

Wikimedia Commons

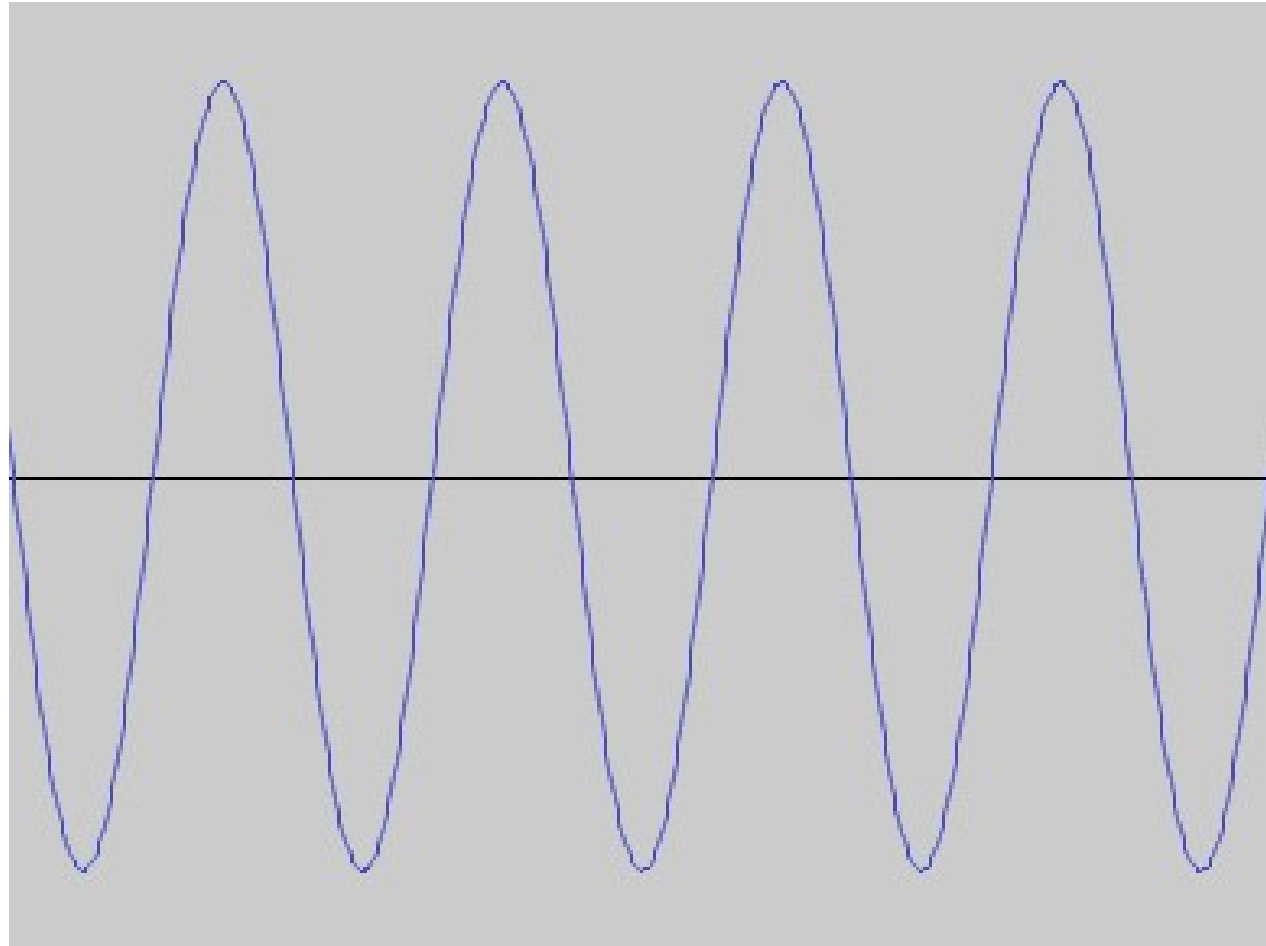
INFORMAÇÃO SONORA

Som



- Som é composto por ondas mecânicas
- Caracterizado por:
 - ▣ Frequência: o quanto grave ou agudo
 - ▣ Amplitude: o quanto alto ou baixo
- Um tom puro possui apenas uma frequência
 - ▣ Sons normalmente possuem várias

