UFABC - CSM - 3QS2021 - Minami

Lab9 - MIDI e IR

Objetivos

1. Tocar notas e gerar músicas simples
2. Gerar arquivos MIDI
3. Reproduzir arquivos MIDI
4. Simular a reprodução em salas com Respostas Impulsivas diversas

Clique duas vezes (ou pressione "Enter") para editar

1 - Tocar notas e gerar músicas simples

- Instalando Magenta

```
print('Installing dependencies...')
!apt-get update -qq && apt-get install -qq libfluidsynth1 fluid-soundfont-gm build-
!pip install -qU pyfluidsynth pretty_midi

!pip install -qU magenta

# Hack to allow python to pick up the newly-installed fluidsynth lib.
# This is only needed for the hosted Colab environment.
import ctypes.util
orig_ctypes_util_find_library = ctypes.util.find_library
def proxy_find_library(lib):
  if lib == 'fluidsynth':
    return 'libfluidsynth.so.1'
  else:
    return orig_ctypes_util_find_library(lib)
ctypes.util.find_library = proxy_find_library

print('Importing libraries and defining some helper functions...')
from google.colab import files

import magenta
import note_seq


print('🎉 Done!')
print(magenta.__version__)
```

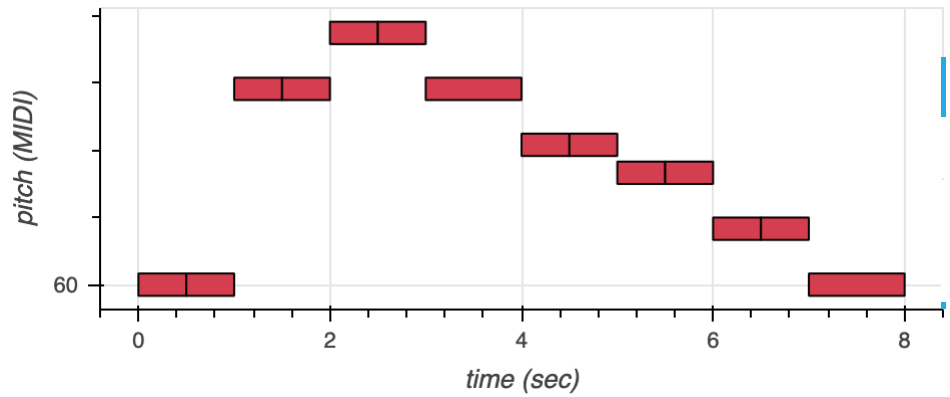
```
Installing dependencies...
Selecting previously unselected package fluid-soundfont-gm.
(Reading database ... 155222 files and directories currently installed.)
Preparing to unpack .../fluid-soundfont-gm_3.1-5.1_all.deb ...
Unpacking fluid-soundfont-gm (3.1-5.1) ...
Selecting previously unselected package libfluidsynth1:amd64.
Preparing to unpack .../libfluidsynth1_1.1.9-1_amd64.deb ...
Unpacking libfluidsynth1:amd64 (1.1.9-1) ...
Setting up fluid-soundfont-gm (3.1-5.1) ...
Setting up libfluidsynth1:amd64 (1.1.9-1) ...
Processing triggers for libc-bin (2.27-3ubuntu1.3) ...
/sbin/ldconfig.real: /usr/local/lib/python3.7/dist-packages/ideep4py/lib/libmk

     |████████████████████████████████| 5.6 MB 6.3 MB/s
     |████████████████████████████████| 51 kB 7.0 MB/s
  Building wheel for pretty-midi (setup.py) ... done
     |████████████████████████████████| 1.4 MB 7.8 MB/s
     |████████████████████████████████| 1.4 MB 31.8 MB/s
     |████████████████████████████████| 69 kB 7.2 MB/s
     |████████████████████████████████| 2.3 MB 65.0 MB/s
     |████████████████████████████████| 254 kB 44.2 MB/s
     |████████████████████████████████| 1.6 MB 50.4 MB/s
     |████████████████████████████████| 352 kB 50.7 MB/s
     |████████████████████████████████| 3.6 MB 31.5 MB/s
     |████████████████████████████████| 210 kB 69.3 MB/s
     |████████████████████████████████| 87 kB 7.7 MB/s
     |████████████████████████████████| 204 kB 62.9 MB/s
     |████████████████████████████████| 20.2 MB 1.5 MB/s
     |████████████████████████████████| 48 kB 4.9 MB/s
     |████████████████████████████████| 79 kB 7.9 MB/s
     |████████████████████████████████| 191 kB 74.3 MB/s
     |████████████████████████████████| 5.8 MB 16.7 MB/s
     |████████████████████████████████| 367 kB 44.5 MB/s
     |████████████████████████████████| 1.1 MB 42.9 MB/s
     |████████████████████████████████| 981 kB 46.7 MB/s
     |████████████████████████████████| 366 kB 57.3 MB/s
     |████████████████████████████████| 251 kB 38.9 MB/s
     |████████████████████████████████| 191 kB 71.2 MB/s
     |████████████████████████████████| 178 kB 69.8 MB/s
  Building wheel for librosa (setup.py) ... done
  Building wheel for mir-eval (setup.py) ... done
  Building wheel for pygtrie (setup.py) ... done
  Building wheel for python-rtmidi (setup.py) ... done
  Building wheel for bz2file (setup.py) ... done
Importing libraries and defining some helper functions...
/usr/local/lib/python3.7/dist-packages/librosa/util/decorators.py:9: NumbaDepr
Import requested from: 'numba.decorators', please update to use 'numba.core.de
  from numba.decorators import jit as optional_jit
/usr/local/lib/python3.7/dist-packages/librosa/util/decorators.py:9: NumbaDepr
Import of 'jit' requested from: 'numba.decorators', please update to use 'numb
  from numba.decorators import jit as optional_jit
🎉 Done!
2.1.3
```

