1. Mount Google Drive

Lab8 - Codificação de Áudio com ffmpeg

Objetivos:

- Conhecer as opções de codificação do ffmpeg
- Verificar várias taxas mpeg 1, camada 3 e conversão entre codecs
- Variação de tempo

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
```

Arquivo .wav full

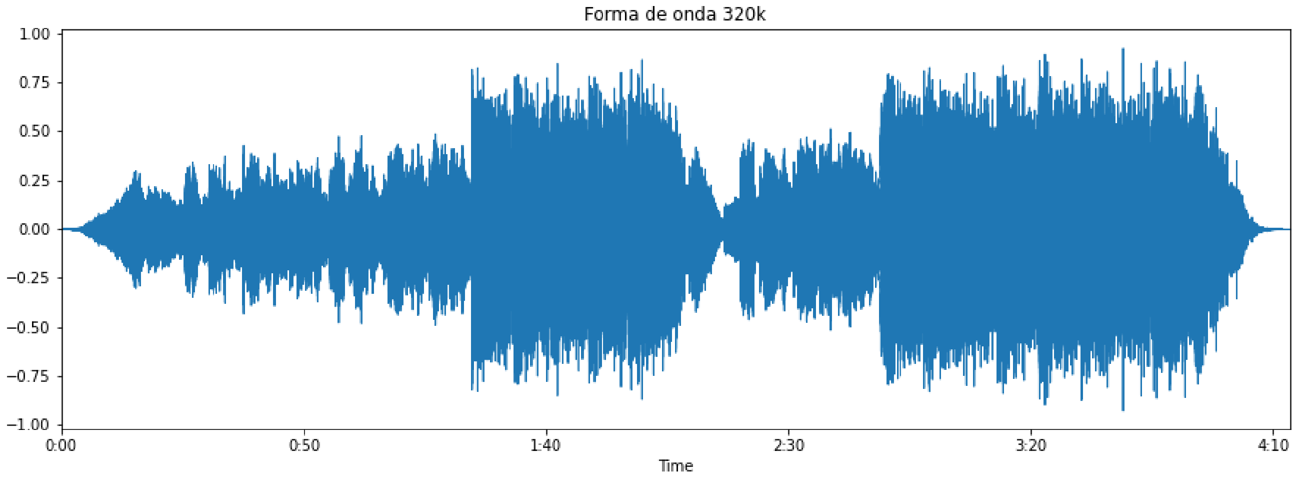
```
import numpy as np
from scipy import signal
import matplotlib.pyplot as plt
import librosa
import librosa.display
import IPython.display
import math
```

```
abba320k = '/content/drive/MyDrive/Colab Notebooks/LAB 8 - CSM/ABBA_Fernando.wav'
fer320 , sr320 = librosa.load(abba320k)
print(type(fer320), type(sr320))
print(fer320.shape, sr320)
# Player será aberto! AGUARDE até abrir!
IPython.display.Audio(data=fer320, rate=sr320)
```

```
pics = '/content/drive/MyDrive/Colab Notebooks/LAB 8 - CSM/pics'
```

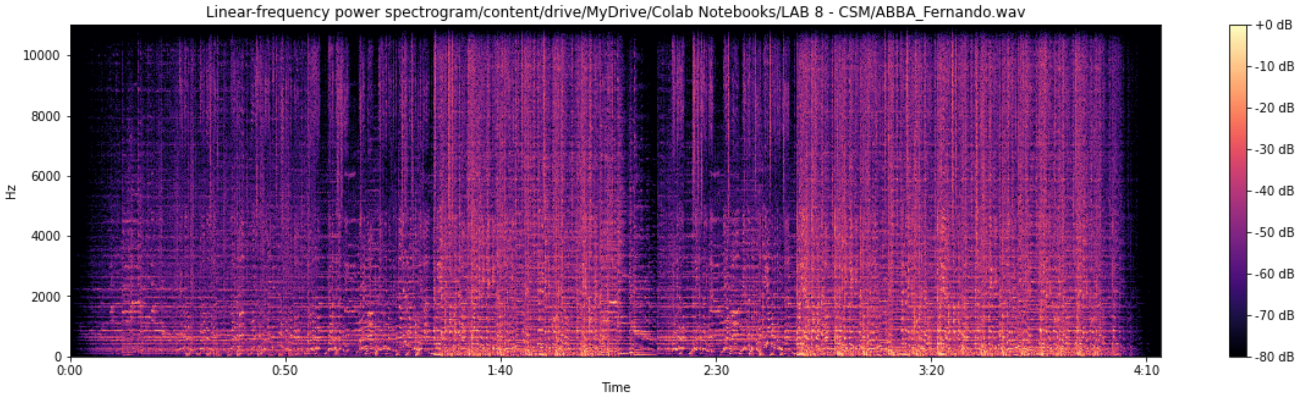
```
plt.figure()
fig, ax = plt.subplots(figsize=(15, 5))
librosa.display.waveplot(fer320, sr=sr320)
plt.title('Forma de onda 320k')
name = 'shapeABBA_320k.png'
plt.savefig(f'{pics}/{name}')
```

```
<Figure size 432x288 with 0 Axes>
```



Espectrograma Original

```
plt.figure(figsize=(20, 5))
D = librosa.amplitude_to_db(np.abs(librosa.stft(fer320)), ref=np.max)
librosa.display.specshow(D, x_axis='time',y_axis='linear')
plt.colorbar(format='%+2.0f dB')
plt.title('Linear-frequency power spectrogram'+ abba320k)
name = 'EspectroABBA_320k.png'
plt.savefig(f'{pics}/{name}')
```



Verificar os codecs

Opções do ffmpeg

Conversão wav em mp3, 128kbps

