# Conversão Analógico-Digital

## Capítulo 7 (Proakis) e Capítulo 5 (Lathi)

Prof. Daniel Costa Araújo

Universidade de Brasília
Laboratório de Telecomunicações

February 19, 2026

# Sumário

1. Amostragem
2. Quantização
3. PCM e Companding

# Capítulo 7

## Conversão Analógico-Digital

### Amostragem, Quantização, PCM e Companding

# Por Que Digitalizar Sinais?

**O mundo é analógico:**

- Voz humana $\rightarrow$ pressão de ar contínua
- Música $\rightarrow$ ondas mecânicas
- Temperatura, pressão, tensão elétrica. . .

**Computadores só entendem 0 e 1.**

Precisamos converter o sinal analógico para digital (ADC).

**Aplicações cotidianas:**

- **Telefonia:** $f_s = 8\,\text{kHz}$, 8 bits
- **CD de áudio:** $f_s = 44{,}1\,\text{kHz}$, 16 bits
- **Wi-Fi / 5G:** ADC em GHz!

**Cadeia ADC — 3 etapas:**

1. Amostragem
2. Quantização
3. Codificação (PCM)

# Modelo Matemático do Amostrador Ideal

Seja $g(t)$ um sinal **passa-baixas limitado em banda**:

$$G(f) = 0 \quad \text{para } |f| \geq W$$

O amostrador ideal multiplica $g(t)$ por um **trem de impulsos de Dirac** ("pente") espaçado de $T_s$:

$$\delta_{T_s}(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \qquad T_s = \frac{1}{f_s}$$

O **sinal amostrado** resultante é:

$$g_s(t) = g(t) \cdot \delta_{T_s}(t)$$

**Questão fundamental:** qual o menor $f_s$ que ainda permite reconstruir $g(t)$ com perfeição?

# Derivação — Passo 1: Sinal Amostrado no Tempo

Expandindo $g_s(t)$ com a definição do trem de impulsos:

$$g_s(t) = g(t) \cdot \sum_{n=-\infty}^{\infty} \delta(t - nT_s) = \sum_{n=-\infty}^{\infty} g(t)\,\delta(t - nT_s)$$

Usamos a **propriedade de filtragem** da delta de Dirac:

$$x(t)\,\delta(t - t_0) = x(t_0)\,\delta(t - t_0)$$

O impulso em $t_0$ "captura" o valor $x(t_0)$ e descarta todo o resto do sinal.

Aplicando a cada termo da soma:

$$g_s(t) = \sum^{\infty} g(nT_s)\,\delta(t - nT_s)$$

# Derivação — Passo 2: Transformada de Fourier e Convolução

No domínio da frequência, **multiplicação no tempo** equivale a **convolução na frequência**:

$$G_s(f) = \mathcal{F}\{g_s(t)\} = \mathcal{F}\{g(t) \cdot \delta_{T_s}(t)\} = G(f) * \mathcal{F}\{\delta_{T_s}(t)\}$$

Para completar, precisamos calcular $\mathcal{F}\{\delta_{T_s}(t)\}$ — a transformada do trem de impulsos.

**Passo 3 — Transformada do Trem de Impulsos**

O trem de impulsos é periódico com período $T_s$. Desenvolvendo-o em Série de Fourier exponencial, todos os coeficientes têm o mesmo módulo $1/T_s = f_s$. Portanto:

$$\mathcal{F}\{\sum_{n=-\infty}^{\infty} \delta(t - nT_s)\} = f_s \sum_{k=-\infty}^{\infty} \delta(f - k\, f_s)$$

*Conclusão elegante: um trem de impulsos no tempo gera outro trem de impulsos na frequência.*

# Derivação — Passo 4: Convolução com o Trem de Impulsos

Substituindo em $G_s(f) = G(f) * \mathcal{F}\{\delta_{T_s}(t)\}$:

$$G_s(f) = G(f) * \left( f_s \sum_{k=-\infty}^{\infty} \delta(f - k\, f_s) \right)$$

Pela **distributividade** da convolução, levamos $*$ para dentro do somatório:

$$G_s(f) = f_s \sum_{k=-\infty}^{\infty} G(f) * \delta(f - k\, f_s)$$

Aplicamos a propriedade de **deslocamento por convolução**:

