

Programação de Computadores

Professores:
Alcides T. B. Jr.
Amilton S. Martha
Ana Paula G. D. Carrare
Cristiane Camilo Hernandez
Manuel F. Paradela Ledón
Vagner da Silva



Motivação

- A programação de computadores é uma atividade que leva à **representação dos passos necessários à resolução de um problema** em **linguagem de programação**.
- Para dar início ao aprendizado dessa atividade, é importante compreender seu **contexto**, seu **propósito**, **os conceitos básicos subjacentes**, bem como tomar contato com as **ferramentas necessárias** para sua realização.

O que é um Programa de Computador?

- **Programa** é uma coleção de instruções que descrevem uma tarefa a ser realizada por um computador.
- Um programa é a formalização de um algoritmo em qualquer **linguagem de programação**, capaz de ser transformado em **instruções** que serão executadas por um computador, gerando os resultados esperados.

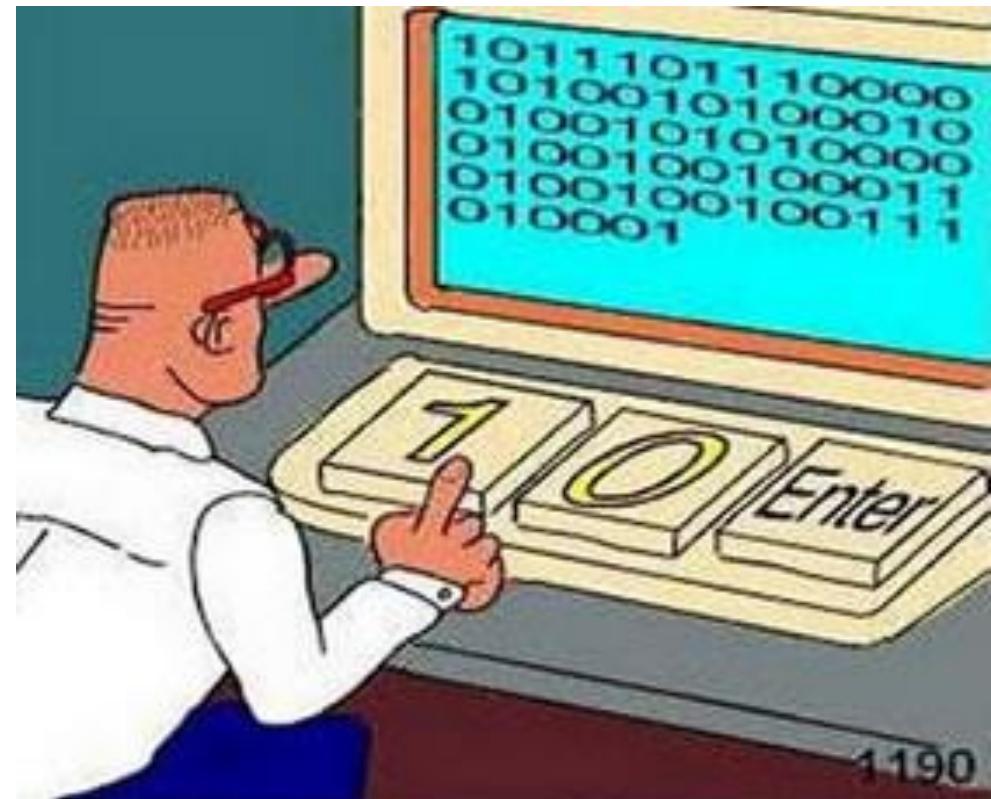
Fonte: Wikipédia http://pt.wikipedia.org/wiki/Programa_de_computador

- Sequência completa de instruções a serem executadas por computador.

Fonte: Miniaurélio Eletrônico versão 5.12 (2004)

O que é um Programa de Computador?

- As **instruções** estão escritas em uma linguagem que permite a **comunicação entre o programador e o computador** (0's e 1's) → Linguagem de programação



Como é construído?

- De forma bem genérica podemos dizer que construir um programa envolve as seguintes etapas:
 - Analisar o problema apresentado em busca de uma solução;
 - Escrever esta solução na forma de um **algoritmo**;
 - Implementar este algoritmo na linguagem de programação escolhida;
 - Realizar testes;
 - Corrigir erros, tanto sintáticos como lógicos;
 - Gerar pacote de instalação.

Algoritmo?

- Em computação pode ser definido como uma sequência de instruções (comandos) ou operações básicas, cuja execução, em tempo finito resolve um problema computacional.
- Ele pode ser representado graficamente (fluxograma), em pseudocódigo ou diretamente na linguagem escolhida para desenvolvimento do programa.

