

INTRODUÇÃO A CIÊNCIA DA COMPUTAÇÃO

INTRODUÇÃO A ALGORITMOS

Maurício Moreira Neto¹

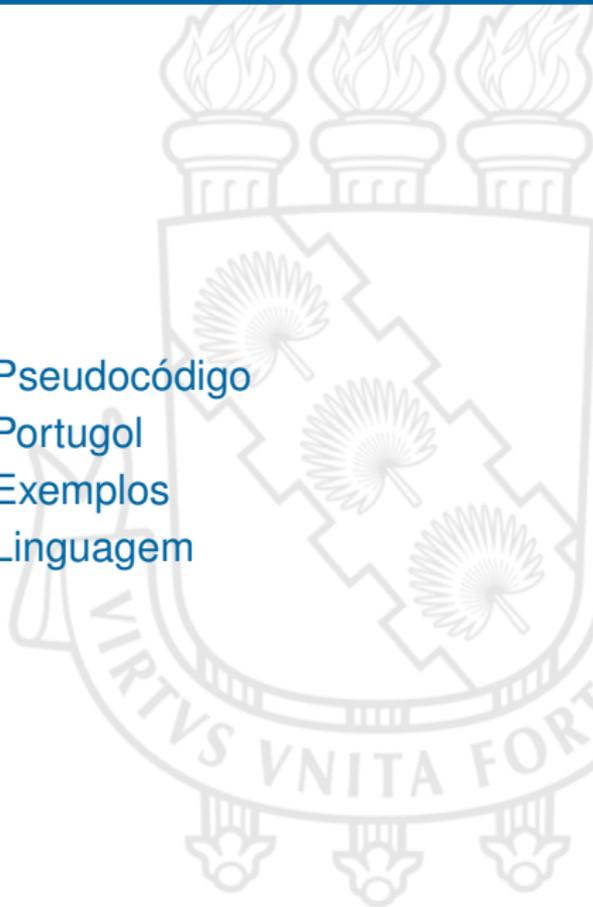
¹**Universidade Federal do Ceará**
Departamento de Computação

31 de janeiro de 2020

Sumário

- 1 Algoritmo
- 2 Tipos de Algoritmos
- 3 Narrativa
- 4 Fluxograma

- 5 Pseudocódigo
- 6 Portugol
- 7 Exemplos
- 8 Linguagem



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)

Algoritmo

- Solucionar algum problema
- Realizar determinada ação
- O número de passos deve ser finito

- Importância na programação
 - Representar o raciocínio lógico
 - Focaliza-se primeiro na resolução algorítmica do problema, possibilitando sua codificação em qualquer linguagem de programação

Método para a Construção de Algoritmos

- **Compreender** o problema
- Definir os dados de **entrada**
- Definir o **processamento**
- Definir os dados de **saída**
- **Construir** o algoritmo
- **Testar** o algoritmo



Algoritmo

- Os algoritmos são comuns no cotidiano
 - Estamos sempre realizando uma sequência de passos para realizar alguma ação

- **Exemplo:** Receita de bolo
 - 1 Bata as claras em neve e reserve
 - 2 Misture as gemas, a margarina e o açúcar até obter uma massa homogênea
 - 3 Acrescente o leite e a farinha de trigo aos poucos, sem parar de bater
 - 4 Por último, adicione as claras em neve e o fermento
 - 5 ...

Algoritmo

Algoritmo: Dobro de um número

1. Digitar um número
2. Guardar o número
3. Pegar o número e multiplicar por 2
4. Guardar o resultado do cálculo
5. Mostrar o resultado do cálculo

