

# Introdução aos algoritmos

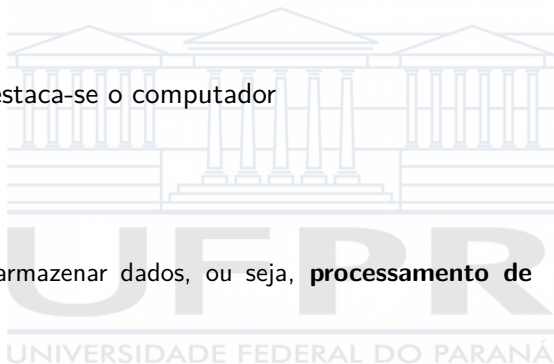
Prof. Robertino Mendes Santiago Jr

robertino@ufpr.br  
Licenciatura em Computação  
Universidade Federal do Paraná

2019/1

# Introdução

- Criação de máquinas para auxiliar o trabalho humano
  - ▶ Diminuição do esforço;
  - ▶ Economia de tempo.
- Dentre essas máquinas, destaca-se o computador
  - ▶ Rápido;
  - ▶ Seguro;
  - ▶ Versátil.
- Finalidade do computador
  - ▶ Receber, manipular e armazenar dados, ou seja, **processamento de dados**.



# Introdução

- O computador possui duas partes distintas que trabalham juntas
  - ▶ **Hardware:** composto pelas partes físicas;
  - ▶ **Software:** composto pelos programas,
- O que é necessário para desenvolver um programa?
  - ▶ **Linguagem de programação:** linguagem que pode ser compreendida tanto pelo computador quanto pelo programador.
- Etapas de desenvolvimento de um programa
  - ▶ **Análise:** estuda-se o enunciado do problema para definir os dados de entrada, processamento e os dados de saída;
  - ▶ **Algoritmo:** descrição do problema com suas soluções;
  - ▶ **Codificação:** transformação do algoritmo em códigos da linguagem de programação.

# Conceito de algoritmo

- **Sequência de passos** que visa atingir um objetivo bem definido.
- Descrição de uma **sequência de passos** que deve ser seguida para a realização de uma tarefa.
- **Sequência finita de instruções ou operações** cuja execução, em tempo finito, resolve um problema computacional, qualquer que seja sua instância.
- É a descrição de um **conjunto de comandos** que, obedecidos, resultam numa **sucessão finita de ações**.

# Algoritmo 1: somar três números

- 1 Receber os três números
- 2 Somar os três números
- 3 Mostrar o resultado obtido



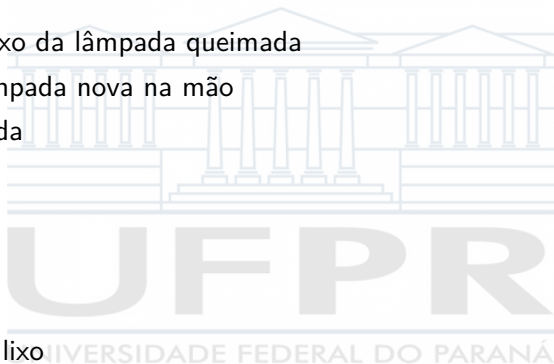
## Algoritmo 2: fazer um sanduíche

- 1 Pegar o pão
- 2 Cortar o pão ao meio
- 3 Pegar a maionese
- 4 Passar a maionese no pão
- 5 Pegar e cortar alface e tomate
- 6 Colocar alface e tomate no pão
- 7 Pegar o hambúrguer
- 8 Fritar o hambúrguer
- 9 Colocar o hambúrguer no pão