$$X(f) * \delta(f - f_0) = X(f - f_0)$$

# Resultado: Espectro do Sinal Amostrado

## Espectro do Sinal Amostrado

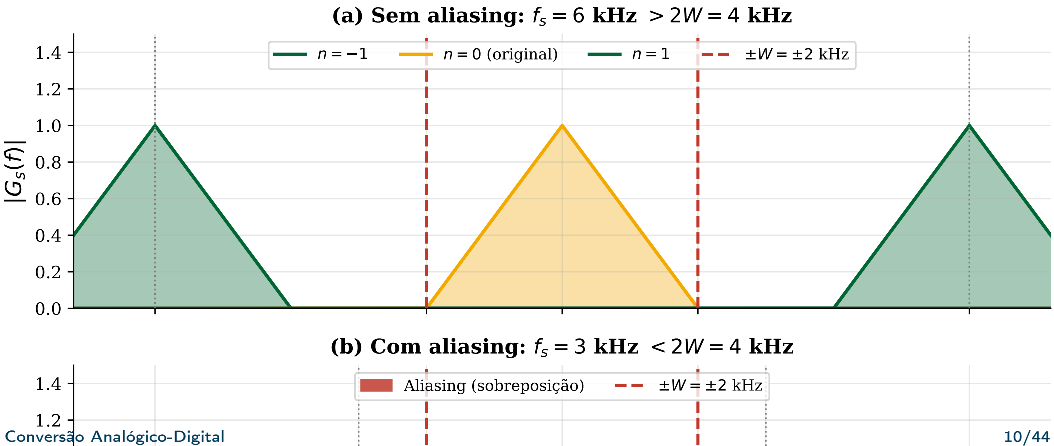
$$G_s(f) = f_s \sum_{k=-\infty}^{\infty} G(f - k\, f_s)$$

O espectro de $g_s(t)$ é uma **soma periódica de réplicas** de $G(f)$, cada uma deslocada de $k\, f_s$ e escalada por $f_s$.

**Consequências imediatas:**

- A amostragem repete o espectro a cada $f_s$ Hz.
- Se as réplicas não se sobrepõem, $G(f)$ pode ser isolada com um filtro passa-baixas ideal de corte em $W$.
- Se houver sobreposição $\Rightarrow$ aliasing — distorção irrecuperável

# Visualização do Espectro Amostrado



# Condição de Nyquist — Derivação

Para reconstrução perfeita, as réplicas **não podem se sobrepor**.

- Réplica em $k = 0$: ocupa $[-W, +W]$
- Réplica em $k = 1$: ocupa $[f_s - W, f_s + W]$

Para que as réplicas em $k = 0$ e $k = 1$ não se sobreponham, o início da réplica em $f_s$ deve estar além do fim da réplica em 0:

$$f_s - W > W \quad \Longrightarrow \quad f_s > 2W$$

## Teorema de Nyquist-Shannon

Um sinal $g(t)$ com espectro limitado a $[-W, W]$ Hz pode ser **perfeitamente reconstruído** a partir de suas amostras $\{g(nT_s)\}$ se, e somente se:

# Aliasing: O Que Acontece Quando $f_s < 2W$?

**Aliasing** = sobreposição das réplicas espectrais.

**Efeito prático:** uma componente de frequência $f_0 > f_s/2$ "se disfarça" de uma frequência espúria $f_{\text{alias}} = |f_0 - f_s|$ no sinal recuperado.

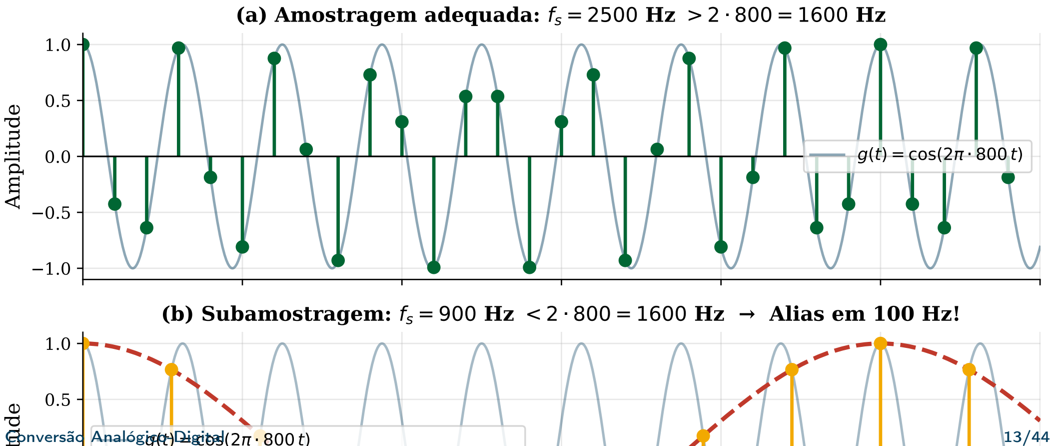
**Exemplo numérico:** sinal em $f_0 = 800$ Hz amostrado com $f_s = 900$ Hz.

$$f_{\text{alias}} = |800 - 900| = 100\,\text{Hz} \qquad \longleftarrow \qquad \text{frequência completamente errada!}$$

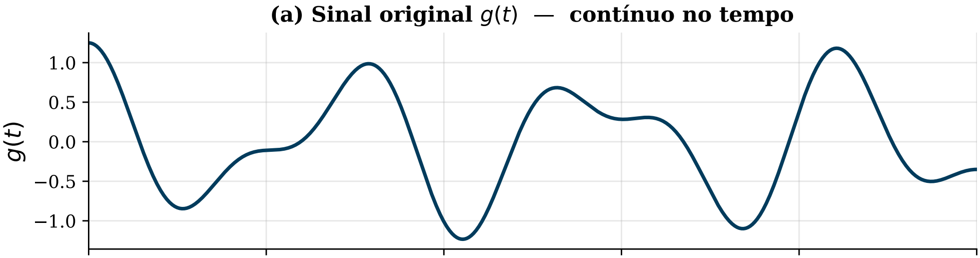
**Observação:** uma vez que o aliasing ocorre, a informação perdida não pode ser recuperada.