Gerando TONS com NoteSequences

Everything in `Magenta` is centered around NoteSequences. This is an abstract representation of a series of notes, each with different pitches, instruments and strike velocities, much like MIDI.

For example, this is a `NoteSequence` that represents "Twinkle Twinkle Little Star".

```
from note_seq.protobuf import music_pb2

twinkle_twinkle = music_pb2.NoteSequence()

# Add the notes to the sequence.
twinkle_twinkle.notes.add(pitch=60, start_time=0.0, end_time=0.5, velocity=80)
twinkle_twinkle.notes.add(pitch=60, start_time=0.5, end_time=1.0, velocity=80)
twinkle_twinkle.notes.add(pitch=67, start_time=1.0, end_time=1.5, velocity=80)
twinkle_twinkle.notes.add(pitch=67, start_time=1.5, end_time=2.0, velocity=80)
twinkle_twinkle.notes.add(pitch=69, start_time=2.0, end_time=2.5, velocity=80)
twinkle_twinkle.notes.add(pitch=69, start_time=2.5, end_time=3.0, velocity=80)
twinkle_twinkle.notes.add(pitch=67, start_time=3.0, end_time=4.0, velocity=80)
twinkle_twinkle.notes.add(pitch=65, start_time=4.0, end_time=4.5, velocity=80)
twinkle_twinkle.notes.add(pitch=65, start_time=4.5, end_time=5.0, velocity=80)
twinkle_twinkle.notes.add(pitch=64, start_time=5.0, end_time=5.5, velocity=80)
twinkle_twinkle.notes.add(pitch=64, start_time=5.5, end_time=6.0, velocity=80)
twinkle_twinkle.notes.add(pitch=62, start_time=6.0, end_time=6.5, velocity=80)
twinkle_twinkle.notes.add(pitch=62, start_time=6.5, end_time=7.0, velocity=80)
twinkle_twinkle.notes.add(pitch=60, start_time=7.0, end_time=8.0, velocity=80)
twinkle_twinkle.total_time = 8

twinkle_twinkle.tempos.add(qpm=60);

# This is a colab utility method that visualizes a NoteSequence.
note_seq.plot_sequence(twinkle_twinkle)

# This is a colab utility method that plays a NoteSequence.
note_seq.play_sequence(twinkle_twinkle,synth=note_seq.fluidsynth)
```

