

▼ UFABC - ESTI019 - Codificação de Sinais Multimídia

Laboratório 6 - PDS da Voz

Prof. Mário Minami

OBJETIVOS:

1. Gravar Arquivos de Áudio com dígitos, números gerais, texto lido e poesia
2. Leitura de Arquivos de Áudio e janelamento
3. Cálculo das Energias de Tempo Curto
4. Cálculo dos Espectrogramas
5. Determinação do Pitch, da Frequência Fundamental e das Formantes (f_1 a f_4) de Algumas Vogais
6. Determinação de fonemas surdos, sonoros, consoantes gerais e plosivos

1. Gravar Arquivos de Áudio com dígitos, números gerais, texto lido e poesia

Usando o Audacity, ou outro programa de áudio, grave arquivos com:

- Dígitos (pronunciados separadamente)
- Números Gerais (p.ex. dia/mês/ano de nascimento, nº celular, RA)
- Texto lido
- Poesia declamada

DICAS:

1. *Ajuste antes o nível do volume do microfone, para uma boa faixa dinâmica, evitando saturar as gravações;*
2. *Para os dígitos isolados, cuide da pausa entre eles, pelo menos uns 100 ms entre um e outro, o que não precisa ocorrer nos números gerais;*
3. *Treinar a leitura antes, tanto do texto quanto da poesia, para obter boas gravações.*

▼ Descrição dos audios:

Dígito: Números do RA dito digito a digito

Número geral: Número do RA dito em pares

Texto:Camoes

Poema:Sagarana

2. Leitura de Arquivos de Áudio e janelamento

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

Mounted at /content/drive

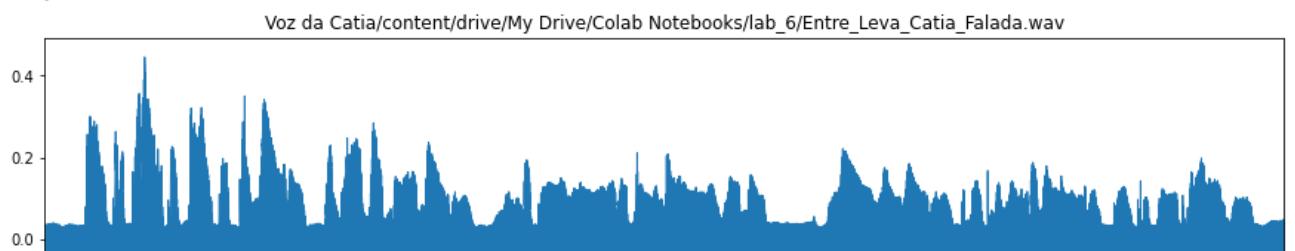
import numpy as np
import matplotlib.pyplot as plt
import librosa
import librosa.display
import IPython.display
#OBS.: insira nome do arquivo WAV
audio1 = '/content/drive/My Drive/Colab Notebooks/lab_6/Entre_Leva_Catia_Falada.wav'
print(audio1)
v1 , sr1 = librosa.load(audio1)
print(type(v1), type(sr1))
print(v1.shape, sr1)
# Player será aberto! AGUARDE até abrir!
IPython.display.Audio(data=v1, rate=sr1)

/content/drive/My Drive/Colab Notebooks/lab_6/Entre_Leva_Catia_Falada.wav
<class 'numpy.ndarray'> <class 'int'>
(292805,) 22050
```

0:00 / 0:13

```
plt.figure()
fig, ax = plt.subplots(figsize=(15, 5))
librosa.display.waveplot(v1, sr=sr1)
plt.title('Voz da Catia' + audio1)
```

```
Text(0.5, 1.0, 'Voz da Catia/content/drive/My Drive/Colab Notebooks/lab_6/Entr  
<Figure size 432x288 with 0 Axes>
```



▼ 2.1 Definição dos Parâmetros da Análise

```
| | | | |
```

```
print('Frequência de Amostragem', sr1)
```

```
Frequência de Amostragem 22050
```

```
fa = sr1  
print(fa)
```

```
22050
```

```
Ts = 0.04 # Tempo de duração do segmento em segundos  
Nj = int(Ts*fa) # Número de pontos da Janela  
print('Tamanho do Segmento', Nj)
```

```
Tamanho do Segmento 882
```

```
Nseg = int(len(v1)/Nj)  
Nover = int(Nj*0.5)  
print('Número de Segmentos no Arquivo', Nseg, '. Pontos Sobrepostos', Nover )
```

```
Número de Segmentos no Arquivo 331 . Pontos Sobrepostos 441
```

2.2 Obtendo uma Janela de Hamming

```
from scipy import signal  
hm = signal.get_window('hamming', Nj)  
plt.plot(hm); plt.title('Janela de Hamming')
```

```
Text(0.5, 1.0, 'Janela de Hamming')
```



3. Energia de Tempo Curto

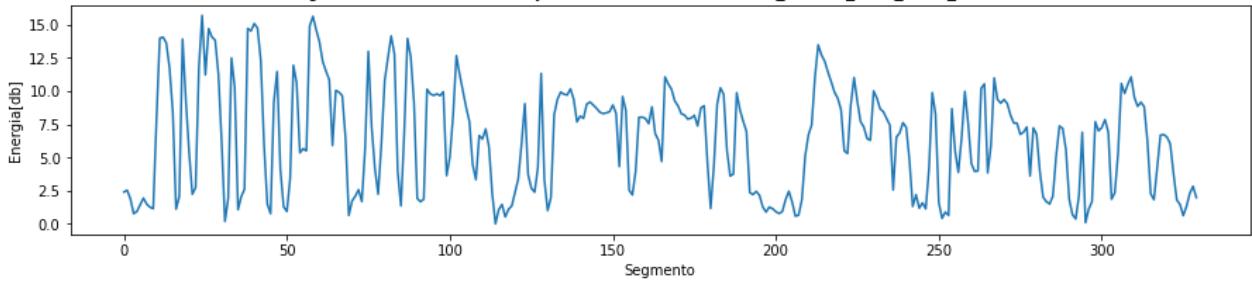
3.1 Cálculo para arquivo de teste

```
Nover = int(Nj*0.5)
Nseg = int(len(v1)/Nj)
E = []
for l in range(1, Nseg):
    xjan = v1[(l-1)*Nj+Nover:l*Nj+Nover]*hm
    x2 = list(np.array(xjan**2))
    aux = sum(x2)/Nj
    E.append(aux)
E = 10*np.log10(E)
Emin = np.min(E) # calcula nível de ruído de fundo
```

```
plt.figure
fig, ax = plt.subplots(figsize=(15, 3))
plt.plot(E - Emin)
plt.title('Energia da Voz' + audio1)
plt.ylabel('Energia[db]'); plt.xlabel('Segmento')
```

```
Text(0.5, 0, 'Segmento')
```

Energia da Voz/content/drive/My Drive/Colab Notebooks/lab_6/Entre_Leva_Catia_Falada.wav



3.2 Agora leia os seus arquivos gravados e:

Determine o Contorno de Energia para três arquivos que vocês gravaram:

1. dígitos
2. voz falada
3. voz declamada (poema)

▼ Lucas Poema

```
audio_poema_lucas = '/content/drive/My Drive/Colab Notebooks/lab_6/Sagarana_LucasEd
print(audio_poema_lucas)
v_poema_lucas , sr_poema_lucas = librosa.load(audio_poema_lucas)
print(type(v_poema_lucas), type(sr_poema_lucas))
print(v_poema_lucas.shape, sr_poema_lucas)
# Player será aberto! AGUARDE até abrir!
IPython.display.Audio(data=v_poema_lucas, rate=sr_poema_lucas)
```

```
/content/drive/My Drive/Colab Notebooks/lab_6/Sagarana_LucasEduardo.wav
/usr/local/lib/python3.7/dist-packages/librosa/core/audio.py:165: UserWarning:
    warnings.warn("PySoundFile failed. Trying audioread instead.")
<class 'numpy.ndarray'> <class 'int'>
(439236,) 22050
```

0:00 / 0:19

▼ Lucas Digito

```
audio_digitos_lucas = '/content/drive/My Drive/Colab Notebooks/lab_6/digitos_LucasE
print(audio_digitos_lucas)
v_digitos_lucas , sr_digitos_lucas = librosa.load(audio_digitos_lucas)
print(type(v_digitos_lucas), type(sr_digitos_lucas))
print(v_digitos_lucas.shape, sr_digitos_lucas)
# Player será aberto! AGUARDE até abrir!
IPython.display.Audio(data=v_digitos_lucas, rate=sr_digitos_lucas)
```

```
/content/drive/My Drive/Colab Notebooks/lab_6/digitos_LucasEduardo.wav
/usr/local/lib/python3.7/dist-packages/librosa/core/audio.py:165: UserWarning:
    warnings.warn("PySoundFile failed. Trying audioread instead.")
<class 'numpy.ndarray'> <class 'int'>
(207711,) 22050
```

0:00 / 0:09

▼ Lucas texto lido negrito

```
audio_texto_lido_lucas =' /content/drive/My Drive/Colab Notebooks/lab_6/Camoes_Lucas
print(audio_texto_lido_lucas)
v_texto_lido_lucas , sr_texto_lido_lucas = librosa.load(audio_texto_lido_lucas)
print(type(v_texto_lido_lucas), type(sr_texto_lido_lucas))
print(v_texto_lido_lucas.shape, sr_texto_lido_lucas)
# Player será aberto! AGUARDE até abrir!
IPython.display.Audio(data=v_texto_lido_lucas, rate=sr_texto_lido_lucas)
```

```
/content/drive/My Drive/Colab Notebooks/lab_6/Camoes_LucasEduardo.wav
/usr/local/lib/python3.7/dist-packages/librosa/core/audio.py:165: UserWarning:
  warnings.warn("PySoundFile failed. Trying audioread instead.")
<class 'numpy.ndarray'> <class 'int'>
(215649,) 22050
```

0:00 / 0:09

▼ Muriel poema

```
audio_poema_muriel = '/content/drive/My Drive/Colab Notebooks/lab_6/Sagarana_Muriel
print(audio_poema_muriel)
v_poema_muriel , sr_poema_muriel = librosa.load(audio_poema_muriel)
print(type(v_poema_muriel), type(sr_poema_muriel))
print(v_poema_muriel.shape, sr_poema_muriel)
# Player será aberto! AGUARDE até abrir!
IPython.display.Audio(data=v_poema_muriel, rate=sr_poema_muriel)
```

