
SENAI

GERENCIADORES DE DEPENDÊNCIAS

TÓPICOS ABORDADOS

- O que são dependências de um projeto?
- O que são gerenciadores de dependências?
  - ➢ Por que são Importantes?
- Principais Gerenciadores de Dependências para Java.
- Maven
  - ➢ O que é o Maven.
  - ➢ Como Funciona o Maven?
  - ➢ Vantagens do Maven.

GERENCIADORES DE DEPENDÊNCIAS: **DEPENDÊNCIAS DE UM PROJETO**

SENAI

GERENCIADORES DE DEPENDÊNCIAS

O QUE SÃO DEPENDÊNCIAS DE UM PROJETO?

- No contexto do desenvolvimento de software, dependências são **bibliotecas** ou **módulos externos** que um projeto precisa para funcionar corretamente.
- Elas fornecem funcionalidades específicas que o desenvolvedor **não precisa implementar do zero**, como acesso a banco de dados, manipulação de arquivos JSON, ou frameworks de desenvolvimento.

GERENCIADORES DE DEPENDÊNCIAS
TIPOS DE DEPENDÊNCIAS

Dependências Diretas:

- São aquelas que o desenvolvedor adiciona explicitamente ao projeto. Exemplo: adicionar o **Hibernate** para gerenciar o banco de dados.

Dependências Transitivas:

- São dependências das dependências. Por exemplo, se o **Hibernate** depende do **slf4j**, ele será incluído automaticamente no projeto.

GERENCIADORES DE DEPENDÊNCIAS: **O QUE SÃO?**

SENAI

GERENCIADORES DE DEPENDÊNCIAS

O QUE SÃO GERENCIADORES DE DEPENDÊNCIAS?

- São ferramentas que automatizam a inclusão, atualização e gerenciamento de **bibliotecas externas** (dependências) em um projeto de software.
- **Simplificam** o processo de **configurar** e **manter** as dependências necessárias para que o projeto funcione corretamente.

GERENCIADORES DE DEPENDÊNCIAS
POR QUE SÃO IMPORTANTES?

SENAI

1. **Automação e Simplificação:**
   - Reduzem a complexidade de gerenciar dependências manualmente. Sem um gerenciador, o desenvolvedor precisaria baixar e configurar cada biblioteca, além de lidar com possíveis conflitos de versões.
2. **Controle de Versões:**
   - Facilitam o uso de versões específicas de uma biblioteca, garantindo que o projeto seja consistente em diferentes ambientes de desenvolvimento.

