



# Algoritmos 2023.2

## Aula 1 – Introdução aos Algoritmos

---

Profª Náthalee Cavalcanti de Almeida Lima

Email: [nathalee.almeida@ufersa.edu.br](mailto:nathalee.almeida@ufersa.edu.br)

04/12/2023



# Agenda

---

- Lógica
- Lógica de Programação
- Algoritmo
- Formas de Representação
  - Descrição Narrativa
  - Fluxograma
  - Pseudocódigo

# Lógica



## Observe as seguintes afirmações:

1. O número 4 é menor que o número 7. O número 9 é maior que o número 7. Logo, o número 4 é menor que os números 9 e 7.
2. Um casal com três filhos notou que um vaso estava quebrado, enquanto duas das crianças estavam na escola. Quem é o culpado?
3. Há uma caneta dentro de uma gaveta fechada. Sabemos que, para pegar a caneta devemos, antes, abrir a gaveta.

# LÓGICA



Segundo Xavier (2005) lógica é “uma sequência coerente, regular e necessária de acontecimentos, de coisas”.

# LÓGICA



Veja, a seguir, uma sequência de passos para comprar um determinado produto em uma loja.

1. Sair da loja;
2. Pagar pelo produto;
3. Entrar na loja;
4. Retirar o dinheiro da carteira;
5. Dirigir-se ao caixa;
6. Escolher o produto na prateleira.



**Seria possível comprar algum produto seguindo essa sequência???**

# LÓGICA



Apesar das ações estarem corretas, a sequência está errada.

1. Entrar na loja;
2. Escolher o produto na prateleira;
3. Dirigir-se ao caixa;
4. Retirar o dinheiro da carteira;
5. Pagar pelo produto;
6. Sair da loja.



# LÓGICA



A **lógica** é utilizada para conduzir pensamentos ou ações voltados para a **solução de problemas**.

Você sabe que a **lógica está correta** em determinada situação, se a **solução do problema tiver sido atingida**.

Na **programação de computadores**, o treino dessa habilidade é **FUNDAMENTAL** para aprimorarmos nossa capacidade de resolver problemas e programar.



# Lógica de Programação

- Técnica de *encadear* pensamentos para atingir determinado *objetivo*.
- Necessária para desenvolver programas e sistemas, pois permite definir a *seqüência lógica* para a solução de um problema.





# Lógica de Programação

SEQÜÊNCIA LÓGICA:      ? → 1. → 2. → 3. → !

- Estes pensamentos podem ser descritos como uma *seqüência de instruções*, que devem ser seguidas para se cumprir uma determinada tarefa.
- *Passos* executados até se atingir um objetivo ou solução de um problema.



# Lógica de Programação

- **Instrução:**

- Cada um dos *passos*, cada uma das ações a tomar (obedecendo a *seqüência lógica*) para ir resolvendo o problema, ou para ir executando a tarefa.
- Em informática, é a informação que indica a um computador uma *operação elementar* a executar
  - Ex.: “somar”, “subtrair”, “comparar se é maior”, etc.
- Uma só instrução não resolve problemas.



