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-jupyter/
```

Analisando as notas em geral

In [1]:

```
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=None)

notas.head()
```

```
c:\users\alexsandro.ignacio\appdata\local\programs\python\python37\lib\site-
packages\ipykernel_launcher.py:13: ParserWarning: Falling back to the 'pytho
n' engine because the 'c' engine does not support sep=None with delim_whites
pace=False; you can avoid this warning by specifying engine='python'.
  del sys.path[0]
```

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.head()
```

Out[5]:

	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 [6]:

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

Out[6]:

```
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 [7]:

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

```
Média 3.501556983616962
Mediana 3.5
```

In [8]:

```
notas.nota
```

Out[8]:

```
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 [9]:

```
notas.nota.head()
```

Out[9]:

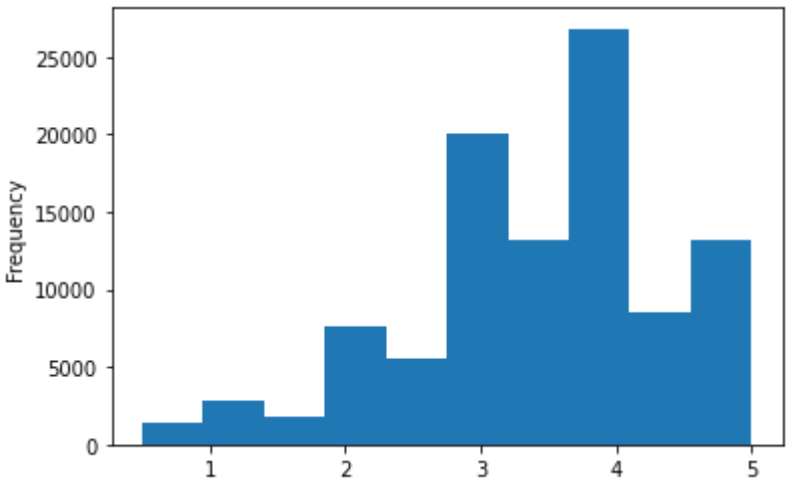
```
0    4.0
1    4.0
2    4.0
3    5.0
4    5.0
Name: nota, dtype: float64
```

In [10]:

```
notas.nota.plot(kind='hist')
```

Out[10]:

```
<AxesSubplot:ylabel='Frequency'>
```



In [11]:

```
notas.nota.describe()
```

Out[11]:

```
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 [12]:

```
#import sys
#!{sys.executable} -m pip install --user seaborn
```

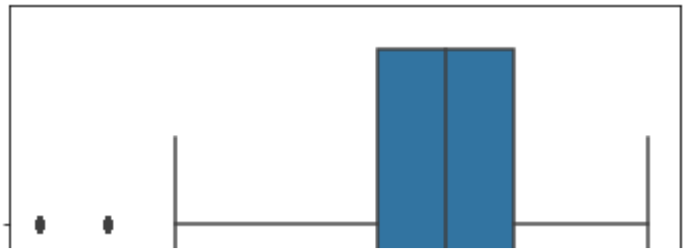
In [13]:

```
sns.boxplot(notas.nota)
```

```
c:\users\alexsandro.ignacio\appdata\local\programs\python\python37\lib\sit
e-packages\seaborn\_decorators.py:43: FutureWarning: Pass the following va
riable as a keyword arg: x. From version 0.12, the only valid positional a
rgument will be `data`, and passing other arguments without an explicit ke
yword will result in an error or misinterpretation.
  FutureWarning
```

Out[13]:

```
<AxesSubplot:xlabel='nota'>
```



In [14]:

```
filmes = pd.read_csv(os.path.join(current_path, 'ml-latest-small','movies.csv'), sep=None)
print(filmes)
```

```
c:\users\alexsandro.ignacio\appdata\local\programs\python\python37\lib\site-
packages\ipykernel_launcher.py:1: ParserWarning: Falling back to the 'pytho
n' engine because the 'c' engine does not support sep=None with delim_whites
pace=False; you can avoid this warning by specifying engine='python'.
  """Entry point for launching an IPython kernel.

