

# Plano de Ensino e Introdução

---

RODRIGO LYRA

# Plano de Ensino - Conteúdos

- Classificação dos paradigmas de programação:
  - Introdução aos paradigmas de programação
  - Principais paradigmas de programação

# Plano de Ensino - Conteúdos

- Paradigmas declarativos e imperativos:
  - Vantagens e Desvantagens de cada paradigma
  - Introdução à orientação a objetos

# Plano de Ensino - Conteúdos

- Programação funcional:
  - Funções de Alta Ordem
  - Currying
  - Funtores e Mônadas

# Plano de Ensino - Conteúdos

- Programação em lógica:
  - Conceitos
  - Aplicações

# Plano de Ensino - Avaliações

- $MF = (M1 + M2 + M3) / 3$

$$M1 = (AC - \text{Definições de Paradigmas} * 4,00 + \\ AC - \text{Programação Funcional 1} * 6,00) / 10,00$$

$$M2 = (AC - \text{Programação Funcional 2} * 5,00 + \\ AC - \text{Programação Funcional 3} * 5,00) / 10,00$$

$$M3 = (\text{Trabalho - Programação Lógica} * 3,00 + \\ \text{Prova - Conceitos} * 7,00) / 10,00$$

- Média Final = 6,00

Limite de Falta = 25%

# Objetivo da disciplina

- Estudar conceitos sobre linguagens de programação e seus paradigmas, tornando-se apto a adotar decisões de projetos que envolvem a seleção de linguagens de programação mais adequadas para o desenvolvimento de aplicações computacionais.

# Organização de Computadores

- Os computadores executam milhões de operações a velocidades incríveis.
- As operações são organizadas por meio de um conjunto de instruções.
- Esse conjunto de instruções é o programa.
- Quem executa o programa é o processador.



