

Apache Maven

Gestão de Construção

Sistemas de Gestão de Construção

- Controla a transformação de itens fonte em itens derivados
- Faz a gestão de dependências dos módulos envolvidos
 - Evita compilações desnecessárias
 - Calcula transitivamente as dependências



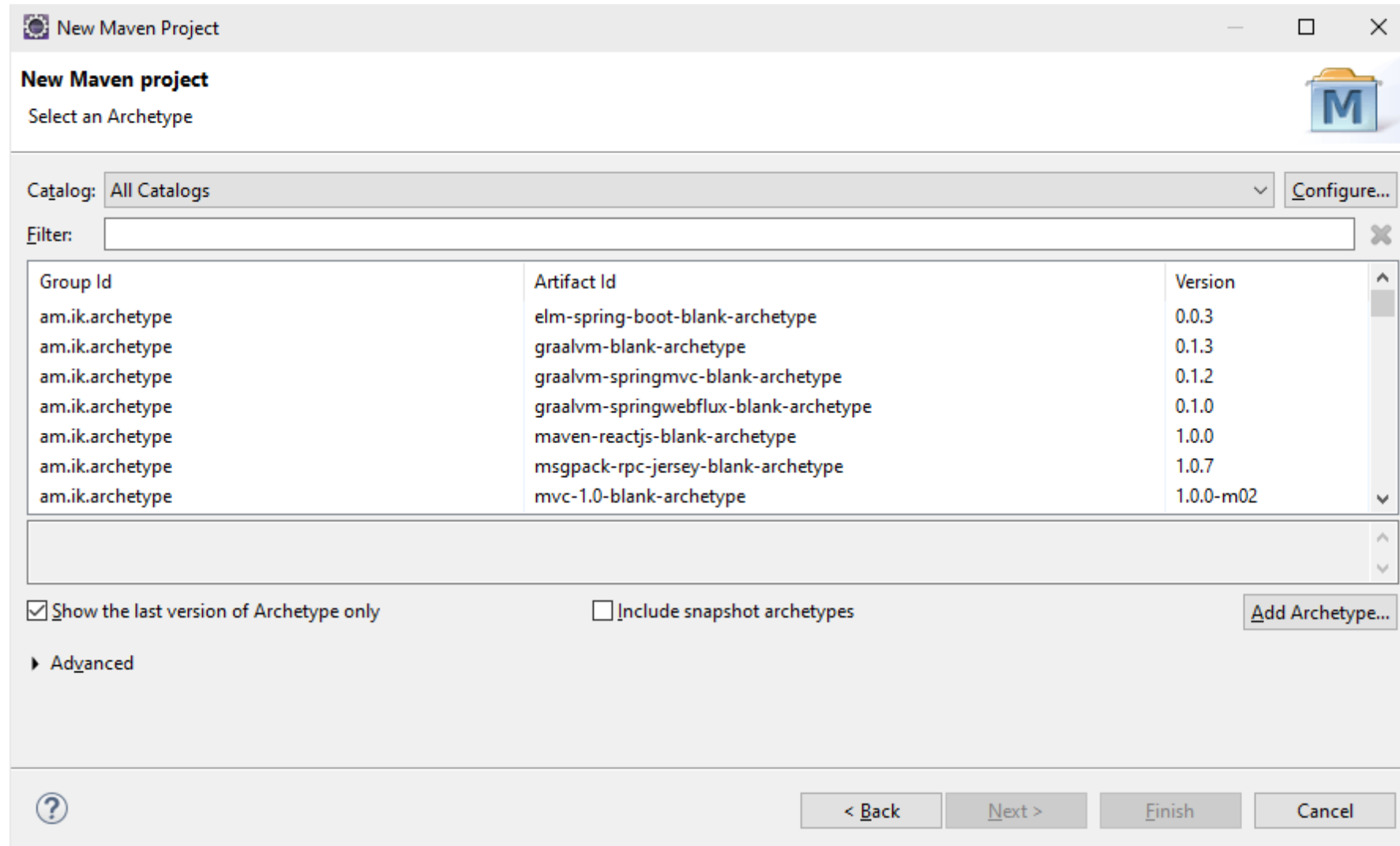
Elementos Básicos

- POM (*Project Object Model*)
 - Descritor XML da estrutura de um projeto
- Ciclo de vida (*lifecycle*)
 - Processo de construção
 - Ex.: compile → test → package → install → deploy
- Fase (*phase*)
 - Passo do processo de construção
 - Ex.: compile
- Plug-in
 - Ferramenta utilizada no processo de construção
 - Ex.: scm
- Meta (*goal*)
 - Funcionalidade provida por uma ferramenta
 - Ex.: scm:checkin

