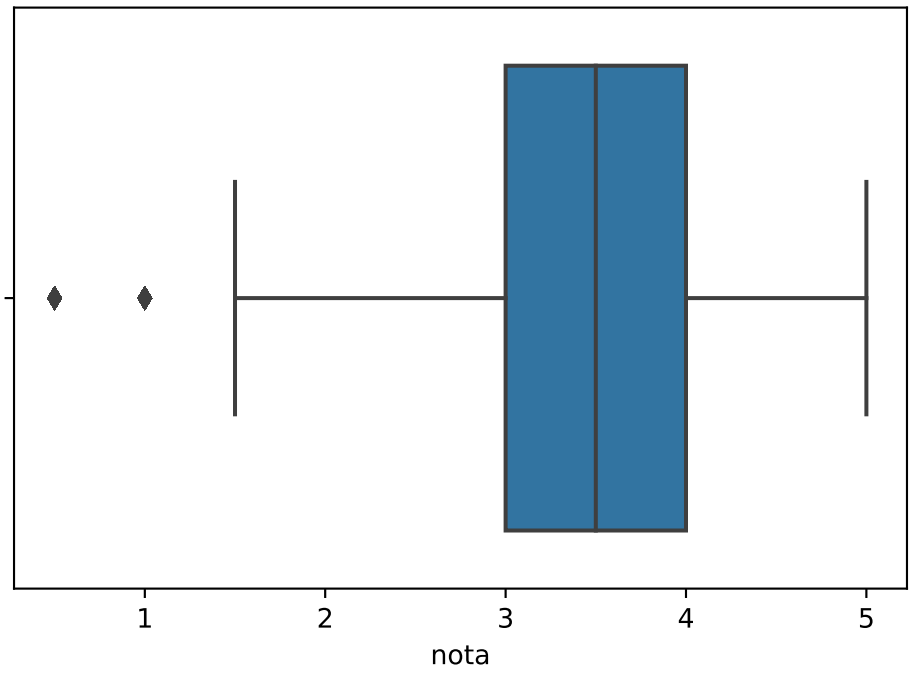
Out[12]:
```
count    100836.000000
mean          3.501557
std           1.042529
min           0.500000
25%           3.000000
50%           3.500000
75%           4.000000
max           5.000000
Name: nota, dtype: float64
```

In [13]:
```
#import sys
#!{sys.executable} -m pip install --user seaborn
```

In [14]:
```
sns.boxplot(notas.nota)
```

```
C:\Users\alexsandro.ignacio\AppData\Local\Programs\Python\Python37\lib\site-packages
\seaborn\_decorators.py:43: FutureWarning: Pass the following variable as a keyword
arg: x. From version 0.12, the only valid positional argument will be `data`, and pa
ssing other arguments without an explicit keyword will result in an error or misinte
rpretation.
  FutureWarning
```

Out[14]: <AxesSubplot:xlabel='nota'>



In [15]:
```
filmes = pd.read_csv(os.path.join(current_path, 'ml-latest-small','movies.csv'), sep
print(filmes)
```

```
      movieId                                      title  \
0           1                           Toy Story (1995)
1           2                             Jumanji (1995)
2           3                    Grumpier Old Men (1995)
3           4                   Waiting to Exhale (1995)
4           5         Father of the Bride Part II (1995)
...       ...                                        ...
9737   193581  Black Butler: Book of the Atlantic (2017)
9738   193583               No Game No Life: Zero (2017)
9739   193585                                 Flint (2017)
9740   193587        Bungo Stray Dogs: Dead Apple (2018)
9741   193609        Andrew Dice Clay: Dice Rules (1991)

                                       genres
0     Adventure|Animation|Children|Comedy|Fantasy
1                      Adventure|Children|Fantasy
2                                  Comedy|Romance
3                            Comedy|Drama|Romance
4                                          Comedy
...                                           ...
9737              Action|Animation|Comedy|Fantasy
9738                     Animation|Comedy|Fantasy
9739                                       Drama
9740                             Action|Animation
9741                                       Comedy

[9742 rows x 3 columns]
C:\Users\alexsandro.ignacio\AppData\Local\Programs\Python\Python37\lib\site-packages
\ipykernel_launcher.py:1: ParserWarning: Falling back to the 'python' engine because
the 'c' engine does not support sep=None with delim_whitespace=False; you can avoid
this warning by specifying engine='python'.
  """Entry point for launching an IPython kernel.
```

Olhando os Filmes

In [16]:
```
filmes.columns = ['filmeId ', 'titulo', 'generos']
filmes.head()
```

Out[16]:

	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 [17]:
```
notas.head()
```

Out[17]:

	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 [18]:
```
notas.query("filmeID==1")
```

Out[18]:

	usuarioiD	filmeID	nota	momento
0	1	1	4.0	964982703
516	5	1	4.0	847434962
874	7	1	4.5	1106635946
1434	15	1	2.5	1510577970
1667	17	1	4.5	1305696483
...	...	...	...	...
97364	606	1	2.5	1349082950
98479	607	1	4.0	964744033
98666	608	1	2.5	1117408267
99497	609	1	3.0	847221025
99534	610	1	5.0	1479542900

215 rows × 4 columns

In [19]:
```
notas.query("filmeID==1").nota
```

Out[19]:
```
0        4.0
516      4.0
874      4.5
1434     2.5
1667     4.5
        ...
97364    2.5
```

```
98479    4.0
98666    2.5
99497    3.0
99534    5.0
Name: nota, Length: 215, dtype: float64
```

Analisando algumas Notas Especificas por filme.

