In [3]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
C:\Users\samue\anaconda3\lib\site-packages\numpy\_distributor_init.py:30: Us
erWarning: loaded more than 1 DLL from .libs:
C:\Users\samue\anaconda3\lib\site-packages\numpy\.libs\libopenblas.el2c6ple4
zyw3eceviv3oxxgrn2nrfm2.gfortran-win_amd64.dll
C:\Users\samue\anaconda3\lib\site-packages\numpy\.libs\libopenblas.FB5AE2TYX
YH2IJRDKGDGQ3XBKLKTF43H.gfortran-win_amd64.dll
C:\Users\samue\anaconda3\lib\site-packages\numpy\.libs\libopenblas.WCDJNK7YV
MPZQ2ME2ZZHJJRJ3JIKNDB7.gfortran-win_amd64.dll
  warnings.warn("loaded more than 1 DLL from .libs:"
```

In [4]:

```
music_df = pd.read_excel('Music_DB.xlsx')
```

Análises Gerais

Verificando os valores únicos de cada coluna

In [5]:

```
music_df.head()
```

Out[5]:

	Energia	Ao_vivo	Tempo	Falado	Orgânico	Instrumental	Compasso	Dançabilidade	Tom
0	0.964	0.1060	167.024	0.0644	0.000071	0.038700	4	0.392	10
1	0.955	0.3800	107.984	0.0653	0.000165	0.000014	4	0.533	8
2	0.939	0.6230	150.184	0.1130	0.000220	0.000000	4	0.524	9
3	0.969	0.0787	95.136	0.0491	0.001400	0.000001	4	0.492	9
4	0.970	0.0994	94.952	0.2120	0.030000	0.000000	4	0.512	8

In [61]:

```
music_df.describe()
```

Out[61]:

	Energia	Ao_vivo	Tempo	Falado	Orgânico	Instrumental	Compa
count	7702.000000	7702.000000	7702.000000	7702.000000	7702.000000	7702.000000	7702.00(
mean	0.461560	0.159612	113.737695	0.066433	0.486352	0.398859	3.84
std	0.335351	0.135808	31.989253	0.063606	0.412783	0.429689	0.56
min	0.000081	0.022400	0.000000	0.000000	0.000000	0.000000	0.00(
25%	0.113000	0.094300	88.666000	0.036900	0.037000	0.000010	4.00(
50%	0.474000	0.110000	114.061500	0.045200	0.458500	0.076500	4.00(
75%	0.770000	0.157000	133.418250	0.066600	0.944000	0.896000	4.00(
max	1.000000	0.988000	236.136000	0.896000	0.996000	1.000000	5.00(

Verificando os tipos de música no dataset

In [6]:

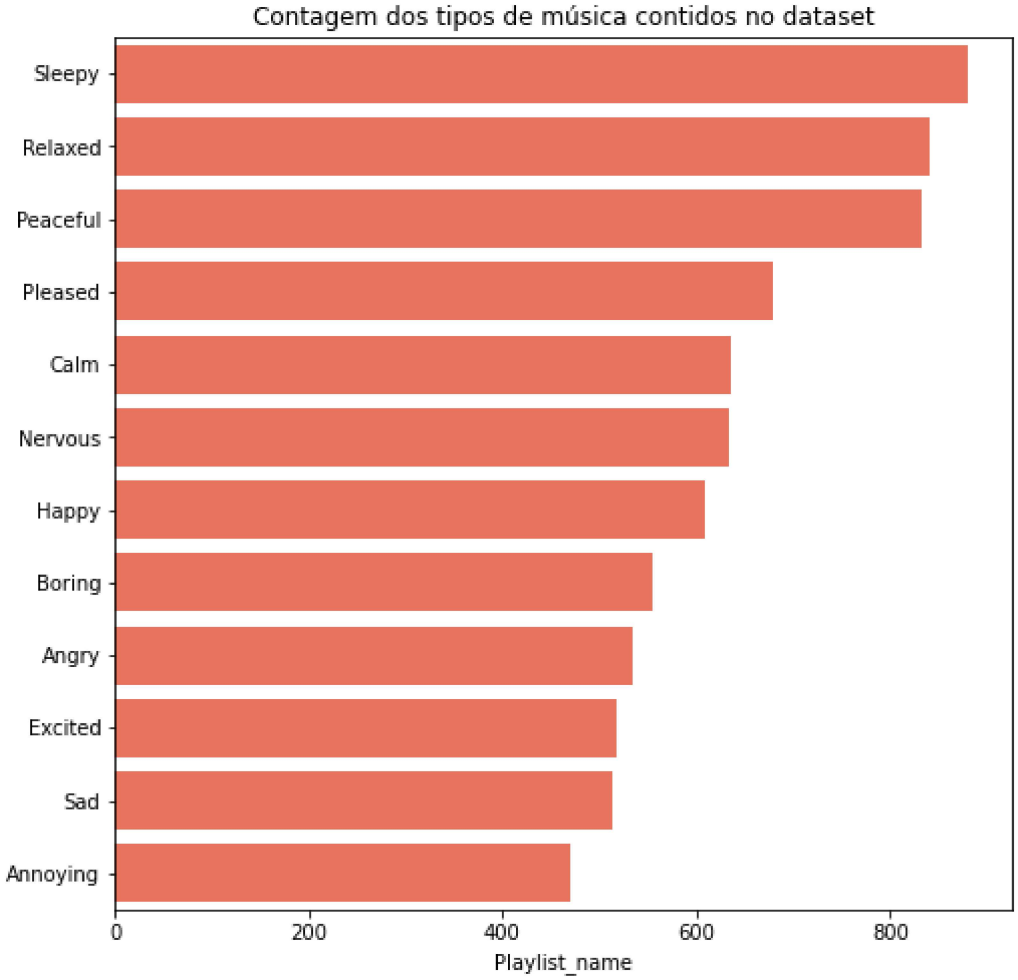
```
music_df.Playlist_name.value_counts()
```

Out[6]:

```
Sleepy      881
Relaxed     840
Peaceful    833
Pleased     678
Calm        635
Nervous     634
Happy       609
Boring      555
Angry       535
Excited     518
Sad         514
Annoying    470
Name: Playlist_name, dtype: int64
```

In [7]:

```
plt.figure(figsize=(8,8))
sns.barplot(y = music_df.Playlist_name.value_counts().index,  x = music_df.Playlist_name.va
plt.title('Contagem dos tipos de música contidos no dataset')
plt.show()
```



Verificando os artistas no dataset

In [8]:

```
music_df.Artista.value_counts().head(10)
```

Out[8]:

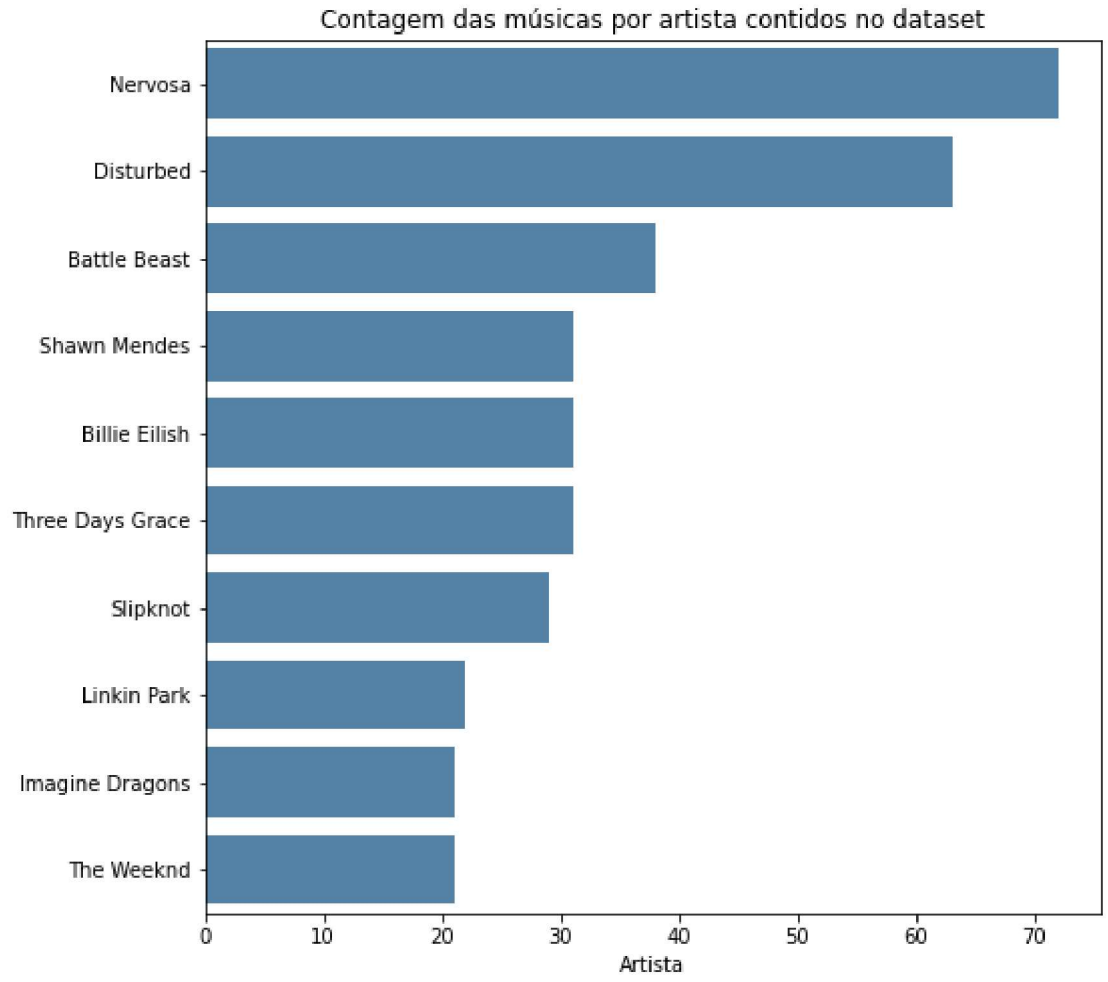
```
Nervosa             72
Disturbed           63
Battle Beast        38
Shawn Mendes        31
Billie Eilish       31
Three Days Grace    31
Slipknot            29
Linkin Park         22
Imagine Dragons     21
The Weeknd          21
Name: Artista, dtype: int64
```

In [128]:

```
top_10_bandas = list(music_df.Artista.value_counts().head(10).index)
```

In [129]:

```
plt.figure(figsize=(8,8))
sns.barplot(y = top_10_bandas,  x = music_df.Artista.value_counts().head(10), color='steelb
plt.title('Contagem das músicas por artista contidos no dataset')
plt.show()
```



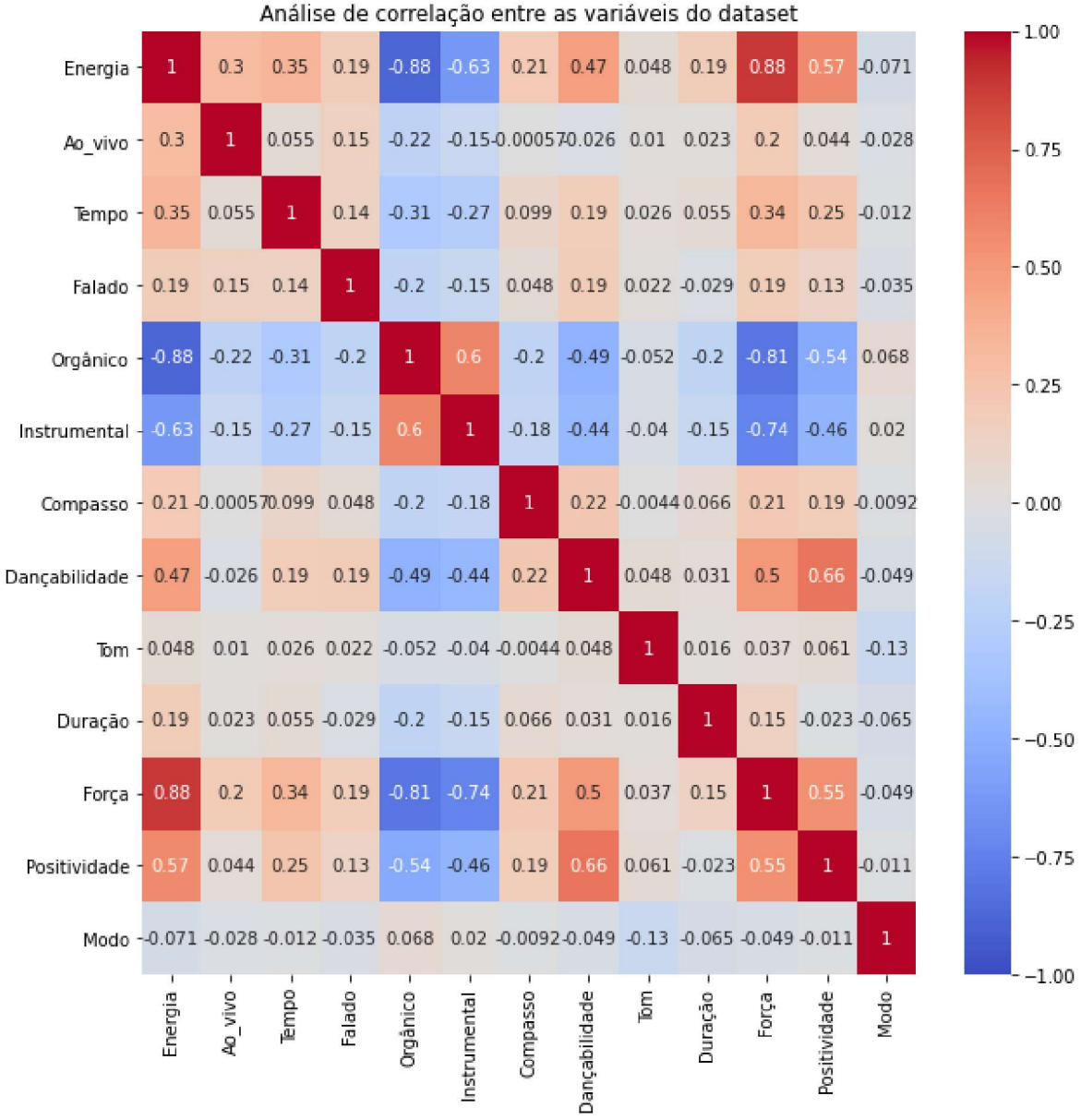
Análise de correlação de pearson entre as variáveis do dataset

In [10]:

```
# plt.figure(figsize=(10,10))
# corr = music_df.corr()
# mask = np.zeros_like(corr)
# mask[np.triu_indices_from(mask)] = True
# with sns.axes_style("white"):
#     f, ax = plt.subplots(figsize=(10,10))
# #     ax = sns.heatmap(corr, mask=mask, vmax=.3, square=True)
#     ax = sns.heatmap(music_df.corr(), vmin = -1, vmax = 1, cmap='coolwarm', annot=True, m
```

In [11]:

```
plt.figure(figsize=(10,10))
sns.heatmap(music_df.corr(), vmin = -1, vmax = 1, cmap='coolwarm', annot=True)
plt.title('Análise de correlação entre as variáveis do dataset')
plt.show()
```



- Podemos ver que as variáveis **Energia e Força** têm uma forte correlação positiva enquanto **Energia e Orgânico** e **Energia e Instrumental** têm correlação negativa
- Podemos ver que **Positividade** está relacionado positivamente com **Força, Energia e Dançabilidade**, e negativamente com **Ôrgânico e Instrumental**

Pairplot das variáveis

In [12]:

```
not_used_cols = ['Tipo', 'Music_id', 'Playlist_name', 'Album_name', 'Artista', 'Music_name'
```

In [19]:

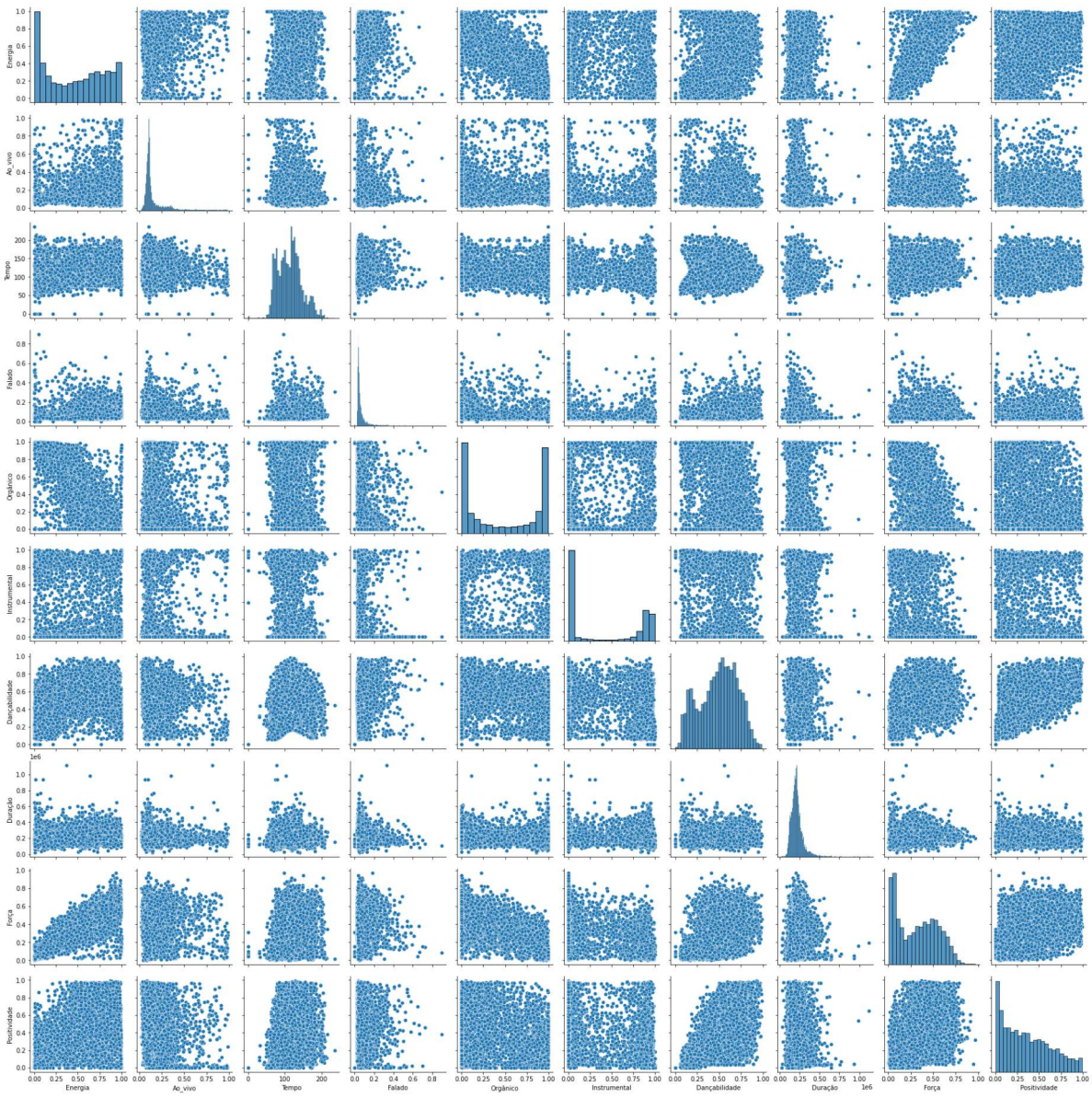
```
numerical_cols = music_df.drop(not_used_cols, axis = 1).columns
```

In [13]:

```
sns.pairplot(music_df.drop(not_used_cols, axis = 1))
```

Out[13]:

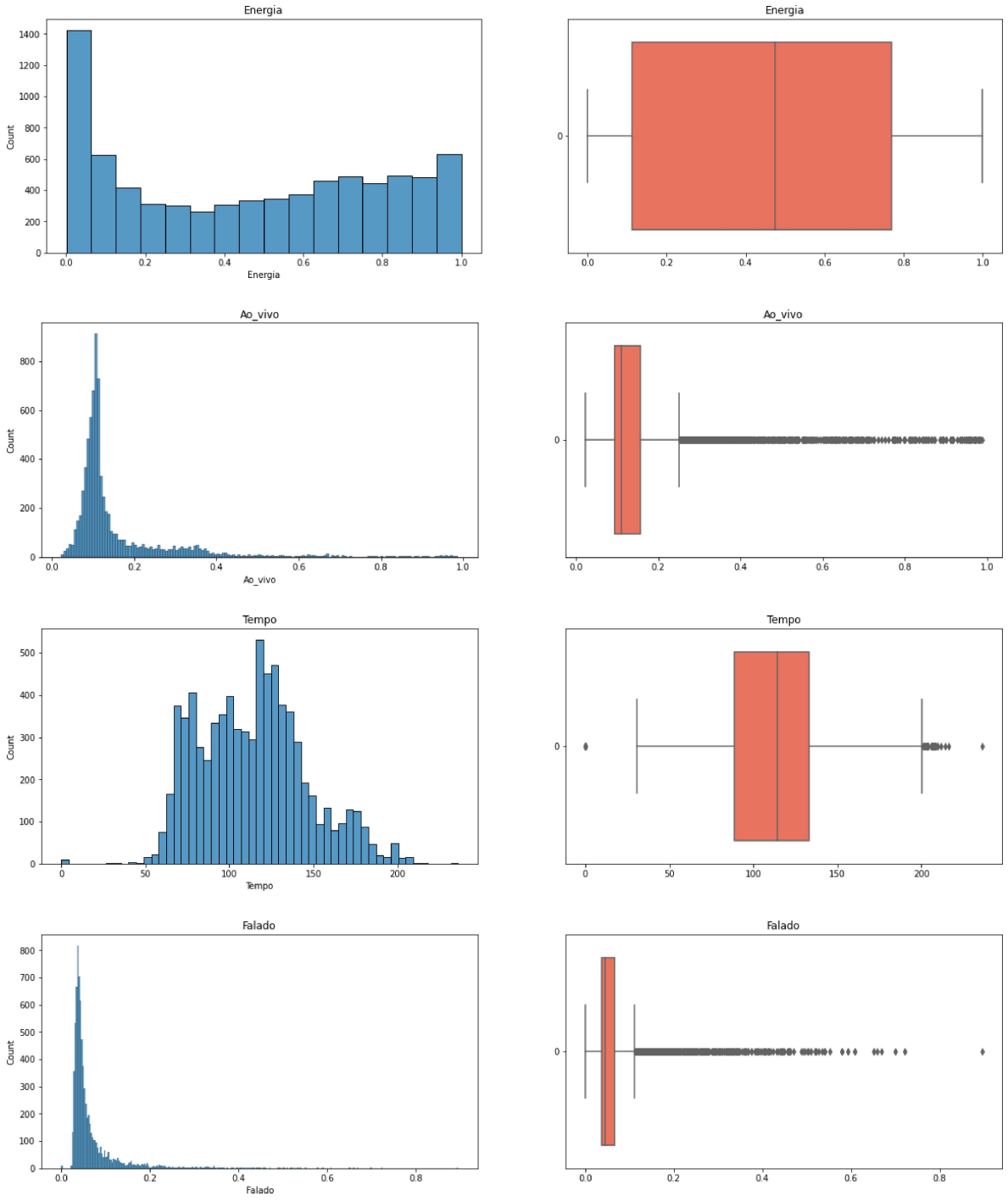
```
<seaborn.axisgrid.PairGrid at 0x1bc7c338a00>
```

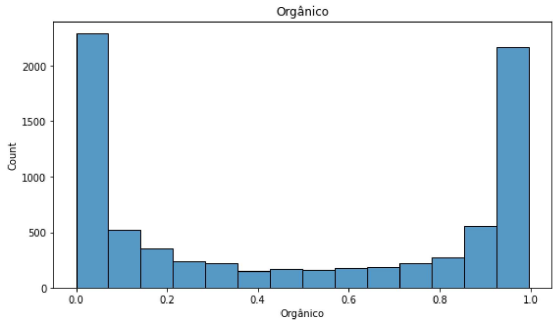


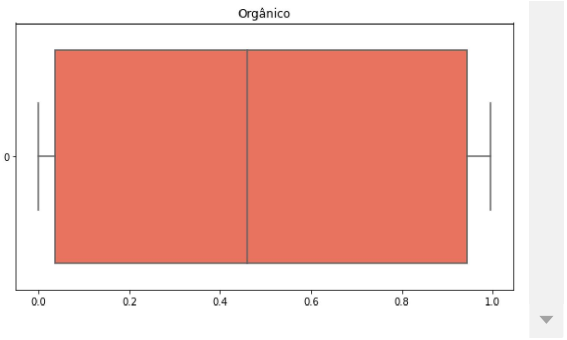
Análise geral de distrbuição

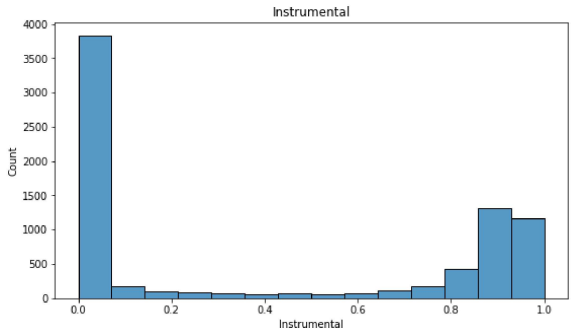
In [35]:

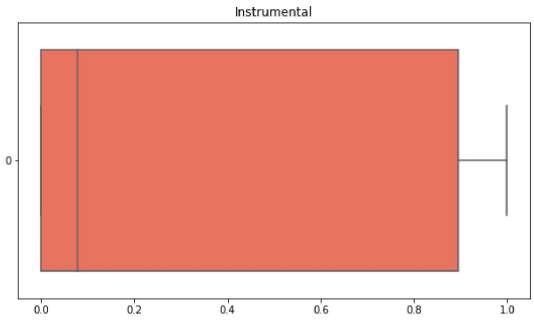
```
for col in numerical_cols:

    plt.figure(figsize=(20,5))
    plt.subplot(1,2,1)
    sns.histplot(music_df[col])
    plt.title(col)

    plt.subplot(1,2,2)
    sns.boxplot(data = music_df[col], orient='h', color = 'tomato')
    plt.title(col)
    plt.show()
```

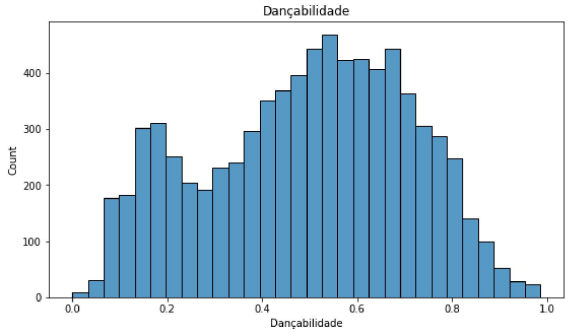


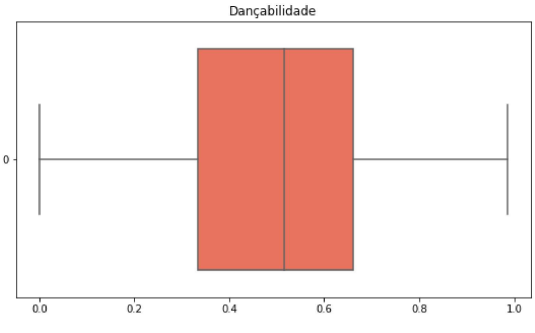

Orgânico
Count
Orgânico

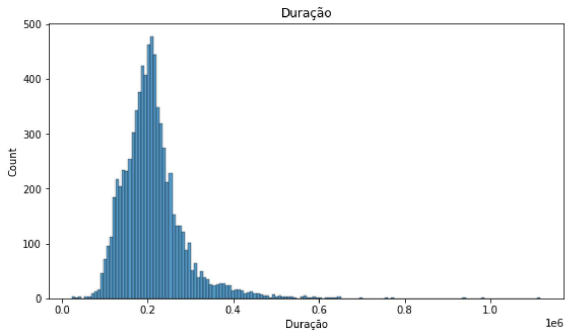

Orgânico
0

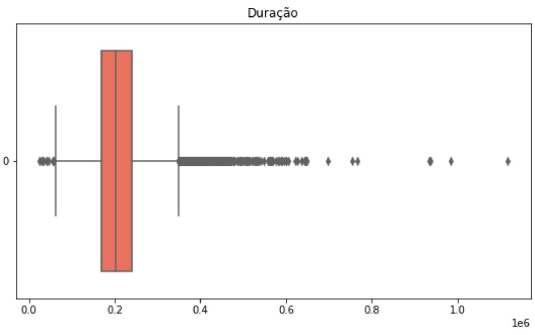

Instrumental
Count
Instrumental

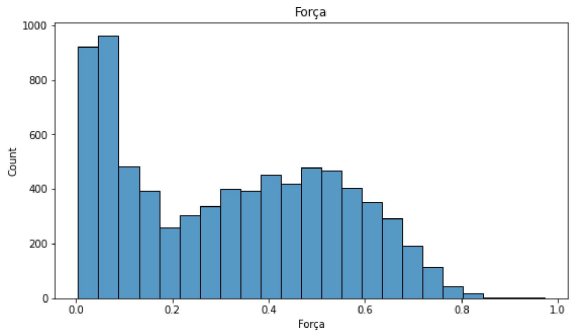

Instrumental
0

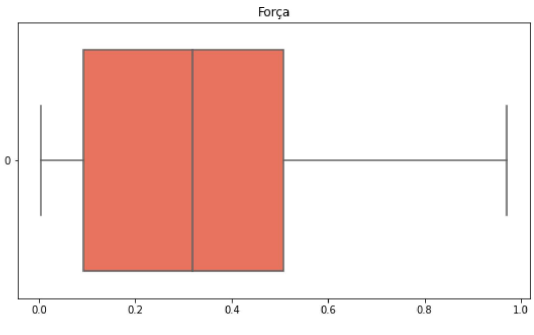

Dançabilidade
Count
Dançabilidade

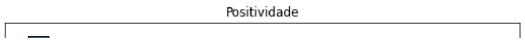

Dançabilidade
0

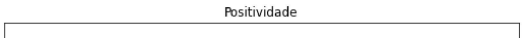

Duração
Count
Duração
1e6


Duração
0
1e6


Força
Count
Força


Força
0





Análise de variáveis categóricas

Podemos considerar variáveis categóricas:

- Compasso
- Tom
- Modo
- Artista
- Album_name
- Playlist_name

Para a finalidade destas análises, o nome da música não faz diferença, por isso não foi considerada.

In [36]:

```
cat_vars = ['Compasso', 'Tom', 'Modo', 'Playlist_name']
all_cat_vars = cat_vars + ['Artista', 'Album_name']
```

In [16]:

```
for col in music_df.columns:
    print(col, len(music_df[col].unique()))
```

```
Energia 1756
Ao_vivo 1100
Tempo 6093
Falado 988
Orgânico 2324
Instrumental 2343
Compasso 5
Dançabilidade 1001
Tom 12
Duração 6029
Força 5724
Positividade 1432
Modo 2
Tipo 1
Music_id 6722
Playlist_name 12
Album_name 4228
Artista 3112
Music_name 4747
```

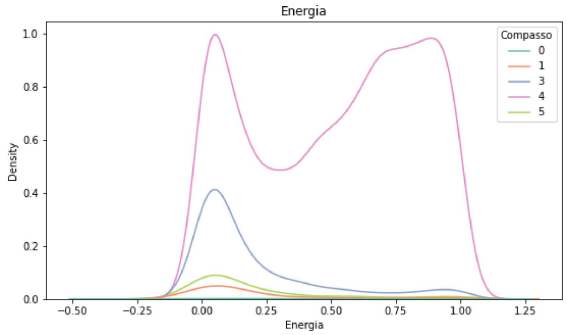
In [45]:

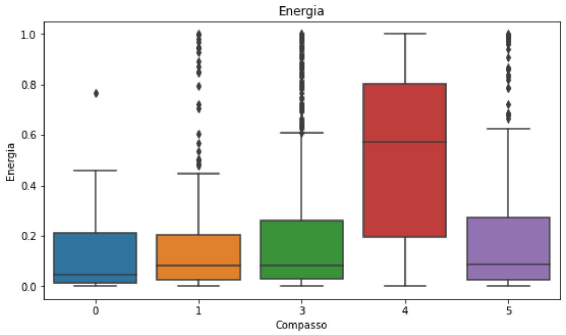
```
for col in cat_vars:
    print(col, (music_df[col].unique()))
```

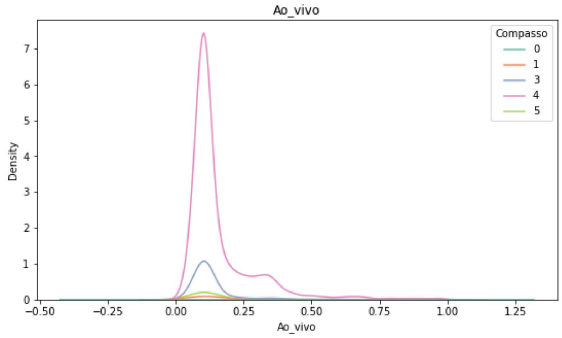
```
Compasso [4 3 5 1 0]
Tom [10  8  9  3 11  6  7  0  2  5  4  1]
Modo [0 1]
Playlist_name ['Angry' 'Annoying' 'Boring' 'Calm' 'Excited' 'Happy' 'Nervou
s' 'Peaceful'
 'Pleased' 'Relaxed' 'Sad' 'Sleepy']
```

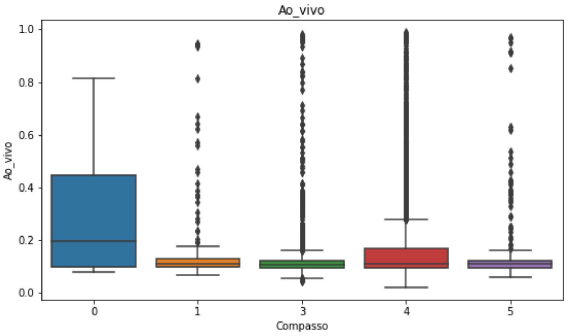
In [60]:

```
for cat_col in cat_vars:
    for num_col in numerical_cols:
        plt.figure(figsize=(20,5))

        plt.subplot(1,2,1)
        sns.kdeplot(data = music_df, x = num_col, hue=cat_col, palette= "Set2")
        plt.title(num_col)

        plt.subplot(1,2,2)
        sns.boxplot(data = music_df, x = cat_col, y = num_col, orient='v')
        plt.title(num_col)
        plt.show()

