



# Fundamentos de Programação 1

## Aula 4B

Lógica e Algoritmos.

Prof<sup>a</sup>. Fabiany  
fabiany@utfpr.edu.br

# Lógica

- A lógica de programação é a técnica de encadear pensamentos para atingir determinado objetivo.
- **Sequência Lógica**  
São passos executados até atingir um objetivo ou a solução de um problema.
- **Instruções**  
Um conjunto de regras ou normas definidas para a realização ou emprego de algo. Em informática, é o que indica a um computador uma ação elementar a executar.

# Algoritmo

- “Algoritmo é uma sequência de passos que visa atingir um objetivo bem definido.” (Forbellone,1999)
- “Algoritmo é a descrição de uma sequência de passos que deve ser seguida para a realização de uma tarefa.” (Ascencio,1999)
- No dia-a-dia executamos vários algoritmos.

# Exemplo Algoritmo

- **Fazer um sanduíche:**
  - Pegar o pão
  - Cortar o pão ao meio
  - Pegar a maionese
  - Abrir a maionese
  - Passar a maionese no pão
  - Pegar o alface e tomate
  - Cortar o tomate
  - Lavar o alface
  - Colocar o tomate e o alface no pão
  - Pegar o hamburger
  - Fritar o hamburger
  - Colocar o hamburger no pão
  - Fechar o pão.

# Exercícios

- Crie uma sequência lógica para tomar banho.
- Faça um algoritmo para trocar uma lâmpada. Descreva com detalhes.



# Método para a construção de algoritmos

- **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 os dados serão fornecidos e quais objetos fazem parte deste 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 os dados serão gerados depois do processamento.
- **Construir o algoritmo utilizando alguns dos tipos de algoritmos existentes.**
- **Testar o algoritmo realizando simulações.**



# Regras para a construção do Algoritmo

- Usar somente um verbo por frase;
- Imaginar que você está desenvolvendo um algoritmo para pessoas que não trabalham com informática;
- Usar frases curtas e simples;
- Ser objetivo;
- Procurar usar palavras que não tenham sentido ambíguo.

# Fases do algoritmo

- Primeiramente precisamos dividir o problema apresentado em três fases fundamentais:



- **Entrada:** são os dados de entrada do algoritmo;
- **Processamento:** são os procedimentos utilizados para chegar ao resultado final;
- **Saída:** são os dados já processados.



# Exemplo 1

- **Problema:** Calcular a média final dos alunos de uma turma. Os alunos realizarão quatro provas: P1, P2, P3 e P4.

Onde:

$$\text{Média Final} = \frac{P1 + P2 + P3 + P4}{4}$$

Para montar o algoritmo, faremos três perguntas:

- a) **Quais são os dados de entrada?**
- b) **Qual será o processamento a ser utilizado?**
- c) **Quais serão os dados de saída?**

# Respostas

**a) Quais são os dados de entrada?**

R.: Os dados de entrada são P1, P2, P3 e P4.

**b) Qual será o processamento a ser utilizado?**

R.: O procedimento será somar todos os dados de entrada e dividi-los por 4.

**c) Quais serão os dados de saída?**

R.: O dado de saída será a média final.

# Algoritmo Exemplo 1

Receba a nota da prova1

Receba a nota da prova2

Receba a nota da prova3

Receba a nota da prova4

Some todas as notas e divida o resultado por 4

Mostre o resultado da divisão.

# Tipos de Algoritmos

- **Descrição narrativa:** consiste em analisar o enunciado do problema e escrever, utilizando uma linguagem natural (ex: língua portuguesa), os passos a serem seguidos para a sua resolução;
- **Fluxograma ou Diagrama de blocos:** consiste em analisar o enunciado do problema e escrever, utilizando símbolos gráficos predefinidos, os passos a serem seguidos para a sua resolução;
- **Pseudocódigo ou português:** consiste em analisar o enunciado do problema e escrever, por meio de regras predefinidas, os passos a serem seguidos para a sua resolução.

# Exemplos de Algoritmos

Fazer um algoritmo para multiplicar dois números.

- Descrição narrativa:

Receba o primeiro número.

Receba o segundo número.

Multiplique os dois números.

