INSTITUTO FEDERAL
Santa Catarina

# Processamento Digital de Sinais

## - Aula 1 -

Professora: Dra. Luciana Menezes Xavier de Souza
e-mail: luciana.xavier@ifsc.edu.br

# Objetivos

- Conhecer e aplicar as ferramentas matemáticas para processamento discreto;

- Analisar e projetar filtros digitais utilizando softwares como ferramenta de desenvolvimento.

# Plano de Aula

1. **Classificação de Sinais**
2. **Sequências básicas**
3. **Operações sobre sequências**
4. Sistemas discretos
5. Sistemas lineares e invariantes no tempo
6. Equação de diferenças
7. Resposta de frequência de sistemas discretos
8. Transformada de Fourier em Tempo Discreto (DTFT – *Discrete-Time Fourier Transform*) de sequências)

# A disciplina

- Conceitos:

  - Sinais e sistemas discretos no tempo

  - Transformada de Fourier a tempo discreto

  - Amostragem e reconstrução de sinais contínuos no tempo

  - Transformada Z

- Aplicações:

  - Transformada Discreta de Fourier (DFT/FFT)

  - Estruturas de implementação e projeto de filtros digitais

# Bibliografia


[1] OPPENHEIM, A. V; SCHAFER, R. W; BUCK, J.R. Discrete-Time Signal Processing. 2.ed. New Jersey: Prentice-Hall, 1999.

[2] DINIZ, P. S. R; SILVA, E. A. B; LIMA NETTO, S. Processamento digital de sinais: Projeto e análise de sistemas. BOOKMAN, 2004.

[3] HAYES, M. H. Processamento Digital de Sinais. 1.ed. São Paulo: Bookman Companhia, 2006.

[4] INGLE, Vinay K.; PROAKIS, John G.. Digital Signal Processing using MATLAB. . Cole Publishing Company. 2000.

# Informações Gerais

- A comunicação com a professora:

  - Por e-mail:

    e-mail da instituição

    luciana.xavier@ifsc.edu.br

# Planejamento da disciplina

- Aulas expositivas e práticas;

  - Aulas semanais:
  - • Terça-feira 15:40 h até 17:30 h
  - • Sexta-feira 13:30 h até 15:20 h

# A disciplina: avaliações

- Provas individuais:
  1. Sinais/sistemas discretos no tempo, Transformada de Fourier a tempo discreto;
  2. Amostragem/reconstrução de sinais, Transformada Z.

- Projetos individuais:
  1. Projeto de filtros digitais.

# Planejamento da disciplina

- Avaliação da disciplina:

  - Duas avaliações e um projeto;

- Composição da avaliação:

$$\textbf{Nota final} = A1 \cdot 0{,}3 + A2 \cdot 0{,}3 + Projeto \cdot 0{,}4$$

# Planejamento da disciplina

- **PROIBIDO USAR** O CELULAR (EM HORÁRIO DE PROVA E NAS AULAS);

# Planejamento da disciplina

**Para ser considerado aprovado, o aluno deverá:**

- Ter o número máximo de faltas na disciplina igual a 25% do total de aulas;
- Ter média final **igual ou superior a 6**;

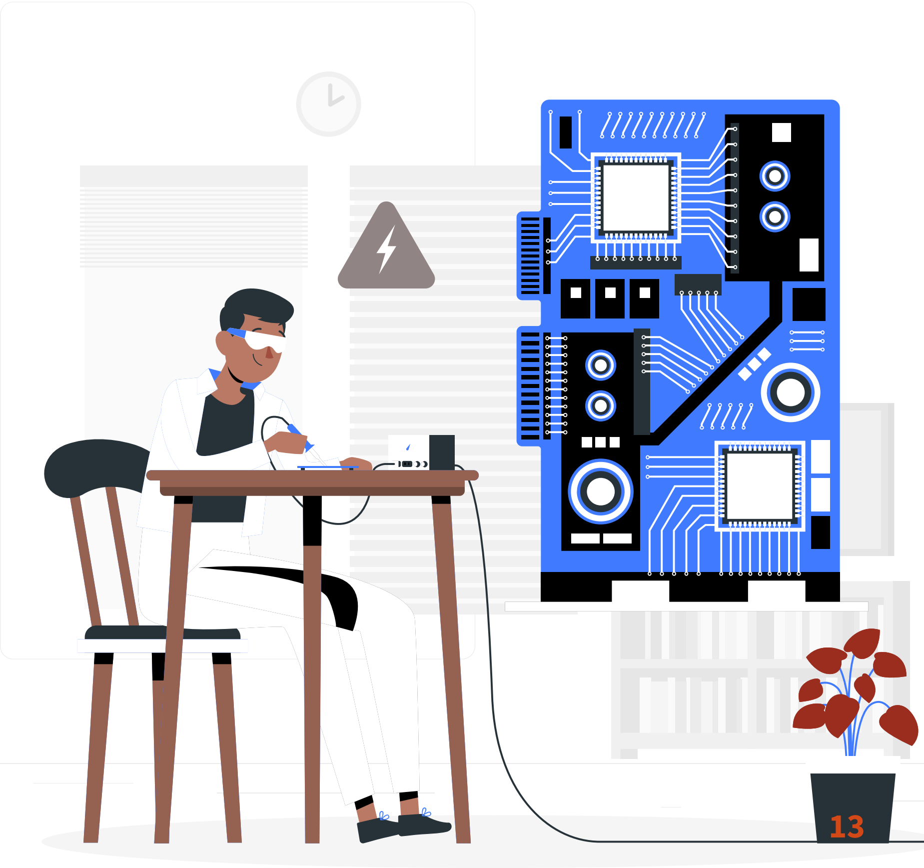
# Planejamento da disciplina

**Data da avaliação:**

- Avaliação 1: **10/06/2025 (Data poderá ser alterada)**.
- Avaliação 2: a ser marcada.

# Conteúdo

- Introdução da disciplina;
- Classificação de sinais;
- Sequências básicas;
- Operações sobre sequências.



# DSP – Uma Visão Histórica

- **Século XVII**: Técnicas Numéricas: Newton e o Método das Diferenças Finitas;
- **Século XVIII**: Euler, Bernoulli, Lagrange, Gauss (1805) base da FFT, Fourier (1822) representação em Série;
- **Até 1950**: Processamento analógico - executado com sistemas analógicos implementados com circuitos eletrônicos ou ainda com dispositivos mecânicos;
- **Até final da década de 1960**: computador digital era usado para aproximar ou simular um sistema de processamento analógico de sinais (não em tempo real).

# DSP – Uma Visão Histórica

- **Década de 1980**: A invenção e subsequente proliferação do microprocessador preparou o caminho para as **implementações de baixo custo** dos sistemas de processamento em tempo discreto de sinais;

- **A complexidade, a velocidade e a capacidade dos chips** de DSP têm crescido exponencialmente desde o **início da década de 1980**. Processamento paralelo e distribuído se tornaram uma tendência significativa no desenvolvimento de algoritmos de processamento de sinais;

- **E o futuro???** A chave para estar pronto para resolver novos problemas de processamento de sinais é, e sempre foi, um **profundo conhecimento da matemática fundamental dos sinais e sistemas e dos projetos e algoritmos** de processamento associados.