```
/content/drive/My Drive/Colab Notebooks/lab_6/Sagarana_MurielDaCosta.wav
/usr/local/lib/python3.7/dist-packages/librosa/core/audio.py:165: UserWarning:
  warnings.warn("PySoundFile failed. Trying audioread instead.")
<class 'numpy.ndarray'> <class 'int'>
(331191,) 22050
```

0:00 / 0:15

▼ Muriel Digito

```
audio_digitos_muriel = '/content/drive/My Drive/Colab Notebooks/lab_6/digitos_Murie
print(audio_digitos_muriel)
v_digitos_muriel , sr_digitos_muriel = librosa.load(audio_digitos_muriel)
print(type(v_digitos_muriel), type(sr_digitos_muriel))
print(v_digitos_muriel.shape, sr_digitos_muriel)
# Player será aberto! AGUARDE até abrir!
IPython.display.Audio(data=v_digitos_muriel, rate=sr_digitos_muriel)
```

```
/content/drive/My Drive/Colab Notebooks/lab_6/digitos_MurielDaCosta.wav
/usr/local/lib/python3.7/dist-packages/librosa/core/audio.py:165: UserWarning:
  warnings.warn("PySoundFile failed. Trying audioread instead.")
<class 'numpy.ndarray'> <class 'int'>
(192276,) 22050
```

0:00 / 0:08

▼ Muriel Texto Lido

```
audio_texto_lido_muriel ='/content/drive/My Drive/Colab Notebooks/lab_6/Camoes_Muri
print(audio_texto_lido_muriel)
v_texto_lido_muriel , sr_texto_lido_muriel = librosa.load(audio_texto_lido_muriel)
print(type(v_texto_lido_muriel), type(sr_texto_lido_muriel))
print(v_texto_lido_muriel.shape, sr_texto_lido_muriel)
# Player será aberto! AGUARDE até abrir!
IPython.display.Audio(data=v_texto_lido_muriel, rate=sr_texto_lido_muriel)

/content/drive/My Drive/Colab Notebooks/lab_6/Camoes_MurielDaCosta.wav
/usr/local/lib/python3.7/dist-packages/librosa/core/audio.py:165: UserWarning:
    warnings.warn("PySoundFile failed. Trying audioread instead.")
<class 'numpy.ndarray'> <class 'int'>
(186984,) 22050
```

0:00 / 0:08

▼ Jefferson Poema

```
audio_poema_jefferson = '/content/drive/My Drive/Colab Notebooks/lab_6/Sagarana_Jef
print(audio_poema_jefferson)
v_poema_jefferson , sr_poema_jefferson = librosa.load(audio_poema_jefferson)
print(type(v_poema_jefferson), type(sr_poema_jefferson))
print(v_poema_jefferson.shape, sr_poema_jefferson)
# Player será aberto! AGUARDE até abrir!
IPython.display.Audio(data=v_poema_jefferson, rate=sr_poema_jefferson)

/content/drive/My Drive/Colab Notebooks/lab_6/Sagarana_JeffersonPaiva.wav
/usr/local/lib/python3.7/dist-packages/librosa/core/audio.py:165: UserWarning:
    warnings.warn("PySoundFile failed. Trying audioread instead.")
<class 'numpy.ndarray'> <class 'int'>
(349272,) 22050
```

0:00 / 0:15

▼ Jefferson Dígito

```
audio_digitos_jefferson = '/content/drive/My Drive/Colab Notebooks/lab_6/digitos_Je
print(audio_digitos_jefferson)
v_digitos_jefferson , sr_digitos_jefferson = librosa.load(audio_digitos_jefferson)
print(type(v_digitos_jefferson), type(sr_digitos_jefferson))
print(v_digitos_jefferson.shape, sr_digitos_jefferson)
```

```
# Player será aberto! AGUARDE até abrir!
IPython.display.Audio(data=v_digitos_jefferson, rate=sr_digitos_jefferson)

/content/drive/My Drive/Colab Notebooks/lab_6/digitos_JeffersonPaiva.wav
/usr/local/lib/python3.7/dist-packages/librosa/core/audio.py:165: UserWarning:
  warnings.warn("PySoundFile failed. Trying audioread instead.")
<class 'numpy.ndarray'> <class 'int'>
(168021,) 22050
```

0:00 / 0:07

▼ Jefferson texto lido

```
audio_texto_lido_jefferson = '/content/drive/My Drive/Colab Notebooks/lab_6/Camoes_J
print(audio_texto_lido_jefferson)
v_texto_lido_jefferson , sr_texto_lido_jefferson = librosa.load(audio_texto_lido_je
print(type(v_texto_lido_jefferson), type(sr_texto_lido_jefferson))
print(v_texto_lido_jefferson.shape, sr_texto_lido_jefferson)
# Player será aberto! AGUARDE até abrir!
IPython.display.Audio(data=v_texto_lido_jefferson, rate=sr_texto_lido_jefferson)

/content/drive/My Drive/Colab Notebooks/lab_6/Camoes_JeffersonPaiva.wav
/usr/local/lib/python3.7/dist-packages/librosa/core/audio.py:165: UserWarning:
  warnings.warn("PySoundFile failed. Trying audioread instead.")
<class 'numpy.ndarray'> <class 'int'>
(196245,) 22050
```

0:00 / 0:08

▼ Victor poema

```
audio_poema_victor = '/content/drive/My Drive/Colab Notebooks/lab_6/Sagarana_Victor
print(audio_poema_victor)
v_poema_victor , sr_poema_victor = librosa.load(audio_poema_victor)
print(type(v_poema_victor), type(sr_poema_victor))
print(v_poema_victor.shape, sr_poema_victor)
# Player será aberto! AGUARDE até abrir!
IPython.display.Audio(data=v_poema_victor, rate=sr_poema_victor)

/content/drive/My Drive/Colab Notebooks/lab_6/Sagarana_VictorMagalhaes.wav
/usr/local/lib/python3.7/dist-packages/librosa/core/audio.py:165: UserWarning:
  warnings.warn("PySoundFile failed. Trying audioread instead.")
<class 'numpy.ndarray'> <class 'int'>
(371763,) 22050
```

0:00 / 0:16

▼ Victor digitos

```
audio_digitos_victor = '/content/drive/My Drive/Colab Notebooks/lab_6/digitos_VictorMagalhaes.wav'
print(audio_digitos_victor)
v_digitos_victor, sr_digitos_victor = librosa.load(audio_digitos_victor)
print(type(v_digitos_victor), type(sr_digitos_victor))
print(v_digitos_victor.shape, sr_digitos_victor)
# Player será aberto! AGUARDE até abrir!
IPython.display.Audio(data=v_digitos_victor, rate=sr_digitos_victor)

/content/drive/My Drive/Colab Notebooks/lab_6/digitos_VictorMagalhaes.wav
/usr/local/lib/python3.7/dist-packages/librosa/core/audio.py:165: UserWarning:
  warnings.warn("PySoundFile failed. Trying audioread instead.")
<class 'numpy.ndarray'> <class 'int'>
(207711,) 22050
```

0:00 / 0:09

▼ Victor texto lido

```
audio_texto_lido_victor = '/content/drive/My Drive/Colab Notebooks/lab_6/Camoes_VictorMagalhaes.wav'
print(audio_texto_lido_victor)
v_texto_lido_victor, sr_texto_lido_victor = librosa.load(audio_texto_lido_victor)
print(type(v_texto_lido_victor), type(sr_texto_lido_victor))
print(v_texto_lido_victor.shape, sr_texto_lido_victor)
# Player será aberto! AGUARDE até abrir!
IPython.display.Audio(data=v_texto_lido_victor, rate=sr_texto_lido_victor)

/content/drive/My Drive/Colab Notebooks/lab_6/Camoes_VictorMagalhaes.wav
/usr/local/lib/python3.7/dist-packages/librosa/core/audio.py:165: UserWarning:
  warnings.warn("PySoundFile failed. Trying audioread instead.")
<class 'numpy.ndarray'> <class 'int'>
(202860,) 22050
```

0:00 / 0:09

4. Espectrogramas

► 4.1 Visualização do Espectrograma para Arquivo de Teste

[] ↗ 1 célula oculta

4.2 Agora com os seus arquivos

Faça os spectrogramas para os arquivos que vocês gravaram e calculem o contorno de energia:

- Dígitos
- Texto Lido
- Poesia

▼ Poemas waveplot, espectrogramas e contorno de energia

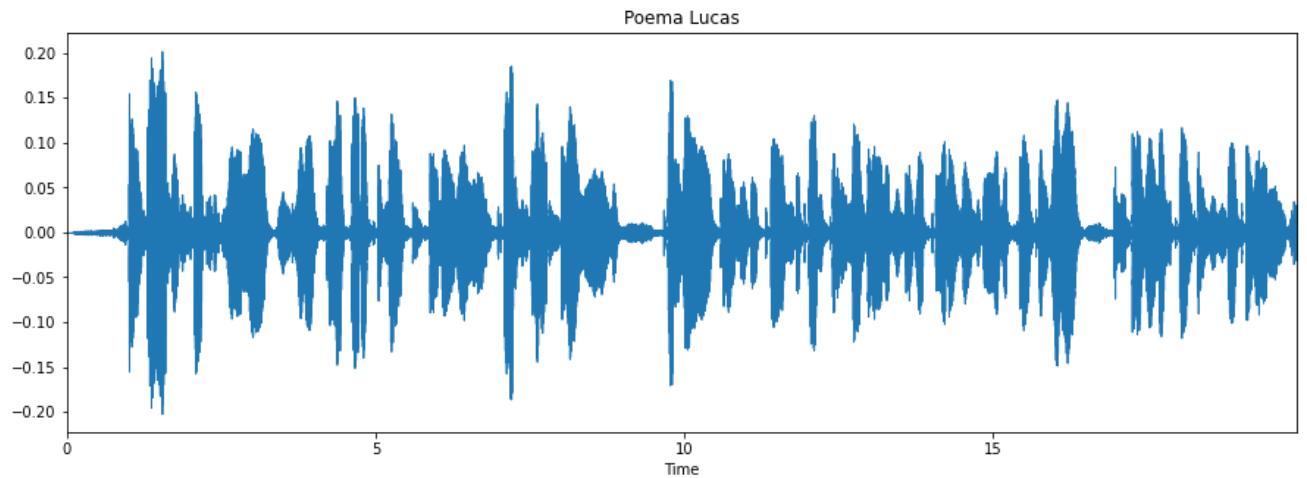
```
plt.figure()
fig, ax = plt.subplots(figsize=(15, 5))
librosa.display.waveplot(v_poema_lucas, sr=sr_poema_lucas)
plt.title('Poema Lucas')
plt.show()
```