Algoritmo: Soma de dois números

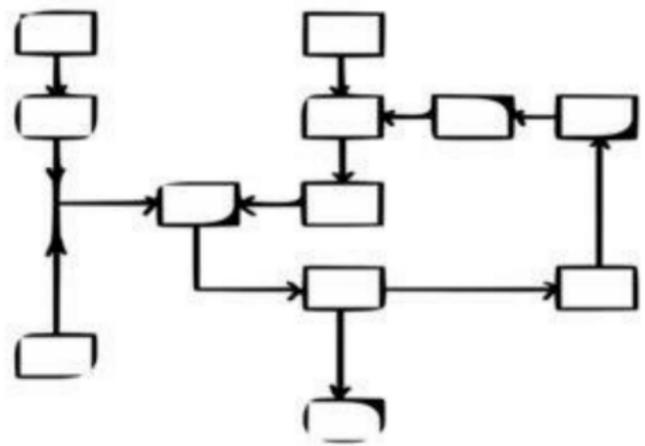
1. Digitar dois números
2. Gravar os números
3. Somar os dois números
4. Gravar o resultado da soma
5. Mostrar o resultado da soma

Algoritmo: Troca de marcha

1. Colocar a marcha no ponto-morto
2. Pressionar a embreagem
3. Deslocar a marcha para o local da próxima marcha
4. Liberar o pé da embreagem

Algoritmo

- Escreva um algoritmo do seu cotidiano



Tipos de Algoritmos

- Descrição narrativa
- Fluxograma
- Pseudocódigo ou portugol



Descrição Narrativa

- Consiste em analisar o enunciado do **problema** e escrever, utilizando uma linguagem natural, os passos a serem seguidos para sua resolução

Descrição Narrativa

■ Descrição Narrativa

- Uma das formas mais simples
- Linguagem natural e no idioma nativo
- Consiste em apenas escrever, de forma textual, o que precisa ser feito
- É pouca utilizada por conta da não eficiência
 - Pode gerar más interpretações, ambiguidade, imprecisão

Descrição Narrativa

■ Exemplo 1

Passo 1 – Receber duas notas

Passo 2 – Calcular a média aritmética

Passo 3 – Mostrar a média aritmética

Passo 4 – Se a média aritmética for maior ou igual a 7, então a situação do aluno é aprovado; caso contrário, a situação é reprovado

Descrição Narrativa

Algoritmo: Receita de bolo

1. Misture os ingredientes
2. Unte a forma com manteiga
3. Despeje a mistura na forma
4. Leve a fôrma ao forno
5. Enquanto não corar deixe a forma no forno
6. Retire do forno
7. Deixe esfriar

Algoritmo: Troca de pneu

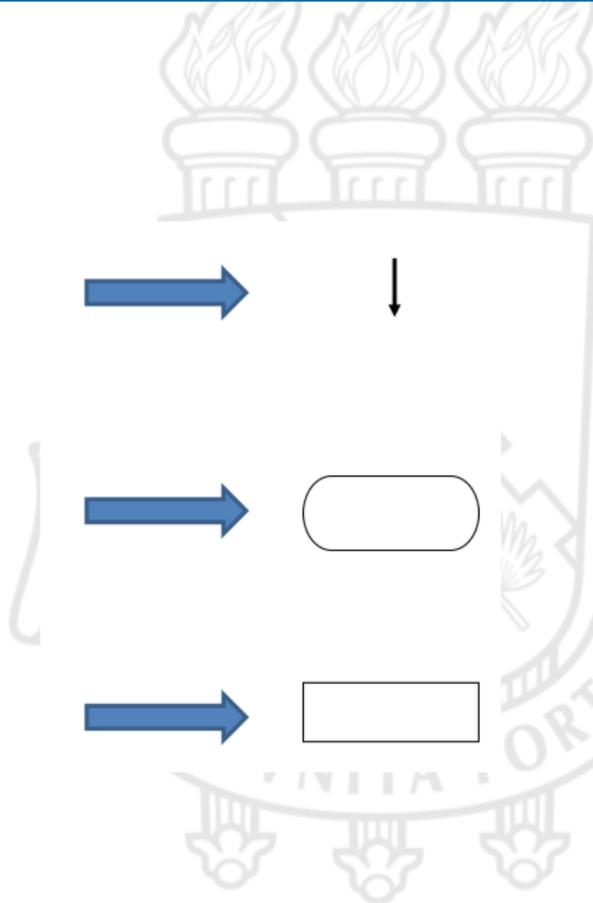
1. Afrouxar ligeiramente as porcas
2. Suspender o carro
3. Retirar as porcas
4. Retirar o pneu
5. Colocar o pneu reserva
6. Apertar as porcas
7. Abaixar o carro
8. Dar o aperto final nas porcas