# Áreas do DSP

- A **interpretação de sinais** é uma área importante do processamento de sinais, em que o objetivo do processamento é obter uma **caracterização do sinal de entrada**;

- A **análise espectral**, baseada no uso da DFT e no uso de **modelos de sinais**, é outro aspecto particularmente importante do processamento de sinais, em que são identificadas quais frequências representam o sinal;

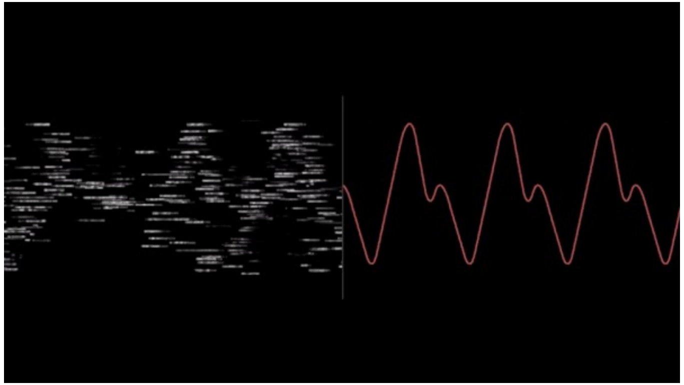
- A **modelagem de sinais** desempenha um papel importante na **compressão e codificação de dados**, bem como em sistemas preditivos;

- Outro tópico avançado de importância considerável é **o processamento adaptativo de sinais** em que o sistema auto ajusta seus parâmetros livres de acordo com o objetivo desejado.

# Definições

## Sinal

É uma função de uma ou mais variáveis, que carrega consigo alguma **informação sobre um fenômeno físico**;



# Definições

◎ **Sistema**

É uma entidade física que é capaz de **manipular um sinal de entrada**, gerando um sinal de saída ou **extraindo alguma informação** sobre o sinal de entrada;

- Geralmente são circuitos eletrônicos, microcontroladores, microprocessadores, entre outros.

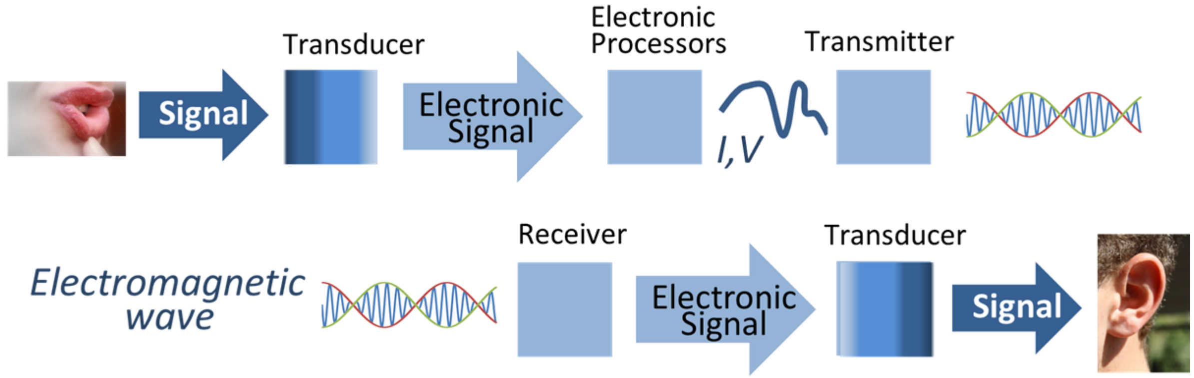
# Definições

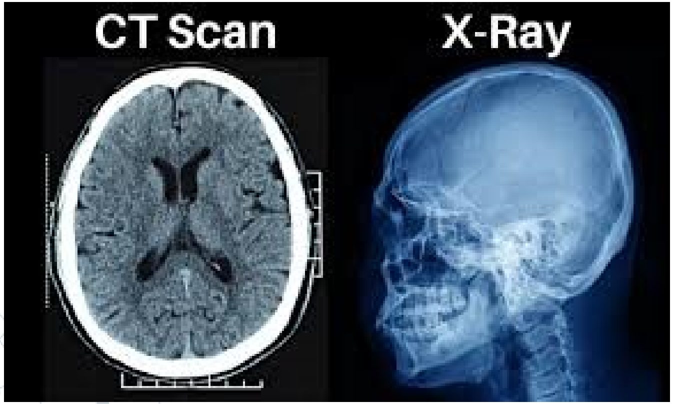
## ◎ Processamento

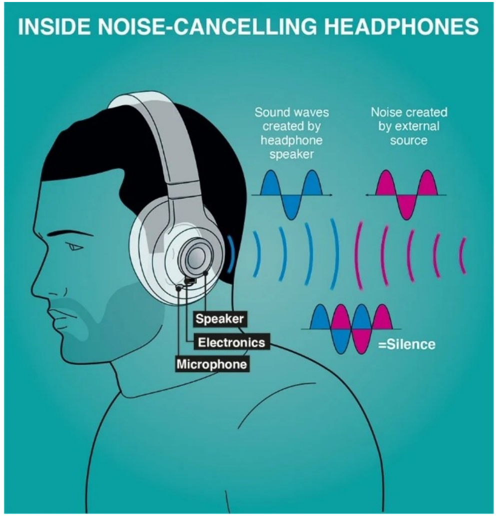
Representação, transformação e manipulação de sinais e da informação contida nos sinais.

- Basicamente, consiste na análise ou modificação de sinais utilizando teoria fundamental, aplicações e algoritmos, de forma a **extrair informações dos mesmos** e/ou torná-los mais **apropriados** para **alguma aplicação específica**.