```
plt.figure()
fig, ax = plt.subplots(figsize=(15, 5))
librosa.display.waveplot(v_poema_muriel, sr=sr_poema_muriel)
plt.title('Poema Muriel')
plt.show()
```

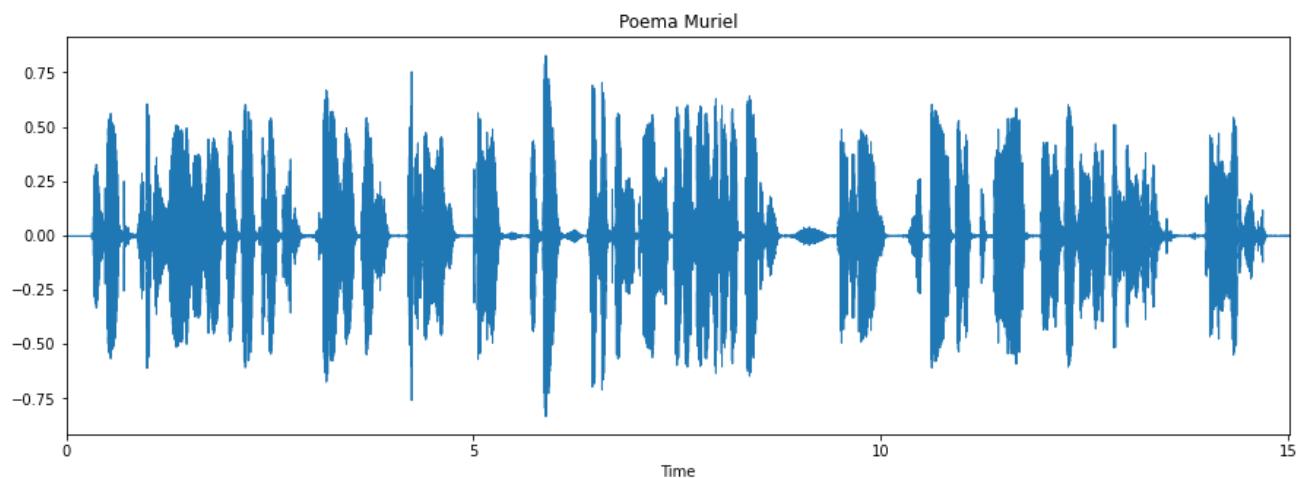
```
plt.figure()
fig, ax = plt.subplots(figsize=(15, 5))
librosa.display.waveplot(v_poema_muriel, sr=sr_poema_muriel)
plt.title('Poema Jefferson')
plt.show()
```

```
plt.figure()
fig, ax = plt.subplots(figsize=(15, 5))
librosa.display.waveplot(v_poema_muriel, sr=sr_poema_muriel)
plt.title('Poema Victor')
plt.show()
```

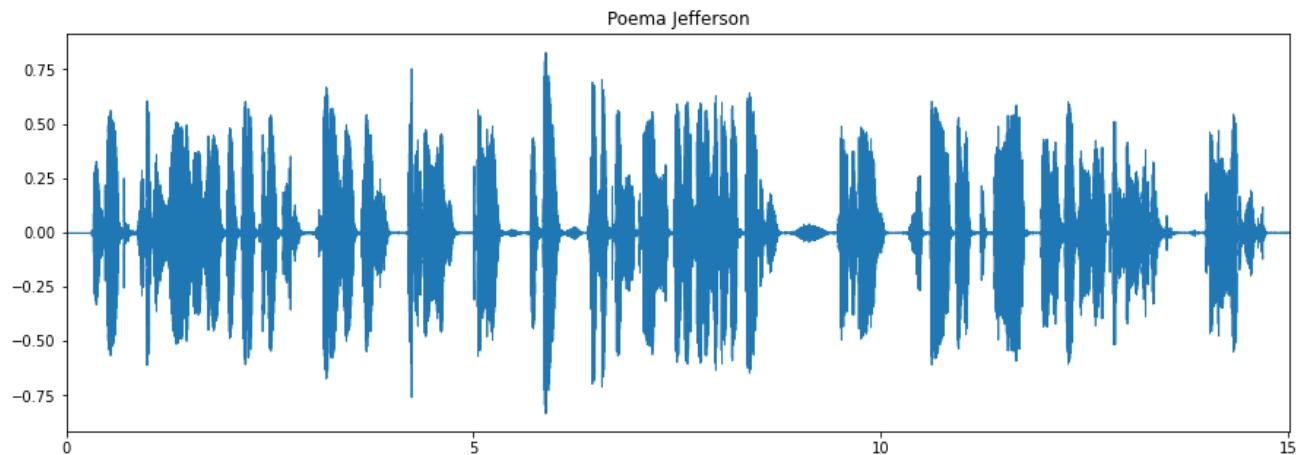
<Figure size 432x288 with 0 Axes>



<Figure size 432x288 with 0 Axes>



<Figure size 432x288 with 0 Axes>



```
#inputs
v_temp=v_poema_lucas
sr_temp=sr_poema_lucas
title='Poema Lucas'
#construcao da energia de voz
fa = sr_temp
Ts = 0.04 # Tempo de duração do segmento em segundos
Nj = int(Ts*fa) # Número de pontos da Janela
Nseg = int(len(v_temp)/Nj)
Nover = int(Nj*0.5)
from scipy import signal
hm = signal.get_window('hamming', Nj)
```

```

Nover = int(Nj*0.5)
Nseg = int(len(v_temp)/Nj)
E = []
for l in range(1, Nseg):
    xjan = v_temp[(l-1)*Nj+Nover:l*Nj+Nover]*hm
    x2 = list(np.array(xjan**2))
    aux = sum(x2)/Nj
    E.append(aux)
E = 10*np.log10(E)
Emin = np.min(E) # calcula nível de ruído de fundo
plt.figure
fig, ax = plt.subplots(figsize=(15, 3))
plt.plot(E - Emin)
plt.title('Energia da Voz' + title)
plt.ylabel('Energia[db]'); plt.xlabel('Segmento')
plt.show()

#inputs
v_temp=v_poema_muriel
sr_temp=sr_poema_muriel
title='Poema muriel'
#construcao da energia de voz
fa = sr_temp
Ts = 0.04 # Tempo de duração do segmento em segundos
Nj = int(Ts*fa) # Número de pontos da Janela
Nseg = int(len(v_temp)/Nj)
Nover = int(Nj*0.5)
from scipy import signal
hm = signal.get_window('hamming', Nj)

Nover = int(Nj*0.5)
Nseg = int(len(v_temp)/Nj)
E = []
for l in range(1, Nseg):
    xjan = v_temp[(l-1)*Nj+Nover:l*Nj+Nover]*hm
    x2 = list(np.array(xjan**2))
    aux = sum(x2)/Nj
    E.append(aux)
E = 10*np.log10(E)
Emin = np.min(E) # calcula nível de ruído de fundo
plt.figure
fig, ax = plt.subplots(figsize=(15, 3))
plt.plot(E - Emin)
plt.title('Energia da Voz' + title)
plt.ylabel('Energia[db]'); plt.xlabel('Segmento')
plt.show()

#inputs
v_temp=v_poema_jefferson
sr_temp=sr_poema_jefferson
title='Poema jefferson'
#construcao da energia de voz
fa = sr_temp
Ts = 0.04 # Tempo de duração do segmento em segundos

```

```

Nj = int(Ts*fa) # Número de pontos da Janela
Nseg = int(len(v_temp)/Nj)
Nover = int(Nj*0.5)
from scipy import signal
hm = signal.get_window('hamming', Nj)

Nover = int(Nj*0.5)
Nseg = int(len(v_temp)/Nj)
E = []
for l in range(1, Nseg):
    xjan = v_temp[(l-1)*Nj+Nover:l*Nj+Nover]*hm
    x2 = list(np.array(xjan**2))
    aux = sum(x2)/Nj
    E.append(aux)
E = 10*np.log10(E)
Emin = np.min(E) # calcula nível de ruído de fundo
plt.figure
fig, ax = plt.subplots(figsize=(15, 3))
plt.plot(E - Emin)
plt.title('Energia da Voz' + title)
plt.ylabel('Energia[db]'); plt.xlabel('Segmento')
plt.show()

