

AULA 1 - COMEÇANDO COM JAVA

1.1 - DOWNLOAD DO INSTALADOR

EXISTEM 2 INSTALADORES: JRE E O JDK;

JRE: USADO PARA EXECUTAR PROGRAMAS JAVA NO COMPUTADOR (SW IMPOSTO RENDA);

JDK: NECESSÁRIO PARA O DESENVOLVIMENTO;

<https://www.oracle.com/java/technologies/downloads/#jdk25-windows>

JDK 25 JDK 21

Java SE Development Kit 25.0.2 downloads

JDK 25 binaries are free to use in production and free to redistribute, at no cost, under the [Oracle No-Fee Terms and Conditions \(NFTC\)](#).

JDK 25 will receive updates under the NFTC, until September 2028, a year after the release of the next LTS. Subsequent JDK 25 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**

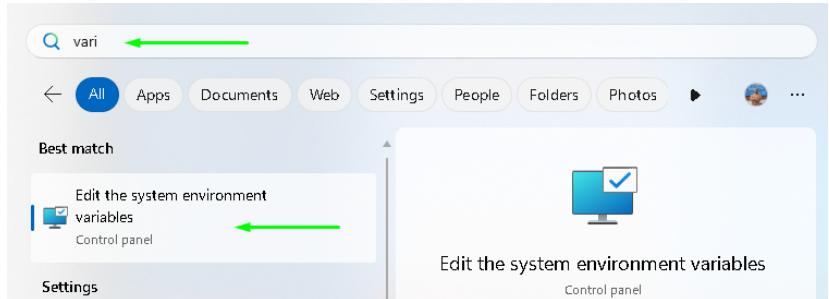
Product/file description	File size	Download
x64 Compressed Archive	204.39 MB	https://download.oracle.com/java/25/latest/jdk-25_windows-x64_bin.zip (sha256)
x64 Installer	153.27 MB	https://download.oracle.com/java/25/latest/jdk-25_windows-x64_bin.exe (sha256)
x64 MSI Installer	152.04 MB	https://download.oracle.com/java/25/latest/jdk-25_windows-x64_bin.msi (sha256)

1.2. CONFIGURAR VARIÁVEL DE AMBIENTE

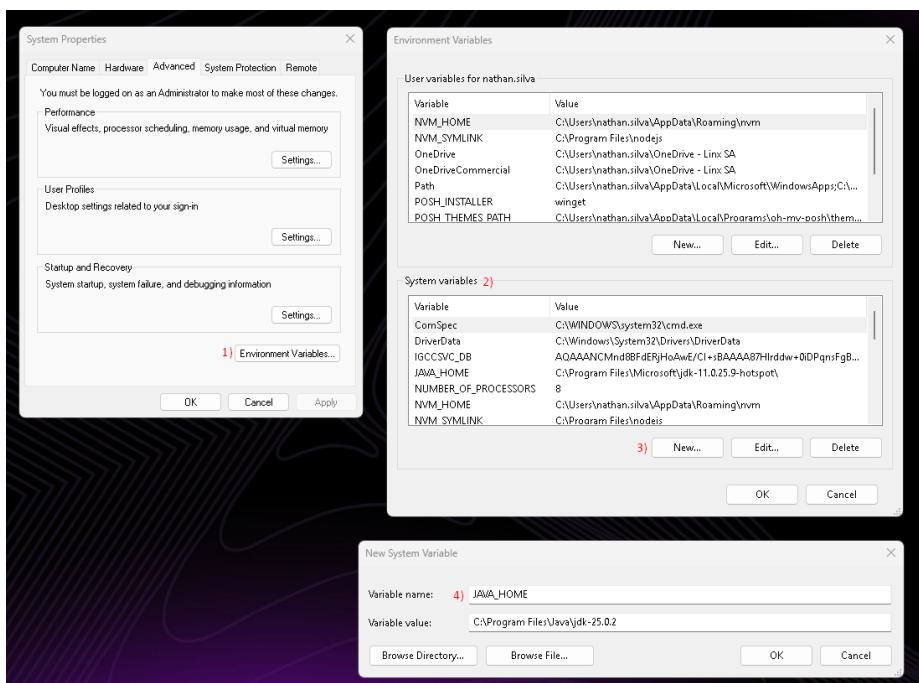
GERALMENTE A INSTALAÇÃO PADRÃO DO JDK OCORRE EM:

C:\ARQUIVOS DE PROGRAMAS\JAVA\JDK-<VERSION>

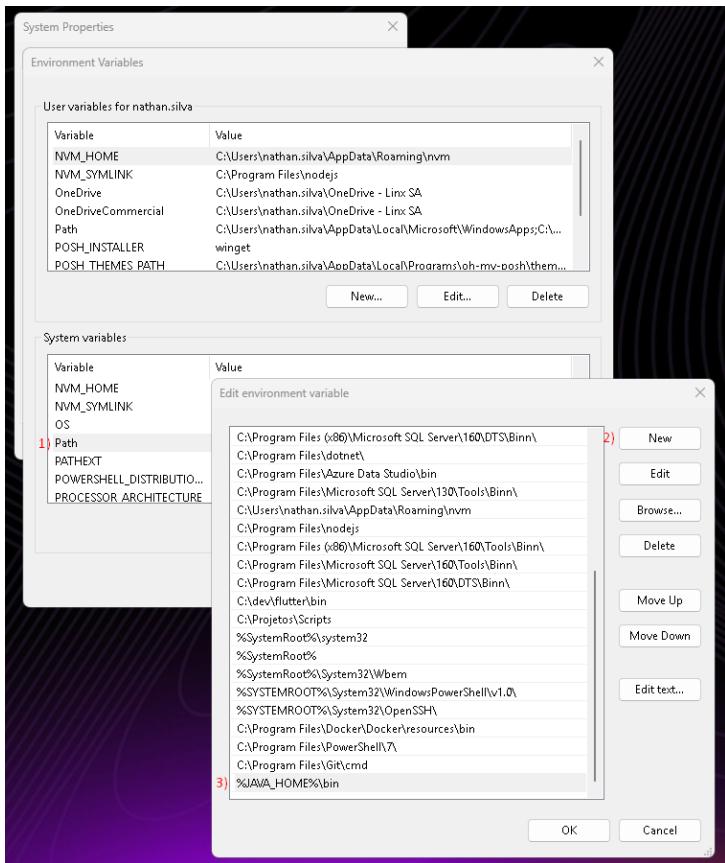
ACESSAR / PESQUISAR POR VARIÁVEIS DE AMBIENTE NO WINDOWS:



VAMOS DIZER AO COMPUTADOR OS ARQUIVOS BATCH QUE PRECISAMOS. NO CASO, ELE PRECISA SABER ONDE ESTÁ O COMPILADOR JAVAC.



AGORA PRECISAMOS ALTERAR A VARIÁVEL PATH, POIS SÃO OS CAMINHOS DOS COMANDOS DISPONÍVEIS ATRAVÉS DA LINHA DE COMANDO.



13. VERIFICANDO A CONFIGURAÇÃO

NO TERMINAL DIGITE OS COMANDOS ABAIXO E VEJA SE HOUVE RETORNO:

JAVA -VERSION

JAVAC -VERSION

```
Command Prompt
Microsoft Windows [Version 10.0.26100.7462]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nathan.silva>java -version
java version "25.0.2" 2026-01-20 LTS
Java(TM) SE Runtime Environment (build 25.0.2+10-LTS-69)
Java HotSpot(TM) 64-Bit Server VM (build 25.0.2+10-LTS-69, mixed mode, sharing)

C:\Users\nathan.silva>javac -version
javac 25.0.2

C:\Users\nathan.silva>
```

1.4. PRIMEIRO PROGRAMA EM JAVA

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

PUBLIC STATIC VOID MAIN(STRING [] ARGS) → É O PONTO DE ENTRADA PARA A EXECUÇÃO DOS PROGRAMAS.

IMPORTANTE: QUANDO CRIAR UMA CLASSE, A PRIMEIRA DE LETRA DE CADA PALAVRA É MAIÚSCULA (CAMELCASE). ALÉM DISSO, O ARQUIVO (OBIGATÓRIAMENTE) SERÁ SALVO COM O NOME DA CLASSE. SEMPRE

1.5. PROCESSO DE COMPILAÇÃO

CONSISTE EM TRANSFORMAR O CÓDIGO QUE ESCRIVEMOS EM UMA LINGUAGEM QUE A JVM COMPREENDE (.CLASS). PARA ISSO, ESTANDO NO DIREtório DO ARQUIVO (.JAVA) EXECUTE:

JAVAC HELLOWORLD.JAVA

Name	Date modified	Type	Size
HelloWorld.java	2/7/2026 4:27 PM	JAVA File	1 KB
HelloWorld.class	2/7/2026 4:48 PM	CLASS File	1 KB

UM PONTO INTERESSANTE A SER OBSERVADO É QUE O ARQUIVO GERADO (.CLASS), NÃO PODE SER FACILMENTE ABERTO. VEJA:

1.6. EXECUÇÃO DO PROGRAMA CRIADO

APÓS TER GERADO O ARQUIVO .CLASS, EXECUTE NA PASTA
EM QUE FOI GERADO O COMANDO:

JAVA HELLOWORLD

```
C:\> C:\Windows\System32\cmd.e <input> + <input>
Microsoft Windows [Version 10.0.26100.7462]
(c) Microsoft Corporation. All rights reserved.

C:\MeusRepositorios\J56C-P00\Aula01>java HelloWorld
Hello World
```

NOTE QUE A SAÍDA GERADA FOI A STRING DEFINIDA NO COMANDO
SYSTEM.OUT.PRINTLN ("HELLO WORLD").

1.7. O QUE ACONTECEU?

HELLO WORLD . JAVA

1

\$ JAVAC Helloworld.java

1

HELLOWORLD.CLASS

1

\$ JAVA HELLOWORLD

HELLO WORLD!

1.7. DETALHANDO A ESTRUTURA DO PROGRAMA

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

- Declaração da Classe
- Nome da Classe
- Método
- Modificadores de Acesso do Método
- Tipo de Retorno do Método
- Nome do Método
- Parâmetros/Argumentos do Método
- Nossa código
- Blocos estão entre {} (chaves)

DEICA: EM NOSSAS AULAS O QUE MUDA IRÁ É APENAS O
NOME DA CLASSE É A STRING DA SAÍDA DESEJADA.

1.8. AGORA É A SUA VEZ!

- 1º) CRIAR UM ARQUIVO CHAMADO OLAMUNDO.JAVA;
- 2º) IMPLEMENTAR UM PROGRAMA JAVA QUE ESCREVA NA TELA "OLA' MUNDO!".
- 3º) COMPILAR E EXECUTAR O PROGRAMA.

1.9. PASSANDO ARGUMENTOS PARA O PROGRAMA

```
class Argumentos {  
    public static void main(String[] args) {  
        System.out.println("Você digitou: " + args[0]);  
    }  
}
```

TODO PROGRAMA JAVA AO SER EXECUTADO PODE RECEBER PARÂMETROS, ELES SÃO REPRESENTADOS PELO **ARGS**.

NOTE QUE **ARGS** É UM ARRAY DE STRING, PORTANTO OS PARÂMETROS PASSADOS NA EXECUÇÃO SÃO CAPTURADOS DE FORMA POSICIONAL (ÍNDICE).

```
C:\MeusRepositorios\J56C-P00\Aula01>javac Argumentos.java ←  
C:\MeusRepositorios\J56C-P00\Aula01>java Argumentos PrimeiroArgumento  
Você digitou: PrimeiroArgumento
```

1.10. AGORA É A SUA VEZ!

- 1º) CRIAR UM ARQUIVO CHAMADO ARGUMENTOS2.JAVA;
- 2º) IMPLEMENTAR UM PROGRAMA QUE ESCREVA NA TELA "OLA' MUNDO"+ O ARGUMENTO PASSADO PARA O PROGRAMA;
- 3º) COMPILE E EXECUTAR O PROGRAMA.

1.11. ENTENDENDO OS ERROS

EXISTEM BASICAMENTE **3 TIPOS DE ERROS**, SENDO:

- ERROS DE SINTAXE;
- ERROS DE SEMÂNTICA;
- ERROS EM TEMPO DE EXECUÇÃO.

ERROS DE SINTAXE

UM PROGRAMA JAVA POSSUI UMA ESTRUTURA QUE É ESPERADA: IMPORTS (PACOTES), CLASSE, MÉTODO MAIN E DENTRO DELE O SEU CÓDIGO.

SE ABRIR CHAVES, ELA DEVE SER FECHADA. SE ABRIR PARÊNTESES, ELE DEVE SER FECHADO. TODA INSTRUÇÃO DEVE SER FINALIZADA COM UM PONTO E VÍRGULA.

CASO SE ESQUEÇA DE ALGO UM ERRO DE SINTAXE IRA' ACONTECER.