```
!ffmpeg -h
```

```
!ffmpeg -codecs
```

```
 DEA..S s302m                SMPTE 302M
 D.A.L. sdx2_dpcm            DPCM Squareroot-Delta-Exact
 D.A..S shorten              Shorten
 D.A.L. sipr                 RealAudio SIPR / ACELP.NET
 D.A.L. smackaudio           Smacker audio (decoders: smackaud )
 ..A.L. smv                  SMV (Selectable Mode Vocoder)
 D.A.L. sol_dpcm             DPCM Sol
 DEA... sonic                Sonic
 .EA... sonicls              Sonic lossless
 DEA.L. speex                Speex (decoders: libspeex ) (encoders: libspeex )
 D.AI.S tak                  TAK (Tom's lossless Audio Kompressor)
 DEA..S truehd               TrueHD
 D.A.L. truespeech           DSP Group TrueSpeech
 DEAI.S tta                  TTA (True Audio)
 D.A.L. twinvq               VQF TwinVQ
 D.A.L. vmdaudio             Sierra VMD audio
 DEA.L. vorbis               Vorbis (decoders: vorbis libvorbis ) (encoders: v
 ..A.L. voxware              Voxware RT29 Metasound
 D.A... wavesynth            Wave synthesis pseudo-codec
 DEAILS wavpack              WavPack (encoders: wavpack libwavpack )
 D.A.L. westwood_snd1        Westwood Audio (SND1) (decoders: ws_snd1 )
 D.A..S wmalossless          Windows Media Audio Lossless
 D.A.L. wmapro               Windows Media Audio 9 Professional
 DEA.L. wmav1                Windows Media Audio 1
 DEA.L. wmav2                Windows Media Audio 2
 D.A.L. wmavoice             Windows Media Audio Voice
 D.A.L. xan_dpcm             DPCM Xan
 D.A.L. xma1                 Xbox Media Audio 1
 D.A.L. xma2                 Xbox Media Audio 2
 ..D... bin_data             binary data
 ..D... dvd_nav_packet       DVD Nav packet
 ..D... klv                  SMPTE 336M Key-Length-Value (KLV) metadata
 ..D... otf                  OpenType font
 ..D... scte_35              SCTE 35 Message Queue
 ..D... timed_id3            timed ID3 metadata
 ..D... ttf                  TrueType font
 DES... ass                  ASS (Advanced SSA) subtitle (decoders: ssa ass )
 DES... dvb_subtitle         DVB subtitles (decoders: dvbsub ) (encoders: dvbs
 D.S... dvb_teletext         DVB teletext (decoders: libzvbi_teletextdec )
 DES... dvd_subtitle         DVD subtitles (decoders: dvdsub ) (encoders: dvds
 D.S... eia_608              EIA-608 closed captions (decoders: cc_dec )
 D.S... hdmv_pgs_subtitle    HDMV Presentation Graphic Stream subtitles (decod
 ..S... hdmv_text_subtitle   HDMV Text subtitle
 D.S... jacosub              JACOsub subtitle
 D.S... microdvd             MicroDVD subtitle
 DES... mov_text             MOV text
 D.S... mpl2                 MPL2 subtitle
 D.S... pjs                  PJS (Phoenix Japanimation Society) subtitle
 D.S... realtext             RealText subtitle
 D.S... sami                 SAMI subtitle
 ..S... srt                  SubRip subtitle with embedded timing
 ..S... ssa                  SSA (SubStation Alpha) subtitle
 D.S... stl                  Spruce subtitle format
 DES... subrip               SubRip subtitle (decoders: srt subrip ) (encoders
 D.S... subviewer            SubViewer subtitle
 D.S... subviewer1           SubViewer v1 subtitle
 DES... text                 raw UTF-8 text
 D.S... vplayer              VPlayer subtitle
```

```
!ffmpeg -i '/content/drive/MyDrive/Colab Notebooks/LAB 8 - CSM/ABBA_Fernando.wav' -
```

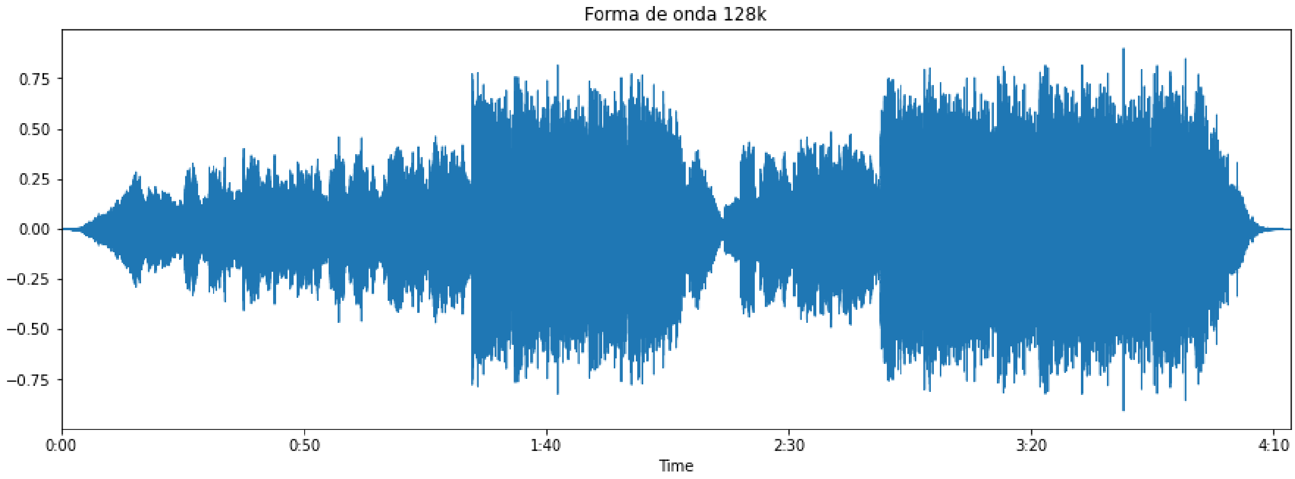
```
ffmpeg version 3.4.8-0ubuntu0.2 Copyright (c) 2000-2020 the FFmpeg developers
  built with gcc 7 (Ubuntu 7.5.0-3ubuntu1~18.04)
  configuration: --prefix=/usr --extra-version=0ubuntu0.2 --toolchain=hardened
  libavutil      55. 78.100 / 55. 78.100
  libavcodec     57.107.100 / 57.107.100
  libavformat    57. 83.100 / 57. 83.100
  libavdevice    57. 10.100 / 57. 10.100
  libavfilter     6.107.100 /  6.107.100
  libavresample   3.  7.  0 /  3.  7.  0
  libswscale      4.  8.100 /  4.  8.100
  libswresample   2.  9.100 /  2.  9.100
  libpostproc    54.  7.100 / 54.  7.100
Guessed Channel Layout for Input Stream #0.0 : stereo
Input #0, wav, from '/content/drive/MyDrive/Colab Notebooks/LAB 8 - CSM/ABBA_F
  Metadata:
    title           : Fernando
    album           : Gold: Greatest Hits
    artist          : ABBA
    date            : 1993
    genre           : Dance
    track           : 12
  Duration: 00:04:13.45, bitrate: 1411 kb/s
    Stream #0:0: Audio: pcm_s16le ([1][0][0][0] / 0x0001), 44100 Hz, stereo, s
Stream mapping:
  Stream #0:0 -> #0:0 (pcm_s16le (native) -> mp3 (libmp3lame))
Press [q] to stop, [?] for help
Output #0, mp3, to 'abba128k.mp3':
  Metadata:
    TIT2            : Fernando
    TALB            : Gold: Greatest Hits
    TPE1            : ABBA
    TDRC            : 1993
    TCON            : Dance
    TRCK            : 12
    TSSE            : Lavf57.83.100
    Stream #0:0: Audio: mp3 (libmp3lame), 44100 Hz, stereo, s16p, 128 kb/s
    Metadata:
      encoder         : Lavc57.107.100 libmp3lame
size=    3961kB time=00:04:13.46 bitrate= 128.0kbits/s speed=36.6x
video:0kB audio:3961kB subtitle:0kB other streams:0kB global headers:0kB muxin
```

Áudio e Espectrograma do arquivo codificado

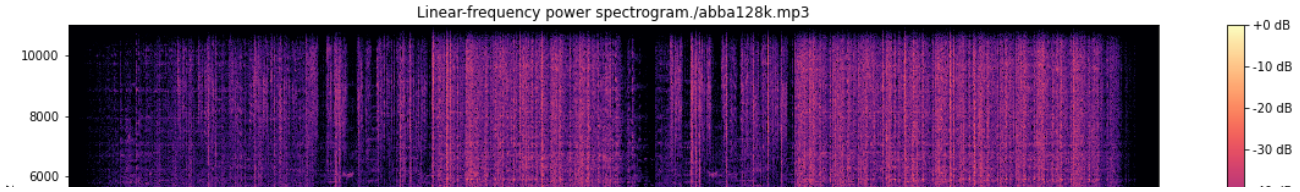
```
abba128k = './abba128k.mp3'
print(abba128k)
fer128 , sr128 = librosa.load(abba128k)
print(type(fer128), type(sr128))
print(fer128.shape, sr128)
# Player será aberto! AGUARDE até abrir!
IPython.display.Audio(data=fer128, rate=sr128)
```