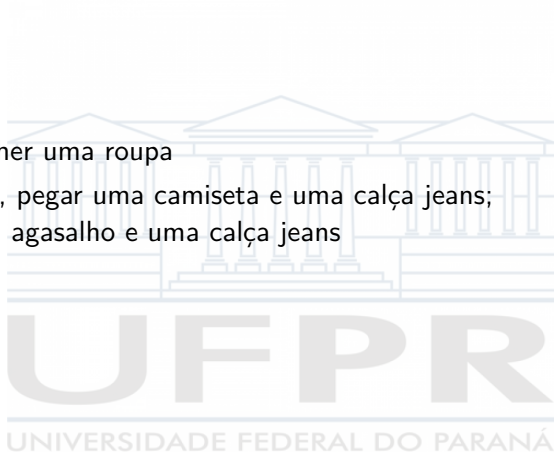
## Algoritmo 3: trocar uma lâmpada

- 1 Pegar uma lâmpada nova
- 2 Pegar uma escada
- 3 Posicionar a escada embaixo da lâmpada queimada
- 4 Subir na escada com a lâmpada nova na mão
- 5 Retirar a lâmpada queimada
- 6 Colocar a lâmpada nova
- 7 Descer da escada
- 8 Testar o interruptor
- 9 Guardar a escada
- 10 Jogar a lâmpada velha no lixo



## Algoritmo 4: ir para a escola

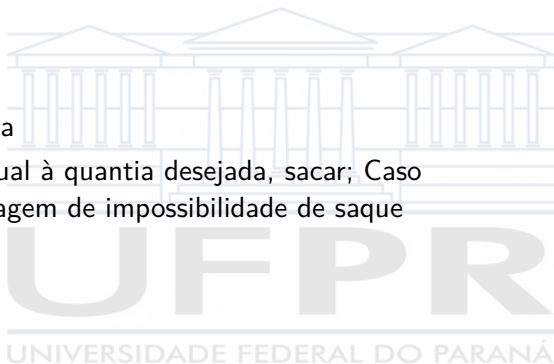
- 1 Acordar cedo
- 2 Ir ao banheiro
- 3 Abrir o armário para escolher uma roupa
- 4 Se o tempo estiver quente, pegar uma camiseta e uma calça jeans;  
Caso contrário, pegar uma agasalho e uma calça jeans
- 5 Vestir a roupa escolhida
- 6 Tomar café
- 7 Pegar uma condução
- 8 Descer próximo à escola





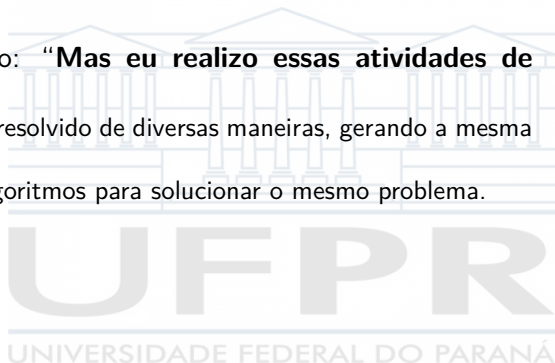
## Algoritmo 5: sacar dinheiro no banco 24 horas

- 1 Ir até um banco 24 horas
- 2 Colocar o cartão
- 3 Digitar a senha
- 4 Solicitar a quantia desejada
- 5 Se o saldo for maior ou igual à quantia desejada, sacar; Caso contrário, mostrar a mensagem de impossibilidade de saque
- 6 Retirar o cartão
- 7 Sair do banco 24 horas



# Observação

- Você pode estar pensando: **“Mas eu realizo essas atividades de maneira diferente!”**
  - ▶ Um problema pode ser resolvido de diversas maneiras, gerando a mesma resposta.
  - ▶ Podem existir vários algoritmos para solucionar o mesmo problema.



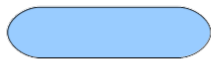
# Método para construção de algoritmos

- Para construir algoritmos é necessário os seguintes passos:
  - ▶ Compreender completamente o problema a ser resolvido, destacando os pontos mais importantes e os objetos que o compõem;
  - ▶ Definir os dados de entrada, ou seja, quais dados serão fornecidos e quais objetos fazem parte desse cenário problema;
  - ▶ Definir o processamento, ou seja, quais cálculos serão efetuados e quais as restrições para esses cálculos. O processamento é responsável pela transformação dos dados de entrada em dados de saída;
  - ▶ Definir os dados de saída, ou seja, quais dados serão gerados depois do processamento;
  - ▶ Construir o algoritmo;
  - ▶ Testar o algoritmo realizando simulações.

