

# Apostila de Java

MARIANA CAVALLE

30 de dezembro de 2023

# Sumário

|                |                                     |          |
|----------------|-------------------------------------|----------|
| <b>Sumário</b> | <b>2</b>                            |          |
| <b>1</b>       | <b>Introdução</b>                   | <b>3</b> |
| <b>2</b>       | <b>Um pouco da história do Java</b> | <b>3</b> |
| 2.1            | Origem do nome Java                 | 3        |
| 2.2            | A tecnologia                        | 3        |
| 2.3            | Mas por quê o Java?                 | 3        |
| <b>3</b>       | <b>Ambiente de desenvolvimento</b>  | <b>4</b> |
| 3.1            | Preparando o ambiente               | 4        |
| 3.2            | Qual IDE utilizar?                  | 4        |
| 3.3            | Primeiro programa em Java           | 4        |

## 1 Introdução

Essa apostila tem como intuito auxiliar iniciantes em Java que, assim como eu, já estiveram perdidos nesse mundo da programação sem saber por onde iniciar. Aqui, pretendo demonstrar conceitos teóricos e exemplificações práticas da linguagem. Como funciona, como exercitar e onde aprender mais.

Se alguém algum dia ler isso, espero que possa ajudar de alguma forma, enquanto também me ajuda a aprofundar meus conhecimentos. Aceito críticas, dúvidas e sugestões para melhorar isso aqui.

## 2 Um pouco da história do Java

### 2.1 Origem do nome Java

Antes de mais nada, é importante falar um pouquinho sobre como a linguagem surgiu. Tudo começou em 1991, na Sun Microsystems que criou a linguagem chamada "Oak" (que significa carvalho, em inglês). O nome foi dado em homenagem à uma árvore de carvalho que tinha perto do escritório e que visível por James Gosling, considerado o pai do Java.

No entanto, não foi possível registrar a linguagem com esse nome, pois já existia uma chamada Oak. Para decidir o novo nome, a equipe foi consultada e decidiram por Java. A ideia do nome é referência ao café de mesmo nome que é originário da ilha de Java. Por isso também a logo do nosso querido Java é uma xícara de café.

### 2.2 A tecnologia

Originalmente, a linguagem Oak foi desenvolvida para ser executada em diferentes aparelhos eletrônicos, permitindo que um só código rodasse em diferentes plataformas, como em televisões, geladeiras, entre outros. Mas essa ideia ainda era muito a frente do seu tempo e acabou não sendo prioridade da Sun Microsystems. Mas em 1994 com o avanço da internet, eles viram uma oportunidade de revisitar a linguagem e adaptá-la para o que fosse uma linguagem que rodasse em mais de uma plataforma e permitisse integração com a Web. Oficialmente, em 1995 a linguagem foi renomeada de Oak para Java e permitia a integração dos conteúdos Web de forma interativa.

### 2.3 Mas por quê o Java?

O Java foi a primeira linguagem a permitir que você rodasse o mesmo código em plataformas diferentes. Isso porque é uma linguagem tanto compilada quanto interpretada. Mas o que isso quer dizer?

Bom, quando eu escrevo um código em Java, eu o compilo (utilizando o `javac`). Esse processo de compilação vai gerar um "bytecode", que é um arquivo com a extensão `.class` na minha máquina para ser executada.

E é aí que entra a parte do "interpretada" na história, pois esse arquivo vai ser lido pela minha Java Virtual Machine (JVM). Ela é a responsável por interpretar todos os bytecodes que gerei anteriormente no meu arquivo `.class` e rodar os comandos em meu computador.

Isso foi algo revolucionário, principalmente porque agora eu poderia escrever um código só e rodar em qualquer Sistema Operacional. O mesmo código que eu escrevo no meu Windows pode ser executado no Linux ou em um MacOS, por exemplo, tendo em vista que os bytecodes gerados podem ser lidos por qualquer sistema operacional desde que ele tenha a JVM instalada. Mas óbvio que essa não é a única vantagem do Java. Além disso ele é orientado a objetos, muito robusto e permite a criação tanto de aplicativos web como aplicações

de desktop, programação de robôs, celulares, entre muitas outras coisas.  
Resumindo, Javinha é bom demais e serve pra tanta coisa. Vale a pena aprender.