```
plt.figure()
fig, ax = plt.subplots(figsize=(15, 5))
librosa.display.waveplot(fer128, sr=sr128)
plt.title('Forma de onda 128k')
name = 'shapeABBA_128k.png'
plt.savefig(f'{pics}/{name}')
```

```
    <Figure size 432x288 with 0 Axes>
```



```
plt.figure(figsize=(20, 5))
D = librosa.amplitude_to_db(np.abs(librosa.stft(fer128)), ref=np.max)
librosa.display.specshow(D, x_axis='time',y_axis='linear')
plt.colorbar(format='%+2.0f dB')
plt.title('Linear-frequency power spectrogram'+ abba128k)
name = 'EspectroABBA_128k.png'
plt.savefig(f'{pics}/{name}')
```



Espectrograma do Arquivo codificado a 128kbps

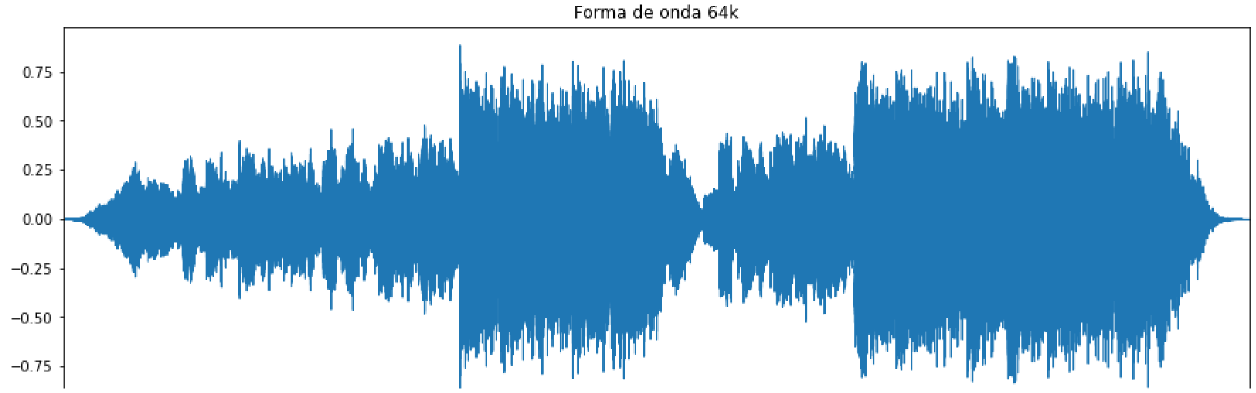
Codificando a 64kbps

```
!ffmpeg -i '/content/drive/MyDrive/Colab Notebooks/LAB 8 - CSM/ABBA_Fernando.wav' -
```

```
ffmpeg version 3.4.8-0ubuntu0.2 Copyright (c) 2000-2020 the FFmpeg developers
  built with gcc 7 (Ubuntu 7.5.0-3ubuntu1~18.04)
  configuration: --prefix=/usr --extra-version=0ubuntu0.2 --toolchain=hardenec
  libavutil      55. 78.100 / 55. 78.100
  libavcodec     57.107.100 / 57.107.100
  libavformat    57. 83.100 / 57. 83.100
  libavdevice    57. 10.100 / 57. 10.100
  libavfilter     6.107.100 /  6.107.100
  libavresample   3.  7.  0 /  3.  7.  0
  libswscale      4.  8.100 /  4.  8.100
  libswresample   2.  9.100 /  2.  9.100
  libpostproc    54.  7.100 / 54.  7.100
Guessed Channel Layout for Input Stream #0.0 : stereo
Input #0, wav, from '/content/drive/MyDrive/Colab Notebooks/LAB 8 - CSM/ABBA_F
  Metadata:
    title           : Fernando
    album           : Gold: Greatest Hits
    artist          : ABBA
    date            : 1993
    genre           : Dance
    track           : 12
  Duration: 00:04:13.45, bitrate: 1411 kb/s
    Stream #0:0: Audio: pcm_s16le ([1][0][0][0] / 0x0001), 44100 Hz, stereo, s
File 'abba64k.mp3' already exists. Overwrite ? [y/N] ^C
```

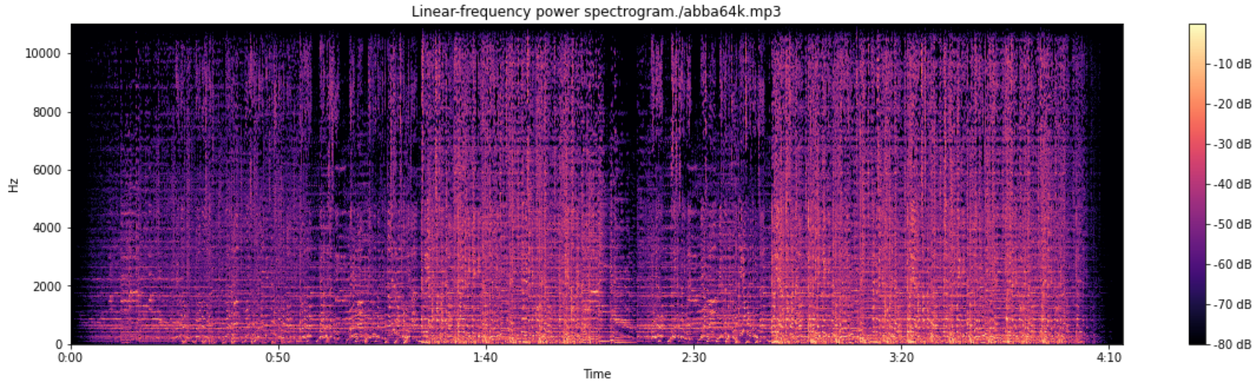
```
abba64k = './abba64k.mp3'
print(abba64k)
fer64 , sr64 = librosa.load(abba64k)
plt.figure()
fig, ax = plt.subplots(figsize=(15, 5))
librosa.display.waveplot(fer64, sr=sr64)
plt.title('Forma de onda 64k')
name = 'shapeABBA_64k.png'
plt.savefig(f'{pics}/{name}')
```

