

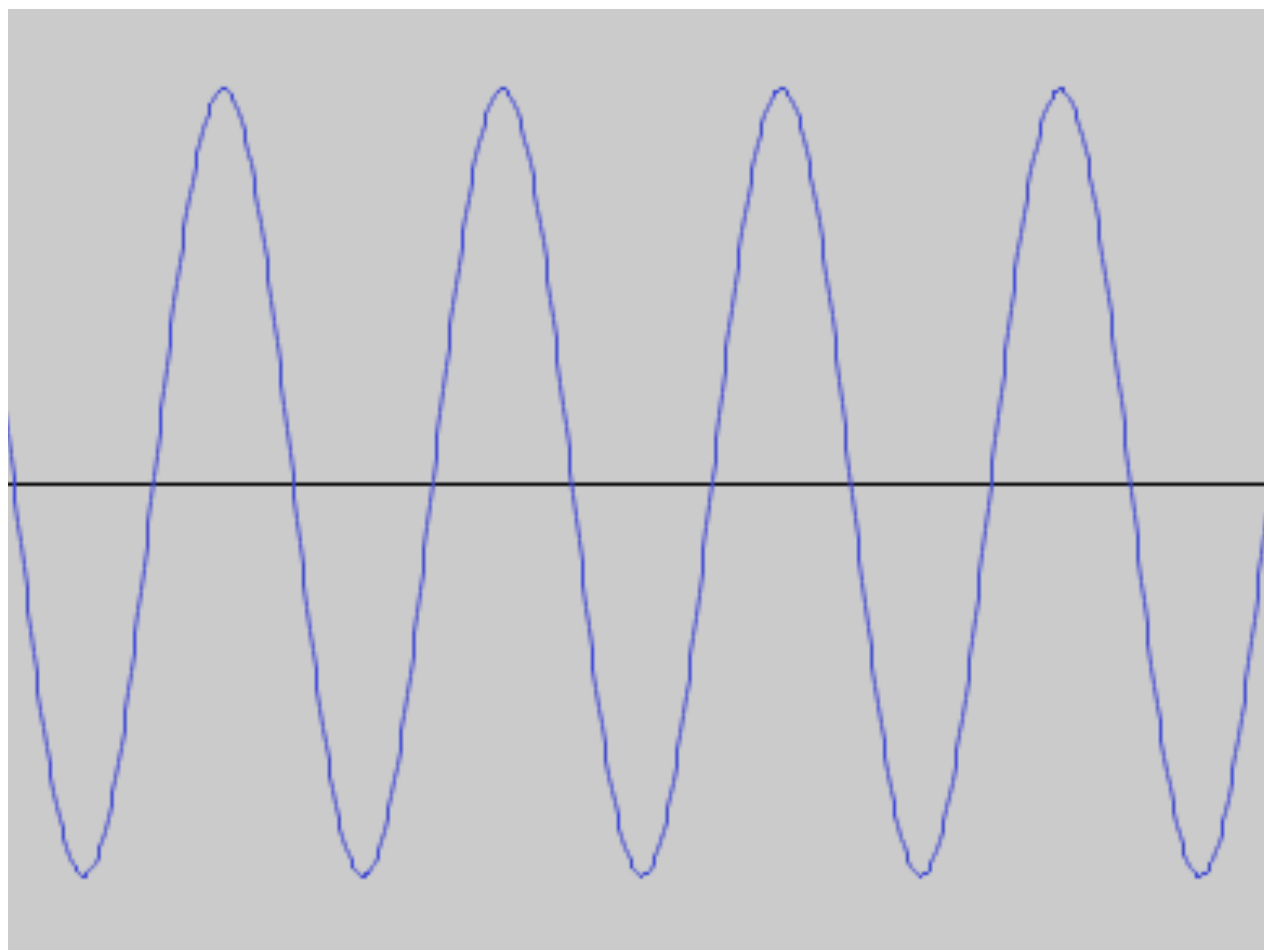
Wikimedia Commons

# INFORMAÇÃO SONORA

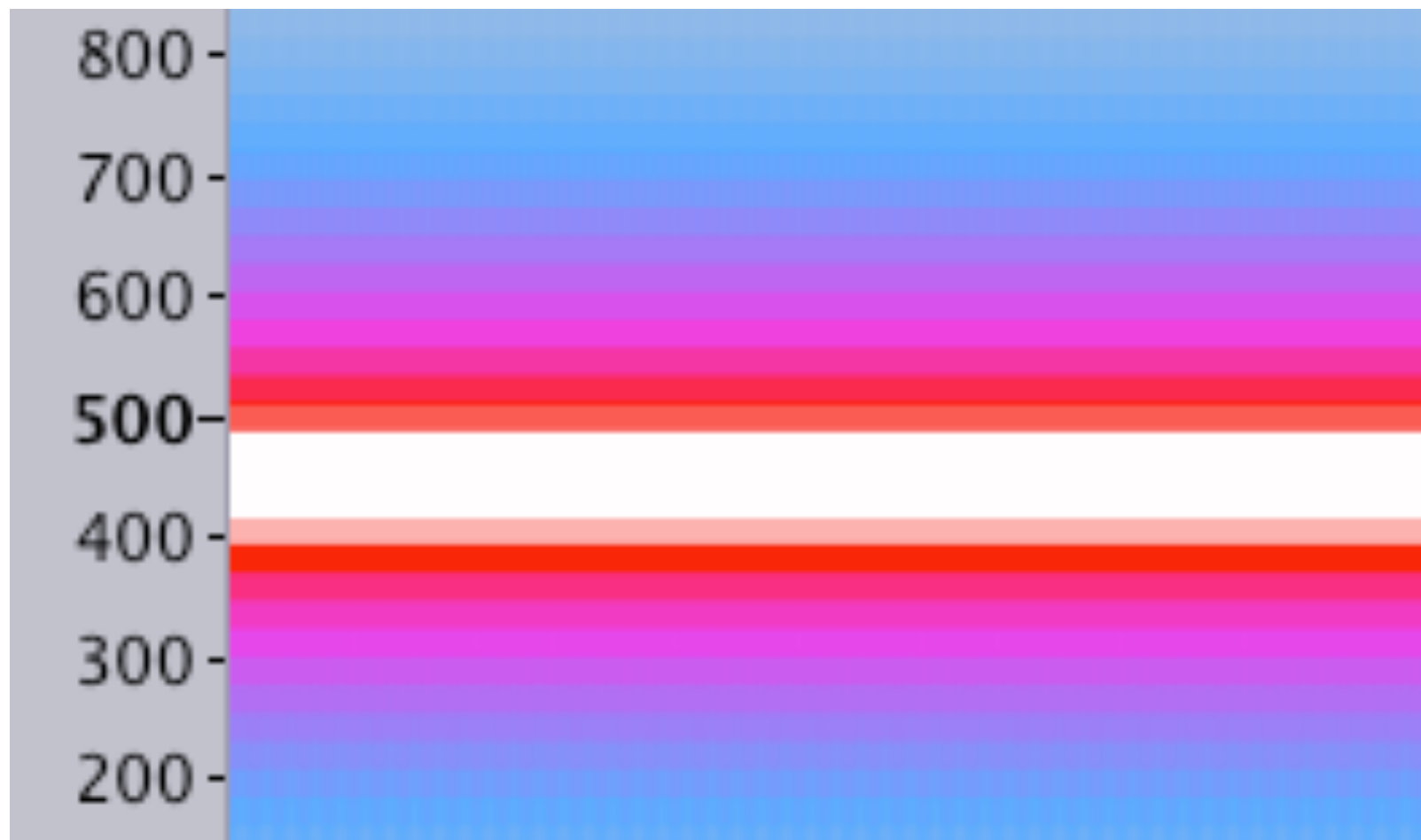
# Som

- 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 440Hz (Amplitude vs Tempo)