#         break
#     break
```

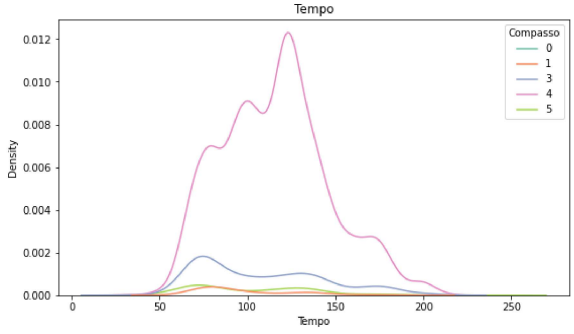


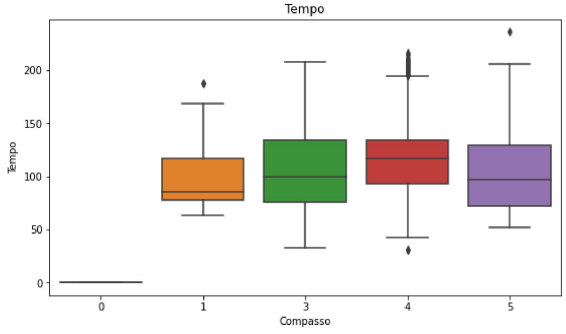




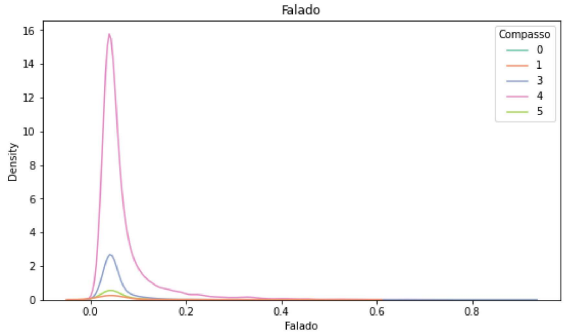


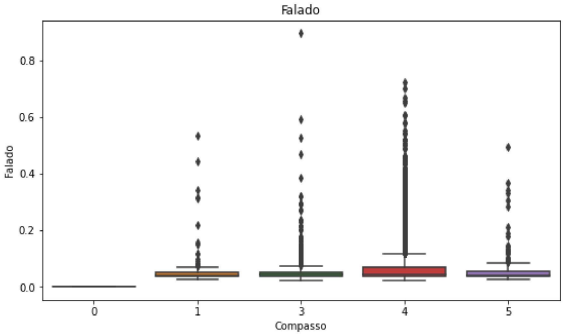

```
C:\Users\samue\anaconda3\lib\site-packages\seaborn\distributions.py:305: Use
rWarning: Dataset has 0 variance; skipping density estimate.
  warnings.warn(msg, UserWarning)
```

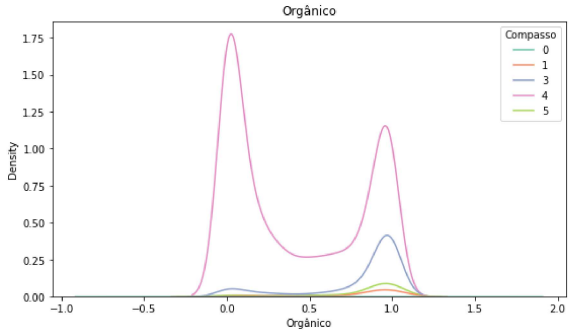


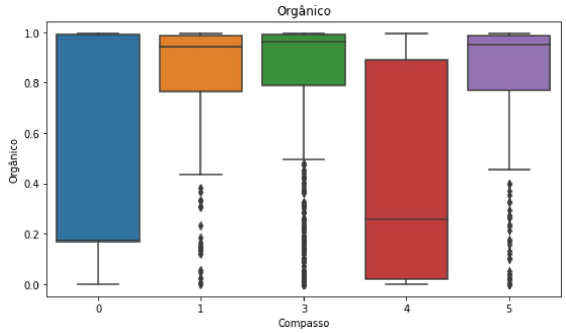


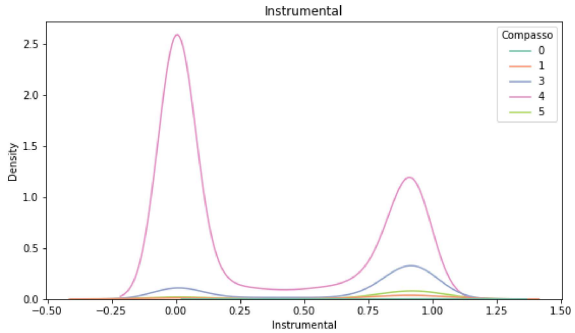

```
C:\Users\samue\anaconda3\lib\site-packages\seaborn\distributions.py:305: Use
rWarning: Dataset has 0 variance; skipping density estimate.
  warnings.warn(msg, UserWarning)
```

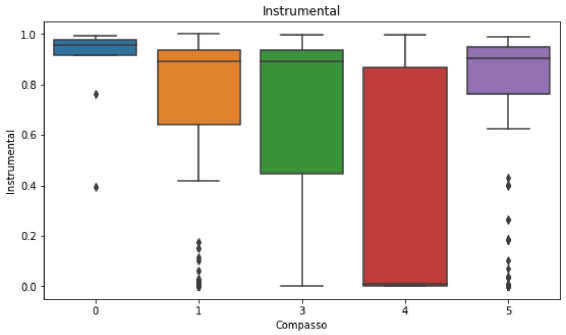




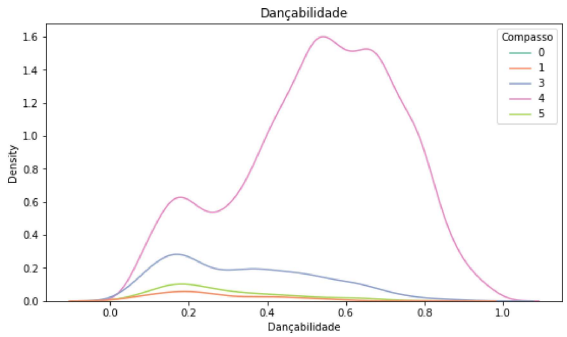


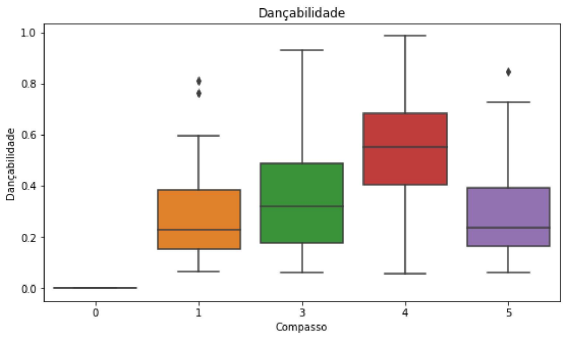


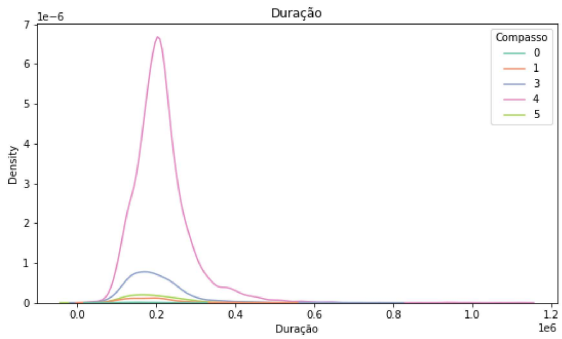


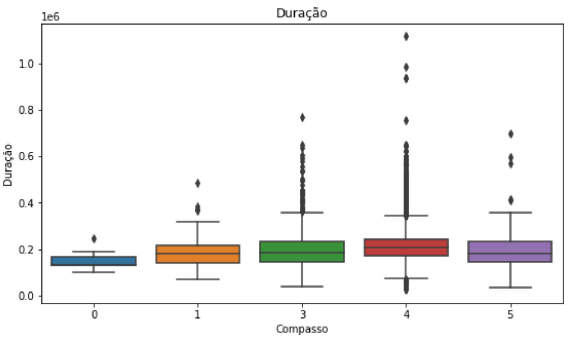


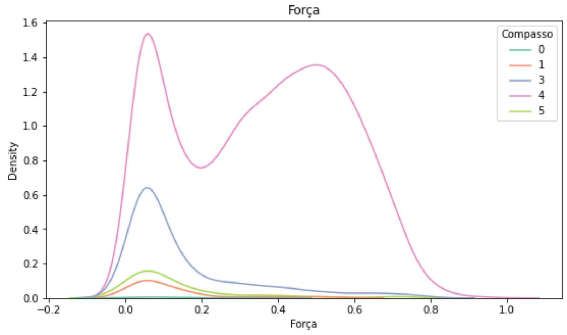

```
C:\Users\samue\anaconda3\lib\site-packages\seaborn\distributions.py:305: Use
rWarning: Dataset has 0 variance; skipping density estimate.
  warnings.warn(msg, UserWarning)
```

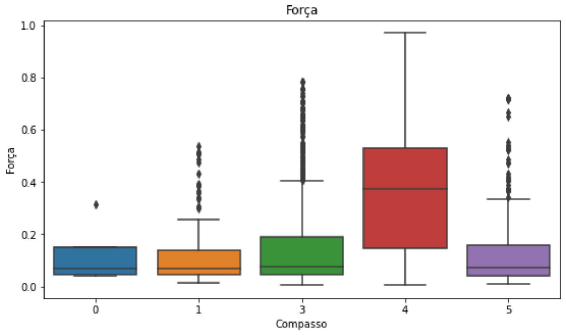




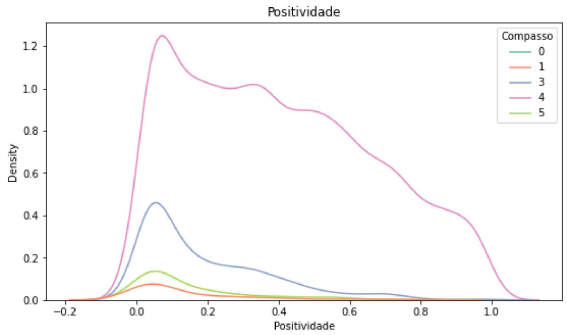


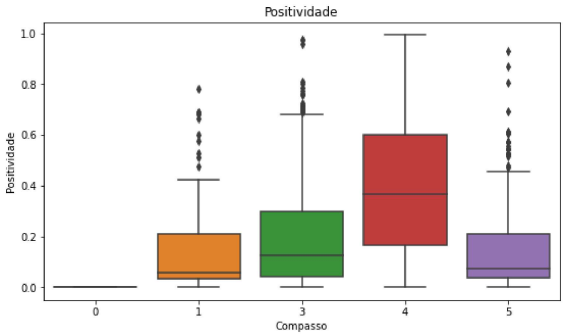


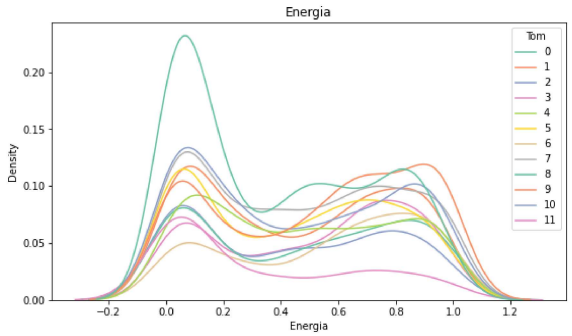


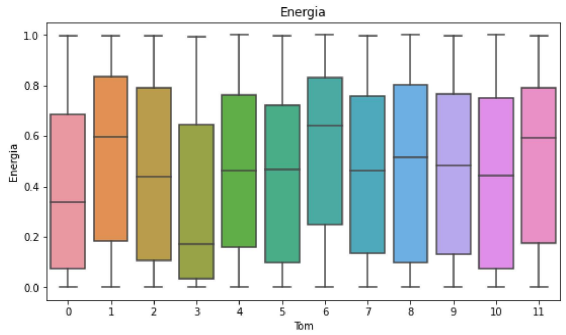


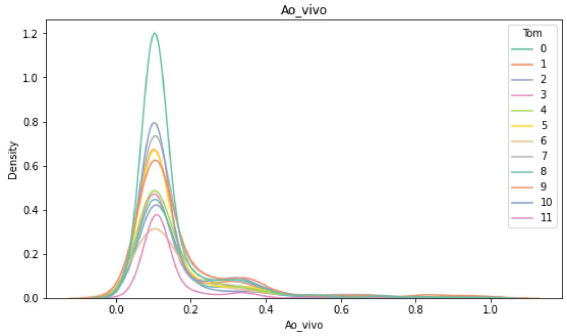