# Tipos de algoritmos

- Os três tipos mais utilizados são:
  - ▶ **Descrição narrativa:** consiste em analisar o enunciado do problema e escrever, utilizando uma linguagem natural, os passos a serem seguidos para sua resolução;
  - ▶ **Fluxograma:** consiste em analisar o enunciado do problema e escrever, utilizando símbolos gráficos predefinidos, os passos a serem seguidos para sua resolução;
  - ▶ **Pseudocódigo:** consiste em analisar o enunciado do problema e escrever, por meio de regras predefinidas, os passos a serem seguidos para sua resolução;

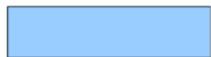
# Conjunto de símbolos utilizados no fluxograma



Símbolo utilizado para indicar o início e fim do algoritmo.



Símbolo que permite indicar o sentido do fluxo de dados. Conecta unicamente símbolos ou blocos.



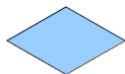
Símbolo utilizado para indicar cálculos e atribuições de valores.



Símbolo utilizado para representar entrada de dados.



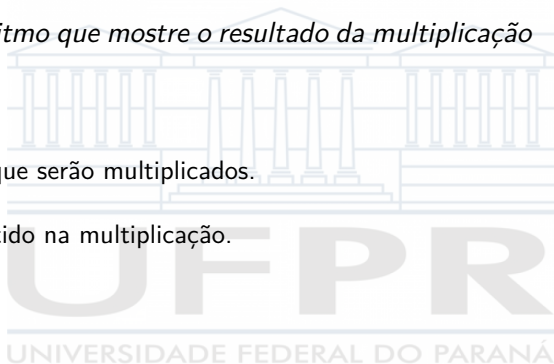
Símbolo utilizado para representar saída de dados.



Símbolo utilizado para indicar que deve ser tomada uma decisão, apontando a possibilidade de desvios.

# Exemplos de algoritmos

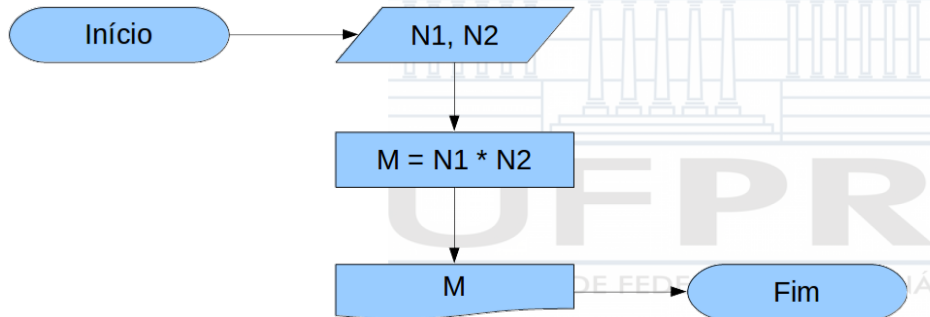
- *Exemplo 1: Faça um algoritmo que mostre o resultado da multiplicação de dois números:*
- **Descrição narrativa:**
  - 1 Receber dois números que serão multiplicados.
  - 2 Multiplicar os números.
  - 3 Mostrar o resultado obtido na multiplicação.



# Exemplos de algoritmos

- Exemplo 1: Faça um algoritmo que mostre o resultado da multiplicação de dois números:

- Fluxograma:**



# Exemplos de algoritmos

- *Exemplo 1: Faça um algoritmo que mostre o resultado da multiplicação de dois números:*

- **Pseudocódigo:**

```
1 algoritmo
2   declare n1, n2, m numerico
3   escreva "Digite dois números: "
4   leia n1, n2
5   m <- n1 * n2
6   escreva "Multiplicação = ", m
7 fim_algoritmo
```