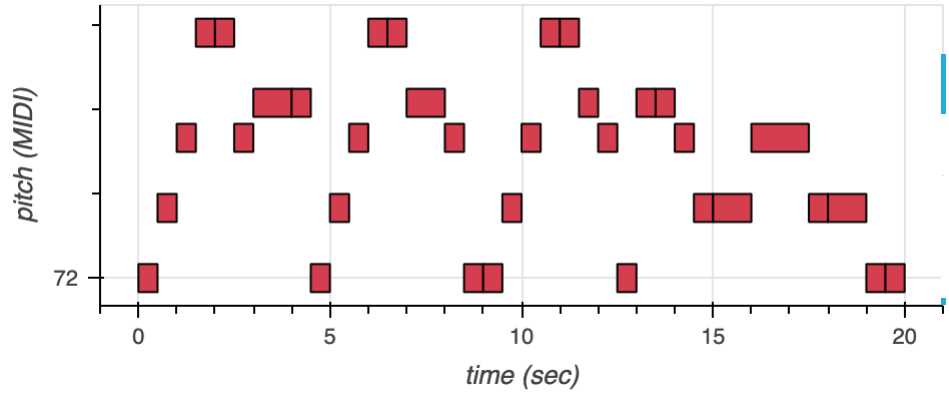


Pode variar os tempos e as notas

Agora com ASA BRANCA

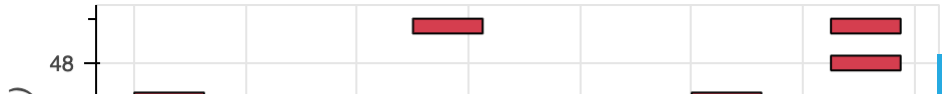
Há um pequeno erro na melodia, procure consertá-lo!

```
# Asa Branca

asa_branca_piano = music_pb2.NoteSequence()

# CifraClub
asa_branca_piano.notes.add(pitch=72, start_time=0.0, end_time=0.5, velocity=80)
asa_branca_piano.notes.add(pitch=74, start_time=0.5, end_time=1.0, velocity=80)
asa_branca_piano.notes.add(pitch=76, start_time=1.0, end_time=1.5, velocity=80)
asa_branca_piano.notes.add(pitch=79, start_time=1.5, end_time=2.0, velocity=80)
asa_branca_piano.notes.add(pitch=79, start_time=2.0, end_time=2.5, velocity=80)
asa_branca_piano.notes.add(pitch=76, start_time=2.5, end_time=3.0, velocity=80)
asa_branca_piano.notes.add(pitch=77, start_time=3.0, end_time=4.0, velocity=80)
asa_branca_piano.notes.add(pitch=77, start_time=4.0, end_time=4.5, velocity=80)
asa_branca_piano.notes.add(pitch=72, start_time=4.5, end_time=5.0, velocity=80)
asa_branca_piano.notes.add(pitch=74, start_time=5.0, end_time=5.5, velocity=80)
asa_branca_piano.notes.add(pitch=76, start_time=5.5, end_time=6.0, velocity=80)
asa_branca_piano.notes.add(pitch=79, start_time=6.0, end_time=6.5, velocity=80)
asa_branca_piano.notes.add(pitch=79, start_time=6.5, end_time=7.0, velocity=80)
asa_branca_piano.notes.add(pitch=77, start_time=7.0, end_time=8.0, velocity=80)
asa_branca_piano.notes.add(pitch=76, start_time=8.0, end_time=8.5, velocity=80)
asa_branca_piano.notes.add(pitch=72, start_time=8.5, end_time=9.0, velocity=80)
asa_branca_piano.notes.add(pitch=72, start_time=9.0, end_time=9.5, velocity=80)
asa_branca_piano.notes.add(pitch=74, start_time=9.5, end_time=10.0, velocity=80)

asa_branca_piano.notes.add(pitch=76, start_time=10.0, end_time=10.5, velocity=80)
asa_branca_piano.notes.add(pitch=79, start_time=10.5, end_time=11.0, velocity=80)
asa_branca_piano.notes.add(pitch=79, start_time=11.0, end_time=11.5, velocity=80)
asa_branca_piano.notes.add(pitch=77, start_time=11.5, end_time=12.0, velocity=80)
asa_branca_piano.notes.add(pitch=76, start_time=12.0, end_time=12.5, velocity=80)
asa_branca_piano.notes.add(pitch=72, start_time=12.5, end_time=13.0, velocity=80)
asa_branca_piano.notes.add(pitch=77, start_time=13.0, end_time=13.5, velocity=80)
asa_branca_piano.notes.add(pitch=77, start_time=13.5, end_time=14.0, velocity=80)
asa_branca_piano.notes.add(pitch=76, start_time=14.0, end_time=14.5, velocity=80)
asa_branca_piano.notes.add(pitch=74, start_time=14.5, end_time=15.0, velocity=80)
asa_branca_piano.notes.add(pitch=74, start_time=15.0, end_time=16.0, velocity=80)
asa_branca_piano.notes.add(pitch=76, start_time=16.0, end_time=17.5, velocity=80)
asa_branca_piano.notes.add(pitch=74, start_time=17.5, end_time=18.0, velocity=80)
asa_branca_piano.notes.add(pitch=74, start_time=18.0, end_time=19.0, velocity=80)
asa_branca_piano.notes.add(pitch=72, start_time=19.0, end_time=19.5, velocity=80)
asa_branca_piano.notes.add(pitch=72, start_time=19.5, end_time=20.0, velocity=80)
asa_branca_piano.total_time = 20.0

asa_branca_piano.tempos.add(qpm=60);

# This is a colab utility method that visualizes a NoteSequence.
note_seq.plot_sequence(asa_branca_piano)

# This is a colab utility method that plays a NoteSequence.
note_seq.play_sequence(asa_branca_piano,synth=note_seq.fluidsynth)
```



