



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
RIO GRANDE DO NORTE  
Campus Nova Cruz



# Fundamentos de Lógica e Algoritmos

## Aula 2.5 – Conceitos Básicos de Algoritmos

Prof. Dr. Bruno Moreno  
[bruno.moreno@ifrn.edu.br](mailto:bruno.moreno@ifrn.edu.br)

# Introdução

- Um algoritmo pode ser definido como uma sequência **finita** de passos (instruções);
- O objetivo de um algoritmo é **resolver um determinado problema**;
- Um algoritmo **eficiente** deve:
  - Ter ações **simples e sem ambiguidade**;
  - Ter ações **ordenadas** corretamente;
  - Ser finito (ter início e fim);
- Um algoritmo também pode ser chamado de **rotina**.

# Introdução

- Um algoritmo pode ser representado por um pseudocódigo:

---

**Algoritmo 1** Troca de pneu do carro.

---

- 1: desligar o carro
- 2: pegar as ferramentas (chave e macaco)
- 3: pegar o estepe
- 4: suspender o carro com o macaco
- 5: desenroscar os 4 parafusos do pneu furado
- 6: colocar o estepe
- 7: enroscar os 4 parafusos
- 8: baixar o carro com o macaco
- 9: guardar as ferramentas

- Observe que:
  - Este algoritmo está em português;
  - Este algoritmo possui um nome, início e fim;
  - Atenção: a numeração de linha está sendo representada apenas para melhor compreensão.

# Introdução

- Algoritmo para pegar um ônibus:

---

## **Algoritmo 2** Pegar um onibus.

---

- 1: ir até a parada
  - 2: **enquanto** ônibus não chega **faça**
  - 3:     esperar ônibus
  - 4: **fim-enquanto**
  - 5: subir no ônibus
  - 6: pegar passagem
  - 7: **se** não há passagem **então**
  - 8:     pegar dinheiro
  - 9: **fim-se**
  - 10: pagar o cobrador
  - 11: troco  $\leftarrow$  dinheiro - passagem
  - 12: **enquanto** banco não está vazio **faça**
  - 13:     ir para o próximo
  - 14: **fim-enquanto**
  - 15: sentar
  - 16: ...
-

# Introdução

- Um algoritmo deve ser capaz de:
  - **Ler e escrever** dados;
  - **Avaliar** expressões **algébricas, relacionais e lógicas**;
  - **Tomar decisões** com base nos resultados das expressões avaliadas;
  - **Repetir** um conjunto de ações de acordo com uma condição;

# Introdução

- Um algoritmo deve ser capaz de:
  - **Ler e escrever** dados;
  - **Avaliar** expressões **algébricas, relacionais e lógicas**;  
COMANDOS DE DECISÃO → ESTRUTURAS DE DECISÃO
  - **Tomar decisões** com base nos resultados das expressões avaliadas;
  - **Repetir** um conjunto de ações de acordo com uma condição;

# Introdução

- Um algoritmo deve ser capaz de:
  - **Ler e escrever** dados;
  - **Avaliar** expressões **algébricas, relacionais e lógicas**;
  - **Tomar decisões** com base nos resultados das expressões avaliadas;
  - **Repetir** um conjunto de ações de acordo com uma condição;

COMANDOS DE REPETIÇÃO → ESTRUTURAS DE REPETIÇÃO

# Introdução

- Considere o algoritmo a seguir:

---

## **Algoritmo 1** Troca de pneu do carro.

---

- 1: desligar o carro
  - 2: pegar as ferramentas (chave e macaco)
  - 3: pegar o estepe
  - 4: suspender o carro com o macaco
  - 5: desenroscar os 4 parafusos do pneu furado
  - 6: colocar o estepe
  - 7: enroscar os 4 parafusos
  - 8: baixar o carro com o macaco
  - 9: guardar as ferramentas
- 

- Este algoritmo possui comandos de decisão ou de repetição?



