Experimento de Processamento de Sinais de Áudio integrado com App Blynk

Instalação das bibliotecas necessárias

In [1]:
```
! pip install hmmlearn
! pip install simplejson
! pip install pydub
! pip install eyed3

from scipy import signal

! pip install pyAudioAnalysis

from pyAudioAnalysis.audioAnalysis import fileChromagramWrapper

from pyAudioAnalysis.audioBasicIO import stereo_to_mono
from pyAudioAnalysis.audioBasicIO import read_audio_file
from pyAudioAnalysis.ShortTermFeatures import feature_extraction
```

```
Requirement already satisfied: hmmlearn in c:\programdata\anaconda3\lib\site-packages (0.2.4)
Requirement already satisfied: scipy>=0.19 in c:\programdata\anaconda3\lib\site-packages (from hmmle
arn) (1.2.1)
Requirement already satisfied: numpy>=1.10 in c:\programdata\anaconda3\lib\site-packages (from hmmle
arn) (1.18.1)
Requirement already satisfied: scikit-learn>=0.16 in c:\programdata\anaconda3\lib\site-packages (fro
m hmmlearn) (0.22.1)
Requirement already satisfied: joblib>=0.11 in c:\programdata\anaconda3\lib\site-packages (from scik
it-learn>=0.16->hmmlearn) (0.14.1)
Requirement already satisfied: simplejson in c:\programdata\anaconda3\lib\site-packages (3.17.2)
Requirement already satisfied: pydub in c:\programdata\anaconda3\lib\site-packages (0.24.1)
Requirement already satisfied: eyed3 in c:\programdata\anaconda3\lib\site-packages (0.9.6)
Requirement already satisfied: coverage[toml]<6.0.0,>=5.3.1 in c:\programdata\anaconda3\lib\site-pac
kages (from eyed3) (5.3.1)
Requirement already satisfied: filetype<2.0.0,>=1.0.7 in c:\programdata\anaconda3\lib\site-packages
(from eyed3) (1.0.7)
Requirement already satisfied: deprecation<3.0.0,>=2.1.0 in c:\programdata\anaconda3\lib\site-packag
es (from eyed3) (2.1.0)
Requirement already satisfied: toml; extra == "toml" in c:\programdata\anaconda3\lib\site-packages
(from coverage[toml]<6.0.0,>=5.3.1->eyed3) (0.10.2)
Requirement already satisfied: packaging in c:\programdata\anaconda3\lib\site-packages (from depreca
tion<3.0.0,>=2.1.0->eyed3) (20.1)
Requirement already satisfied: six in c:\programdata\anaconda3\lib\site-packages (from packaging->de
precation<3.0.0,>=2.1.0->eyed3) (1.14.0)
Requirement already satisfied: pyparsing>=2.0.2 in c:\programdata\anaconda3\lib\site-packages (from
packaging->deprecation<3.0.0,>=2.1.0->eyed3) (2.4.6)
Requirement already satisfied: pyAudioAnalysis in c:\programdata\anaconda3\lib\site-packages (0.3.6)
```

In [4]:
```
from IPython.display import Audio

!pip install sounddevice
```

```
Requirement already satisfied: sounddevice in c:\programdata\anaconda3\lib\site-packages (0.4.1)
Requirement already satisfied: CFFI>=1.0 in c:\programdata\anaconda3\lib\site-packages (from soundde
vice) (1.14.0)
Requirement already satisfied: pycparser in c:\programdata\anaconda3\lib\site-packages (from CFFI>=
1.0->sounddevice) (2.19)
```

In [5]:
```
def play_audio(file):
    return Audio(file, autoplay=True)
```

Passo 2: Processamento do Áudio de uma Música

Download de músicas em MP3 com licença Creative Commons do site: https://freemusicarchive.org/ (https://freemusicarchive.org/)

- **Genre - Author - Music Name**
- blues - Pierce Murphy - Ashes Of Paradise
- classical - Crowander - Jerry's Back

- country - Lobo Loco - Verona - Intro (ID 1407)
- electronic - Xylo Ziko - Alternate
- hiphop - Eaters - Dogstarmegalazer
- jazz - Pierce Murphy - Among The Stars
- pop - Scott Holmes Music - We Are One
- rock - Blue Wave Theory - Jazz Hole

Foi necessária conversão de MP3 para WAV para utilizar a bilbioteca pyAudioAnalysis

Escolha um dos arquivos de música que foram baixados:

In [23]:
```
file = "blues.wav"

play_audio(file)
```

Out[23]:

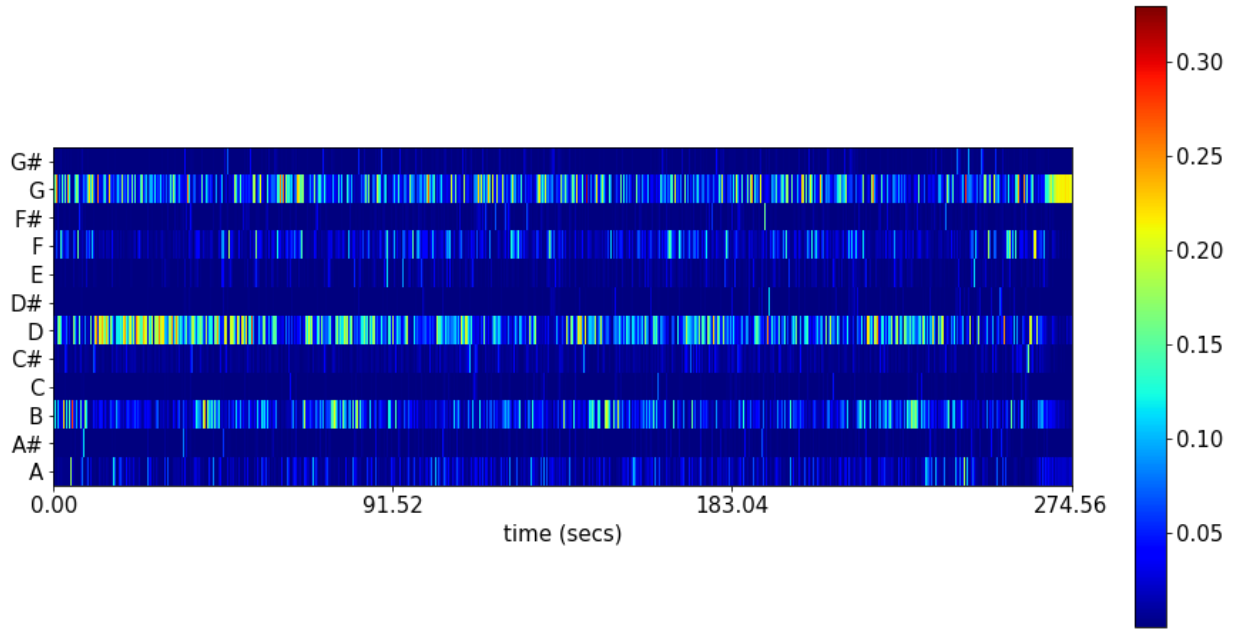
0:00 / 4:34

Processamento do Áudio e visualização do tempo necessário (em segundos):

OBS: Esta etapa levará em média de 1 a 3 minutos, dependendo da música escolhida

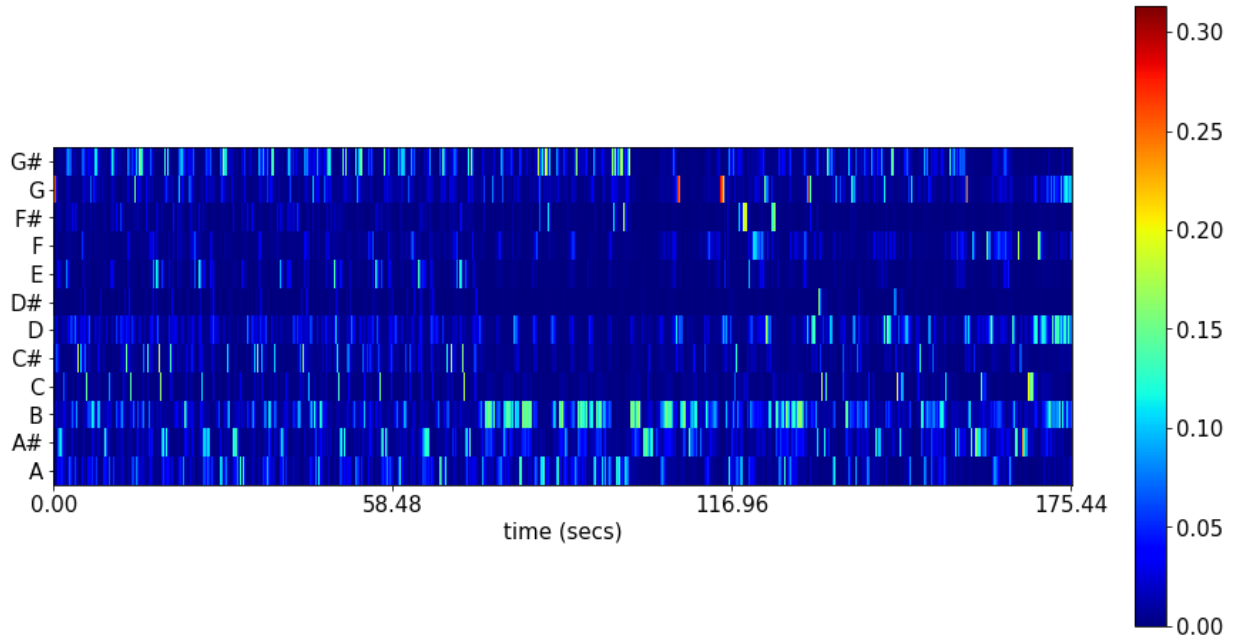
In [20]:
```
import time
import matplotlib.pyplot as plt

def chromagram(file):

    start = time.time()

    plt.rcParams["figure.figsize"] = [16,8]
    plt.rcParams.update({'font.size': 15})

    fileChromagramWrapper(file)

    end = time.time()

    print(end - start)
```