ERROS DE SEMÂNTICA

IMAGINE, POR EXEMPLO, DECLARAR UMA VARIÁVEL DE UMA FORMA QUE O JAVA NÃO ESTÁ ESPERANDO. DE REPENTE AÍ É ESTÁ DECLARADA CORRETAMENTE, MAS NÃO ESTÁ GUARDANDO NENHUM VALOR E VOCÊ TENTA USÁ-LA MESMO ASSIM.

ERROS EM TEMPO DE EXECUÇÃO

VOCÊ CRIOU UMA CLASSE, MAS NÃO DEclarou o MÉTODO MAIN. FOI IMPLEMENTADO ALGO QUE NÃO TEM como o PROGRAMA SABER, como: DIVISÃO POR ZERO.

OBS → PARA ESSE TIPO DE ERRO É QUE USAMOS TRATAMENTO DE EXCEÇÕES.

EXEMPLO DE ERRO DE SINTAXE

```
J Erro01.java X
no elements
1 < class Erro01{
2   < public static void main(String[]args){
3     < System.out.println("Hello World")
4   }
5 }
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE
PS C:\MeusRepositorios\J56C-POO\Aula01> javac Erro01.java
Erro01.java:3: error: unclosed string literal
    System.out.println("Hello World"
                           ^
1 error
```

UM PONTO INTERESSANTE A SER OBSERVADO É QUE QUANDO ESSE TIPO DE ERRO OCORRE, O ARQUIVO .CLASS NÃO É GERADO, INDICANDO QUE O PROGRAMA NÃO FOI COMPIULADO.

EXEMPLO DE ERRO DE SEMÂNTICA

TODA LINGUAGEM POSSUI FORMAS E REGRAS ESPECÍFICAS PARA A SUA UTILIZAÇÃO. A DECLARAÇÃO DE VARIAVEIS É UM EXEMPLO DISSO.

```
J Erro02.java X
no elements
1 class Erro02{
2   public static void main(String[]args){
3     var int cont;
4     System.out.println("Contador: " + cont);
5   }
6 }
```

```
PS C:\MeusRepositorios\J56C-POO\Aula01> javac Erro02.java
Erro02.java:3: error: not a statement
    var int cont;
           ^
Erro02.java:3: error: ';' expected
    var int cont;
           ^
2 errors
```

NO ENTANTO, MESMO ASUSTRANDO A SUA DECLARAÇÃO DE SINTAXE, PODEMOS NOS DEPARAR COM O SEGUINTE ERRO DE SEMÂNTICA.

```
J Erro02.java X
no elements
1 class Erro02{
2   public static void main(String[]args){
3     int cont; ←
4     System.out.println("Contador: " + cont);
5   }
6 }
```

```
PS C:\MeusRepositorios\J56C-POO\Aula01> javac Erro02.java
Erro02.java:4: error: variable cont might not have been initialized
    System.out.println("Contador: " + cont);
                           ^
1 error
```

EXEMPLO DE ERRO EM TEMPO DE EXECUÇÃO

```
J Erro03.java U X
no elements
1 class Erro03{
2     public static void Main(String[] args){
3         System.out.println("O nome é: Nathan Cirillo");
4     }
5 }
```

NESSÉ EXEMPLO É IMPORTANTE NOTAR QUE O PROGRAMA PODE COMPILADO E APENAS NA EXECUÇÃO OBTIVEMOS O ERRO!

```
● PS C:\MeusRepositorios\J56C-POO\Aula01> javac Erro03.java
○ PS C:\MeusRepositorios\J56C-POO\Aula01> java Erro03
Error: Main method not found in class Erro03, please define the main method as:
public static void main(String[] args)
or a JavaFX application class must extend javafx.application.Application
```

UM OUTRO EXEMPLO CLÁSSICO É A DIVISÃO POR 0. ELE TAMBÉM IRÁ COMPIALAR, MAS NA EXECUÇÃO TEREMOS PROBLEMAS.

```
J Erro03.java U X
no elements
1 class Erro03{
2     public static void main(String[] args){
3         System.out.println("O valor é: " + 10/0);
4     }
5 }
```

```
● PS C:\MeusRepositorios\J56C-POO\Aula01> javac Erro03.java
○ PS C:\MeusRepositorios\J56C-POO\Aula01> java Erro03
Exception in thread "main" java.lang.ArithmetricException: / by zero
at Erro03.main(Erro03.java:3)
```