Tom 440 Hz (Amplitude vs Tempo)



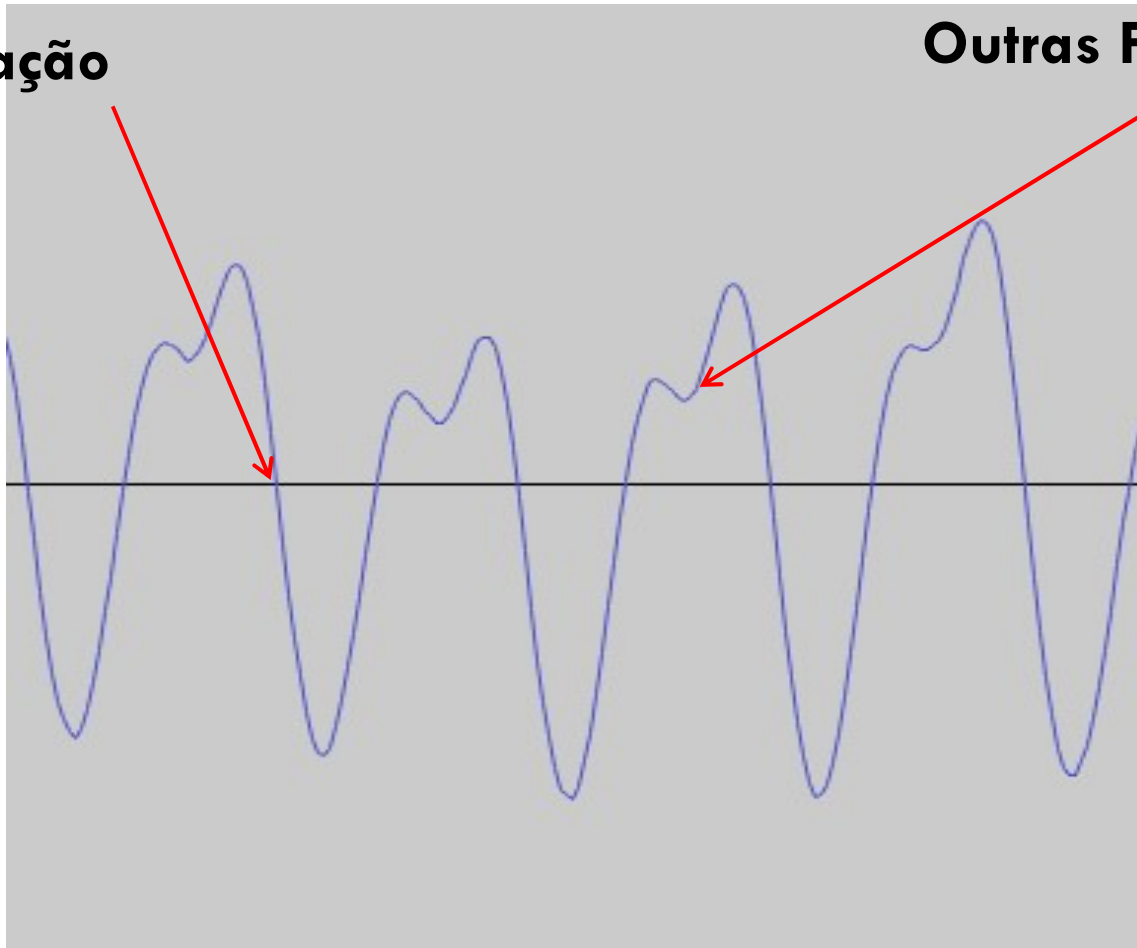
Tom 440 Hz (Frequência vs Tempo)