In [20]:
```
notas.query("filmeID==1").nota.mean()
```

Out[20]:
```
3.9209302325581397
```

In [21]:
```
notas.query("filmeID==2").nota.mean()
```

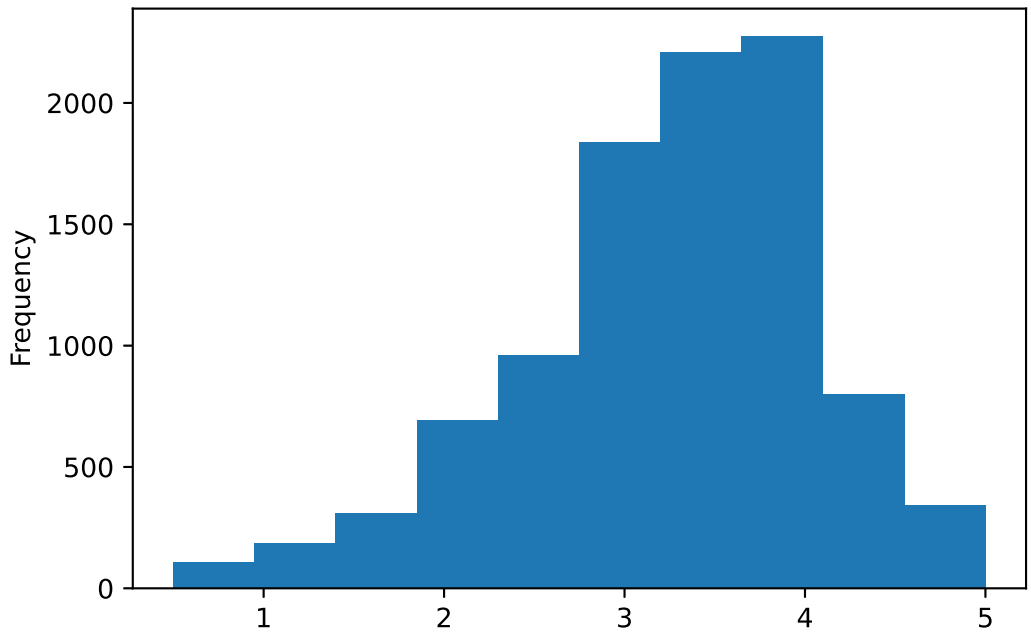
Out[21]:
```
3.4318181818181817
```

In [22]:
```
medias_por_filme = notas.groupby("filmeID").nota.mean() # ou mean()['nota']
medias_por_filme.head()
```

Out[22]:
```
filmeID
1    3.920930
2    3.431818
3    3.259615
4    2.357143
5    3.071429
Name: nota, dtype: float64
```

In [23]:
```
medias_por_filme.plot(kind='hist')
```

Out[23]:
```
<AxesSubplot:ylabel='Frequency'>
```

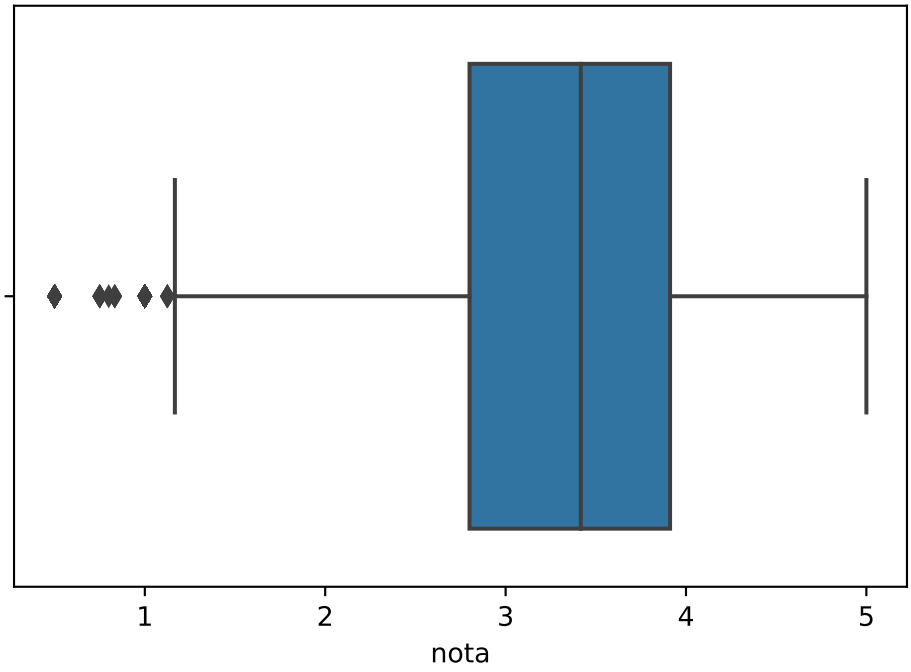


In [24]:
```
sns.boxplot(medias_por_filme)
```

```
C:\Users\alexsandro.ignacio\AppData\Local\Programs\Python\Python37\lib\site-packages
\seaborn\_decorators.py:43: FutureWarning: Pass the following variable as a keyword
```

