

Fundamentos para Construção de Algoritmos

Agenda

- Conceitos iniciais;
- Algoritmos: formas de representação;
- Técnicas de desenvolvimento de algoritmos;
- Exercícios.

Conceitos Iniciais



Problema:

- Um problema será uma pergunta (ou situação) de caráter geral a ser respondida (ou resolvida);
- Um problema é descrito identificando-se:
 - quais são os possíveis dados (ou entradas);
 - quais são os possíveis resultados (ou saídas);
 - que condições (ou relações) devem ser satisfeitas para que um particular resultado seja uma saída aceitável para uma certa entrada.

Conceitos Iniciais

Lógica de Programação:

- Lógica de programação é a técnica de encadear pensamentos para atingir determinado objetivo. Ou seja, permite definir a sequência lógica para o desenvolvimento de sistemas e programas computacionais.



Conceitos Iniciais

Algoritmo:

- É uma sequência de passos finitos com o objetivo de solucionar um problema;
- Quando se tem um problema, o objetivo é solucioná-lo;
- Portanto, algoritmo é um caminho para a solução de um problema e, em geral, os caminhos que levam a uma solução são muitos.

Exemplo para somar dois números:

Ler dois números

Calcular a soma dos números lidos

Imprimir o resultado da soma

Conceitos Iniciais

Implementação:

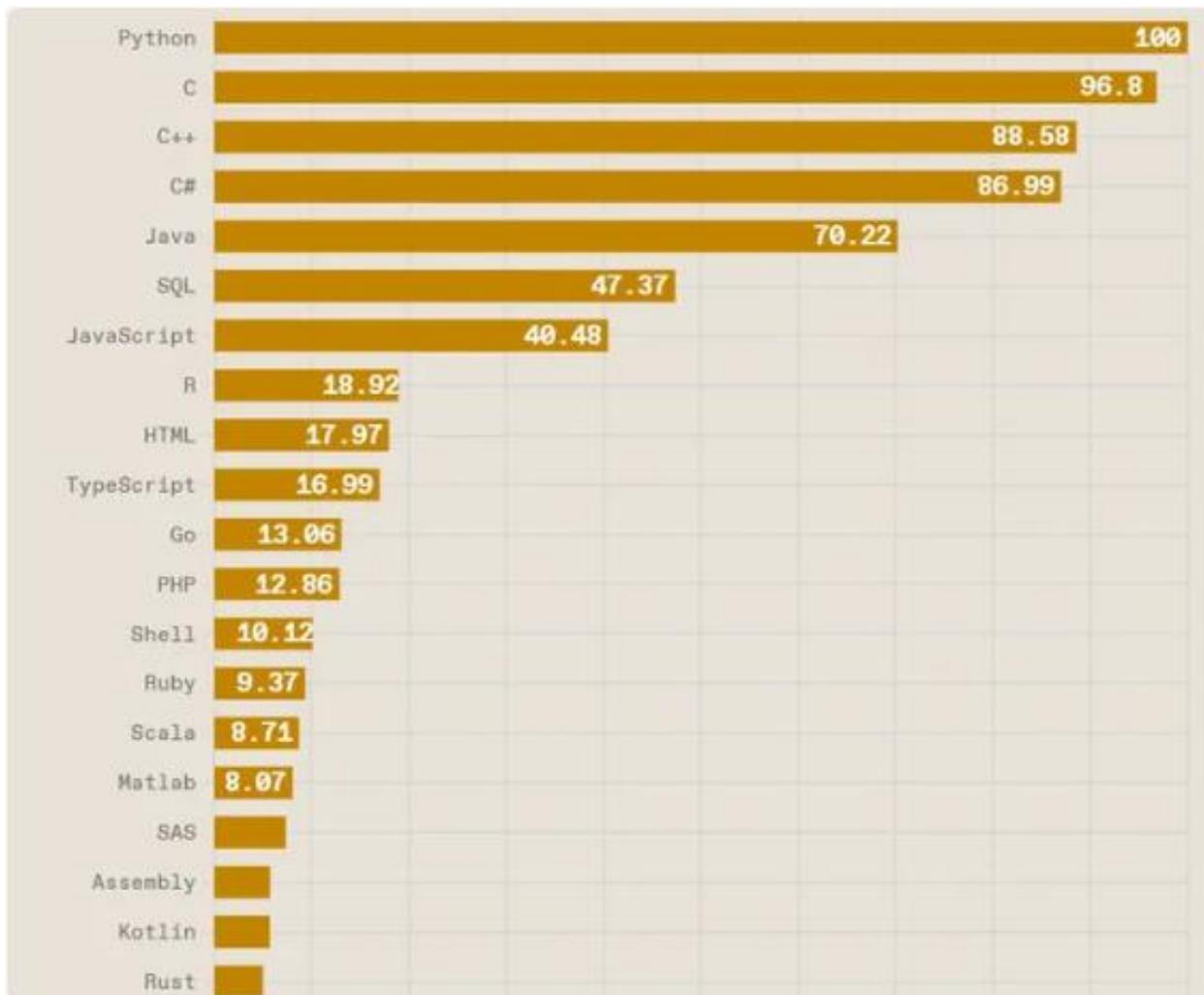
- Codificação;
- Tradução do algoritmo para uma linguagem de programação específica;
- É importante demonstrar (ou verificar) que o algoritmo (ou programa) realmente resolve o problema proposto.
 - **Teste de Mesa** (simula a execução do algoritmo sem usar o computador empregando apenas “papel e lápis”).

