

Unidade 01 | Capítulo 01



Introdução ao Java

Prof.: Allberson Dantas





Sumário

- História do Java;
- Características da Linguagem Java;
- Configuração do Ambiente e Primeiro Programa;
- Tipos Primitivos, Variáveis e Operadores).

O que é o Java?

Java é uma linguagem de programação de alto nível, orientada a objetos, baseada em classes, amplamente utilizada no desenvolvimento de aplicações para diversos tipos de plataformas.

Criada por **James Gosling** e sua equipe na **Sun Microsystems** (posteriormente adquirida pela Oracle).



<https://www.techtudo.com.br/tudo-sobre/java/>

História do Java



1990

Projeto
"Green"

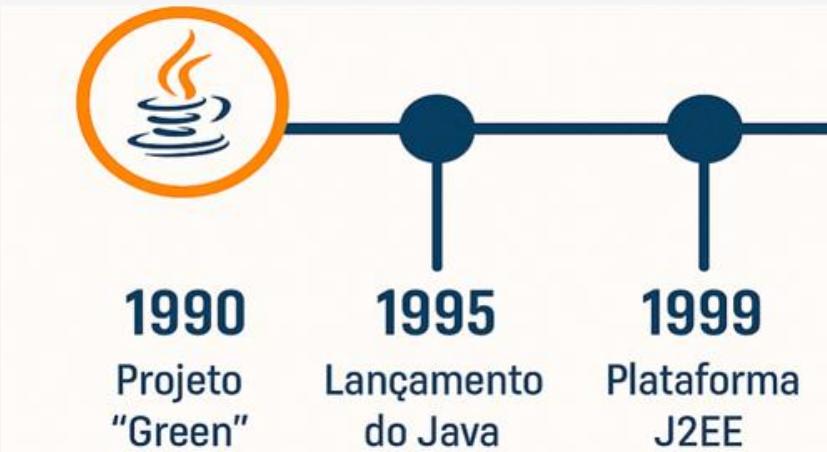
Criada por uma equipe de engenheiros liderada por James Gosling na empresa Sun Microsystems, com o objetivo inicial de desenvolver uma linguagem para pequenos dispositivos eletrônicos inteligentes.

História do Java



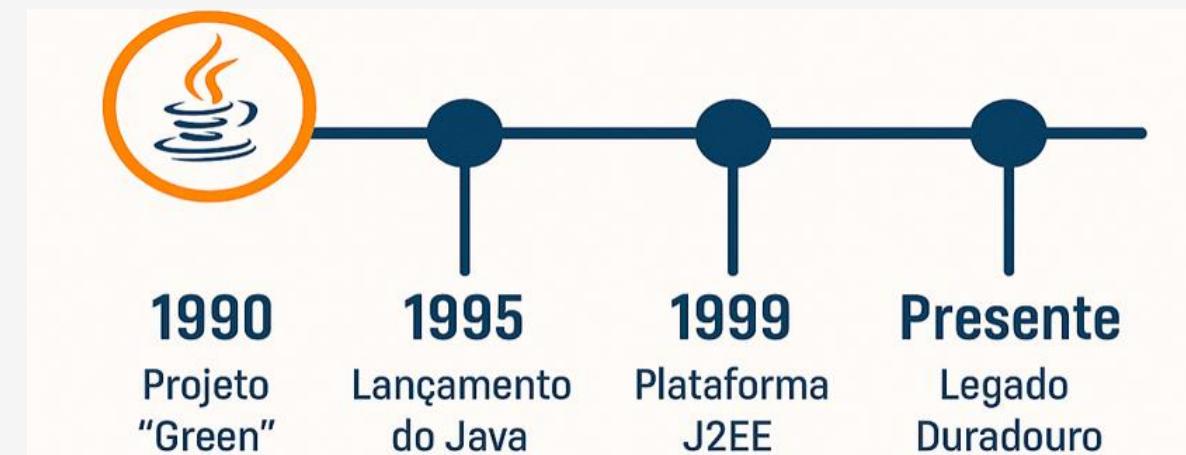
O lançamento oficial do Java aconteceu em 1995, e rapidamente a linguagem ganhou destaque entre os desenvolvedores. Um dos principais motivos foi o seu diferencial: portabilidade.

História do Java



Surgiram diferentes edições da plataforma, como a Java 2 Enterprise Edition (J2EE), voltada para aplicações corporativas de grande porte.

História do Java



Hoje, Java é amplamente utilizado em sistemas bancários, aplicações empresariais, dispositivos Android, sistemas embarcados, entre muitos outros. Sua influência é tão grande que ainda é uma das linguagens mais populares do mundo.