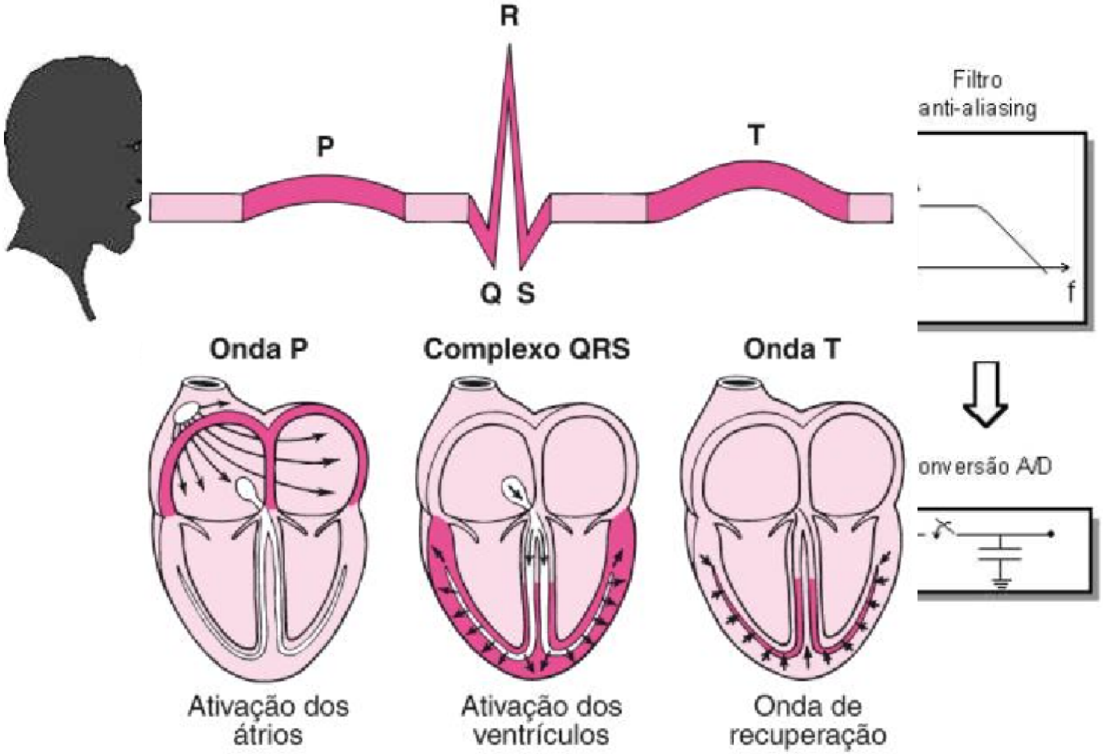
# **Processamento**: Exemplos



# **Processamento**: Exemplos



# Classificação de sinais

- **Amplitude**

- Valor do sinal, função de valores das componentes independentes;

- **Forma de onda**

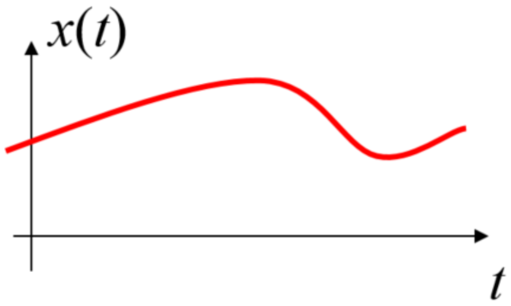
- Variação da amplitude com os valores das variáveis independentes;

- **Tempo**

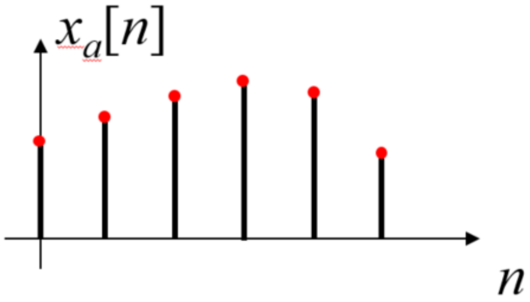
- Normalmente a variável independente em sinais 1D.