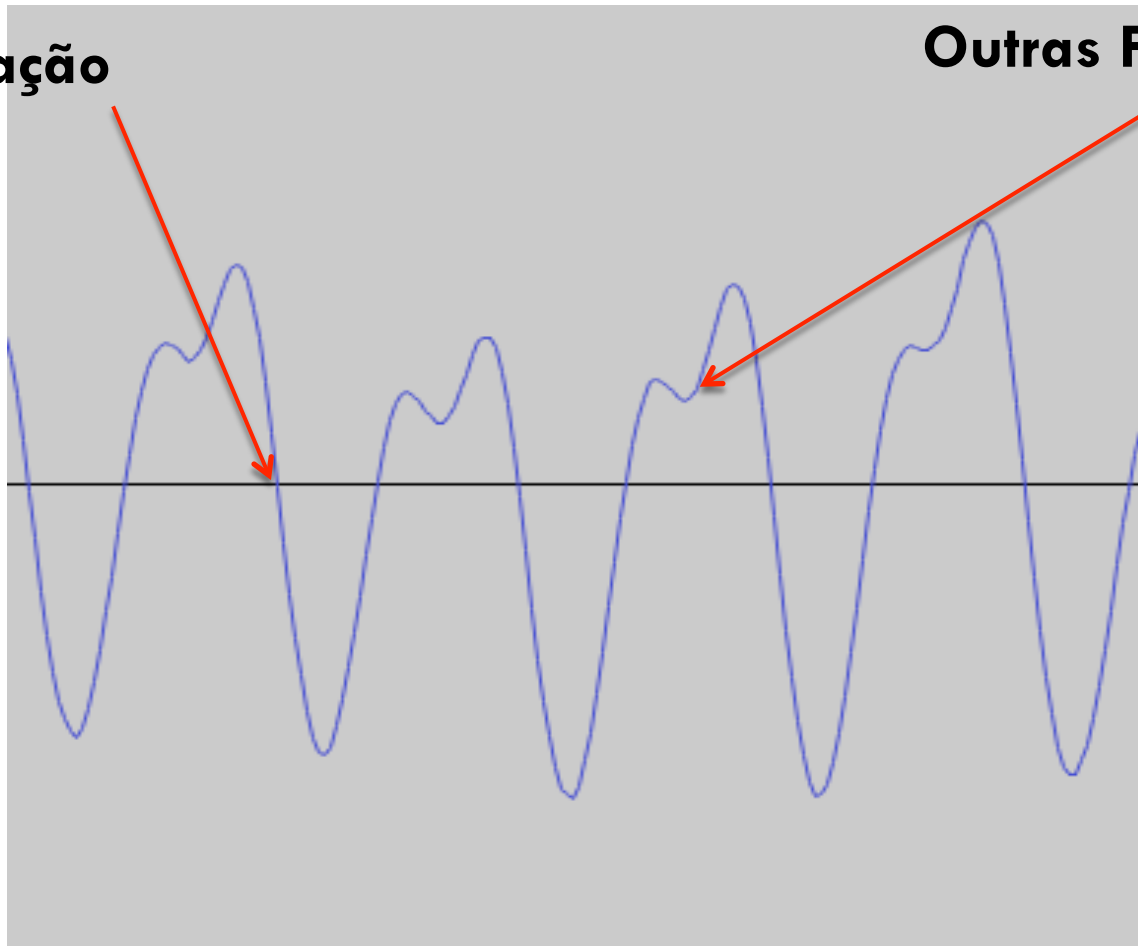
# Tom 440Hz (Frequência vs Tempo)