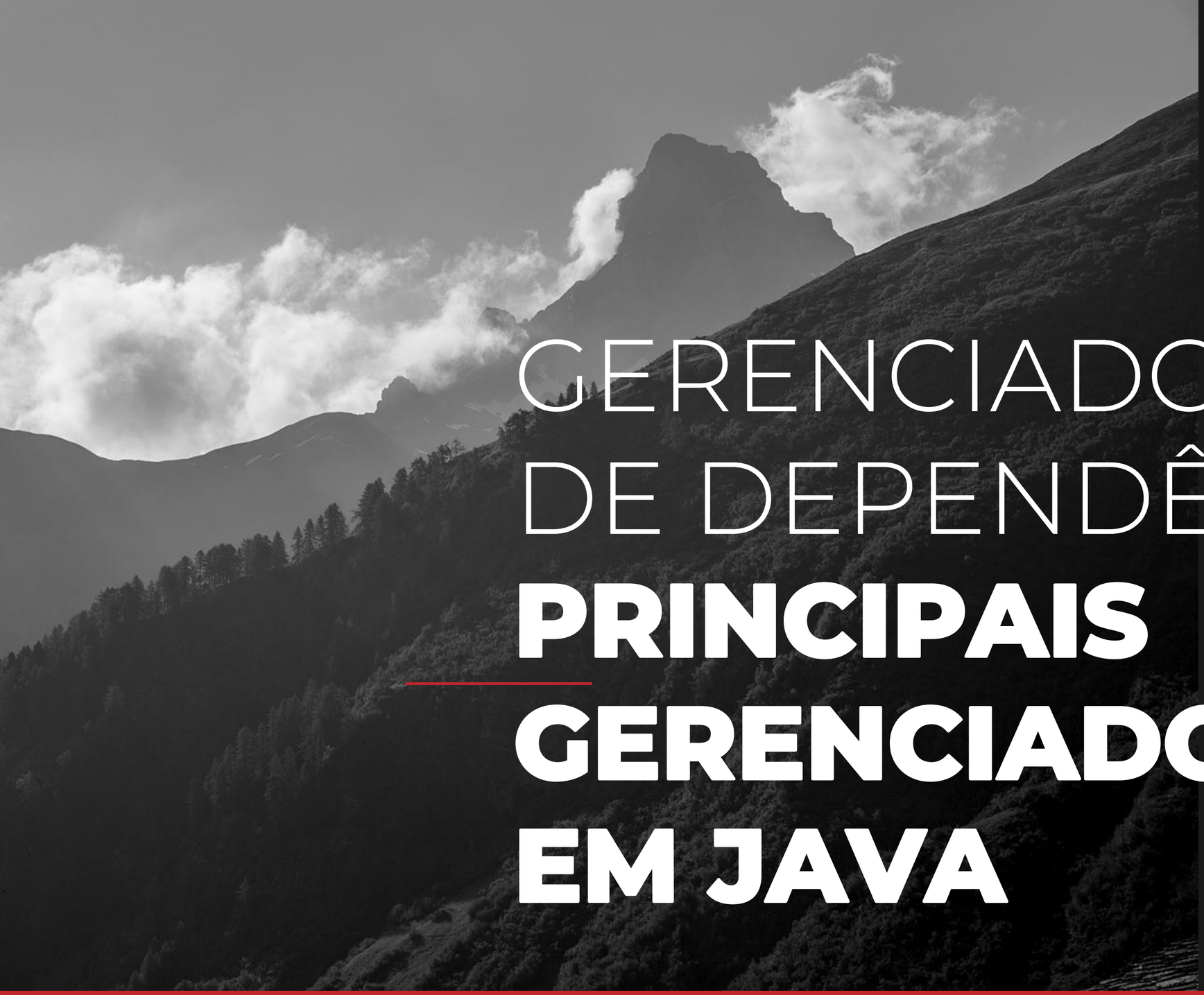
GERENCIADORES DE DEPENDÊNCIAS
POR QUE SÃO IMPORTANTES?

3. **Resolução de Conflitos:**
   - Quando duas bibliotecas dependem de diferentes versões de uma mesma dependência, os gerenciadores podem resolver esses conflitos automaticamente
4. **Centralização e Reutilização**:
   - Permitem a centralização das dependências em arquivos de configuração, tornando o projeto mais organizado e reusável.

GERENCIADORES DE DEPENDÊNCIAS
POR QUE SÃO IMPORTANTES?

SENAI

5. **Integração com Repositórios Remotos:**
   - Os gerenciadores baixam as dependências diretamente de repositórios remotos, como o **Maven Central**, evitando que o desenvolvedor precise procurá-las manualmente.



GERENCIADORES DE DEPENDÊNCIAS: **PRINCIPAIS GERENCIADORES EM JAVA**

SENAI

GERENCIADORES DE DEPENDÊNCIAS

PRINCIPAIS GERENCIADORES EM JAVA

1. **Maven**
   - **Uso:** Gerenciamento de dependências, automação de builds e integração contínua (pipelines).
   - **Arquivo principal:** pom.xml.

2. **Gradle**
   - **Uso:** Alternativa ao Maven, com maior flexibilidade e performance.
   - **Arquivo principal:** build.gradle.

3. **Ivy**
   - **Uso:** Extensão do Apache Ant para gerenciar dependências.
   - **Arquivo principal:** ivy.xml.

GERENCIADORES DE DEPENDÊNCIAS
PRINCIPAIS GERENCIADORES EM JAVA

SENAI

4. **SBT (Scala Build Tool)**
   - **Uso:** Focado em projetos Scala, mas suporta Java.
   - **Arquivo principal:** build.sbt.

5. **Ant com Ivy**
   - **Uso:** Permite gerenciamento de dependências em projetos Ant.
   - **Arquivo principal:** build.xml com ivy.xml.

GERENCIADORES DE DEPENDÊNCIAS: **MAVEN**

SENAI

GERENCIADORES DE DEPENDÊNCIAS

O QUE É O MAVEN?