```
./abba64k.mp3
/usr/local/lib/python3.7/dist-packages/librosa/core/audio.py:165: UserWarni
  warnings.warn("PySoundFile failed. Trying audioread instead.")
<Figure size 432x288 with 0 Axes>
```



Espectrograma do arquivo a 64kbps

```
plt.figure(figsize=(20, 5))
D = librosa.amplitude_to_db(np.abs(librosa.stft(fer64)), ref=np.max)
librosa.display.specshow(D, x_axis='time',y_axis='linear')
plt.colorbar(format='%+2.0f dB')
plt.title('Linear-frequency power spectrogram'+ abba64k)
name = 'EspectroABBA_64k.png'
plt.savefig(f'{pics}/{name}')
```



Audio do arquivo

```
IPython.display.Audio(data=fer64, rate=sr64)
```

Conversão de Codecs

```
!ffmpeg -i abba128k.mp3 -c:a libvorbis abba128k.ogg
```

```
ffmpeg version 3.4.8-0ubuntu0.2 Copyright (c) 2000-2020 the FFmpeg developers
  built with gcc 7 (Ubuntu 7.5.0-3ubuntu1~18.04)
  configuration: --prefix=/usr --extra-version=0ubuntu0.2 --toolchain=hardenec
  libavutil      55. 78.100 / 55. 78.100
  libavcodec     57.107.100 / 57.107.100
  libavformat    57. 83.100 / 57. 83.100
  libavdevice    57. 10.100 / 57. 10.100
  libavfilter     6.107.100 /  6.107.100
  libavresample   3.  7.  0 /  3.  7.  0
  libswscale      4.  8.100 /  4.  8.100
  libswresample   2.  9.100 /  2.  9.100
  libpostproc    54.  7.100 / 54.  7.100
Input #0, mp3, from 'abba128k.mp3':
  Metadata:
    title           : Fernando
    album           : Gold: Greatest Hits
    artist          : ABBA
    date            : 1993
    genre           : Dance
    track           : 12
    encoder         : Lavf57.83.100
  Duration: 00:04:13.49, start: 0.025057, bitrate: 128 kb/s
    Stream #0:0: Audio: mp3, 44100 Hz, stereo, s16p, 128 kb/s
    Metadata:
      encoder         : Lavc57.10
Stream mapping:
  Stream #0:0 -> #0:0 (mp3 (native) -> vorbis (libvorbis))
Press [q] to stop, [?] for help
Output #0, ogg, to 'abba128k.ogg':
  Metadata:
    title           : Fernando
    album           : Gold: Greatest Hits
    artist          : ABBA
    date            : 1993
    genre           : Dance
    track           : 12
    encoder         : Lavf57.83.100
    Stream #0:0: Audio: vorbis (libvorbis), 44100 Hz, stereo, fltp
    Metadata:
      encoder         : Lavc57.107.100 libvorbis
      title           : Fernando
      album           : Gold: Greatest Hits
      artist          : ABBA
      date            : 1993
      genre           : Dance
      TRACKNUMBER     : 12
size=    3267kB time=00:04:13.43 bitrate= 105.6kbits/s speed=43.1x
video:0kB audio:3234kB subtitle:0kB other streams:0kB global headers:4kB muxir
```

```
!ls -l
```

```
total 20124
-rw-r--r-- 1 root root 4056451 Dec 18 20:00 abba128k.mp3
-rw-r--r-- 1 root root 3345162 Dec 18 21:47 abba128k.ogg
-rw-r--r-- 1 root root 8110864 Dec 18 20:50 abba128k_vh.mp3
-rw-r--r-- 1 root root 1014336 Dec 18 20:43 abba16k.mp3
```

```
-rw-r--r-- 1 root root 1014336 Dec 18 20:42 abba32k.mp3
-rw-r--r-- 1 root root 2028304 Dec 18 20:11 abba64k.mp3
-rw-r--r-- 1 root root 1014336 Dec 18 20:47 abba8k.mp3
drwx------ 5 root root    4096 Dec 18 19:44 drive
drwxr-xr-x 1 root root    4096 Dec  3 14:33 sample_data
```

Faça o procedimento para mais três taxas, 32k, 16k e 8kbps.

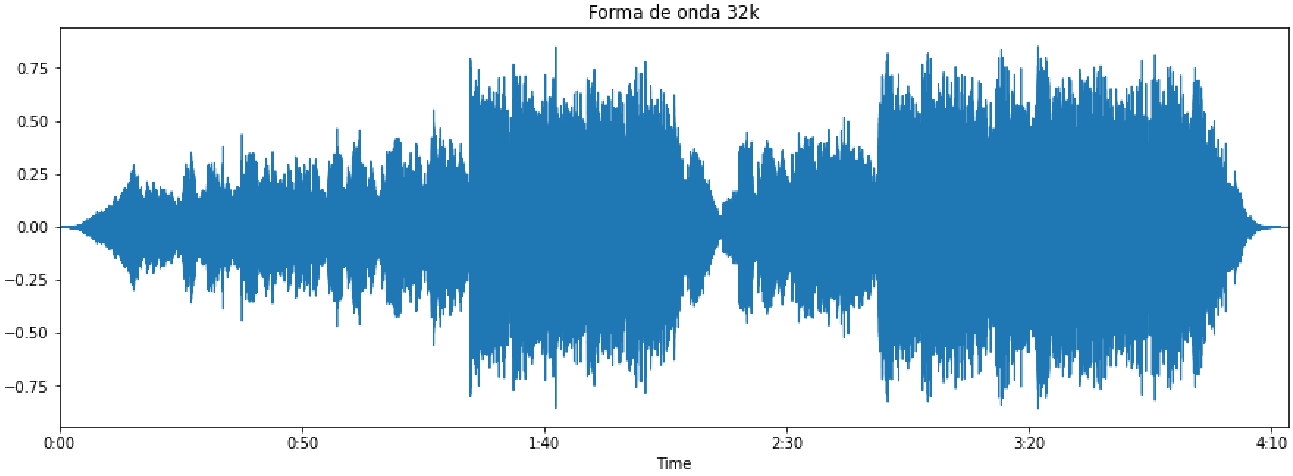
```
!ffmpeg -i '/content/drive/MyDrive/Colab Notebooks/LAB 8 - CSM/ABBA_Fernando.wav' -
```

