

## **Aula 2**

**Prof. Daniel Cavalcanti Jeronymo**

# **Fundamentos de Programação**

CP41F

Conceito de algoritmo. Raciocínio lógico na construção de algoritmos. Estrutura de algoritmos.

**Universidade Tecnológica Federal do Paraná (UTFPR)**  
Engenharia de Computação – 1º Período

- Desafio
- Conceito de algoritmo
- Características de algoritmos
- Representação de algoritmos
- Raciocínio lógico na construção de algoritmos
- Estrutura de algoritmos
- Tipos de dados
- Comandos

# Desafio

- Elabore uma sequência de passos para obter 4L de água a partir de um galão de 5L e outro galão de 3L



# Desafio

Como desenvolver uma solução que seja reproduzível para todos os casos desta situação?

Desejável: solução na forma de receita de bolo, basta seguir passos bem definidos.

Elementos do problema:

- **Entradas** - dois galões
- **Operações** - encher, esvaziar, transferir
- **Saída** desejada - 4 litros

# Desafio

Solução possível:

1. Iniciar com os galões vazios
2. Encher o galão de 5 litros
3. Transferir conteúdo do galão de 5 litros para o de 3 litros
4. Esvaziar o galão de 3 litros
5. Transferir conteúdo do galão de 5 litros para o de 3 litros
6. Encher o galão de 5 litros
7. Transferir conteúdo do galão de 5 litros para o de 3 litros

# Desafio

- Como verificar o algoritmo?
- Utilizar o teste de mesa (*table test*):

1. Iniciar com os galões vazios	g5 = ?	g3 = ?
2. Encher o galão de 5 litros	g5 = ?	g3 = ?
3. Transferir conteúdo do galão de 5 litros para o de 3 litros	g5 = ?	g3 = ?
4. Descartar a água do galão de 3 litros	g5 = ?	g3 = ?
5. Transferir conteúdo do galão de 5 litros para o de 3 litros	g5 = ?	g3 = ?
6. Encher o galão de 5 litros	g5 = ?	g3 = ?
7. Transferir conteúdo do galão de 5 litros para o de 3 litros	g5 = ?	g3 = ?

# Desafio

- Cada linha é um passo
- Cada coluna uma variável
- **Execute o teste de mesa!**

1. Iniciar com os galões vazios	$g5 = ?$	$g3 = ?$
2. Encher o galão de 5 litros	$g5 = ?$	$g3 = ?$
3. Transferir conteúdo do galão de 5 litros para o de 3 litros	$g5 = ?$	$g3 = ?$
4. Descartar a água do galão de 3 litros	$g5 = ?$	$g3 = ?$
5. Transferir conteúdo do galão de 5 litros para o de 3 litros	$g5 = ?$	$g3 = ?$
6. Encher o galão de 5 litros	$g5 = ?$	$g3 = ?$
7. Transferir conteúdo do galão de 5 litros para o de 3 litros	$g5 = ?$	$g3 = ?$

# Desafio

- Resultado final

1. Iniciar com os galões vazios	$g5 = 0$	$g3 = 0$
2. Encher o galão de 5 litros	$g5 = 5$	$g3 = 0$
3. Transferir conteúdo do galão de 5 litros para o de 3 litros	$g5 = 2$	$g3 = 3$
4. Descartar a água do galão de 3 litros	$g5 = 2$	$g3 = 0$
5. Transferir conteúdo do galão de 5 litros para o de 3 litros	$g5 = 0$	$g3 = 2$
6. Encher o galão de 5 litros	$g5 = 5$	$g3 = 2$
7. Transferir conteúdo do galão de 5 litros para o de 3 litros	$g5 = 4$	$g3 = 3$



# Conceito de Algoritmo

## Definição:

*Conjunto de regras e operações bem definidas e ordenadas cuja execução passo a passo objetiva a solução de um problema em um número finito de etapas.*