# Classificação de sinais

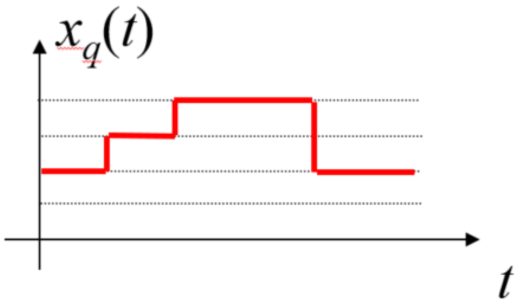
Sinal analógico contínuo



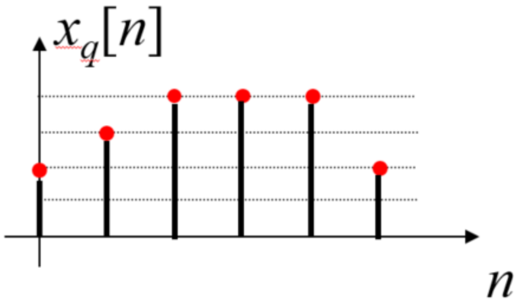

Sinal amostrado

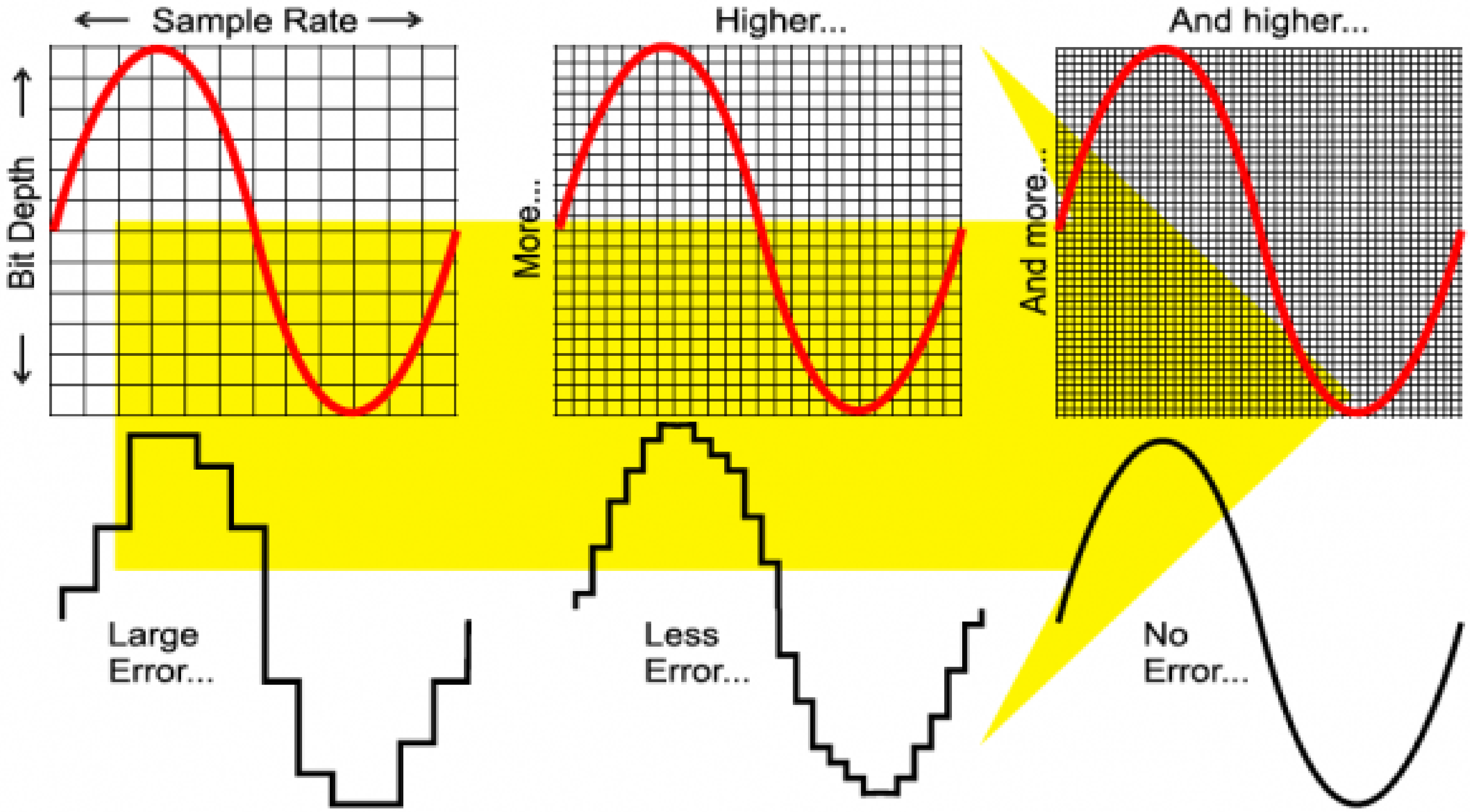



Sinal analógico quantizado



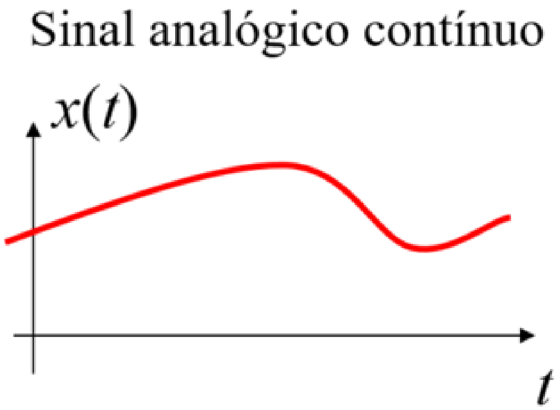

Sinal digital




Sample Rate
Bit Depth
Higher...
More...
And higher...
And more...
Large Error...
Less Error...
No Error...

# Classificação de sinais

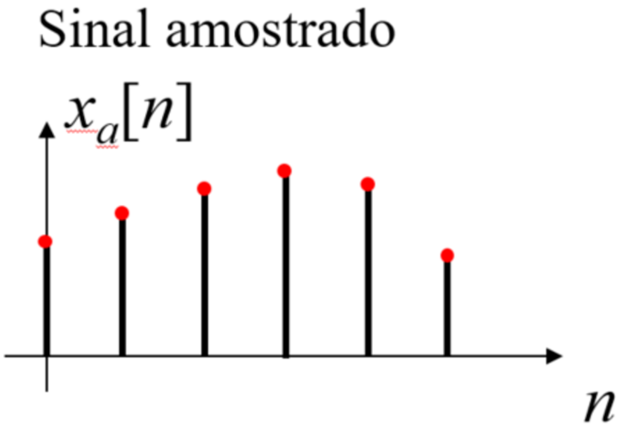
◎ ***Sinal analógico*** ou ***contínuo:*** definido para todo $t \in \mathbb{R}$;




- Sinal elétrico que varia no tempo e é proporcional à grandeza medida;
- Exemplos: corrente ou tensão elétrica.

# Classificação de sinais

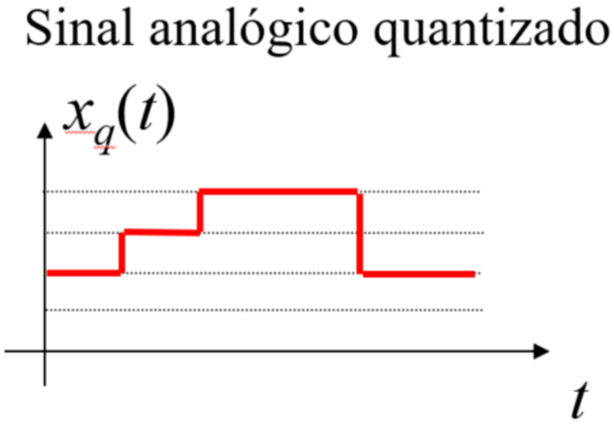
- ***Sinal amostrado:*** definido para valores de $t = nT$, sendo $T$ o período de amostragem e $n \in \mathbb{Z}$;




- Sinal analógico que foi medido em intervalos de tempo pré determinados;
- O **conjunto de pontos** de dados amostrados **reflete o sinal original** no digital.

# Classificação de sinais

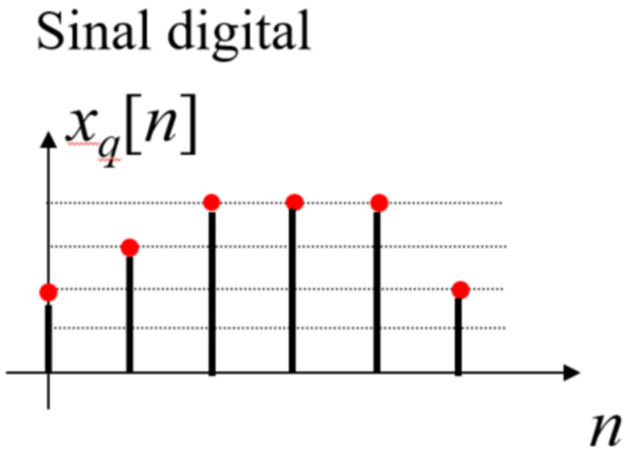
- ***Sinal quantizado:*** sinal contínuo para o qual as amplitudes são aproximadas por valores fixos discretos;




- **Sinal amostrado** que foi **arredondado** para um **conjunto finito** de valores;
- Cada nível é designado por um código digital "*n*" bits.

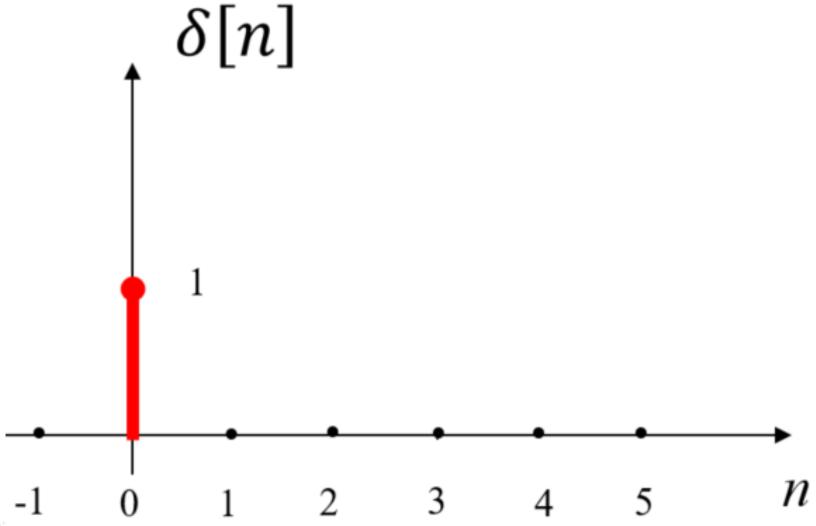
# Classificação de sinais

- ***Sinal digital*** ou ***numérico:*** as amplitudes e o tempo são discretos;