Apache **Maven** é uma ferramenta de gerenciamento de projetos e automação de builds para Java. Além de compilar e construir o projeto, ele é amplamente utilizado como gerenciador de dependências.

Estrutura de um Projeto Maven:

- **pom.xml (Project Object Model):** Arquivo XML que centraliza as configurações do projeto. Nele, você define:
  - As dependências do projeto.
  - Plugins e configurações de build.
  - Informações do projeto (nome, versão, descrição, etc.).

GERENCIADORES DE DEPENDÊNCIAS
COMO FUNCIONA O MAVEN?

- **Definição de Dependências:** O desenvolvedor define as bibliotecas no **pom.xml**.
- **Resolução de Dependências:** O Maven **baixa automaticamente** essas bibliotecas e suas dependências transitivas (bibliotecas necessárias para que as dependências principais funcionem).
- **Build e Execução:** Com todas as dependências resolvidas, o **Maven compila**, **testa** e **empacota** o projeto.

GERENCIADORES DE DEPENDÊNCIAS

VANTAGENS DO MAVEN?

1. **Facilidade de Configuração:** Com poucas linhas de XML, é possível **adicionar dependências** e configurar o projeto.
2. **Repositórios Centralizados:** Maven busca bibliotecas em **repositórios públicos** (ex.: Maven Central), garantindo acesso fácil a milhares de pacotes.
3. **Portabilidade:** Qualquer desenvolvedor pode clonar o repositório do projeto, executar um comando Maven **(mvn clean install**), e ter o projeto pronto para rodar com todas as dependências configuradas.
4. **Automação do Build:** O Maven permite automatizar processos como **compilação**, **testes**, **empacotamento**, e **publicação**.

GERENCIADORES DE DEPENDÊNCIAS
EXEMPLO PRÁTICO

Considere um projeto que precise de uma biblioteca para **manipular JSON**, como a **Gson**. Basta adicionar a dependência no ***pom.xml***:

```
<dependency>
    <groupId>com.google.code.gson</groupId>
    <artifactId>gson</artifactId>
    <version>2.8.9</version>
</dependency>
```

Após salvar, execute o comando: ***mvn compile***
O Maven fará o download da biblioteca Gson e a integrará ao projeto.