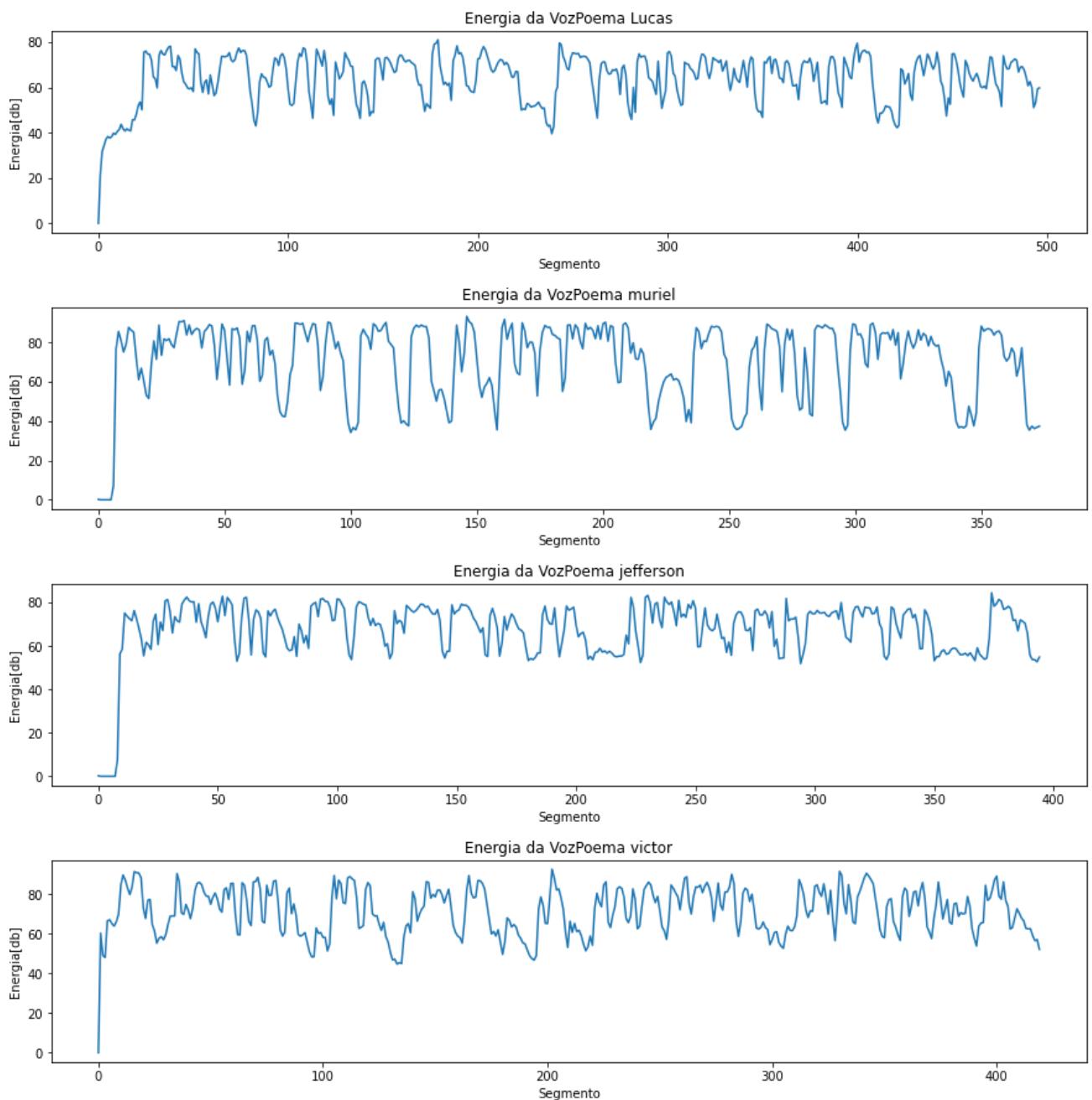
```

```

#inputs
v_temp=v_poema_victor
sr_temp=sr_poema_victor
title='Poema victor'
#construcao da energia de voz
fa = sr_temp
Ts = 0.04 # Tempo de duração do segmento em segundos
Nj = int(Ts*fa) # Número de pontos da Janela
Nseg = int(len(v_temp)/Nj)
Nover = int(Nj*0.5)
from scipy import signal
hm = signal.get_window('hamming', Nj)

Nover = int(Nj*0.5)
Nseg = int(len(v_temp)/Nj)
E = []
for l in range(1, Nseg):
    xjan = v_temp[(l-1)*Nj+Nover:l*Nj+Nover]*hm
    x2 = list(np.array(xjan**2))
    aux = sum(x2)/Nj
    E.append(aux)
E = 10*np.log10(E)
Emin = np.min(E) # calcula nível de ruído de fundo
plt.figure
fig, ax = plt.subplots(figsize=(15, 3))
plt.plot(E - Emin)
plt.title('Energia da Voz' + title)
plt.ylabel('Energia[db]'); plt.xlabel('Segmento')
plt.show()

```



```
#inputs
v_temp=v_poema_lucas
sr_temp=sr_poema_lucas
title='Poema Lucas'
```

```

plt.figure(figsize=(12, 8))
D = librosa.amplitude_to_db(np.abs(librosa.stft(v_temp)), ref=np.max)
fig, ax = plt.subplots(figsize=(15, 10))
librosa.display.specshow(D, x_axis='time',y_axis='linear')
plt.colorbar(format='%+2.0f dB')
plt.title('Potência e Espectrograma Linear na Frequênci' + title)
plt.show()

#inputs
v_temp=v_poema_muriel
sr_temp=sr_poema_muriel
title='Poema muriel'

plt.figure(figsize=(12, 8))
D = librosa.amplitude_to_db(np.abs(librosa.stft(v_temp)), ref=np.max)
fig, ax = plt.subplots(figsize=(15, 10))
librosa.display.specshow(D, x_axis='time',y_axis='linear')
plt.colorbar(format='%+2.0f dB')
plt.title('Potência e Espectrograma Linear na Frequênci' + title)
plt.show()

#inputs
v_temp=v_poema_jefferson
sr_temp=sr_poema_jefferson
title='Poema jefferson'

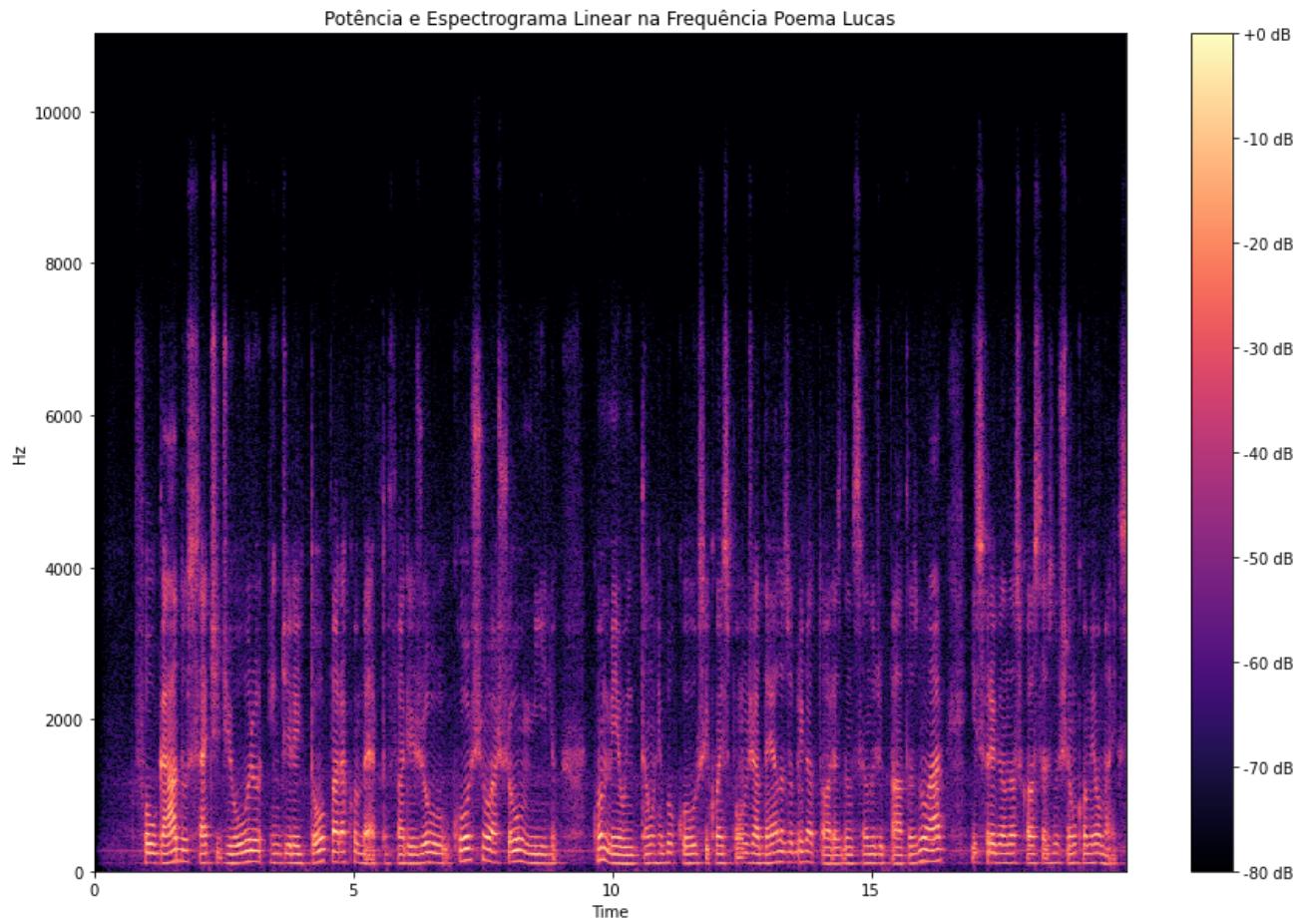
plt.figure(figsize=(12, 8))
D = librosa.amplitude_to_db(np.abs(librosa.stft(v_temp)), ref=np.max)
fig, ax = plt.subplots(figsize=(15, 10))
librosa.display.specshow(D, x_axis='time',y_axis='linear')
plt.colorbar(format='%+2.0f dB')
plt.title('Potência e Espectrograma Linear na Frequênci' + title)
plt.show()

#inputs
v_temp=v_poema_victor
sr_temp=sr_poema_victor
title='Poema victor'

