



# CONCEITOS DE LINGUAGEM DE PROGRAMAÇÃO

FAPESC – DESENVOLVEDORES PARA TECNOLOGIA DA INFORMAÇÃO

HABNER FABRÍCIO BOESING

[habner.boesing@unoesc.edu.br](mailto:habner.boesing@unoesc.edu.br)

# OBJETIVOS

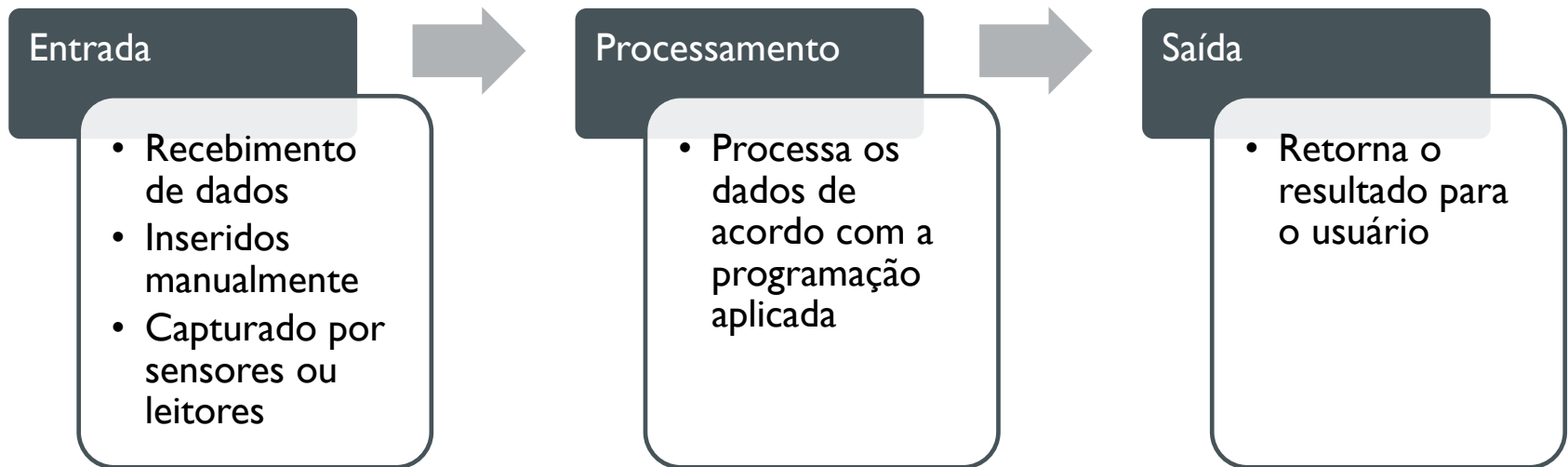
- Conceito de dados e informação
- Processamento de dados
- Lógica, proposições e algoritmos
- Conceitos de linguagem de programação;
- Tipos de Linguagens - Classificações;
  - Grau de Abstração
  - Estrutura
- Características da Programação Desktop e Web

# DADO X INFORMAÇÃO

- Dados não possuem uma classificação ou ordenação lógica.
- Informação é formada com a classificação e união de dados.