```
C:\Users\samue\anaconda3\lib\site-packages\seaborn\distributions.py:305: Use
rWarning: Dataset has 0 variance; skipping density estimate.
  warnings.warn(msg, UserWarning)
```

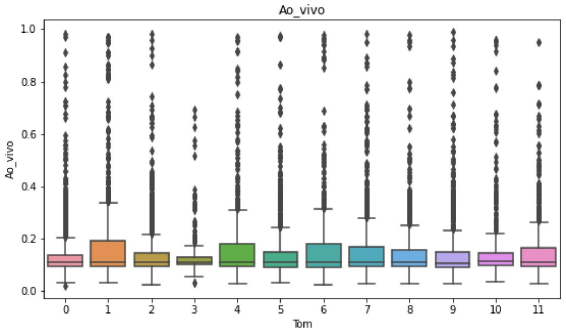


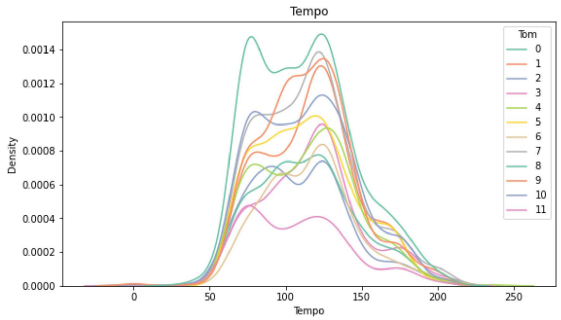


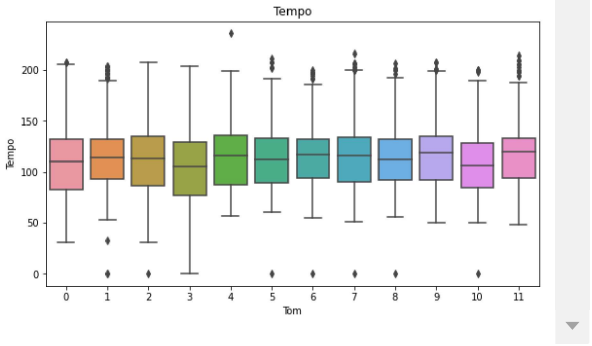


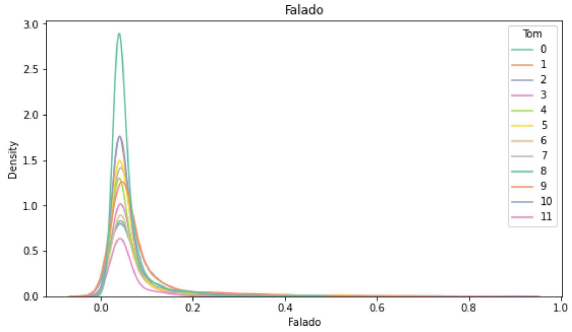


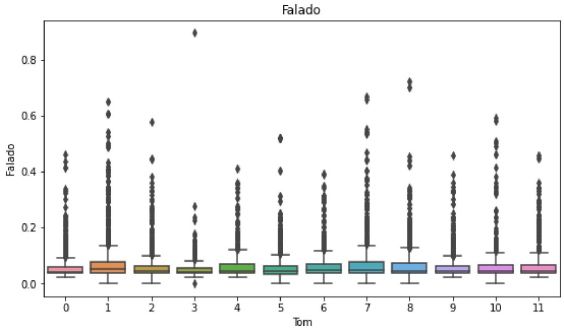


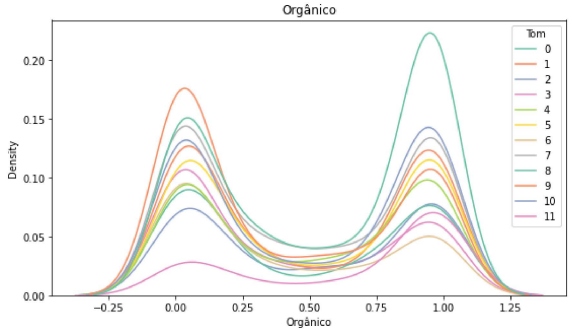


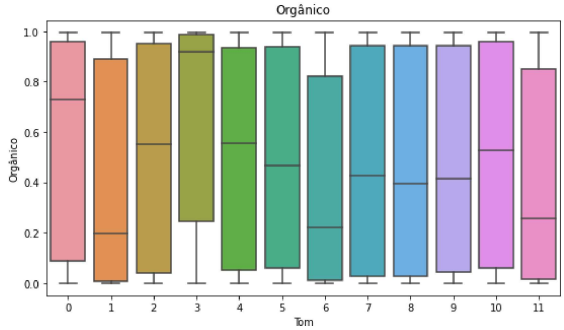

Tempo
Tom
0
1
2
3
4
5
6
7
8
9
10
11
Density
Tempo

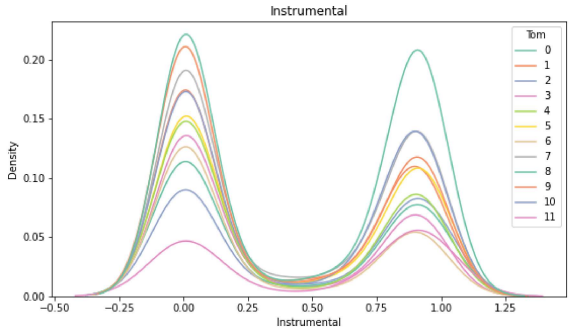

Tempo
Tempo
Tom

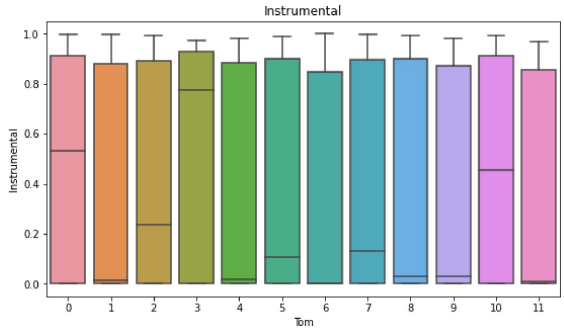

Falado
Tom
Density
Falado

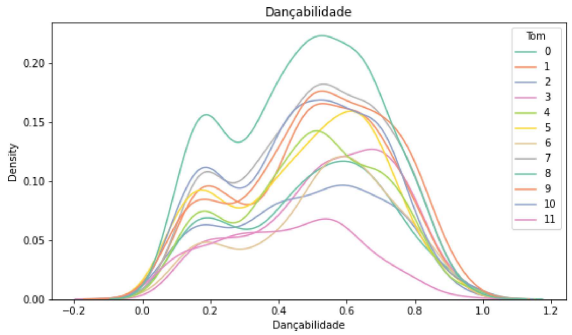

Falado
Falado
Tom

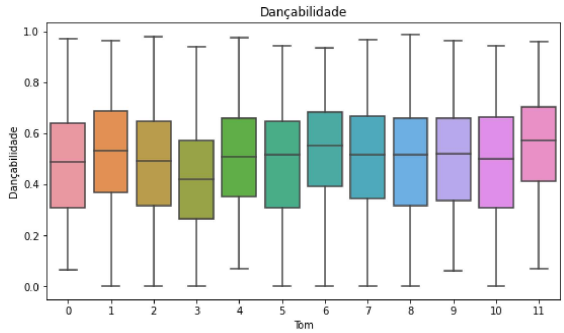

Orgânico
Tom
Density
Orgânico

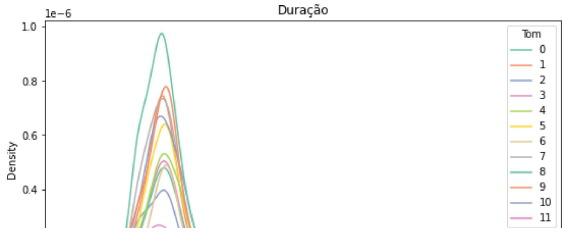

Orgânico
Orgânico
Tom

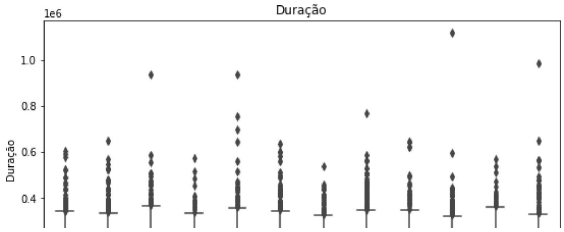

Instrumental
Tom
Density
Instrumental

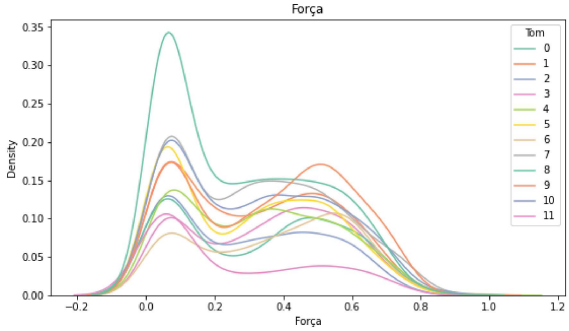

Instrumental
Instrumental
Tom

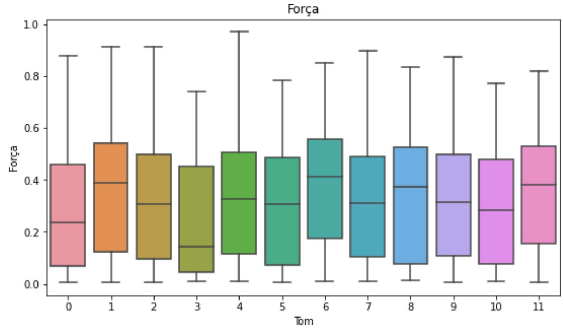

Dançabilidade
Tom
Density
Dançabilidade

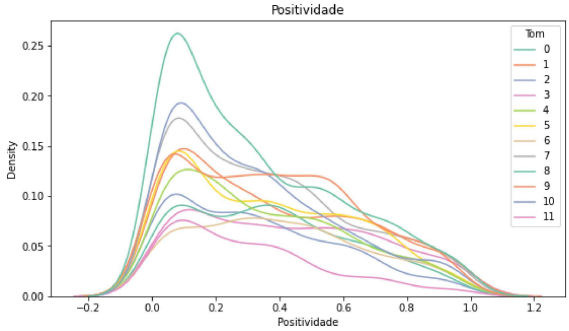

Dançabilidade
Dançabilidade
Tom

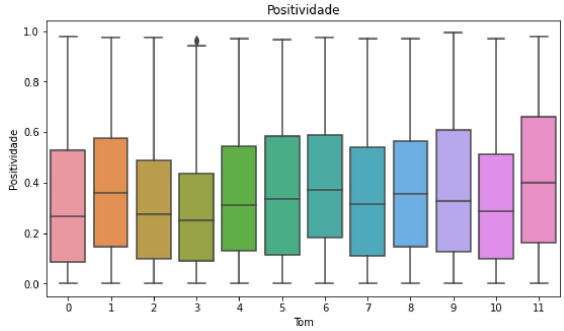

Duração
1e−6
Density
Tom
0
1
2
3
4
5
6
7
8
9
10
11

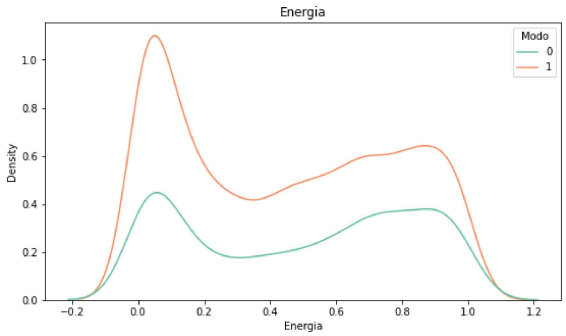

Duração
1e6
Duração

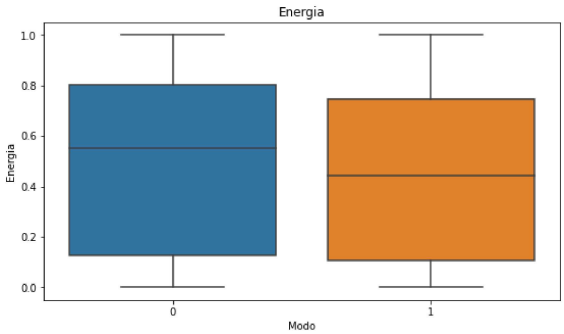

Força
Density
Força
Tom
0
1
2
3
4
5
6
7
8
9
10
11

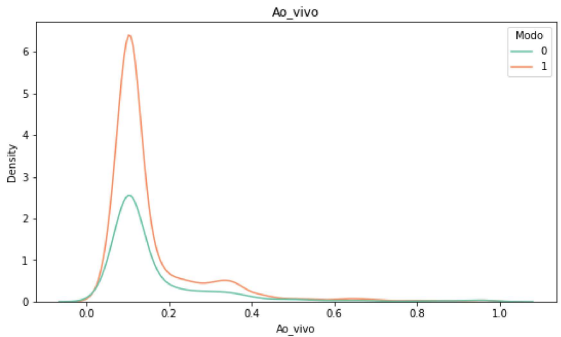

Força
Força
Tom

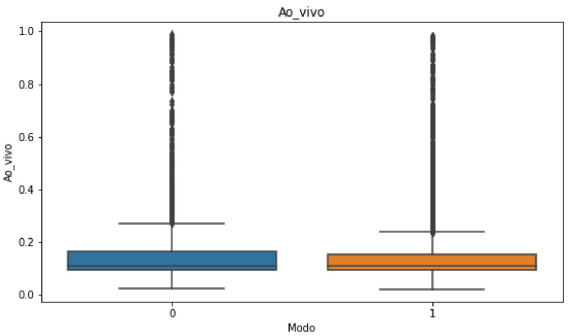

Positividade
Density
Positividade
Tom
0
1
2
3
4
5
6
7
8
9
10
11

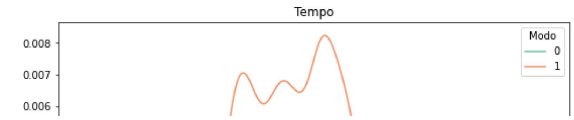

Positividade
Positividade
Tom

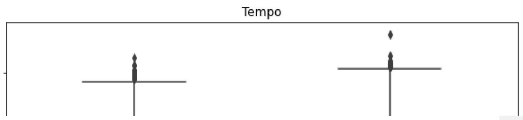

Energia
Density
Energia
Modo
0
1