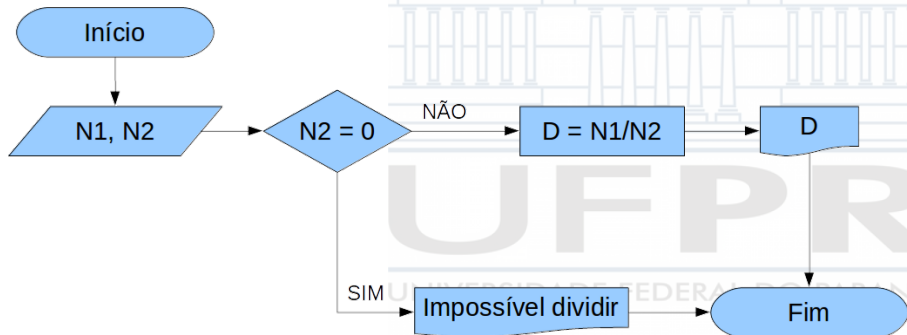
# Exemplos de algoritmos

- *Exemplo 2: Faça um algoritmo que mostre o resultado da divisão de dois números:*
- **Descrição narrativa:**
  - 1 Receber os dois números que serão divididos.
  - 2 Se o segundo número for igual a zero, não poderá ser feita a divisão, pois não existe divisão por zero; caso contrário, dividir os números e mostrar o resultado da divisão.

# Exemplos de algoritmos

- Exemplo 2: Faça um algoritmo que mostre o resultado da divisão de dois números:

- Fluxograma:**



# Exemplos de algoritmos

- Exemplo 2: Faça um algoritmo que mostre o resultado da divisão de dois números:

- Pseudocódigo:**

```
1 algoritmo
2   declare n1, n2, d numerico
3   escreva "Digite dois números: "
4   leia n1, n2
5   se n2 = 0
6   entao escreva "Impossível dividir"
7   senao inicio
8       d <- n1 / n2
9       escreva "Divisão = ", d
10  fim
11 fim_algoritmo
```

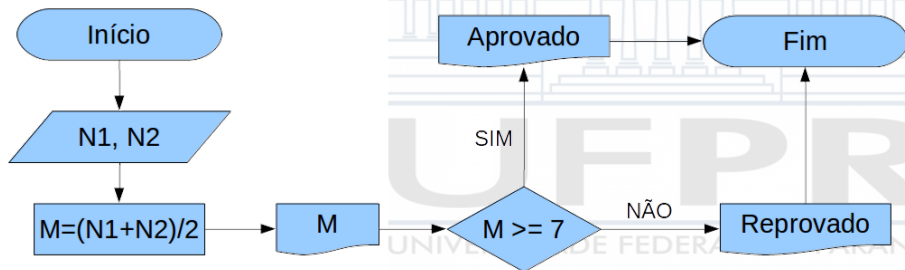
# Exemplos de algoritmos

- *Exemplo 3: Faça um algoritmo para calcular a média aritmética entre duas notas de um aluno e mostrar sua situação, que pode ser aprovado ou reprovado:*
- **Descrição narrativa:**
  - 1 Receber as duas notas.
  - 2 Calcular a média aritmética.
  - 3 Mostrar a média aritmética.
  - 4 Se a média aritmética for maior ou igual a 7, então a situação do aluno é aprovado; caso contrário, a situação é reprovado.

# Exemplos de algoritmos

- Exemplo 3: Faça um algoritmo para calcular a média aritmética entre duas notas de um aluno e mostrar sua situação, que pode ser aprovado ou reprovado:

- Fluxograma:**



# Exemplos de algoritmos

- *Exemplo 3: Faça um algoritmo para calcular a média aritmética entre duas notas de um aluno e mostrar sua situação, que pode ser aprovado ou reprovado:*
- **Pseudocódigo:**