Características do Java

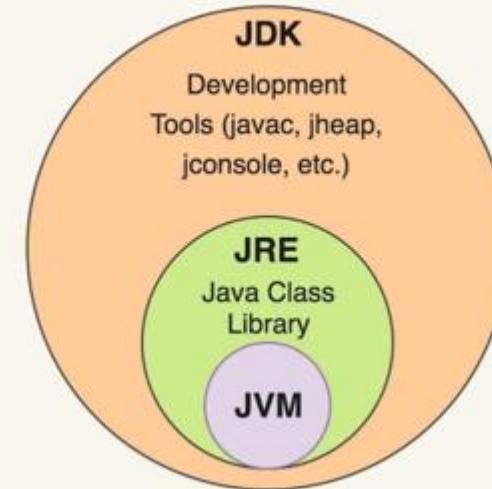


Configuração do Ambiente

Instalação do JDK (Java Development Kit)

Para programar em Java, é necessário instalar o JDK, que inclui:

- O compilador Java (javac)
- A JVM (Java Virtual Machine)
- Ferramentas auxiliares de desenvolvimento



<https://www.techtudo.com.br/tudo-sobre/java/>

Configuração do Ambiente

Instalação do JDK (Java Development Kit):

Para fazer o download do pacote JDK, acesse:

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

É importante citar que devemos dar prioridade as versões Long-Term Support(LTS), por terem suporte a longo prazo.

Configuração do Ambiente

Vamos trabalhar com a versão JDK 21. Escolha de acordo com sua máquina.

The screenshot shows the Oracle Java downloads page. At the top, there's a navigation bar with links for Produtos, Setores, Recursos, Clientes, Parceiros, Desenvolvedores, Empresa, and language options (Portuguese). Below the navigation bar, there are three main categories: Java downloads, Tools and resources, and Java archive. The Java downloads section is highlighted with a yellow background. It features a banner for "Java 24, Java 21, and earlier versions available now". Below the banner, there are descriptions for JDK 24, JDK 21 (which is the latest LTS release), and earlier JDK versions. A red arrow points to the "JDK 21" link. Below this, there's a section for "Java SE Development Kit 21.0.7 downloads" with links for Linux, macOS, and Windows. A second red arrow points to the "Windows" link.

ORACLE

Produtos Setores Recursos Clientes Parceiros Desenvolvedores Empresa

Java downloads Tools and resources Java archive

Java 24, Java 21, and earlier versions available now

JDK 24 is the latest release of the Java SE Platform.

JDK 21 is the latest *Long-Term Support (LTS)* release of the Java SE Platform.

Earlier JDK versions are available below.

[Learn about Java SE Subscription](#)

JDK 24 JDK 21 GraalVM for JDK 24 GraalVM for JDK 21

Java SE Development Kit 21.0.7 downloads

JDK 21 binaries are free to use in production and free to redistribute, at no cost, under the [Oracle No-Fee Terms and Conditions \(NFTC\)](#).

JDK 21 will receive updates under the NFTC, until September 2026, a year after the release of the next LTS. Subsequent JDK 21 updates will be licensed under the [Java SE OTN License \(OTN\)](#) and production use beyond the [limited free grants](#) of the OTN license will [require a fee](#).

Linux macOS Windows

Configuração do Ambiente

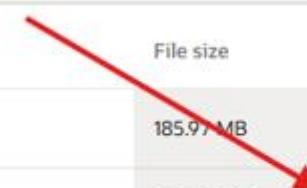
Instalação do JDK (**J**ava **D**evelopment **K**it):
Faça o download e instale o JDK.

JDK 21 binaries are free to use in production and free to redistribute, at no cost, under the Oracle Non-Free Terms and Conditions (NFTC).

JDK 21 will receive updates under the NFTC, until September 2026, a year after the release of the next LTS. Subsequent JDK 21 updates will be licensed under the Java SE OTN License (OTN) and production use beyond the [limited free grants](#) of the OTN license will require a fee.

Linux	macOS	Windows
Product/file description	File size	Download
x64 Compressed Archive	185.97 MB	https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.zip (sha256)
x64 Installer	164.35 MB	https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.exe (sha256)
x64 MSI Installer	163.09 MB	https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.msi (sha256)

[Documentation Download](#)