Fluxograma

- Consiste em analisar o enunciado do **problema** e escrever, utilizar símbolos gráficos predefinidos, os passos a serem seguidos para sua resolução
- Tal propriedade facilita o entendimento das ideias contidas nos algoritmos e justifica sua popularidade

Fluxograma

- Simbologia:
 - Fluxo de dados
 - Indica o sentido do fluxo de dados
 - Conecta os demais símbolos
 - Terminal
 - Indica o INÍCIO ou FIM de um processamento
 - **Exemplo:** Início do algoritmo
 - Processamento
 - Processamento em geral
 - **Exemplo:** Cálculo de dois números



Fluxograma

■ Simbologia:

■ Entrada

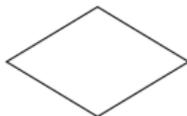
- Operação de entrada de dados
- **Exemplo:** Leitura de dados pelo teclado

■ Saída

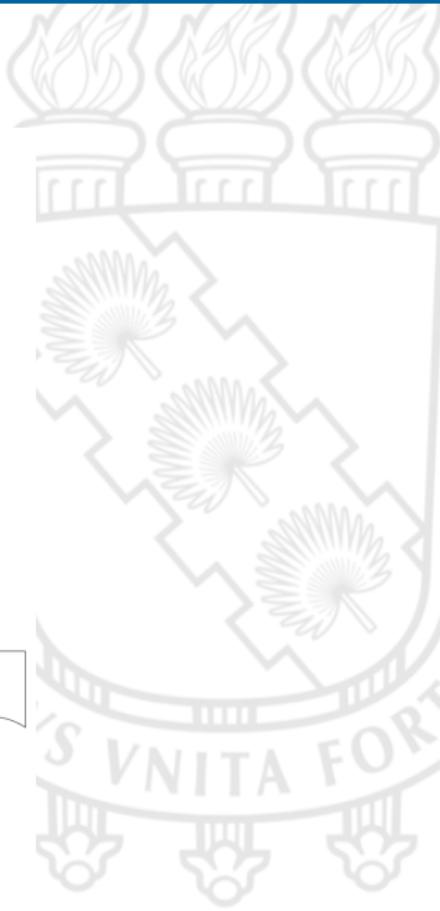
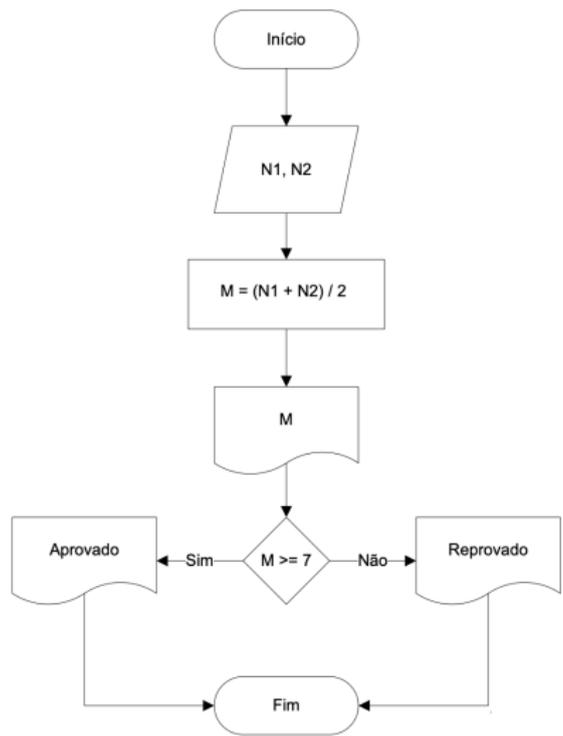
- Mostra informações ou resultados
- **Exemplo:** Mostre o resultado do cálculo