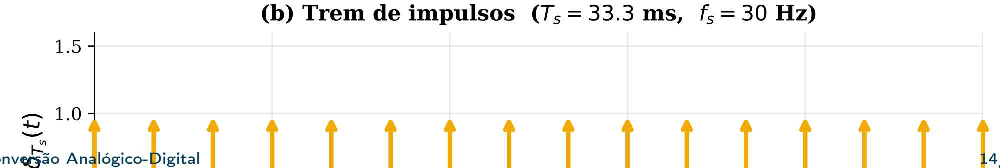
**Solução na prática:** inserir um **filtro anti-aliasing** (passa-baixas analógico com corte em $f_s/2$) *antes* do amostrador, garantindo que o sinal seja estritamente limitado em banda.

# Aliasing — Demonstração Visual



# Amostragem no Domínio do Tempo



# Reconstrução: Filtro Passa-Baixas Ideal

Se $f_s > 2W$, podemos **recuperar** $g(t)$ **exatamente** filtrando $g_s(t)$ com um filtro passa-baixas ideal de corte em $W$:

$$H(f) = \frac{1}{f_s}\,\mathrm{rect}\left(\frac{f}{2W}\right) \quad \Longrightarrow \quad \hat{G}(f) = G_s(f)\cdot H(f) = G(f)$$

No domínio do tempo, o filtro ideal corresponde a um pulso sinc:

$$h(t) = \mathcal{F}^{-1}\{H(f)\} = \frac{2W}{f_s}\,\mathrm{sinc}(2Wt)$$

A reconstrução completa do sinal contínuo é a **interpolação sinc**:

$$g(t) = \sum_{n=-\infty}^{\infty} g(nT_s)\,\mathrm{sinc}\left(\frac{t-nT_s}{T_s}\right)$$

# Por Que Quantizar?

Após a amostragem, temos $g[n] = g(nT_s)$: discreto no tempo, mas com **amplitude ainda contínua**.

**O problema:**

- Um processador só armazena valores discretos
- Com $n$ bits há apenas $2^n$ valores representáveis
- Amplitudes contínuas **precisam ser arredondadas**

**Analogia cotidiana:**

- Régua com marcas a cada 1 mm
- 23,731 mm $\rightarrow$ 23,7 mm
- O erro é inevitável — só reduzível

**A questão central:**

1. Quanto ruído a quantização introduz?

# O Quantizador Uniforme

Seja $g[n]$ com faixa de amplitude $[V_{\min}, V_{\max}]$.

Dividindo a faixa total $V_{pp} = V_{\max} - V_{\min}$ em $L$ intervalos **iguais**, o tamanho de cada degrau é:

$$\Delta = \frac{V_{pp}}{L}$$

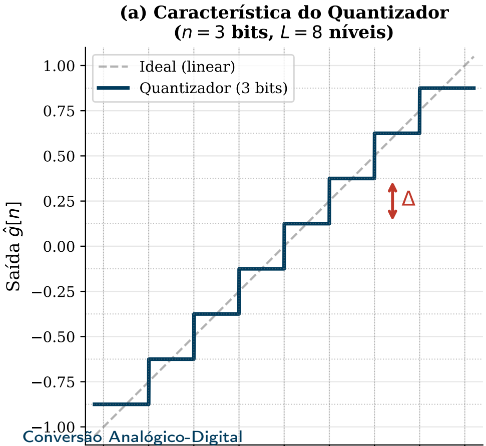
O $k$-ésimo intervalo cobre $\left[V_{\min} + k\Delta,\ V_{\min} + (k+1)\Delta\right)$ e é mapeado para o seu **ponto médio**:

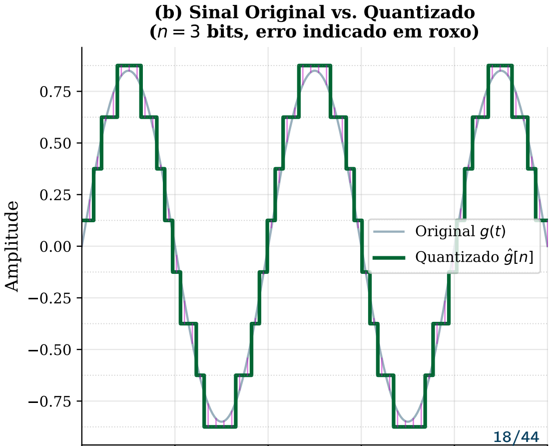
$$\hat{g}[n] = V_{\min} + \left(k + \tfrac{1}{2}\right)\Delta, \quad k = 0, 1, \ldots, L-1$$

A saída é uma **função escada**: cada amplitude de entrada é arredondada para o nível de quantização mais próximo.

