

1.5 Setup

Para começar a desenvolver em Java, é necessário que tenhamos o JDK instalado.

🔔 Atenção

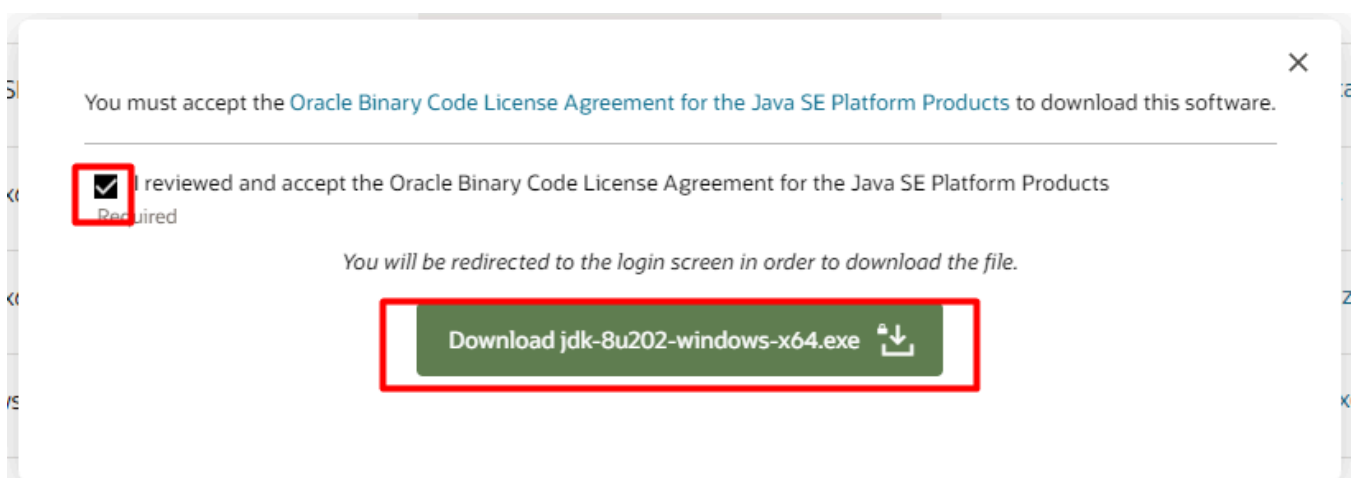
É extremamente relevante, determinar a versão do Java que precisa utilizar diante de alguns requisitos do projeto.

1.5.1 Instalando o JDK

ℹ Informação

Algumas IDEs como o VSCode o usuário consegue baixar e instalar o JDK e realizar a configuração de forma fácil.

- Busque no Google por Java **JDK xxx** (versão desejada);
- Selecione a opção **.exe** de acordo com o seu sistema operacional. No meu exemplo eu baixei a versão: **jdk-8u202-windows-x64.exe**.



- Depois, você precisará realizar um login no site da **Oracle**.

Início de sessão na conta Oracle

Nome de utilizador



Senha



Iniciar sessão

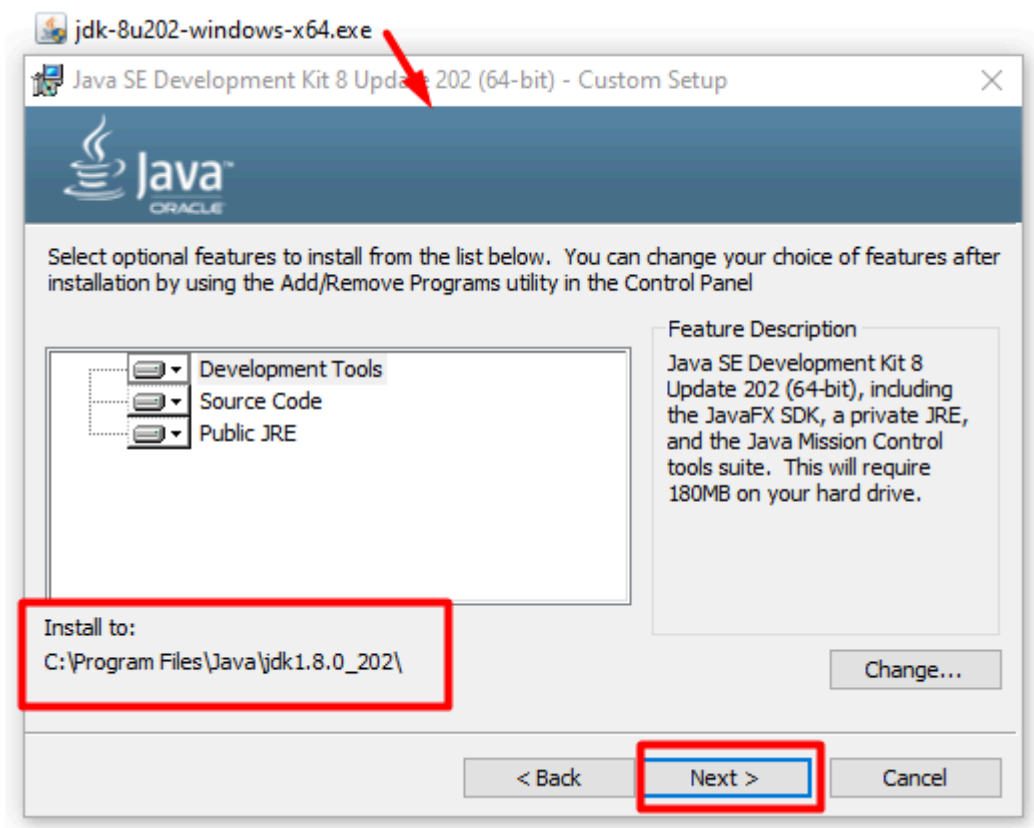
[Precisa de ajuda?](#)