**Passo a passo:** cada passo é completado antes que o próximo inicie.

**Bem definidas:** cada passo é completamente definido a partir da entrada atual e passos anteriores.

# Conceito de Algoritmo

## Algoritmo:

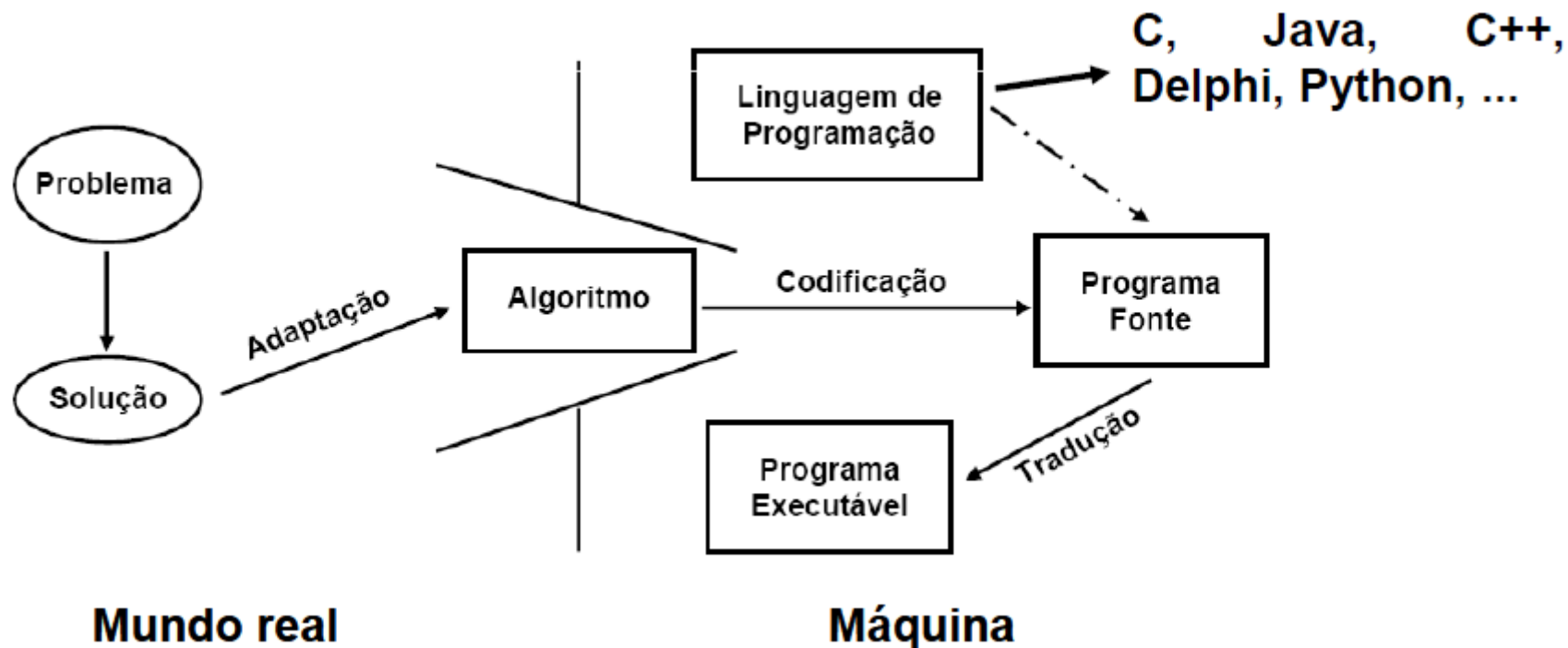
uma representação de solução para um problema.

## Programa:

Sequência de instruções a serem executadas por um computador  
– de acordo com um algoritmo

# Conceito de Algoritmo

O algoritmo no contexto computacional desempenha um papel fundamental: é o **elo de ligação** entre o mundo real e o computacional.



# Características de Algoritmos

**Finitude:** um algoritmo deve sempre terminar após um número finito de passos.

**Definição:** cada passo de um algoritmo deve ser precisamente definido, sem ambiguidades.

**Entradas:** um algoritmo deve ter zero ou mais entradas.

**Saídas:** um algoritmo deve ter uma ou mais saídas.

**Efetividade:** cada operação deve executar em tempo finito

# Representação de Algoritmos

Há quatro maneiras de representação:

- Descrição Narrativa ou Linguagem Natural
- Fluxograma
- Pseudocódigo
- Linguagem de Programação

# Representação de Algoritmos

Descrição Narrativa ou Linguagem Natural:

*É a descrição daquilo que se quer fazer, similar a comunicação humana como em uma receita de bolo, manual de produto, direções para um lugar, regras de um jogo, etc.*

