

1 Introdução

Neste material iremos estudar sobre lógica de programação, algoritmos e suas possíveis representações e o desenvolvimento de programas de computador utilizando linguagens de programação.

2 Definições

2.1 Lógica: Raciocínio realizado de acordo com princípios muito bem definidos e considerados válidos, verdadeiros.

2.2 Algoritmo: Sequência de passos finita construída com o objetivo de solucionar um problema.

2.3 Linguagem de Programação: Ferramenta que permite que um algoritmo seja expresso de modo que o computador seja capaz de entendê-lo e executá-lo.

3 Exemplos de algoritmos

No dia a dia, nós desenvolvemos e aplicamos muitos algoritmos com o objetivo de solucionar problemas comuns. Veja alguns exemplos.

3.1 Algoritmo (sequência de passos) para fritar um ovo.

- 1 – Pegar frigideira, ovo, óleo e sal
- 2 – Colocar óleo na frigideira
- 3 – Acender o fogo
- 4 – Colocar a frigideira no fogo
- 5 – Esperar o óleo esquentar
- 6 – Colocar o ovo
- 7 – Retirar quando pronto

3.2 Algoritmo (sequência de passos) para mascar um chiclete.

- 1 – Pegar o chiclete
- 2 – Retirar o papel
- 3 – Mastigar
- 4 – Jogar o papel no lixo

3.3 Algoritmo (sequência de passos) para trocar uma lâmpada.

- 1 – Se (lâmpada estiver fora de alcance)
 Pegar a escada
- 2 – Pegar a lâmpada
- 3 – Se (lâmpada estiver quente)
 Pegar pano
- 4 – Tirar lâmpada queimada
- 5 – Colocar lâmpada boa

3.4 Algoritmo (sequência de passos) para o fim de semana.

- 1 – Ver a previsão do tempo
- 2 – Se (fizer Sol)
 Vou à praia
- 3 – Senão
 Vou estudar
- 4 – Ver televisão
- 5 – Dormir

3.5 Algoritmo (sequência de passos) para descascar batatas.

- 1 – Pegar faca, bacia e batatas
- 2 – Colocar água na bacia
- 3 – Enquanto (houver batatas)
 - descascar próxima batata
 - colocar batata descascada na água

3.6 Algoritmo (sequência de passos) para fazer uma prova.

- 1 – Ler a prova
- 2 – Pegar a caneta
- 3 – Enquanto (houver questão em branco) e (tempo não terminou) faça
 - Se (souber a resposta para uma questão)
 - Resolvê-la
 - Senão
 - Pular para outra ainda não avaliada
- 4 – Entregar a prova
- 5 - Rezar

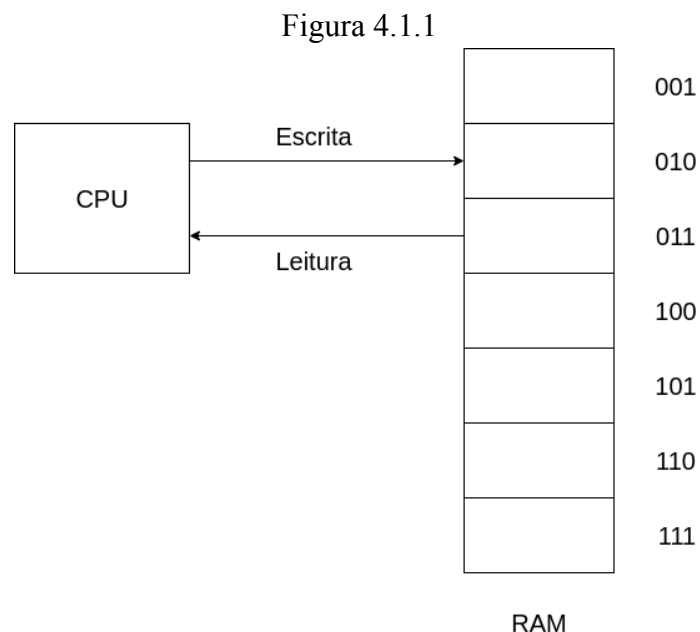
Exercícios propostos:

- 1 Escreva um algoritmo que descreva o que você pretende fazer ao longo do semestre para ser aprovado nas duas UCs.
- 2 Escreva um algoritmo que descreva o que você pretende fazer para alcançar o seu maior objetivo.