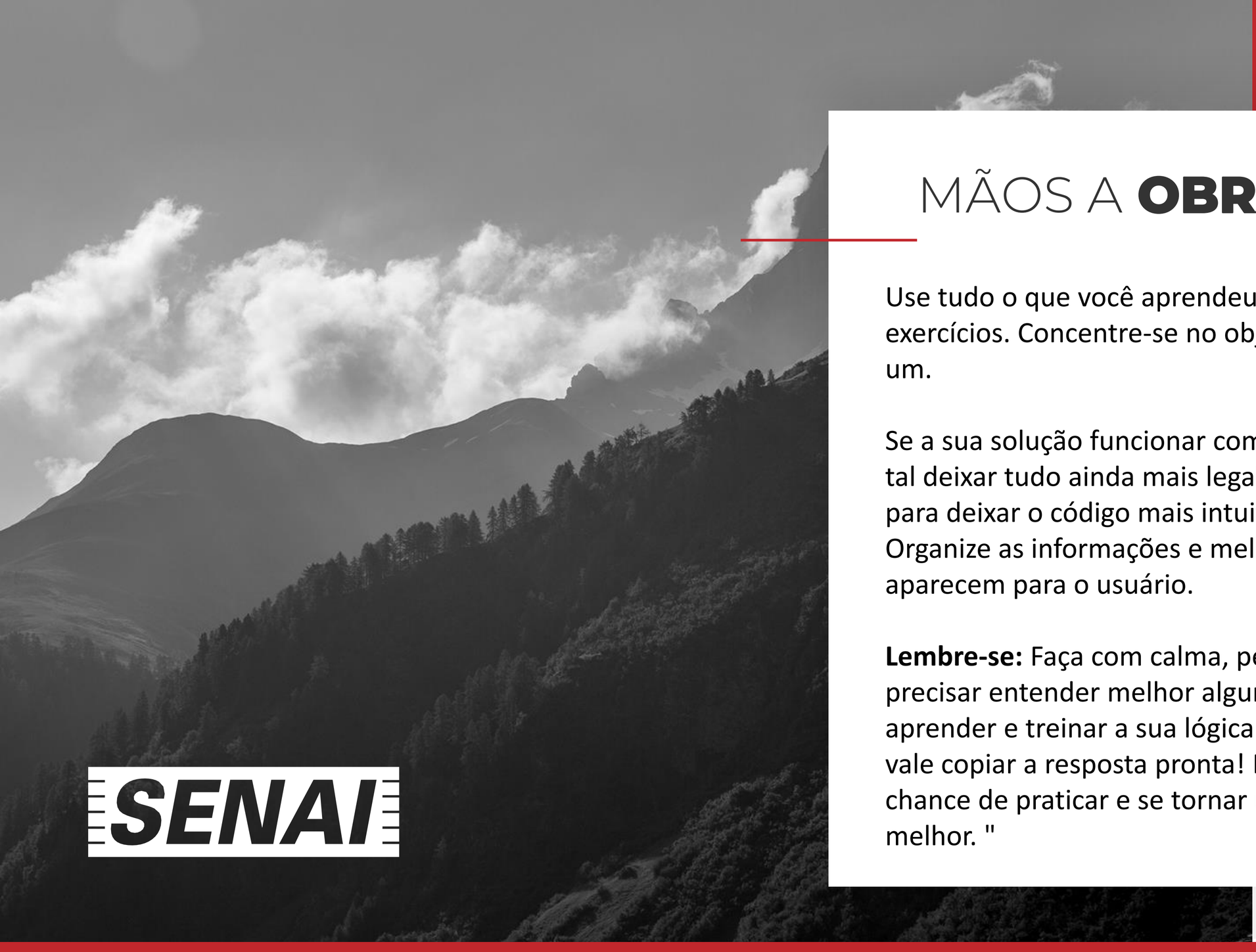
GERENCIADORES DE DEPENDÊNCIAS

EXERCÍCIOS:

USO DO **MAVEN:**

- ADICIONAR DEPENDÊNCIAS
- BUILDAR PROJETO – (.jar)