Conceitos Iniciais

Linguagem de Programação:

- Por meio de símbolos (convenções ou palavras reservadas), permite exprimir de uma forma “natural” o raciocínio algorítmico para o computador.
- Linguagens Estruturadas
 - C, Pascal, Clipper, Cobol
- Linguagens Orientadas a Objetos
 - C++, Delphi, Java, R, Jasp, Python

Linguagens de Programação em 2022-2023:



Confira as 20 linguagens de programação mais populares de acordo com o Instituto de Engenheiros Elétricos e Eletrônicos (IEEE). (Imagen: IEEE Spectrum/Divulgação)

Conceitos Iniciais

Programa:

- É uma sequência de instruções codificadas em uma linguagem de programação, e para ser executado, precisa ser armazenado na memória do computador.

Exemplo para somar dois números em Python:

```
Número1 = int(input("Digite um número: "))
Número2 = int(input("Digite outro número: "))
Soma = Número1 + Número2
print("O resultado da soma é" , Soma)
```

Conceitos Iniciais

Processamento de Dados:

Conjunto de dados de entrada que são transformados por meio do processamento de algoritmos (ou programas) em um conjunto de informações de saída.

- Entrada: dados inseridos pelo usuário, como por ex.: nota de um aluno, nome, preço de custo de um produto, preço de venda etc.;
- Processamento: cálculos e manipulação dos dados de entrada, como por ex.: calcular a média das notas de um aluno, onde como entrada seriam as notas, dividido pela quantidade de notas;
- Saída: impressão (na tela) do resultado obtido no processamento.



Conceitos Iniciais

Sistemas de Informação:

- Conjunto organizado de pessoas, hardware, software, redes de comunicações e recursos de dados que **coleta** (entrada), **transforma** (algoritmo) e **dissemina** (saída) as informações em uma organização.



Conceitos Iniciais

- ✓ Problema
- ✓ Lógica
- ✓ Algoritmo
- ✓ Programa
- ✓ Implementação
- ✓ Linguagem de Programação
- ✓ Processamento de Dados
- ✓ Sistemas de Informação

Formas de Representação

Formas de Representação de Algoritmos:

Os três tipos mais comuns de algoritmos são:

- descrição narrativa;
- fluxograma; e
- pseudocódigo ou portugol.

Formas de Representação

Descrição Narrativa:

Consiste em analisar o enunciado do problema e escrever, utilizando uma linguagem natural (por exemplo, a língua portuguesa), os passos que devem ser seguidos para a resolução do problema.

Segue exemplo para a *Troca de um pneu furado*:

- Afrouxar ligeiramente as porcas
- Suspender o carro
- Retirar as porcas e o pneu
- Colocar o pneu reserva
- Apertar as porcas
- Abaixar o carro
- Dar o aperto final nas porcas



Formas de Representação

Descrição Narrativa:

- Segue outro exemplo para a *Tomar banho*:

Início

- Pegar a toalha e roupas;
- Entrar no banheiro e fechar a porta;
- Tirar a roupa que está usando;
- Ligar o chuveiro e esperar a água ficar na temperatura certa;
- Entrar no box (ou espaço destinado para o banho);
- Se molhar, ensaboar, enxaguar = banho;
- Fechar o chuveiro;
- Pegar a toalha e se secar;
- Sair do box;
- Se vestir com a roupa limpa;
- Sair do banheiro.