■ Decisão

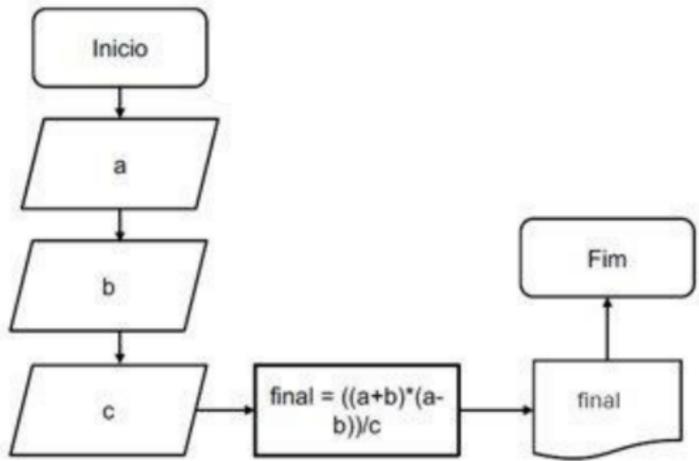
- Permite elaborar processos de decisão



Fluxograma



Fluxograma



Pseudocódigo

■ Pseudocódigo

- Esta forma de **representação** de algoritmos é rica em **detalhes**. Por assemelhar-se bastante à forma em que os programas são escritos, encontra muita **aceitação**
- É linguagem **intermediária** entre a linguagem natural e uma linguagem de programação
- É mais simples a implementação
- As regras do pseudocódigo devem ser **aprendidas!**

Pseudocódigo

```
Algoritmo <nome_do_algoritmo>;  
<declaração_de_variaveis>;  
<subalgoritmos>  
início  
<corpo_do_algoritmo>  
fim
```

<nome_do_algoritmo> é um identificador dado ao algoritmo com a finalidade de distingui-los dos demais.

<declaração_de_variáveis> consiste em uma porção opcional onde são declaradas as variáveis globais usadas no algoritmo principal e, eventualmente, nos subalgoritmos.

<subalgoritmos> consiste de uma porção opcional do pseudocódigo onde são definidos os subalgoritmos.

Início e Fim são respectivamente as palavras que delimitam o início e o término do conjunto de instruções do corpo do algoritmo.

Pseudocódigo

```
Algoritmo Soma;  
var:  
    x,y,final : inteiro;  
início:  
    escreva("digite um valor para x: ")  
    leia(x)  
    escreva("digite um valor para y: ")  
    leia(y)  
    final <- x+y  
    escreva("o resultado da soma de x e y  
é: ",final)  
fim
```

```
Algoritmo Delta;  
var:  
    a,b,c,delta : real;  
início:  
    escreva("digite o valor de a: ")  
    leia(a)  
    escreva("digite o valor de b: ")  
    leia(b)  
    escreva("digite o valor de c: ")  
    leia(c)  
    delta <- ((b * b) - (4 * a * c))  
    escreva("o resultado de delta é: ",  
delta)  
fim
```

Pseudocódigo

Escreva um algoritmo pseudocódigo



Vantagens e Desvantagens

- Vantagens e Desvantagens de cada tipo de representação

	Vantagens	Desvantagens
Narrativa	Fácil Escrever Linguagem Natural	Más Interpretações Ambiguidade
Pseudocódigo	Fácil transição para a linguagem de programação	Necessário aprender regras do pseudocódigo
Fluxograma	Fácil de Visualizar Fluxo Lógico	Aprender os Símbolos

Portugal

- Consiste em analisar o enunciado do **problema** e escrever, por meio de regras predefinidas, os passos a serem seguidos para sua resolução

Portugol

■ Exemplo 1

ALGORITMO

DECLARE N1, N2, M **NUMÉRICO**

ESCREVA "Digite as duas notas"

LEIA N1, N2

M ← (N1 + N2) / 2

ESCREVA "Média = ", M

SE M ≥ 7

ENTÃO ESCREVA "Aprovado"

SENÃO ESCREVA "Reprovado"