SENAI



SENAI

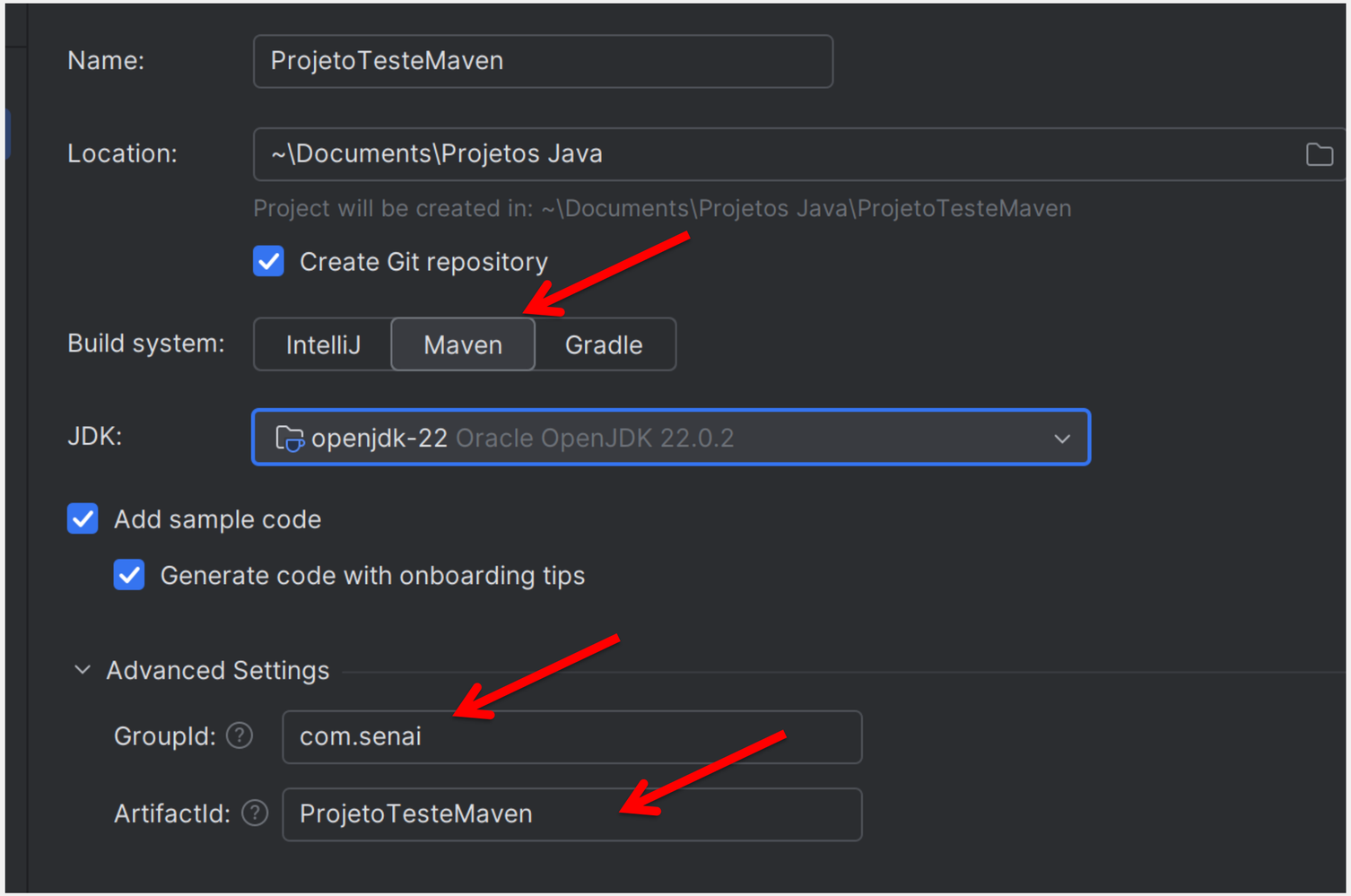
MÃOS A **OBRA**

Use tudo o que você aprendeu para resolver os exercícios. Concentre-se no objetivo principal de cada um.

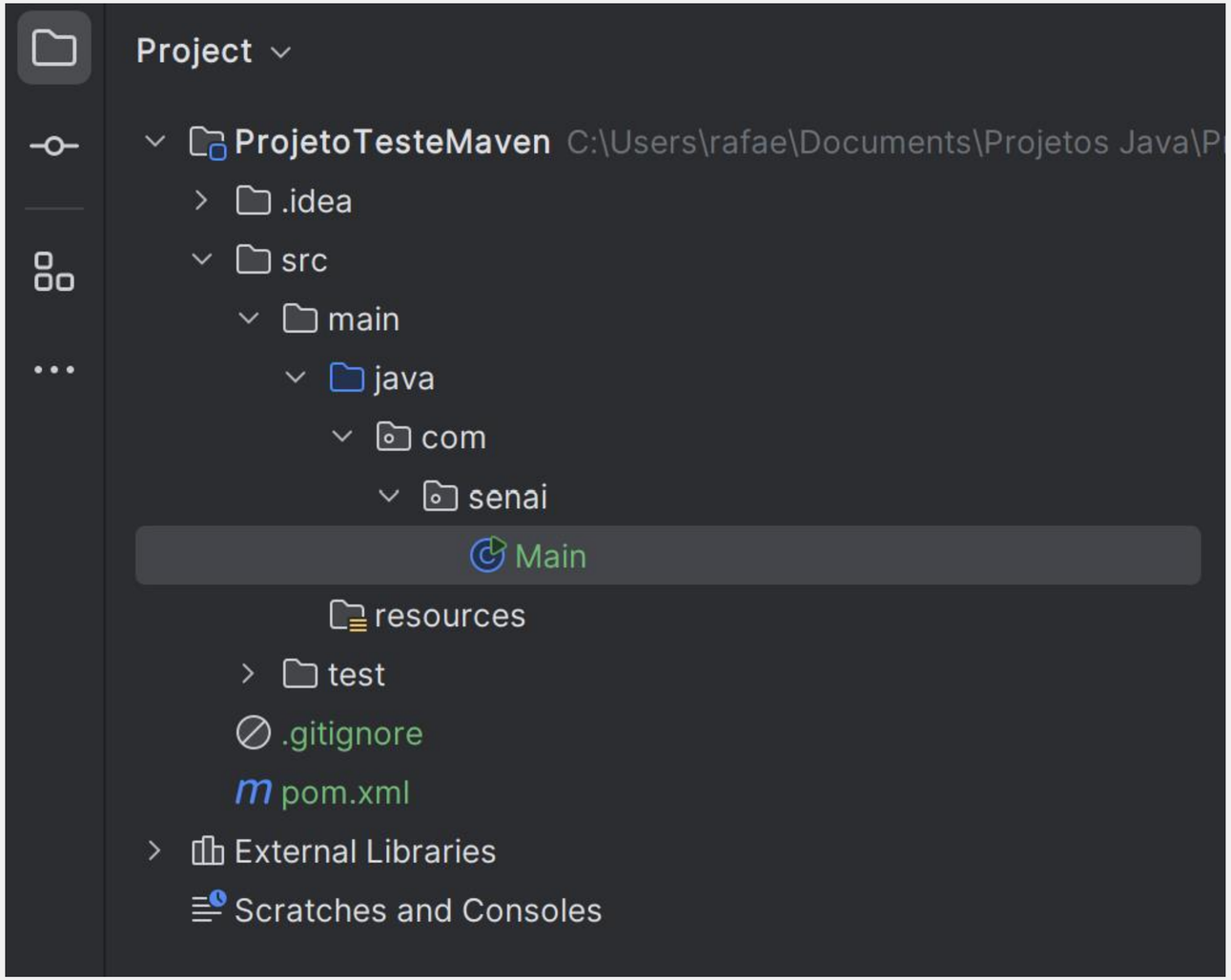
Se a sua solução funcionar como o exercício pede, que tal deixar tudo ainda mais legal? Use a sua criatividade para deixar o código mais intuitivo e fácil de entender. Organize as informações e melhore os textos que aparecem para o usuário.