In [22]:
```
chromagram("blues.wav")
```



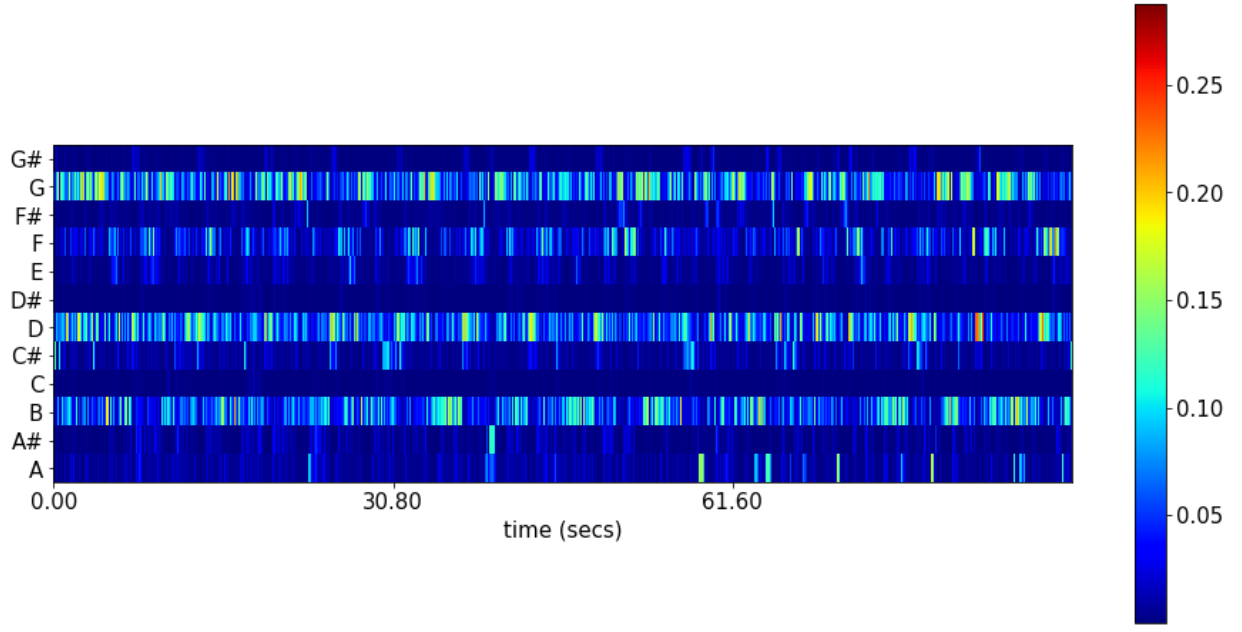

```
8.04853892326355
```

In [24]:
```
chromagram("classical.wav")
```



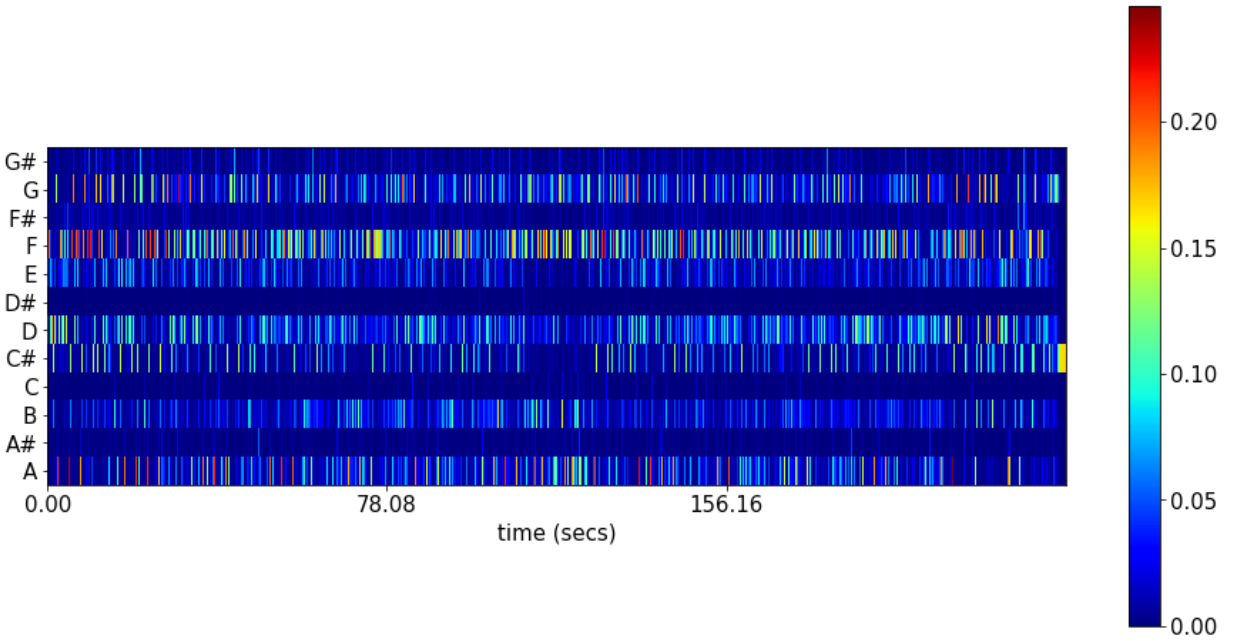

```