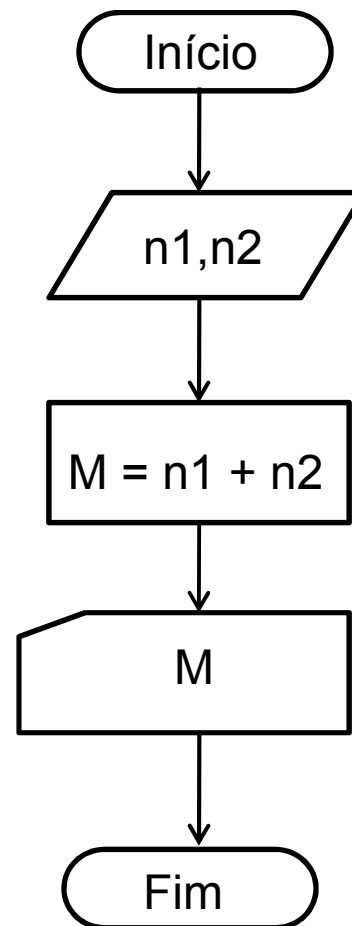
Mostre o resultado da multiplicação.


- Pseudocódigo:

```
Inicio {  
    inteiro n1, n2, m;  
    leia (n1, n2);  
    m = n1 * n2;  
    escreva("Resultado é: ", m);  
}
```

# Exemplos de Algoritmos

- Fluxograma ou diagrama de blocos:











# Diagrama de Bloco ou Fluxograma

- O diagrama de blocos é uma forma padronizada e eficaz para representar os passos lógicos de um determinado processamento.
- Podemos definir uma sequência de símbolos, com significado bem definido, portanto, sua principal função é a de facilitar a visualização dos passos de um algoritmo (processamento).

# Diagrama de Bloco ou Fluxograma

## Simbologia

Símbolo	Função
	Símbolo utilizado para indicar o início e o fim do algoritmo.
	Permite indicar o sentido do fluxo de dados. Serve exclusivamente para conectar os símbolos ou blocos existentes.
	Símbolo utilizado para indicar cálculos e atribuições de valores.
	Símbolo utilizado para representar a entrada de dados.
	Símbolo utilizado para representar a saída de dados.
	Símbolo utilizado pra indicar que deve ser tomada uma decisão, apontando a possibilidade de desvios.





# Variáveis e Constantes

- Variáveis e constantes são os elementos básicos que um programa manipula.
- Um programa deve conter declarações que especificam de que tipo são as variáveis que ele utilizará e as vezes um valor inicial.
- Tipos podem ser por exemplo: inteiros, reais, caracteres, etc.
- As expressões combinam variáveis e constantes para calcular novos valores.

# Constantes

- Constante é um determinado valor fixo que não se modifica ao longo do tempo, durante a execução de um programa.

# Variáveis

- Uma variável é um espaço reservado na memória do computador para armazenar um tipo de dado determinado, cujo conteúdo pode se alterado ao longo do tempo durante a execução de um programa.
- Variáveis devem receber nomes para poderem ser referenciadas e modificadas quando necessário.



# Variáveis

- Embora uma variável possa assumir diferentes valores, ela só pode armazenar um valor a cada instante.
- As variáveis só podem armazenar valores de um mesmo tipo.

# Operadores

- Os operadores são meios pelo qual realizamos cálculos, comparações e avaliações dos dados no nosso programa. Temos três tipos de operadores:
  1. Operadores Aritméticos
  2. Operadores Relacionais
  3. Operadores Lógicos

# Operadores Aritméticos

Operação	Símbolo
Adição	+
Subtração	-
Multiplicação	*
Divisão	/

# Operadores Relacionais

Descrição	Símbolo
Igual a	=
Diferente de	<> ou !=
Maior que	>
Menor que	<
Maior ou igual a	>=
Menor ou igual a	<=

# Operadores Lógicos

Operação	Operador
E	E
OU	OU
NÃO	NOT

E / AND: uma expressão AND (E) é verdadeira se todas as condições forem verdadeiras.

OR/OU: uma expressão OR (OU) é verdadeira se pelo menos uma condição for verdadeira.

NOT: um expressão NOT (NÃO) inverte o valor da expressão ou condição, se verdadeira inverte para falsa e vice-versa.



# Referência Bibliográfica

- **FOBERLLONE**, André Luiz Villar; **EBERSPÄCHER**, Henri Frederico. Lógica de programação a construção de algoritmos e estrutura de dados. 3. ed. Makron, 2000.
- **ASCENCIO**, Ana Fernanda Gomes; **CAMPOS**, Edilene Aparecida Veneruchi de. Fundamentos da programação de computadores. 2. ed. Pearson Prentice Hall, 2007.
- **MORAES**, Paulo Sérgio de. Curso Básico de Lógica de Programação. Unicamp, 2000.