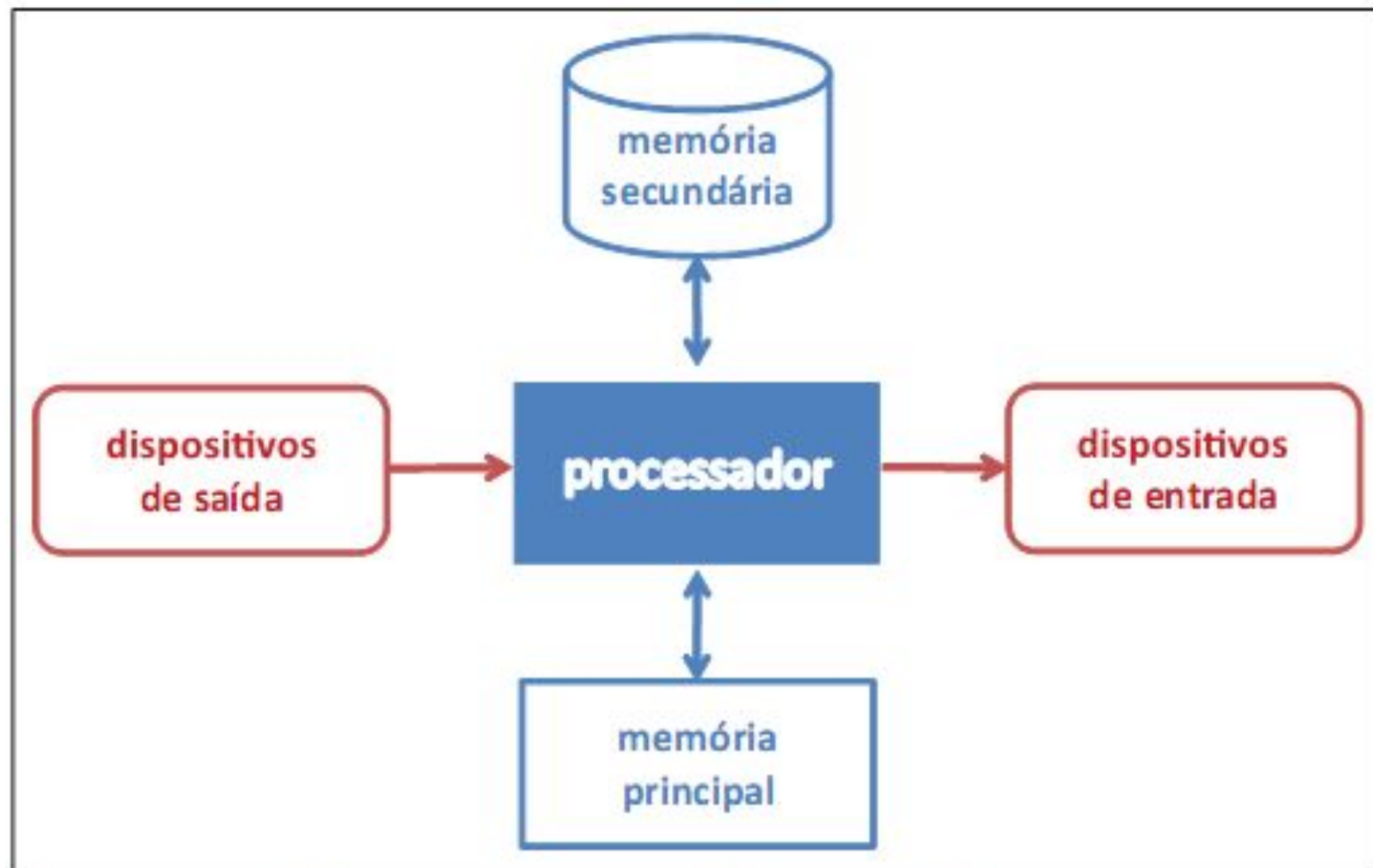


Figura 1 Representação dos principais componentes de um computador.

# Linguagens de Programação

- É necessário que os programas sejam escritos em uma linguagem que o computador entenda.
- Essa linguagem é chamada linguagem de máquina.
  - 0100 1111 1010 0110
- É pouco intuitiva para os seres humanos.
- Foram desenvolvidas linguagens de mais alto-nível.

# Linguagens de Programação

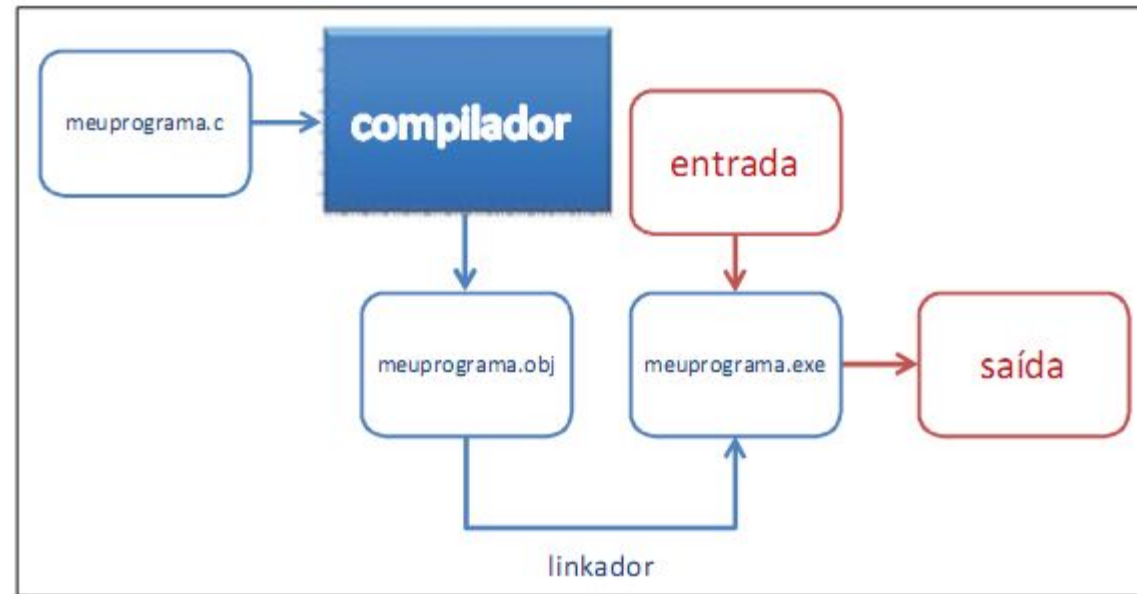
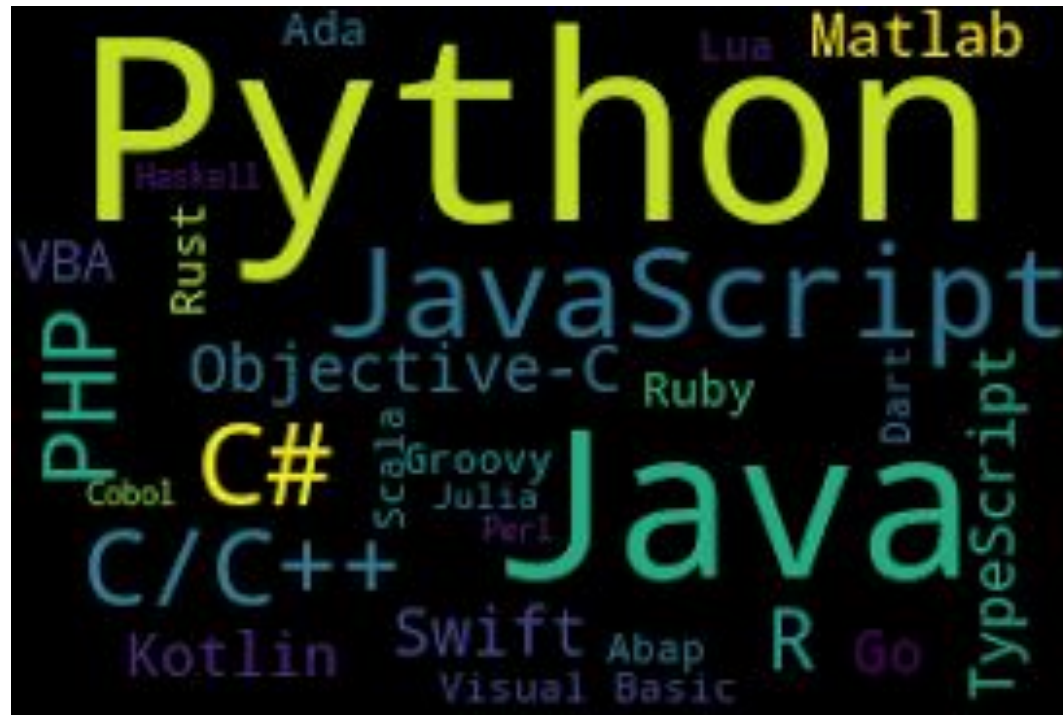