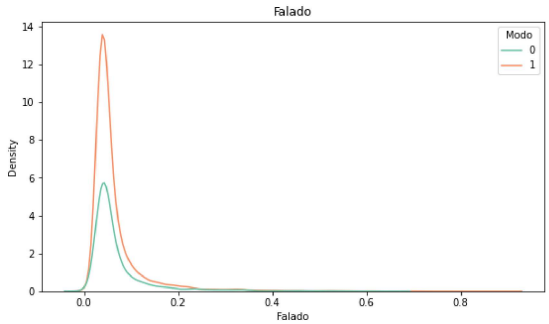

Energia
Energia
Modo

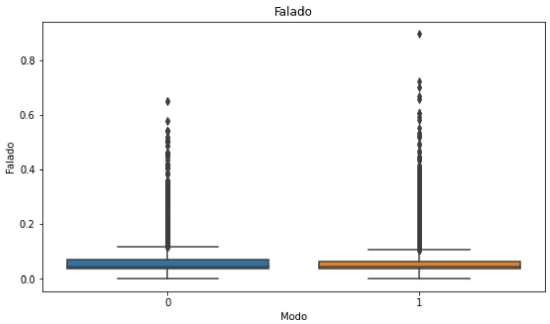

Ao_vivo
Density
Ao_vivo
Modo
0
1

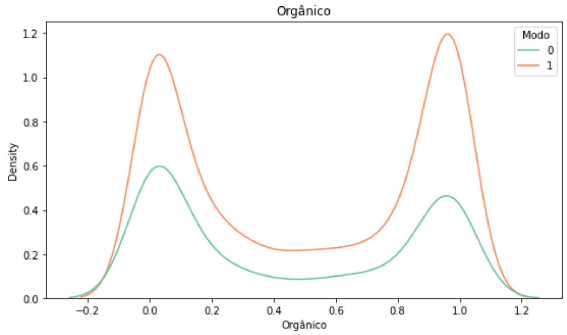

Ao_vivo
Ao_vivo
Modo

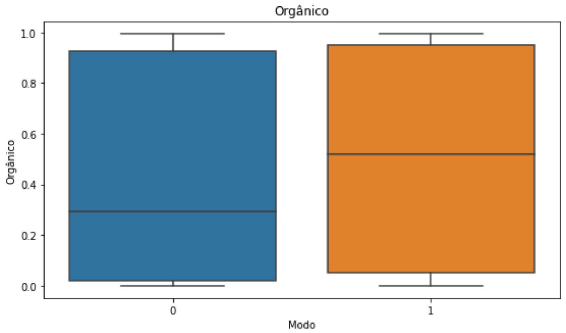

Tempo
Modo
0
1
0.008
0.007
0.006

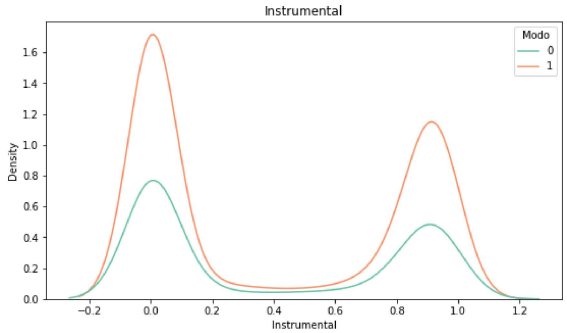

Tempo
200

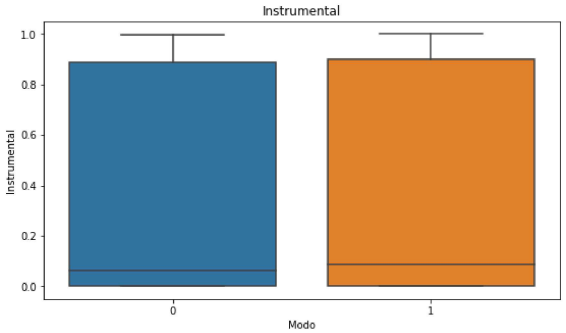

Falado
Modo
0
1
Density
Falado

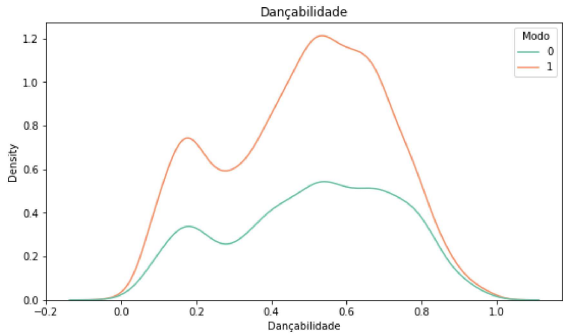

Falado
Falado
Modo

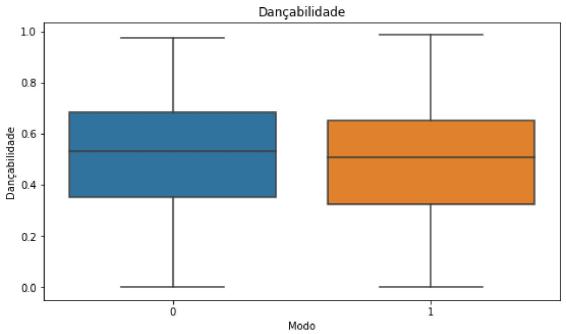

Orgânico
Modo
0
1
Density
Orgânico

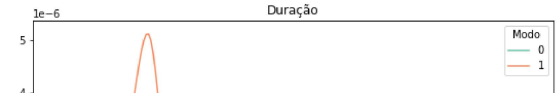

Orgânico
Orgânico
Modo

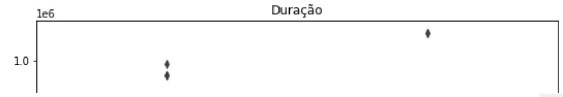

Instrumental
Modo
0
1
Density
Instrumental

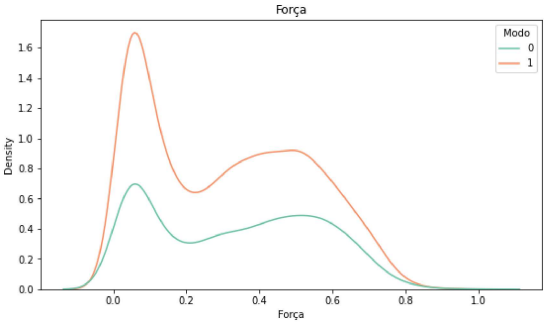

Instrumental
Instrumental
Modo

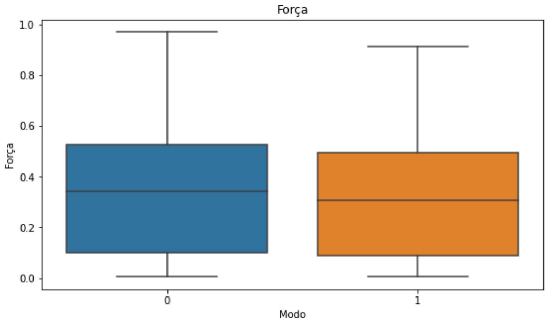

Dançabilidade
Modo
0
1
Density
Dançabilidade

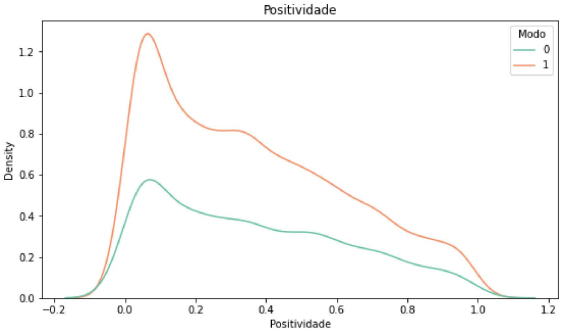

Dançabilidade
Dançabilidade
Modo

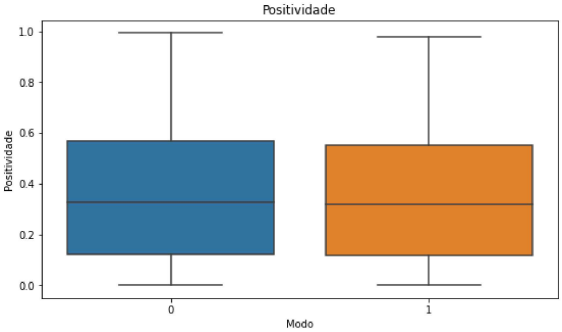

Duração
1e−6
Modo
0
1

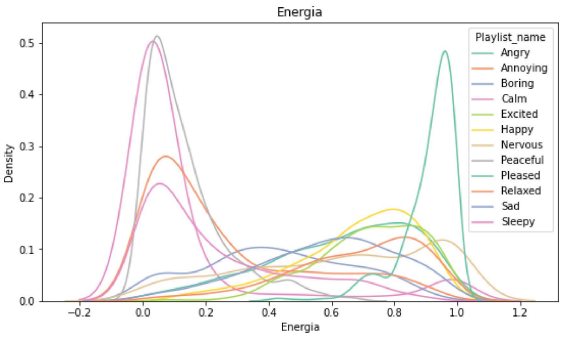

Duração
1e6

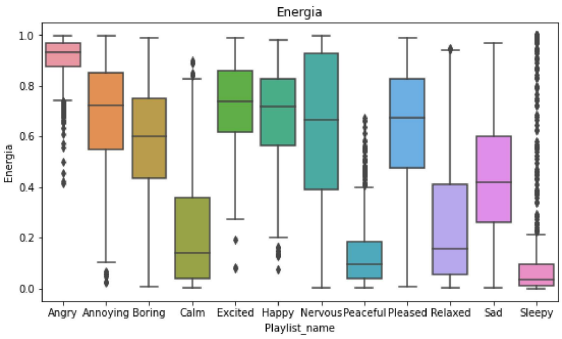

Força
Modo
0
1
Density
Força

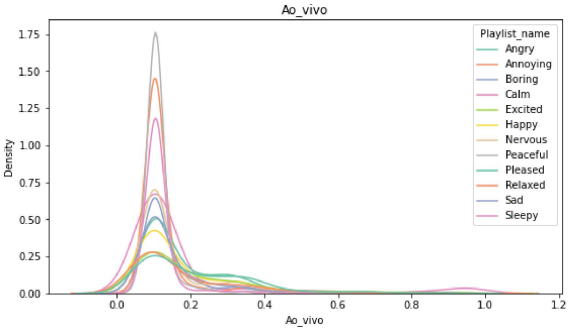

Força
Força
Modo

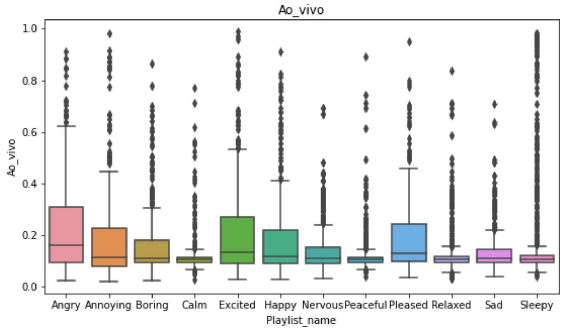

Positividade
Modo
0
1
Density
Positividade

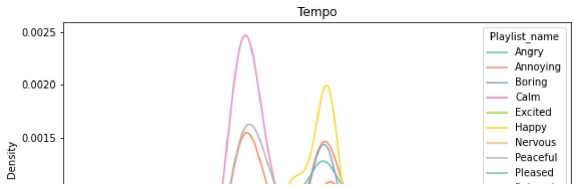

Positividade
Positividade
Modo

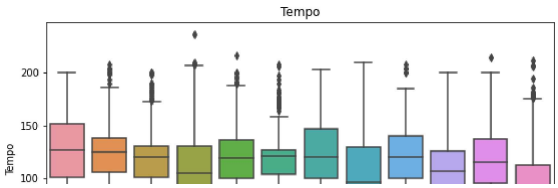

Energia
Playlist_name
Angry
Annoying
Boring
Calm
Excited
Happy
Nervous
Peaceful
Pleased
Relaxed
Sad
Sleepy
Density
Energia

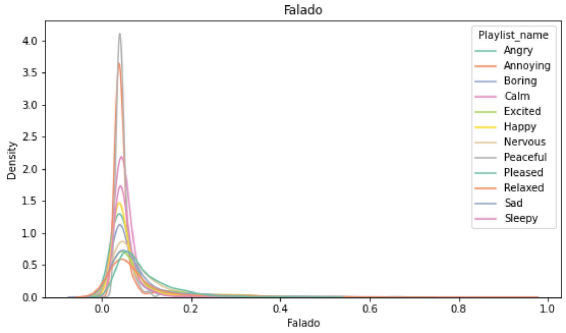

Energia
Energia
Angry Annoying Boring Calm Excited Happy Nervous Peaceful Pleased Relaxed Sad Sleepy
Playlist_name

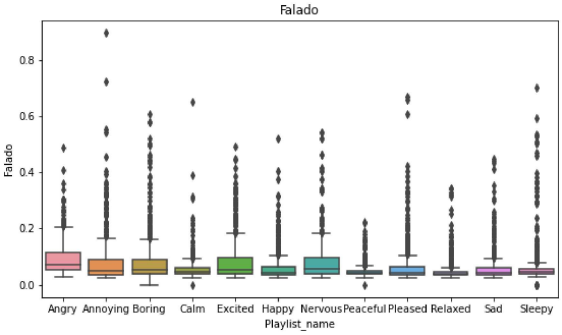

Ao_vivo
Playlist_name