```
ffmpeg version 3.4.8-0ubuntu0.2 Copyright (c) 2000-2020 the FFmpeg developers
  built with gcc 7 (Ubuntu 7.5.0-3ubuntu1~18.04)
  configuration: --prefix=/usr --extra-version=0ubuntu0.2 --toolchain=hardened
  libavutil      55. 78.100 / 55. 78.100
  libavcodec     57.107.100 / 57.107.100
  libavformat    57. 83.100 / 57. 83.100
  libavdevice    57. 10.100 / 57. 10.100
  libavfilter     6.107.100 /  6.107.100
  libavresample   3.  7.  0 /  3.  7.  0
  libswscale      4.  8.100 /  4.  8.100
  libswresample   2.  9.100 /  2.  9.100
  libpostproc    54.  7.100 / 54.  7.100
Guessed Channel Layout for Input Stream #0.0 : stereo
Input #0, wav, from '/content/drive/MyDrive/Colab Notebooks/LAB 8 - CSM/ABBA_F
  Metadata:
    title           : Fernando
    album           : Gold: Greatest Hits
    artist          : ABBA
    date            : 1993
    genre           : Dance
    track           : 12
  Duration: 00:04:13.45, bitrate: 1411 kb/s
    Stream #0:0: Audio: pcm_s16le ([1][0][0][0] / 0x0001), 44100 Hz, stereo, s
File 'abba32k.mp3' already exists. Overwrite ? [y/N] y
Stream mapping:
  Stream #0:0 -> #0:0 (pcm_s16le (native) -> mp3 (libmp3lame))
Press [q] to stop, [?] for help
Output #0, mp3, to 'abba32k.mp3':
  Metadata:
    TIT2            : Fernando
    TALB            : Gold: Greatest Hits
    TPE1            : ABBA
    TDRC            : 1993
    TCON            : Dance
    TRCK            : 12
    TSSE            : Lavf57.83.100
    Stream #0:0: Audio: mp3 (libmp3lame), 44100 Hz, stereo, s16p, 32 kb/s
    Metadata:
      encoder         : Lavc57.107.100 libmp3lame
size=     991kB time=00:04:13.46 bitrate=  32.0kbits/s speed=50.1x
video:0kB audio:990kB subtitle:0kB other streams:0kB global headers:0kB muxing
```

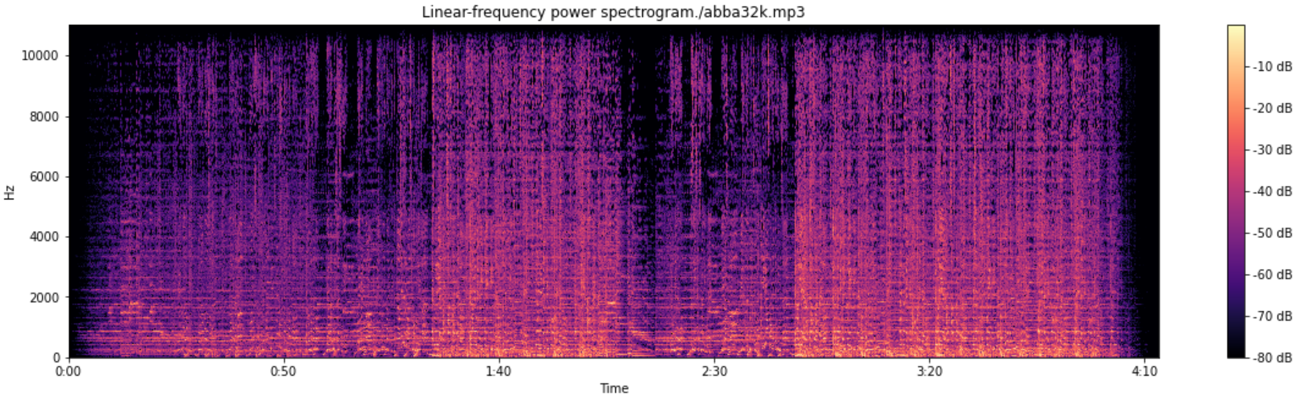
```
abba32k = './abba32k.mp3'
print(abba32k)
fer32 , sr32 = librosa.load(abba32k)
plt.figure()
```

```
fig, ax = plt.subplots(figsize=(15, 5))
librosa.display.waveplot(fer32, sr=sr32)
plt.title('Forma de onda 32k')
name = 'shapeABBA_32k.png'
plt.savefig(f'{pics}/{name}')
```

```
./abba32k.mp3
/usr/local/lib/python3.7/dist-packages/librosa/core/audio.py:165: UserWarning:
  warnings.warn("PySoundFile failed. Trying audioread instead.")
<Figure size 432x288 with 0 Axes>
```



```
plt.figure(figsize=(20, 5))
D = librosa.amplitude_to_db(np.abs(librosa.stft(fer64)), ref=np.max)
librosa.display.specshow(D, x_axis='time',y_axis='linear')
plt.colorbar(format='%+2.0f dB')
plt.title('Linear-frequency power spectrogram'+ abba32k)
name = 'EspectroABBA_32k.png'
plt.savefig(f'{pics}/{name}')
```



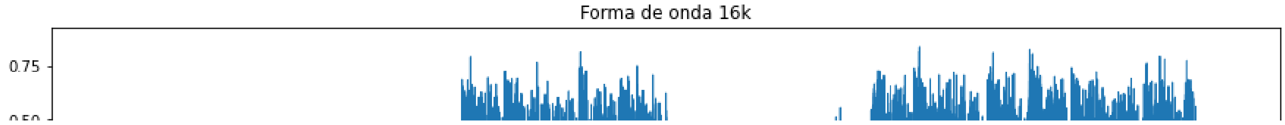
```
IPython.display.Audio(data=fer32, rate=sr32)
```