5.210365056991577
```

In [25]:
```
chromagram("country.wav")
```



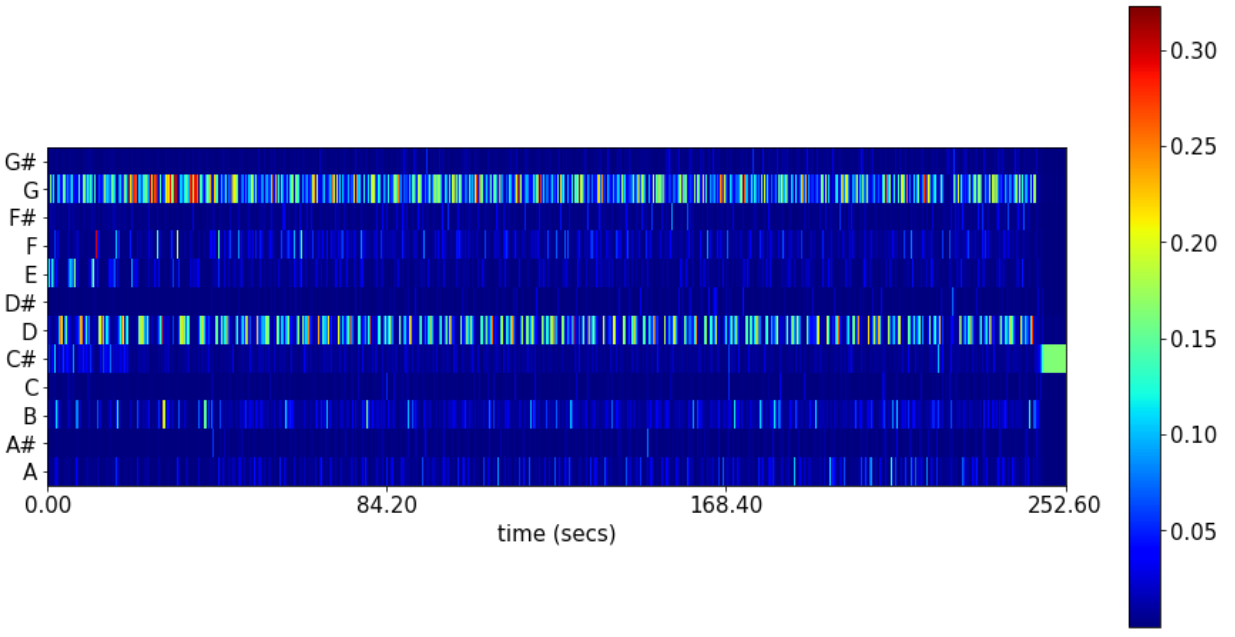

```
2.9912617206573486
```

In [26]:
```
chromagram("electronic.wav")
```



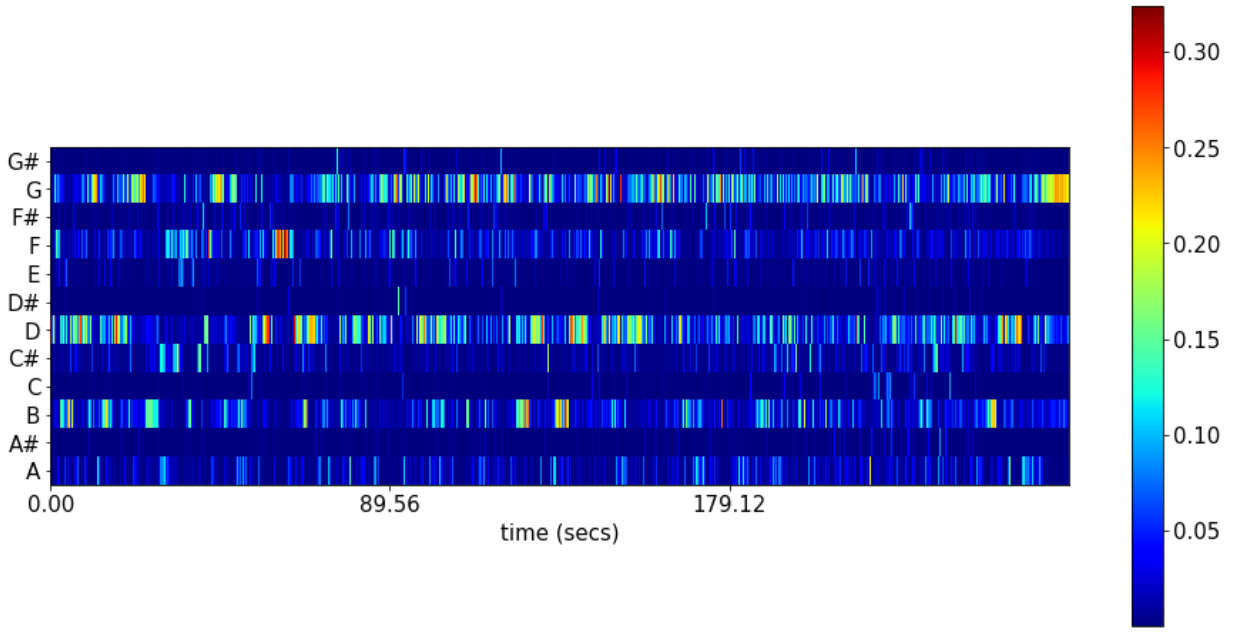

