

# Java Básico

*Daniel Pereira*

## 1 – Introdução ao Java

Nos primórdios da computação, os computadores eram operados de forma muito rudimentar, utilizando plugs e botões em vez de linguagens de programação. O ENIAC, um dos primeiros computadores eletrônicos, usava a base decimal e era bastante complexo, com um grande número de válvulas.

Com o tempo, à medida que os desktops se tornaram mais comuns, surgiram linguagens de programação de alto nível, como Pascal, facilitando a programação e o desenvolvimento de software. Uma linguagem de alto nível é mais próxima da linguagem humana, facilitando a programação. Ela abstrai os detalhes do hardware e é convertida em código de máquina por um compilador ou interpretador, permitindo que os programadores se concentrem na lógica e funcionalidade do software. No entanto, a diversidade de compiladores para diferentes máquinas dificultava a compatibilidade e portabilidade dos programas.

Em 1990, a Sun Microsystems, sob a liderança de James Gosling, começou a desenvolver o Greentalk,, que mais tarde se tornou o Java. Lançado em 1995, o Java se destacou pela sua portabilidade, com o slogan "escreva uma vez, execute em qualquer lugar", e rapidamente se tornou uma das linguagens de programação mais populares, utilizada em uma ampla variedade de aplicações

O Java se tornou um projeto de código aberto em 2006, quando a Sun Microsystems lançou seu código sob a Licença Pública Geral GNU (GPL), permitindo contribuições dos desenvolvedores. Em 2010, a Oracle adquiriu a Sun e continuou a evolução da linguagem, com atualizações regulares. O Java se consolidou como uma das principais linguagens de programação, expandindo-se para plataformas como dispositivos móveis (Java ME) e aplicações corporativas (Java EE), e tornando-se essencial em ambientes de servidor e no desenvolvimento de aplicações web, mantendo sua popularidade ao longo dos anos.

## 2 – Funcionamento do Java

No Java, o funcionamento basicamente consiste em:

1. **Código:** O programador escreve o código-fonte em arquivos com extensão `.java`.
2. **Compilação:** O código-fonte é compilado por um compilador Java (**javac**), que o transforma em bytecode, um formato intermediário que é armazenado em arquivos `.class`.
3. **Execução:** O bytecode é executado na Java Virtual Machine (**JVM**). A JVM interpreta ou compila o bytecode em código de máquina que o sistema operacional pode entender.

Esse processo permite que o código Java seja executado em qualquer dispositivo que tenha uma JVM, tornando-o altamente portátil. Isso significa que um programa escrito em Java funcionará em diferentes sistemas operacionais, como Windows, macOS ou Linux, sem precisar de alterações no código.

No Java, os usuários são separados em dois grupos:

**JRE** (Java Runtime Environment): É voltado para usuários finais que apenas desejam executar aplicativos Java. Ele inclui a Java Virtual Machine (JVM) e as bibliotecas necessárias para a execução dos programas, mas não oferece ferramentas para desenvolvimento.

**JDK** (Java Development Kit): É destinado a desenvolvedores e inclui tudo que o JRE oferece, além de ferramentas adicionais, como compiladores, depuradores e outras utilidades que permitem escrever, compilar e testar aplicativos Java.

Portanto, **JRE** é usado para executar aplicações, enquanto **JDK** é necessário para o desenvolvimento.

### IDE e NetBeans:

Existem ferramentas que facilitam o desenvolvimento de software, e uma delas é a **IDE**.

Uma **IDE** (Integrated Development Environment) é uma ferramenta abrangente que combina editor de código, depurador e compilador para facilitar o desenvolvimento de software.

O **NetBeans** é uma IDE de código aberto distribuída oficialmente pela Oracle, amplamente utilizada para desenvolvimento em Java, mas que também suporta outras linguagens, como PHP, HTML5 e JavaScript, por meio de plugins. Oferece recursos como edição de código com completamento automático, refatoração, depuração, gerenciamento de projetos e integração com sistemas de controle de versão (como Git). Sua interface amigável e suporte para aplicações desktop, web e móveis, aliados a uma comunidade ativa, fazem do NetBeans uma opção robusta para desenvolvedores. Por essa razão, será utilizada neste curso.

### 3 – Downloads e Instalações Necessárias

01 - Começando pelo JDK, a versão atual (setembro/2024) é a JDK Development Kit 23

<https://www.oracle.com/br/java/technologies/downloads>

Windows x64 Installer

[https://download.oracle.com/java/23/latest/jdk-23\\_windows-x64\\_bin.exe](https://download.oracle.com/java/23/latest/jdk-23_windows-x64_bin.exe)

Importante instalar a JDK antes do Netbeans! É recomendado instalar tudo padrão sem mudar o caminho da instalação.

02 - Em seguida o Netbeans, a versão atual (setembro/2024) é a NetBeans 23

<https://netbeans.apache.org/front/main/download/nb23>

Windows x64 Installer

<https://www.apache.org/dyn/closer.lua/netbeans/netbeans-installers/23/Apache-NetBeans-23-bin-windows-x64.exe>

É recomendado instalar tudo padrão sem mudar o caminho da instalação.

03 - Por último, instalar o JavaFX Scene Builder. Será necessário uma conta Oracle para fazer o download

[www.oracle.com/java/technologies/javafxscenebuilder-1x-archive-downloads.html](http://www.oracle.com/java/technologies/javafxscenebuilder-1x-archive-downloads.html)

Instalar tudo por padrão. Importante ser instalado depois do NetBeans!

04 - Caso o JavaFX não funcione ou tenha algum problema relacionado com o JavaFX FXML, recomenda-se instalar o JDK8

<https://www.oracle.com/br/java/technologies/javase/javase8-archive-downloads.html>

05 - Caso tenha algum problema para executar os arquivos .jar com JavaFX, talvez seja necessário atualizar a JRE.