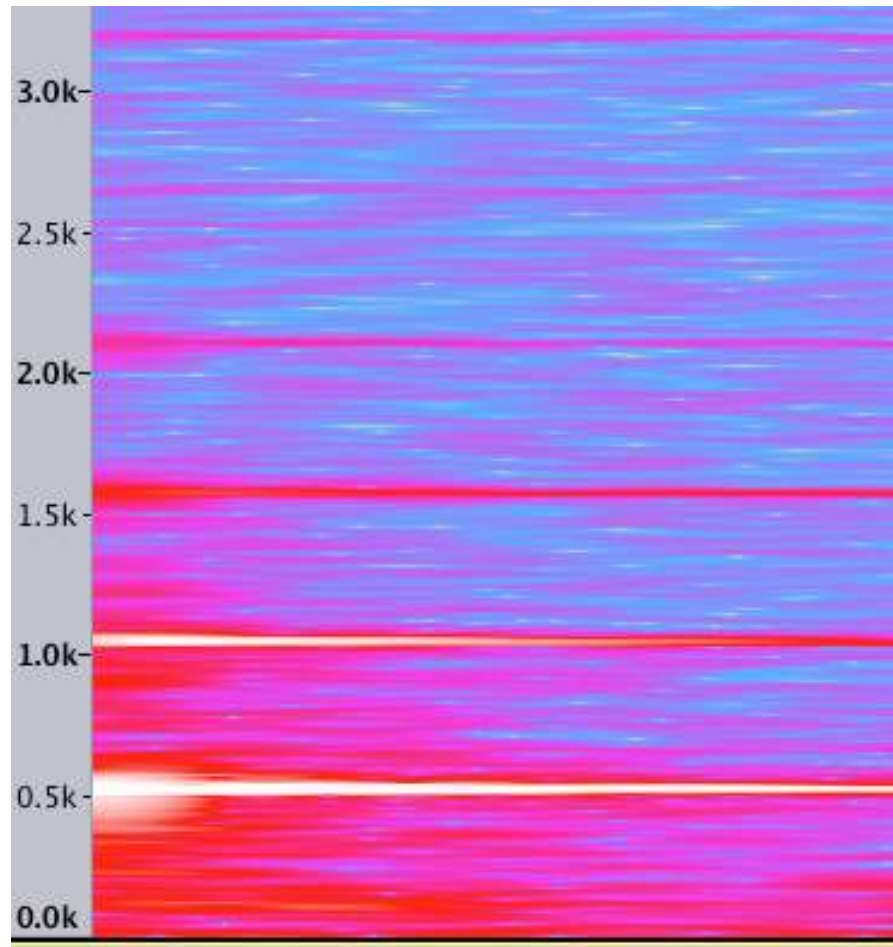
Piano (C5, 523 Hz)