Exemplo de algoritmo do dia a dia

ALGORITMO 1.1 Troca de lâmpada

- pegar uma escada;
- posicionar a escada embaixo da lâmpada;
- buscar uma lâmpada nova;
- subir na escada;
- retirar a lâmpada velha;
- colocar a lâmpada nova.



- descer da escada
- guardar a escada

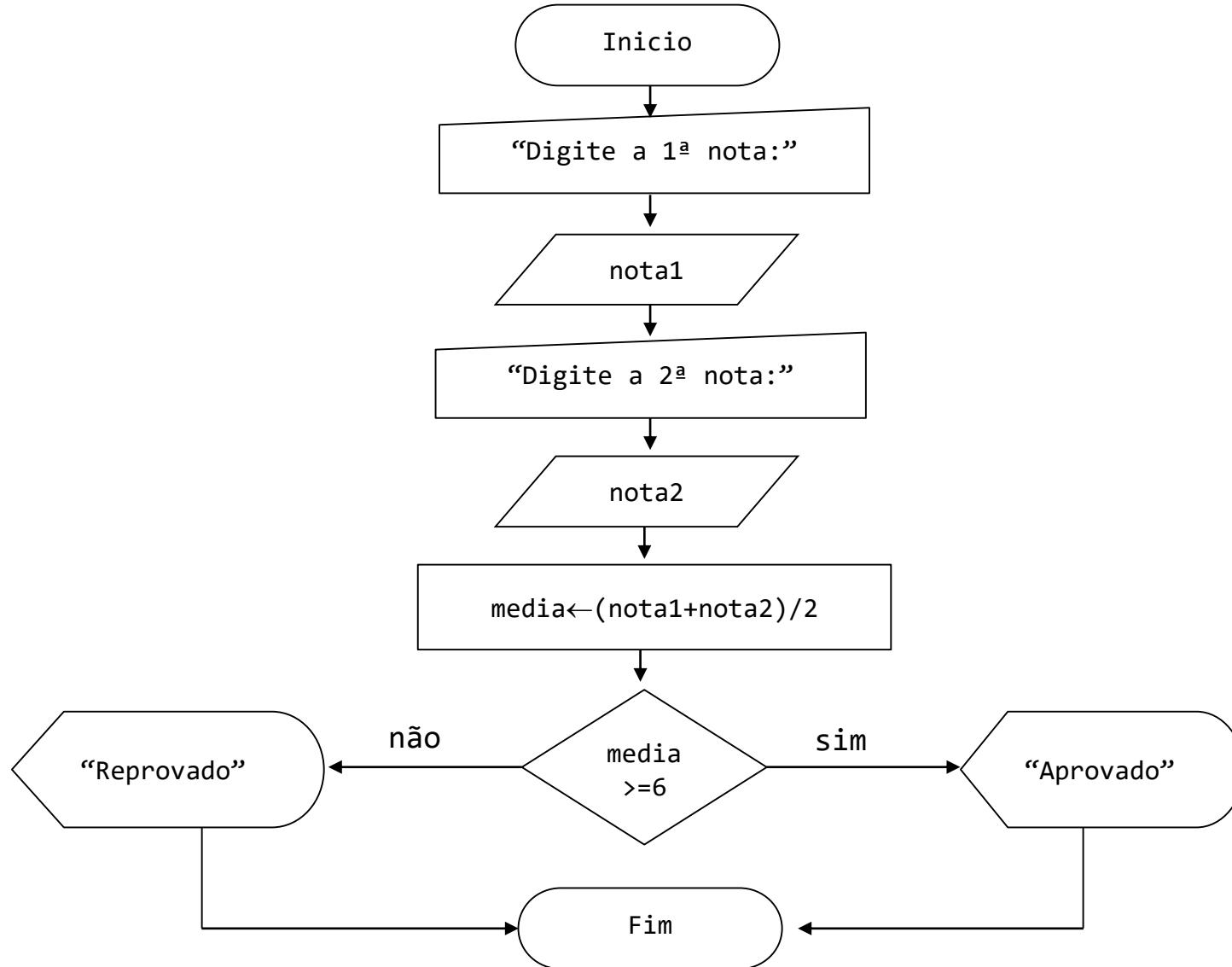
muitas vezes
podemos
aprimorar um
algoritmo...

Algoritmo – Representação Pseudo-Código (Portugol)

Algoritmo exemplo

```
real media, nota1, nota2
início
    escreva (“Digite a 1a nota: ”)
    leia (nota1)
    escreva (Digite a 2a nota: ”)
    leia (nota2)
    media ← (nota1 + nota2)/2
    se (media >= 6)
        escreva (“Aprovado”)
    senão
        escreva (“Reprovado”)
    fim se
fim
```

Algoritmo – Representação Gráfica (Fluxograma)



Por que o algoritmo é importante ?