      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]
```

Olhando os Filmes

In [15]:

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

Out[15]:

	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 [16]:

```
notas.head()
```

Out[16]:

	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 [17]:

```
notas.query("filmeID==1")
```

Out[17]:

	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 [18]:

```
notas.query("filmeID==1").nota
```

Out[18]:

```
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 [19]:

```
notas.query("filmeID==1").nota.mean()
```

Out[19]:

```
3.9209302325581397
```

In [20]:

```
notas.query("filmeID==2").nota.mean()
```

Out[20]:

```
3.4318181818181817
```

In [21]:

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

Out[21]:

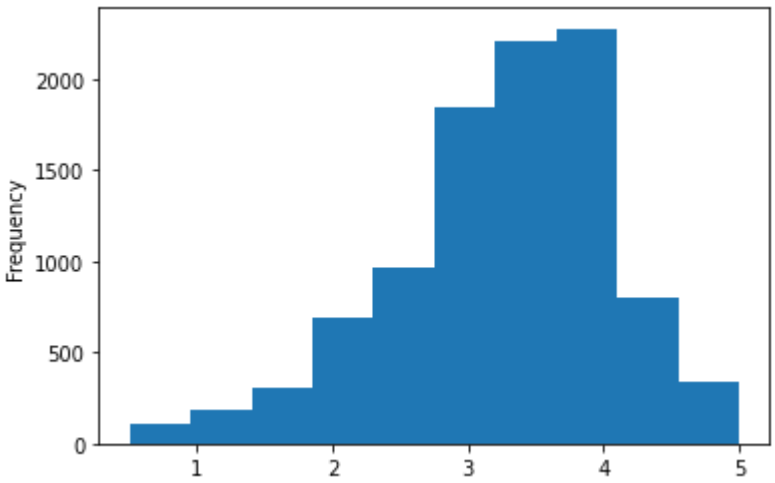
```
filmeID
1    3.920930
2    3.431818
3    3.259615
4    2.357143
5    3.071429
Name: nota, dtype: float64
```

In [22]:

```
medias_por_filme.plot(kind='hist')
```

Out[22]:

```
<AxesSubplot:ylabel='Frequency'>
```



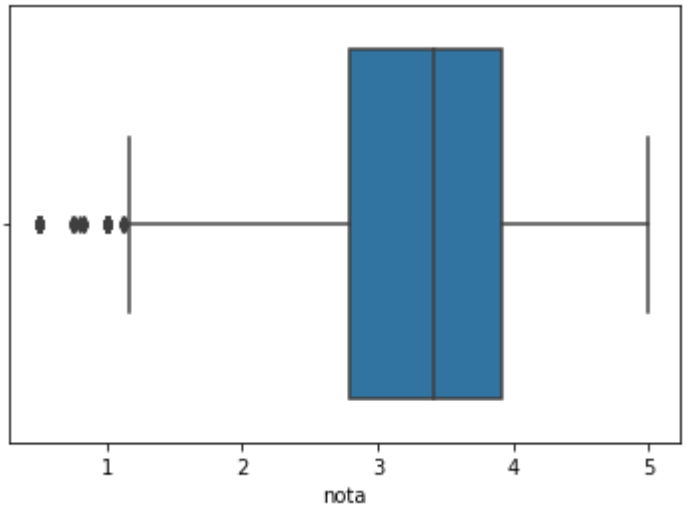
In [23]:

```
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 variab
le as a keyword arg: x. From version 0.12, the only valid positional argumen
t will be `data`, and passing other arguments without an explicit keyword wi
ll result in an error or misinterpretation.
  FutureWarning
```