# Sequência básicas – Sinais importantes no tempo discreto

- **Impulso discreto** (ou unitário)



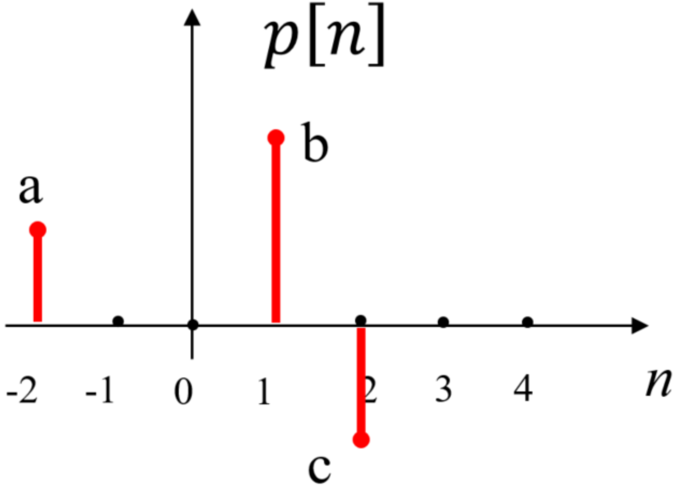

$$\delta[n] = \begin{cases} 0, & n \neq 0 \\ 1, & n = 0 \end{cases}$$

# Sequência básicas

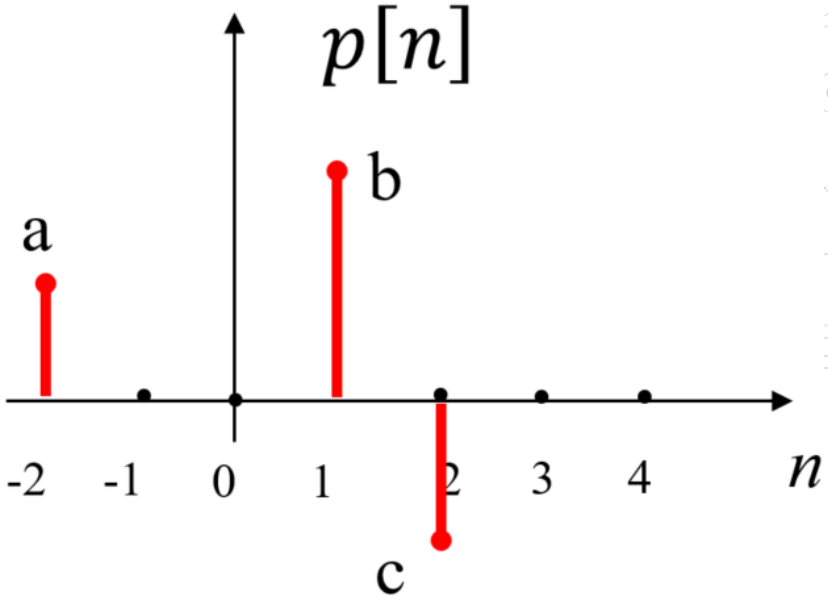
Qualquer sinal pode ser expresso como uma soma de impulsos:

$$x[n] = \sum_{k=-\infty}^{\infty} x[k].\delta[n-k]$$

Exercício:



$$x[n]=\sum_{k=-\infty}^{\infty}x[k].\delta[n-k]$$



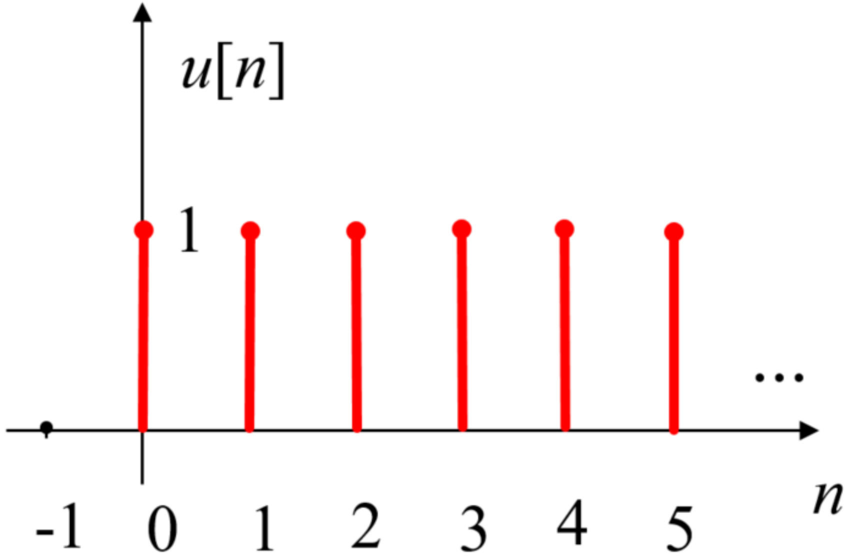

Solução:

$$p[n]=\mathrm{a}.\,\delta[n+2]+b.\,\delta[n-1]-c.\,\delta[n-2]$$

# Sequência básicas

- **Salto ou degrau unitário:**

$$u[n] = \begin{cases} 0, & n < 0 \\ 1, & n \geq 0 \end{cases}$$




Como: $$u[n] = \sum_{k=-\infty}^{n} \delta[k] = \sum_{k=0}^{\infty} \delta[n-k] = \delta[n] + \delta[n-1] + \delta[n-2] + \ldots$$