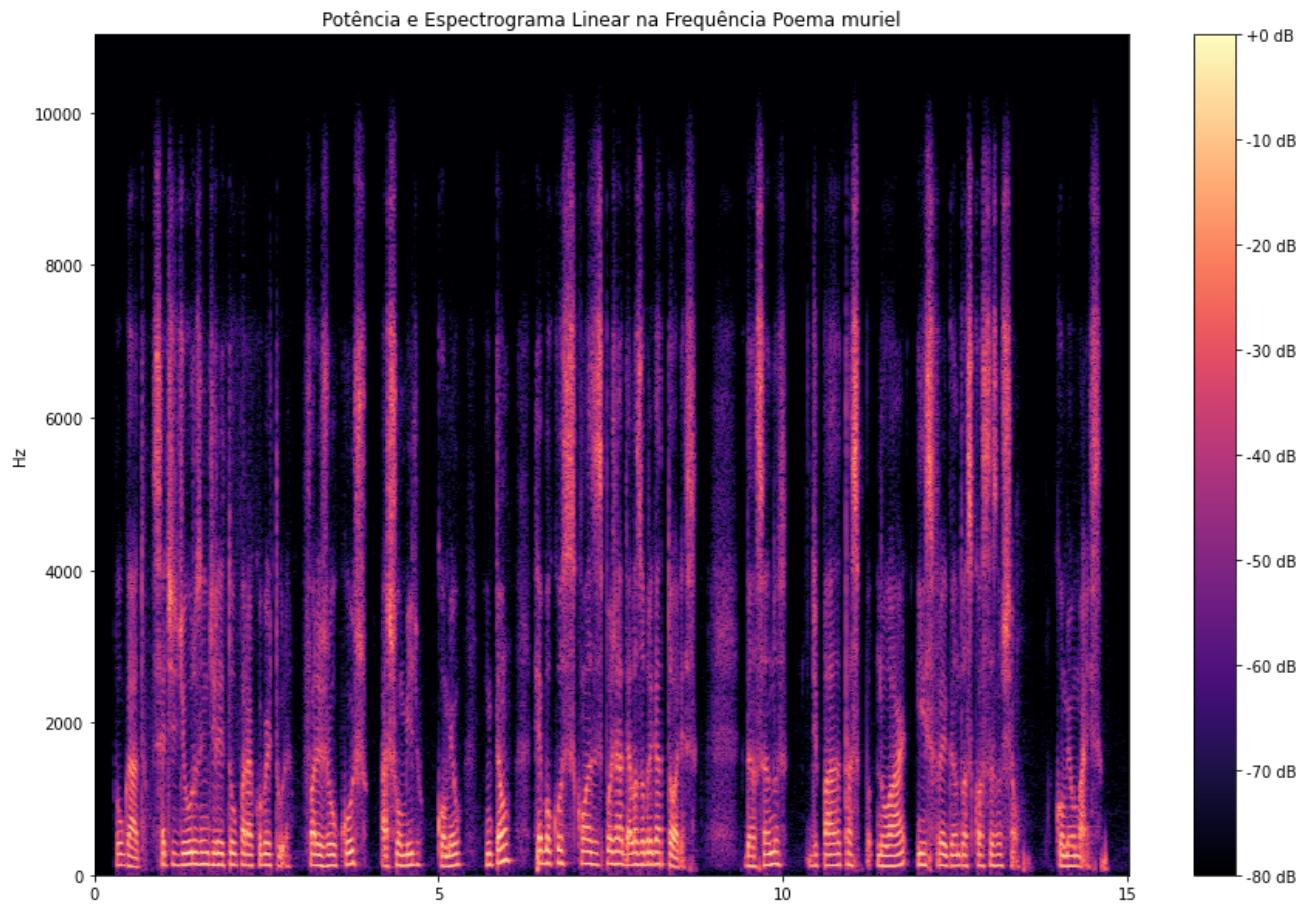
plt.figure(figsize=(12, 8))
D = librosa.amplitude_to_db(np.abs(librosa.stft(v_temp)), ref=np.max)
fig, ax = plt.subplots(figsize=(15, 10))
librosa.display.specshow(D, x_axis='time',y_axis='linear')
plt.colorbar(format='%+2.0f dB')
plt.title('Potência e Espectrograma Linear na Frequênci' + title)
plt.show()

```

<Figure size 864x576 with 0 Axes>



<Figure size 864x576 with 0 Axes>



Texto lido waveplot, espectrogramas e contorno de energia

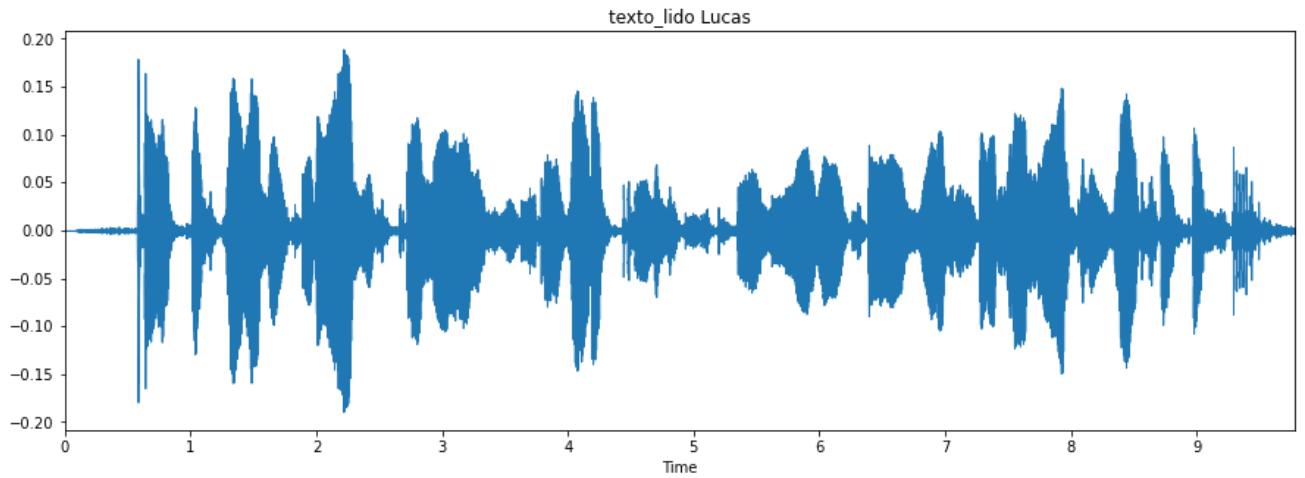
```
plt.figure()
fig, ax = plt.subplots(figsize=(15, 5))
librosa.display.waveplot(v_texto_lido_lucas, sr=sr_texto_lido_lucas)
plt.title('texto_lido Lucas')
plt.show()

plt.figure()
fig, ax = plt.subplots(figsize=(15, 5))
librosa.display.waveplot(v_texto_lido_muriel, sr=sr_texto_lido_muriel)
plt.title('texto_lido Muriel')
plt.show()

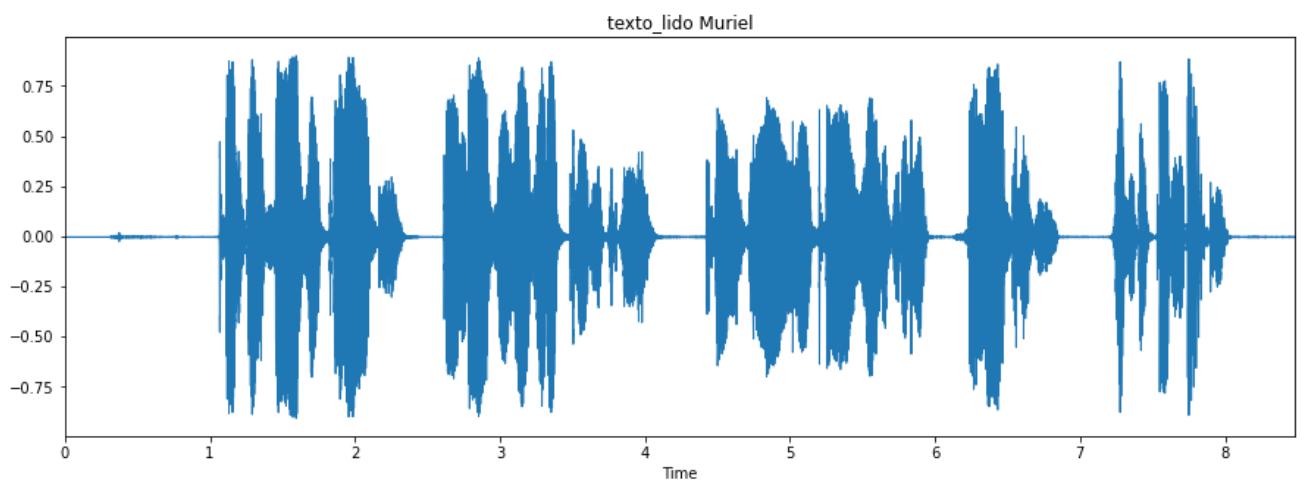
plt.figure()
fig, ax = plt.subplots(figsize=(15, 5))
librosa.display.waveplot(v_texto_lido_muriel, sr=sr_texto_lido_muriel)
plt.title('texto_lido Jefferson')
plt.show()

plt.figure()
fig, ax = plt.subplots(figsize=(15, 5))
librosa.display.waveplot(v_texto_lido_muriel, sr=sr_texto_lido_muriel)
plt.title('texto_lido Victor')
plt.show()
```

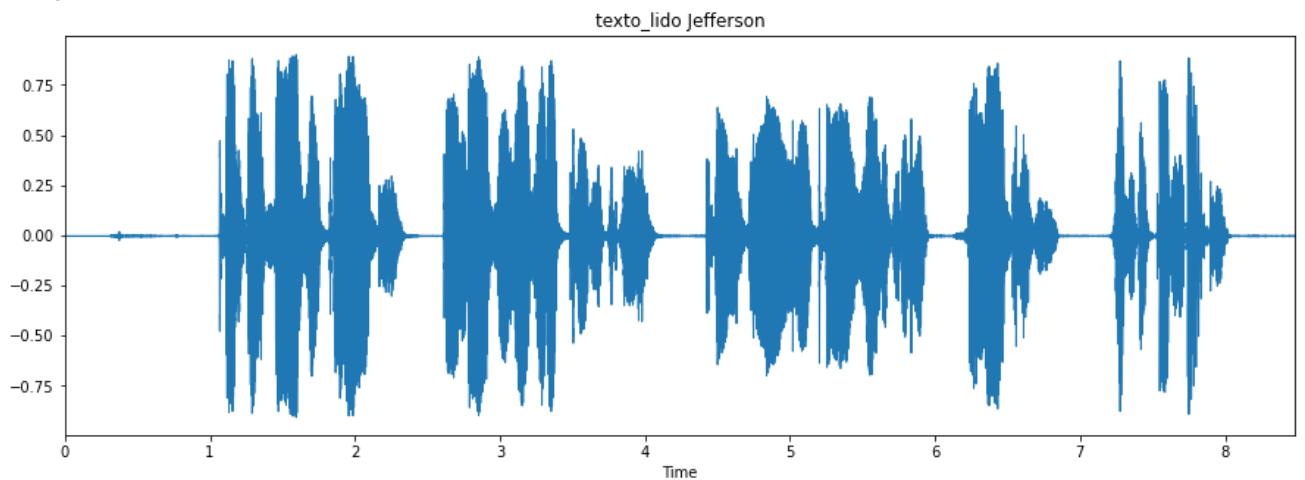
<Figure size 432x288 with 0 Axes>



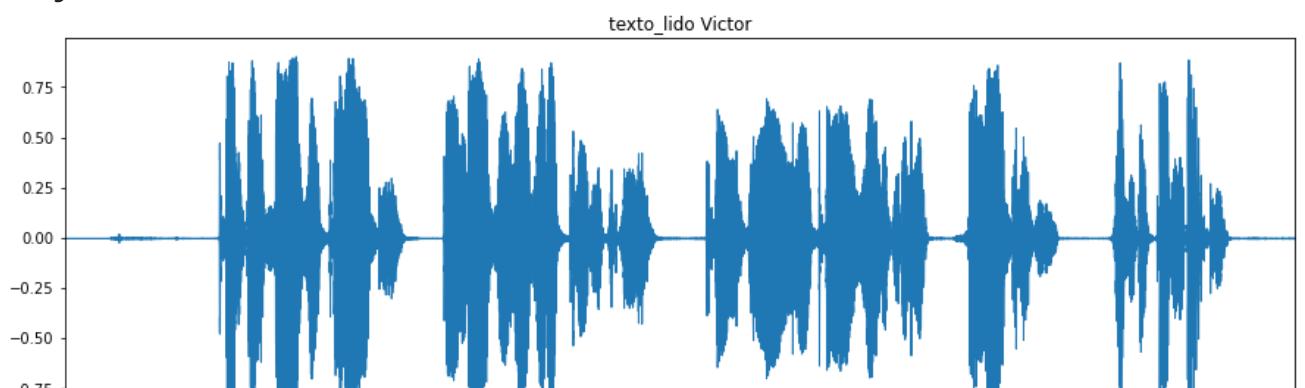
<Figure size 432x288 with 0 Axes>



<Figure size 432x288 with 0 Axes>



<Figure size 432x288 with 0 Axes>