Out[23]:

```
<AxesSubplot:xlabel='nota'>
```

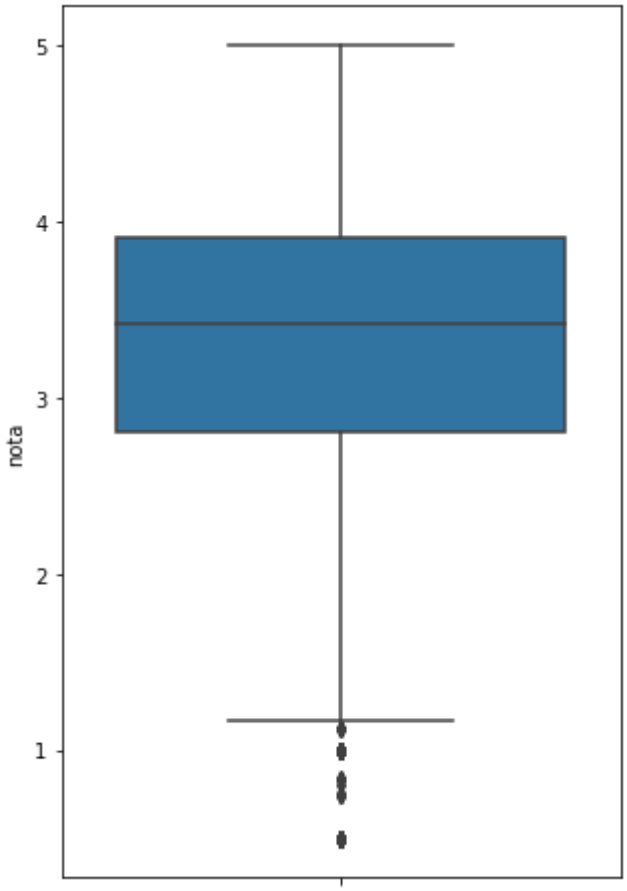


In [24]:

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

Out[24]:

```
<AxesSubplot:ylabel='nota'>
```



In [25]:

```
medias_por_filme.describe()
```

Out[25]:

```
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
```

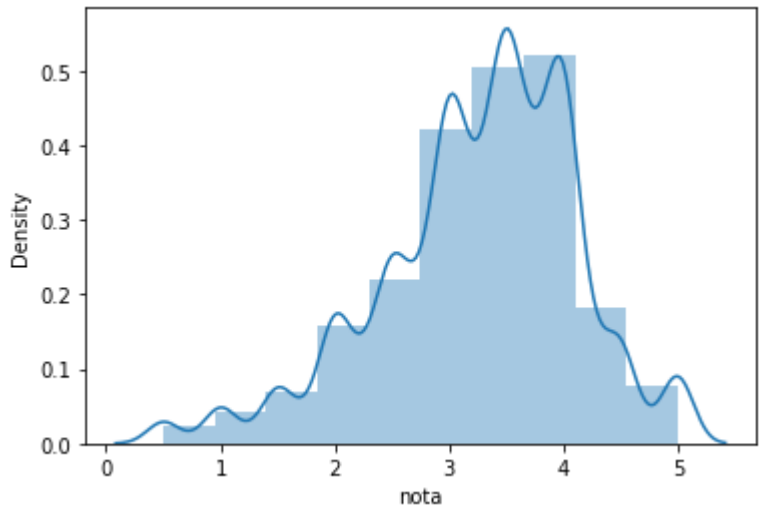
In [26]:

```
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 depre
cated function and will be removed in a future version. Please adapt your co
de to use either `displot` (a figure-level function with similar flexibilit
y) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[26]:

```
<AxesSubplot:xlabel='nota', ylabel='Density'>
```



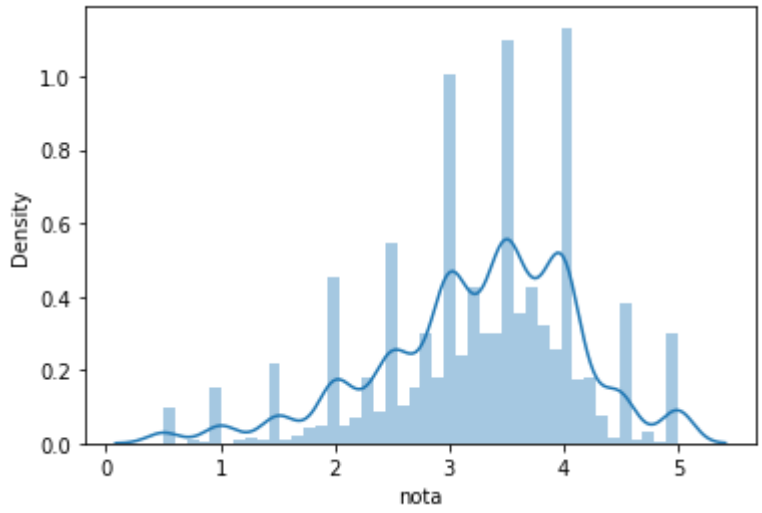
In [27]:

```
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 depre
cated function and will be removed in a future version. Please adapt your co
de to use either `displot` (a figure-level function with similar flexibilit
y) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[27]:

```
<AxesSubplot:xlabel='nota', ylabel='Density'>
```

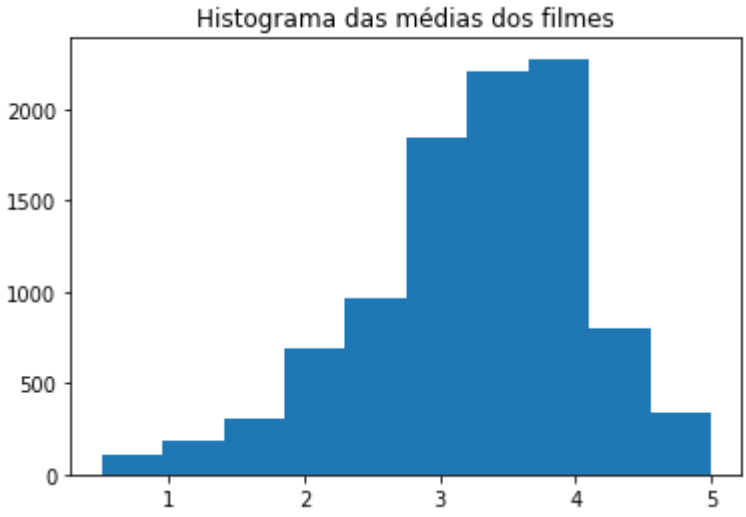


In [28]:

```
mat = plt.hist(medias_por_filme)
plt.title("Histograma das médias dos filmes")
```

Out[28]:

```
Text(0.5, 1.0, 'Histograma das médias dos filmes')
```



Começando nova Análise exploratória de dados

In [29]:

```
filmes.head(2)
```

Out[29]:

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

In [30]:

```
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 [31]:

```
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 [32]:

```
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()}\nDesvi
```

```
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 [33]:

```
np.array([2.5] * 10)
```

Out[33]:

```
array([2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5])
```

In [34]:

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

Out[34]:

```
2.5
```

In [35]:

```
np.array([3.5] * 10)
```

Out[35]:

```
array([3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5])
```

In [36]:

```
round(random.random()*100,2)
```

Out[36]:

```
11.37
```

In [37]:

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

Out[37]:

```
[17.4, 3.96, 0.85, 43.71, 5.3, 39.86, 37.93, 28.85, 60.77, 93.22]
```

In [38]:

```
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: [17.4, 3.96, 0.85, 43.71, 5.3, 39.86, 37.93, 28.85, 60.77, 93.22]
```

In [39]:

```
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 Padrão (
```

```
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
```

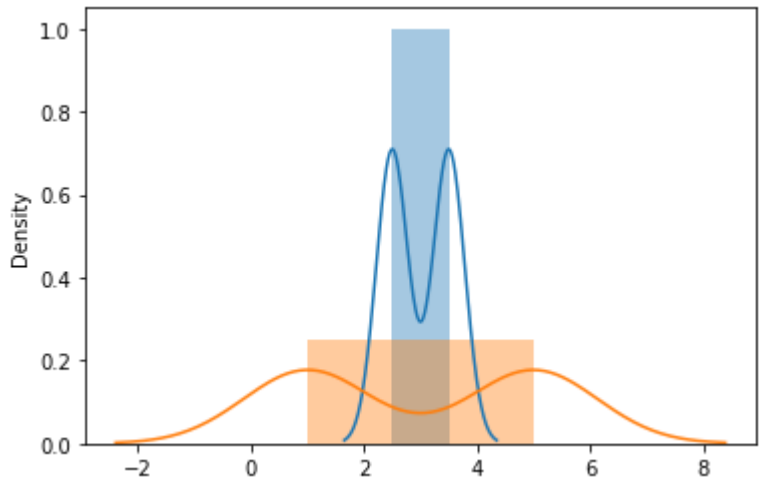
In [40]:

```
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 depre
cated function and will be removed in a future version. Please adapt your co
de to use either `displot` (a figure-level function with similar flexibilit
y) or `histplot` (an axes-level function 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 depre
cated function and will be removed in a future version. Please adapt your co
de to use either `displot` (a figure-level function with similar flexibilit
y) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[40]:

```
<AxesSubplot:ylabel='Density'>
```



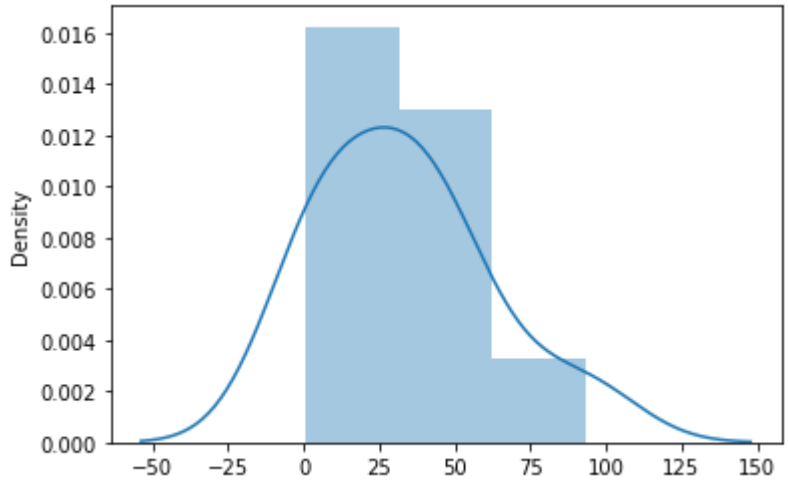
In [41]:

```
sns.distplot(filme3)
```

```
c:\users\alexsandro.ignacio\appdata\local\programs\python\python37\lib\site-
packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a depre
cated function and will be removed in a future version. Please adapt your co
de to use either `displot` (a figure-level function with similar flexibilit
y) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[41]:

```
<AxesSubplot:ylabel='Density'>
```

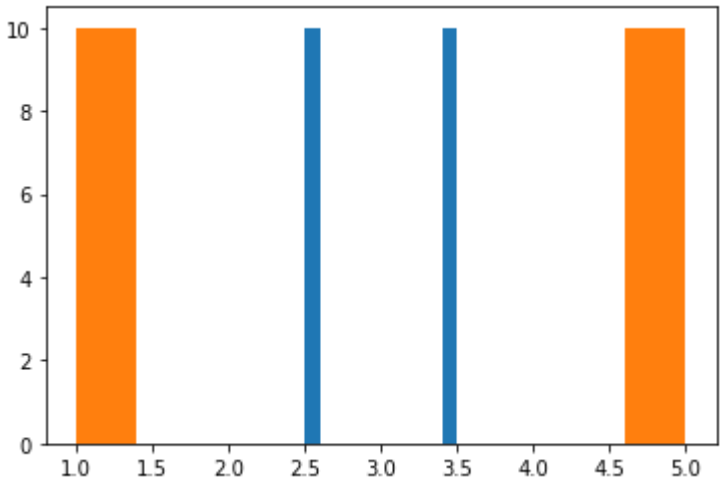


In [42]:

```
plt.hist(filme1)
plt.hist(filme2)
```

Out[42]:

```
(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>)
```

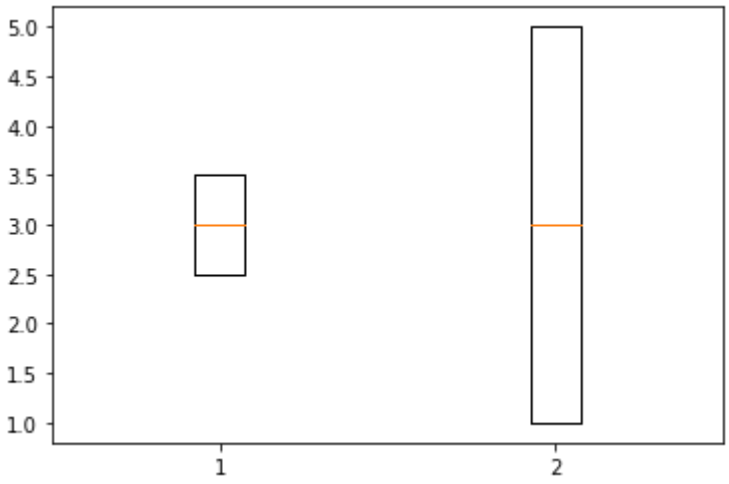


In [43]:

```
plt.boxplot([filme1, filme2])
```

Out[43]:

```
{'whiskers': [<matplotlib.lines.Line2D at 0x19243cdf2c8>,
  <matplotlib.lines.Line2D at 0x19243ce9948>,
  <matplotlib.lines.Line2D at 0x19243cf0f08>,
  <matplotlib.lines.Line2D at 0x19243cee308>],
 'caps': [<matplotlib.lines.Line2D at 0x19243ce9ac8>,
  <matplotlib.lines.Line2D at 0x19243ceb708>,
  <matplotlib.lines.Line2D at 0x19243cf1c48>,
  <matplotlib.lines.Line2D at 0x19243cf1cc8>],
 'boxes': [<matplotlib.lines.Line2D at 0x19243ce9148>,
  <matplotlib.lines.Line2D at 0x19243b5ea88>],
 'medians': [<matplotlib.lines.Line2D at 0x19243ceb808>,
  <matplotlib.lines.Line2D at 0x19243cf1e08>],
 'fliers': [<matplotlib.lines.Line2D at 0x19243cee608>,
  <matplotlib.lines.Line2D at 0x19243cf3a48>],
 'means': []}
```



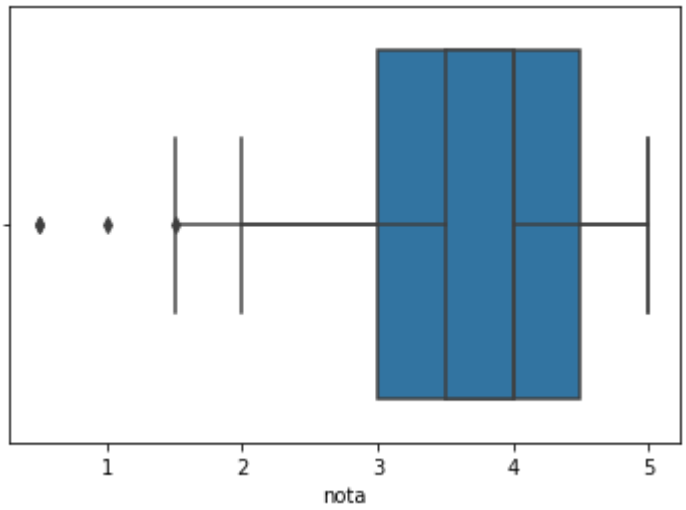
In [44]:

```
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 variab
le as a keyword arg: x. From version 0.12, the only valid positional argumen
t will be `data`, and passing other arguments without an explicit keyword wi
ll result in an error or misinterpretation.
  FutureWarning
c:\users\alexsandro.ignacio\appdata\local\programs\python\python37\lib\site-
packages\seaborn\_decorators.py:43: FutureWarning: Pass the following variab
le as a keyword arg: x. From version 0.12, the only valid positional argumen
t will be `data`, and passing other arguments without an explicit keyword wi
ll result in an error or misinterpretation.
  FutureWarning
```

Out[44]:

```
<AxesSubplot:xlabel='nota'>
```

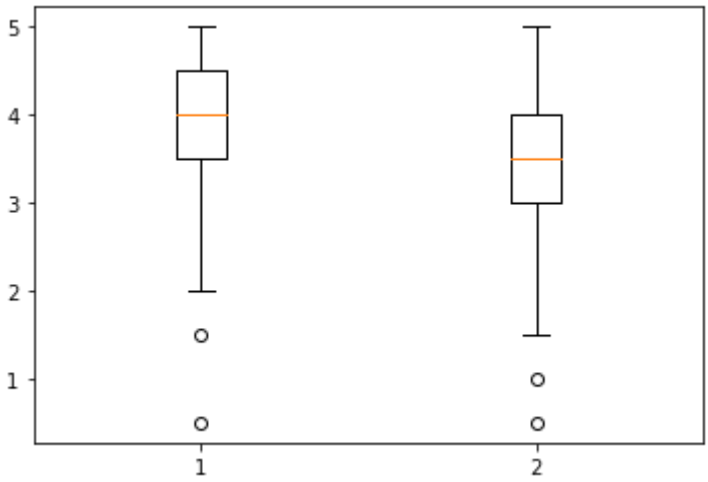


In [45]:

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

Out[45]:

```
{'whiskers': [<matplotlib.lines.Line2D at 0x19243d481c8>,
  <matplotlib.lines.Line2D at 0x19243d52f48>,
  <matplotlib.lines.Line2D at 0x19243d57888>,
  <matplotlib.lines.Line2D at 0x19243d566c8>],
 'caps': [<matplotlib.lines.Line2D at 0x19243d53b88>,
  <matplotlib.lines.Line2D at 0x19243d53bc8>,
  <matplotlib.lines.Line2D at 0x19243d5a508>,
  <matplotlib.lines.Line2D at 0x19243d5a648>],
 'boxes': [<matplotlib.lines.Line2D at 0x19243d52488>,
  <matplotlib.lines.Line2D at 0x19243d2c488>],
 'medians': [<matplotlib.lines.Line2D at 0x19243d53d48>,
  <matplotlib.lines.Line2D at 0x19243d5bd88>],
 'fliers': [<matplotlib.lines.Line2D at 0x19243d56a48>,
  <matplotlib.lines.Line2D at 0x19243d5be08>],
 'means': []}
```

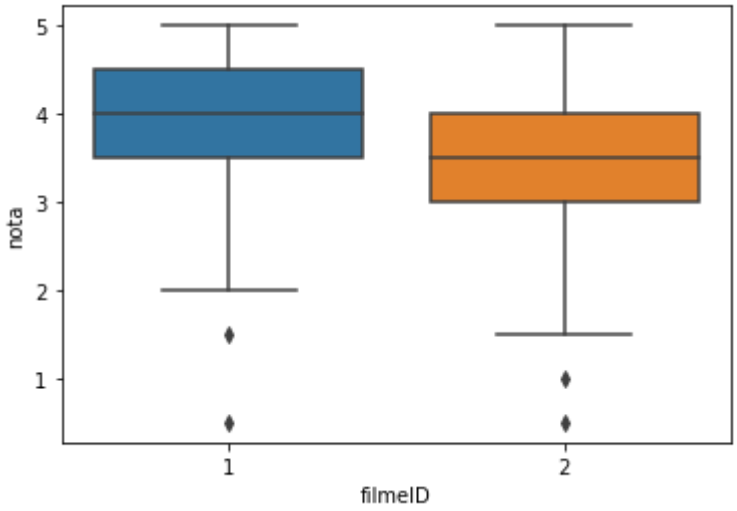


In [46]:

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

Out[46]:

```
<AxesSubplot:xlabel='filmeID', ylabel='nota'>
```



In [47]:

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

Out[47]:

```
<AxesSubplot:xlabel='filmeID', ylabel='nota'>
```