FIM
ALGORITMO

Etapas no Desenvolvimento de um Programa

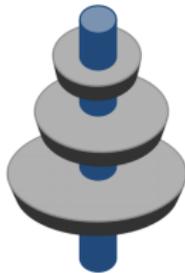
- **Análise:** estuda-se o **problema** para definir os dados de **entrada**, o **processamento** e os dados de **saída**
- **Algoritmo:** Utiliza-se para descrever o **problema** com suas soluções
- **Codificação:** Transforma o **algoritmo** em códigos de uma **linguagem de programação**

Exemplo de Problemas

■ Problema da Torre de Hanói

■ Seja a seguinte situação:

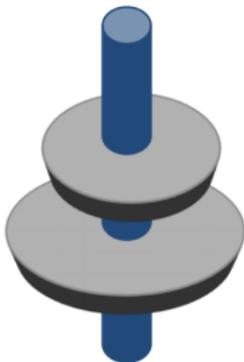
- Deve-se mover todos os discos do primeiro eixo para o terceiro mantendo-se a ordem original
- Em cada movimento, pode-se mover apenas um disco
- Um disco nunca poderá ser sobreposto por outro maior



Exemplo de Problemas

■ Passo 1:

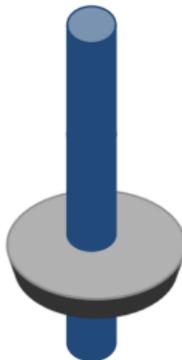
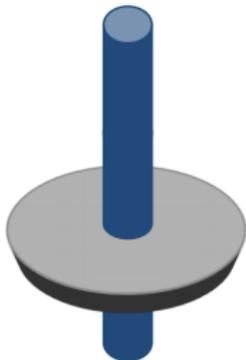
- mova disco menor para terceiro eixo



Exemplo de Problemas

■ Passo 2:

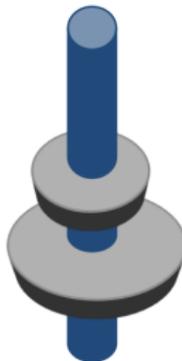
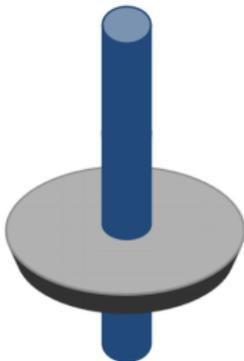
- **mova disco médio para segundo eixo**



Exemplo de Problemas

■ Passo 3:

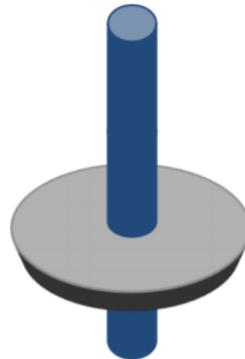
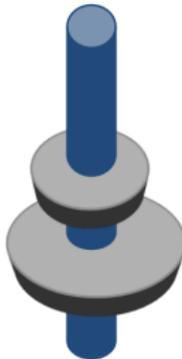
- **move disco menor para segundo eixo**



Exemplo de Problemas

■ Passo 4:

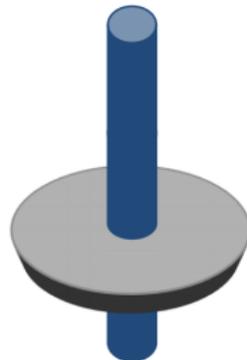
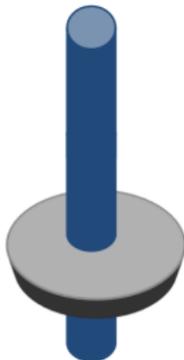
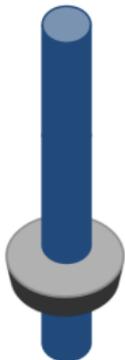
- **mova disco maior para terceiro eixo**



Exemplo de Problemas

■ Passo 5:

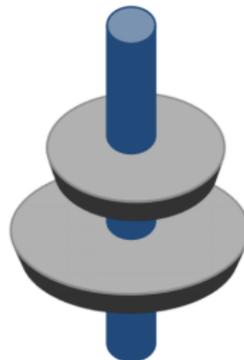
- **mova disco menor para primeiro eixo**



