

# ALGORITMOS E PROGRAMAÇÃO DE COMPUTADORES

**Disciplina: 113476**

Profa. Carla Denise Castanho

Universidade de Brasília - UnB  
Instituto de Ciências Exatas - IE  
Departamento de Ciência da Computação - CIC

# 1. REPRESENTAÇÃO DE ALGORITMOS

# Lógica

## ► Lógica:

- Relaciona-se com a **razão**, investiga a **correção** do **pensamento/raciocínio**, ou seja, quais operações são **válidas** ou não.
- Estuda e ensina a colocar **ordem** no pensamento.

# Algoritmo

## ► Algoritmo:

- É uma **sequência** finita de **passos** para realizar uma tarefa ou **solucionar um problema**.
- Na prática, criamos algoritmos com o objetivo de **automatizar**: queremos que o resultado de uma tarefa possa ser **conhecido** ou obtido repetidamente, no mesmo **intervalo de tempo**, com a mesma **qualidade**.

Todos os passos conhecidos



Automação de uma Tarefa



Repetibilidade

# Algoritmo

## ▶ Algoritmo (cont.):

- ▶ Não é exclusivo da computação!
- ▶ Fazem parte do dia-a-dia de todas as pessoas:
  - ▶ Instruções para uso de medicamentos;
  - ▶ Indicações de como montar um móvel;
  - ▶ Receitas culinárias;
- ▶ A **ordem** é muito importante! Se você trocar os passos, o algoritmo pode não dar certo!

Não são automatizados,  
mas são algoritmos!

# Dado e Informação

- ▶ Um **dado** é uma **sequência de símbolos**. São elementos conhecidos de um problema (são os fatos em sua forma primária, como observamos no mundo).
- ▶ Quando organizamos os dados atribuímos **significado**, eles ganham **valor**, tornam-se **informação**. Ex.:
  - ▶ 1,89 - dado
  - ▶ “a altura de José é 1,89 m” - informação
- ▶ Um mesmo conjunto de dados pode ser utilizado para representar informações de natureza distintas. Ex: *alfabeto vs línguas*.
  - ▶ O C R T D M A U P O - dados
  - ▶ C O M P U T A D O R - informação

**DADO** e **INFORMAÇÃO** não são termos exclusivos da computação, foram cunhados por outras áreas, tais como a *semiótica*.

# Dados de Entrada e Saída

- ▶ A solução de qualquer problema envolve um **processamento**, que manipula **dados de entrada** e gera **dados de saída**:



# Lógica de Programação

- ▶ Nesta disciplina, queremos aprender a:
  - ▶ Dado um **problema**,
  - ▶ determinar uma **sequência de instruções** tal que,
  - ▶ fornecidos os **dados de entrada**,
  - ▶ a **execução** da sequência da instruções
  - ▶ resulte como **saída** a **solução** do problema.



# Lógica de Programação

## ► Exemplo:

### Algoritmo para trocar um pneu

#### **Algoritmo para trocar um Pneu**

**Entrada:** carro, macaco mecânico, pneu reserva

**Saída:** carro com o pneu trocado

Faça:

```
Afrouxar ligeiramente os parafusos;  
Suspender o carro;  
Retirar os parafusos e o pneu;  
Colocar o pneu reserva;  
Apertar os parafusos;  
Abaixar o carro;  
Dar o aperto final nos parafusos.
```

# Representação de Algoritmos

- ▶ Existem diversas formas de **representação** de algoritmos.
- ▶ Não existe um consenso sobre qual é a melhor forma, cada uma tem vantagens e desvantagens.
- ▶ Algumas formas tratam os problemas apenas no nível lógico, **abstraindo os detalhes** relacionados de implementação.
- ▶ Outras formas entram em mais detalhes e são mais **específicas**, porém que podem **obscurecer a idéia principal**.