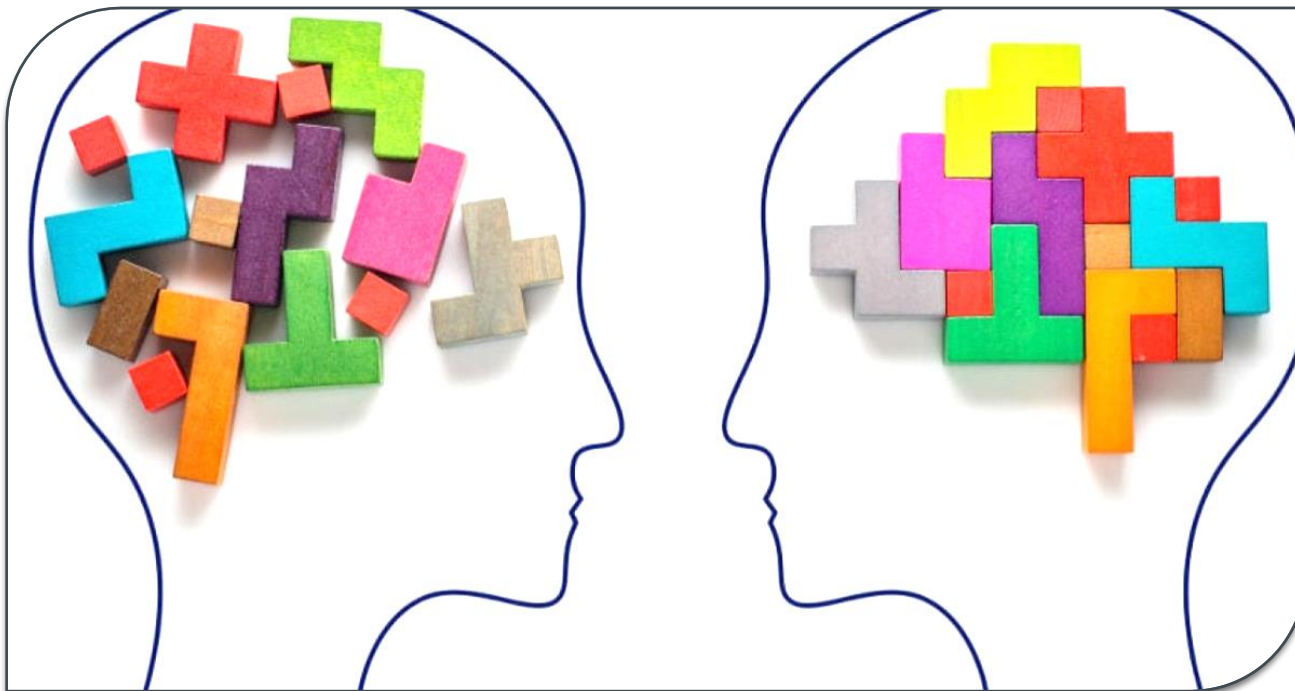
Tipo	Dado	Informação
Números	11202220	22/11/2020
Letras	R, O, P, D, E	PEDRO

# PROCESSAMENTO DE DADOS



# LÓGICA

- Área da filosofia que visa estudar a estrutura formal dos enunciados (proposições) e suas regras.
- Busca atribuir valores verdadeiros ou falsos para proposições.



# PROPOSIÇÕES

Proposição é uma **sentença declarativa**, seja ela expressa de forma **afirmativa** ou **negativa**, na qual podemos atribuir um valor lógico “**V**” (**verdadeiro**) ou “**F**” (**falso**)

## Exemplos:

1) Brasília é a capital do Brasil.

■ Sentença afirmativa. Verdadeira

2) A Argentina não é um país pertencente ao continente Africano.

■ Sentença negativa. Verdadeira

3) 10 é um número par positivo.

■ Sentença afirmativa. Verdadeira

4)  $7+5 = 10$ .