1. Ler A e B
2. Somar A e B
3. Caso a soma seja maior que cinco, imprimir “Maior que 5”
4. Caso contrário, imprimir “Menor que 5”

# Representação de Algoritmos

Fluxograma:

- São representações gráficas do algoritmo, enfatizando passos individuais e o **fluxo de controle** (execução).
- Permitem a visualização do fluxo lógico através do tempo

# Representação de Algoritmos

Fluxograma - símbolos:



início / fim



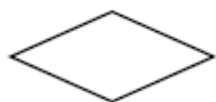
ação



conector



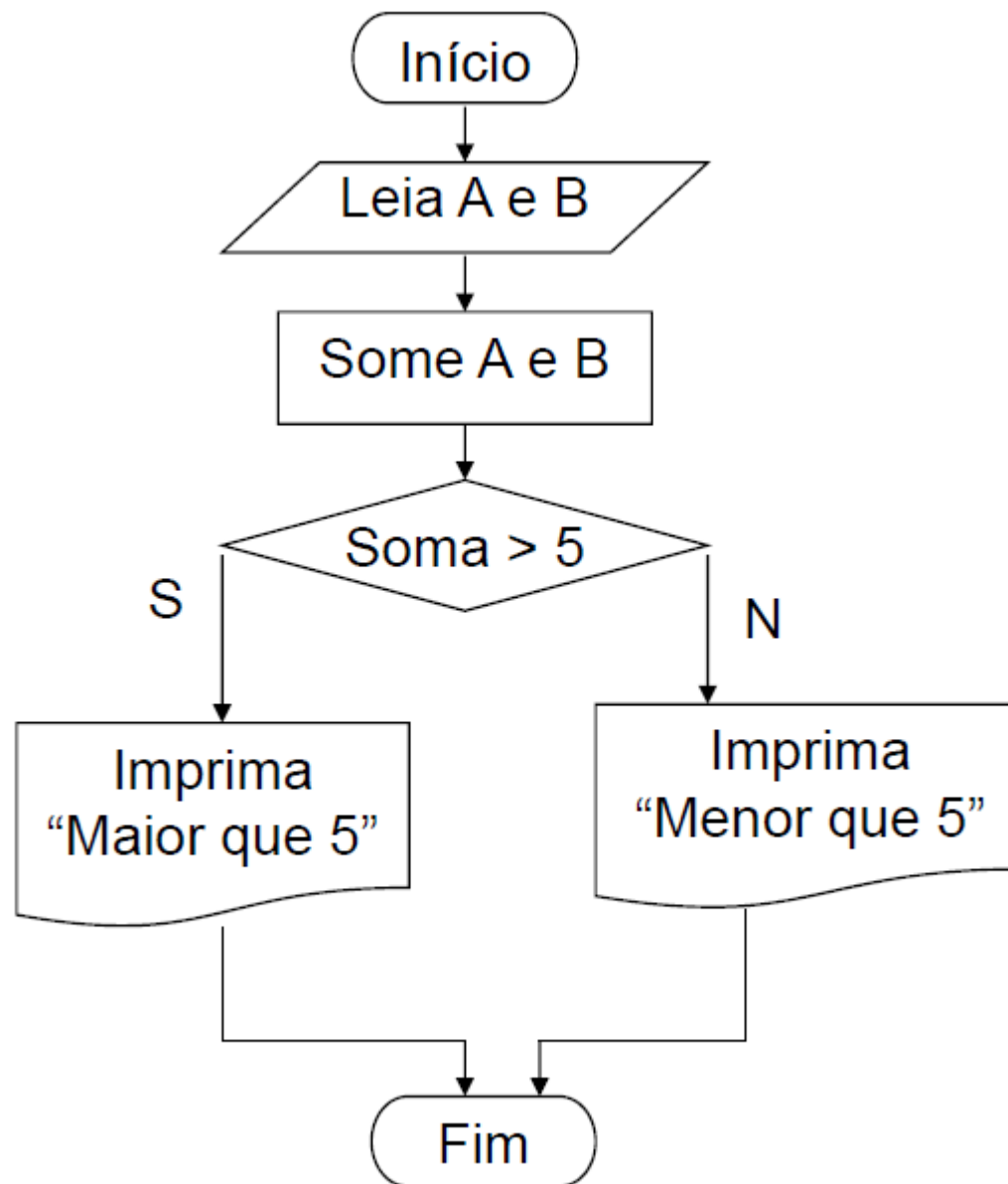
leitura



decisão



impressão





# Representação de Algoritmos

Pseudocódigo:

- Linguagem intermediária entre linguagem natural e linguagens de programação para descrição de algoritmos.
- Utiliza a estrutura de linguagens de programação porém visa a interpretação por humanos e não por computadores.

```
ALGORITMO SOMA
Entradas: nenhuma
Saídas: Soma
Declarar N1, N2, Inteiros
LER N1, N2
Soma <- N1 + N2
  SE Soma > 5 ENTÃO
    IMPRIMIR "Maior que 5"
  SENÃO
    IMPRIMIR "Menor que 5"
  FIM SE
FIM ALGORITMO
```

# Representação de Algoritmos

Linguagem de programação:

- Linguagem de alto ou baixo nível que pode descrever algoritmos.
- Utiliza estruturas e pode ser processada por computadores, sendo compilada ou interpretada.

```
#include <stdio.h>
int main(void)
{
    int a=0,b=0,soma=0;
    scanf("%d, %d", &a, &b);
    soma = a + b;
    if(soma > 5)
        printf("%d é maior que 5\n", soma);
    else
        printf("%d é menor que 5\n", soma);
    return soma;
}
```

**Leitura: Joel on Software – The Law of Leaky Abstractions**

<https://www.joelonsoftware.com/2002/11/11/the-law-of-leaky-abstractions/>

# Raciocínio lógico na construção de algoritmos

## Lógica:

"A lógica nos ensina a colocar ordem no pensamento"

↳ "pensar com ordem"

## Lógica de programação:

Objetiva a construção de algoritmos para solução de problemas através da ordenação do pensamento.

## Lógica matemática:

Prova de declarações matemáticas

# Raciocínio lógico na construção de algoritmos

Análise do pseudocódigo anterior:

```
ALGORITMO SOMA
Entradas: nenhuma
Saídas: Soma
Declarar N1, N2, Inteiros
LER N1, N2
Soma <- N1 + N2
SE Soma > 5 ENTÃO
    IMPRIMIR "Maior que 5"
SENÃO
    IMPRIMIR "Menor que 5"
FIM SE
FIM ALGORITMO
```

Lógica de programação:  
ordem de declarações  
para resolver um  
problema.

Lógica matemática:  
expressões que podem ser  
avaliadas como verdadeiras  
ou falsas

# Estrutura de Algoritmos

Teorema do Programa Estruturado:

*Três estruturas são suficientes para representar qualquer função computável.*

sequência



seleção



iteração



Primeira observação baseada no comportamento de processadores e da máquina de Turing.

## Flow diagrams, turing machines and languages with only two formation rules

Full Text: PDF

Authors: [Corrado Böhm](#) [International Computation Centre and Istituto Nazionale per le Applicazioni del Calcolo, Roma, Italy](#)  
[Giuseppe Jacopini](#) [International Computation Centre and Istituto Nazionale per le Applicazioni del Calcolo, Roma, Italy](#)



1966 Article

Published in:



· Magazine  
 Communications of the ACM [CACM Homepage archive](#)  
 Volume 9 Issue 5, May 1966  
 Pages 366-371  
[ACM](#) New York, NY, USA  
[table of contents](#) doi> [10.1145/355592.365646](https://doi.org/10.1145/355592.365646)

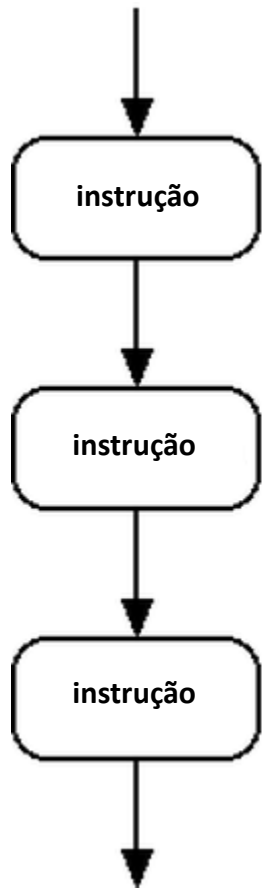
Bibliometrics

- Downloads (6 Weeks): 27
- Downloads (12 Months): 181
- Downloads (cumulative): 3,315
- Citation Count: 217