4 Conceitos Fundamentais para a programação de computadores

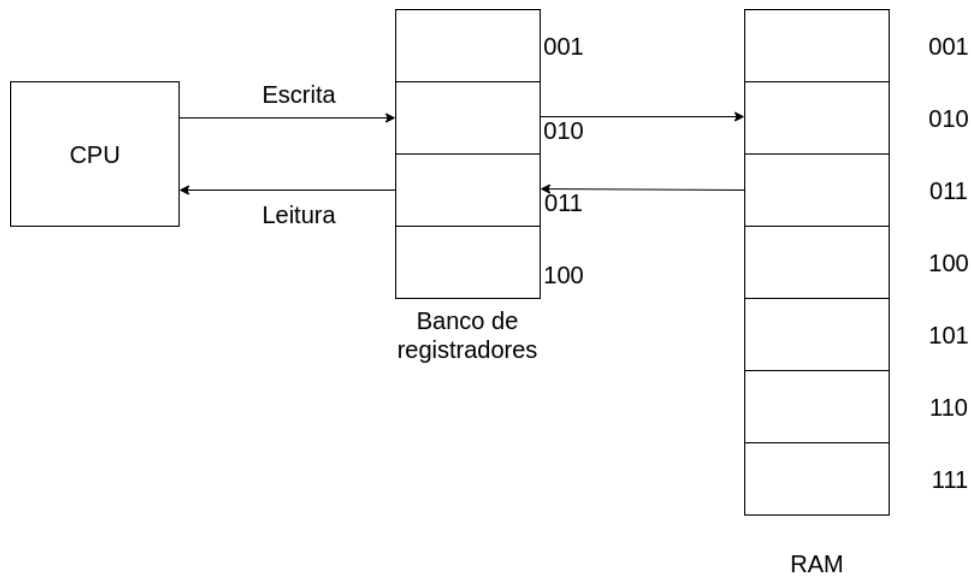
Nesta seção iremos estudar sobre o funcionamento básico de um computador, considerando os aspectos mais importantes para o desenvolvedor de software.

4.1 Programar um computador significa especificar o que ele deve fazer. Para isso, precisamos conhecer as partes mais importantes dele, que são a memória e a CPU. Além disso, precisamos conhecer como ambas interagem, que é por meio de operações comumente chamadas de **leitura** e **escrita**. Veja a Figura 4.1.1. Note que a memória é dividida em pequenos blocos, cada qual com o seu endereço. Em cada um deles podemos guardar um dado de interesse a fim de viabilizar a solução computacional de algum problema.



4.2 (O banco de registradores) A Figura 4.1.1 ilustra os componentes fundamentais de um computador. Porém há um detalhe muito importante a ser citado. A velocidade de processamento da CPU, nos dias atuais, é muito maior do que a velocidade com que a memória é capaz de disponibilizar os dados para que o processamento ocorra. Por exemplo, para calcular 10% de aumento sobre o salário de uma pessoa que está armazenado na memória, a CPU precisa fazer uma operação de leitura nela. Por ser muito mais rápida do que a memória, a tendência é que a CPU passe algum tempo ociosa aguardando até que o salário esteja disponível para ela poder processar, o que é um desperdício computacional. Para resolver esse problema, os computadores empregam um recurso conhecido como **hierarquia de memória**. O banco de registradores faz parte dele. Trata-se de um bloco de memória parecido com a memória RAM, porém menor e muito mais rápido. Ela fica entre a CPU e a memória. Veja a Figura 4.2.1.

Figura 4.2.1



Como a Figura 4.2.1 ilustra, a fim de realizar alguma computação, é preciso transportar dados da memória RAM para o banco de registradores para que só então a CPU possa manipulá-los.