Lembre-se: Faça com calma, peça ajuda quando precisar entender melhor alguma coisa. O importante é aprender e treinar a sua lógica de programação. Não vale copiar a resposta pronta! Esses exercícios são a sua chance de praticar e se tornar um programador ainda melhor. "

GERENCIADORES DE DEPENDÊNCIAS: EXERCÍCIOS



GERENCIADORES DE DEPENDÊNCIAS: EXERCÍCIOS



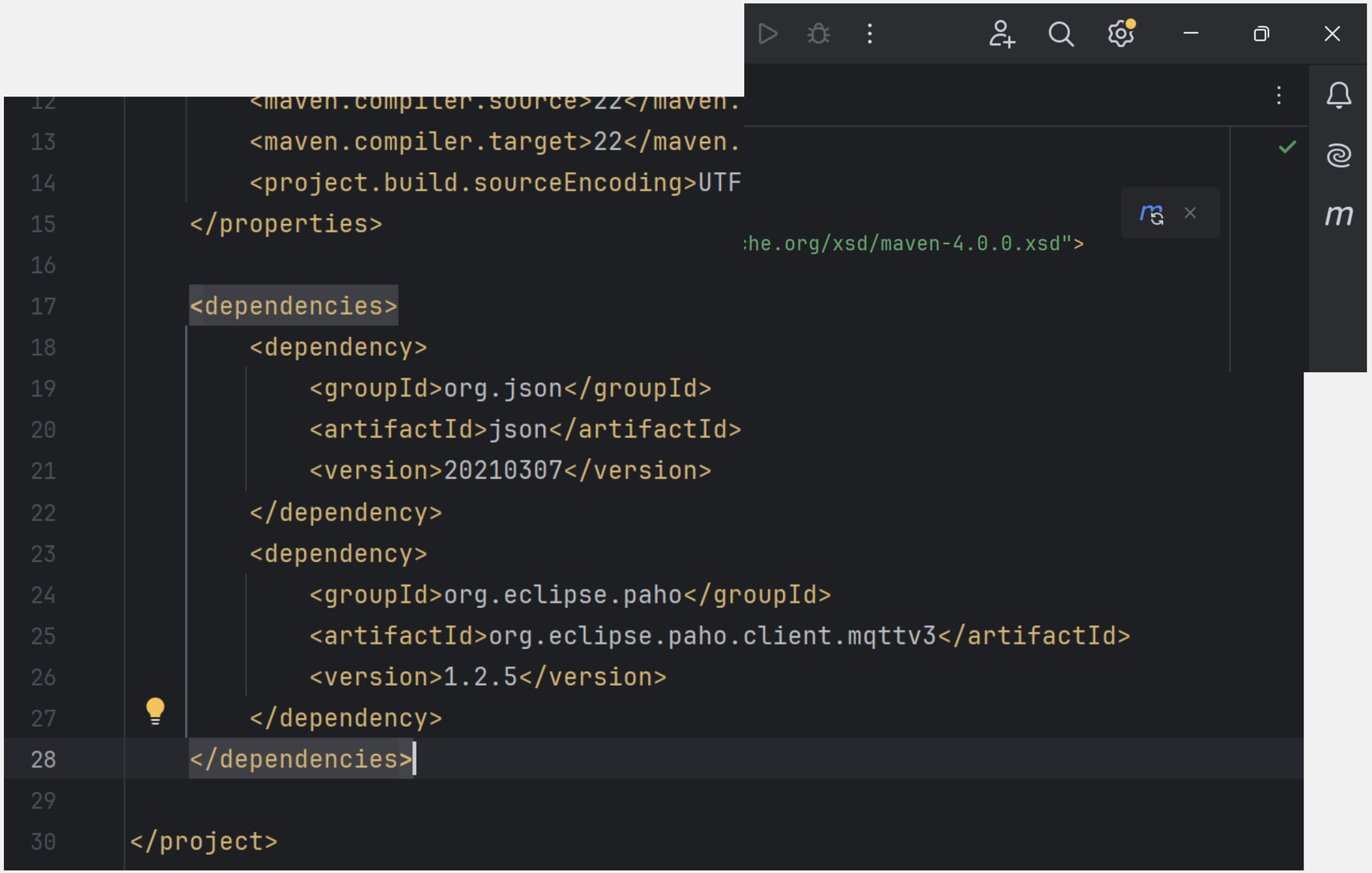
GERENCIADORES DE DEPENDÊNCIAS: EXERCÍCIOS