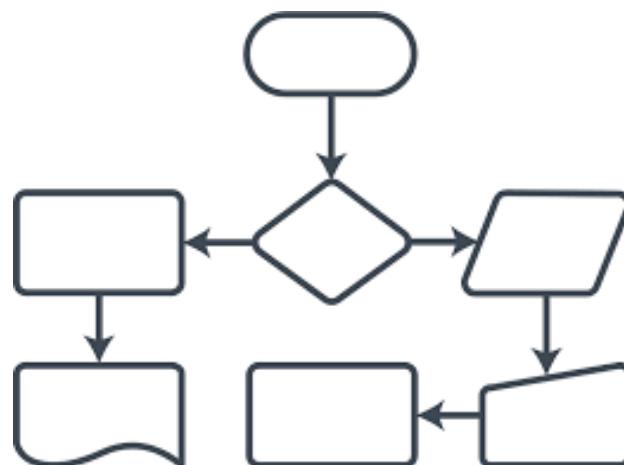
Fim



Formas de Representação

Fluxograma:

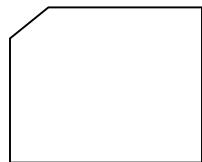
- Consiste em analisar o enunciado do problema e escrever, utilizando símbolos gráficos predefinidos, os passos que devem ser seguidos para a resolução do problema.



Fluxograma – Símbologia Básica



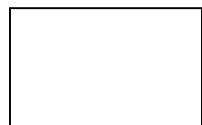
Início e final do fluxograma



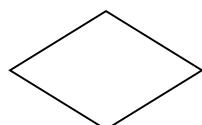
Entrada de dados: digitação durante a execução



