



**Certified Tech
Developer**
The Ultimate Degree

Maven

À medida que o sistema que estamos desenvolvendo cresce e precisa atender a mais necessidades, aumenta a complexidade para construção de novas funcionalidades e sua manutenção. Vejamos a seguinte situação:

Como o Maven pode nos ajudar nessa situação?

Usando o Maven, podemos adicionar automaticamente as bibliotecas ao projeto. Este se encarregará de localizá-las e baixá-las.

Desta forma, qualquer pessoa que deseja executar nosso código não precisará localizar e adicionar cada biblioteca manualmente.

O Maven se encarrega de compilar nosso código e executar todos os testes que dispomos. Assim teremos a certeza de que as bibliotecas foram adicionadas ao projeto e que estão funcionando corretamente.

Como funciona?

O Maven baseia-se no conceito de arquivos POM (Project Object Model). Um arquivo POM é como representamos um projeto para o Maven, este arquivo contém todas as configurações mínimas para que ele funcione corretamente. O arquivo POM, quando criamos, é herdado do Super POM, desta forma, temos apenas que adicionar alguns detalhes de configuração. Vamos ver um exemplo:

```
<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
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
```

```
  <groupId>com.empresa.project-group</groupId>
  <artifactId>project</artifactId>
  <version>1.0</version>
```

Configuração obrigatória
para a criação de um
projeto Maven



```
  <dependencies>
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.16</version>
  </dependency>
```

```
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.12</version>
    <scope>test</scope>
  </dependency>
</dependencies>
</project>
```

Dentro do grupo de
dependências, adicionaremos
as dependências (bibliotecas)
de que precisamos. Atenção! O
Maven não usa o conceito de
biblioteca, pois uma
dependência pode ser muito
maior do que uma biblioteca.

```
<groupId>com.empresa.project-group</groupId>
<artifactId>project</artifactId>
<version>1.0</version>

<dependencies>
<dependency>

    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.16</version>
</dependency>

<dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.12</version>
    <scope>test</scope>
</dependency>
</dependencies>
</project>
```

GroupId: é utilizado para indicar o nome da organização ou empresa e do projeto.

ArtifactId: é utilizado para atribuir uma identificação ao projeto.

Versão: é usado para indicar a versão do projeto.

```
<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
https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.empresa.project-group</groupId>
    <artifactId>project</artifactId>
    <version>1.0</version>
<dependencies>
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.16</version>
</dependency>
```

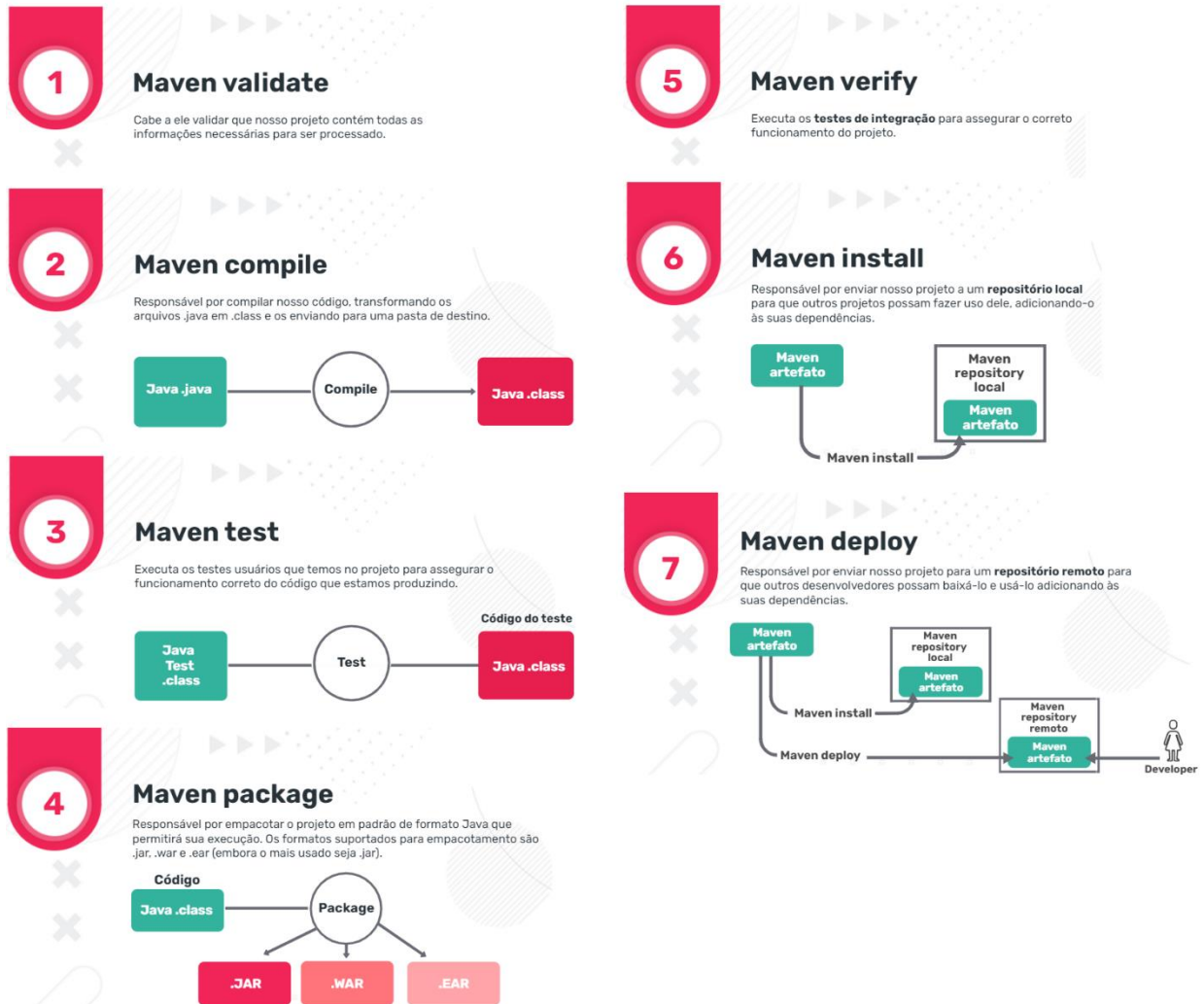
```
<dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <version>4.12</version>
  <scope>test</scope>
</dependency>
</dependencies>
</project>
```

Ciclo de vida de um projeto Maven

O Maven define uma série de fases pelas quais nosso código passará. As mais utilizadas são:

- Maven validate;
- Maven compile;
- Maven test;
- Maven package;
- Maven verify;
- Maven install;
- Maven deploy.

Vamos conhecer cada uma em detalhes.



Repositórios

Como vimos nas fases de instalação (Maven install) e implantação (Maven deploy), o Maven envia nosso projeto para repositórios, locais ou remotos. Os repositórios são locais onde o Maven irá buscar as dependências que adicionamos ao POM.

Conclusão

Depois de ver o material, podemos concluir que usar o Maven em nossos projetos nos permite focar no que realmente importa para nós, a lógica de nossa aplicação. Além disso, incentiva a reutilização de código e se encarrega da maioria das etapas relacionadas à construção de nosso sistema.

Mãos à obra!

Agora é sua hora de praticar. Sugerimos que você faça o exercício dado na aula de Padrão DAO sobre “Voos fretados”, mas usando o Maven. Para isso, devemos criar o arquivo POM e incluir as bibliotecas utilizadas anteriormente como dependências.

Atenção! Você pode copiar e colar o código Java desenvolvido.