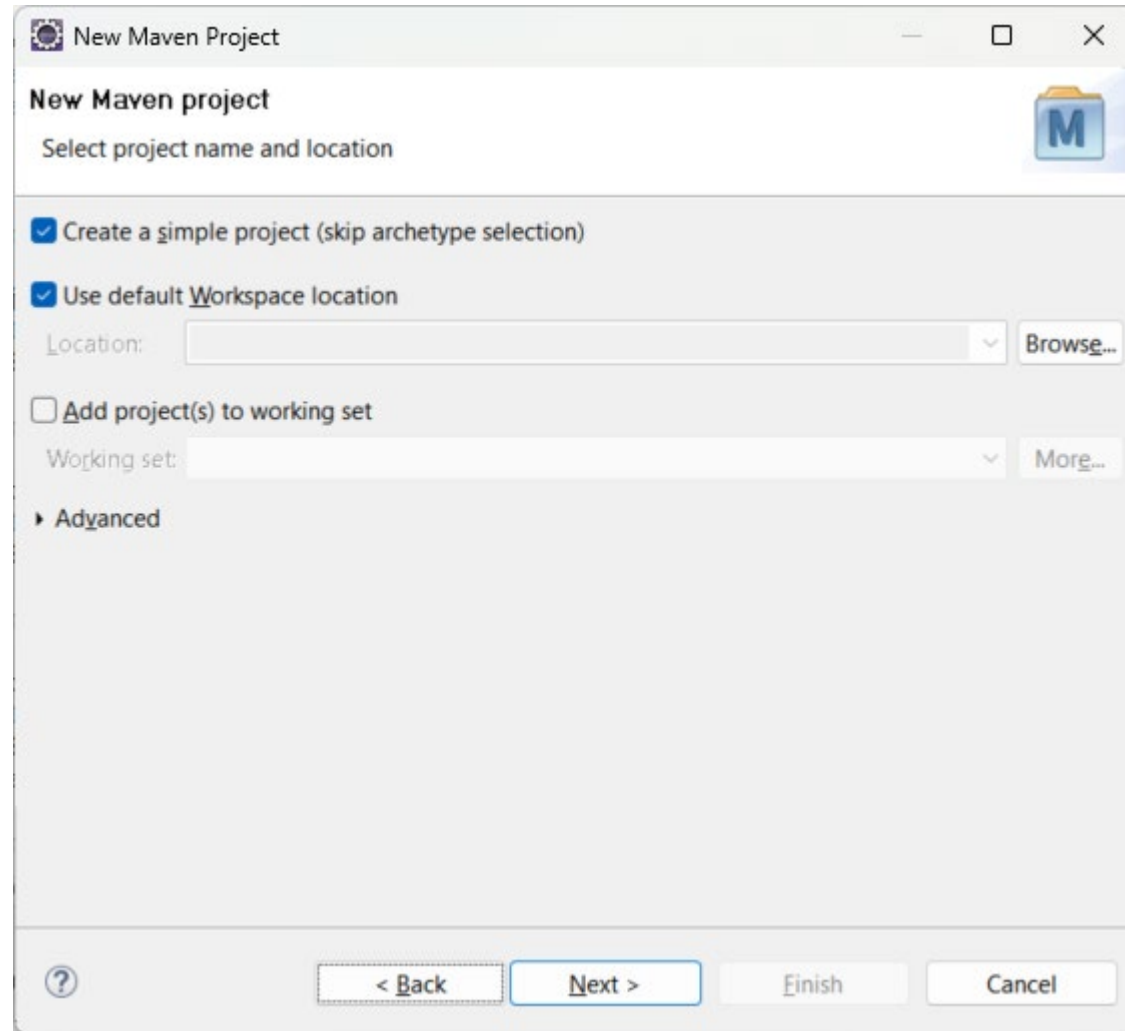
Geração da Estrutura do Projeto

- Problema: “Onde cada artefato do projeto deve ser colocado?”
- Maven permite gerar o **esqueleto do projeto** (layout) aderente às suas convenções
- Fornece 292 layouts, dentre eles:
 - Aplicação Java, Groovy, Ruby, Scala
 - Site web
 - Documento DocBook e LaTeX
 - Maven Archetype (ou seja, é possível criar o seu próprio layout)

Geração da Estrutura do Projeto (Eclipse)



Geração da Estrutura do Projeto (Eclipse)



Geração da Estrutura do Projeto (Eclipse)

New Maven Project

New Maven project

Enter a group id for the artifact.

Artifact

Group Id:

Artifact Id:

Version: 0.0.1-SNAPSHOT

Packaging: war

Name:

Description:

Parent Project

Group Id:

Artifact Id:

Version: Browse... Clear

Advanced

< Back Next > Finish Cancel

Name: É o nome do projeto no contexto da Workspace. Pode ser o mesmo do Artifact Id. Não precisa ser único

Description: Descrição do projeto

Artifact Id: É o nome do Projeto sem a versão. Deve ser único

Version: Versão da distribuição

Group Id: Identifica exclusivamente o seu projeto em todos os projetos. Um ID de grupo deve seguir as regras de nome de pacote do Java. Isso significa que começa com um nome de domínio invertido que você controla.

Packaging: Define o formato de distribuição:
Jar – Desktop
War – Web
Pom - Pacote

Estrutura do Projeto

src/main/java	Código fonte da aplicação (no exemplo, código java)
src/main/resources	Recursos da aplicação (imagens, sons, etc.)
src/test/java	Código de teste (no exemplo, testes junit)
src/test/resources	Recursos de teste
src/site	Site do projeto
target	Diretório com arquivos gerados pelo processo de build
LICENSE.txt	Licença do projeto
README.txt	Visão geral do projeto
pom.xml	Descritor Maven do projeto

Pom.xml