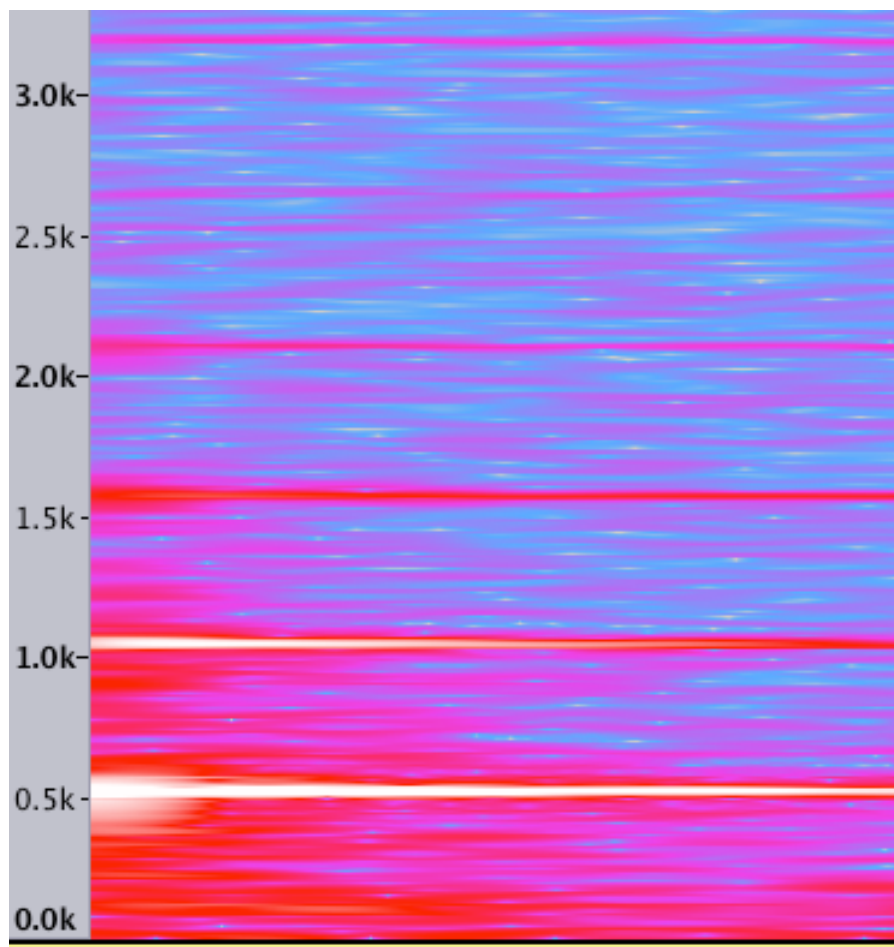
# Piano (C5, 523Hz)