- Executar um *conjunto de instruções*
- Executar em uma *seqüência lógica*



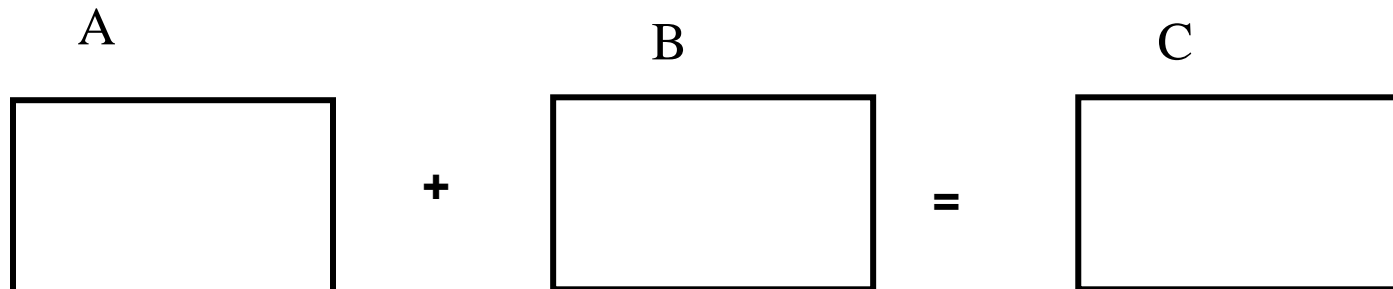
# Lógica de Programação

- **Exemplo: “Fazer omelete”**
  - Instruções: “quebrar ovos”, “bater ovos”, “colocar sal”, “ligar fogão”, “colocar óleo na frigideira”, “pôr frigideira no fogo”, “fritar ovos batidos”, etc...
- Quanto às instruções isoladas:
  - Só “quebrar ovos”, ou só “pôr óleo na frigideira”, não é suficiente para cumprir a tarefa “fazer omelete”
- Quanto à seqüência lógica:
  - Se executarmos “fritar ovos batidos” antes de “bater ovos”, ou pior, antes de “quebrar ovos”, não iremos cumprir a tarefa “fazer omelete”



# Algoritmo

- Sequência finita de passos que levam à execução de uma tarefa.
- Claro e preciso. Ex. “somar dois números”:
  - Escrever primeiro número no retângulo A
  - Escrever segundo número no retângulo B
  - Somar o número do retângulo A com o número do retângulo B e escrever o resultado no retângulo C



# Algoritmo



## Definições:

- “Um algoritmo é um processo sistemático para a resolução de um problema”. (Szwarcfiter, J. L. *et. al.*, 1994)
- “ Um algoritmo computa uma saída, o resultado do problema, a partir de uma entrada, as informações inicialmente conhecidas e que permitem encontrar a solução do problema”. (Szwarcfiter, J. L. *et. al.*, 1994)

# Algoritmo

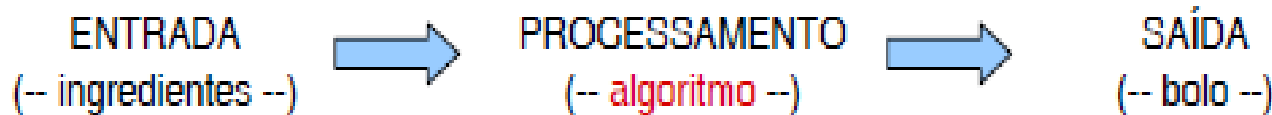


“Um algoritmo é um conjunto finito de regras que fornece uma sequência de operações para resolver um problema específico. É algo como uma receita, ou uma rotina”(IMD, UFRN 2014)



# Exemplo de Algoritmo

## ■ Vamos fazer um bolo!



1. Bater as claras em neve
2. Misturar açúcar e manteiga
3. Adicionar as gemas
4. Adicionar metade da farinha de trigo
5. Adicionar metade do leite
6. Adicionar a outra metade da farinha de trigo
7. Adicionar a outra metade do leite
8. Ligar o forno a 180°
9. Adicionar as claras
10. Despejar bolo na forma
11. Colocar a forma no forno





# Exemplo de Algoritmo

Um motorista que necessita efetuar a troca de um pneu furado segue uma rotina para realizar essa tarefa:

1. Verifica qual pneu está furado
2. Posiciona o macaco para levantar o carro
3. Pega o estepe
4. Solta os parafusos
5. Substitui o pneu furado
6. Recoloca os parafusos
7. Desce o carro
8. Guarda o macaco e o pneu furado