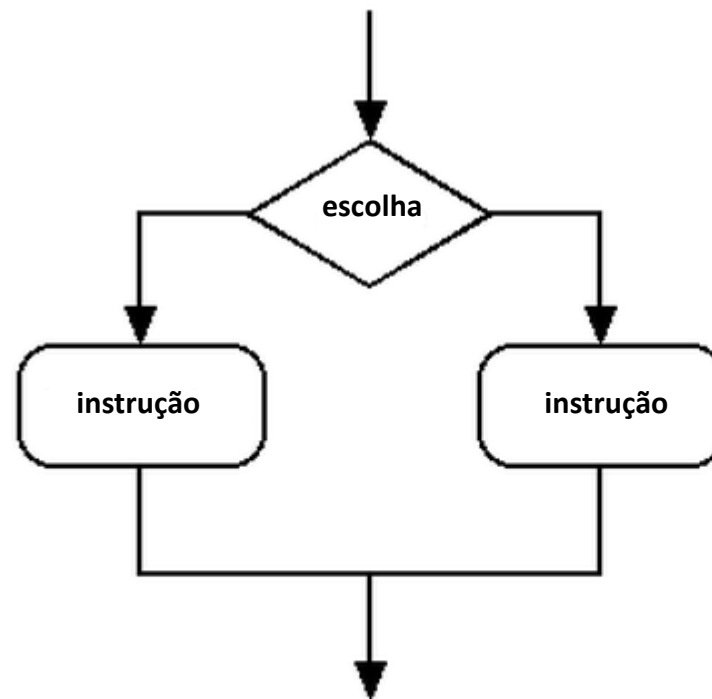
# Estrutura de Algoritmos

Teorema do Programa Estruturado:

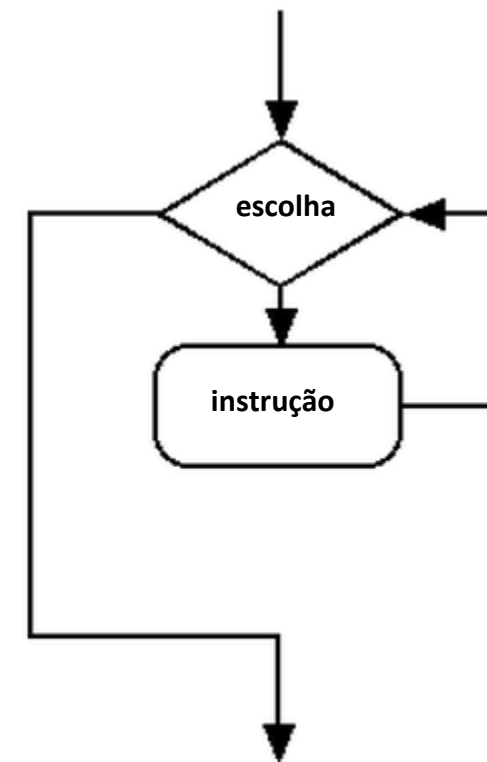
sequência



seleção



iteração



A iteração é uma seleção onde uma das instruções retorna a escolha e a outra sai da estrutura

# Tipos de Dados

## Variável:

- Porção de memória onde um valor pode ser armazenado e lido.
- No programa, esse espaço de memória é associado a um nome (identificador) e um tipo.
- No executável, é apenas um endereço e um número.

# Tipos de Dados

## Identificadores:

Nomes utilizados para referenciar variáveis, funções ou outros objetos.

- Letras, dígitos e sublinhados (\_).
- Não podem começar com dígito. (impediria a especificação de literais)
- Não podem ter espaços.
- Não podem ser iguais a palavras-chave e nem repetidos.



# Tipos de Dados

Dados primitivos:

- Caractere, inteiro, real, enumerado
- Constantes

Geralmente a declaração do dado é feita junto com o identificador:

- real a;
- caractere x,z;

A linguagem C usa tipos estáticos e exige a declaração de variáveis com tipos.

# Comandos

São disponibilizados em bibliotecas ou definidos em outros algoritmos:

- ler
- escrever