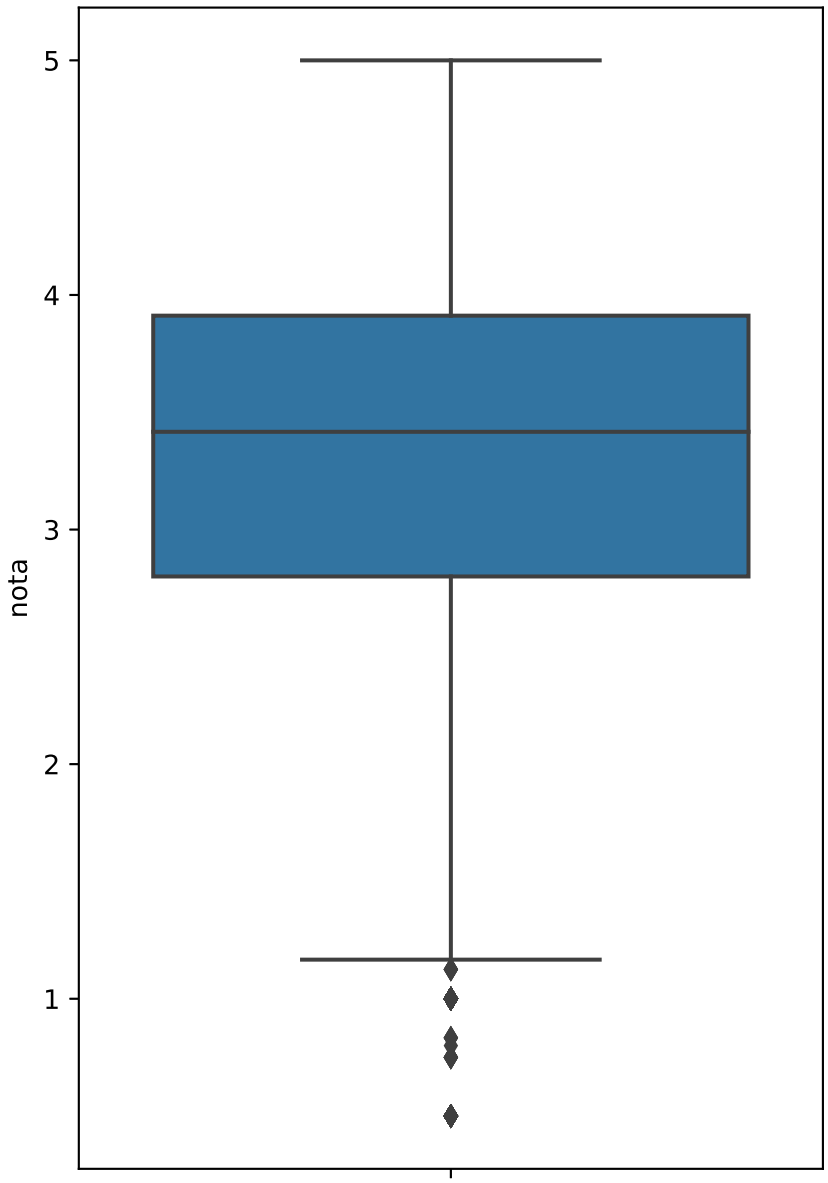
```
arg: x. From version 0.12, the only valid positional argument will be `data`, and pa
ssing other arguments without an explicit keyword will result in an error or misinte
rpretation.
  FutureWarning
```

Out[24]: <AxesSubplot:xlabel='nota'>



In [25]:
```
plt.figure(figsize=(5,8))
sns.boxplot(y=medias_por_filme)
```

Out[25]: <AxesSubplot:ylabel='nota'>



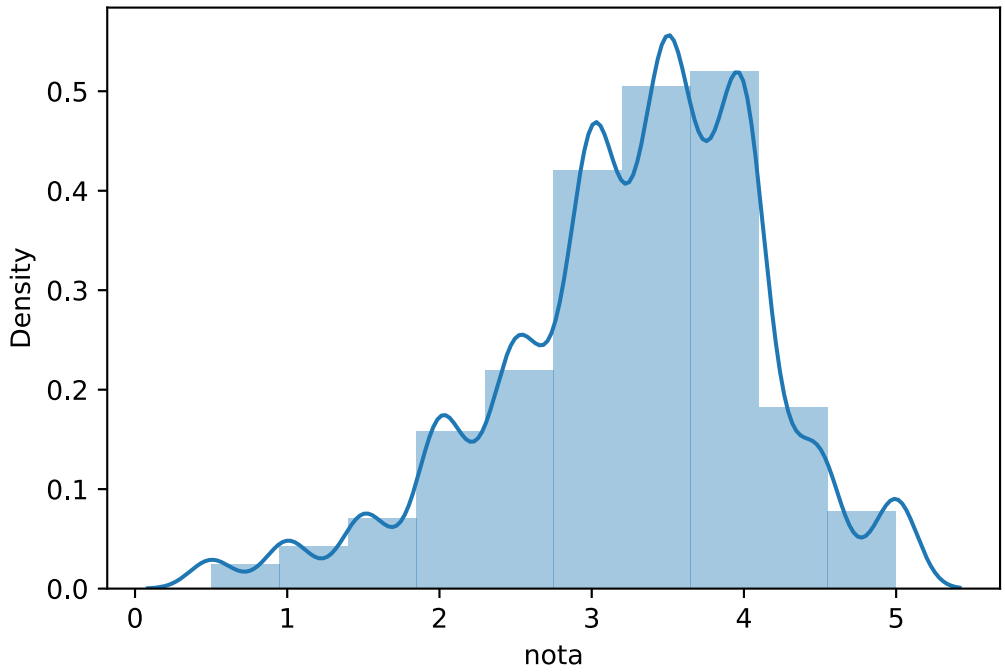
In [26]:
```
medias_por_filme.describe()
```

Out[26]:
```
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 [27]:
```
sns.distplot(medias_por_filme, bins=10)
```