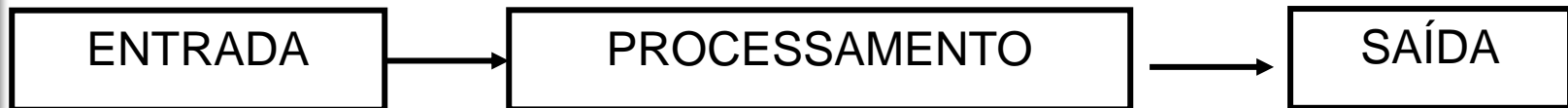


# Algoritmos



**FASES** para desenvolver o algoritmo:

- 1 - Determinar o problema, defini-lo bem.
- 2 - Dividir a solução nas três fases:



# Algoritmos



## Exemplo:

- **Problema:** calcular a média de quatro números
- **Dados de entrada:** os números,  $N1$ ,  $N2$ ,  $N3$  e  $N4$
- **Processamento:** somar os quatro números e dividir a soma por 4
- **Dados de saída:** a média final

# Algoritmos x Programas



- Não confunda **algoritmos** com **programas de computadores!!**
- O **algoritmo** é um conjunto de instruções para realizar uma tarefa.
- Um algoritmo escrito em uma linguagem de programação e traduzido para linguagem de máquina é chamado de **programa**.
- Chegando a uma solução algorítmica é possível traduzir para qualquer linguagem de programação.

# Algoritmos x Programas



- A **linguagem de programação** é a ponte entre o computador e o humano.
- **Linguagens de programação:**
  - Linguagens de alto nível (java, c++, python, html, php, ...)
  - Linguagens de baixo nível (linguagem binária, assembly, ...): interpretadas diretamente pelo computador que só entende zeros e uns.

# Formas de Representação de Algoritmos

---



1. DESCRIÇÃO NARRATIVA

2. FLUXOGRAMA

3. PSEUDOCÓDIGO

# DESCRIÇÃO NARRATIVA



- Analisamos o enunciado do problema;
- Descrevemos a sequência de passos em nossa língua nativa.

## Vantagens:

- Não precisamos aprender nenhum conceito novo, pois já temos domínio sobre a nossa língua nativa.

# DESCRIÇÃO NARRATIVA



## Desvantagens:

- A língua natural pode ser interpretada de diferentes maneiras.
- Para a linguagem de programação, a linguagem natural é abstrata, imprecisa e pouco confiável. Isso poderia trazer problemas na hora de transcrever o algoritmo para o programa (em uma linguagem de programação).

# DESCRIÇÃO NARRATIVA - EXEMPLO



Ingredientes:

Modo de preparo:

**Passo 2:** Aqueça o forno a 180 graus;

**Passo 3:** Bata as claras em neve e reserve;

**Passo 4:** Em uma travessa, bata o açúcar, a manteiga e as gemas;

**Passo 5:** Misture a farinha e o leite;

**Passo 6:** Bata bem, até ficar bem homogêneo;

**Passo 7:** Com a ajuda de uma colher, acrescente o fermento;

**Passo 8:** Por último, adicione as claras em neve e mexa cuidadosamente;

**Passo 9:** Coloque em uma forma untada com manteiga e farinha de trigo e leve ao forno médio para assar por aproximadamente 35 minutos ou até que, ao espetar um palito, esse saia seco;

**Passo 10:** Após assado, desligue o forno e deixe o bolo esfriar;

**Passo 11:** Desenforme e saboreie.

FIM



# DESCRIÇÃO NARRATIVA - EXEMPLO



Passo 9: Coloque em uma forma untada com manteiga e farinha de trigo e leve ao forno médio para assar por **aproximadamente** 35 minutos ou até que, ao espetar um palito, esse saia seco;

- Informação em negrito **imprecisa** e pode ser interpretada de diferentes formas por diferentes pessoas.
- Essa **imprecisão** dificultaria, portanto, a transcrição para uma linguagem de programação.







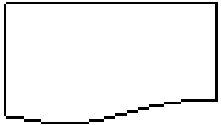
# FLUXOGRAMA



É uma forma universal de representação, pois se utiliza de figuras geométricas para ilustrar passos a serem seguidos para a resolução de problemas.