SENAI

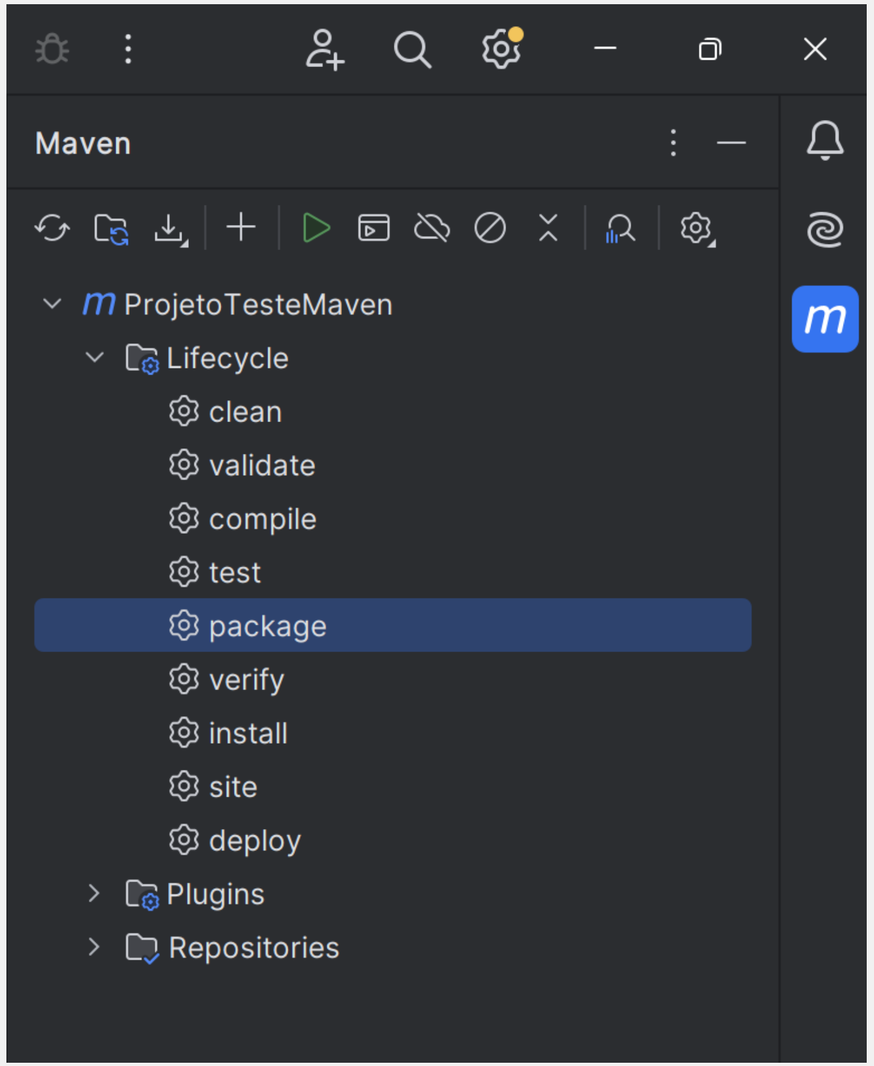
lain.java × m pom.xml (ProjetoTesteMaven) ×