# Representação de Algoritmos

- ▶ Entre as formas mais conhecidas, podemos citar:
  - ▶ Descrição Narrativa;
  - ▶ Fluxograma Convencional;
  - ▶ Pseudocódigo, também conhecido como *portugol*.

# Descrição Narrativa

- ▶ Nesta forma de representação os algoritmos são expressos diretamente em **linguagem natural**.
- ▶ Uso da linguagem natural pode gerar **ambiguidades** e **imprecisões**.
- ▶ Por exemplo, no algoritmo do pneu, o que significa “Afrouxar *ligeiramente* o parafuso”?
- ▶ Não seria mais preciso dizer “Afrouxar o parafuso, girando o mesmo  $33.8^\circ$  no sentido anti-horário”?

# Descrição Narrativa

- ▶ Quando criamos algoritmos, frequentemente precisamos utilizar estruturas **condicionais** e de **repetição**. Isto é, precisamos **tomar decisões** ou **repetir passos**.
- ▶ Vamos ver um exemplo...

# Descrição Narrativa

## Algoritmo para trocar uma lâmpada

### Algoritmo para trocar uma lâmpada:

Faça:

**Se** a lâmpada antiga estiver queimada, faça:

Pegar uma escada

Posicionar a escada debaixo da lâmpada

Buscar uma lâmpada nova

Subir na escada

Retirar a lâmpada antiga

Colocar a lâmpada nova

**Enquanto** a lâmpada nova não acender, faça:

Retirar a lâmpada antiga

Colocar uma lâmpada nova

**Caso contrario**, não é necessário fazer nada.

# Descrição Narrativa

## Algoritmo para trocar uma lâmpada

### Algoritmo para trocar uma lâmpada:

Faça:

**Se** a lâmpada antiga estiver queimada, faça:

Pegar uma escada

Posicionar a escada debaixo da lâmpada

Buscar uma lâmpada nova

Subir na escada

Retirar a lâmpada antiga

Colocar a lâmpada nova

**Enquanto** a lâmpada nova não acender, faça:

Retirar a lâmpada antiga

Colocar uma lâmpada nova

**Caso contrario,** não é necessário fazer nada.

CONDICIONAL

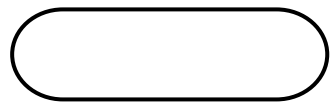
REPETIÇÃO

# Fluxograma

- ▶ É uma representação **gráfica** de algoritmos, que utiliza **formas geométricas** diferentes para representar **ações** (instruções, comandos) e **setas** para representar a **sequência**.
- ▶ Essa organização geralmente facilita o entendimento das idéias contidas nos algoritmos, o que justifica sua popularidade.
- ▶ É uma forma intermediária entre a descrição narrativa e o pseudocódigo (que veremos em seguida), pois é **menos imprecisa** que a primeira, porém, ainda não se preocupa com **detalhes** de implementação do programa.