Saída de dados: impressão no vídeo



Atribuição: para cálculos e atribuição de valores

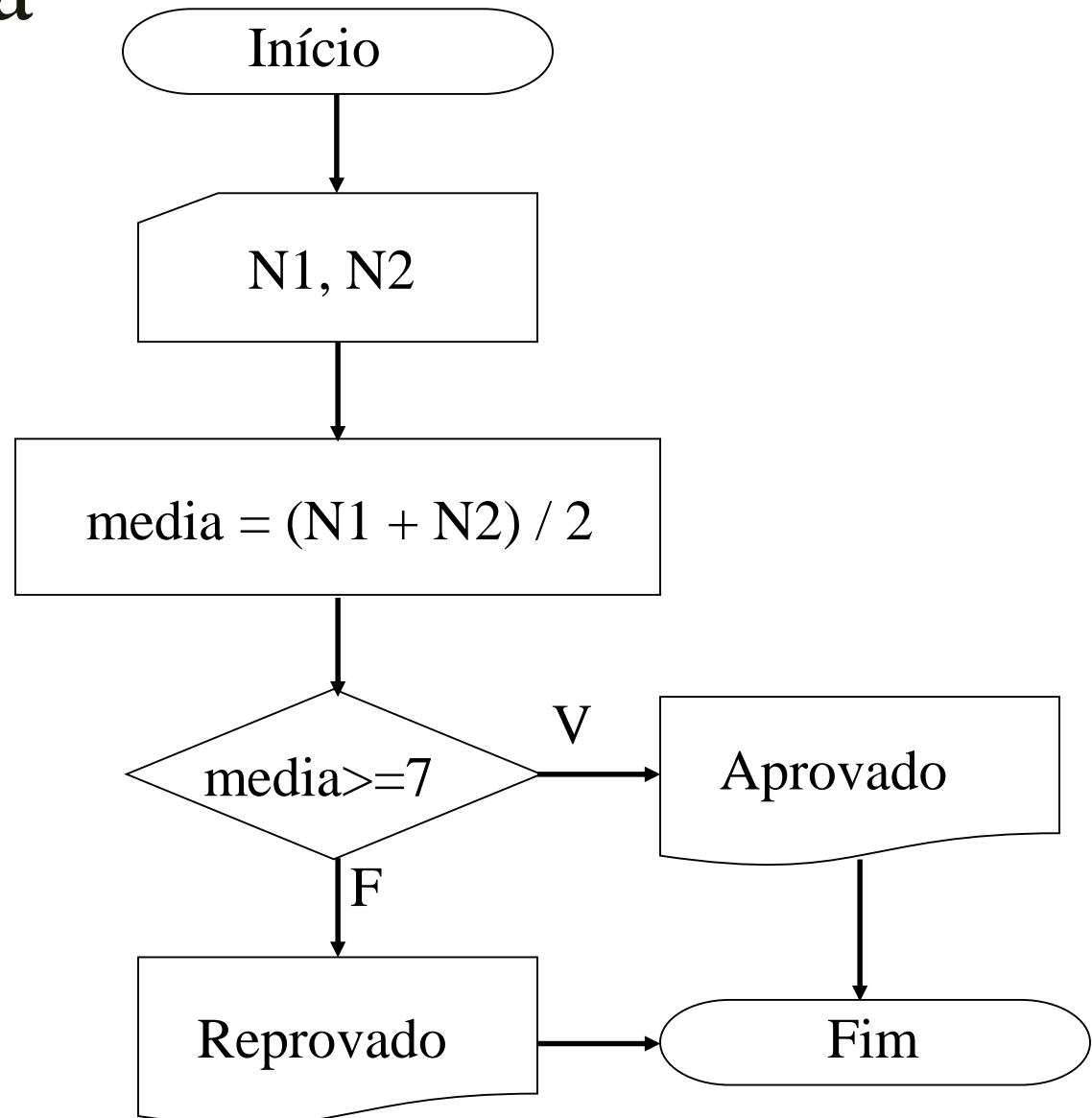


Desvio condicional: indica a tomada de decisão



Fluxo da informação: indica o sentido dos dados

Fluxograma – Exemplo



Formas de Representação

Pseudocódigo (Portugol):

- Assemelha-se à forma em que os programas são escritos, segue exemplo:

leia N1

leia N2

media = (N1 + N2) / 2

se media >= 7 :

imprima (“Aprovado”)

senao :

imprima (“Reprovado”)

Formas de Representação

Exemplo / Exercício:

Algoritmo para mostrar o resultado da multiplicação de dois números.

1) Descrição Narrativa

Passo 1 - Obter os dois números que serão multiplicados.

Passo 2 - Multiplicar os números.

Passo 3 - Mostrar o resultado obtido na multiplicação.

2) Faça o fluxograma.

3) Faça o pseudocódigo.

Desenvolvendo Algoritmos

Um algoritmo correto deve possuir 4 qualidades:

- 1- O algoritmo deve ter um início;
- 2- Cada passo do algoritmo deve ser uma instrução que possa ser realizada;
- 3- A ordem dos passos deve ser precisamente determinada;
- 4- O algoritmo deve ter um fim.

Desenvolvendo Algoritmos

Passos para desenvolver algoritmos:

Passo 1: ler a especificação do problema até o final;

Passo 2: levantar e analisar todas as **saídas** exigidas na especificação do problema;

Passo 3: levantar e analisar todas as **entradas** citadas na especificação do problema;

Passo 4: verificar se é necessário gerar **valores internamente** ao algoritmo e identificar os valores iniciais de cada variável;

Passo 5: levantar e analisar todas as **transformações** necessárias para, dadas as entradas produzir as saídas especificadas (**entrada → saída**);

Passo 6: **testar** cada passo do algoritmo (“teste de mesa”), verificando se as transformações/cálculos estão conduzindo aos objetivos desejados (*utilizar, sempre que possível, valores de teste que permitam prever os resultados*).

Tipos de Erros

Início

media = (N1 + N2) / 2

leia N1

leia N2

se media >= 7 :

imprima (“Aprovado”)

seno

imprima (“Em exame”)

Fim

Erro de
Lógica/Semântica

Erro de Sintaxe

Referências

- Lopes, A. & Garcia, G. – Introdução a Programação.
- Saliba, W. – Técnicas de Programação.
- Schildt – C Completo e Total.

Exercícios

1. Elabore algoritmo para fazer um bolo (descrição narrativa).
2. Elabore algoritmo para sacar dinheiro em um caixa de banco 24 horas (descrição narrativa).
3. Elabore um algoritmo para fazer pipoca numa panela de fogão, usando manteiga, sal e milho de pipoca (descrição narrativa).
4. Elabore algoritmos (3 tipos) para calcular a média da idade de três alunos, estas informadas durante a execução do mesmo.

Exercícios

5. Faça o “teste de mesa” para identificar os dados de entrada, processamento e saída no algoritmo a seguir:

- 1) Entre com o código do produto
- 2) Entre com o preço do produto
- 3) Entre com a quantidade do produto
- 4) Calcule o valor total (preço x quantidade)
- 5) Mostre o código do produto e o valor total

TESTE DE MESA		
Entrada	Processamento	Saída