```
C:\Users\alexsandro.ignacio\AppData\Local\Programs\Python\Python37\lib\site-packages
\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function a
nd will be removed in a future version. Please adapt your code to use either `displo
t` (a figure-level function with similar flexibility) or `histplot` (an axes-level f
unction for histograms).
  warnings.warn(msg, FutureWarning)
```

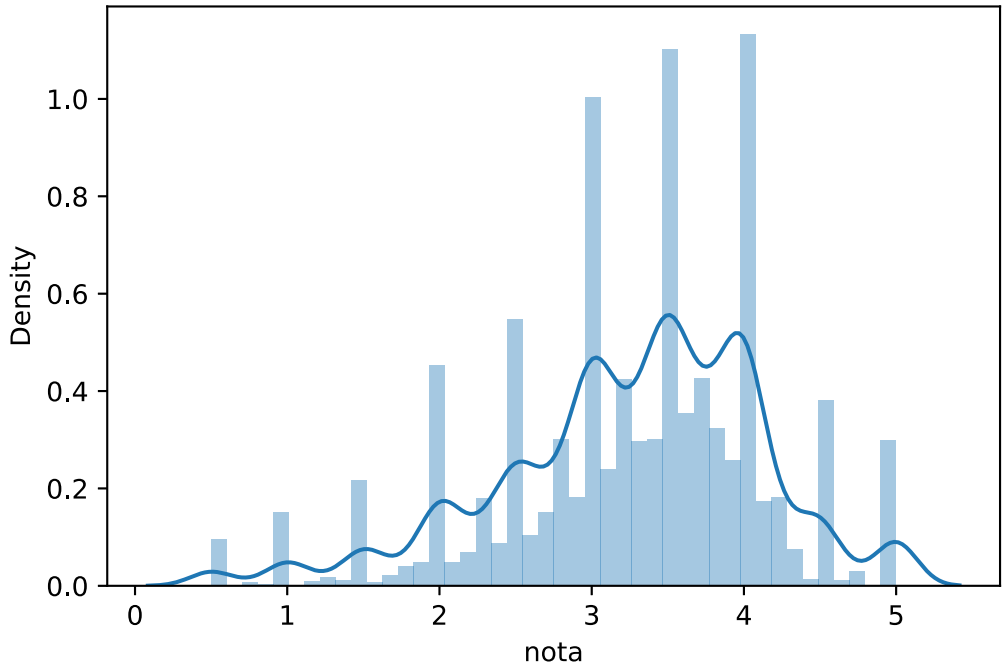
Out[27]:
```
<AxesSubplot:xlabel='nota', ylabel='Density'>
```



In [28]:
```
sns.distplot(medias_por_filme)
```

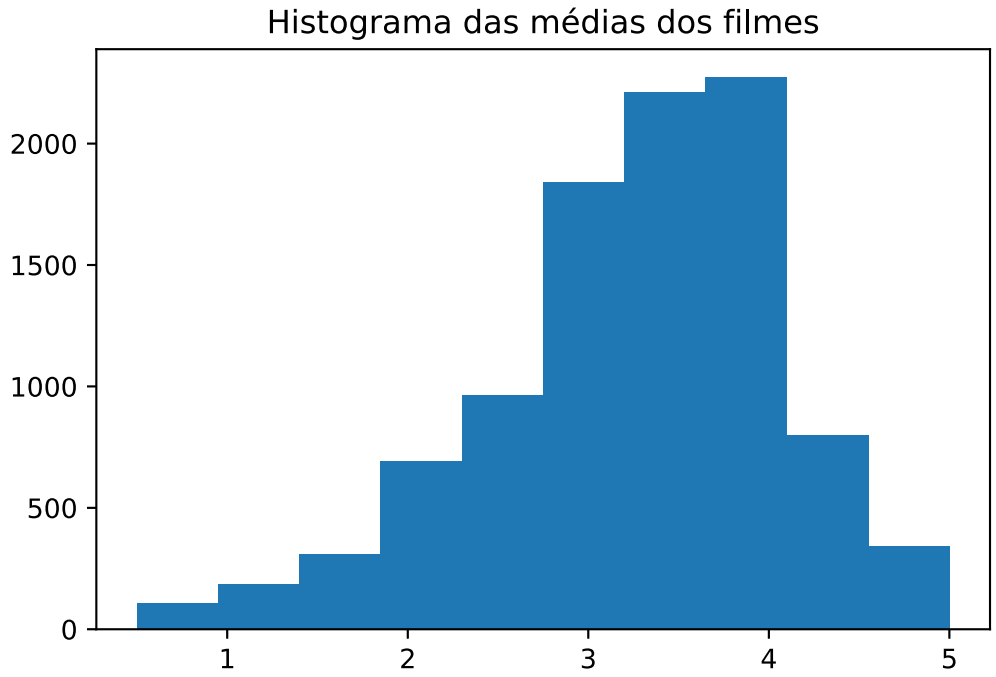
```
C:\Users\alexsandro.ignacio\AppData\Local\Programs\Python\Python37\lib\site-packages
\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function a
nd will be removed in a future version. Please adapt your code to use either `displo
t` (a figure-level function with similar flexibility) or `histplot` (an axes-level f
unction for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[28]: <AxesSubplot:xlabel='nota', ylabel='Density'>



In [29]:
```
mat = plt.hist(medias_por_filme)
plt.title("Histograma das médias dos filmes")
```

Out[29]: Text(0.5, 1.0, 'Histograma das médias dos filmes')



Começando nova Análise exploratória de dados

In [30]:
```
filmes.head(2)
```

Out[30]:

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

In [31]:
```
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 [32]:
```
print("             Table Toy Story\n")
print(notas_do_toy_story.head())
print("\n              Table Jumanji\n")
print(notas_do_Jumanji.head())
```

```
             Table Toy Story

      usuarioiD  filmeID  nota     momento
0             1        1   4.0   964982703
516           5        1   4.0   847434962
874           7        1   4.5  1106635946
1434         15        1   2.5  1510577970
1667         17        1   4.5  1305696483