Com $n$ bits podemos representar $L = 2^n$ níveis.

# Característica do Quantizador



# O Erro de Quantização

O **erro de quantização** é a diferença entre o valor quantizado e o valor real:

$$e_q[n] = \hat{g}[n] - g[n]$$

Como o quantizador mapeia cada amostra para o **ponto médio** de seu intervalo $\Delta$, o erro fica sempre limitado a:

$$-\frac{\Delta}{2} \leq e_q[n] \leq +\frac{\Delta}{2}$$

**Compromisso fundamental:**

- Erro menor $\Rightarrow$ $\Delta$ menor $\Rightarrow$ mais níveis $L = V_{pp}/\Delta$ $\Rightarrow$ mais bits $n = \log_2 L$
- Aumentar $n$ em 1 bit $\Rightarrow$ $\Delta$ cai pela metade $\Rightarrow$ erro máximo cai pela metade

# Modelo Estatístico do Erro de Quantização

**Hipóteses do modelo de ruído de quantização** (válidas para $L$ grande e sinal variando rapidamente):

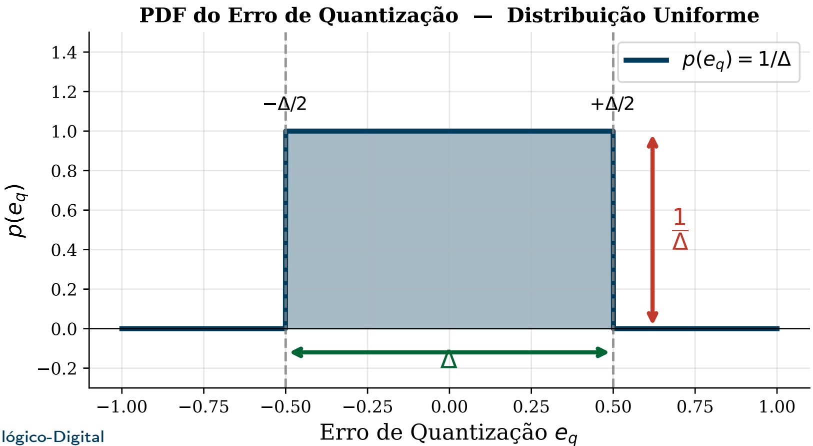
1. $e_q[n]$ é uma **variável aleatória uniforme** em $[-\Delta/2, +\Delta/2]$
2. Erros em amostras consecutivas são **não-correlacionados**
3. O erro é **independente** do sinal de entrada

Sob essas hipóteses, a função de densidade de probabilidade (PDF) é:

$$p(e_q) = \begin{cases} \dfrac{1}{\Delta}, & \text{se } -\dfrac{\Delta}{2} \leq e_q \leq \dfrac{\Delta}{2} \\ 0, & \text{caso contrário} \end{cases}$$

Cada valor do erro é **igualmente provável** dentro do intervalo.

# PDF do Erro de Quantização



# Derivação de $P_q$ — Passo 1: Definição Formal

A **potência do ruído de quantização** $P_q$ é o valor esperado do *quadrado* do erro:

$$P_q = E\left[e_q^2\right]$$

Para uma variável aleatória contínua, o valor esperado de uma função $f(e_q)$ é calculado como:

$$E[f(e_q)] = \int_{-\infty}^{+\infty} f(e_q)\, p(e_q)\, de_q$$

Portanto, a definição formal de $P_q$ é:

$$P_q = \int_{-\infty}^{+\infty} e_q^2\, p(e_q)\, de_q$$

*Interpretação:* pesamos $e_q^2$ pela sua probabilidade $p(e_q)$ e somamos (integramos) sobre todos

# Derivação de $P_q$ — Passo 2: Substituição da PDF

Como $p(e_q) = 1/\Delta$ somente no intervalo $[-\Delta/2, +\Delta/2]$ e zero fora dele, podemos restringir os limites da integral:

$$P_q = \int_{-\Delta/2}^{+\Delta/2} e_q^2 \cdot \frac{1}{\Delta}\, de_q$$

**Passo 3 — Resolver a integral**

A constante $1/\Delta$ sai da integral:

$$P_q = \frac{1}{\Delta} \int_{-\Delta/2}^{+\Delta/2} e_q^2\, de_q$$

A primitiva de $e_q^2$ é $e_q^3/3$. Aplicando o teorema fundamental do cálculo:

$$\frac{1}{\Delta}\left[e_q^3\right]^{+\Delta/2} \quad \frac{1}{\Delta} \quad (\Delta/2)^3 - (-\Delta/2)^3$$