Oscilação

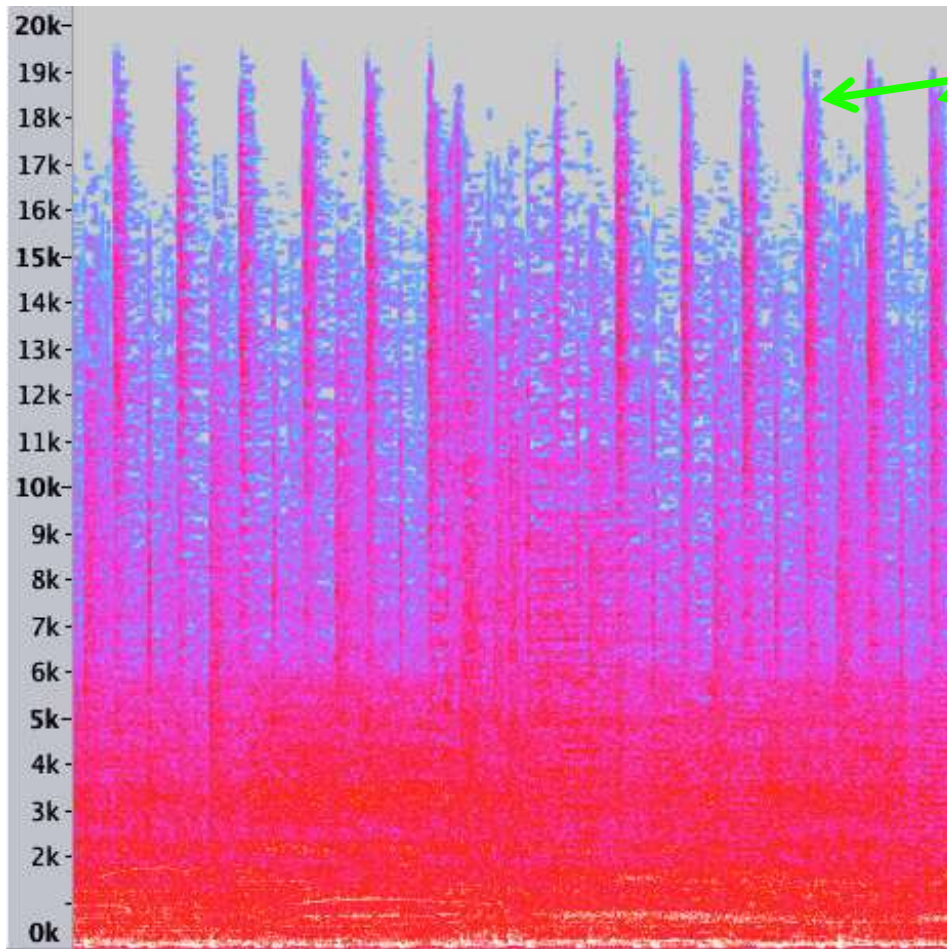
Outras Frequências



Piano (C5, 523 Hz)



Música



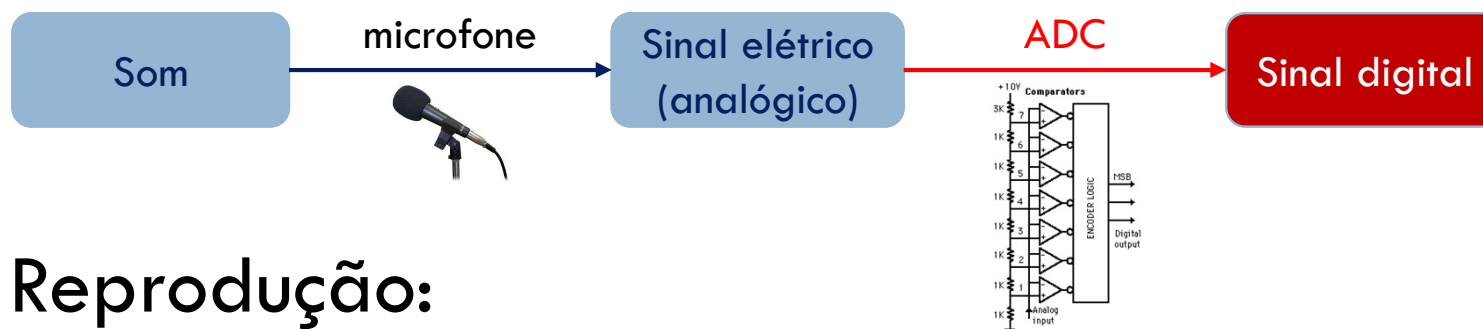
Ritmo da música

**Muito mais complexo!
Instrumentos, vozes,
efeitos, etc...**

ADC e DAC

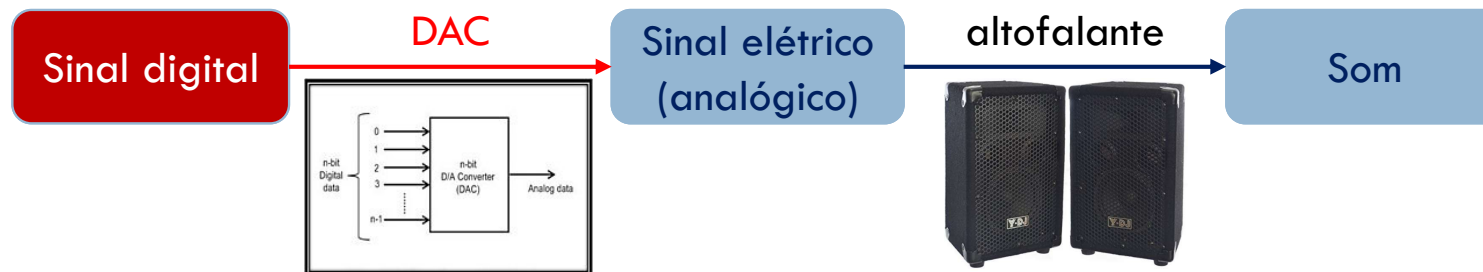
□ Aquisição:

▣ Analog to Digital Conversion (ADC)



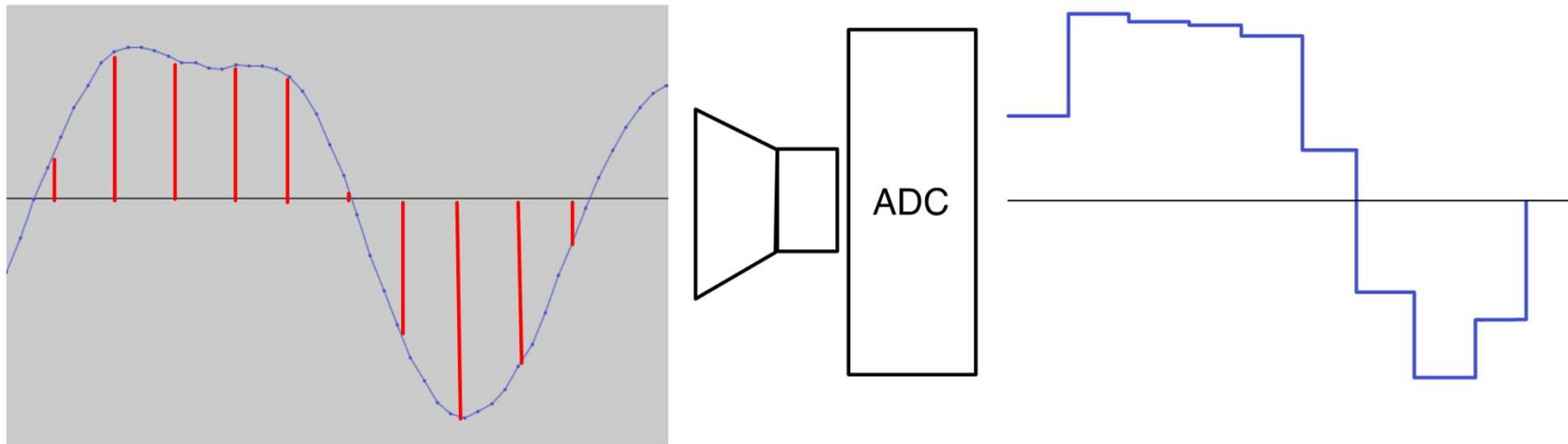
□ Reprodução:

▣ Digital to Analog Conversion (DAC)



Digitalização

- ADC amostra sinal a uma frequência específica
- Converte tensão medida para um valor
 - ▣ Processo não é perfeito...