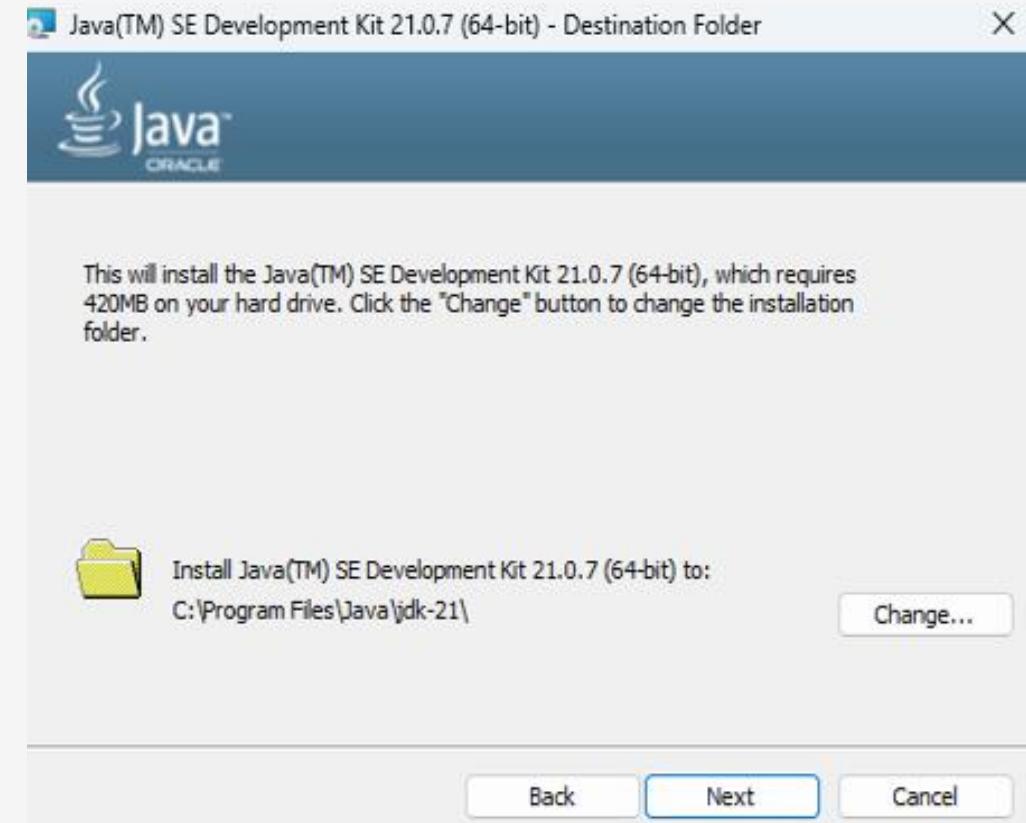
Configuração do Ambiente

Instalação do JDK (**Java Development Kit**):
Faça o download e instale o JDK.



Configuração do Ambiente

Instalação do JDK (**Java Development Kit**):
Faça o download e instale o JDK.



Configuração do Ambiente

Instalação do JDK (**Java Development Kit**):
Faça o download e instale o JDK.



Configuração do Ambiente

IDE:

Embora seja possível programar em Java usando apenas um editor de texto e o terminal, é altamente recomendável o uso de uma IDE para aumentar a produtividade. Vamos utilizar o NetBeans.