```
6.822643995285034
```

In [27]:
```
chromagram("hiphop.wav")
```



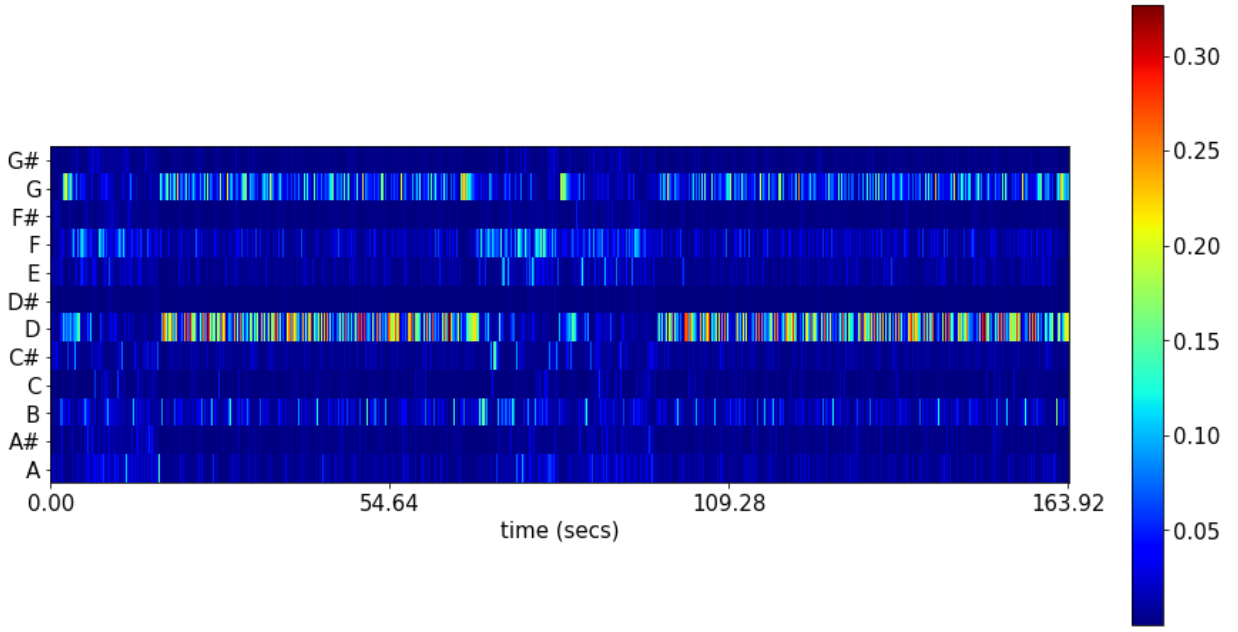

```
7.598628759384155
```

In [28]:
```
chromagram("jazz.wav")
```