```
!ffmpeg -i '/content/drive/MyDrive/Colab Notebooks/LAB 8 - CSM/ABBA_Fernando.wav' -
```

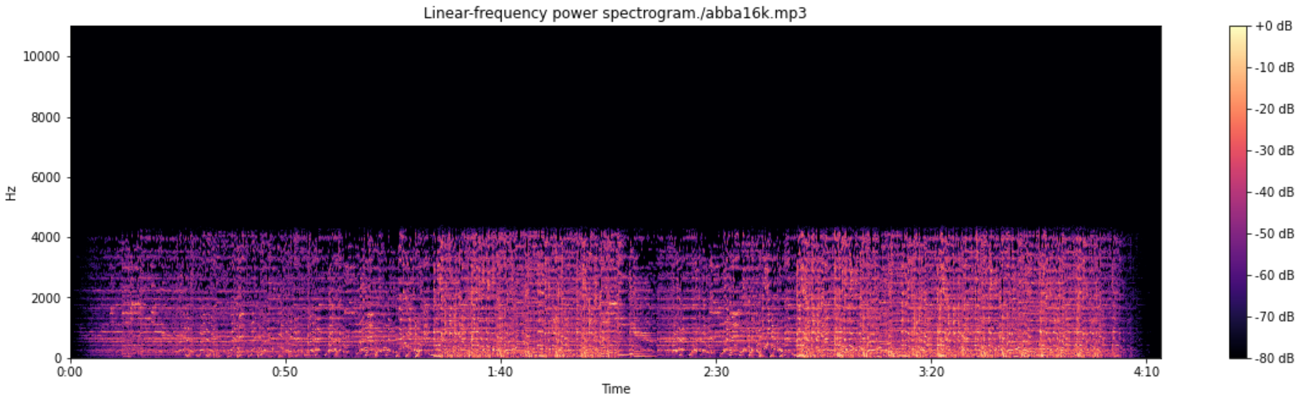
```
ffmpeg version 3.4.8-0ubuntu0.2 Copyright (c) 2000-2020 the FFmpeg developers
  built with gcc 7 (Ubuntu 7.5.0-3ubuntu1~18.04)
  configuration: --prefix=/usr --extra-version=0ubuntu0.2 --toolchain=hardenec
  libavutil      55. 78.100 / 55. 78.100
  libavcodec     57.107.100 / 57.107.100
  libavformat    57. 83.100 / 57. 83.100
  libavdevice    57. 10.100 / 57. 10.100
  libavfilter     6.107.100 /  6.107.100
  libavresample   3.  7.  0 /  3.  7.  0
  libswscale      4.  8.100 /  4.  8.100
  libswresample   2.  9.100 /  2.  9.100
  libpostproc    54.  7.100 / 54.  7.100
Guessed Channel Layout for Input Stream #0.0 : stereo
Input #0, wav, from '/content/drive/MyDrive/Colab Notebooks/LAB 8 - CSM/ABBA_F
  Metadata:
    title           : Fernando
    album           : Gold: Greatest Hits
    artist          : ABBA
    date            : 1993
    genre           : Dance
    track           : 12
  Duration: 00:04:13.45, bitrate: 1411 kb/s
    Stream #0:0: Audio: pcm_s16le ([1][0][0][0] / 0x0001), 44100 Hz, stereo, s
File 'abba16k.mp3' already exists. Overwrite ? [y/N] ^C
```

```
abba16k = './abba16k.mp3'
print(abba16k)
fer16 , sr16 = librosa.load(abba16k)
plt.figure()
fig, ax = plt.subplots(figsize=(15, 5))
librosa.display.waveplot(fer16, sr=sr16)
plt.title('Forma de onda 16k')
name = 'shapeABBA_16k.png'
plt.savefig(f'{pics}/{name}')
```

```
./abba16k.mp3
/usr/local/lib/python3.7/dist-packages/librosa/core/audio.py:165: UserWarning:
  warnings.warn("PySoundFile failed. Trying audioread instead.")
<Figure size 432x288 with 0 Axes>
```



```
plt.figure(figsize=(20, 5))
D = librosa.amplitude_to_db(np.abs(librosa.stft(fer16)), ref=np.max)
librosa.display.specshow(D, x_axis='time',y_axis='linear')
plt.colorbar(format='%+2.0f dB')
plt.title('Linear-frequency power spectrogram'+ abba16k)
name = 'EspectroABBA_16k.png'
plt.savefig(f'{pics}/{name}')
```



```
IPython.display.Audio(data=fer16, rate=sr16)
```

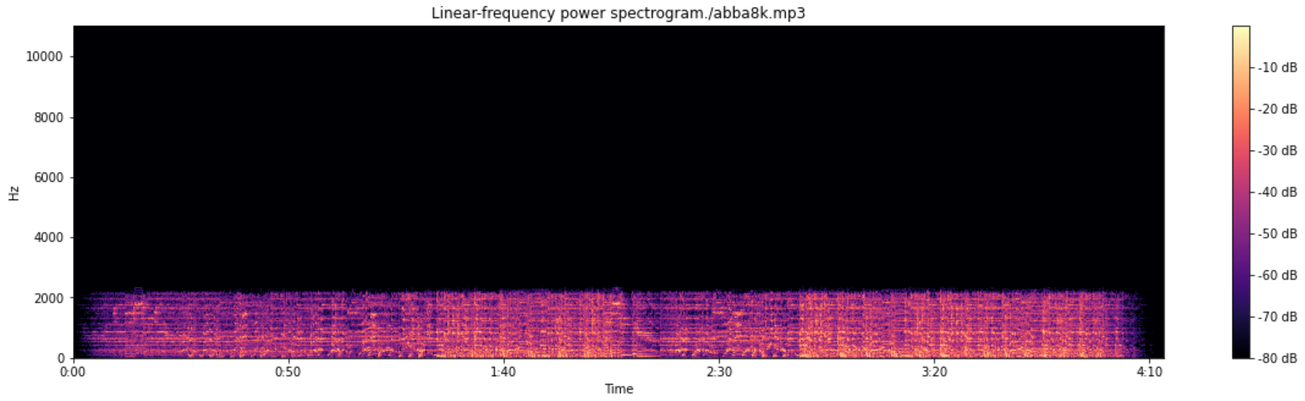
```
!ffmpeg -i '/content/drive/MyDrive/Colab Notebooks/LAB 8 - CSM/ABBA_Fernando.wav' -
```

```
ffmpeg version 3.4.8-0ubuntu0.2 Copyright (c) 2000-2020 the FFmpeg developers
  built with gcc 7 (Ubuntu 7.5.0-3ubuntu1~18.04)
  configuration: --prefix=/usr --extra-version=0ubuntu0.2 --toolchain=hardened
  libavutil      55. 78.100 / 55. 78.100
  libavcodec     57.107.100 / 57.107.100
  libavformat    57. 83.100 / 57. 83.100
  libavdevice    57. 10.100 / 57. 10.100
  libavfilter     6.107.100 /  6.107.100
  libavresample   3.  7.  0 /  3.  7.  0
  libswscale      4.  8.100 /  4.  8.100
  libswresample   2.  9.100 /  2.  9.100
  libpostproc    54.  7.100 / 54.  7.100
Guessed Channel Layout for Input Stream #0.0 : stereo
Input #0, wav, from '/content/drive/MyDrive/Colab Notebooks/LAB 8 - CSM/ABBA_F
  Metadata:
    title           : Fernando
    album           : Gold: Greatest Hits
    artist          : ABBA
    date            : 1993
    genre           : Dance
    track           : 12
  Duration: 00:04:13.45, bitrate: 1411 kb/s
    Stream #0:0: Audio: pcm_s16le ([1][0][0][0] / 0x0001), 44100 Hz, stereo, s
Stream mapping:
  Stream #0:0 -> #0:0 (pcm_s16le (native) -> mp3 (libmp3lame))
Press [q] to stop, [?] for help
Output #0, mp3, to 'abba8k.mp3':
  Metadata:
    TIT2            : Fernando
    TALB            : Gold: Greatest Hits
    TPE1            : ABBA
    TDRC            : 1993
    TCON            : Dance
    TRCK            : 12
    TSSE            : Lavf57.83.100
    Stream #0:0: Audio: mp3 (libmp3lame), 44100 Hz, stereo, s16p, 8 kb/s
    Metadata:
      encoder         : Lavc57.107.100 libmp3lame
size=     991kB time=00:04:13.46 bitrate=  32.0kbits/s speed=47.1x
video:0kB audio:990kB subtitle:0kB other streams:0kB global headers:0kB muxing
```