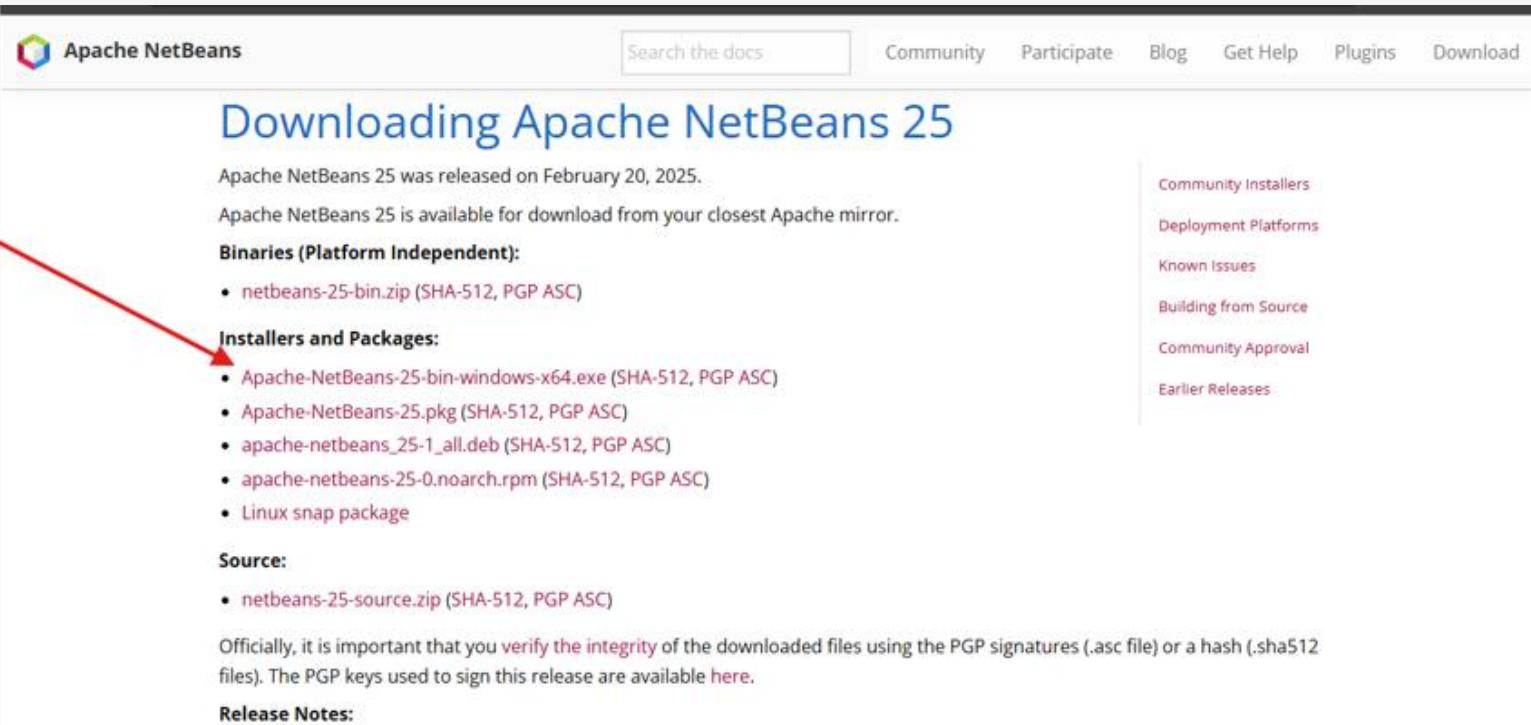
Para download basta acessar:

<https://netbeans.apache.org/front/main/index.html>



Configuração do Ambiente

Escolha o instalador de acordo com sua máquina:



The screenshot shows the Apache NetBeans 25 download page. At the top, there's a navigation bar with links for 'Community', 'Participate', 'Blog', 'Get Help', 'Plugins', and 'Download'. Below the navigation, the title 'Downloading Apache NetBeans 25' is displayed. A red arrow points from the text 'Escolha o instalador de acordo com sua máquina:' to the 'Binaries (Platform Independent)' section, which lists 'netbeans-25-bin.zip (SHA-512, PGP ASC)'. The page also lists 'Installers and Packages' (including Windows, Mac, Linux, and snap versions), 'Source' (netbeans-25-source.zip), and 'Release Notes'. On the right side, there's a sidebar with links for 'Community Installers', 'Deployment Platforms', 'Known Issues', 'Building from Source', 'Community Approval', and 'Earlier Releases'.

Apache NetBeans 25 was released on February 20, 2025.

Apache NetBeans 25 is available for download from your closest Apache mirror.

Binaries (Platform Independent):

- [netbeans-25-bin.zip \(SHA-512, PGP ASC\)](#)

Installers and Packages:

- [Apache-NetBeans-25-bin-windows-x64.exe \(SHA-512, PGP ASC\)](#)
- [Apache-NetBeans-25.pkg \(SHA-512, PGP ASC\)](#)
- [apache-netbeans_25-1_all.deb \(SHA-512, PGP ASC\)](#)
- [apache-netbeans-25-0.noarch.rpm \(SHA-512, PGP ASC\)](#)
- [Linux snap package](#)

Source:

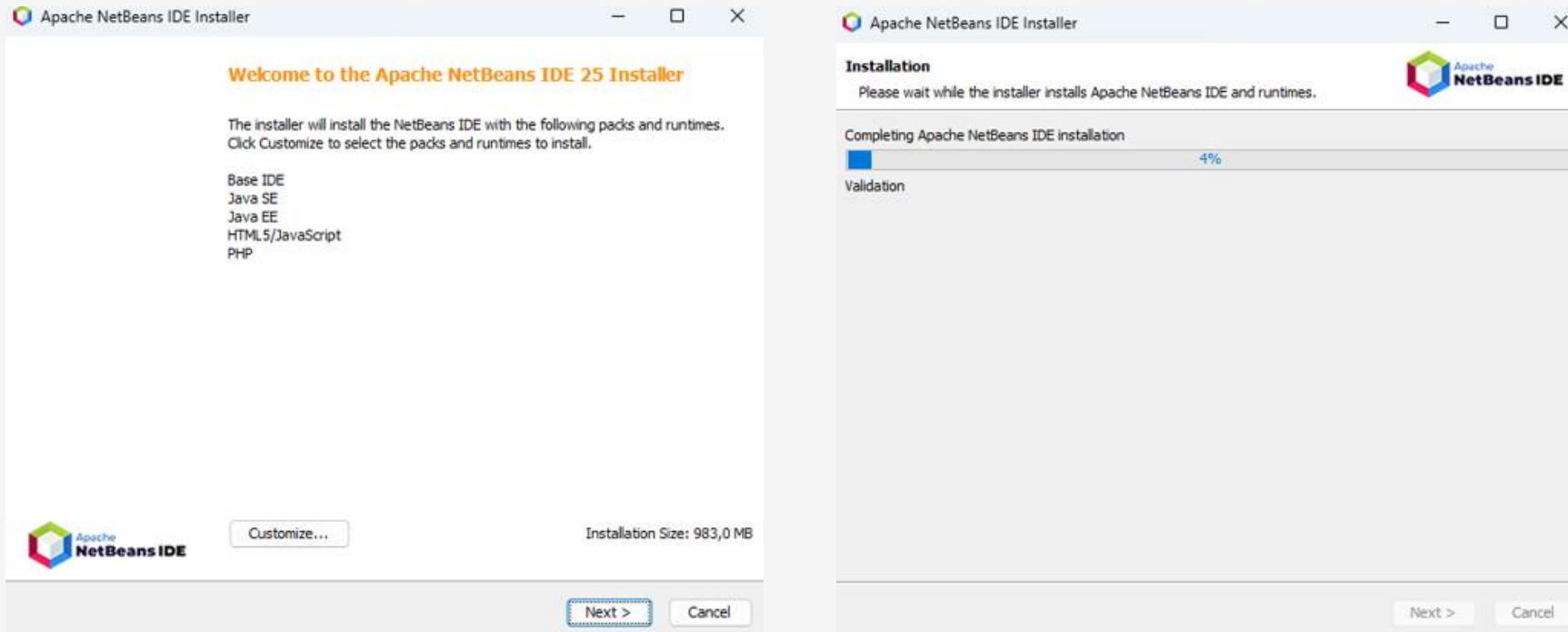
- [netbeans-25-source.zip \(SHA-512, PGP ASC\)](#)

Officially, it is important that you [verify the integrity](#) of the downloaded files using the PGP signatures (.asc file) or a hash (.sha512 files). The PGP keys used to sign this release are available [here](#).

Release Notes:

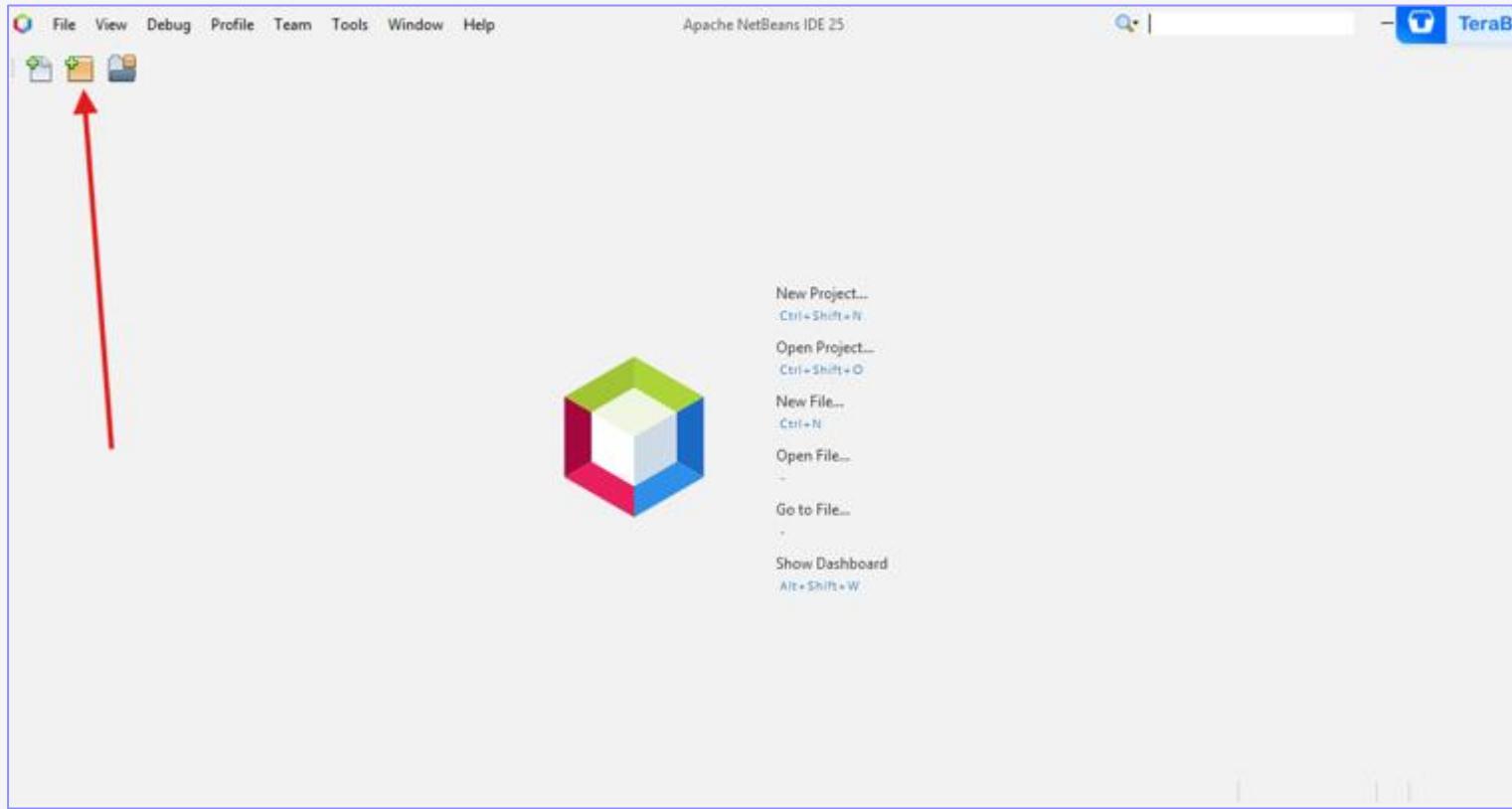
Configuração do Ambiente

Durante a instalação siga os passos, basicamente basta ir em next, aceitar os termos e seguir clicando em next:



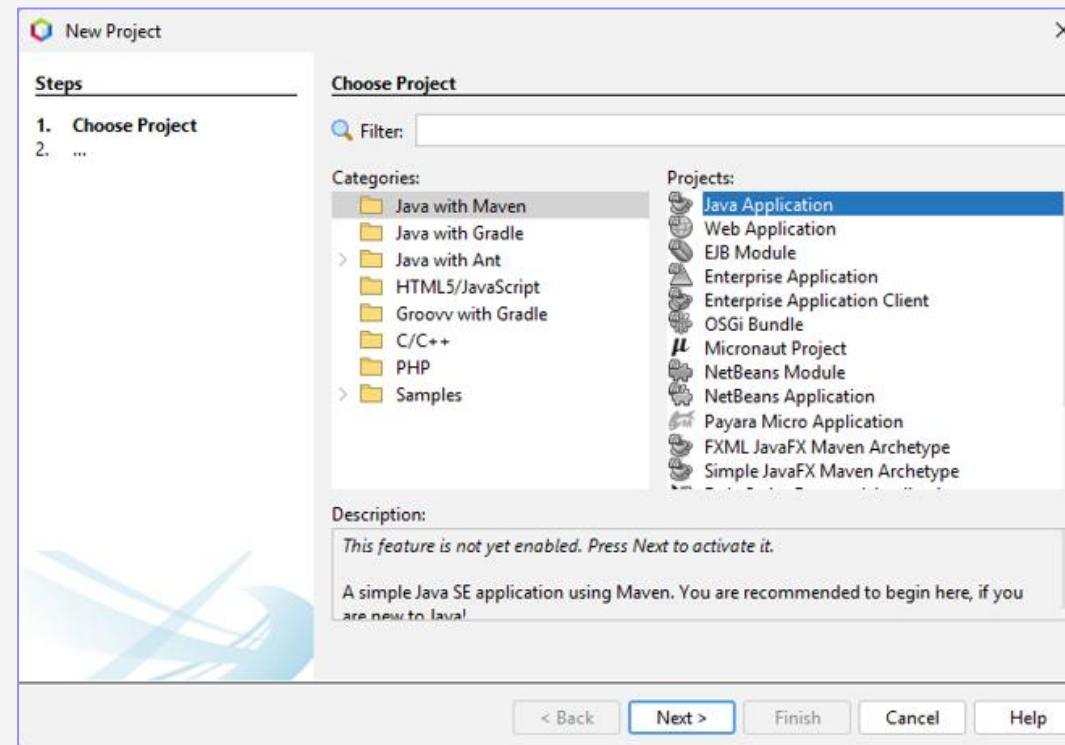
Primeiro Programa em Java

Inicialmente vamos criar um novo projeto no NetBeans



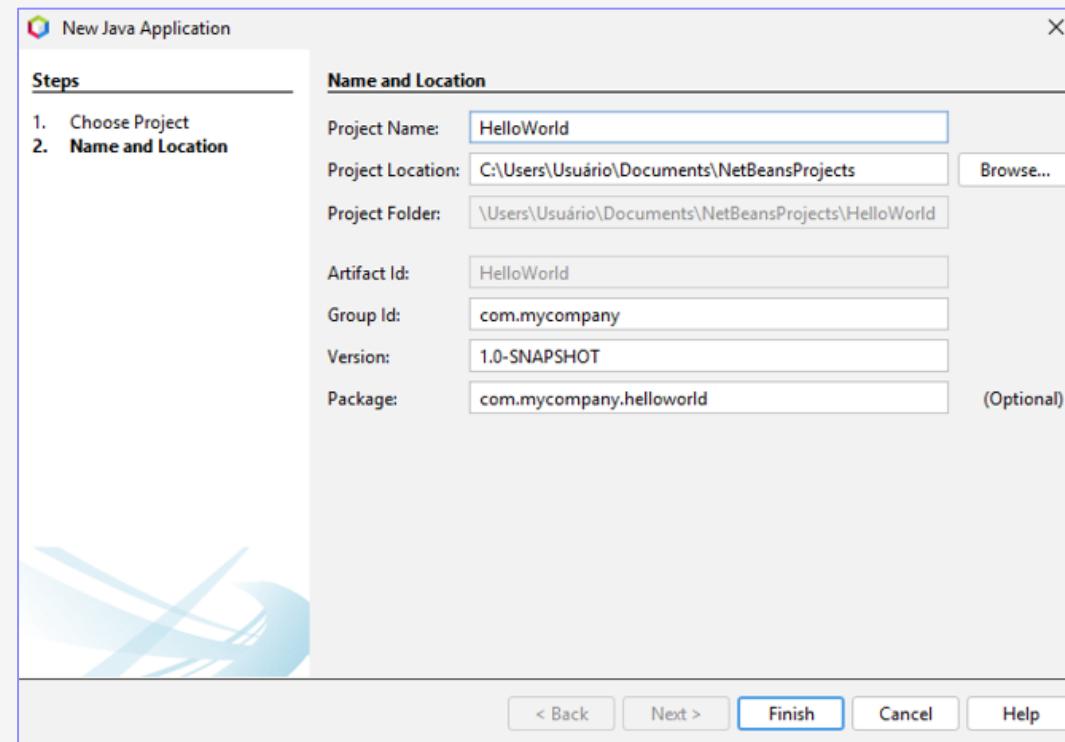
Primeiro Programa em Java

Inicialmente vamos criar um novo projeto no NetBeans



Primeiro Programa em Java

Inicialmente vamos criar um novo projeto no NetBeans



Primeiro Programa em Java

O objetivo do nosso primeiro programa é imprimir “Olá, mundo!”.

The screenshot shows a Java code editor and a terminal window. The code editor contains the following Java code:

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

The terminal window, titled "Output - Run (HelloWorld)", shows the execution results:

```
--- exec:java:exec (default-cli) & HelloWorld ---  
[1] 1000 java -jar HelloWorld.jar  
Olá, mundo!  
-----  
BUILD SUCCESS  
-----  
Total time: 2.231 s  
Finished at: 2025-05-12T14:48:42-03:00  
-----
```

Tipos Primitivos

Java possui 8 tipos primitivos básicos:

Tipo	Tamanho (bits)	Exemplo de valor
byte	8	127
short	16	32_000
int	32	2_000_000_000
long	64	9_000_000_000L
float	32	3.14f
double	64	3.141592653589
char	16	'A'
boolean	1	true / false

Variáveis

- Uma variável é um nome simbólico onde guardamos algum dado.
- Em termos técnicos, variáveis são espaços de memórias para armazenar valores.
- Você pode pensar no seguinte, imagine uma caixa e uma bola, você deseja guardar a bola na caixa para não perdê-la. A bola é a variável e a caixa o espaço de memória.

Para declarar variáveis em Java, usamos o tipo seguido do nome da variável:

```
public static void main(String[] args) {  
    int idade = 25;  
    double altura = 1.75;  
    boolean ativo = true;  
}
```

Exemplo Variáveis

```
public class Ex01Variaveis {

    public static void main(String[] args) {
        // Declaração e atribuição de variáveis
        int idade = 25;
        double altura = 1.75;
        char genero = 'M';
        boolean estudante = true;
        String nome = "João";

        // Exibindo os valores das variáveis
        System.out.println("Nome: " + nome);
        System.out.println("Idade: " + idade + " anos");
        System.out.println("Altura: " + altura + " metros");
        System.out.println("Gênero: " + genero);
        System.out.println("É estudante? " + estudante);
    }
}
```

Material de Apoio: Para rodar o código, execute o arquivo da pasta
Exemplos: Ex01Variaveis.java

Operadores Aritméticos

Operadores Aritméticos:

- Adição (+): Soma dois valores.
- Subtração (-): Subtrai um valor do outro.
- Multiplicação (*): Multiplica dois valores.
- Divisão (/): Divide um valor pelo outro.
- Módulo (%): Retorna o resto da divisão de dois números inteiros.