Figura 9 Processo de compilação de um código-fonte na linguagem de programação C.

# Linguagens de Programação

- Compiladas: todas as instruções são traduzidas para instruções de baixo nível antes da execução do programa.
- Interpretadas: as instruções são traduzidas para instruções de baixo nível à medida que vão sendo executadas.

# Linguagens de Programação



# Métodos de Programação

- A elaboração de programas complexos requer a utilização de um método sistemático de programação que permita obter programas confiáveis, flexíveis e eficientes. Salvetti (1998, p. 5)
- Sugere-se adotar um método de programação em etapas.

# Métodos de Programação

1. Análise do problema a ser resolvido;
2. Projeto do programa que resolverá o problema, incluindo a elaboração de um algoritmo e a definição das estruturas de dados a serem utilizadas;
3. Implementação (codificação) do programa;
4. Testes.

# Algoritmo

O que é um algoritmo?

- É a descrição de uma sequência de passos ordenadas e finita que deve ser seguida para a realização de uma tarefa.
- É uma descrição de como realizar uma tarefa independente da linguagem/hardware.
- Não se preocupa com detalhes de implementação em uma linguagem.



# Pseudocódigo

Faz uso de uma notação baseada em linguagem natural.

Um algoritmo é composto por um nome, as variáveis e o bloco principal

{Este algoritmo apresenta a estrutura básica presente em todos os algoritmos criados}

**Algoritmo** AlgoritmoGenerico → nome do algoritmo

**Declare**

variavel1, variavel2: **tipo1**;  
variavel3, variavel4: **tipo2**;

} variáveis

**Início**

instrução1;  
instrução2;

**Fim**

} bloco principal

# Paradigmas de Linguagem de Programação

- Fornece e determina a visão que o programador possui sobre a estruturação e execução do programa.
- Assim como diferentes grupos em engenharia de software propõem diferentes metodologias, diferentes linguagens de programação propõem diferentes paradigmas de programação.