```
    https://maven.apache.org/xsd/maven-4.0.0.xsd (xsi:schemaLocation)
10 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3
2     <modelVersion>4.0.0</modelVersion>
3     <groupId>br.edu.fateczl.AulaViewMaven</groupId>
4     <artifactId>AulaViewMaven</artifactId>
5     <version>1.0</version>
6     <packaging>war</packaging>
7     <name>AulaViewMaven</name>
8     <description>AulaViewMaven</description>
9
10    <properties>
11        <java.version>11</java.version>
12        <maven.compiler.source>${java.version}</maven.compiler.source>
13        <maven.compiler.target>${java.version}</maven.compiler.target>
14    </properties>
15
16    |
17 </project>
```

Project Facets

Deve ser ajustado com o web.xml (Generate Deployment Descriptor Stub)

Properties for AulaViewMaven

Project Facets

Configuration: <custom> Save As... Delete

Project Facet	Version
<input type="checkbox"/> Application Client module	6.0
<input type="checkbox"/> Axis2 Web Services	
<input type="checkbox"/> CXF 2.x Web Services	1.0
<input checked="" type="checkbox"/> Dynamic Web Module	5.0
<input type="checkbox"/> EAR	6.0
<input type="checkbox"/> EJB Module	3.1
<input type="checkbox"/> EJBDoclet (XDoclet)	1.2.3
<input checked="" type="checkbox"/> Java	11
<input checked="" type="checkbox"/> JavaScript	1.0
<input type="checkbox"/> JavaServer Faces	2.3
<input type="checkbox"/> JAX-RS (REST Web Services)	1.1
<input type="checkbox"/> JAXB	2.2
<input type="checkbox"/> JCA Module	1.6
<input type="checkbox"/> JPA	2.2
<input type="checkbox"/> Static Web Module	
<input type="checkbox"/> Utility Module	
<input type="checkbox"/> Web Fragment Module	3.0
<input type="checkbox"/> WebDoclet (XDoclet)	1.2.3

Details Runtimes

Java 11

Adds support for writing applications using Java programming language.