**Oscilação**

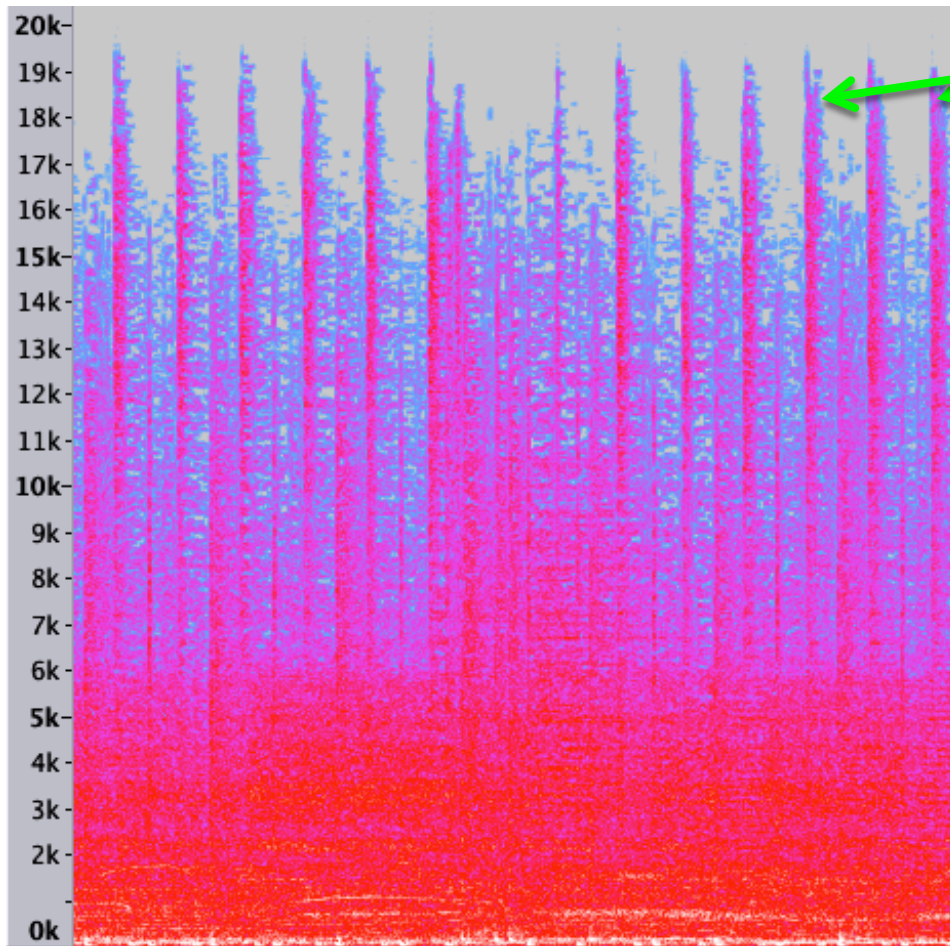
**Outras Frequências**



# Piano (C5, 523Hz)



# Música



**Ritmo da música**

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

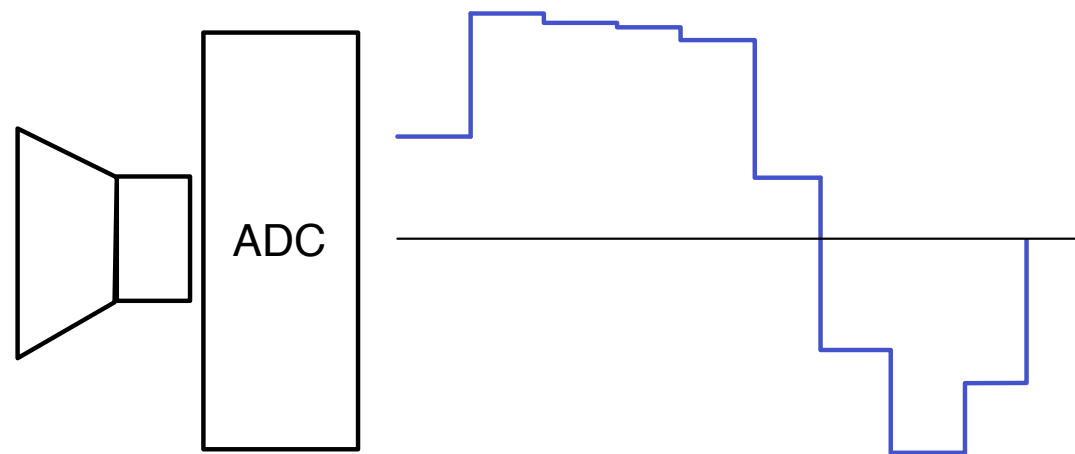
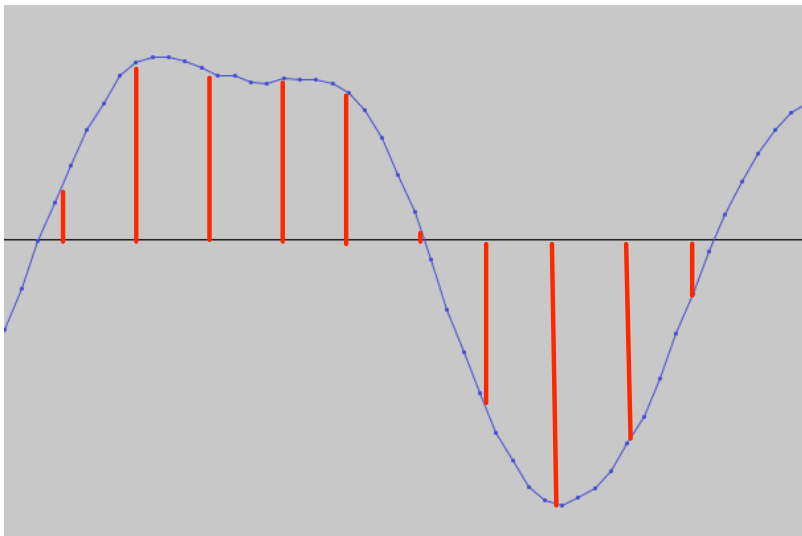
# ADC e DAC