Pode usar outro instrumento. Por exemple, uma bateria

```
drums = music_pb2.NoteSequence()

drums.notes.add(pitch=36, start_time=0, end_time=0.125, is_drum=True, instrument=10
drums.notes.add(pitch=38, start_time=0, end_time=0.125, is_drum=True, instrument=10
drums.notes.add(pitch=42, start_time=0, end_time=0.125, is_drum=True, instrument=10
drums.notes.add(pitch=46, start_time=0, end_time=0.125, is_drum=True, instrument=10
drums.notes.add(pitch=42, start_time=0.25, end_time=0.375, is_drum=True, instrument
drums.notes.add(pitch=42, start_time=0.375, end_time=0.5, is_drum=True, instrument=
drums.notes.add(pitch=42, start_time=0.5, end_time=0.625, is_drum=True, instrument=
drums.notes.add(pitch=50, start_time=0.5, end_time=0.625, is_drum=True, instrument=
drums.notes.add(pitch=36, start_time=0.75, end_time=0.875, is_drum=True, instrument
drums.notes.add(pitch=38, start_time=0.75, end_time=0.875, is_drum=True, instrument
drums.notes.add(pitch=42, start_time=0.75, end_time=0.875, is_drum=True, instrument
drums.notes.add(pitch=45, start_time=0.75, end_time=0.875, is_drum=True, instrument
drums.notes.add(pitch=36, start_time=1, end_time=1.125, is_drum=True, instrument=10
drums.notes.add(pitch=42, start_time=1, end_time=1.125, is_drum=True, instrument=10
drums.notes.add(pitch=46, start_time=1, end_time=1.125, is_drum=True, instrument=10
drums.notes.add(pitch=42, start_time=1.25, end_time=1.375, is_drum=True, instrument
drums.notes.add(pitch=48, start_time=1.25, end_time=1.375, is_drum=True, instrument
drums.notes.add(pitch=50, start_time=1.25, end_time=1.375, is_drum=True, instrument
drums.total_time = 1.375

drums.tempos.add(qpm=60)

# This is a colab utility method that visualizes a NoteSequence.
note_seq.plot_sequence(drums)


# This is a colab utility method that plays a NoteSequence.
note_seq.play_sequence(drums,synth=note_seq.fluidsynth)
```



* Agora "Escreva" uma música simples, e toque-a

2. Gerando arquivos MIDI: `NoteSequence` -> MIDI

When you called the "play_sequence" method above, this converted the `NoteSequence` to MIDI, and created an HTML widget to play it. This method is specially made for colab notebooks, so it won't work inside your Python script. That method uses the Magenta built-in conversion methods, which you can use in your python script:

```
# This creates a file called `asa_branca_sample_output.mid`, containing the drums s
note_seq.sequence_proto_to_midi_file(asa_branca_piano, 'asa_branca_sample_output.mi

# This is a colab utility method to download that file. In your Python script, you
# would just write it to disk.
files.download('asa_branca_sample_output.mid')

# This creates a file called `drums_sample_output.mid`, containing the drums solo w
note_seq.sequence_proto_to_midi_file(drums, 'drums_sample_output.mid')

# This is a colab utility method to download that file. In your Python script, you
# would just write it to disk.
files.download('drums_sample_output.mid')
```

* Salve a sua "composição" musical

Useful helpers

There are a lot of other helper methods sprinkled around the `note_seq` codebase that you might need but not know where to find. Here are some of our favourites:

- converting between MIDI and NoteSequences
- trimming, concatenating and expanding NoteSequences
- colab notebook utils

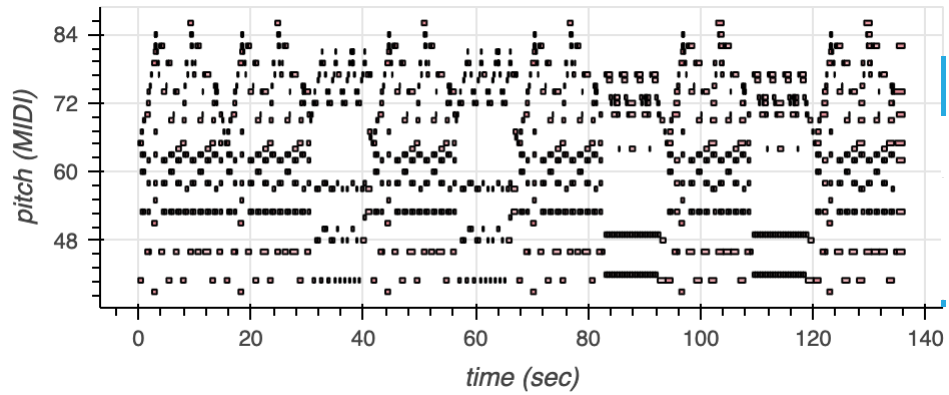
3. Reproduzindo Arquivos MIDI

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

```
Mounted at /content/drive
```

Carregando MIDI

```
midi_file = note_seq.midi_file_to_note_sequence('/content/drive/MyDrive/lab09_csm/c
# Handle sustain pedal in midi_file
midi_file = note_seq.apply_sustain_control_changes(midi_file)
note_seq.play_sequence(
    midi_file,
    synth=note_seq.fluidsynth, sample_rate=22050)
note_seq.plot_sequence(midi_file)
```