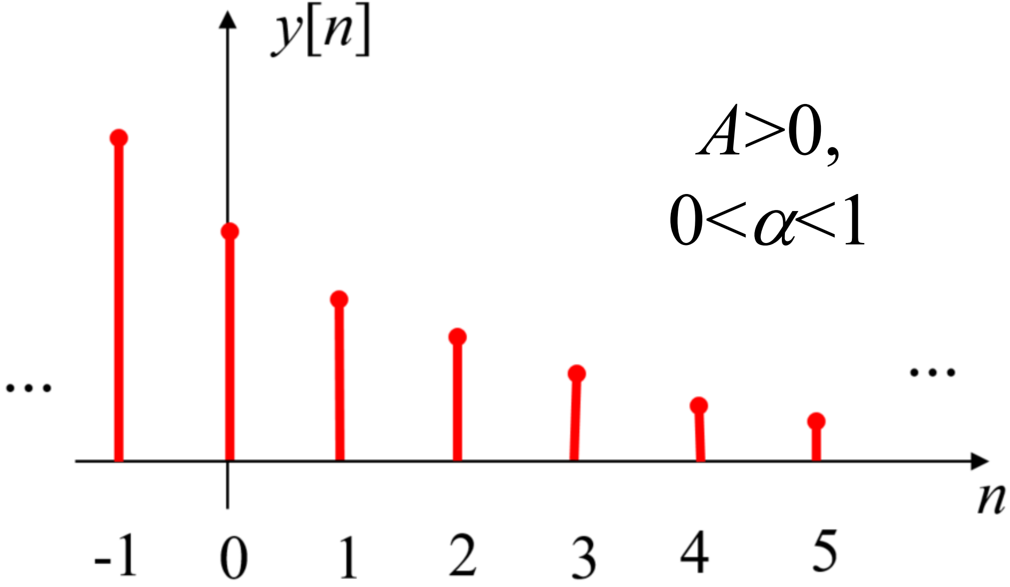
# Sequência básicas

- **Exponencial real**

$$x[n] = A \cdot \alpha^{n}$$

Se **α < 1: sinal decrescente**

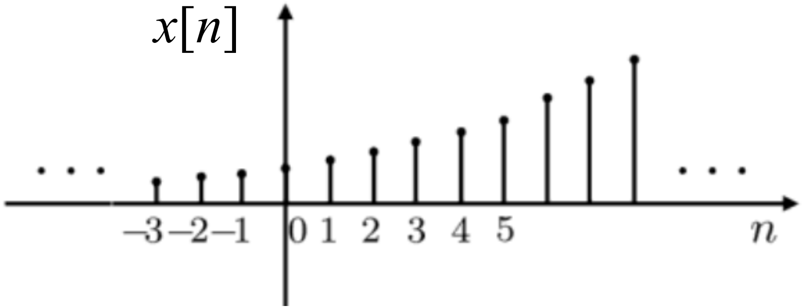
Se α > 1: sinal crescente



# Sequência básicas

- **Exponencial real**

$$x[n] = A \cdot \alpha^n$$




Se α < 1: sinal decrescente

Se **α > 1: sinal crescente**

# Sequência básicas

onde $\omega_0$ é a freq. angular, em rad/amostra

- **Exponencial complexa**

$$x[n] = A.\alpha^n \text{ com } A = |A|e^{j\varphi} \text{ e } \alpha = |\alpha|e^{j\omega_0}$$

$$x[n] = |A|e^{j\varphi}\,|\alpha|^n e^{j\omega_0 n} = |A|\,|\alpha|^n \underbrace{e^{j(\omega_0 n+\varphi)}}$$

Euler: $e^{j\omega n} = \cos(\omega n) + j\text{sen}(\omega n)$

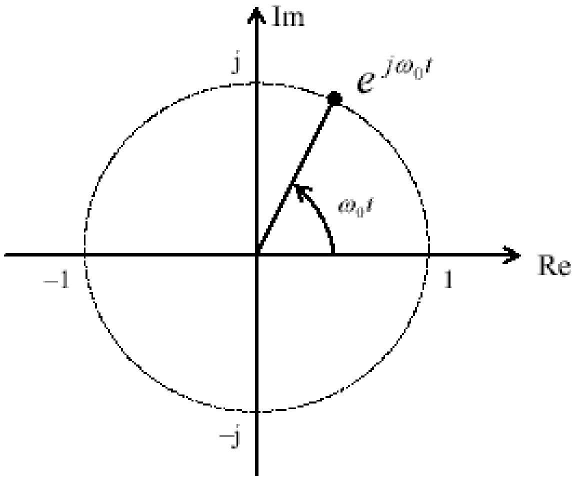
$$\boxed{x[n] = |A||\alpha|^n\left(\cos(\omega_0 n + \varphi) + j\,\text{sen}(\omega_0 n + \varphi)\right)}$$

# Sequência básicas

- **Exponencial complexa**

Se $|\alpha| = 1$, então:

$$x[n] = |A|\,|\alpha| e^{j(\omega_0 n+\varphi)} = |A|\, e^{j(\omega_0 n+\varphi)}$$




que é uma **senoide complexa** com frequência de oscilação $\omega_0$:

$$x[n] = |A|cos(\omega_0 n + \varphi) + j|A|sen(\omega_0 n + \varphi)$$

# Sinais periódicos digitais

- Diferentemente dos sinais contínuos, as sequências exponenciais complexas digitais com frequências $(\omega_0 + 2\pi r)$, com $r$ inteiro – são indistinguíveis entre si, pois

$$x[n] = Ae^{j(\omega_0+2\pi)n}$$
$$= Ae^{j\omega_0 n}.\underbrace{e^{j2\pi n}}_{e^{j2\pi n}=\cos(2\pi n)+j\sin(2\pi n)} = Ae^{j\omega_0 n}$$

OBS: Sabemos que o cosseno e o seno de múltiplos inteiros de 2π são:
cos $(2\pi n)$= 1 e sin $(2\pi n)$= 0
Portanto: $e^{j2\pi n}=1$
A simplificação final é: $Ae^{j2\pi n}=\mathrm{A}\cdot 1=\mathrm{A}$

# Sinais periódicos digitais

- Também vale para a sequência senoidal (sequência real):

$$
\begin{aligned}
x[n] &= A.cos[(\omega_0 + 2\pi r)n + \varphi] \\
&= A.cos[\omega_0 n + \varphi]
\end{aligned}
$$

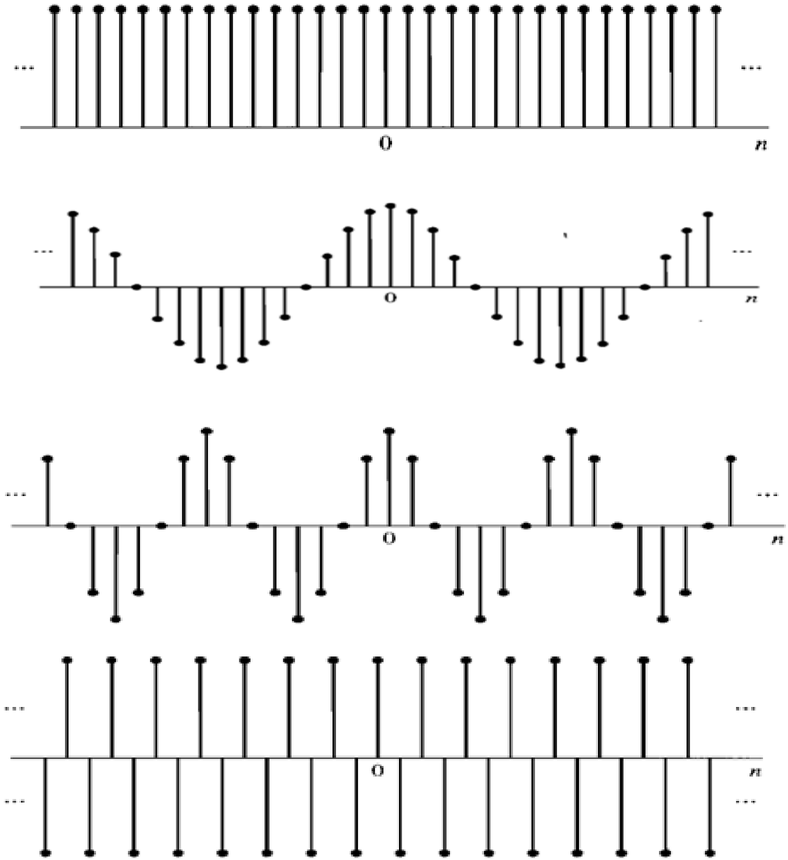
# Frequência Digital

- Além disso, em sinais contínuos como por exemplo $x(t) = A.cos(\Omega_0 t + \phi)$, a **frequência de oscilação do sinal aumenta com a frequência** $\Omega_0$.