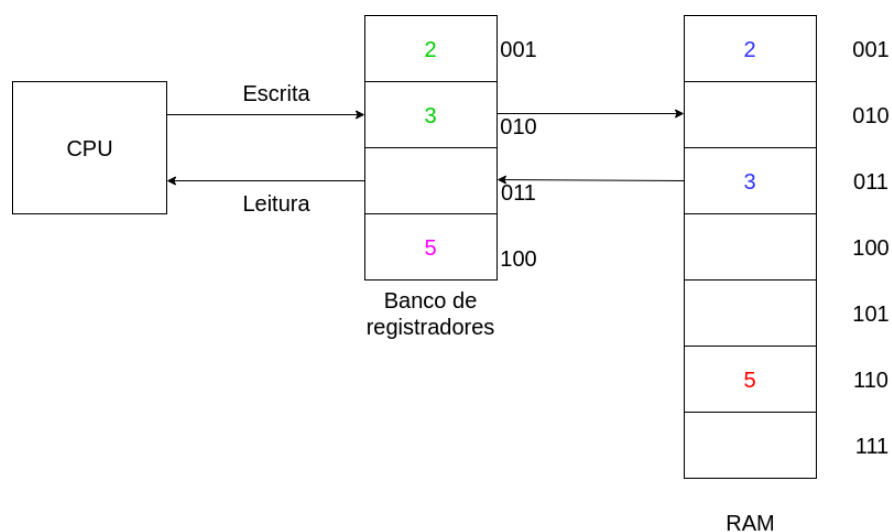
4.3 (Conversando com a CPU: somando dois números com linguagem de máquina) Para programar um computador utilizamos uma linguagem de programação. Uma ferramenta que nos permite especificar à CPU o que ela deve fazer. A linguagem que a CPU entende nativamente é composta somente de 0s e 1s. Essa é a conhecida **linguagem de máquina**. A fim de ilustrar o uso de uma linguagem de máquina, vamos escrever um programa que realiza a soma de dois valores existentes na memória, armazenando o resultado nela no final. Para tal, precisamos de comandos para as operações descritas na Tabela 4.3.1.

Tabela 4.3.1

Comando	Finalidade	Como utilizar?
load	Transportar um dado da memória para o banco de registradores	1001
store	Transportar um dado do banco de registradores para a memória	1100
add	Somar dois valores	1111

Na Tabela 4.3.1 especificamos três comando para a nossa CPU fictícia. CPUs reais também possuem uma tabela dessa, bem mais complexa e com mais operações. Mas a ideia é exatamente a mesma. A Figura 4.3.1 mostra o que desejamos fazer.

Figura 4.3.1



De acordo com a linguagem que especificamos para a nossa CPU, o programa que realiza essas operações é exibido na Listagem 4.3.1.

Listagem 4.3.1

```
1001 001 001
1001 011 010
1111 001 010 100
1100 100 110
```

4.4 (Conversando com a CPU: somando dois números com linguagem de montagem) Perceba como a programação de computadores utilizando linguagem de máquina pode ser difícil e propensa a erros. Por essa razão, as CPUs oferecem um mecanismo conhecido como **montador** ou **assembler**. Isso permite que o programador especifique seu programa sem dizer diretamente os números que representam as operações que deseja. Como exemplo, vamos usar a linguagem de montagem especificada na Tabela 4.4.1. Note que as operações agora têm um apelido.

Tabela 4.4.1

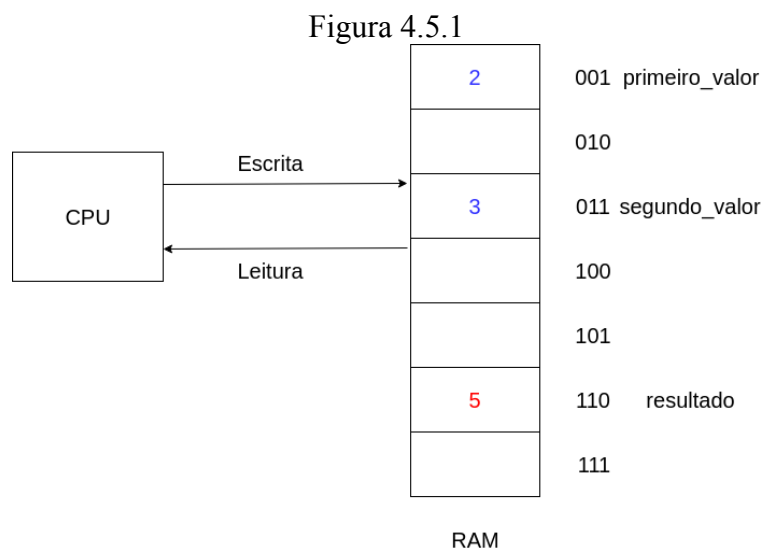
Comando	Finalidade	Como utilizar?
load	Transportar um dado da memória para o banco de registradores	lw
store	Transportar um dado do banco de registradores para a memória	sw
add	Somar dois valores	add

Utilizando essa linguagem de montagem, nosso programa agora fica como mostra a Listagem 4.4.1.