```

#inputs
v_temp=v_texto_lido_lucas
sr_temp=sr_texto_lido_lucas
title='texto_lido Lucas'
#construcao da energia de voz
fa = sr_temp
Ts = 0.04 # Tempo de duração do segmento em segundos
Nj = int(Ts*fa) # Número de pontos da Janela
Nseg = int(len(v_temp)/Nj)
Nover = int(Nj*0.5)
from scipy import signal
hm = signal.get_window('hamming', Nj)

Nover = int(Nj*0.5)
Nseg = int(len(v_temp)/Nj)
E = []
for l in range(1, Nseg):
    xjan = v_temp[(l-1)*Nj+Nover:l*Nj+Nover]*hm
    x2 = list(np.array(xjan**2))
    aux = sum(x2)/Nj
    E.append(aux)
E = 10*np.log10(E)
Emin = np.min(E) # calcula nível de ruído de fundo
plt.figure
fig, ax = plt.subplots(figsize=(15, 3))
plt.plot(E - Emin)
plt.title('Energia da Voz ' + title)
plt.ylabel('Energia[db]'); plt.xlabel('Segmento')
plt.show()

```

```

#inputs
v_temp=v_texto_lido_muriel
sr_temp=sr_texto_lido_muriel
title='texto_lido muriel'
#construcao da energia de voz
fa = sr_temp
Ts = 0.04 # Tempo de duração do segmento em segundos
Nj = int(Ts*fa) # Número de pontos da Janela
Nseg = int(len(v_temp)/Nj)
Nover = int(Nj*0.5)
from scipy import signal
hm = signal.get_window('hamming', Nj)

Nover = int(Nj*0.5)
Nseg = int(len(v_temp)/Nj)
E = []
for l in range(1, Nseg):
    xjan = v_temp[(l-1)*Nj+Nover:l*Nj+Nover]*hm
    x2 = list(np.array(xjan**2))
    aux = sum(x2)/Nj
    E.append(aux)

```

```

E = 10*np.log10(E)
Emin = np.min(E) # calcula nível de ruído de fundo
plt.figure
fig, ax = plt.subplots(figsize=(15, 3))
plt.plot(E - Emin)
plt.title('Energia da Voz ' + title)
plt.ylabel('Energia[db]'); plt.xlabel('Segmento')
plt.show()

#inputs
v_temp=v_texto_lido_jefferson
sr_temp=sr_texto_lido_jefferson
title='texto_lido jefferson'
#construcao da energia de voz
fa = sr_temp
Ts = 0.04 # Tempo de duração do segmento em segundos
Nj = int(Ts*fa) # Número de pontos da Janela
Nseg = int(len(v_temp)/Nj)
Nover = int(Nj*0.5)
from scipy import signal
hm = signal.get_window('hamming', Nj)

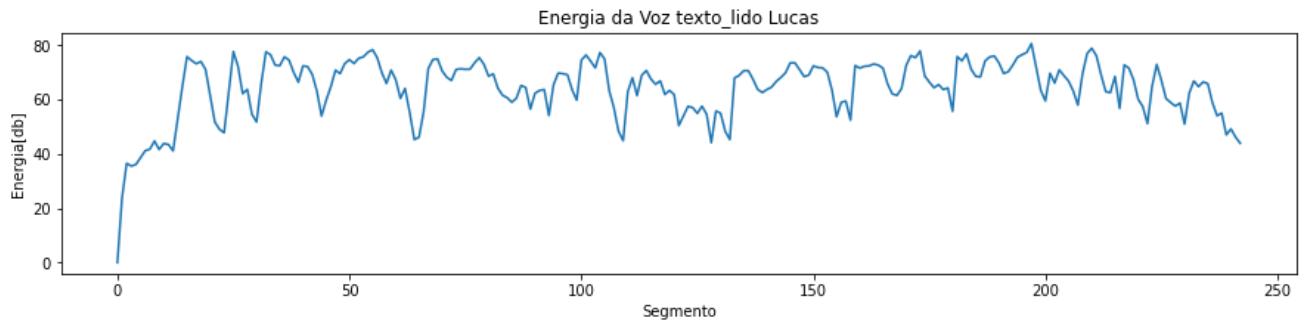
Nover = int(Nj*0.5)
Nseg = int(len(v_temp)/Nj)
E = []
for l in range(1, Nseg):
    xjan = v_temp[(l-1)*Nj+Nover:l*Nj+Nover]*hm
    x2 = list(np.array(xjan**2))
    aux = sum(x2)/Nj
    E.append(aux)
E = 10*np.log10(E)
Emin = np.min(E) # calcula nível de ruído de fundo
plt.figure
fig, ax = plt.subplots(figsize=(15, 3))
plt.plot(E - Emin)
plt.title('Energia da Voz ' + title)
plt.ylabel('Energia[db]'); plt.xlabel('Segmento')
plt.show()

#inputs
v_temp=v_texto_lido_victor
sr_temp=sr_texto_lido_victor
title='texto_lido victor'
#construcao da energia de voz
fa = sr_temp
Ts = 0.04 # Tempo de duração do segmento em segundos
Nj = int(Ts*fa) # Número de pontos da Janela
Nseg = int(len(v_temp)/Nj)
Nover = int(Nj*0.5)
from scipy import signal
hm = signal.get_window('hamming', Nj)

Nover = int(Nj*0.5)
Nseg = int(len(v_temp)/Nj)

```

```
E = []
for l in range(1, Nseg):
    xjan = v_temp[(l-1)*Nj+Nover:l*Nj+Nover]*hm
    x2 = list(np.array(xjan**2))
    aux = sum(x2)/Nj
    E.append(aux)
E = 10*np.log10(E)
Emin = np.min(E) # calcula nível de ruído de fundo
plt.figure
fig, ax = plt.subplots(figsize=(15, 3))
plt.plot(E - Emin)
plt.title('Energia da Voz ' + title)
plt.ylabel('Energia[db]'); plt.xlabel('Segmento')
plt.show()
```



Energia da Voz texto_lido muriel



```
#inputs
v_temp=v_texto_lido_lucas
sr_temp=sr_texto_lido_lucas
title='texto_lido Lucas'

plt.figure(figsize=(12, 8))
D = librosa.amplitude_to_db(np.abs(librosa.stft(v_temp)), ref=np.max)
fig, ax = plt.subplots(figsize=(15, 10))
librosa.display.specshow(D, x_axis='time',y_axis='linear')
plt.colorbar(format='%+2.0f dB')
plt.title('Potência e Espectrograma Linear na Frequênci a '+ title)
plt.show()

#inputs
v_temp=v_texto_lido_muriel
sr_temp=sr_texto_lido_muriel
title='texto_lido muriel'

plt.figure(figsize=(12, 8))
D = librosa.amplitude_to_db(np.abs(librosa.stft(v_temp)), ref=np.max)
fig, ax = plt.subplots(figsize=(15, 10))
librosa.display.specshow(D, x_axis='time',y_axis='linear')
plt.colorbar(format='%+2.0f dB')
plt.title('Potência e Espectrograma Linear na Frequênci a '+ title)
plt.show()

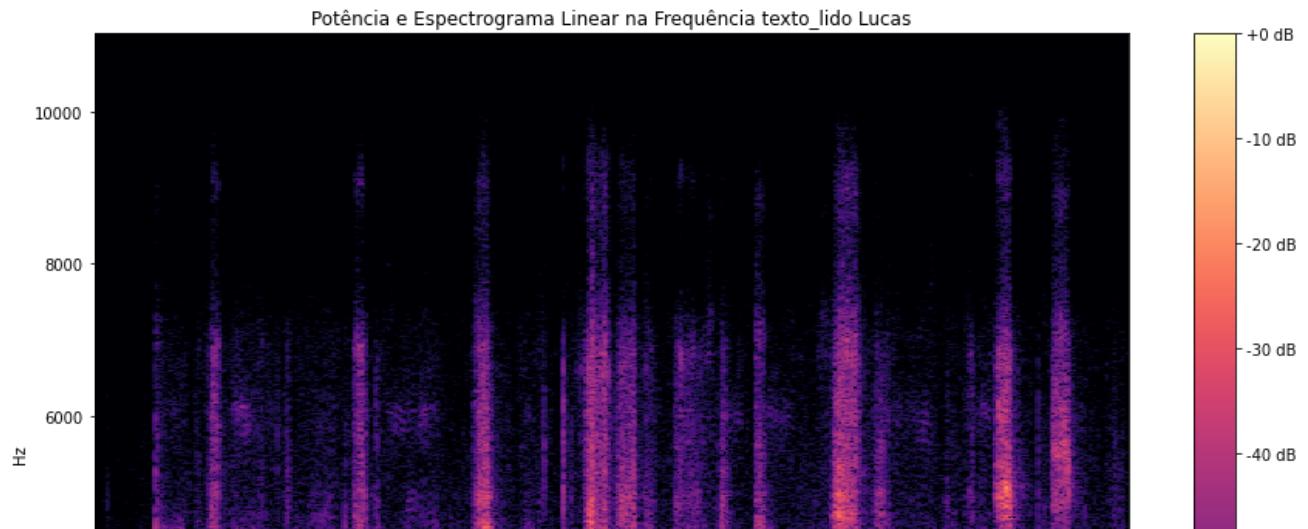
#inputs
v_temp=v_texto_lido_jefferson
sr_temp=sr_texto_lido_jefferson
title='texto_lido jefferson'

plt.figure(figsize=(12, 8))
D = librosa.amplitude_to_db(np.abs(librosa.stft(v_temp)), ref=np.max)
fig, ax = plt.subplots(figsize=(15, 10))
librosa.display.specshow(D, x_axis='time',y_axis='linear')
plt.colorbar(format='%+2.0f dB')
plt.title('Potência e Espectrograma Linear na Frequênci a '+ title)
plt.show()
```