- Mas na sequência senoidal digital equivalente $x[n] = A.cos(\omega_0 n + \varphi)$ :
  - $0 \leq \omega_0 \leq \pi$ : **aumento** da oscilação
  - $\pi \leq \omega_0 \leq 2\pi$ : **redução** da oscilação!

- OBS: período $T = \frac{2\pi}{\omega_0}$.

$$x[n] = A.\,cos(\omega_0 n + \varphi)$$



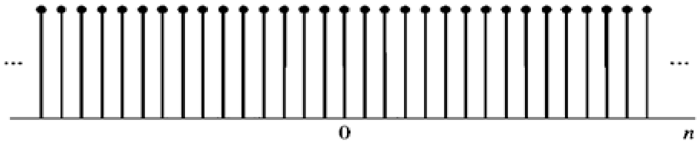
$\omega_0 = 0$

$\omega_0 = \pi/8$

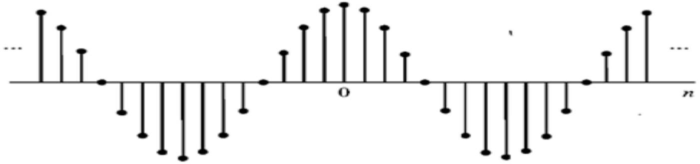
$\omega_0 = \pi/4$

$\omega_0 = \pi$

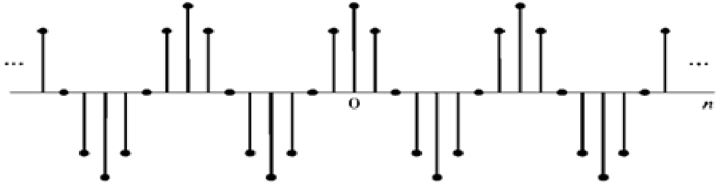
$$x[n] = A.\cos(\omega_0 n + \varphi)$$



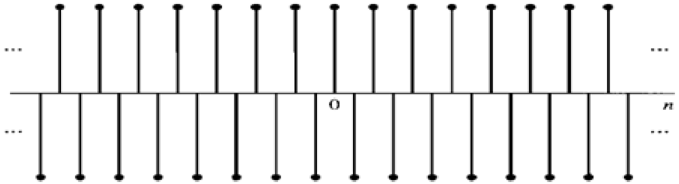

$$\omega_0 = 2\pi$$




$$\omega_0 = 15\pi/8$$




$$\omega_0 = 7\pi/4$$




$$\omega_0 = \pi$$

# Periodicidade de uma sequência

- Uma sequência (digital ou contínua) é dita periódica se $x[n] = x[n - N]$, para todo $n$.

- No caso da sequência senoidal $x[n] = A.cos(\omega_0 n + \varphi)$ :

$$= A.cos(\omega_0(n - N) + \varphi)$$
$$= A.cos(\omega_0 n - \omega_0 N + \varphi)$$

- Só será verdade se: $\omega_0 \text{ N} = 2\pi \, m$

$$\therefore \ \omega_0 = 2\pi m \, / \, \text{N} \text{ , com N e } m \in \text{Z}$$

- Apenas para alguns valores de $\omega_0$ múltiplos inteiros de $\pi$.

# Operações sobre sequências

1. Soma: $w[n] = x[n] + y[n]$

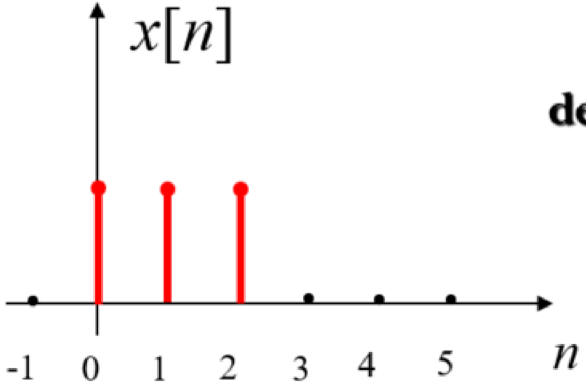
2. Multiplicação por constante: $k[n] = \alpha \cdot x[n]$

3. Deslocamento no tempo: $y[n] = x[n - n_0]$

4. Multiplicação entre sequências: $z[n] = x[n] \cdot y[n]$

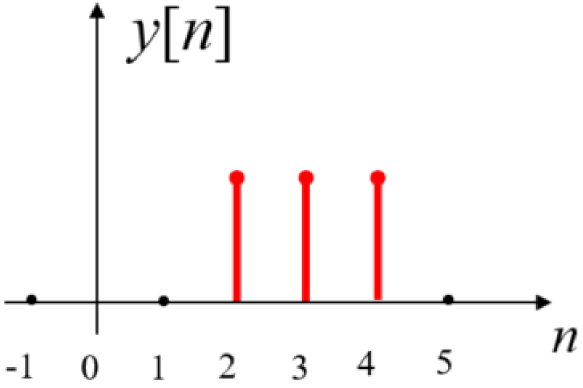
# Operações sobre sequências

Deslocamento no tempo: $y[n] = x[n - n_0]$




**Exemplo de deslocamento no tempo:**

$y[n] = x[n\text{-}2]$



# Exercícios

- Usamos várias sequências elementares em processamento de sinais digitais para propósitos de análise. Suas definições e representações no MATLAB seguem.

1. **Sequência de amostra unitária**:

$$\delta(n) = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases} = \left\{ \dots, 0, 0, \underset{\uparrow}{1}, 0, 0, \dots \right\}$$

No MATLAB, a função zeros(1,N) gera um vetor linha de N zeros, que pode ser usado para implementar $\delta(n)$ em um intervalo finito. No entanto, a relação lógica $n$==0 é uma maneira elegante de implementar $\delta(n)$.Por exemplo, para implementar

$$\delta(n - n_0) = \begin{cases} 1, & n = n_0 \\ 0, & n \neq n_0 \end{cases}$$

no intervalo $n_1 \leq n_0 \leq n_2$, usaremos a seguinte função do MATLAB.

```matlab
%Unit Sample Sequence

function [x,n] = impseq(n0,n1,n2)
% Generates x(n) = delta(n-n0); n1 <= n <= n2
% ----------------------------------------------
% [x,n] = impseq(n0,n1,n2)
%
n = [n1:n2];
x = [(n-n0) == 0];
```

# 2. Sequência de step unitária:

$$u(n) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases} = \{\ldots, 0, 0, \underset{\uparrow}{1}, 1, 1, \ldots\}$$

No MATLAB, a função ones(1,N) gera um vetor linha de N uns. Ela pode ser usada para gerar $u(n)$ em um intervalo finito. Mais uma vez, uma abordagem elegante é usar a relação lógica n>=0. Para implementar

$$u(n - n_0) = \begin{cases} 1, & n \geq n_0 \\ 0, & n < n_0 \end{cases}$$

no intervalo $n_1 \leq n_0 \leq n_2$, usaremos a seguinte função MATLAB.