- Aquisição: Som é digitalizado por um microfone
  - ▣ Analog Digital Conversion (ADC)
- Reprodução: Níveis eléctricos são definidos no altifalante
  - ▣ Digital to Analog Conversion (DAC)



# Digitalização

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

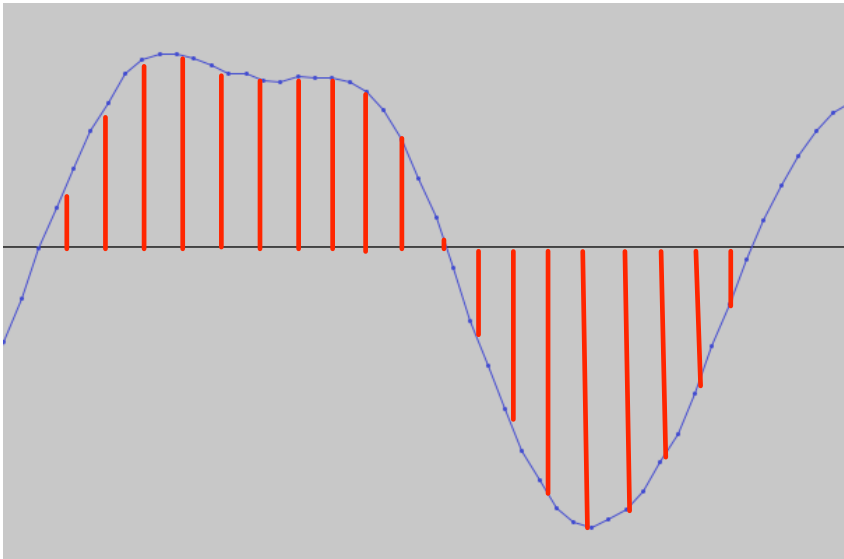


# Frequência de Amostragem

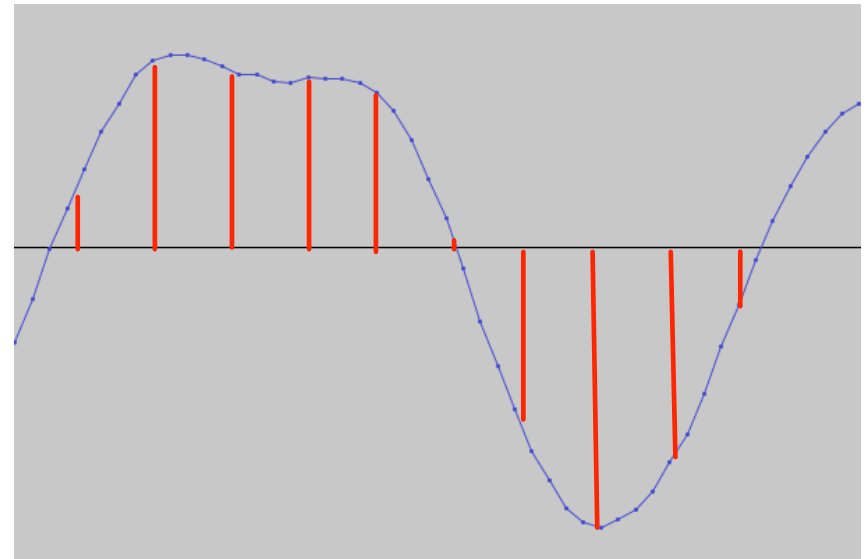
Quantas vezes por segundo se lê o valor de tensão