# Derivação de $P_q$ — Passo 4: Simplificação

Calculamos os cubos:

$$\left(\frac{\Delta}{2}\right)^3 = \frac{\Delta^3}{8}, \qquad \left(-\frac{\Delta}{2}\right)^3 = -\frac{\Delta^3}{8}$$

Substituindo na expressão anterior:

$$P_q = \frac{1}{\Delta} \cdot \frac{\frac{\Delta^3}{8} - \left(-\frac{\Delta^3}{8}\right)}{3} = \frac{1}{\Delta} \cdot \frac{2 \cdot \frac{\Delta^3}{8}}{3} = \frac{1}{\Delta} \cdot \frac{\Delta^3}{12}$$

Cancelando um fator $\Delta$:

$$P_q = \frac{\Delta^2}{12}$$

# Relação Sinal-Ruído de Quantização (SQNR)

A qualidade da quantização é medida pelo **SQNR** (*Signal-to-Quantization-Noise Ratio*):

$$\mathrm{SQNR} = \frac{P_s}{P_q} = \frac{P_s}{\Delta^2/12} = \frac{12\,P_s}{\Delta^2}$$

Expressando $\Delta$ em termos do número de bits $n$ ($L = 2^n$ níveis, faixa $V_{pp}$):

$$\Delta = \frac{V_{pp}}{2^n} \implies \mathrm{SQNR} = \frac{12\,P_s}{(V_{pp}/2^n)^2} = \frac{12\,P_s}{V_{pp}^2} \cdot 2^{2n}$$

**Observação:** $2^{2n} = 4^n$. Adicionar 1 bit multiplica o SQNR por 4 (ou seja, +6 dB).

# SQNR para um Sinal Senoidal — Derivação

**Caso prático:** senoide $g(t) = A\cos(\omega t)$ que ocupa **toda a faixa** do quantizador.

Parâmetros:

$$V_{pp} = A - (-A) = 2A, \qquad P_s = \frac{A^2}{2} \quad \text{(potência média da senoide)}$$

Substituindo na expressão geral do SQNR:

$$\mathrm{SQNR} = \frac{12 \cdot (A^2/2)}{(2A)^2} \cdot 2^{2n} = \frac{6A^2}{4A^2} \cdot 2^{2n} = \frac{3}{2} \cdot 2^{2n} = 1{,}5 \cdot 4^n$$

Convertendo para decibéis ($X_{\mathrm{dB}} = 10\log_{10} X$):

$$\mathrm{SQNR}_{\mathrm{dB}} = 10\log_{10}(1{,}5 \cdot 4^n) = 10\log_{10}(1{,}5) + n \cdot 10\log_{10}(4)$$

# A Regra dos 6 dB por Bit

Calculando os valores numéricos dos logaritmos:

$$10\log_{10}(1{,}5) \approx 1{,}76 \text{ dB}, \qquad 10\log_{10}(4) \approx 6{,}02 \text{ dB}$$

Portanto:

$$\mathrm{SQNR}_{\mathrm{dB}} \approx 1{,}76 + 6{,}02\, n$$

**Regra dos 6 dB/bit (senoide em faixa cheia)**

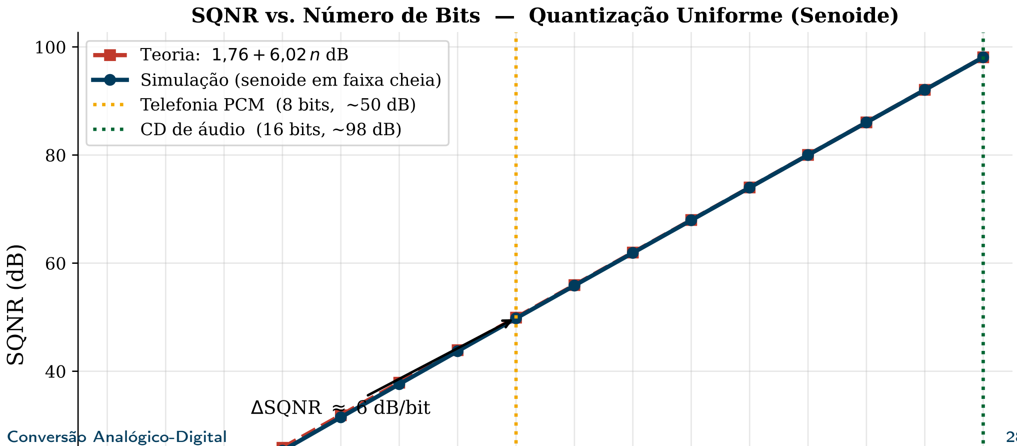
$$\mathrm{SQNR}_{\mathrm{dB}} \approx 1{,}76 + 6{,}02\, n \text{ dB}$$

Cada bit adicional eleva o SQNR em $\approx 6$ **dB**.