Não tem uma Conta Oracle?

Criar Conta

© Oracle | [Condições de Utilização](#) | [Política de Privacidade](#)

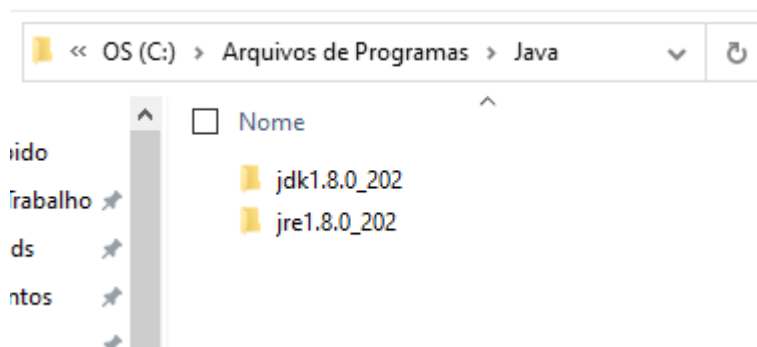
- Após realizar o download, executar o instalador para instalar o Java no Windows. Este processo instalará tanto o **JDK** quanto a **JRE**.



⚠ Atenção

Evite mudar o diretório de instalação.

- Em seu explorer, deve ter algo mais ou menos assim:



- Precisamos agora, validar se a instalação também já configurou nossa variável de ambiente, para poder executar o Java pelo Prompt de comando ou PowerShell do Windows.
- Abra o Prompt de comando e execute o comando `java -version`.

C:\> Prompt de Comando

```
C:\Users\gleys>java -version
java version "1.8.0_202"
Java(TM) SE Runtime Environment (build 1.8.0_202-b08)
Java HotSpot(TM) 64-Bit Server VM (build 25.202-b08, mixed mode)
```

Informação

Não iremos configurar as variáveis de ambiente, pois não é um pré-requisito para desenvolver em Java, considerando que iremos usufruir das mais poderosas IDEs para aumentar nossa produtividade em codificar a linguagem.

1.5.2 Escolha sua IDE

Para aumento de produtividade, podemos escolher algumas ferramentas denominadas de IDE - Integrated Development Environment ou Ambiente de Desenvolvimento Integrado.

É um software que dispõe de recursos que auxilia muito nosso desenvolvimento como:

- Aceleração de nossa escrita com **Auto-complete** ou **IntelliSense**
- Formatação de palavras e blocos de códigos
- Análise de erro de sintaxe
- Compilação de programa
- Depuração (acompanhamento) de execução do programa.

Cada IDE dispõe de recursos comuns ou plugins específicos para te auxiliar na agilidade de desenvolvimento.



Informação

Antes de escolher sua IDE recomendamos realizar uma pesquisa e ter uma experiência em cada uma delas.

- Eclipse: Mais utilizada no ambiente corporativo com interface bem interativa.
- NetBeans: É mais utilizada em ambiente acadêmico e com versão de menu e assistentes em português.
- IntelliJ: Oferece muitos recursos de inteligência para agilidade de nosso desenvolvimento com sugestões de ações bem assertivas.
- VS Code: É muito utilizada por desenvolvedores que já atuam com NodeJs ou plataformas de front-end.

Eclipse

O **Eclipse IDE** é uma alternativa muito comum de encontrar ambientes de desenvolvimento, principalmente pelo aspecto de uma interface simples, plugins para recursos adicionais, atalhos para operações de menu e de escrita e para quem trabalha com Java Swing você pode adicionar o plugin **WindowBuilder** que dispõe de uma interface gráfica para criar as telas gerando o código de layout para o desenvolvedor.