Revert Apply

Apply and Close Cancel

Project Explorer context menu:

- New
- Go Into
- Show In (Alt+Shift+W)
- Show in Local Terminal
- Copy (Ctrl+C)
- Copy Qualified Name
- Paste (Ctrl+V)
- Delete (Delete)
- Build Path
- Source (Alt+Shift+S)
- Refactor (Alt+Shift+T)
- Import
- Export
- Refresh (F5)
- Close Project
- Close Unrelated Projects
- Coverage As
- Run As
- Debug As
- Profile As
- Restore from Local History...
- Java EE Tools
- Maven
- Team
- Compare With
- Configure
- Source
- Validate
- Properties (Alt+Enter)
- Update EAR Libraries
- Generate Deployment Descriptor Stub

```
3 <groupId>br.edu.fate
4 <artifactId>AulaView
5 <version>1.0</versio
6 <packaging>war</pack
7 <name>AulaViewMaven<
8 <description>AulaVie
9
10 <properties>
11   <java.version>11.
12   <maven.compiler.
13   <maven.compiler.
14 </properties>
15
16
17 </project>
```

Overview Dependencies Dependency Hie

Markers Properties Servers

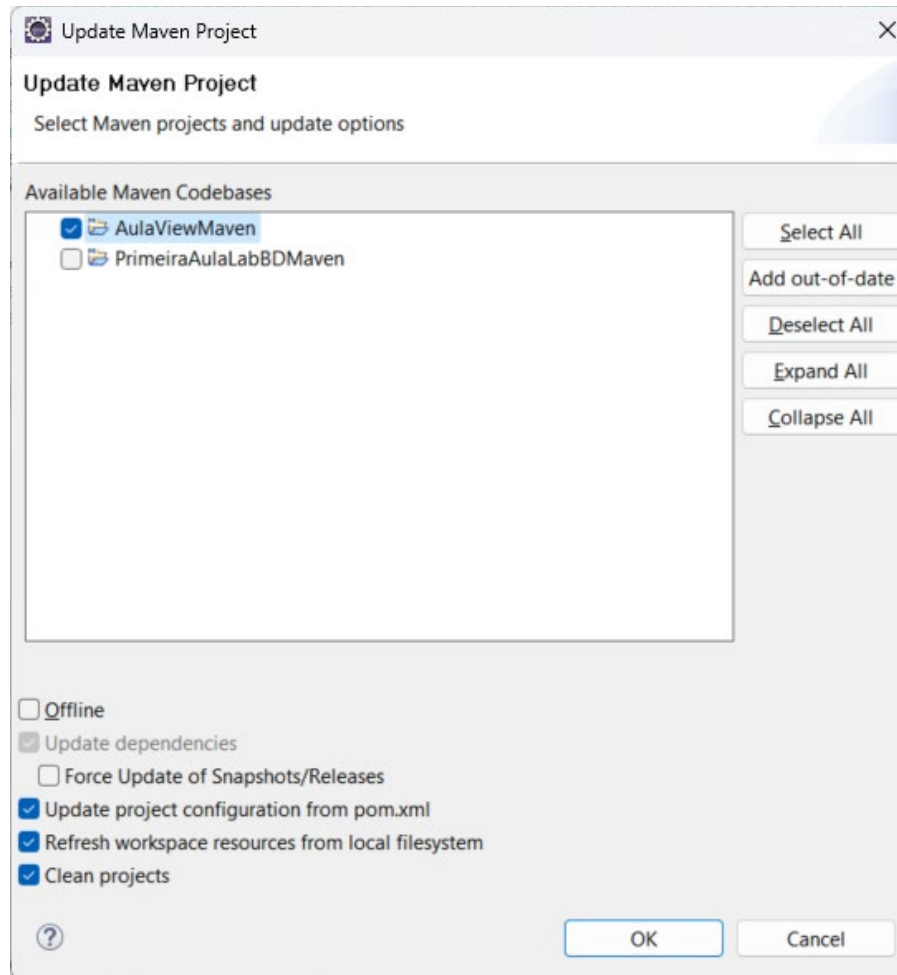
4 errors, 18 warnings, 10 others

Description

> JSP Problem (14 items)