4. Tocando a música em alguns ambientes

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
import librosa
import librosa.display
import IPython.display
```

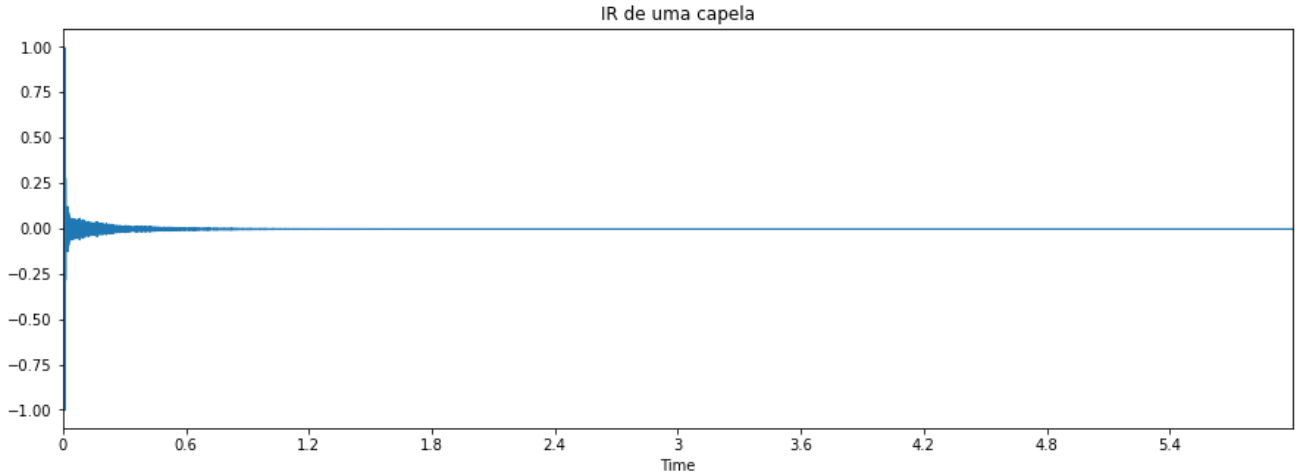
```
# Resposta Impulsiva de uma capela de Igreja
ir_chapel, sr_chapel = librosa.load('/content/drive/MyDrive/lab09_csm/stalbans_omni
```

```
# Normalização da IR
ir_chapel = librosa.util.normalize(ir_chapel)
```

```
IPython.display.Audio(ir_chapel, rate = sr_chapel)
```

```
plt.figure()
fig, ax = plt.subplots(figsize=(15, 5))
librosa.display.waveplot(ir_chapel, sr=sr_chapel)
plt.title('IR de uma capela')
```

```
Text(0.5, 1.0, 'IR de uma capela')
<Figure size 432x288 with 0 Axes>
```

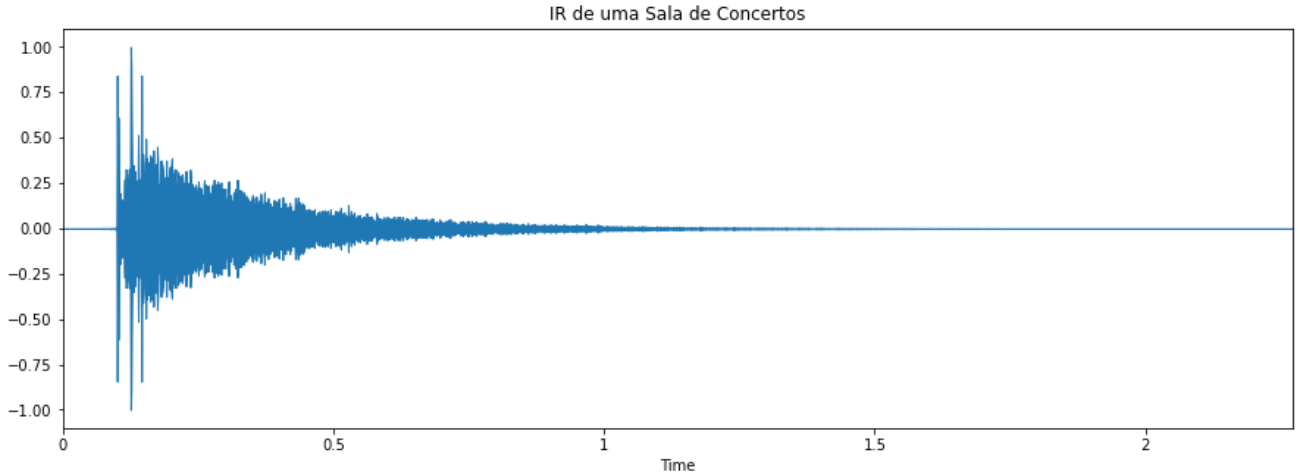



```
# "Tocando" Chopin na Capela

Mazurca, sr_Mazurca = librosa.load('/content/drive/MyDrive/lab09_csm/Mazurca_Chopin

Mazurca = librosa.util.normalize(Mazurca)

reproducao_Chopin_chapel = np.convolve(Mazurca, ir_chapel, mode = 'full')


IPython.display.Audio(reproducao_Chopin_chapel, rate = sr_Mazurca)
```

0:00 / 2:43