# FLUXOGRAMA

	Início ou fim do algoritmo
	Indica o sentido do fluxo de execução do algoritmo. Conecta os objetos gráficos
	Representa a entrada de dados
	Indica cálculos e atribuições de valores (processamento)
	Indica desvios ou tomadas de decisões (Por exemplo: SE isso, ENTÃO aquilo)
	Representa a saída de dados, no Portugol IDE
	Também representa a saída de dados

**Fonte:** Material Didático, IMD, UFRN, 2012.

# FLUXOGRAMA



## Vantagens:

- mais fácil entender um conteúdo descrito de forma gráfica do que um descrito textualmente;
- os fluxogramas obedecem a um **padrão** mundial, quanto à simbologia.

# FLUXOGRAMA



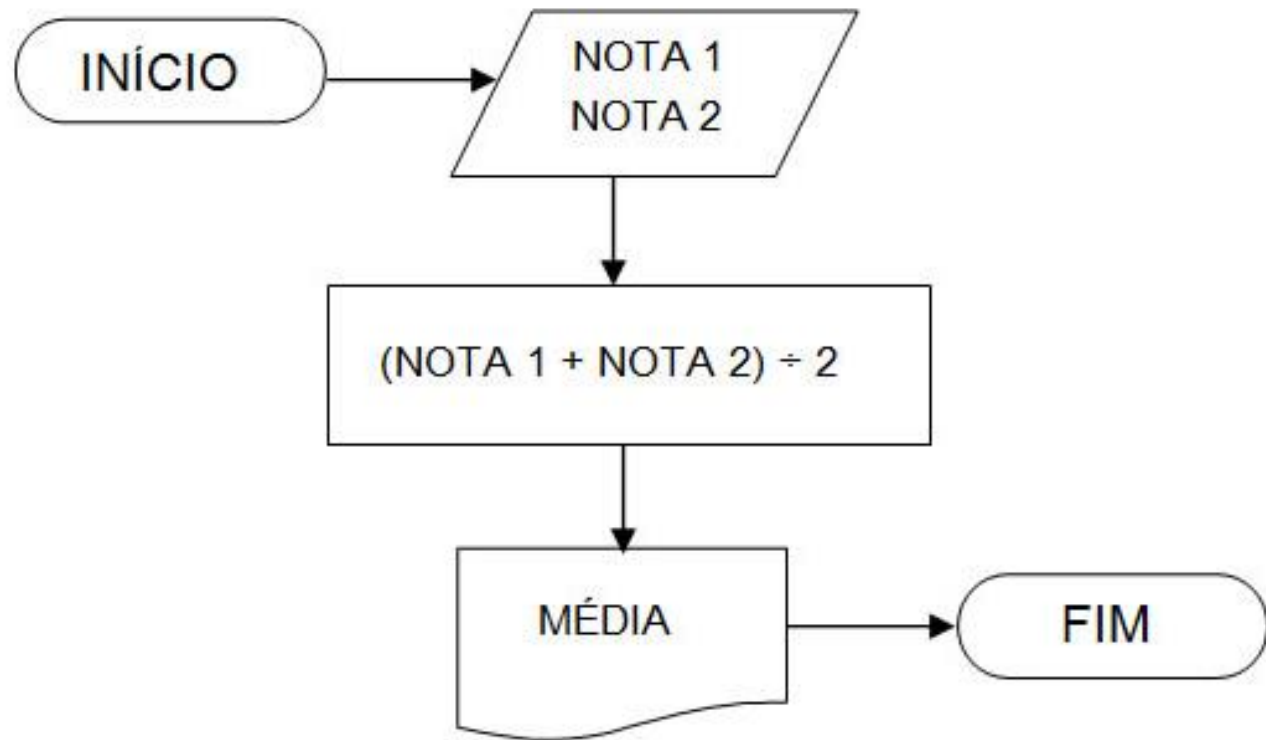
## Desvantagens:

- os dados podem não ser suficientemente detalhados, dificultando, assim, a transcrição do algoritmo para o programa a ser desenvolvido;
- é necessário aprender a simbologia dos fluxogramas;
- para algoritmos mais extensos, a construção do fluxograma pode se tornar mais complicada.