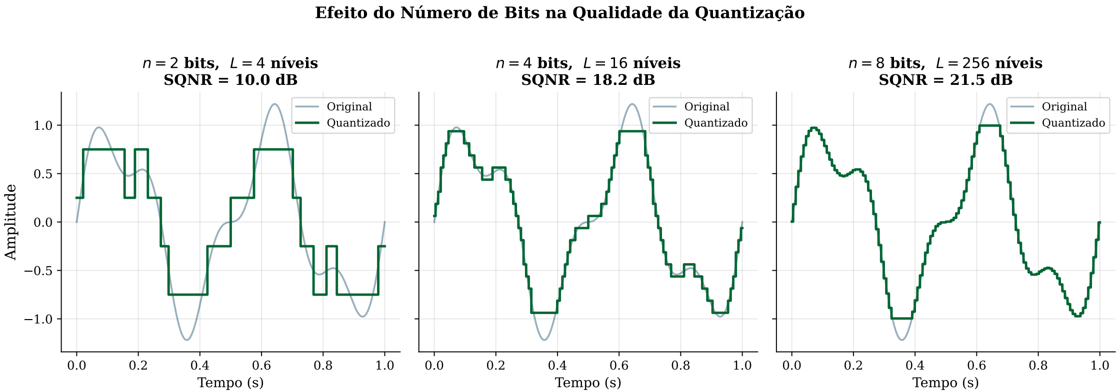
**Exemplos práticos:**

- $n = 8$ bits: $\mathrm{SQNR} \approx 50$ dB (telefonia PCM)

# SQNR vs. Número de Bits — Confirmação Numérica



# Efeito da Resolução na Qualidade do Sinal




Com $n = 2$ bits a forma de onda é severamente distorcida. Com $n = 8$ bits a diferença para o sinal original é

# Implementação em Python

```
import numpy as np

def quantize_uniform(x, n_bits, v_min=-1.0, v_max=1.0):
    L     = 2**n_bits                          # numero de niveis
    delta = (v_max - v_min) / L                # tamanho do passo
    x_c   = np.clip(x, v_min, v_max)           # limitar faixa
    idx   = np.floor((x_c - v_min)/delta).astype(int)
    idx   = np.clip(idx, 0, L-1)
    return v_min + (idx + 0.5) * delta          # ponto medio

# Calcular SQNR para varios numeros de bits
t = np.linspace(0, 100, 1_000_000)
g = np.sin(2*np.pi*t)                          # senoide em faixa cheia

for n in [4, 8, 16]:
    g_q  = quantize_uniform(g, n)
    sqnr = 10 * np.log10(np.mean(g**2) / np.mean((g_q-g)**2))
    teo  = 1.76 + 6.02*n
```

# A Cadeia ADC Completa

$g(t) \rightarrow$ Filtro Anti-aliasing $\xrightarrow{g(t) \text{ filtrado}}$ Amostrador $f_s > 2W$ $\xrightarrow{g[n]}$ Quantizador $L = 2^n$ $\xrightarrow{\hat{g}[n]}$ Codificador PCM

$f_c \leq f_s/2$ | $T_s = 1/f_s$ | $\Delta = V_{pp}/L$ | $n$ bits/amostra

Cada etapa tem parâmetros de projeto que determinam a qualidade e a taxa de bits resultante.

# Codificação Binária: PCM

Após a quantização, cada nível $\hat{g}[n]$ recebe uma **palavra binária de $n$ bits** (representação PCM).

Com $n$ bits podemos representar $L = 2^n$ níveis:

$$L = 2^n \quad \Longleftrightarrow \quad n = \log_2 L$$

**Exemplo com $n = 3$ bits ($L = 8$ níveis):**

| Índice $k$ | Código binário | Nível $\hat{g}$ |
|---|---|---|
| 0 | 000 | $V_{\text{min}} + 0{,}5\Delta$ |
| 1 | 001 | $V_{\text{min}} + 1{,}5\Delta$ |
| $\vdots$ | $\vdots$ | $\vdots$ |
| 7 | 111 | $V_{\text{min}} + 7{,}5\Delta$ |

# Taxa de Bits e Largura de Banda PCM

A **taxa de bits** (bit rate) do canal PCM é:

$$R_b = n \cdot f_s \quad \text{[bits/s]}$$

onde $n$ é o número de bits por amostra e $f_s$ é a taxa de amostragem.