■ Sentença afirmativa. Falsa

# ALGORITMOS

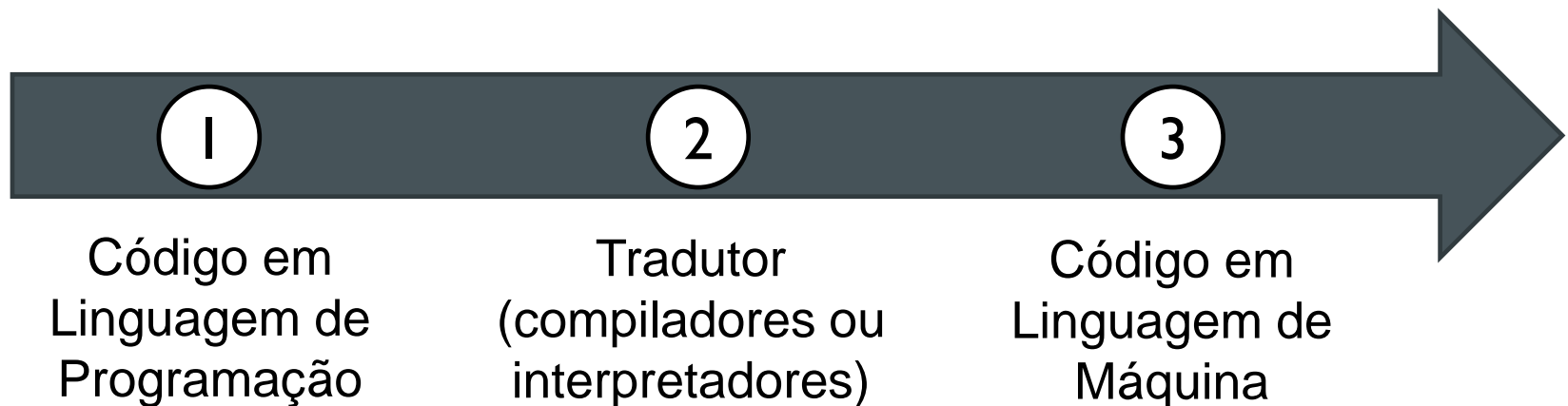
- Sequência finita de ações executáveis que visam obter uma solução para um determinado tipo de problema.
- Utilizando-se das proposições conseguimos montar algoritmos para solucionar problemas.

Exemplo:

- Problema: Trocar o pneu de um carro
- Solução: Utilizando de proposições defina uma sequência de ações que podem ser realizadas para solucionar este problema.
- Algoritmo: Organizando as proposições numa ordem lógica é possível criar o algoritmo que irá solucionar este problema.
- **Qual é a solução que pode ser utilizada? Defina e enumere os passos para realizar este procedimento.**

# LINGUAGEM DE PROGRAMAÇÃO

- Os algoritmos devem ser criados utilizando um conjunto de palavras específicas da linguagem de programação (sintaxe).
- A programação dos algoritmos compõem o código fonte de um software e traduzidos e interpretados pelo hardware.





# SINTAXE X SEMÂNTICA

- **Sintaxe:** regras que regem a composição dos textos (comandos) de uma linguagem de programação.

Exemplo: Sintaxe utilizada na linguagem de programação PHP

*echo "Olá Mundo!"*

- **Semântica:** Significado que cada comando terá ao ser interpretado pelo compilador da linguagem de programação.

Exemplo: Entendemos que por meio do comando echo o PHP fará com que uma mensagem seja exibida na tela, neste caso:

*Olá Mundo!*

“É como se aprendêssemos uma linguagem em comum que possa ser interpretada por humanos e computadores ao mesmo tempo”

# COMPARATIVO DE COMANDOS

- Os três comandos abaixo foram escritos em linguagens de programação diferentes, mas exibem a mesma mensagem na tela
- Resumindo: sintaxes diferentes, mas semânticas iguais.



```
echo "Olá, Mundo!";
```



```
System.out.println("Olá, Mundo!");
```



```
writeln('Olá, Mundo!');
```

# TIPOS DE LINGUAGENS

## GRAU DE ABSTRAÇÃO

Existem 2 tipos principais de classificações de linguagem quanto ao seu grau de abstração:

### **Alto Nível:**

- Mais próximo à linguagem humana.
- Não precisa conhecer características do processador.
- Exemplos: Java, C#, PHP, Python

### **Baixo Nível:**

- Mais próximo à linguagem de máquina.
- É necessário conhecer as características do hardware
- Exemplo: Assembly

# TIPOS DE LINGUAGENS

## GRAU DE ABSTRAÇÃO

### Exemplos:

#### Alto Nível:



```
1 print("Hello World!")
```

#### Baixo Nível:

# Assembly

```
1 lea si, string
2 call printf
3 hlt
4 string db "Ola mundo!", 0
5 printf PROC
6     mov AL, [SI]
7     cmp AL, 0
8     je pfend
9     mov AH, 0Eh
10    int 10h
11    inc SI
12    jmp printf
13 pfend:
14    ret
15 printf ENDP
```

# TIPOS DE LINGUAGENS

## ESTRUTURA

Existem 2 tipos principais de estrutura de linguagens de programação:

### **Estruturada:**

- Estruturas básicas de controle: sequência, condição e repetição.
- Programa orientada a procedimentos.
- Possui melhor desempenho de processamento.
- Exemplos: *Lisp*, *Cobol*, *C*, *Pascal*.