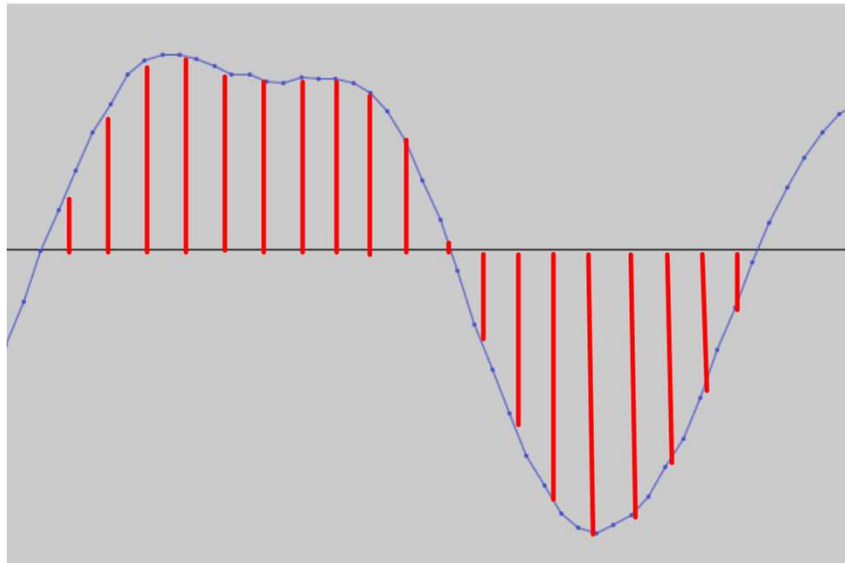


Frequência de Amostragem

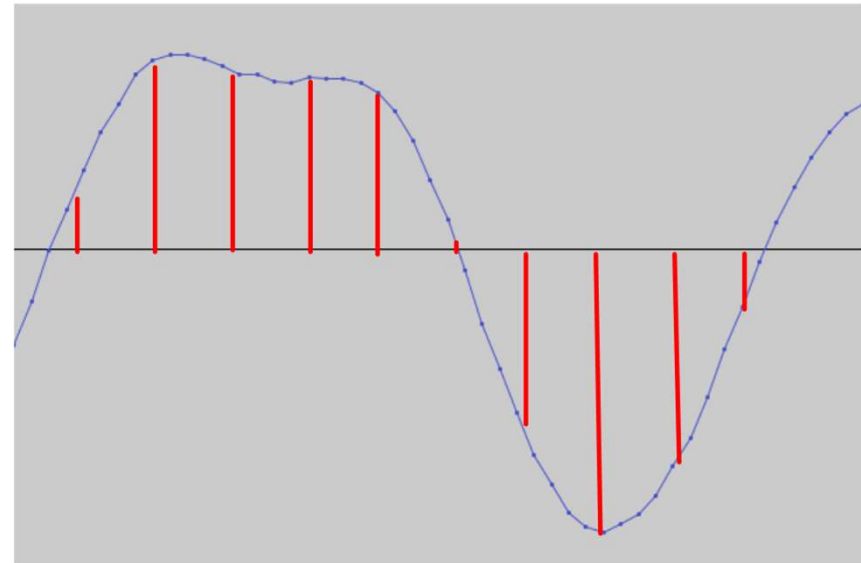
Quantas vezes por segundo se lê o valor de tensão

Maior frequência → melhor digitalização → maiores ficheiros

Maior Frequência



Menor Frequência



Frequência de Amostragem

- Deve ser o dobro da frequência do som a digitalizar
 - ▣ Teorema de Nyquist
 - ▣ no mínimo
- Ouvido humano distingue até 20 KHz
 - ▣ Digitalizar música: 44100 Hz, 48000 Hz
 - ▣ Gravação “normal” num PC
- Voz humana entre 1 KHz e 3 KHz
 - ▣ Digitalizar voz: 8 KHz
 - ▣ Comunicações por voz (telemóveis, skype, etc...)

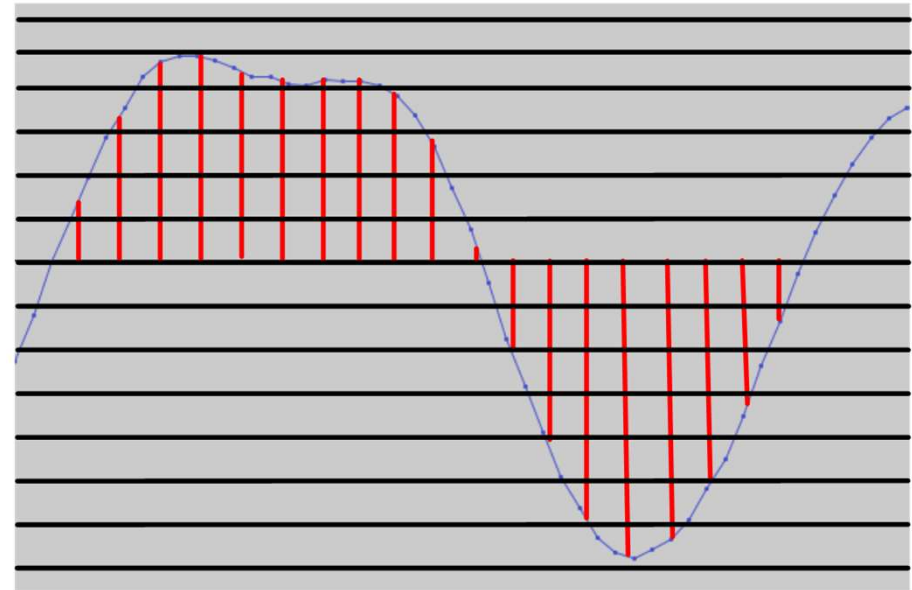
Resolução

Quantos valores diferentes se consideram?