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 Shannon-Hartley
  - ▣ no mínimo
- Ouvido humano distingue até 20KHz
  - ▣ Digitalizar música: 44100Hz, 48000Hz
  - ▣ Gravação “normal” num PC
- Voz humana entre 1KHz e 3KHz
  - ▣ Digitalizar voz: 8KHz
  - ▣ Comunicações por voz (telemóveis, skype, etc...)

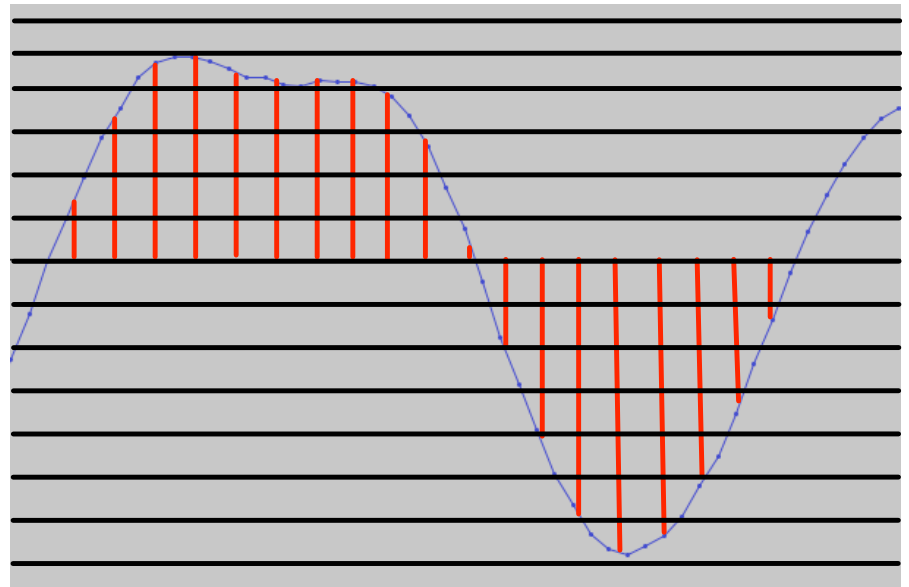
# Resolução

Quantos valores diferentes se consideram?

Valores medidos são arredondados de acordo com a resolução

+ bits --- melhor digitalização --- 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

- Música a 44100Hz, Mono, 16bit, 12 segundos

Formato	Tamanho
WAV	1.058.444 octetos
FLAC	350.451 octetos
MP3	107.040 octetos

- Música a 44100Hz, Stereo, 16bit, ~4:30m

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 (Stereo = 2, Surround = 3 a 24)
  - ▣ Frequências até 4.3Ghz
  - ▣ Resolução até 32bits
  - ▣ 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	Subchunk1ID	4	The "fmt" 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	
big	36	Subchunk2ID	4	
little	40	Subchunk2 Size	4	
little	44	data		The "data" sub-chunk