- Porque **prepara uma lógica adequada**, correta, para resolver um determinado problema.
- A partir dele será construído um **programa**, que estará escrito em alguma **linguagem de programação** para que possa ser executado em um computador.
- Pode ser programado em diferentes linguagens de programação.
- É imprescindível considerar todas as operações ou passos necessários de um algoritmo e a ordem em que deverão ser executadas estas operações pelo computador.

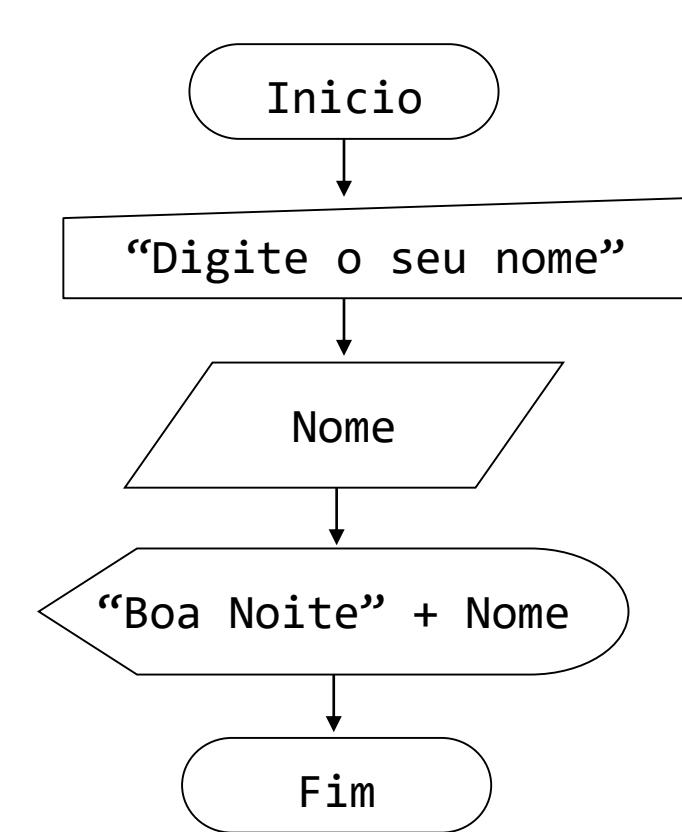
Exemplo

Queremos fazer um algoritmo que o usuário entre com o seu nome e o sistema imprima na tela “Boa noite “ e o nome dessa pessoa.

Algoritmo – Representação em Algoritmo e Fluxograma

Algoritmo exemplo
caractere nome
início

```
escreva("Digite o seu nome:")
leia(nome)
escreva("Boa Noite " + nome)
fim
```



Java

```
import javax.swing.*;  
public class Teste {  
    public static void main(String args[ ]) {  
        String nome;  
        nome = JOptionPane.showInputDialog("Digite o seu nome");  
        JOptionPane.showMessageDialog(null, "Boa noite " + nome);  
    }  
}
```

C

```
#include<stdio.h>

void main()
{
    char nome[30];
    printf("Digite o seu nome: ");
    gets(nome);
    printf("Boa Noite %s", nome);
}
```

C++

```
#include<iostream>
using namespace std;

void main()
{
    char nome[30];
    cout << "Digite o seu nome: ";
    gets(nome);
    cout << "Boa Noite" << nome;
}
```

Javascript

```
<script language="Javascript">  
    var nome;  
    nome = prompt("Digite o seu nome");  
    alert("Boa noite " + nome);  
</script>
```

Python

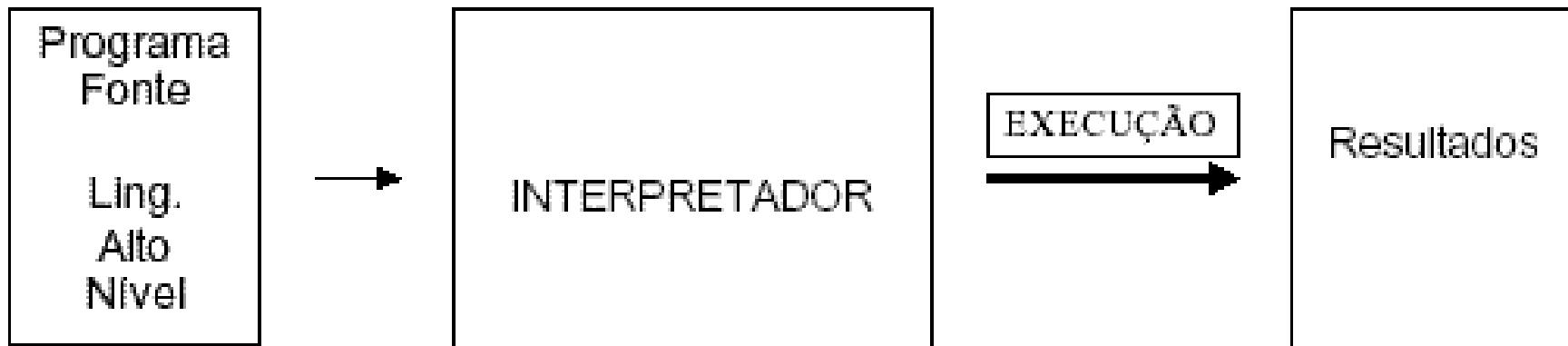
```
nome = input("Digite o seu nome: ")
print("Boa noite", nome)
```