Listagem 4.4.1

```
lw 001 001
lw 011 010
add 001 010 100
sw 100 110
```

4.5 (Conversando com a CPU: somando dois números com uma linguagem de alto nível) Embora a linguagem de montagem tenha simplificado a programação até certo ponto, quando a utiliza, o programador ainda tem de se preocupar com detalhes muito próximos do hardware. Por exemplo, ele tem de saber da existência do banco de registradores e fazer uso apropriado dele. Além disso, é muito inconveniente, difícil e propenso a erros ter de lembrar os endereços em que está armazenado cada um dos valores de interesse. Por essa razão, é comum o uso de **linguagens de programação de alto nível**. Elas têm como principal característica permitir ao programador desenvolver seus programas sem se preocupar com muitos detalhes de hardware. Dizemos que elas são mais próximas do ser humano e mais distantes dos detalhes de funcionamento do computador. Porém, é necessário que seja feita uma espécie de **tradução** do programa escrito em alto nível para uma linguagem que o computador entenda. Em geral, esse processo é feito por um componente de software chamado **compilador**. Um dos recursos mais importantes existentes em uma linguagem de alto nível é a **declaração de variáveis**. Trata-se de um recurso que permite que sejam atribuídos nomes ou apelidos a posições de memória de modo que não seja necessário lembrar de seus endereços. Veja como ficaria nosso programa de soma de dois números usando uma possível linguagem de programação de alto nível na Listagem 4.5.1, de acordo com a organização da Figura 4.5.1. Ele usa uma operação de atribuição.



Listagem 4.5.1

```
resultado = primeiro_valor + segundo_valor
```

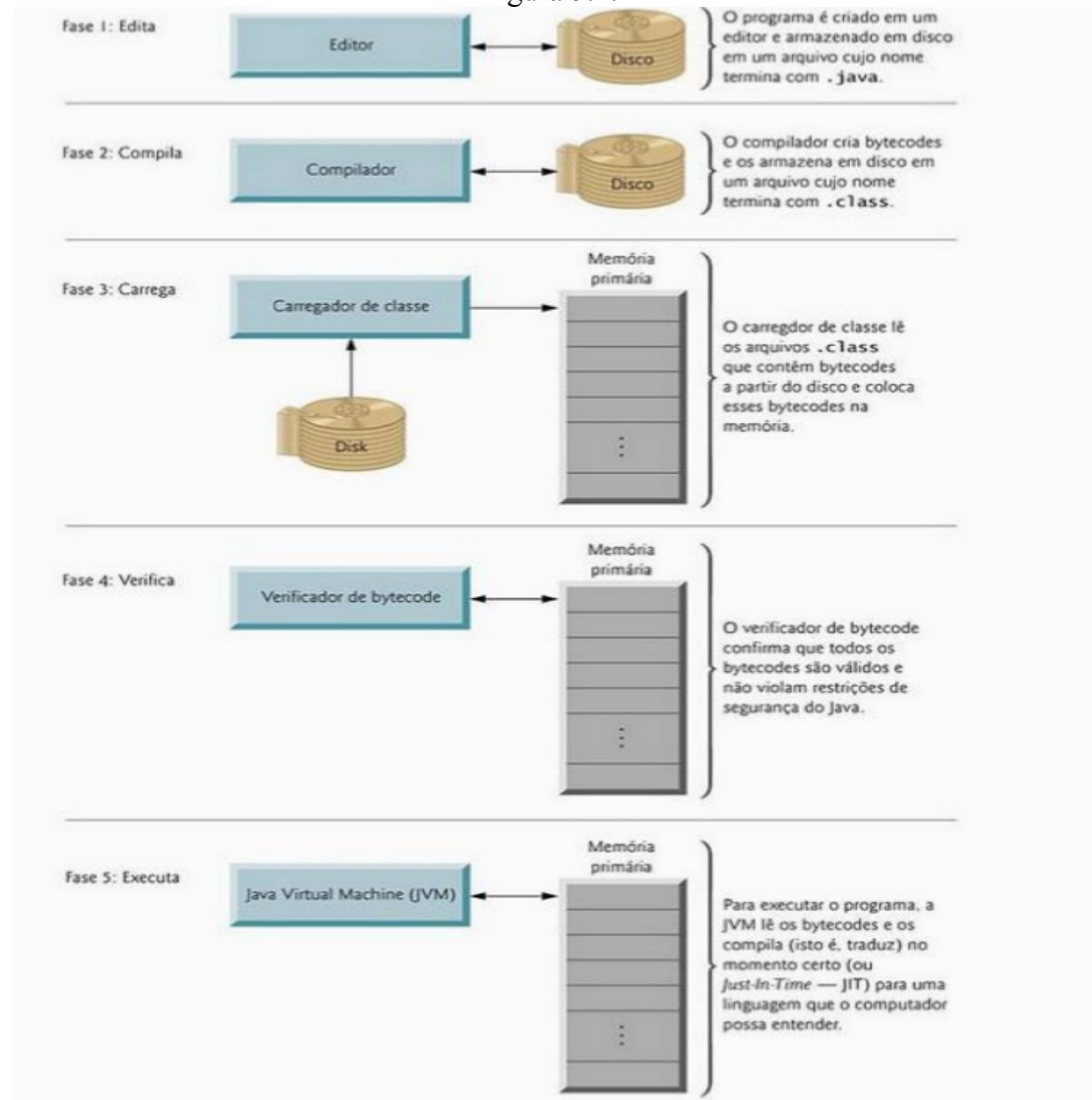
Assim, concluímos nosso estudo inicial sobre as características principais de funcionamento de um computador.