### 3 Ambiente de desenvolvimento

#### 3.1 Preparando o ambiente

Para começar a programar em Java, primeiro precisamos instalar o Java Development Kit (JDK). O download pode ser feito pelo site da Oracle, por esse link: <https://www.oracle.com/br/java/technologies/downloads/>. Depois de realizar o processo de instalação, para confirmar que está tudo certo, basta abrir o prompt de comando e digitar "java -version". Se aparecer a versão instalada, está tudo pronto:

```
C:\Users\miojo>java -version
openjdk version "17.0.6" 2023-01-17
OpenJDK Runtime Environment Temurin-17.0.6+10 (build 17.0.6+10)
OpenJDK 64-Bit Server VM Temurin-17.0.6+10 (build 17.0.6+10, mixed mode, sharing)
```

#### 3.2 Qual IDE utilizar?

Há diversas IDEs que podem ser utilizadas no desenvolvimento do Java, sendo algumas das mais famosas:

- NetBeans
- Eclipse
- IntelliJ
- VSCode

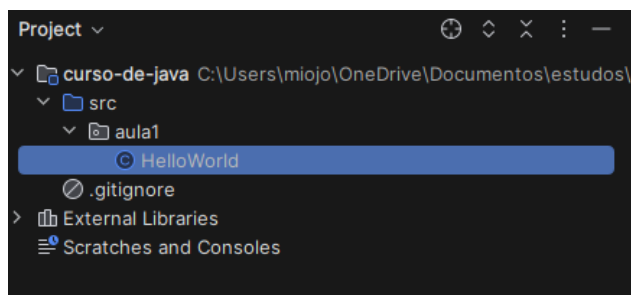
Ainda tem alguns que se arriscam a usar o Neovim, pra deixar tudo mais personalizado.

Aqui no meu caso, vou usar o IntelliJ para os exercícios e exemplos, porque foi a que mais me adaptei. Mas você pode testar e usar a que preferir.

#### 3.3 Primeiro programa em Java

Lógico que o primeiro programa tem que ser o clássico "Hello, World!". Dizem que se o seu primeiro programa em qualquer linguagem não for o famoso hello world, terá 7 anos de azar.

Dentro da IDE de sua escolha, crie um novo projeto e dentro da pasta "src" crie uma nova classe chamada "HelloWorld", conforme o exemplo abaixo:



As Classes em Java são onde você irá escrever o programa e os nomes normalmente se iniciam em letra maiúscula, seguindo o padrão CamelCase. Portanto, sempre que eu tenho uma nova palavra, ela se inicia com letra maiúscula. Para esse programa inicial, iremos escrever esse código:

---

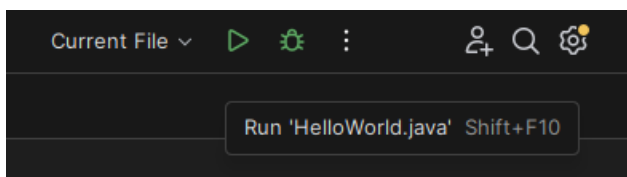
```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, World!");  
    }  
}
```

---

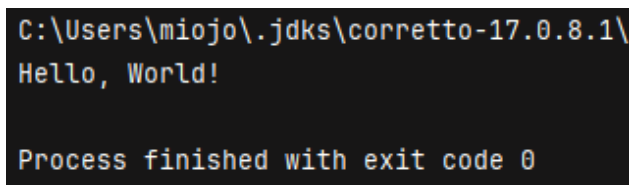
O método "public static void main(String[] args)" indica para o Java que aquele é o programa executável, é onde eu digo para o programa que aquelas são as linhas que devem ser seguidas quando iniciar o programa. Acredito que mais pra frente os métodos e classes ficarão mais claros.

Já para imprimir a informação na tela, utilizamos o "System.out.println();

Simples, né? Agora já posso rodar em minha IDE. No caso do IntelliJ, basta apertar "Shift + 10" ou clicar nessa setinha verde no topo, com a opção current file selecionada:



Assim que compilar, a frase será exibida no terminal. Prontinho, temos nosso primeiro programa em Java!



```
C:\Users\miojo\.jdk\corretto-17.0.8.1\bin  
Hello, World!  
  
Process finished with exit code 0
```