Angry
Annoying
Boring
Calm
Excited
Happy
Nervous
Peaceful
Pleased
Relaxed
Sad
Sleepy
Density
Ao_vivo

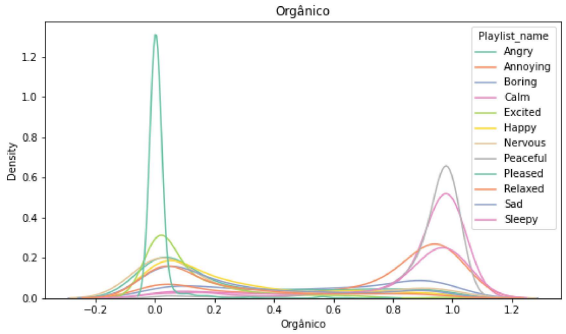

Ao_vivo
Ao_vivo
Angry Annoying Boring Calm Excited Happy Nervous Peaceful Pleased Relaxed Sad Sleepy
Playlist_name

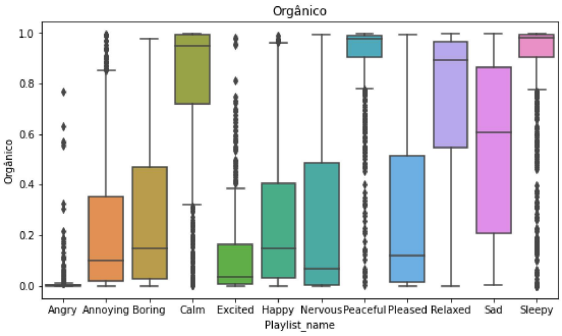

Tempo
Density
Playlist_name
Angry
Annoying
Boring
Calm
Excited
Happy
Nervous
Peaceful
Pleased

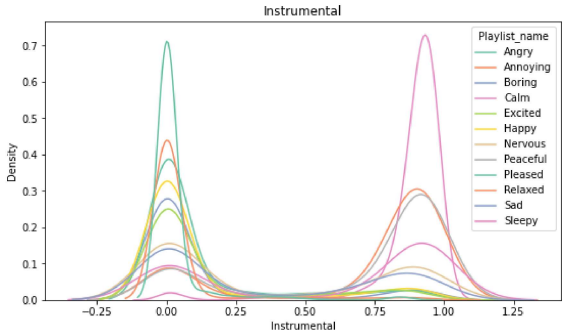

Tempo
Tempo
Angry Annoying Boring Calm Excited Happy Nervous Peaceful Pleased Relaxed Sad Sleepy
Playlist_name

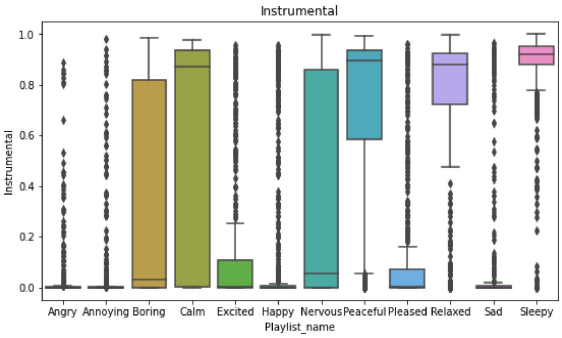

Falado
Density
Falado
Playlist_name
Angry
Annoying
Boring
Calm
Excited
Happy
Nervous
Peaceful
Pleased
Relaxed
Sad
Sleepy

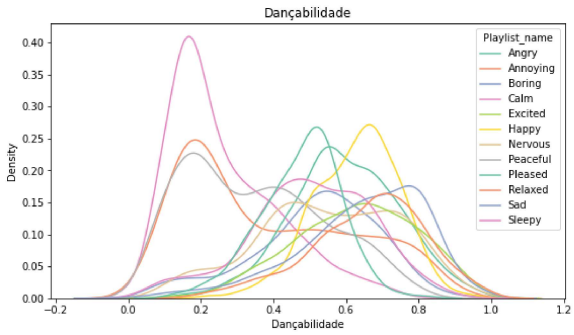

Falado
Falado
Angry Annoying Boring Calm Excited Happy Nervous Peaceful Pleased Relaxed Sad Sleepy
Playlist_name

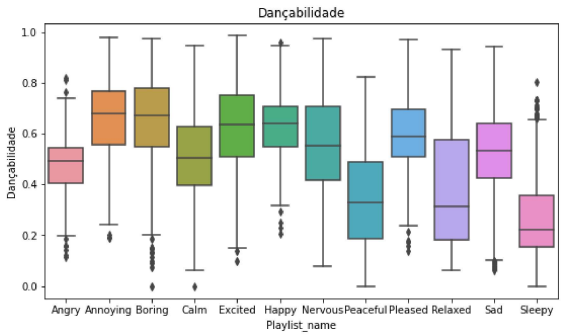

Orgânico
Density
Orgânico
Playlist_name
Angry
Annoying
Boring
Calm
Excited
Happy
Nervous
Peaceful
Pleased
Relaxed
Sad
Sleepy

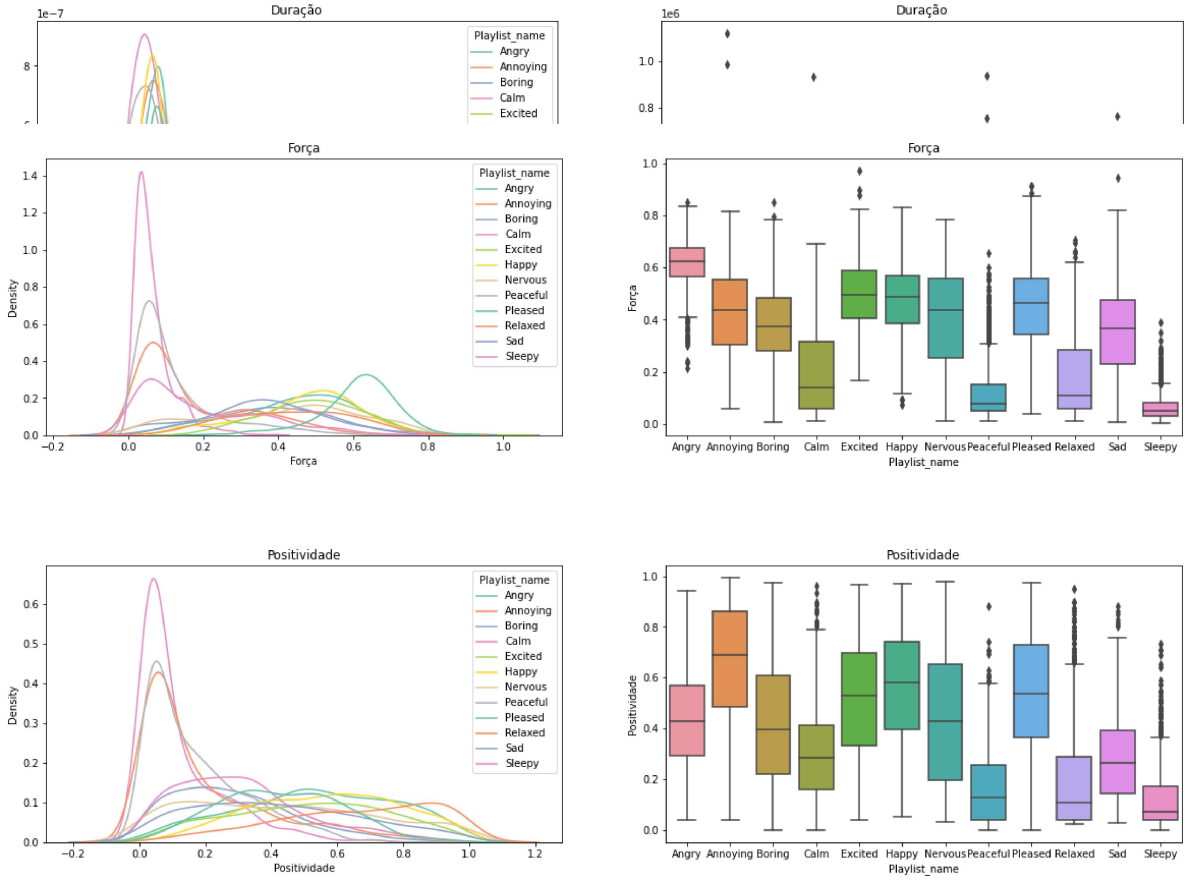

Orgânico
Orgânico
Angry Annoying Boring Calm Excited Happy Nervous Peaceful Pleased Relaxed Sad Sleepy
Playlist_name


Instrumental
Density
Instrumental
Playlist_name
Angry
Annoying
Boring
Calm
Excited
Happy
Nervous
Peaceful
Pleased
Relaxed
Sad
Sleepy


Instrumental
Instrumental
Angry Annoying Boring Calm Excited Happy Nervous Peaceful Pleased Relaxed Sad Sleepy
Playlist_name


Dançabilidade
Density
Dançabilidade
Playlist_name
Angry
Annoying
Boring
Calm
Excited
Happy
Nervous
Peaceful
Pleased
Relaxed
Sad
Sleepy


Dançabilidade
Dançabilidade
Angry Annoying Boring Calm Excited Happy Nervous Peaceful Pleased Relaxed Sad Sleepy
Playlist_name



In [65]:

```
music_df.columns
```

Out[65]:

```
Index(['Energia', 'Ao_vivo', 'Tempo', 'Falado', 'Orgânico', 'Instrumental',
       'Compasso', 'Dançabilidade', 'Tom', 'Duração', 'Força', 'Positividad
e',
       'Modo', 'Tipo', 'Music_id', 'Playlist_name', 'Album_name', 'Artista',
       'Music_name'],
      dtype='object')
```

In [122]:

```
mean_df = music_df[list(numerical_cols ) + ['Playlist_name']].groupby('Playlist_name').mean
std_df = music_df[list(numerical_cols ) + ['Playlist_name']].groupby('Playlist_name').std()
```

In [123]:

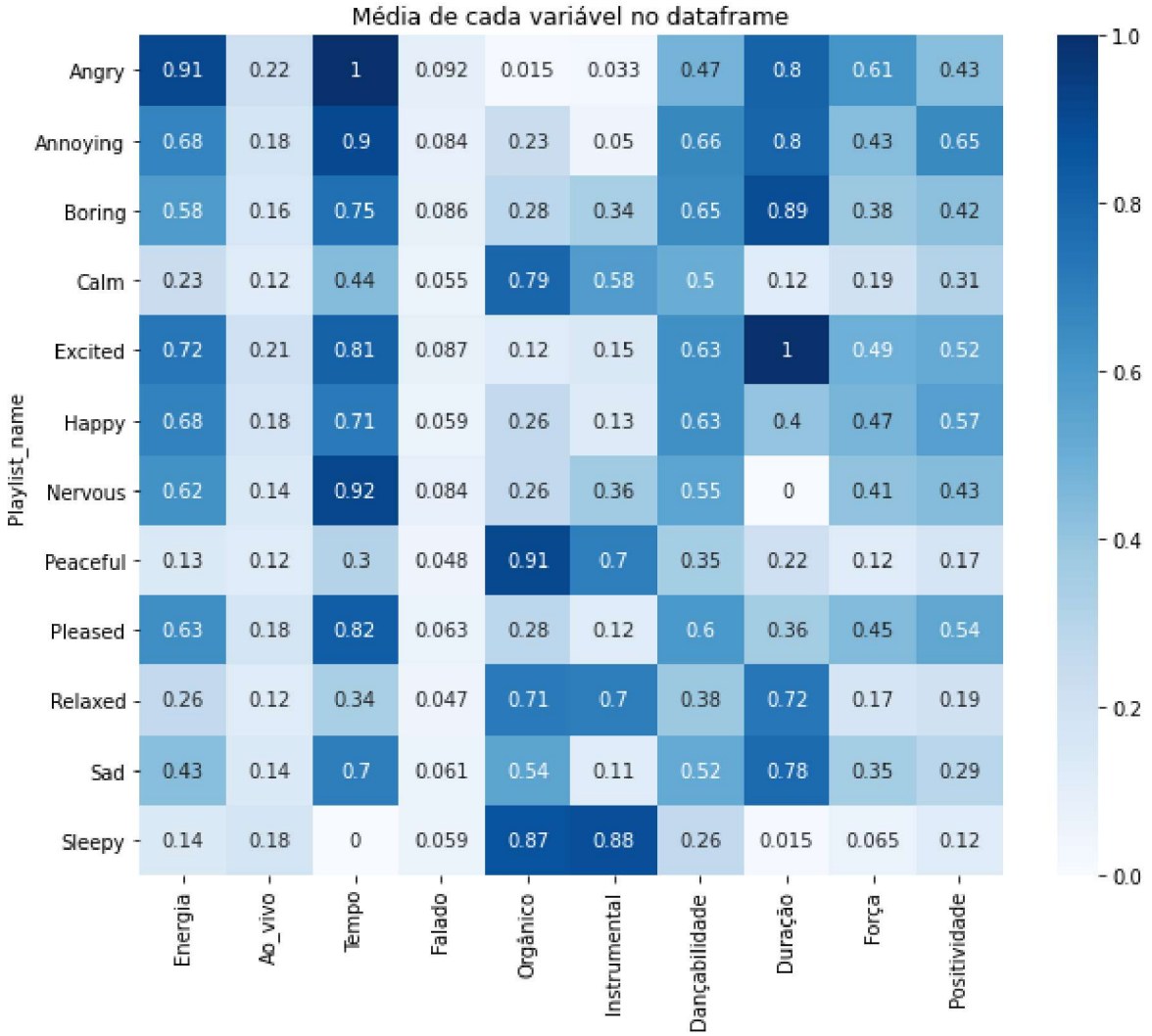
```
from sklearn.preprocessing import MinMaxScaler
```

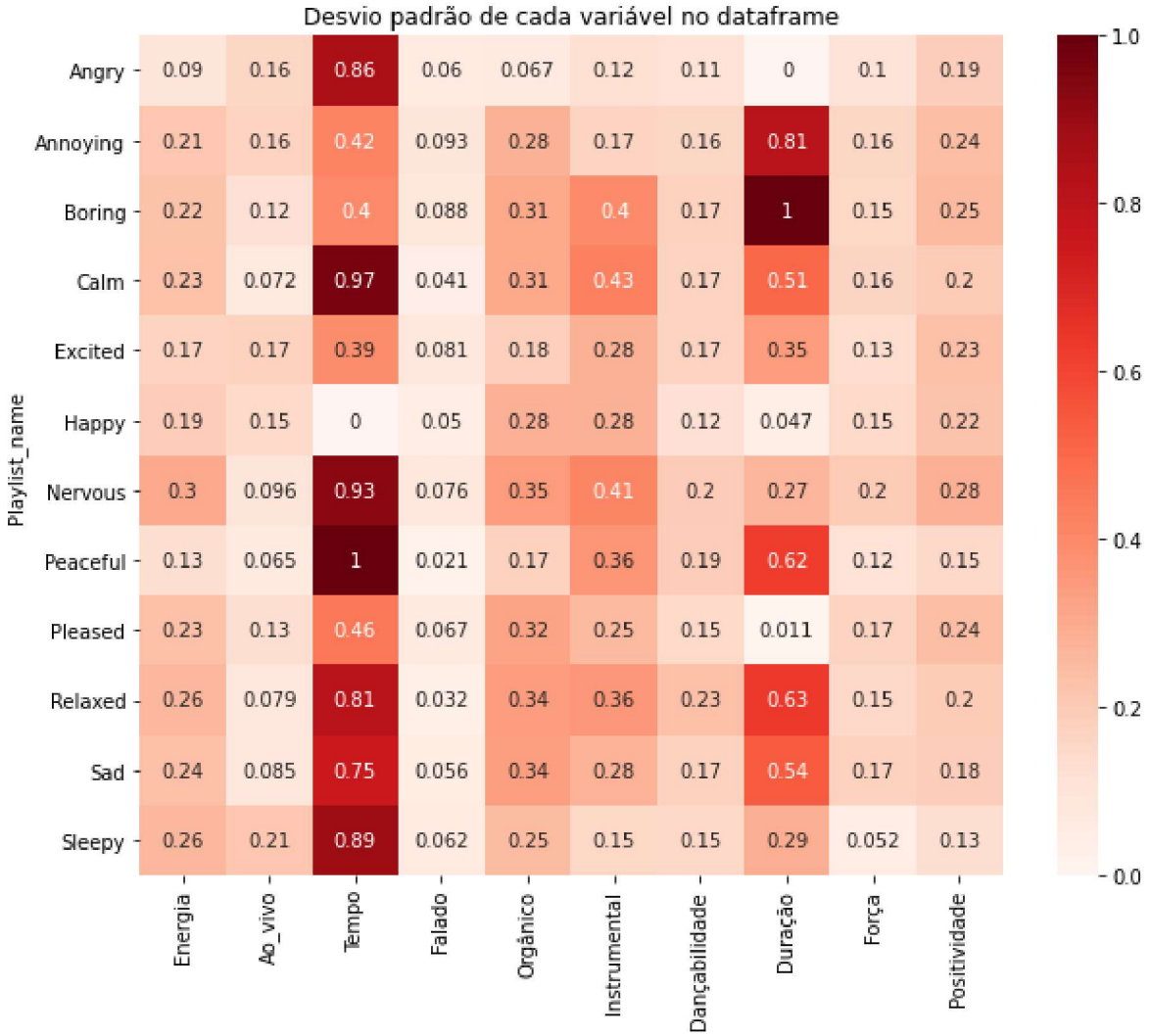
In [124]:

```
# as variáveis duração e tempo foram transormadas para ficarem na mesma ordem de grandeza d
for col in ['Duração', 'Tempo']:
    for df in [mean_df, std_df]:
        scaler = MinMaxScaler()
        df[col] = scaler.fit_transform(df[[col]]).reshape(1,-1)[0]
        del scaler
```

In [125]:

```
plt.figure(figsize=(10,8))
sns.heatmap(
    data = mean_df,
    cmap= "Blues",
    annot=True
);
plt.title('Média de cada variável no dataframe');

plt.figure(figsize=(10,8))
sns.heatmap(
    data = std_df,
    cmap= "Reds",
    annot=True
);
plt.title('Desvio padrão de cada variável no dataframe');
```





Principais insights:

- Músicas com o **Compasso** da classe '4' tendem a ter maior **Energia, Dançabilidade, Força e Positividade**;
- Músicas com **Tons** diferentes apresentam menor variação nas variáveius numéricas que músicas com **Compassos** diferentes;
- De maneira geral, músicas, **Calm, Peacefuk, Relaxed e Sleepy** têm valores de **Energia, Dançabilidade, Força e Positividade** mais baixas, enquanto **Angry, Annoying, Excited, Happy, Nervous e Pleased** têm valores mais altos.
- Músicas classificadas como **Angry** têm distribuições de *Energia* e *Orgânica* praticamente 'opostas' às músicas marcadas como **Sleepy e Peaceful**

Verificando o tipo de música do Top 10 bandas no Dataset

In [140]:

```
music_df.loc[music_df['Artista'].isin(top_10_bandas)].Playlist_name.value_counts().plot(kin
```