5 Programação com Java

A linguagem de programação que utilizaremos neste curso chama-se **Java**. Trata-se de uma linguagem de programação de alto nível com as características vistas na seção anterior. Há alguns conceitos fundamentais que todo desenvolvedor deve conhecer, os quais estudaremos nesta seção.

5.1 O ambiente de desenvolvimento típico Java é ilustrado na Figura 5.1.1.

Figura 5.1.1



Assim, concluímos que Java é uma linguagem **compilada** e **interpretada**.

5.2 As principais siglas encontradas em um ambiente de desenvolvimento Java são as seguintes.

JDK – Java Development Kit – Contém as ferramentas necessárias para o desenvolvimento e execução de programas em Java.

JVM – Java Virtual Machine – Responsável por interpretar programas Java compilados.

JRE – Java Runtime Environment – Inclui a JVM, bibliotecas (código escrito previamente e reutilizável) entre outras coisas que viabilizam a execução de aplicativos Java.

5.3 Java não é somente uma linguagem de programação. Dizemos que Java é uma **Plataforma**, composta pela linguagem, a API e a máquina virtual. Java possui uma especificação e muitas possíveis implementações como a oficial mantida pela Oracle e o OpenJDK.

5.4 Java possui um lema conhecido como “Write Once, Run Anywhere”, o que a caracteriza como uma linguagem **portável**.

5.5 Hoje, Java é usado, por exemplo, nos seguintes cenários:

- Desenvolvimento de aplicações WEB
- Computação em Nuvem
- Computação para Dispositivos Móveis
- Java Embedded (Raspberry PI)
- Eletrônicos em Geral (TV, máquina de lavar etc.)
- Veículos

5.6 Veja no link 5.6.1 um ranking de popularidade de linguagens de programação.

Link 5.6.1

<https://www.tiobe.com/tiobe-index/>

5.7 Os links a seguir são fundamentais para todo desenvolvedor Java.

- Download do JDK (Você precisa desse para programar em Java)
<https://www.oracle.com/java/technologies/javase-jdk13-downloads.html>

- Especificação da linguagem e plataforma

<https://docs.oracle.com/javase/specs/>

- The Java Tutorials

<https://docs.oracle.com/javase/tutorial/>

- Javadoc

<https://docs.oracle.com/en/java/javase/13/docs/api/index.html>

5.8 Tradicionalmente, quando aprendemos uma nova tecnologia para o desenvolvimento de software, começamos o aprendizado por meio do desenvolvimento de uma aplicação “Hello, World”. Trata-se de uma aplicação que entra em execução e exibe o texto “Hello, World”. Junto com o seu professor, abra o IDE escolhido e desenvolva um programa “Hello, World” em Java. Faça variações que exibem a mensagem tanto no terminal quanto em uma interface gráfica.

Referências

DEITEL, P. e DEITEL, H. **Java Como Programar**. 8ª Edição. São Paulo, SP: Pearson, 2010.

LOPES, A. e GARCIA, G. **Introdução à Programação – 500 Algoritmos Resolvidos**. 1ª Edição. São Paulo, SP: Elsevier, 2002.