Operadores Aritméticos - Exemplo

```
public class Ex02OperadoresAritmeticos {

    public static void main(String[] args) {
        int a = 10;
        int b = 3;

        System.out.println("Soma: " + (a + b));          // 13
        System.out.println("Subtracao: " + (a - b));      // 7
        System.out.println("Multiplicacao: " + (a * b)); // 30
        System.out.println("Divisao: " + (a / b));        // 3 (divisão inteira)
        System.out.println("Modulo: " + (a % b));         // 1 (resto da divisão)
    }
}
```

Material de Apoio: Para rodar o código, execute o arquivo da pasta
Exemplos: Ex02OperadoresAritmeticos.java

Operadores Relacionais

Operadores Relacionais:

- **Igualdade (=):** Verifica se dois valores são iguais.
- **Diferente (!=):** Verifica se dois valores são diferentes.
- **Maior que (>):** Verifica se um valor é maior que outro.
- **Menor que (<):** Verifica se um valor é menor que outro.
- **Maior ou igual que (>=):** Verifica se um valor é maior ou igual a outro.
- **Menor ou igual que (<=):** Verifica se um valor é menor ou igual a outro.

Operadores Relacionais - Exemplo

```
public class Ex03OperadoresLogicos {

    public static void main(String[] args) {
        int idade = 18;

        System.out.println("Maior de idade? " + (idade >= 18)); // true
        System.out.println("Menor de idade? " + (idade < 18)); // false
        System.out.println("Idade é igual a 18? " + (idade == 18)); // true
        System.out.println("Idade é diferente de 21? " + (idade != 21)); // true
    }
}
```

Material de Apoio: Para rodar o código, execute o arquivo da pasta
Exemplos: Ex03OperadoresRelacionais.java

Operadores Lógicos

Os operadores lógicos são utilizados para combinar expressões lógicas e retornar um valor lógico.

O operador **E** em Java é representado por `&&`

O operador **OU** é representado por `||`

O operador **NÃO** é representado por `!`

Operadores Lógicos

Operador E (&&)

Tabela Verdade da Operação E:

p	operação	q	Resultado
V	E	V	V
V	E	F	F
F	E	V	F
F	E	F	V

Operadores Lógicos

Operador OU (||)

Tabela Verdade da Operação OU:

p	operação	q	Resultado
V	OU	V	V
V	OU	F	V
F	OU	V	V
F	OU	F	F

Operadores Lógicos

Operador NÃO (!)

Tabela Verdade da Operação NÃO:

P	operação	Resultado
V	NÃO	F
F	NÃO	V

Operadores Lógicos - Exemplo

```
public class Ex04OperadoresRelacionais {

    public static void main(String[] args) {
        boolean temCarteira = true;
        boolean fezTestePsicotecnico = false;

        // Operador AND (&&) - Ambos precisam ser verdadeiros
        boolean podeDirigir = temCarteira && fezTestePsicotecnico;
        System.out.println("Pode dirigir? " + podeDirigir); // false

        // Operador OR (||) - Basta um ser verdadeiro
        boolean acessoRestrito = temCarteira || fezTestePsicotecnico;
        System.out.println("Tem algum acesso? " + acessoRestrito); // true

        // Operador NOT (!) - Inverte o valor booleano
        System.out.println("Nao fez teste? " + !fezTestePsicotecnico); // true
    }
}
```

Material de Apoio: Para rodar o código, execute o arquivo da pasta
Exemplos: Ex04OperadoresLogicos.java



Lista de Exercícios e Tarefas

Material de Apoio:

Lista de Exercícios: Lista 01 - Introdução ao Java

Tarefas: TAREFA 01 - Introdução ao Java



CAPACITA

iRede

INovação e TECNOLOGIA

RESIDÊNCIA EM TIC 20

EXECUTOR



INICIATIVA



COORDENADORA PPI



MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÃO



*Este projeto foi apoiado pelo Ministério da Ciência, Tecnologia e Inovações, com recursos da Lei nº 8.248, de 23 de outubro de 1991, no âmbito do PPI-SOFTEX, coordenado pela Softex e publicado Residência em TIC 20, DOU 01245.002631/2023-58.