Atenção

Em caso de erro quanto a versão do Java no Eclipse [veja o vídeo pelo link](#)

IntelliJ

O **IntelliJ** é uma plataforma de código aberto para criação de IDEs e ferramentas para desenvolvedores, mas nem todos os recursos estão disponíveis de forma gratuita.

Mesmo com a proposta de ser inteligente, talvez em um primeiro contato, seja difícil desfrutar de todos os recursos, fazendo com você opte por mudar ou voltar para uma IDE, como o Eclipse. Claro que não podemos deixar de destacar que muitas coisas a gente não precisa, como: Salvar arquivo, encode padrão UTF-8 e sugestões que vão surgindo ao longo da nossa interação com a ferramenta.

Visual Studio Code

Desenvolvido pela Microsoft, é uma editor de texto muito simples e interativo que conta com inúmeras extensões, temas variados e integração com tecnologias como NodeJS e Java.

Comparando-se em performance, o VS Code demonstra um grande desempenho de inicialização, codificação e execução dos programas desenvolvidos.

Resumindo:

- **Eclipse:** Mais utilizada no ambiente corporativo com interface bem interativa.
- **NetBeans:** É mais utilizada em ambiente acadêmico e com versão de menu e assistentes em português.
- **IntelliJ:** Oferece muitos recursos de inteligência para agilidade de nosso desenvolvimento, como sugestões de ações bem assertivas.

1.5.3 Mão na massa

Vamos explorar alguns principais recursos utilizando as IDEs, Eclipse, IntelliJ e VSCode.



- O que são workspaces ?
- Definir a JDK na IDE
- Criar \ abrir um projeto Java
- Criar \ importar um projeto Maven
- Conhecer os principais atalhos
- Code Snippet
- Executar \ Depurar nosso programa
- Conhecer alguns atalhos

Maven, é uma ferramenta de automação de compilação utilizada primariamente em projetos Java, mas hoje também é utilizada para construir e gerenciar projetos escritos em C#, Ruby, Scala e outras linguagens

1.5.4 Atalhos

Todas as IDEs possuem combinações de comandos que são facilitadores para criação de arquivo, digitação de códigos, localização e execução das classes em nosso programa, além de comandos já pré-definidos de linhas de código (code snippet).

Code Snippets, que numa tradução livre seria algo como “retalhos de código”, são pedaços de código pré-fabricados e parametrizáveis que podemos inserir em nossa aplicação

Linha do código

Comparação de alguns atalhos

Comando	Eclipse	IntelliJ
Cria o método principal main	main, ctrl + espaço, enter	psvm
Localiza recurso \ arquivo	ctrl + shit + R	ctrl+shift + N
Método de impressão no Console	sysout, ctrl + espaço	sout
Renomeando variável ou método	alt+shift+R	shift + F6
Localizar a utilização de um método	ctrl+shift+G	alt + F7
Entrar na implementação do método	ctrl+click	ctrl+click

1.5.5 Debug

Acompanhar o estado de nossos objetos, o valor das nossas variáveis e até aonde o nosso programa está executado sem erros, é algo extremamente importante para

O termo Debug deriva da expressão "procurar e remover insetos", termo este utilizado devido localização de uma mariposa (**Bug**) presa no relé em um dos grandes computadores da época como o Mark II de Harvard, ocasionando assim falhas no processo.



Depurar é a capacidade que o desenvolvedor adquire em prever e constatar em um determinado ponto na aplicação, o estado atual das variáveis que representam os valores e objetos existentes ao longo da execução de um programa.

Primeiro, inicie o seu programa em modo **debug**.

1. **Funcionalidade de Step Over (F8)** – Avança para a próxima linha.
2. **Funcionalidade de Step Into (F7)** – Avança para a próxima linha considerando o interior da próxima chamada.

3. Funcionalidade de Resume Program (F9) – Libera a execução do programa até que encontre outro breakpoint

Abaixo iremos ilustrar que com uma aplicação simples de operações matemáticas podemos praticar depuração de código.

java

```
1  public class OperacaoMatematica {
2      public static void main(String[] args) {
3          int numero = 0;
4
5          numero = 1+1 ;
6
7          numero = multiplicar(3,5);
8
9          numero = numero - 5;
10
11         numero = numero / 2;
12
13         // qual o resultado que será impresso?
14         // e quais foram o valores da variável número ao longo da execução ?
15         System.out.println(numero);
16
17     }
18     static int multiplicar(int numeroUm, int numeroDois){
19         int resultado = numeroUm * numeroDois;
20         return resultado;
21     }
22 }
```

Pare para pensar:

1. Qual seria o valor da variável `numero` na linha 5?
2. Qual seria o valor da variável `numero` na linha 7?
3. Qual seria o valor da variável `numero` na linha 15?

Sucesso

Depurar o código, é a capacidade e a percepção de prevê o resultado da execução em ponto específico.