```
#inputs
v_temp=v_texto_lido_victor
sr_temp=sr_texto_lido_victor
title='texto_lido victor'

plt.figure(figsize=(12, 8))
D = librosa.amplitude_to_db(np.abs(librosa.stft(v_temp)), ref=np.max)
fig, ax = plt.subplots(figsize=(15, 10))
librosa.display.specshow(D, x_axis='time',y_axis='linear')
plt.colorbar(format='%+2.0f dB')
plt.title('Potência e Espectrograma Linear na Frequênci' + title)
plt.show()
```

```
<Figure size 864x576 with 0 Axes>
```



▼ Dígitos waveplot, espectrogramas e contorno de energia



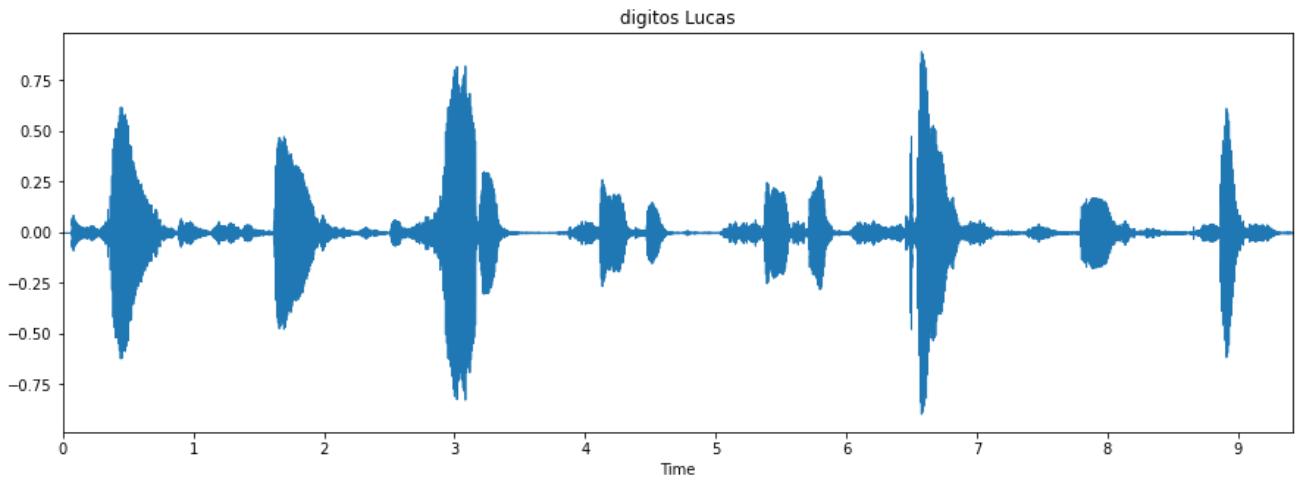
```
plt.figure()
fig, ax = plt.subplots(figsize=(15, 5))
librosa.display.waveplot(v_digitos_lucas, sr=sr_digitos_lucas)
plt.title('dígitos Lucas')
plt.show()
```

```
plt.figure()
fig, ax = plt.subplots(figsize=(15, 5))
librosa.display.waveplot(v_digitos_muriel, sr=sr_digitos_muriel)
plt.title('dígitos Muriel')
plt.show()
```

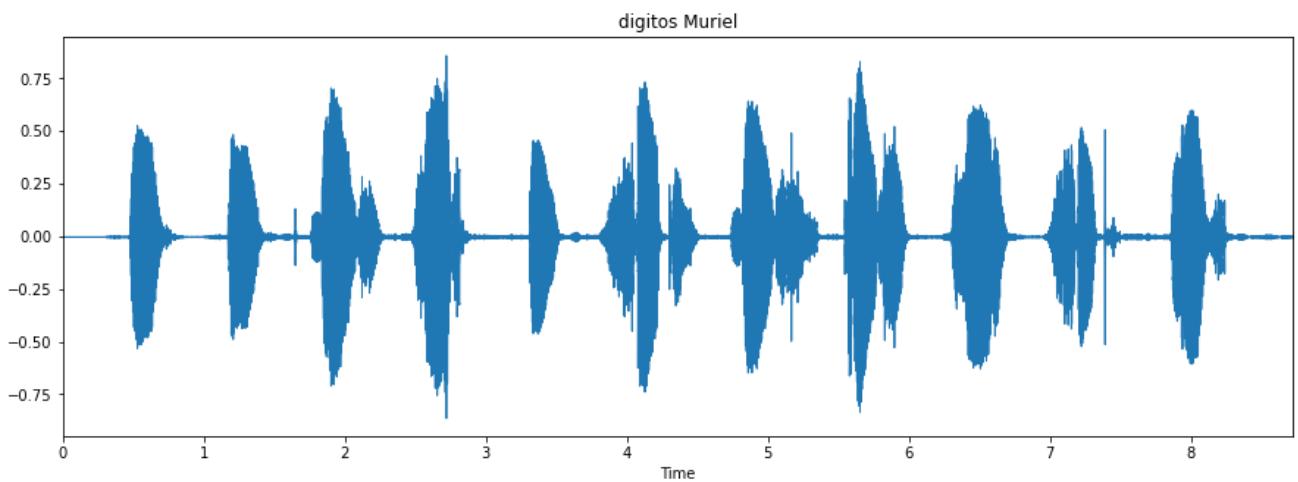
```
plt.figure()
fig, ax = plt.subplots(figsize=(15, 5))
librosa.display.waveplot(v_digitos_muriel, sr=sr_digitos_muriel)
plt.title('dígitos Jefferson')
plt.show()
```

```
plt.figure()
fig, ax = plt.subplots(figsize=(15, 5))
librosa.display.waveplot(v_digitos_muriel, sr=sr_digitos_muriel)
plt.title('dígitos Victor')
plt.show()
```

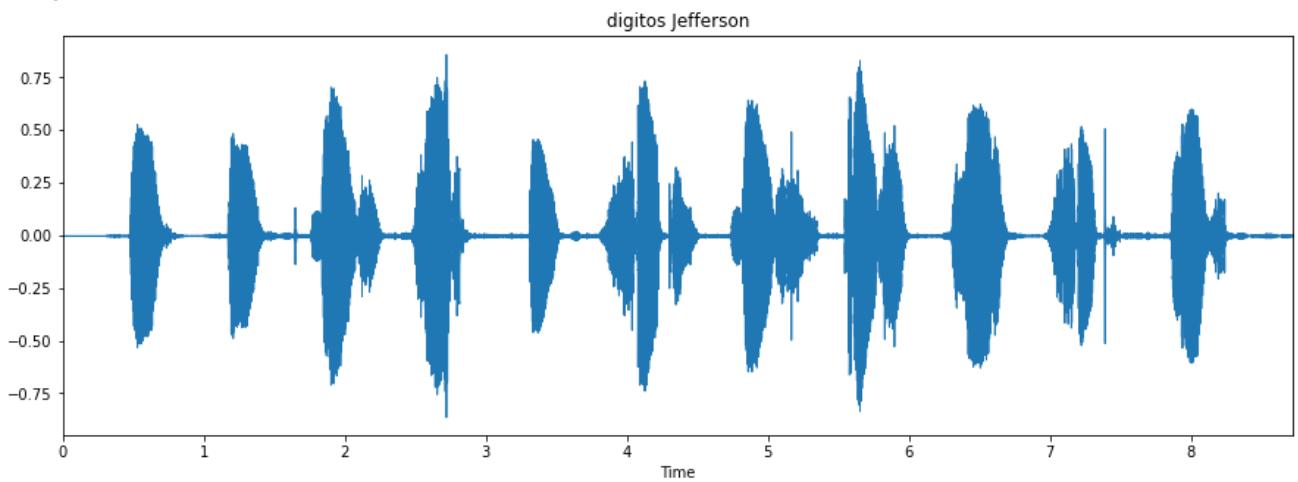
<Figure size 432x288 with 0 Axes>



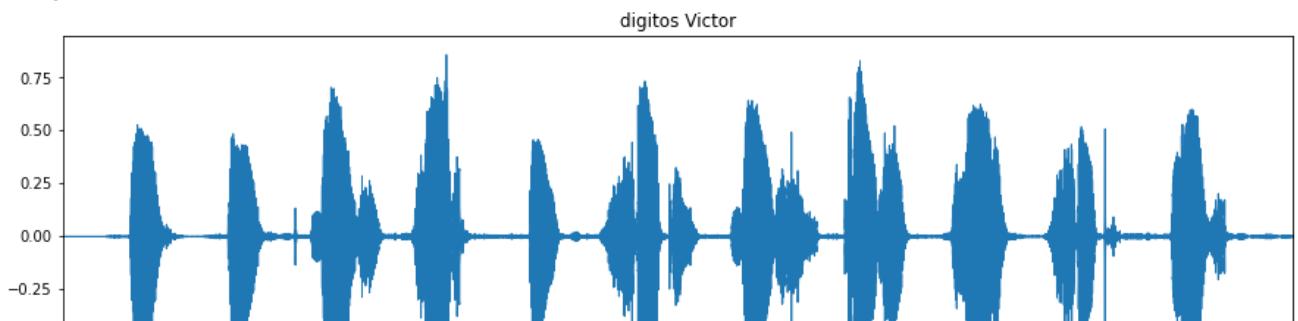
<Figure size 432x288 with 0 Axes>



<Figure size 432x288 with 0 Axes>



<Figure size 432x288 with 0 Axes>



#inputs

```

v_temp=v_digitos_lucas
sr_temp=sr_digitos_lucas
title='digitos Lucas'
#construcao da energia de voz
fa = sr_temp
Ts = 0.04 # Tempo de duração do segmento em segundos
Nj = int(Ts*fa) # Número de pontos da Janela
Nseg = int(len(v_temp)/Nj)
Nover = int(Nj*0.5)
from scipy import signal
hm = signal.get_window('hamming', Nj)