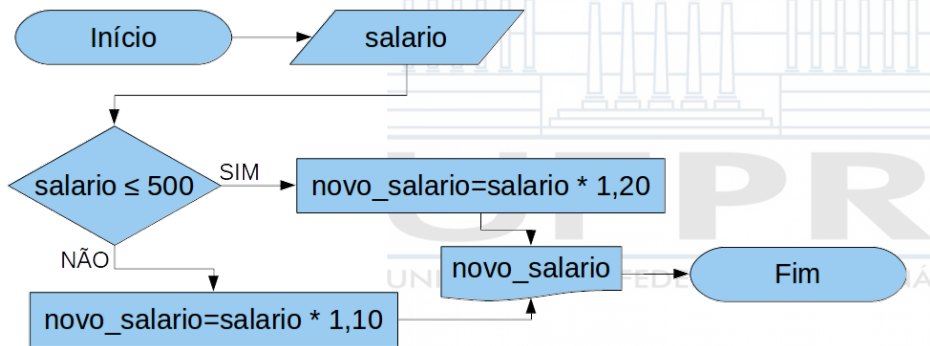
```
1 algoritmo
2   declare n1, n2, m numerico
3   escreva "Digite as duas notas: "
4   leia n1, n2
5   m <- (n1 + n2) / 2
6   escreva "Média = ", m
7   se m >= 7
8     entao escreva "Aprovado"
9     senao escreva "Reprovado"
10 fim_algoritmo
```

# Exemplos de algoritmos

- *Exemplo 4: Faça um algoritmo para calcular o novo salário de um funcionário. Sabe-se que os funcionários que recebem atualmente salário de até R\$ 500 terão aumento de 20%; os demais terão aumento de 10%:*
- **Descrição narrativa:**
  - 1 Receber o salário atual do funcionário.
  - 2 Se o salário atual do funcionário for de até R\$ 500, calcular o novo salário com percentual de aumento de 20%; caso contrário, calcular o novo salário com percentual de aumento de 10%.
  - 3 Mostrar o valor do novo salário.

# Exemplos de algoritmos

- Exemplo 4: Faça um algoritmo para calcular o novo salário de um funcionário. Sabe-se que os funcionários que recebem atualmente salário de até R\$ 500 terão aumento de 20%; os demais terão aumento de 10%:
- Fluxograma:**





# Exemplos de algoritmos

- Exemplo 4: *Faça um algoritmo para calcular o novo salário de um funcionário. Sabe-se que os funcionários que recebem atualmente salário de até R\$ 500 terão aumento de 20%; os demais terão aumento de 10%:*
- Pseudocódigo:**

```
1 algoritmo
2   declare salario, novo_salario numerico
3   escreva "Digite o salário atual do funcionário: "
4   leia salario
5   se salario <= 500
6   entao novo_salario <- salario * 1.20
7   senao novo_salario <- salario * 1.10
8   escreva "Novo salário = ", novo_salario
9 fim_algoritmo
```

# Variáveis

- Uma variável representa uma posição na memória.
  - ▶ Possui nome e tipo;
  - ▶ O conteúdo armazenado na variável pode variar durante a execução do programa;
  - ▶ Entretanto, a variável só pode armazenar um valor a cada instante.
- Tabela de alocação.
  - ▶ Nome da variável;
  - ▶ Tipo de dado (necessário para calcular o espaço a ser utilizado);
  - ▶ Endereço inicial de armazenamento.

# Tipos de dados

- Numéricos: inteiros ou reais;
  - ▶ Inteiros: 123, -2, 567, 0, -238
  - ▶ Reais: 3.4, -8.9, 4567.153, 1.1
- Lógico (booleanos): verdadeiro ou falso;
- Literais ou caracteres:
  - ▶ Único caractere: 'a', 'b', 'c', '3', '@'
  - ▶ Cadeia de caracteres: "aa", "ufpr", "Brasil", "323"

# Formação de identificadores

- Os identificadores são os nomes das variáveis, dos programas, das constantes, das rotinas, das unidades, etc. As regras básicas para a formação de identificadores são:
  - ▶ Os caracteres permitidos são: os números, as letras maiúsculas, as letras minúsculas e o caractere sublinhado;
  - ▶ O primeiro caractere deve ser sempre uma letra ou o caractere sublinhado;
  - ▶ Não são permitidos espaços em branco e caracteres especiais (@, \$, +, -, %, !);
  - ▶ Não podemos usar as palavras reservadas nos identificadores, ou seja, palavras que pertençam à linguagem de programação.

# Referências I



Ana Fernanda Gomes Ascencio and Edilene Aparecida Veneruchi de Campos.

*Fundamentos da Programação de Computadores: algoritmos, Pascal, C/C++ (Padrão ANSI) e Java.*

Pearson Education, São Paulo, 3 edition, 2012.