              Table Jumanji

      usuarioiD  filmeID  nota     momento
560           6        2   4.0   845553522
1026          8        2   4.0   839463806
1773         18        2   3.0  1455617462
```

```
2275         19        2   3.0   965704331
2977         20        2   3.0  1054038313
```

In [33]:
```
print("Nota média do Toy Story %.2f" % notas_do_toy_story.nota.mean())
print("Nota média do Jumanji %.2f" % notas_do_Jumanji.nota.mean())
print("Nota mediana do Toy Story %.2f" % notas_do_toy_story.nota.median())
print("Nota mediana do Jumanji %.2f" % notas_do_Jumanji.nota.median())
print(f"Desvio Padrão Toy Story (Standard Deviation):{notas_do_toy_story.nota.std()}
```

```
Nota média do Toy Story 3.92
Nota média do Jumanji 3.43
Nota mediana do Toy Story 4.00
Nota mediana do Jumanji 3.50
Desvio Padrão Toy Story (Standard Deviation):0.8348591407114047
Desvio Padrão Jumanji (Standard Deviation):0.8817134921476455
```

In [34]:
```
np.array([2.5] * 10)
```

Out[34]:
```
array([2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5])
```

In [35]:
```
np.array([2.5] * 10).mean()
```

Out[35]:
```
2.5
```

In [36]:
```
np.array([3.5] * 10)
```

Out[36]:
```
array([3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5])
```

In [37]:
```
round(random.random()*100,2)
```

Out[37]:
```
68.64
```

In [38]:
```
valor_random = []
for i in range(0,10):
    valor_random.append(round(random.random()*100,2))
    #print(valor)
print()
valor_random
```

Out[38]:
```
[25.57, 33.86, 49.22, 99.25, 94.15, 92.11, 47.14, 99.07, 44.35, 47.8]
```

In [39]:
```
filme1 = np.append(np.array([2.5] * 10), np.array([3.5] * 10))
filme2 = np.append(np.array([5] * 10), np.array([1] * 10))
filme3 = valor_random
print(f"filme1: {filme1}\nfilme2: {filme2}\nfilme3: {filme3}")
```

```
filme1: [2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 3.5 3.5 3.5 3.5 3.5 3.5 3.5 3.5
 3.5 3.5]
filme2: [5 5 5 5 5 5 5 5 5 5 1 1 1 1 1 1 1 1 1 1]
filme3: [25.57, 33.86, 49.22, 99.25, 94.15, 92.11, 47.14, 99.07, 44.35, 47.8]
```

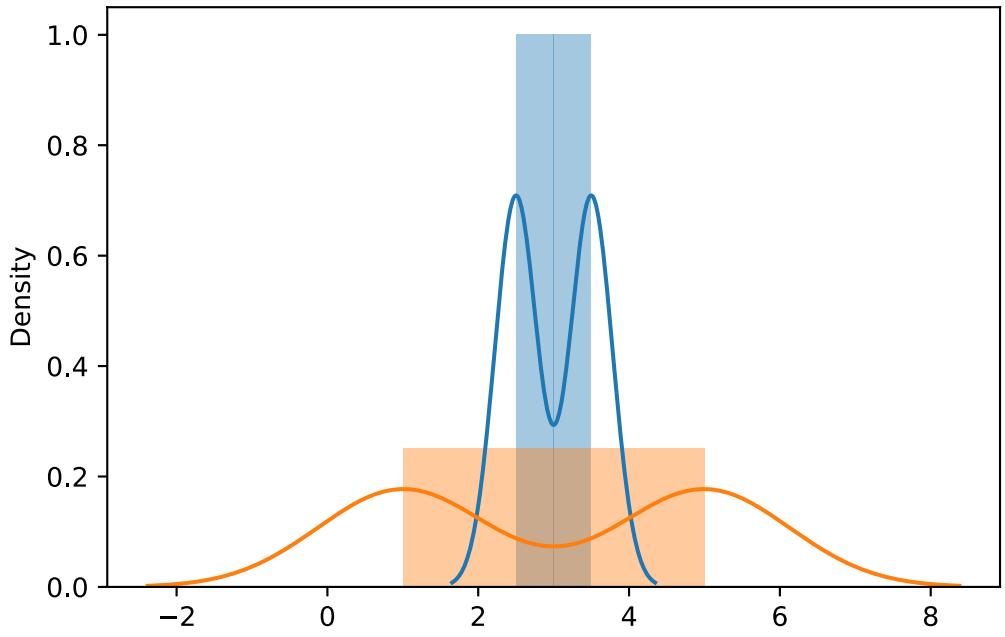
In [40]:
```
print(f"filme1 média: {filme1.mean()}\nfilme2 média: {filme2.mean()}")
print(f"filme1 mediana: {np.median(filme1)}\nfilme2 mediana: {np.median(filme2)}")
print(f"filme1 Desvio Padrão (Standard Deviation): {np.std(filme1)}\nfilme2 Desvio P
```

```
filme1 média: 3.0
filme2 média: 3.0
filme1 mediana: 3.0
filme2 mediana: 3.0
filme1 Desvio Padrão (Standard Deviation): 0.5
filme2 Desvio Padrão (Standard Deviation): 2.0
```

In [41]:
```
sns.distplot(filme1)
sns.distplot(filme2)
```