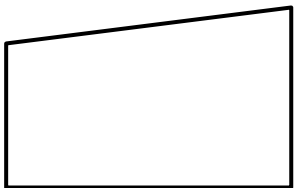
# Fluxograma



Início e Fim



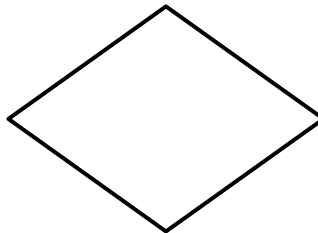
Operação de  
Atribuição



Processamento de  
entrada de dados



Processamento de  
saída de dados



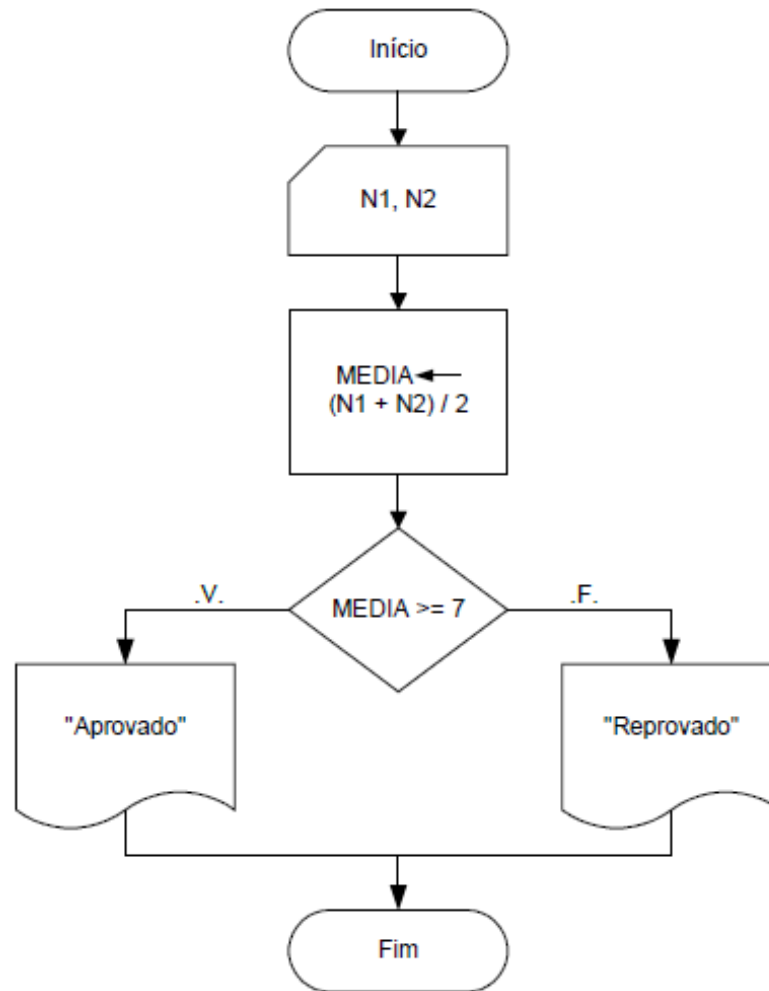
Decisão



Seta de Fluxo de  
Dados

# Fluxograma

- ▶ Ex.: Cálculo da média de um aluno sob a forma de um fluxograma, sendo que o aluno é aprovado se a média for maior ou igual a 7 (sete).
- ▶ Partindo do símbolo inicial, há sempre um único caminho orientado a ser seguido, representando a existência de uma única seqüência de execução das instruções



# Pseudocódigo

- ▶ Esta forma de representação é rica em **detalhes**, tais como definição dos **tipos** e **nomes** das **variáveis** usadas no algoritmo e controle de **índices** e **tamanho** dos dados.
- ▶ É uma representação mais **precisa** e detalhada que as demais, porém, suficientemente **genérica** para permitir a **tradução** de um algoritmo para várias **linguagens de programação** específicas, de maneira praticamente direta.
- ▶ Também conhecida como *Portugol*, *Linguagem Estruturada* ou *Linguagem Algorítmica*.

**ESTA É A REPRESENTAÇÃO QUE  
UTILIZAREMOS NA DISCIPLINA!**

# Pseudocódigo

- ▶ A forma geral da representação de um algoritmo em pseudocódigo:

## Pseudocódigo - Estrutura Básica

```
Algoritmo <nome do algoritmo>.  
<declaração de variáveis (nome e tipo)>  
<subalgoritmos>  
Início  
<corpo do algoritmo>  
Fim.
```

- ▶ *Algoritmo*, *Início* e *Fim* são **palavras-chave** (ou *palavras reservadas*), utilizadas respectivamente para: *declarar* e dar um **nome** ao algoritmo, e *delimitar* o **corpo** do algoritmo.
- ▶ Como veremos, algoritmos utilizam **variáveis** para manipular valores, e podem ser divididos em tarefas menores, os **subalgoritmos**.
- ▶ Nas próximas aulas, estudaremos em detalhes esta representação.