Nover = int(Nj*0.5)
Nseg = int(len(v_temp)/Nj)
E = []
for l in range(1, Nseg):
    xjan = v_temp[(l-1)*Nj+Nover:l*Nj+Nover]*hm
    x2 = list(np.array(xjan**2))
    aux = sum(x2)/Nj
    E.append(aux)
E = 10*np.log10(E)
Emin = np.min(E) # calcula nível de ruído de fundo
plt.figure
fig, ax = plt.subplots(figsize=(15, 3))
plt.plot(E - Emin)
plt.title('Energia da Voz ' + title)
plt.ylabel('Energia[db]'); plt.xlabel('Segmento')
plt.show()

```

```

#inputs
v_temp=v_digitos_muriel
sr_temp=sr_digitos_muriel
title='digitos muriel'
#construcao da energia de voz
fa = sr_temp
Ts = 0.04 # Tempo de duração do segmento em segundos
Nj = int(Ts*fa) # Número de pontos da Janela
Nseg = int(len(v_temp)/Nj)
Nover = int(Nj*0.5)
from scipy import signal
hm = signal.get_window('hamming', Nj)

Nover = int(Nj*0.5)
Nseg = int(len(v_temp)/Nj)
E = []
for l in range(1, Nseg):
    xjan = v_temp[(l-1)*Nj+Nover:l*Nj+Nover]*hm
    x2 = list(np.array(xjan**2))
    aux = sum(x2)/Nj
    E.append(aux)
E = 10*np.log10(E)
Emin = np.min(E) # calcula nível de ruído de fundo
plt.figure
fig, ax = plt.subplots(figsize=(15, 3))

```

```

plt.plot(E - Emin)
plt.title('Energia da Voz ' + title)
plt.ylabel('Energia[db]'); plt.xlabel('Segmento')
plt.show()

#inputs
v_temp=v_digitos_jefferson
sr_temp=sr_digitos_jefferson
title='digitos jefferson'
#construcao da energia de voz
fa = sr_temp
Ts = 0.04 # Tempo de duração do segmento em segundos
Nj = int(Ts*fa) # Número de pontos da Janela
Nseg = int(len(v_temp)/Nj)
Nover = int(Nj*0.5)
from scipy import signal
hm = signal.get_window('hamming', Nj)

Nover = int(Nj*0.5)
Nseg = int(len(v_temp)/Nj)
E = []
for l in range(1, Nseg):
    xjan = v_temp[(l-1)*Nj+Nover:l*Nj+Nover]*hm
    x2 = list(np.array(xjan)**2)
    aux = sum(x2)/Nj
    E.append(aux)
E = 10*np.log10(E)
Emin = np.min(E) # calcula nível de ruído de fundo
plt.figure
fig, ax = plt.subplots(figsize=(15, 3))
plt.plot(E - Emin)
plt.title('Energia da Voz ' + title)
plt.ylabel('Energia[db]'); plt.xlabel('Segmento')
plt.show()

#inputs
v_temp=v_digitos_victor
sr_temp=sr_digitos_victor
title='digitos victor'
#construcao da energia de voz
fa = sr_temp
Ts = 0.04 # Tempo de duração do segmento em segundos
Nj = int(Ts*fa) # Número de pontos da Janela
Nseg = int(len(v_temp)/Nj)
Nover = int(Nj*0.5)
from scipy import signal
hm = signal.get_window('hamming', Nj)

Nover = int(Nj*0.5)
Nseg = int(len(v_temp)/Nj)
E = []
for l in range(1, Nseg):
    xjan = v_temp[(l-1)*Nj+Nover:l*Nj+Nover]*hm
    x2 = list(np.array(xjan)**2))

```

```
aux = sum(x2)/Nj
E.append(aux)
E = 10*np.log10(E)
Emin = np.min(E) # calcula nível de ruído de fundo
plt.figure
fig, ax = plt.subplots(figsize=(15, 3))
plt.plot(E - Emin)
plt.title('Energia da Voz ' + title)
plt.ylabel('Energia[db]'); plt.xlabel('Segmento')
plt.show()
```



```
#inputs
v_temp=v_digitos_lucas
sr_temp=sr_digitos_lucas
title='digitos Lucas'

plt.figure(figsize=(12, 8))
D = librosa.amplitude_to_db(np.abs(librosa.stft(v_temp)), ref=np.max)
fig, ax = plt.subplots(figsize=(15, 10))
librosa.display.specshow(D, x_axis='time',y_axis='linear')
plt.colorbar(format='%+2.0f dB')
plt.title('Potência e Espectrograma Linear na Frequênci' + title)
plt.show()
```

```
#inputs
v_temp=v_digitos_muriel
sr_temp=sr_digitos_muriel
title='digitos muriel'

plt.figure(figsize=(12, 8))
D = librosa.amplitude_to_db(np.abs(librosa.stft(v_temp)), ref=np.max)
fig, ax = plt.subplots(figsize=(15, 10))
librosa.display.specshow(D, x_axis='time',y_axis='linear')
plt.colorbar(format='%+2.0f dB')
plt.title('Potência e Espectrograma Linear na Frequênci' + title)
plt.show()
```

```
#inputs
v_temp=v_digitos_jefferson
sr_temp=sr_digitos_jefferson
title='digitos jefferson'

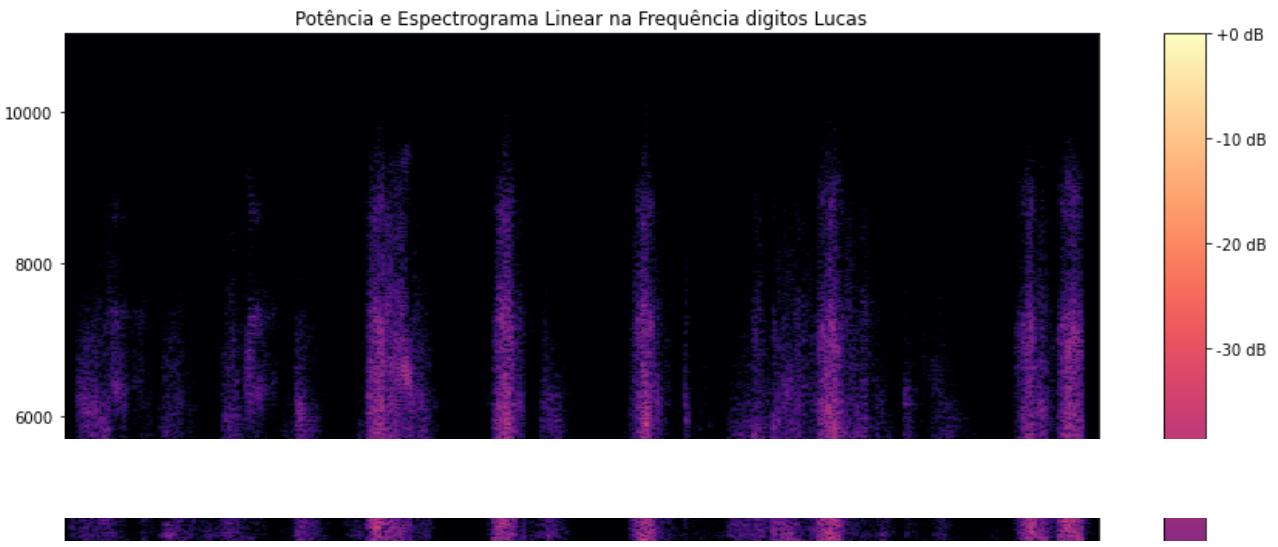
plt.figure(figsize=(12, 8))
D = librosa.amplitude_to_db(np.abs(librosa.stft(v_temp)), ref=np.max)
fig, ax = plt.subplots(figsize=(15, 10))
librosa.display.specshow(D, x_axis='time',y_axis='linear')
plt.colorbar(format='%+2.0f dB')
plt.title('Potência e Espectrograma Linear na Frequênci' + title)
plt.show()
```

```
#inputs
v_temp=v_digitos_victor
sr_temp=sr_digitos_victor
```

```
title='digitos victor'

plt.figure(figsize=(12, 8))
D = librosa.amplitude_to_db(np.abs(librosa.stft(v_temp)), ref=np.max)
fig, ax = plt.subplots(figsize=(15, 10))
librosa.display.specshow(D, x_axis='time',y_axis='linear')
plt.colorbar(format='%+2.0f dB')
plt.title('Potência e Espectrograma Linear na Frequênci' + title)
plt.show()
```

<Figure size 864x576 with 0 Axes>



5. Determinação do Pitch e da F0

► 5.1 Pitch do Arquivo Teste1

[] ↳ 2 células ocultas



O Período de Pitch (T_0) será o intervalo entre picos sucessivos.

A Frequência Fundamental (f_0) será o inverso do período de Pitch



► 5.2 Pitch do Arquivo Teste2

[] ↳ 2 células ocultas



Determine o Período de Pitch e a Fundamental desta vogal.

► Agora com cada um dos seus arquivos:

Determine o Pitch e a f_0 das vogais que desejarem (três vogais diferentes) dos seus arquivos, para as versões:

1. Dígitos
2. Falada
3. Declamada

[] ↳ 4 células ocultas

Time

► 5.3 Determinação das Formantes

[] ↳ 6 células ocultas

► 6. Determinação de Consoantes e Plosivos

↳ 1 célula oculta

▼ 7. RELATÓRIO

- A áudio descrição pode ser mais simplificada, pois neste Lab são muitos os arquivos de áudio.
- Na página do Laboratório (LEMBREM-SE que os notebooks são ANEXOS, e não páginas html), incluir os arquivos de voz de cada um dos integrantes.
- Fazer uma tabela com todos os valores de pitch e f_0 de todos os integrantes, de acordo com o item 5.2., e comparem-nas.
- Escolher um conjunto de formantes (f_1 a f_4) de cada integrante, para a mesma vogal, e comparem-nas.
- Fazer uma seção de **Comentários e Conclusões** incluindo a análise dos espectrogramas das consoantes.

▼ Conclusão

Os experimentos consolidam os conhecimentos abordados na disciplina de CSM, como o comportamento da frequência fundamental do espectrograma, a menor componente do sinal de fala, observa-se a diferença na entonação dos membros do grupo, mesmo com texto semelhantes a diferença de entonação e de microfone causa observáveis. Dentre os integrantes, o Muriel apresentou a menor frequência de F0 em 526 Hz, correspondendo a voz mais grave do grupo, enquanto o Lucas apresentou a voz mais aguda dentre os demais chegando até 3333 Hz.