A largura de banda mínima necessária para transmitir o PCM (sem inter-símbolos) é:

$$B_{\text{min}} = \frac{R_b}{2} = \frac{n\, f_s}{2}$$

**Compromisso clássico:**

- Qualidade alta $\Rightarrow$ $n$ grande $\Rightarrow$ $R_b$ alto $\Rightarrow$ mais largura de banda
- Largura de banda limitada $\Rightarrow$ $n$ reduzido $\Rightarrow$ mais ruído de quantização

# Exemplo de Projeto PCM: Telefonia Digital

**Parâmetros do canal de voz (ITU-T G.711):**

- Largura de banda da voz: $W = 4\,\text{kHz}$
- Taxa de amostragem escolhida: $f_s = 8\,\text{kHz}$ (garante $f_s > 2W$, com margem)
- Resolução: $n = 8$ bits/amostra ($L = 256$ níveis)

**Taxa de bits por canal:**

$$R_b = n \cdot f_s = 8 \times 8\,000 = 64\,\text{kb/s}$$

**SQNR máximo (senoide):**

$$\mathrm{SQNR}_{\mathrm{dB}} \approx 1{,}76 + 6{,}02 \times 8 \approx 50\,\text{dB}$$

**Observação:** 50 dB é adequado para inteligibilidade da voz, mas ainda abaixo da qualidade de áudio de alta fidelidade (CD: ∼98 dB).

# Por Que Companding? Motivação

**Problema com quantização uniforme para a voz:**

A amplitude da voz humana **não tem distribuição uniforme** — amplitudes pequenas são muito mais frequentes do que amplitudes grandes.

Com quantização uniforme:

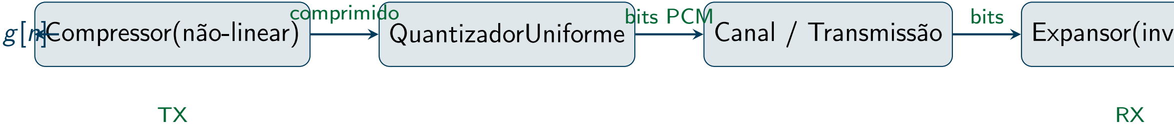
- Os poucos níveis disponíveis se distribuem **igualmente** por toda a faixa
- A maioria das amostras (pequena amplitude) usa apenas uma pequena fração dos níveis
- Para sinais fracos, o SQNR cai drasticamente

**Solução:** usar passos $\Delta$ **menores perto de zero** e maiores para amplitudes altas. Isso é a quantização não-uniforme.

Na prática, implementada via **companding**: **comp**ression + exp**anding**.

# O Conceito de Companding

**Companding = Compressão + Quantização Uniforme + Expansão**




- **Compressor:** amplifica amplitudes pequenas e comprime amplitudes grandes antes da quantização
- **Expansor:** aplica a função inversa no receptor para restaurar as amplitudes originais

# Lei $\mu$: Definição Matemática

Padrão nos EUA, Canadá e Japão (ITU-T G.711, $\mu = 255$):

$$C_\mu(x) = \text{sgn}(x)\,\frac{\ln(1+\mu\,|x|)}{\ln(1+\mu)}, \qquad |x| \leq 1$$

**Propriedades:**

- Para $\mu|x| \gg 1$ (sinal grande): $C_\mu(x) \approx \text{sgn}(x)\,\dfrac{\ln(\mu|x|)}{\ln(1+\mu)}$ — compressão logarítmica
- Para $\mu|x| \ll 1$ (sinal pequeno): $C_\mu(x) \approx x$ — aproximadamente linear
- Caso limite: $\mu \to 0$ recupera a quantização uniforme

A **inversão** (expansor) é:

$$C_\mu^{-1}(y) = \text{sgn}(y)\,\frac{(1+\mu)^{|y|} - 1}{\mu}$$

# Lei A: Definição Matemática

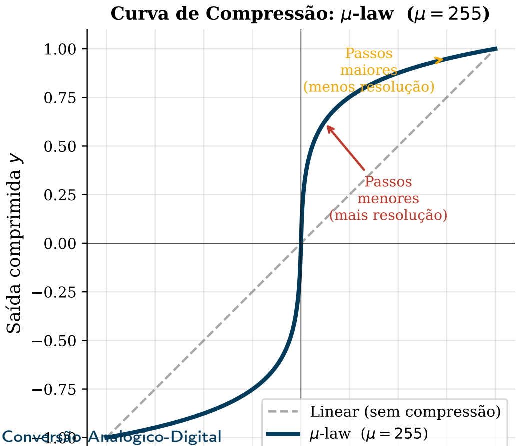
Padrão na Europa e em sistemas legados no Brasil (ITU-T G.711, $A = 87{,}6$):

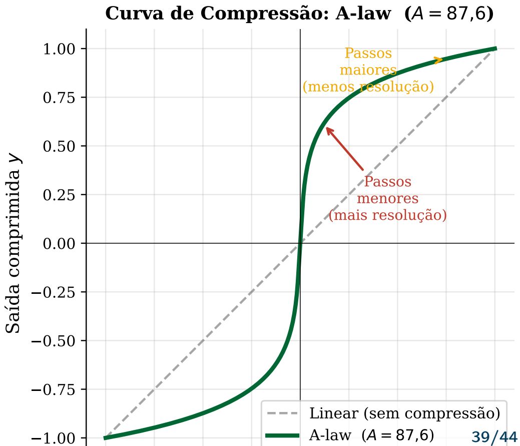
$$C_A(x) = \mathrm{sgn}(x) \cdot \begin{cases} \dfrac{A\,|x|}{1 + \ln A}, & 0 \leq |x| < \dfrac{1}{A} \\ \dfrac{1 + \ln(A\,|x|)}{1 + \ln A}, & \dfrac{1}{A} \leq |x| \leq 1 \end{cases}$$