# Introdução

- Identifique comandos que representam **decisão** ou **repetição** no algoritmo ao lado
  - Atente para as palavras-chave:
    - Se ... então;
    - Enquanto ... faça;

---

## Algoritmo 2 Pegar um onibus.

---

```
1: ir até a parada
2: enquanto ônibus não chega faça
3:     esperar ônibus
4: fim-enquanto
5: subir no ônibus
6: pegar passagem
7: se não há passagem então
8:     pegar dinheiro
9: fim-se
10: pagar o cobrador
11: troco ← dinheiro - passagem
12: enquanto banco não está vazio faça
13:     ir para o próximo
14: fim-enquanto
15: sentar
16: ...
```

---

# Partes de um Algoritmo

- Um algoritmo é constituído pelo menos das 3 partes, sendo elas:
  - 1- Entrada de dados;
  - 2- Processamento de dados;
  - 3- Saída de dados;



# Partes de um Algoritmo

## 1- Entrada de dados

- Informações necessárias para execução do algoritmo;
- São fornecidas quando o programa é executado ou podem estar embutidas dentro do mesmo;

## 2- Processamento

- São avaliadas todas as expressões algébricas, relacionais e lógicas;
- São avaliadas todas as estruturas de controle existentes no algoritmo
  - De condição e/ou repetição.

## 3- Saída de dados

- Todos os resultados do processamento (ou parte deles) são enviados para um ou mais dispositivos de saída
  - Monitor, impressora, ou até mesmo a própria memória do computador

# Partes de um Algoritmo

- Exemplo: cálculo da área de uma circunferência
  - $A = \pi.R^2$
  - Para calcular  $A$ , é necessário saber os valores de  $\pi$  e  $R$ ;
  - $\pi$  tem valor constante: 3,14
    - Pode ser definido dentro do próprio algoritmo;
  - $R$  é o raio e será informado pelo usuário
    - Dispositivo de entrada: teclado;
  - O valor de  $A$  será escrito no monitor
    - Dispositivo de saída.

# Partes de um Algoritmo

- Exemplo: cálculo da área de uma circunferência
  - $A = \pi.R^2$

---

## Algoritmo 3 Calcula Área de uma Circunferência.

---

|                             |                                |
|-----------------------------|--------------------------------|
| 1: $\pi \leftarrow 3.14$    | {entrada para o processamento} |
| 2: <b>leia</b> $R$          | {entrada para o processamento} |
| 3: $A \leftarrow \pi * R^2$ | {processamento}                |
| 4: <b>escreva</b> $A$       | {saída}                        |

---

# Partes de um Algoritmo

- Exemplo: cálculo da área de uma circunferência
  - $A = \pi \cdot R^2$       **Quase sempre um algoritmo possui variáveis.**

---

## Algoritmo 3 Calcula Área de uma Circunferência.

---

- |                             |                                |
|-----------------------------|--------------------------------|
| 1: $\pi \leftarrow 3.14$    | {entrada para o processamento} |
| 2: <b>leia</b> $R$          | {entrada para o processamento} |
| 3: $A \leftarrow \pi * R^2$ | {processamento}                |
| 4: <b>escreva</b> $A$       | {saída}                        |
-

# Pseudocódigos

- Os algoritmos vistos até aqui não são considerados códigos de programas
  - i.e. não utilizam uma linguagem de programação específica;
- São considerados pseudocódigos;
- Pseudocódigos são comumente utilizados para representar um algoritmo de forma genérica
  - Pode ser implementado em qualquer LP.

# Pseudocódigos

- O pseudo código a seguir contém conceitos que a maioria das LPs contém
  - Leitura de dados;
  - Escrita de dados;
  - Estruturas condicionais.

---

## Algoritmo 4 Exemplo de Pseudocódigo.

---

```
leia (x, y) {Esta linha é um comentário}
se x > y então
    escreva ("x é maior")
senão
    se y > x então
        escreva ("y é maior")
    senão
        escreva ("x e y são iguais")
fim-se
fim-se
```

---



# Exercício

- Utilizando como base o Algoritmo 4, faça o que se pede:
  - 1- Construa um Algoritmo que resolve uma eq. do 2º grau;
  - 2- Construa um Algoritmo que calcula a média do aluno do IF e que informa se, baseado nesta regra, o mesmo está aprovado ou reprovado.
- Escreva o comando para receber os dados necessários;
- Utilize estruturas de condição e de repetição, se necessárias;
- Escreva o comando para imprimir o resultado.

---

## Algoritmo 4 Exemplo de Pseudocódigo.

---

```
leia (x, y)
se  $x > y$  então
    escreva ("x é maior")
senão
    se  $y > x$  então
        escreva ("y é maior")
    senão
        escreva ("x e y são iguais")
fim-se
fim-se
```

---