# FLUXOGRAMA - EXEMPLO

**Algoritmo para calcular a média aritmética de um aluno**



# PSEUDOCÓDIGO



- A **descrição narrativa** pode ser interpretada de diversas maneiras. Isso pode gerar ambiguidades.
- Já o **fluxograma** tem maior precisão, mas é pouco descritivo, o que pode torná-lo insuficiente, além de complicar-se, conforme o crescimento do algoritmo.
- O **pseudocódigo** é uma combinação das melhores características das duas formas de representação anteriores.

# PSEUDOCÓDIGO



- Português estruturado, linguagem estruturada ou pseudolinguagem.
- O pseudocódigo obedece a regras predefinidas de estrutura para descrever um algoritmo.
- Facilita descrever o algoritmo antes de passá-lo para uma linguagem de programação;
- **Intermediária**: linguagem natural – linguagem de programação;
- Pseudocódigo = “código falso”



# PSEUDOCÓDIGO



## Vantagens:

- **Descrição do algoritmo** menos rigorosa que na linguagem de programação (código fonte);
- Fácil de entender e fácil de *codificar* depois;
- Independente da linguagem de programação;
- Simples e objetivo → Técnicas:
- Pode definir quais os dados a ser utilizados e como eles vão estar estruturados, além de utilizar o português como base.

# PSEUDOCÓDIGO



## Desvantagens:

- precisamos aprender as regras dessa forma de representação;
- não padronização de sua estruturação. Isso quer dizer que você encontrará um mesmo termo descrito de formas diferentes em diferentes literaturas.

# PSEUDOCÓDIGO



**algoritmo** “Média”

**Var**

nota1: **real**

nota2: **real**

media: **real**

**Início**

**escreva** (“digite o valor da primeiro nota:”)

**leia**(nota1)

**escreva** (“digite o valor da primeiro nota:”)

**leia**(nota2)

media<- (nota1+nota2)/2

**escreva** (“A média d aluno é:”, media)

**fimalgoritmo**



# Programa

- *Algoritmo* escrito em uma **linguagem** de computador (linguagem de programação - C, Pascal, COBOL, Fortran, Basic, Java, etc.)
- Interpretado e executado por um computador
- Interpretação rigorosa, exata do computador  $\Rightarrow$   
 $\Rightarrow$  escrita do algoritmo na linguagem de prog. tem que seguir regras mais rigorosas

# Exercícios



- A. Escreva a seqüência de passos para que uma pessoa consiga fazer um download de um arquivo.
- B. Escreva os passos necessários para uma pessoa colocar combustível na sua moto.
- C. Escreva os passos necessários para fazer uma feira no supermercado.
- D. Escreva os passos para que, dado um número, seja exibido seu antecessor.

# Exercícios



1) Identifique os dados de entrada, processamento e saída no algoritmo abaixo

- Calcule o valor total da peça (Quantidade \* Valor da peça)
- Receba código da peça
- Receba Quantidade de peças
- Mostre o código da peça e seu valor total
- Receba valor da peça

2) Faça um algoritmo para “Calcular o estoque médio de uma peça”, sendo que

$$\text{ESTOQUE MÉDIO} = (\text{QUANTIDADE MÍNIMA} + \text{QUANTIDADE MÁXIMA}) / 2$$

# Referências



- *Salvetti, D. D.; Barbosa, L. M. Algoritmos. São Paulo: Makron Books, 2004. 300p;*
- *Forbellone, A. L. V.; Eberspächer, H. F. **Lógica de Programação – A construção de algoritmos e estruturas de dados.** 3.ed. São Paulo: Makron Books, 2005.*
- *Manzano, J. A; Oliveira, J. F. **Algoritmos - Estudo dirigido.** 15ª edição revisada. Editora Érica, 2012.*
- Instituto Metrópole Digital (IMD), Material Didático, UFRN, 2012.