```
abba8k = './abba8k.mp3'
print(abba8k)
fer8 , sr8 = librosa.load(abba8k)
plt.figure()
fig, ax = plt.subplots(figsize=(15, 5))
librosa.display.waveplot(fer8, sr=sr8)
plt.title('Forma de onda 8k')
name = 'shapeABBA_8k.png'
plt.savefig(f'{pics}/{name}')
```

```
    ./abba8k.mp3
    /usr/local/lib/python3.7/dist-packages/librosa/core/audio.py:165: UserWarning:
```

```
plt.figure(figsize=(20, 5))
D = librosa.amplitude_to_db(np.abs(librosa.stft(fer8)), ref=np.max)
librosa.display.specshow(D, x_axis='time',y_axis='linear')
plt.colorbar(format='%+2.0f dB')
plt.title('Linear-frequency power spectrogram'+ abba8k)
name = 'EspectroABBA_8k.png'
plt.savefig(f'{pics}/{name}')
```



```
IPython.display.Audio(data=fer8, rate=sr8)
```

Alterando a velocidade

```
!ffmpeg -i abba128k.mp3 -filter:a "atempo=2.0" -vn abba128k_v2.mp3
```

Audio v2

```
abba128k_2 = './abba128k_v2.mp3'
print(abba128k_2)
fer128_2 , sr128_2 = librosa.load(abba128k_2)
IPython.display.Audio(data=fer128_2, rate=sr128_2)
```

Mude para um tempo mais lento

```
!ffmpeg -i abba128k.mp3 -filter:a "atempo=0.5" -vn abba128k_vh.mp3
```

```
    ffmpeg version 3.4.8-0ubuntu0.2 Copyright (c) 2000-2020 the FFmpeg developers
      built with gcc 7 (Ubuntu 7.5.0-3ubuntu1~18.04)
```

```
  configuration: --prefix=/usr --extra-version=0ubuntu0.2 --toolchain=hardenec
  libavutil      55. 78.100 / 55. 78.100
  libavcodec     57.107.100 / 57.107.100
  libavformat    57. 83.100 / 57. 83.100
  libavdevice    57. 10.100 / 57. 10.100
  libavfilter     6.107.100 /  6.107.100
  libavresample   3.  7.  0 /  3.  7.  0
  libswscale      4.  8.100 /  4.  8.100
  libswresample   2.  9.100 /  2.  9.100
  libpostproc    54.  7.100 / 54.  7.100
Input #0, mp3, from 'abba128k.mp3':
  Metadata:
    title           : Fernando
    album           : Gold: Greatest Hits
    artist          : ABBA
    date            : 1993
    genre           : Dance
    track           : 12
    encoder         : Lavf57.83.100
  Duration: 00:04:13.49, start: 0.025057, bitrate: 128 kb/s
    Stream #0:0: Audio: mp3, 44100 Hz, stereo, s16p, 128 kb/s
    Metadata:
      encoder         : Lavc57.10
Stream mapping:
  Stream #0:0 -> #0:0 (mp3 (native) -> mp3 (libmp3lame))
Press [q] to stop, [?] for help
Output #0, mp3, to 'abba128k_vh.mp3':
  Metadata:
    TIT2            : Fernando
    TALB            : Gold: Greatest Hits
    TPE1            : ABBA
    TDRC            : 1993
    TCON            : Dance
    TRCK            : 12
    TSSE            : Lavf57.83.100
    Stream #0:0: Audio: mp3 (libmp3lame), 44100 Hz, stereo, s16p
    Metadata:
      encoder         : Lavc57.107.100 libmp3lame
size=    7921kB time=00:08:26.88 bitrate= 128.0kbits/s speed=  25x
video:0kB audio:7920kB subtitle:0kB other streams:0kB global headers:0kB muxir
```

```
abba128k_h = './abba128k_vh.mp3'
print(abba128k_h)
fer128_h , sr128_h = librosa.load(abba128k_h)
IPython.display.Audio(data=fer128_h, rate=sr128_h)
```

Repita todos os passos para um arquivo de áudio original de preferência de um CD, mas ao invés da opção codificação com taxa fixa (-b:a), use -q:a N (com N um número de 0 a 9), de 320 a 65kbps.

Relatório

- Comparar os espectrogramas e as taxas de compressão nos arquivos gravados (façam tabelas e as analisem!).
- Apresentar as formas de onda e os espectrogramas para os arquivos com tempo modificados.
- Apresentar aquivos de áudio diferentes para cada integrante do grupo: sugerimos escolherem músicas com intrumental e ritmos diferentes.

```
import numpy as np
from scipy import signal
import matplotlib.pyplot as plt
import librosa
import librosa.display
import IPython.display
import math
```

```
  !ffmpeg -h
  !ffmpeg -codecs
```

```
!ffmpeg -i '/content/drive/MyDrive/Colab Notebooks/LAB 8 - CSM/Mazurca_Chopin_opus7
chopin320k = './chopin320k.mp3'
chp320 , sr320 = librosa.load(chopin320k)
print(type(chp320), type(sr320))
print(chp320.shape, sr320)
# Player será aberto! AGUARDE até abrir!
IPython.display.Audio(data=chp320, rate=sr320)
```

```
y
yfor N in range(10):
  !ffmpeg -i "/content/drive/MyDrive/Colab Notebooks/LAB 8 - CSM/Mazurca_Chopin_opu
  chopin320k = './chopin320k.mp3'
  print(chopin320k)
  chp320 , sr320 = librosa.load(chopin320k)
  print(type(chp320), type(sr320))
  print(chp320.shape, sr320)
  IPython.display.Audio(data=chp320, rate=sr320)
  plt.figure(figsize=(20, 5))
  D = librosa.amplitude_to_db(np.abs(librosa.stft(fer320)), ref=np.max)
  librosa.display.specshow(D, x_axis='time',y_axis='linear')
  plt.colorbar(format='%+2.0f dB')
  plt.title('Linear-frequency power spectrogram'+ chopin320k)
```

```
    File "<ipython-input-95-6bd5a8ba7ed6>", line 3
      chopin{N} = './chopin320k.mp3'
            ^
  SyntaxError: invalid syntax
```

```
!ffmpeg -i '/content/drive/MyDrive/Colab Notebooks/LAB 8 - CSM/01. Green Light.mp3'
green320k = './green320k.mp3'
fer320 , sr320 = librosa.load(green320k)
print(type(fer320), type(sr320))
print(fer320.shape, sr320)
# Player será aberto! AGUARDE até abrir!
IPython.display.Audio(data=fer320, rate=sr320)
```