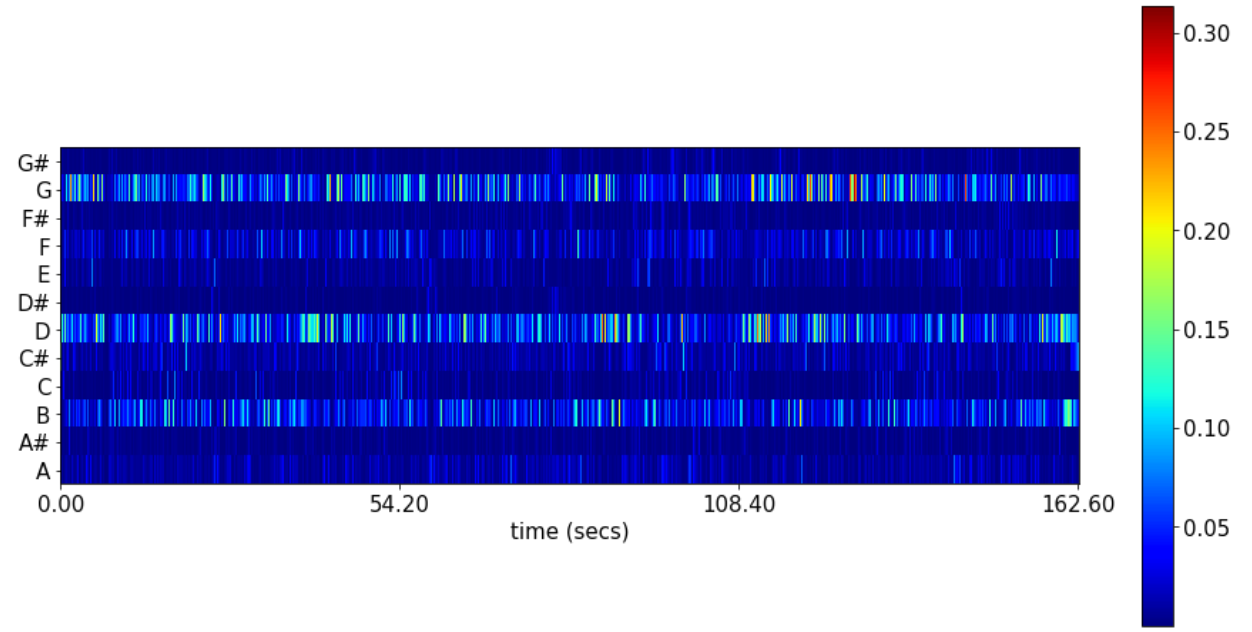

```
8.427712440490723
```

In [29]:
```
chromagram("pop.wav")
```




```
5.099345922470093
```

In [30]:
```
chromagram("rock.wav")
```




```
4.838857173919678
```

In [ ]:
```

```