```
# "Tocando" numa Sala de Concertos
ir_hall, sr_hall = librosa.load('/content/drive/MyDrive/lab09_csm/usina_main_s1_p5.
ir_hall = librosa.util.normalize(ir_hall)
plt.figure()
fig, ax = plt.subplots(figsize=(15, 5))
librosa.display.waveplot(ir_hall, sr=sr_hall)
plt.title('IR de uma Sala de Concertos')
```

```
Text(0.5, 1.0, 'IR de uma Sala de Concertos')
<Figure size 432x288 with 0 Axes>
```




```
IPython.display.Audio(ir_hall, rate = sr_hall)
```

```
reproducao_Chopin_hall = np.convolve(Mazurca, ir_hall, mode = 'full')
```

```
IPython.display.Audio(reproducao_Chopin_hall, rate = sr_Mazurca)
```

* Reproduza a sua música neste dois ambientes diferentes

5. Relatório

a) Apresentem as músicas de todos os integrantes do grupo, identificando quem "criou" cada uma;

b) Pesquise numa base de dados aberta como: https://www.openair.hosted.york.ac.uk/?page_id=36, e encontre três IRs de ambientes diferentes (pequeno-fechado, auditório médio, igreja grande);

c) "Criar" mais uma música (de maior duração) e "reproduzi-la" em cada ambiente, comentando as diferenças;

d) Repita o mesmo procedimento (c) para um outro .midi que desejar (da web).

e) Usando algum programa que lê partituras midi, por exemplo "Finale Notepad" ou "MuseScore", faça uma das duas opções:

i) leia uma música com pelo menos quatro vozes, e troque um instrumento qualquer, gere o arquivo midi com o instrumento "trocado" e coloque as duas músicas no site

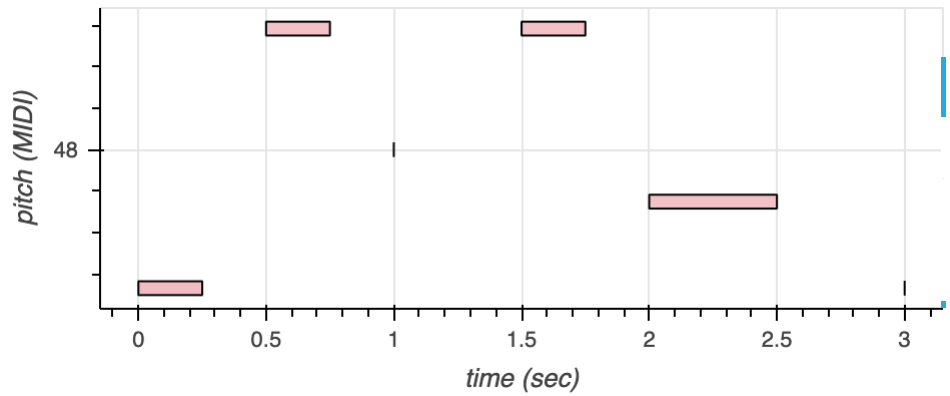
ii) ou faça um acompanhamento (Loops) de batidas original

-X-X-X-

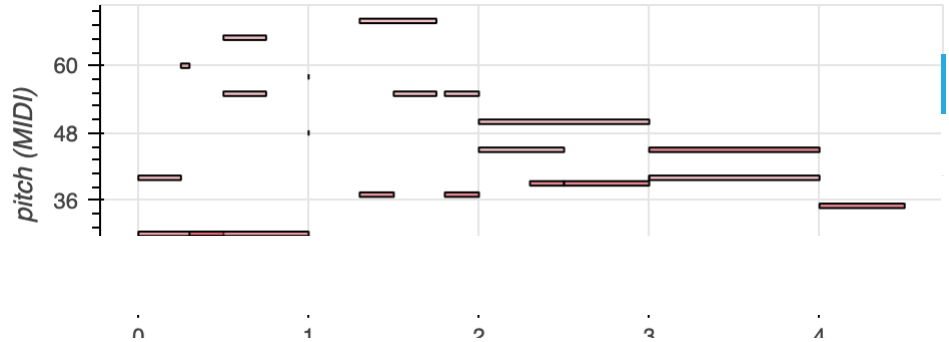
```
#musica jefferson

jefferson_song = music_pb2.NoteSequence()

# Add the notes to the sequence.
jefferson_song.notes.add(pitch=30, start_time=0.0, end_time=0.5, velocity=65)
jefferson_song.notes.add(pitch=30, start_time=0.3, end_time=1.0, velocity=70)
jefferson_song.notes.add(pitch=37, start_time=1.3, end_time=1.5, velocity=70)
jefferson_song.notes.add(pitch=37, start_time=1.8, end_time=2.0, velocity=80)
jefferson_song.notes.add(pitch=39, start_time=2.3, end_time=2.5, velocity=80)
jefferson_song.notes.add(pitch=39, start_time=2.5, end_time=3.0, velocity=80)
jefferson_song.notes.add(pitch=45, start_time=3.0, end_time=4.0, velocity=80)
jefferson_song.notes.add(pitch=35, start_time=4.0, end_time=4.5, velocity=80)
jefferson_song.total_time = 8

jefferson_song.tempos.add(qpm=60);