Valores medidos são arredondados de acordo com a resolução

+ bits \rightarrow melhor digitalização \rightarrow maiores ficheiros

- 8 bits: -128 a 127
- 16 bits: -32768 a 32767
- 32 bits: $-(2^{31})$ a $2^{31} - 1$



Armazenamento



- Formatos: WAV, MP3, FLAC, ...
- Não comprimidos: WAV
- Comprimidos:
 - ▣ Sem perdas: FLAC, WAV
 - “Semelhante” a um ZIP (codificação de entropia)
 - ▣ Com perdas: MP3
 - Descartada informação “não audível”
 - Codificação de entropia, quantificação dinâmica, etc...

Armazenamento

- Amostragem a 44100 Hz, Mono, 16 bits, 12 segundos

Formato	Tamanho
WAV	1.058.444 octetos
FLAC	350.451 octetos
MP3	107.040 octetos

- Amostragem a 44100 Hz, Estéreo, 16 bits, ~4:30 minutos

Formato	Tamanho
WAV	48.545.436 octetos
FLAC	27.196.823 octetos
MP3	4.411.858 octetos

Tamanho WAV = Freq * Canais * Resolução * Tempo

WAV: WAVEform audio file format

- Características:

- Até 65535 canais (Estéreo = 2, Surround = 3 a 24)
- Frequências até 4.3 Ghz
- Resolução até 32 bits
- Codificação típica LPCM
 - Linear Pulse Code Modulation
 - Pode usar outras