Exemplo de Problemas

■ Passo 6:

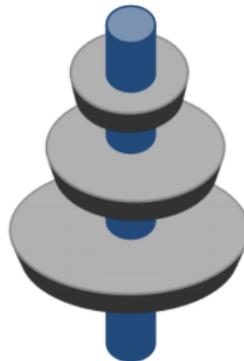
- `mova disco médio para terceiro eixo`



Exemplo de Problemas

■ Passo 7:

- **move disco menor para terceiro eixo**



Exemplo de Problemas

■ Sequência de Passos Completa:

- **Passo 1:** mova disco menor para terceiro eixo
- **Passo 2:** mova disco médio para segundo eixo
- **Passo 3:** mova disco menor para segundo eixo
- **Passo 4:** mova disco maior para terceiro eixo
- **Passo 5:** mova disco menor para primeiro eixo
- **Passo 6:** mova disco médio para terceiro eixo
- **Passo 7:** mova disco menor para terceiro eixo

Jogo da Travessia do Rio

■ Como jogar:

■ As regras são as seguintes:

- 1 Somente o **pai**, a **mãe** e o **policial** sabem pilotar o barco
- 2 A **mãe** não pode ficar sozinha com os **filhos**
- 3 O **pai** não pode ficar sozinho com as **filhas**
- 4 O **prisioneiro** não pode ficar sozinho com nenhum **integrante da família**
- 5 O **barco** só pode transportar 2 pessoas por vez
- 6 Você pode ir e vir com as pessoas quantas vezes precisar

Jogo da Travessia do Rio

Jogo da Travessia do Rio

Vamos ao desafio!!



Jogo da Travessia do Rio

- 1 ida = policial + bandida
- 2 volta = policial
- 3 ida = policial + menino
- 4 volta = policial + bandida
- 5 ida = pai + menino
- 6 volta = pai
- 7 ida = pai + mãe
- 8 volta = mãe
- 9 ida = policial + bandida

- 10 volta = pai
- 11 ida = pai + mãe
- 12 volta = mãe
- 13 ida = mãe + menina
- 14 volta = policial + bandida
- 15 ida = policial + menina
- 16 volta = policial
- 17 ida = policial + bandida

Linguagens de Programação

- Utilizada para **escrever programas** onde o **computador** e o **criador** deste programa compreendam
- Linguagens **Compiladas** e Linguagens **Interpretadas**
- Paradigmas de Programação (dois mais conhecidos)
 - Programação Estruturada
 - **Exemplos:** Pascal e C
 - Programação Orientada a Objetos
 - **Exemplos:** C++ e Java

Compiladores

- Um compilador tem a finalidade:
 - Converter uma linguagem – **Linguagem Fonte** – de fácil escrita e leitura para os programadores, numa linguagem – **Linguagem alvo ou objeto** – que possa ser executada pelas máquinas
- O código executável gerado pelo compilador é **dependente** do **Sistema Operacional** e da linguagem de máquina para o qual o código fonte foi traduzido
- Exemplos de Linguagens Compiladas:
 - Pascal
 - C, C++

Interpretadores

- O funcionamento dos **interpretadores** é muito parecido ao dos **compiladores**
- O interpretador traduz o código linha a linha
- O código fonte não é totalmente traduzido antes de ser executado
- Não existem fases distintas nem se produz **código intermediário**
- Passa o tempo todo a ler e a traduzir código
- Exemplo de Linguagens Interpretadas:
 - HTML
 - PHP
 - ASP

Compiladores x Interpretadores

	Vantagens	Desvantagens
Compiladores	Execução mais rápida	Várias etapas de tradução
	Permite a otimização do código fonte	Processo de correção de erros e depuração é mais demorado
Interpretadores	Depuração do programa é mais simples	Execução do programa é mais lenta
	Resultado imediato do programa ou rotina desenvolvida	Necessário fornecer o programa fonte ao utilizador

Obrigado!

maumneto@gmail.com