# This is a colab utility method that visualizes a NoteSequence.
note_seq.plot_sequence(jefferson_song)

# This is a colab utility method that plays a NoteSequence.
note_seq.play_sequence(jefferson_song,synth=note_seq.fluidsynth)



#musica muriel

muriel_song = music_pb2.NoteSequence()

# Add the notes to the sequence.
muriel_song.notes.add(pitch=20, start_time=0.0, end_time=0.25, velocity=45)
muriel_song.notes.add(pitch=25, start_time=0.3, end_time=0.75, velocity=40)
```

```
muriel_song.notes.add(pitch=28, start_time=1.3, end_time=1, velocity=40)
muriel_song.notes.add(pitch=25, start_time=1.8, end_time=1.75, velocity=40)
muriel_song.notes.add(pitch=25, start_time=2.3, end_time=2, velocity=40)
muriel_song.notes.add(pitch=28, start_time=2.5, end_time=2.2, velocity=40)
muriel_song.notes.add(pitch=28, start_time=3.0, end_time=2.8, velocity=40)
muriel_song.notes.add(pitch=40, start_time=4.0, end_time=3, velocity=40)
muriel_song.total_time = 8

muriel_song.tempos.add(qpm=60);

# This is a colab utility method that visualizes a NoteSequence.
note_seq.plot_sequence(muriel_song)

# This is a colab utility method that plays a NoteSequence.
note_seq.play_sequence(muriel_song,synth=note_seq.fluidsynth)




#musica lucas

lucas_song = music_pb2.NoteSequence()

# Add the notes to the sequence.
lucas_song.notes.add(pitch=60, start_time=0.3, end_time=0.25, velocity=45)
lucas_song.notes.add(pitch=65, start_time=0.5, end_time=0.75, velocity=40)
lucas_song.notes.add(pitch=58, start_time=1.0, end_time=1, velocity=40)
lucas_song.notes.add(pitch=68, start_time=1.3, end_time=1.75, velocity=40)
lucas_song.notes.add(pitch=55, start_time=1.8, end_time=2, velocity=40)
lucas_song.notes.add(pitch=50, start_time=2.0, end_time=3, velocity=40)
lucas_song.total_time = 8

lucas_song.tempos.add(qpm=60);

# This is a colab utility method that visualizes a NoteSequence.
note_seq.plot_sequence(lucas_song)

# This is a colab utility method that plays a NoteSequence.
note_seq.play_sequence(lucas_song,synth=note_seq.fluidsynth)




#musica victor

victor_song = music_pb2.NoteSequence()

# Add the notes to the sequence.
victor_song.notes.add(pitch=40, start_time=0.0, end_time=0.25, velocity=45)
victor_song.notes.add(pitch=55, start_time=0.5, end_time=0.75, velocity=40)
victor_song.notes.add(pitch=48, start_time=1.0, end_time=1, velocity=40)
victor_song.notes.add(pitch=55, start_time=1.5, end_time=1.75, velocity=40)
victor_song.notes.add(pitch=45, start_time=2.5, end_time=2, velocity=40)
victor_song.notes.add(pitch=40, start_time=3.0, end_time=3, velocity=40)
victor_song.total_time = 8
```

```
victor_song.tempos.add(qpm=60);

# This is a colab utility method that visualizes a NoteSequence.
note_seq.plot_sequence(victor_song)

# This is a colab utility method that plays a NoteSequence.
note_seq.play_sequence(victor_song,synth=note_seq.fluidsynth)
```

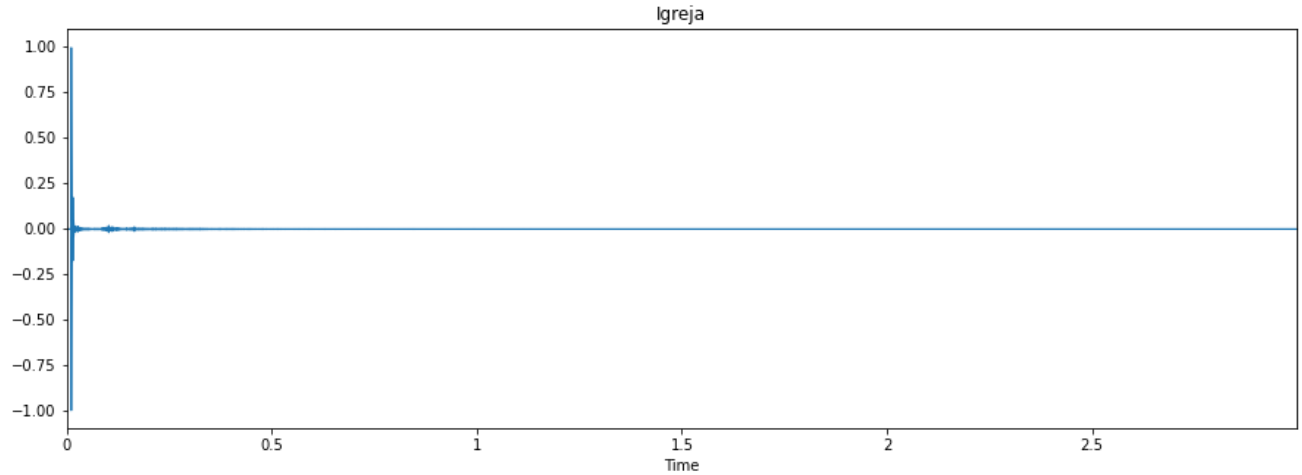