```
C:\Users\alexsandro.ignacio\AppData\Local\Programs\Python\Python37\lib\site-packages
\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function a
nd will be removed in a future version. Please adapt your code to use either `displo
t` (a figure-level function with similar flexibility) or `histplot` (an axes-level f
unction for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\alexsandro.ignacio\AppData\Local\Programs\Python\Python37\lib\site-packages
\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function a
nd will be removed in a future version. Please adapt your code to use either `displo
t` (a figure-level function with similar flexibility) or `histplot` (an axes-level f
unction for histograms).
  warnings.warn(msg, FutureWarning)
```

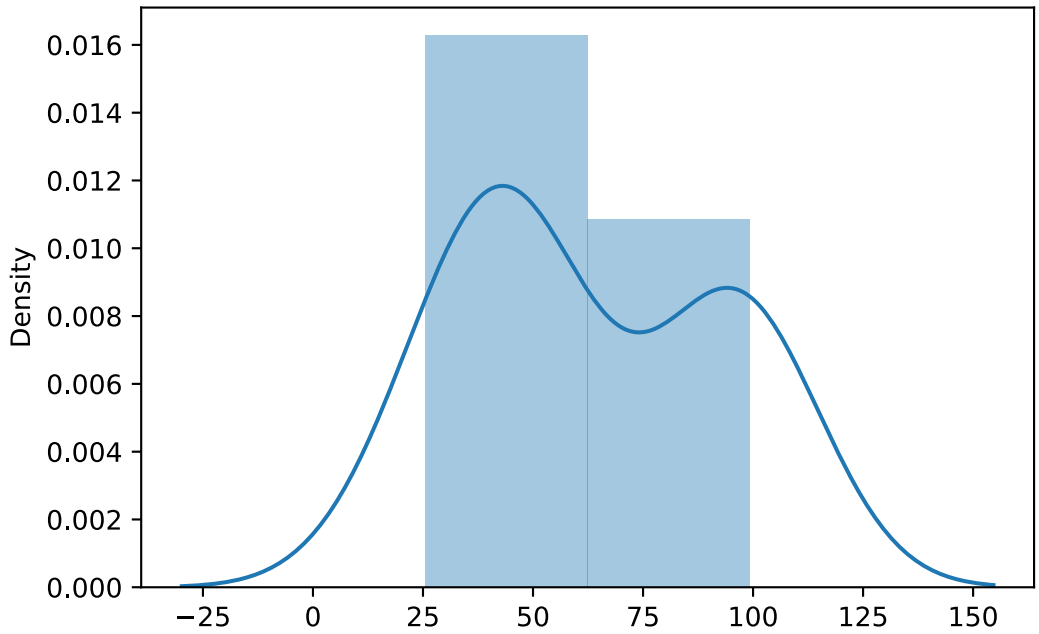
Out[41]: <AxesSubplot:ylabel='Density'>



In [42]:
```
sns.distplot(filme3)
```

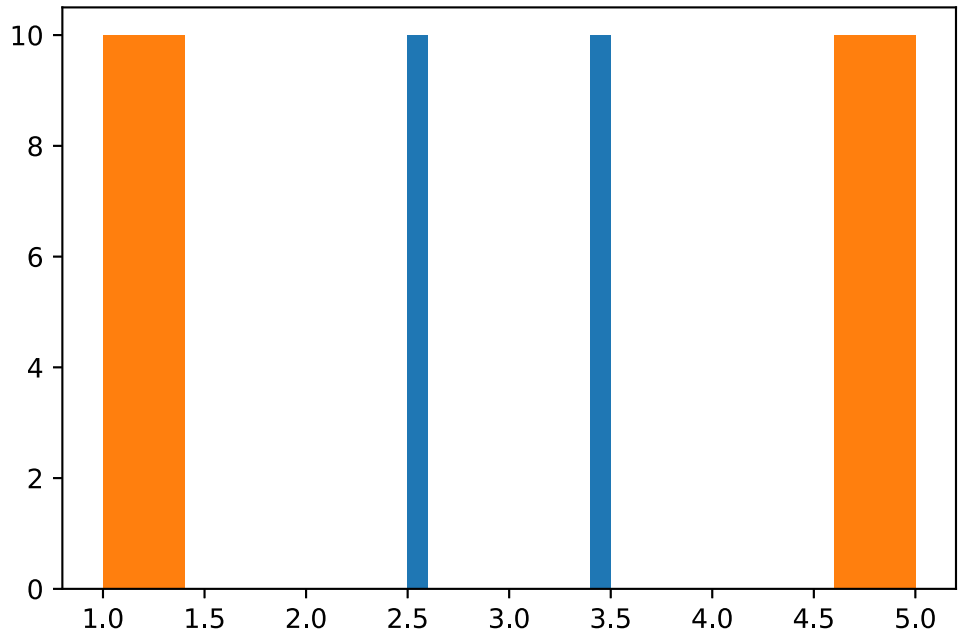
```
C:\Users\alexsandro.ignacio\AppData\Local\Programs\Python\Python37\lib\site-packages
\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function a
nd will be removed in a future version. Please adapt your code to use either `displo
t` (a figure-level function with similar flexibility) or `histplot` (an axes-level f
unction for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[42]: <AxesSubplot:ylabel='Density'>