Como a máquina entende os códigos?

- Para que o computador "entenda" um programa escrito em uma linguagem (de alto nível) é necessário um meio de **tradução** entre a linguagem de alto nível utilizada no programa e a linguagem de máquina.
- Para essa tarefa temos basicamente dois métodos:
 - Compilador
 - Interpretador

Interpretador

- Traduz e faz a checagem da sintaxe e envia para execução, instrução por instrução.
- Precisa estar presente todas as vezes que vamos executar o programa e o processo acima é repetido.



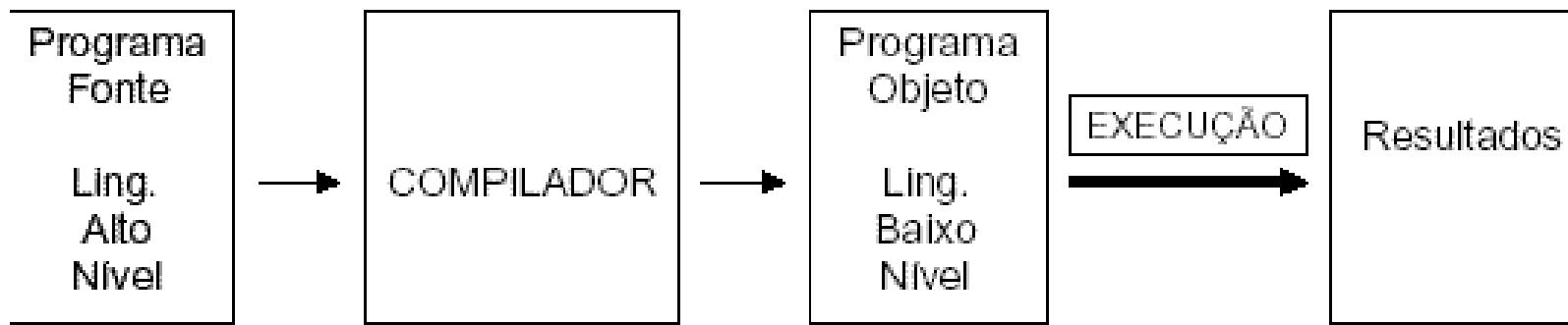
Vantagem: consome menos memória

Desvantagem: execução mais lenta

Exemplo: Uma página html é interpretada pelo Navegador.

Compilador

- Traduz o programa escrito em uma linguagem de programação para um programa equivalente escrito em linguagem de máquina (programa-objeto).



Vantagens:

- Velocidade de execução
- Oculta o código fonte

Desvantagem:

- A cada alteração no programa fonte é necessário gerar novamente o programa-objeto



Python

- É considerada uma linguagem interpretada, pois os programas em Python são executados por um interpretador.
- Existem duas maneiras de usar o interpretador:
 - no modo de linha de comando
 - no modo de script: arquivo .py

Exemplo 1

- Vamos criar um programa que apresente uma mensagem e faça a leitura de dados fornecidos pelo usuário. O algoritmo em pseudo-código ficará assim:

Algoritmo exemplo

caracter resposta

Inicio

escreva “Sejam bem vindos ao mundo da programação”

escreva “Digite seu nome”

leia resposta

escreva resposta

fim

Exercícios

1. Faça um programa em Python que escreva a mensagem “Análise e Desenvolvimento de Sistemas - Unicsul” na tela.

2. Faça um programa em Python que solicite ao usuário a sua profissão e mostre na tela a informação digitada.

3. Faça um programa em Python que solicite ao usuário a sua idade e apresente a informação na tela.

Exercícios

4. Faça um programa em Python que solicite o último sobrenome do usuário e mostre na tela a mensagem: “Família” e na sequência o sobrenome digitado.

5. Faça um programa em Python que solicite o esporte favorito de uma pessoa e apresente-o na tela.