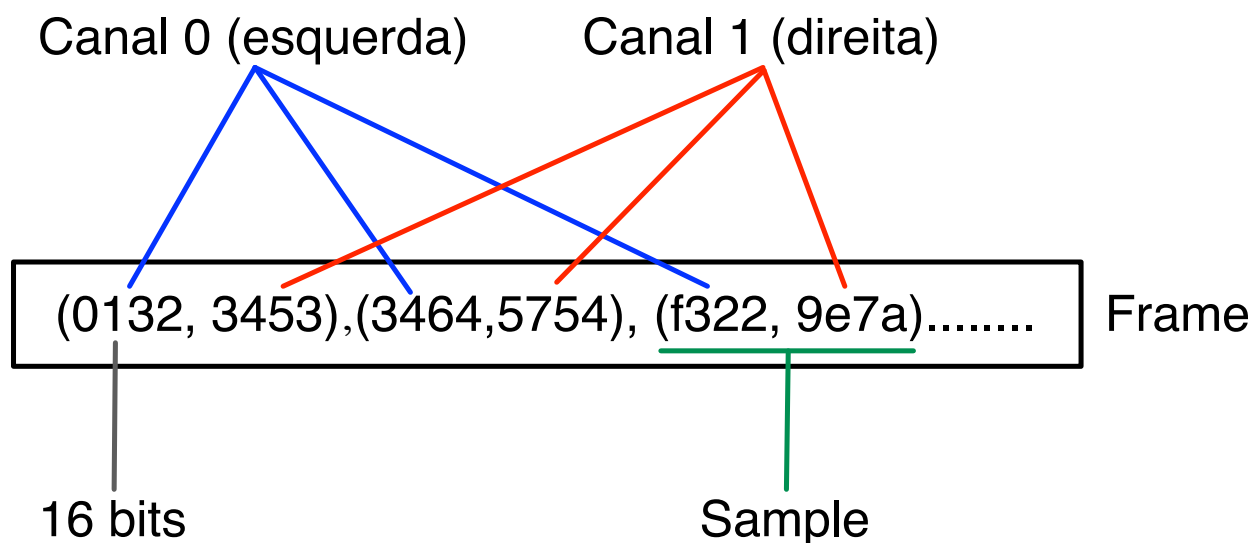
Subchunk2Size

Indicates the size of the sound information and contains the raw sound data



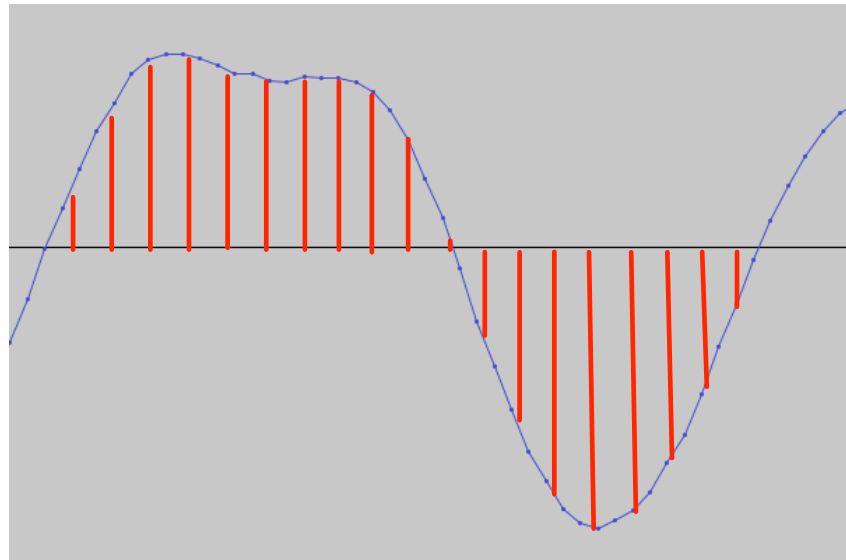
# WAV

- Informação sonora organizada em Frames
- Frames contêm Samples
- Samples contêm canais



# LPCM

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



# LPCM

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

# LPCM: Geração de tom 440Hz

```
import math

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

while i < rate*duration:
    data.append( vol*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 += pack('h',v)

# Converter LPCM binário em lista
fmt = '%ih' % len(wvData)
data = unpack(fmt, wvData)
```