### **Orientada à objeto:**

- Estruturas relacionais: objetos e métodos.
- Programação orienta a objetos.
- Possui melhor entendimento de linguagem, em detrimento do processamento.
- Exemplos: *Java*, *C#*, *Python*, *Ruby*.

# TIPOS DE LINGUAGENS ESTRUTURA

## Estruturada:



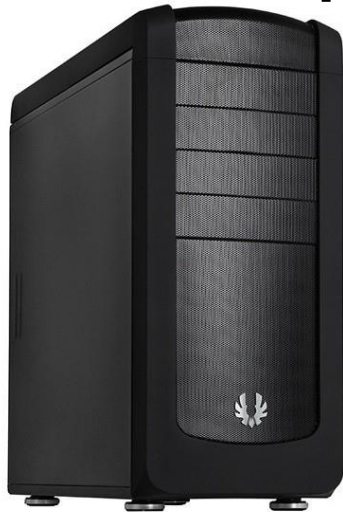
## Orientada à objeto:



# TIPOS DE PROGRAMAÇÃO

Existem 2 tipos principais de programação referentes ao seu modo de implementação:

**Desktop**



**X**

**Web**



# PROGRAMAÇÃO DESKTOP X WEB

## CARACTERÍSTICAS

### **Desktop:**

- Geralmente de acesso local, tendo dificuldades para conexões remotas.
- Depende do desempenho da máquina em que está sendo executada.
- Altamente dependente da compatibilidade de hardware e software.
- Suporta maiores cargas de acesso por se tratar de um dispositivo local.
- Manutenções podem exigir um esforço extra

### **Web:**

- Acesso global
- Depende do desempenho do servidor
- Costuma ser acessível por qualquer dispositivo que possua um navegador compatível e um acesso à rede
- Pode sofrer com sobrecarga de acessos
- Manutenção centralizada, facilitando a aplicação das modificações



# PROGRAMAÇÃO DESKTOP

## VANTAGENS E DESVANTAGENS

### Vantagens:



- Consegue trabalhar melhor com especificação de hardwares.
- Independe do acesso à internet.
- Permite processamento superior por utilizar melhor os recursos locais.
- Suporta altas taxas de tráfegos de dados.

### Desvantagens:



- Acessibilidade
- Distribuição da aplicação é crítica por necessitar por muitas vezes atualização máquina à máquina
- Trabalho extra para desenvolver softwares multiplataforma

# PROGRAMAÇÃO WEB

## VANTAGENS E DESVANTAGENS

### Vantagens:



- Pode ser acessível em qualquer parte do mundo.
- Atualização facilitada por centralizar a distribuição de recursos.
- Desenvolvimento facilitado para multiplataforma
- Podem rodar em dispositivos mais simples, sem exigir tanto processamento local

### Desvantagens:



- Dificuldades para se comunicar com hardware local
- Sistema inoperante em casos de problemas de acesso à web
- Sistema centralizado, pode afetar diversos usuário ao mesmo tempo
- Sobrecargas de acesso (DDoS)

# EXERCÍCIOS

- 1) Cite o nome de 5 linguagens de programação desktop.
- 2) Cite o nome de 5 linguagens de programação mobile.
- 3) Cite o nome de 5 linguagens de programação web.
- 4) Na programação web os desenvolvedores costumam ser divididos em front-end, back-end e full stack. Explique qual é a diferença entre cada um deles.
- 5) Quais são as linguagens de programação mais utilizadas por programadores front-end?
- 6) Quais são as linguagens de programação mais utilizadas por programadores back-end?
- 7) O que é pseudocódigo?
- 8) Para que o pseudocódigo é utilizado?
- 9) Quais são as vantagens de se utilizar pseudocódigo ao invés de linguagem de programação para iniciantes?
- 10) Quais programas podem ser utilizados para executar pseudocódigo?