# Paradigmas de Linguagem de Programação

- Um paradigma específico:

- Java- Orientações a objetos
- Haskell- Funcional

- Múltiplos paradigmas:

- LISP, Perl, Python, C++, Oz

# Paradigmas de Linguagem de Programação

- Os paradigmas de programação são muitas vezes diferenciados pelas técnicas de programação que proíbem ou permitem.
- O relacionamento entre paradigmas de programação e linguagens de programação pode ser complexo.

# Paradigmas de Linguagem de Programação

- Se dividem em dois grandes grupos:
  - Imperativo
  - Declarativo

# Paradigmas de Linguagem de Programação

- Os paradigmas imperativos são aqueles que facilitam a computação por meio de mudanças de estado. Se dividem principalmente em:
  - O paradigma estruturado: Algol 60.
  - O paradigma procedural: Fortran e o Basic.
  - O paradigma de orientação a objetos: C++, Java, Python e Ruby

# Paradigmas de Linguagem de Programação

- A arquitetura dos computadores exerceu um efeito crucial sobre o projeto das linguagens de programação.
- Arquitetura de Von Neumann se caracteriza pela possibilidade de uma máquina digital armazenar seus programas no mesmo espaço de memória que os dados, podendo assim manipular tais programas.

# Paradigmas de Linguagem de Programação

- O modelo imperativo de programação baseia-se no modo de funcionamento do computador.
- O paradigma imperativo é predominante nas LPs, pois tais linguagens são mais fáceis de traduzir para uma forma adequada para execução na máquina.



# Paradigmas de Linguagem de Programação

- Linguagens imperativas são caracterizadas por três conceitos: variáveis, atribuições e sequências.
- O estado de um programa imperativo é mantido em variáveis de programa que são associados com localizações de memória que correspondem a um endereço e um valor de armazenamento.
- O valor da variável pode ser acessado direta e indiretamente, e pode ser alterado através de um comando de atribuição. O comando de atribuição introduz uma dependência de ordem no programa: o valor de uma variável é diferente antes e depois de um comando de atribuição.

# Paradigmas de Linguagem de Programação

- Já as funções de linguagem de programação imperativas são descritas como algoritmos que especificam como processar um intervalo de valores, a partir de um valor de domínio, com uma série de passos prescritos.
- A repetição, ou laço, é usada extensivamente para processar os valores desejados. Laços são usados para varrer uma sequência de localizações de memória tal como vetores, ou para acumular um valor em uma variável específica.

# Paradigmas de Linguagem de Programação

- As características centrais das linguagens imperativas são:
  - As variáveis, que modelam as células de memória;
  - Comandos de atribuição, que são baseados nas operações de transferência dos dados e instruções;
  - A execução sequencial de instruções;
  - A forma iterativa de repetição, que é o método mais eficiente desta arquitetura.

# Paradigmas de Linguagem de Programação

- Vantagens

- Eficiência (embute o modelo de Von Neumann)
- Mais fácil de traduzir para a linguagem de máquina
- Paradigma dominante e bem estabelecido
- Modelagem “Natural” de aplicações do mundo real
- Também é muito flexível

- Desvantagens

- Descrições demasiadamente operacionais
- Focaliza o “como” e não o “quê”

# Paradigmas de Linguagem de Programação

- Alguns exemplos de linguagens de programação que baseiam-se no modelo imperativo:
  - Ada
  - Algol
  - Basic
  - C/C++
  - PHP
  - Java
  - Cobol
  - Fortran
  - Pascal
  - Python
  - Lua
  - Mathematica

# Programação Declarativa

- Já a Programação declarativa é um paradigma de programação baseado em programação funcional, programação lógica ou programação restritiva. Um programa se diz declarativo quando descreve o que ele faz e não como seus procedimentos funcionam. Na programação declarativa, não há a ideia de estado do programa, como ocorre na programação imperativa.

# Programação Declarativa

- Vantagens:

- Reduzir efeito colaterais
- Minimizar mutabilidade
- Mais legibilidade e menos bugs

- Desvantagens:

- Não oferece alocação explícita de memória ou declaração explícita de variáveis, necessários para a resolução de muitos problemas reais.