Maven Update

Atualiza o projeto após modificações (no Pom.xml, por exemplo)



Gestão de Dependências

- Problema: “Como lidar com situações onde o projeto depende de bibliotecas externas?”
- Maven permite definir dependências para um projeto
 - As dependências são definidas no pom.xml
 - O Maven calcula as dependências considerando transitividade
 - As dependências são baixadas de repositórios centrais por demanda

Maven Repository

Repositório de bibliotecas para os projetos

The screenshot displays the Maven Repository website interface. At the top, there is a search bar with the text "Search for groups, artifacts, categories" and a "Search" button. To the right of the search bar are links for "Categories", "Popular", and "Contact Us".

On the left side, there is a sidebar with a blue header "MVN REPOSITORY" and a search bar. Below the header, there is a section "Indexed Artifacts (32.6M)" with a line graph showing the number of artifacts indexed over time from 2006 to 2018. Below the graph is a section "Popular Categories" with a list of categories including "Testing Frameworks & Tools", "Android Packages", "Logging Frameworks", "Java Specifications", "JSON Libraries", "Core Utilities", "JVM Languages", "Mocking", "Language Runtime", "Web Assets", "Annotation Libraries", "Logging Bridges", "HTTP Clients", "Dependency Injection", "XML Processing", "Web Frameworks", "I/O Utilities", "Defect Detection Metadata", "Configuration Libraries", "Code Generators", "OSGi Utilities", and "Android Platform".

The main content area is titled "What's New in Maven" and displays a list of artifacts. Each artifact entry includes a Kermit logo, the artifact name, version, and release date. The artifacts listed are:

- Kermit** (co.touchlab » kermit » 2.0.0-RC2) with 30 usages and last release on Mar 5, 2023.
- Kermit** (co.touchlab » kermit-iosx64 » 2.0.0-RC2) with 24 usages and last release on Mar 5, 2023.
- Kermit** (co.touchlab » kermit-iosarm64 » 2.0.0-RC2) with 23 usages and last release on Mar 5, 2023.
- Kermit** (co.touchlab » kermit-android » 2.0.0-RC2) with 20 usages and last release on Mar 5, 2023.
- Kermit** (co.touchlab » kermit-jvm » 2.0.0-RC2) with 19 usages and last release on Mar 5, 2023.
- Kermit** (co.touchlab » kermit-iOSSimulatorarm64 » 2.0.0-RC2) with 17 usages and last release on Mar 5, 2023.

On the right side, there is a section "Indexed Repositories (1886)" with a list of repositories including "Central", "Atlassian", "Sonatype", "Hortonworks", "Spring Plugins", "Spring Lib M", "JCenter", "JBossEA", "Atlassian Public", and "BeDataDriven". Below this is a section "Popular Tags" with a list of tags including "aar", "amazon", "android", "apache", "api", "application", "arm", "assets", "atlassian", "aws", "build", "build-system", "client", "clojure", "cloud", "config", "cran", "data", "database", "eclipse", "example", "extension", "github", "gradle", "groovy", "http", "io", "jboss", "kotlin", "library", "logging", "maven", "module", "npm", "persistence", "platform", "plugin", "rest", and "riano".