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.senai</groupId>
    <artifactId>ProjetoTesteMaven</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <maven.compiler.source>22</maven.compiler.source>
        <maven.compiler.target>22</maven.compiler.target>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>

</project>
```

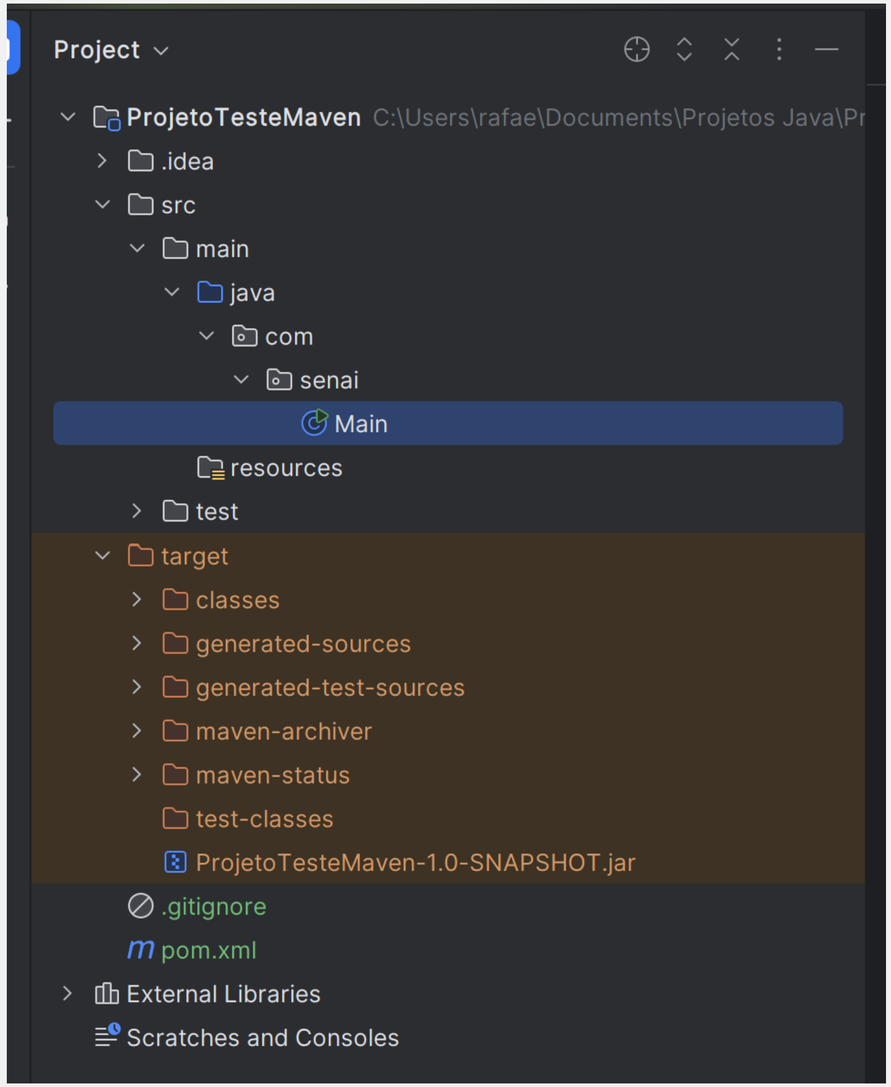
GERENCIADORES DE DEPENDÊNCIAS: EXERCÍCIOS



GERENCIADORES DE DEPENDÊNCIAS: EXERCÍCIOS



GERENCIADORES DE DEPENDÊNCIAS: EXERCÍCIOS




SENAI
DEPARTAMENTO REGIONAL
DE SÃO PAULO
www.sp.senai.br