## 2. **Sequência de step unitária**:

```
%Unit step sequence

function [x,n] = stepseq(n0,n1,n2)
% Generates x(n) = u(n-n0); n1 <= n <= n2
% ------------------------------------------
% [x,n] = stepseq(n0,n1,n2)
%
n = [n1:n2];
x = [(n-n0) >= 0];
```

## 3. **Sequência exponencial de valor real**:

$$x(n) = a^n, \forall n; \quad a \in \mathbb{R}$$

No MATLAB, um operador de matriz ".^" é necessário para implementar uma sequência exponencial real.

Por exemplo, para gerar $x(n) = (0{,}9)^n$, $0 \leq n \leq 10$, precisaremos do seguinte script do MATLAB:

```
n = [0:10];
x = (0.9).^n;
stem(x)
```

4. **Sequência exponencial de valor complexo**:

$$x(n) = e^{(\sigma + j\omega_0)n}, \forall n$$

onde σ produz uma atenuação (se <0) ou amplificação (se >0) e $\omega_0$ é a frequência em radianos. Uma função *exp* do MATLAB é usada para gerar sequências exponenciais.

Por exemplo, para gerar $x(n)=exp[(2+j3)*n]$, $0 \le n \le 10$, precisaremos do seguinte script do MATLAB:

```
n = [0:10];
x = exp((2+3j)*n);
```

5. **Sequência sinusoidal**:

$$x(n) = A\cos(\omega_0 n + \theta_0), \forall n$$

onde A é uma amplitude e $\theta_0$ é a fase em radianos. A função no MATLAB cos (ou sin) é usada para gerar sequências senoidais.

Por exemplo, para gerar $x(n) = 3*\cos(0{,}1\pi n + \pi/3) + 2\ \sin(0{,}5\pi n)$, $0 \le n \le 10$, precisaremos do seguinte script MATLAB:

```
n = [0:10];
x = 3*cos(0.1*pi*n+pi/3) + 2*sin(0.5*pi*n);
stem(x)
```

## Exercício 2

Gere e plote cada uma das seguintes sequências no intervalo indicado.

a) $x(n) = 2\delta(n+2) - \delta(n-4), \quad -5 \leq n \leq 5.$

```
n = [-5:5];
x = 2*impseq(-2,-5,5) - impseq(4,-5,5);
stem(n,x);
title('Sequence problema 2.a')
xlabel('n');
ylabel('x(n)');
```

## Exercício 2

Gere e plote cada uma das seguintes sequências no intervalo indicado.

b) $x(n) = n\,[u(n) - u(n-10)] + 10e^{-0.3(n-10)}[u(n-10) - u(n-20)]$, $0 \leq n \leq 20.$

```
n = [0:20];
x1 = n.*(stepseq(0,0,20)-stepseq(10,0,20));
x2 = 10*exp(-0.3*(n-10)).*(stepseq(10,0,20)-stepseq(20,0,20));
x = x1+x2;
stem(n,x);
title('Sequencia problema 2.b')
xlabel('n');
ylabel('x(n)');
```

## Exercício 2

Gere e plote cada uma das seguintes sequências no intervalo indicado.

c) $x(n) = \cos(0.04\pi n) + 0.2w(n),\ 0 \leq n \leq 50.$

```
n = [0:50];
x = cos(0.04*pi*n)+0.2*randn(size(n));
stem(n,x);
title('Sequencia no problema 2.c')
xlabel('n');
ylabel('x(n)');
```

## Exercício 2

Gere e plote cada uma das seguintes sequências no intervalo indicado.

d) $\tilde{x}(n) = \{\ldots, 5, 4, 3, 2, 1, 5, 4, 3, 2, 1, 5, 4, 3, 2, 1, \ldots\}; \;\; -10 \leq n \leq 9.$

```
n = [-10:9];
x = [5,4,3,2,1];
xtilde = x' * ones(1,4);
xtilde = (xtilde(:))';
stem(n,xtilde);
title('Sequência problema 2.d')
xlabel('n');
ylabel('xtilde(n)');
```

## Exercício 2.2

- Seja $x(n) = \{1, 2, 3, 4, 5, 6, 7, 6, 5, 4, 3, 2, 1\}$. Determine e plote as seguintes sequências.

a) $x_1(n) = 2x(n-5) - 3x(n+4)$;

b) $x_2(n) = x(3-n) + x(n)\,x(n-2)$;

A sequência $x(n)$ é diferente de zero em $-2 \leq n \leq 10$.

## Exercício 2.2 a) $x_1(n) = 2x(n-5) - 3x(n+4)$;

A primeira parte é obtida deslocando $x(n)$ por 5 e a segunda parte deslocando $x(n)$ por $-4$. Esse deslocamento e a adição podem ser facilmente feitos usando as funções sigshift e sigadd.

```
 n = -2:10;
 x = [1:7,6:-1:1];

[x11,n11] = sigshift(x,n,5);
[x12,n12] = sigshift(x,n,-4);
[x1,n1] = sigadd(2*x11,n11,-3*x12,n12);
stem(n1,x1);
title('Sequencia problema 2.2a');
xlabel('n');
ylabel('x1(n)');
```

## Exercício 2.2 b) $x_2(n) = x(3 - n) + x(n)\,x(n - 2);$

- O primeiro termo pode ser escrito como $x(-(n - 3))$. Portanto, ele é obtido primeiro dobrando $x(n)$ e depois deslocando o resultado em 3.
- A segunda parte é uma multiplicação de $x(n)$ e $x(n-2)$, ambos com o mesmo comprimento, mas suporte diferente (ou posições de amostra).

Essas operações podem ser facilmente feitas usando as funções sigfold e sigmult.

```
[x21,n21] = sigfold(x,n);
[x21,n21] = sigshift(x21,n21,3);
[x22,n22] = sigshift(x,n,2);
[x22,n22] = sigmult(x,n,x22,n22);
[x2,n2]   = sigadd(x21,n21,x22,n22);
stem(n2,x2);
title('Sequencia problema 2.2b');
xlabel('n');
ylabel('x2(n)');
```

# Exercício 2.3

Gere o sinal de valor complexo

$$x(n) = e^{(-0,1+j0,3)n}, \ -10 \le n \le 10$$

plote sua magnitude, fase, a parte real e a parte imaginária em quatro sub gráficos separados.

```
n = [-10:1:10]; alpha = -0.1+0.3j;
x = exp(alpha*n);

subplot(2,2,1); stem(n,real(x));            title('real part');      xlabel('n');
subplot(2,2,2); stem(n,imag(x));            title('imaginary part');xlabel('n');
subplot(2,2,3); stem(n,abs(x));             title('magnitude part');xlabel('n');
subplot(2,2,4); stem(n,(180/pi)*angle(x));  title('phase part');     xlabel('n');
```