```
#juntando as musicas
composicao_grupo = music_pb2.NoteSequence()

composicao_grupo.MergeFrom(victor_song)
composicao_grupo.MergeFrom(lucas_song)
composicao_grupo.MergeFrom(jefferson_song)
composicao_grupo.MergeFrom(muriel_song)


note_seq.plot_sequence(composicao_grupo)


note_seq.play_sequence(composicao_grupo,synth=note_seq.fluidsynth)

#baixando as musicas
midi_file=note_seq.sequence_proto_to_midi_file(victor_song, 'victor_song.mid')
files.download('victor_song.mid')

midi_file=note_seq.sequence_proto_to_midi_file(lucas_song, 'lucas_song.mid')
files.download('lucas_song.mid')

midi_file=note_seq.sequence_proto_to_midi_file(jefferson_song, 'jefferson_song.mid'
files.download('jefferson_song.mid')

midi_file=note_seq.sequence_proto_to_midi_file(muriel_song, 'muriel_song.mid')
files.download('muriel_song.mid')
```



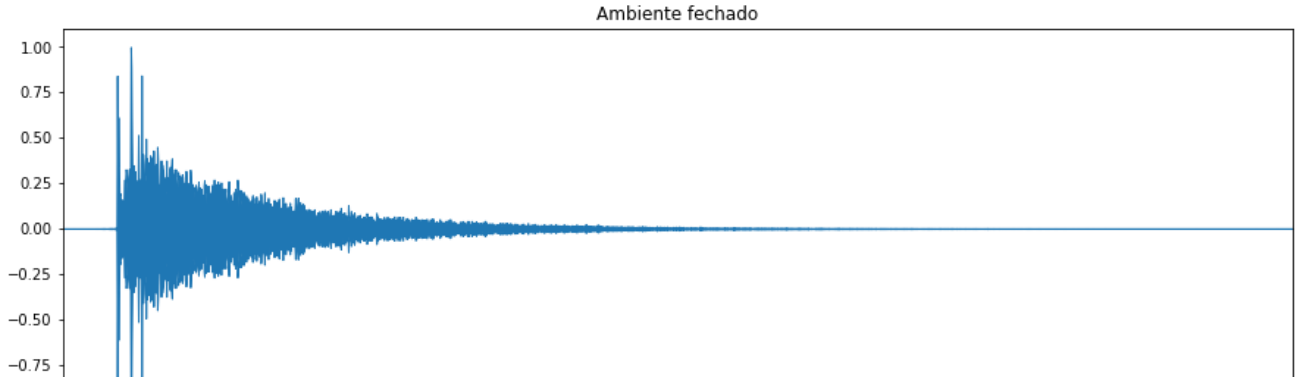

```
# "Tocando" numa Sala de Concertos
ir_igreja, sr_igreja = librosa.load('/content/drive/MyDrive/lab09_csm/1st_baptist_n
ir_hir_igrejaall = librosa.util.normalize(ir_igreja)
plt.figure()
fig, ax = plt.subplots(figsize=(15, 5))
librosa.display.waveplot(ir_igreja, sr=sr_igreja)
plt.title('Igreja')
```

```
Text(0.5, 1.0, 'Igreja')
<Figure size 432x288 with 0 Axes>
```




```
# "Tocando" numa Sala de Concertos
ir_fechado, sr_fechado = librosa.load('/content/drive/MyDrive/lab09_csm/usina_main_
ir_fechado = librosa.util.normalize(ir_fechado)
plt.figure()
fig, ax = plt.subplots(figsize=(15, 5))
librosa.display.waveplot(ir_fechado, sr=sr_fechado)
plt.title('Ambiente fechado')
```

```
Text(0.5, 1.0, 'Ambiente fechado')
<Figure size 432x288 with 0 Axes>
```




```
# "Tocando" numa Sala de Concertos
ir_hall, sr_hall = librosa.load('/content/drive/MyDrive/lab09_csm/ir_row_1l_sl_cent
ir_hall = librosa.util.normalize(ir_hall)
plt.figure()
fig, ax = plt.subplots(figsize=(15, 5))
librosa.display.waveplot(ir_hall, sr=sr_hall)
plt.title('Auditorio')
```

Clique duas vezes (ou pressione "Enter") para editar