In [43]:
```
plt.hist(filme1)
plt.hist(filme2)
```

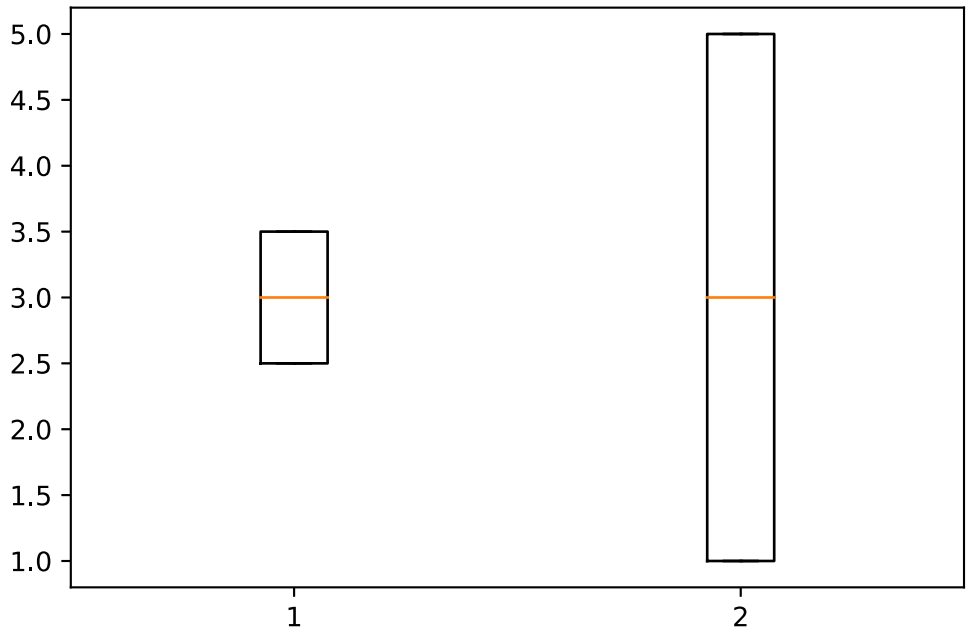
Out[43]:
```
(array([10.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0., 10.]),
 array([1. , 1.4, 1.8, 2.2, 2.6, 3. , 3.4, 3.8, 4.2, 4.6, 5. ]),
 <BarContainer object of 10 artists>)
```