- Muito comum em trabalho profissional

- Armazenamento simples, compatível e sem perdas

- Muito raro na Internet

- Ficheiros muito grandes

WAV

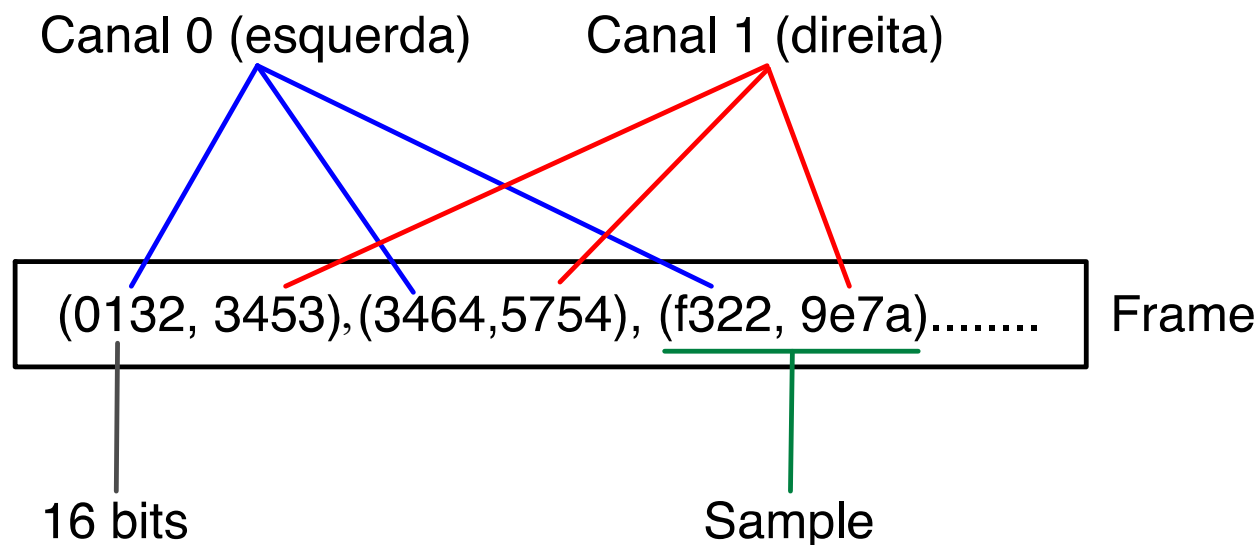
Organizado em Chunks

The Canonical WAVE file format

endian	File offset (bytes)	field name	Field Size (bytes)	
big	0	ChunkID	4	The "RIFF" chunk descriptor
little	4	ChunkSize	4	
big	8	Format	4	
big	12	Subchunk1 ID	4	The "fmt" sub-chunk describes the format of the sound information in the data sub-chunk
little	16	Subchunk1 Size	4	
little	20	AudioFormat	2	
little	22	NumChannels	2	
little	24	SampleRate	4	
little	28	ByteRate	4	
little	32	BlockAlign	2	
little	34	BitsPerSample	2	The "data" sub-chunk
big	36	Subchunk2ID	4	
little	40	Subchunk2 Size	4	
little	44	data		Indicates the size of the sound information and contains the raw sound data

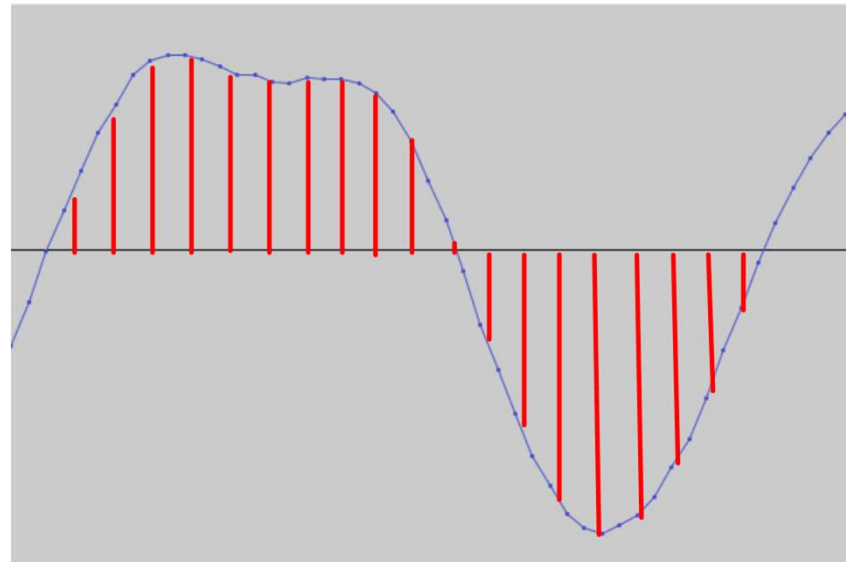
WAV

- Informação sonora organizada em Frames (janelas)
- Frames contêm Samples (amostras)
- Samples contêm Channels (canais)



LPCM

- Valores de tensão em cada ponto de digitalização
- Pode ser interpretado como uma série de valores (lista/array)



LPCM

- Possibilita a criação de efeitos manipulando a lista de valores
 - ▣ Normalização
 - ▣ Volume
 - ▣ Fade In/Fade Out
 - ▣ Echo/Reverb
 - ▣ Compressão, Expansão, etc..
- Possibilita uma fácil geração de sons
 - ▣ Variação de amplitude $x \sin(x)$

LPCM: Geração de tom 440 Hz

```
import math

data = []
rate = 44100
duration = 2 # segundos
vol = 32767 # Amplitude (volume)
i = 0

while i < rate*duration:
    data.append( vol*math.sin(2*math.pi*440.0*i/rate) )
    i += 1
```

Python Audio

- Dados LPCM codificados num formato binário
 - ▣ 8 bit unsigned, 16 bit signed, etc...
 - ▣ Necessário converter ao ler e escrever

```
from struct import pack, unpack

# Converter lista (data) em LPCM binário (wvData)
wvData=[]
for v in data:
    wvData.append(pack("h", int(v) ))

# Converter dados para escrever no ficheiro
data = bytearray(wvData)
```