```
for N in range(10):
  !ffmpeg -i '/content/drive/MyDrive/Colab Notebooks/LAB 8 - CSM/01. Green Light.mp
  green320k = './green320k.mp3'
  print(green320k)
  fer320 , sr320 = librosa.load(green320k)
  print(type(fer320), type(sr320))
  print(fer320.shape, sr320)
  # Player será aberto! AGUARDE até abrir!
  IPython.display.Audio(data=fer320, rate=sr320)
  plt.figure(figsize=(20, 5))
  D = librosa.amplitude_to_db(np.abs(librosa.stft(fer320)), ref=np.max)
  librosa.display.specshow(D, x_axis='time',y_axis='linear')
  plt.colorbar(format='%+2.0f dB')
  plt.title('Linear-frequency power spectrogram'+ green320k)
```

```
!ffmpeg -i '/content/drive/MyDrive/Colab Notebooks/LAB 8 - CSM/02. Youth.mp3' -code
youth320k = './youth320k.mp3'
fer320 , sr320 = librosa.load(youth320k)
print(type(fer320), type(sr320))
print(fer320.shape, sr320)
# Player será aberto! AGUARDE até abrir!
IPython.display.Audio(data=fer320, rate=sr320)
```

```
for N in range(10):
  !ffmpeg -i '/content/drive/MyDrive/Colab Notebooks/LAB 8 - CSM/02. Youth.mp3' -co
  youth320k = './youth320k.mp3'
  print(youth320k)
  fer320 , sr320 = librosa.load(youth320k)
  print(type(fer320), type(sr320))
  print(fer320.shape, sr320)
  # Player será aberto! AGUARDE até abrir!
  IPython.display.Audio(data=fer320, rate=sr320)
  plt.figure(figsize=(20, 5))
  D = librosa.amplitude_to_db(np.abs(librosa.stft(fer320)), ref=np.max)
  librosa.display.specshow(D, x_axis='time',y_axis='linear')
```

```
plt.colorbar(format='%+2.0f dB')
plt.title('Linear-frequency power spectrogram'+ youth320k)
```

```
ffmpeg version 3.4.8-0ubuntu0.2 Copyright (c) 2000-2020 the FFmpeg developers
  built with gcc 7 (Ubuntu 7.5.0-3ubuntu1~18.04)
  configuration: --prefix=/usr --extra-version=0ubuntu0.2 --toolchain=hardened
  libavutil      55. 78.100 / 55. 78.100
  libavcodec     57.107.100 / 57.107.100
  libavformat    57. 83.100 / 57. 83.100
  libavdevice    57. 10.100 / 57. 10.100
  libavfilter     6.107.100 /  6.107.100
  libavresample   3.  7.  0 /  3.  7.  0
  libswscale      4.  8.100 /  4.  8.100
  libswresample   2.  9.100 /  2.  9.100
  libpostproc    54.  7.100 / 54.  7.100
Input #0, mp3, from '/content/drive/MyDrive/Colab Notebooks/LAB 8 - CSM/02. Yo
  Metadata:
    track           : 02
    title           : Youth
    artist          : Glass Animals
    album           : How To Be A Human Being
    genre           : Electronic
    date            : 2016
  Duration: 00:03:48.81, start: 0.025056, bitrate: 322 kb/s
    Stream #0:0: Audio: mp3, 44100 Hz, stereo, s16p, 320 kb/s
    Metadata:
      encoder         : LAME3.99r
    Side data:
      replaygain: track gain - -7.600000, track peak - unknown, album gain - u
    Stream #0:1: Video: mjpeg, yuvj444p(pc, bt470bg/unknown/unknown), 600x600
    Metadata:
      title           : Glass Animals
      comment         : Cover (front)
File 'youth320k.mp3' already exists. Overwrite ? [y/N] y
Stream mapping:
  Stream #0:1 -> #0:0 (mjpeg (native) -> png (native))
  Stream #0:0 -> #0:1 (mp3 (native) -> mp3 (libmp3lame))
Press [q] to stop, [?] for help
[swscaler @ 0x55a0632de000] deprecated pixel format used, make sure you did se
[mp3 @ 0x55a06328a600] Frame rate very high for a muxer not efficiently suppor
Please consider specifying a lower framerate, a different muxer or -vsync 2
Output #0, mp3, to 'youth320k.mp3':
  Metadata:
    TRCK            : 02
    TIT2            : Youth
    TPE1            : Glass Animals
    TALB            : How To Be A Human Being
    TCON            : Electronic
    TDRC            : 2016
    TSSE            : Lavf57.83.100
    Stream #0:0: Video: png, rgb24(progressive), 600x600 [SAR 1:1 DAR 1:1], q=
    Metadata:
      title           : Glass Animals
      comment         : Cover (front)
      encoder         : Lavc57.107.100 png
    Stream #0:1: Audio: mp3 (libmp3lame), 44100 Hz, stereo, s16p
    Metadata:
      encoder         : Lavc57.107.100 libmp3lame
    Side data:
      replaygain: track gain - -7.600000, track peak - unknown, album gain - u
```

```
[mp3 @ 0x55a0632b9900] Header missing
```

```
!ffmpeg -i '/content/drive/MyDrive/Colab Notebooks/LAB 8 - CSM/01. Green Light.mp3'
push320k = './push320k.mp3'
fer320 , sr320 = librosa.load(push320k)
print(type(fer320), type(sr320))
print(fer320.shape, sr320)
# Player será aberto! AGUARDE até abrir!
IPython.display.Audio(data=fer320, rate=sr320)
```

```
for N in range(10):
  !ffmpeg -i '/content/drive/MyDrive/Colab Notebooks/LAB 8 - CSM/09 - Push.mp3' -co
  push320k = './push320k.mp3'
  print(push320k)
  fer320 , sr320 = librosa.load(push320k)
  print(type(fer320), type(sr320))
  print(fer320.shape, sr320)
  # Player será aberto! AGUARDE até abrir!
  IPython.display.Audio(data=fer320, rate=sr320)
  plt.figure(figsize=(20, 5))
  D = librosa.amplitude_to_db(np.abs(librosa.stft(fer320)), ref=np.max)
  librosa.display.specshow(D, x_axis='time',y_axis='linear')
  plt.colorbar(format='%+2.0f dB')
  plt.title('Linear-frequency power spectrogram'+ push320k)
```

Conclusão:

Como pudemos perceber as representações no espectômetro são bem semelhantes enquanto variamos o N. Se levarmos em conta a variação da taxa de compressão, cada vez que ela diminui os valores de frequencia também diminuem tornando as ondas menos largas em relação ao eixo y tornando assim o som um pouco mais grave mesmo que no geral a variação seja quase imperceptível.

● ✕