**Diferença em relação à lei $\mu$:**

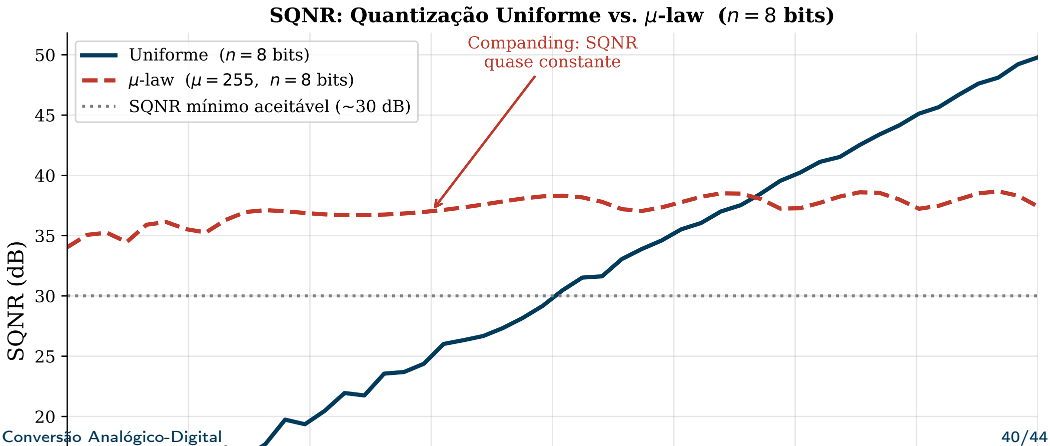
- A lei A é **linear** para amplitudes muito pequenas ($|x| < 1/A$) e logarítmica para amplitudes maiores
- A lei $\mu$ é logarítmica em toda a faixa
- Ambas garantem SQNR quase constante em uma ampla faixa de níveis de sinal

# Curvas de Compressão



# Comparação de SQNR: Uniforme vs. $\mu$-law



# Implementação em Python: $\mu$-law Companding

```python
import numpy as np

def mu_law_compress(x, mu=255):
    return np.sign(x) * np.log(1 + mu*np.abs(x)) / np.log(1 + mu)

def mu_law_expand(y, mu=255):
    return np.sign(y) * (1/mu) * ((1+mu)**np.abs(y) - 1)

def quantize_uniform(x, n_bits, v_min=-1.0, v_max=1.0):
    L   = 2**n_bits
    d   = (v_max - v_min) / L
    idx = np.clip(np.floor((np.clip(x, v_min, v_max)-v_min)/d), 0, L-1)
    return v_min + (idx.astype(int) + 0.5) * d

# Pipeline de companding
g          = 0.05 * np.sin(2*np.pi*np.linspace(0,100,1_000_000))
g_comp     = mu_law_compress(g)              # compressao
g_q_comp = quantize_uniform(g_comp, 8) # quantizacao uniforme
```

# Resumo: Conversão Analógico-Digital

## Resultado 1 — Amostragem (Nyquist-Shannon)

$$f_s > 2W \qquad \Longrightarrow \qquad G_s(f) = f_s \sum_{k=-\infty}^{\infty} G(f - k\, f_s)$$

## Resultado 2 — Ruído de Quantização

$$P_q = \frac{\Delta^2}{12}, \qquad \Delta = \frac{V_{pp}}{2^n}$$

## Resultado 3 — SQNR (senoide em faixa cheia)

# Agradecimentos

Obrigado pela atenção!

**Contato:**
daniel.araujo@unb.br

**Laboratório de Telecomunicações**
Universidade de Brasília

**Dúvidas?**

# Referências I

- J. G. Proakis and M. Salehi, *Communication Systems Engineering*. Upper Saddle River, NJ: Pearson, 2nd ed., 2014.
- B. P. Lathi and Z. Ding, *Modern Digital and Analog Communication Systems*. New York: Oxford University Press, 4th ed., 2009.
- C. E. Shannon, "Communication in the presence of noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1949.
- H. Nyquist, "Certain topics in telegraph transmission theory," *Transactions of the AIEE*, vol. 47, no. 2, pp. 617–644, 1928.

# Referências II

- W. R. Bennett, "Spectra of quantized signals," *Bell System Technical Journal*, vol. 27, no. 3, pp. 446–472, 1948.
- A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Upper Saddle River, NJ: Prentice Hall, 2nd ed., 1999.
- S. Haykin and M. Moher, *Communication Systems*. Hoboken, NJ: Wiley, 5th ed., 2013.