Pom.xml com Dependências

```
<dependencies>
  <!-- https://mvnrepository.com/artifact/jakarta.servlet/jakarta.servlet-api >
  <dependency>
    <groupId>jakarta.servlet</groupId>
    <artifactId>jakarta.servlet-api</artifactId>
    <version>6.0.0</version>
    <scope>provided</scope>
  </dependency>
  <!-- https://mvnrepository.com/artifact/jakarta.servlet.jsp/jakarta.servlet.jsp-api >
  <dependency>
    <groupId>jakarta.servlet.jsp</groupId>
    <artifactId>jakarta.servlet.jsp-api</artifactId>
    <version>3.1.1</version>
    <scope>provided</scope>
  </dependency>
  <!-- https://mvnrepository.com/artifact/jakarta.servlet.jsp.jstl/jakarta.servlet.jsp.jstl-api >
  <dependency>
    <groupId>jakarta.servlet.jsp.jstl</groupId>
    <artifactId>jakarta.servlet.jsp.jstl-api</artifactId>
    <version>3.0.0</version>
  </dependency>
  <!-- https://mvnrepository.com/artifact/org.glassfish.web/jakarta.servlet.jsp.jstl >
  <dependency>
    <groupId>org.glassfish.web</groupId>
    <artifactId>jakarta.servlet.jsp.jstl</artifactId>
    <version>3.0.0</version>
  </dependency>
  <!-- https://mvnrepository.com/artifact/jakarta.el/jakarta.el-api >
  <dependency>
    <groupId>jakarta.el</groupId>
    <artifactId>jakarta.el-api</artifactId>
    <version>5.0.1</version>
  </dependency>
  <!-- https://mvnrepository.com/artifact/jakarta.annotation/jakarta.annotation-api >
  <dependency>
    <groupId>jakarta.annotation</groupId>
    <artifactId>jakarta.annotation-api</artifactId>
    <version>2.1.1</version>
  </dependency>
  <!-- https://mvnrepository.com/artifact/jakarta.xml.bind/jakarta.xml.bind-api >
  <dependency>
    <groupId>jakarta.xml.bind</groupId>
    <artifactId>jakarta.xml.bind-api</artifactId>
    <version>4.0.0</version>
  </dependency>
  <!-- https://mvnrepository.com/artifact/net.sourceforge.jtds/jtds >
  <dependency>
    <groupId>net.sourceforge.jtds</groupId>
    <artifactId>jtds</artifactId>
    <version>1.3.1</version>
  </dependency>
</dependencies>
```

Dependencies scope

- **Compile:** Este é o escopo padrão, usado se nenhum for especificado. As dependências de compilação estão disponíveis em todos os classpaths de um projeto. Além disso, essas dependências são propagadas para projetos dependentes.
- **Provided:** Semelhante à compilação, mas indica que você espera que o JDK ou um contêiner forneça a dependência em tempo de execução. Por exemplo, ao construir um aplicativo da web para o Java Enterprise Edition, você configuraria a dependência na API do Servlet e nas APIs Java EE relacionadas para o escopo fornecido porque o contêiner da web fornece essas classes.
- **Runtime:** Esse escopo indica que a dependência não é necessária para compilação, mas para execução. O Maven inclui uma dependência com esse escopo no tempo de execução e nos classpaths de teste, mas não no classpath de compilação.
- **Test:** Este escopo indica que a dependência não é necessária para o uso normal do aplicativo e está disponível apenas para as fases de compilação e execução do teste.
- **System:** Este escopo é semelhante ao fornecido, exceto que você deve fornecer o JAR que o contém explicitamente. O artefato está sempre disponível e não é pesquisado em um repositório.
- **Import:** Este escopo é suportado apenas em uma dependência do tipo pom na seção <dependencyManagement>. Ele indica que a dependência deve ser substituída pela lista efetiva de dependências na seção <dependencyManagement> do POM especificado.

Gestão do Projeto de Construção

- **validate:** verifica se o projeto está correto e se os dados estão disponíveis
- **compile:** compila o código do projeto
- **test:** executa testes de unidade
- **package:** empacota o código compilado em um formato apropriado (ex.: jar)
- **integration-test:** implanta o pacote em um ambiente apropriado e executa testes de integração
- **verify:** executa verificações de qualidade sobre o pacote
- **install:** instala o pacote no repositório local
- **deploy:** disponibiliza o pacote em um repositório remoto
- **clean:** remove artefatos criados por processos anteriores de construção
- **site:** gera o site do projeto

Geração do Site do Projeto

- Problema: “Como criar um site com dados técnicos sobre o projeto?”
- Maven permite gerar automaticamente um site com os dados principais do projeto
 - Equipe
 - Bibliotecas utilizadas
 - Resultados dos testes
 - Repositório de código fonte
 - Repositório de solicitações de modificação