In [44]:
```
plt.boxplot([filme1, filme2])
```

Out[44]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x16e3f06d0c8>,
  <matplotlib.lines.Line2D at 0x16e3f076e48>,
  <matplotlib.lines.Line2D at 0x16e3f07ae88>,
  <matplotlib.lines.Line2D at 0x16e3f07a548>],
 'caps': [<matplotlib.lines.Line2D at 0x16e3f076c88>,
  <matplotlib.lines.Line2D at 0x16e3f078c88>,
  <matplotlib.lines.Line2D at 0x16e3f07ba48>,
  <matplotlib.lines.Line2D at 0x16e3f07df48>],
 'boxes': [<matplotlib.lines.Line2D at 0x16e3f076d88>,
  <matplotlib.lines.Line2D at 0x16e3f07aa48>],
 'medians': [<matplotlib.lines.Line2D at 0x16e3f078b08>,
  <matplotlib.lines.Line2D at 0x16e3f07d788>],
```

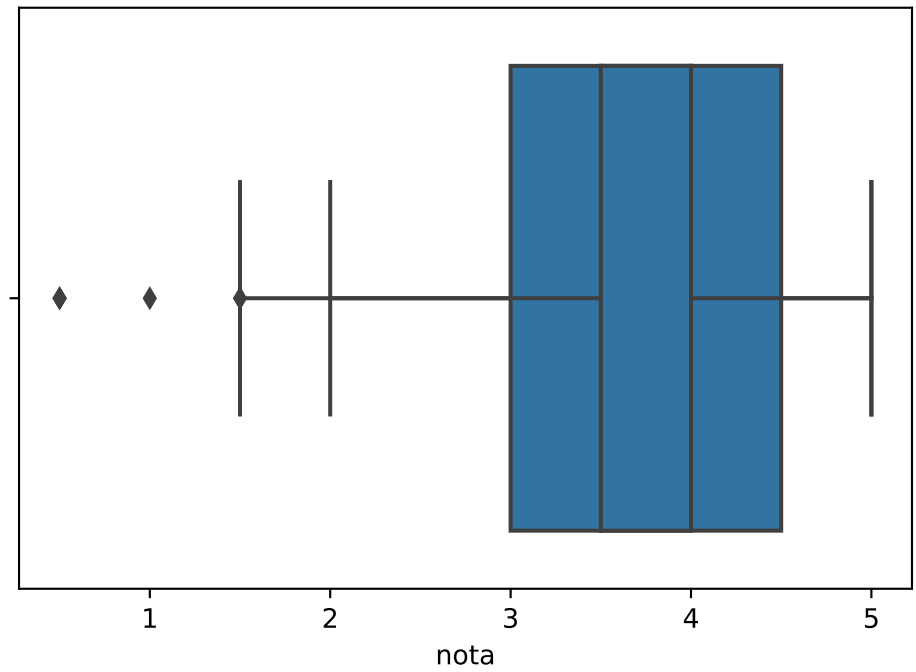
```
'fliers': [<matplotlib.lines.Line2D at 0x16e3f07abc8>,
 <matplotlib.lines.Line2D at 0x16e3f07fe48>],
'means': []}
```



In [45]:
```
sns.boxplot(notas_do_toy_story.nota)
sns.boxplot(notas_do_Jumanji.nota)
```

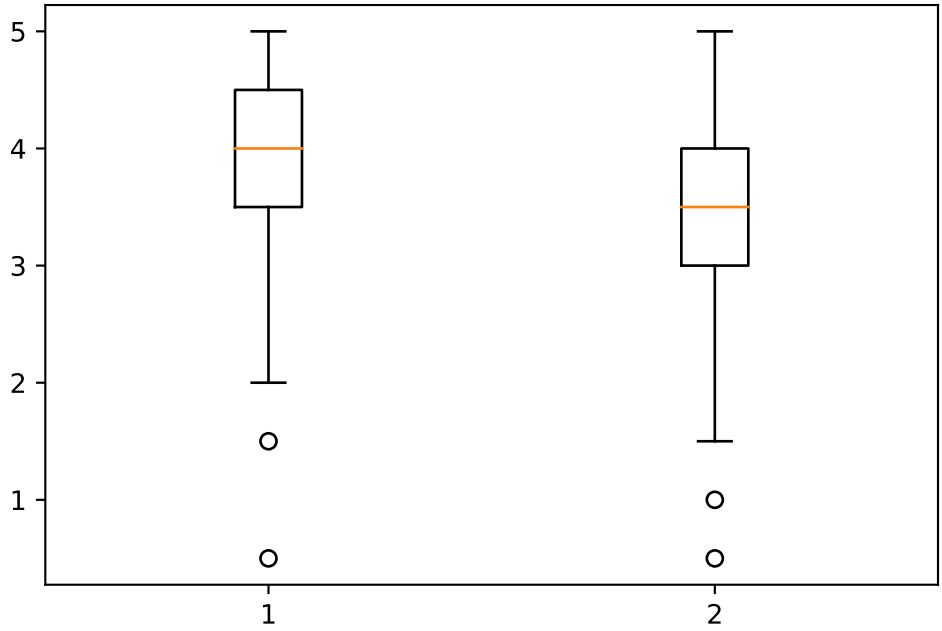
```
C:\Users\alexsandro.ignacio\AppData\Local\Programs\Python\Python37\lib\site-packages
\seaborn\_decorators.py:43: FutureWarning: Pass the following variable as a keyword
arg: x. From version 0.12, the only valid positional argument will be `data`, and pa
ssing other arguments without an explicit keyword will result in an error or misinte
rpretation.
  FutureWarning
C:\Users\alexsandro.ignacio\AppData\Local\Programs\Python\Python37\lib\site-packages
\seaborn\_decorators.py:43: FutureWarning: Pass the following variable as a keyword
arg: x. From version 0.12, the only valid positional argument will be `data`, and pa
ssing other arguments without an explicit keyword will result in an error or misinte
rpretation.
  FutureWarning
```

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



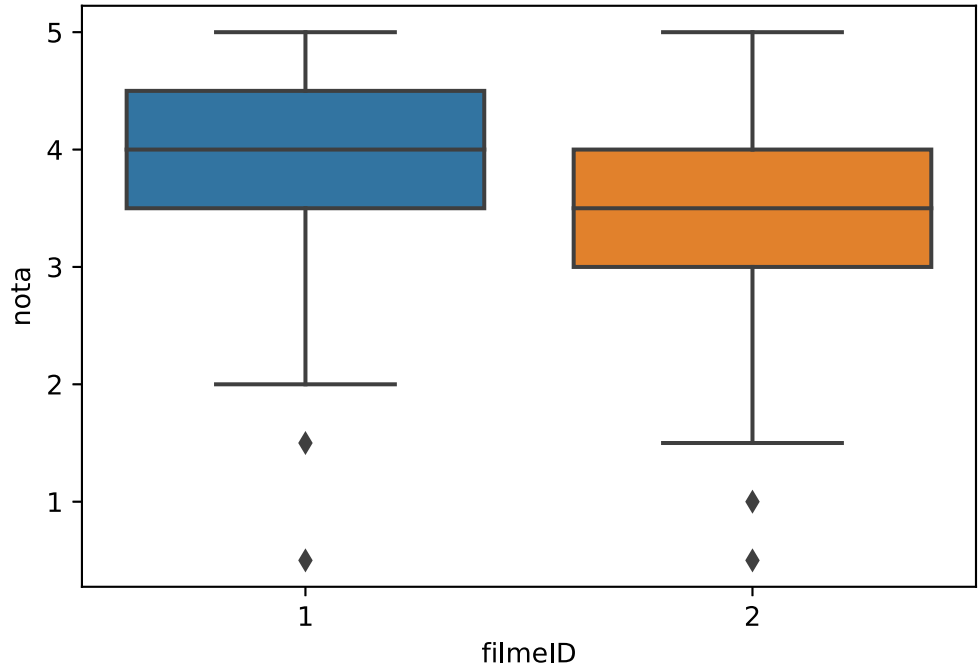
In [46]:
```
plt.boxplot([notas_do_toy_story.nota, notas_do_Jumanji.nota])
```

Out[46]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x16e3f0ca1c8>,
  <matplotlib.lines.Line2D at 0x16e3f0d4e48>,
  <matplotlib.lines.Line2D at 0x16e3f0d7e08>,
  <matplotlib.lines.Line2D at 0x16e3f0d76c8>],
 'caps': [<matplotlib.lines.Line2D at 0x16e3f0d4d08>,
  <matplotlib.lines.Line2D at 0x16e3f0d5c88>,
  <matplotlib.lines.Line2D at 0x16e3f0d8988>,
  <matplotlib.lines.Line2D at 0x16e3f0dbe48>],
 'boxes': [<matplotlib.lines.Line2D at 0x16e3f0d4dc8>,
  <matplotlib.lines.Line2D at 0x16e3f0d7888>],
 'medians': [<matplotlib.lines.Line2D at 0x16e3f0d5b48>,
  <matplotlib.lines.Line2D at 0x16e3f0dbd08>],
 'fliers': [<matplotlib.lines.Line2D at 0x16e3f0d7a48>,
  <matplotlib.lines.Line2D at 0x16e3f0ddb48>],
 'means': []}
```



In [47]:
```
sns.boxplot(x= "filmeID", y= "nota", data= notas.query("filmeID in [1,2]"))
```

Out[47]:
```
<AxesSubplot:xlabel='filmeID', ylabel='nota'>
```



In [48]:
```
sns.boxplot(x= "filmeID", y= "nota", data= notas.query("filmeID in [1,2,3,4,5]"))
```

Out[48]: <AxesSubplot:xlabel='filmeID', ylabel='nota'>