

# POM, Dependências e Repositórios

## O POM

### Entendendo o POM (Project Object Model)

- Unidade fundamental de trabalho do Apache Maven.
- Formato XML.
- Detalha o projeto, na raiz do seu projeto.
- Detalha como construir o projeto.
- Maven sempre procura pelo pom.xml para realizar sua execução

### Mais detalhes pom.xml

- Nome do projeto
- Dependências
- Módulos
- Configurações de build
- Detalhes do projeto (nome, descrição, licença, url)
- Configurações de ambiente (repositórios, tracking, profiles)
- Exemplo

### Pom.xml básico

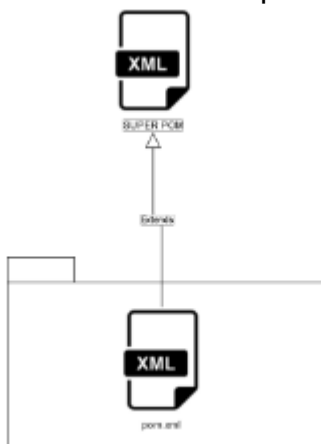
Para funcionar precisamos detalhar apenas essas informações.

```
<project>
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.mycompany.app</groupId>
  <artifactId>my-app</artifactId>
  <version>1</version>
</project>
```

## O Super POM

O maven trabalha com o conceito de Herança, que se estende o pom.

Se declaramos o pom apenas com as informações básicas, ele irá pegar as outras informações necessárias do Super POM.



<https://maven.apache.org/ref/3.0.4/maven-model-builder/super-pom.html>

## Repositórios

### O que são repositórios?

- São locais onde podemos encontrar plugins e bibliotecas que o Maven provê.

### 2 Tipos

#### Repositório remoto

- É o local central utilizado pelo Maven para buscar os artefatos.
- Configurado automaticamente pelo Super POM para utilizar o Maven Central

## Configuração

- Via pom.xml do projeto

```
<repositories>
  <repository>
    <id>central</id>
    <name>Central Repository</name>
    <url>http://repo.maven.apache.org/maven2</url>
    <layout>default</layout>
    <snapshots>
      <enabled>>false</enabled>
    </snapshots>
  </repository>
</repositories>
```

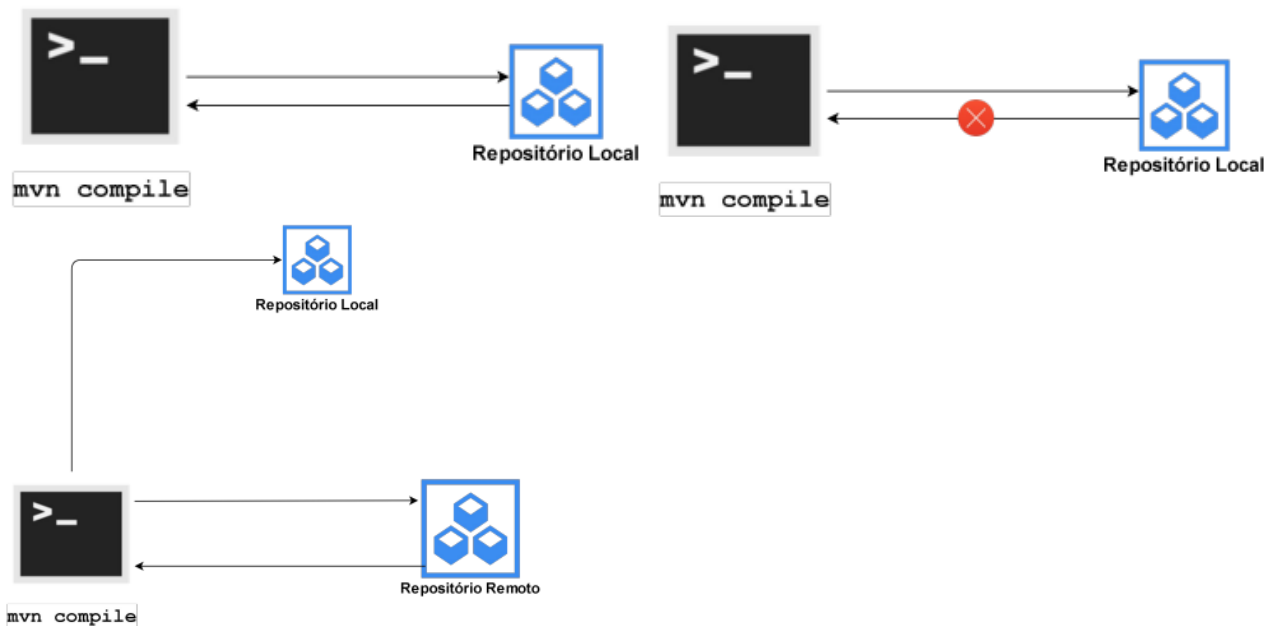
```
<pluginRepositories>
  <pluginRepository>
    <id>central</id>
    <name>Central Repository</name>
    <url>http://repo.maven.apache.org/maven2</url>
    <layout>default</layout>
    <snapshots>
      <enabled>>false</enabled>
    </snapshots>
    <releases>
      <updatePolicy>never</updatePolicy>
    </releases>
  </pluginRepository>
</pluginRepositories>
```

- Via settings.xml
  - Localização: pasta\_apache\_maven/conf/settings.xml

## Repositório local

- É o repositório na máquina utilizado pelo Maven para buscar os artefatos.
- Estratégia de caching.
- Localizações:
  - Windows: %USERPROFILE%\m2\repository
  - Linux: \$HOME/.m2/repository

## Como isso funciona?



## Como adicionar dependências

### Configurando

```
...
<dependencies>
  <dependency>
    <groupId>one.digitalinnovation</groupId>
    <artifactId>app-spring</artifactId>
    <version>1.0.0</version>
  </dependency>
...

```


## Propriedades

- groupId: É como se fosse o id da organização. Segue as regras de nomes de pacote Java
- artifactId: Nome do projeto em si
- Version: Número da versão que será utilizada



## Passo a Passo

1. <https://mvnrepository.com/>
2. Seleccione o Repository e copie o a tag do mesmo.

 **Hibernate Validator Engine » 8.0.0.CR2**  
Hibernate's Jakarta Bean Validation reference implementation.

License	Apache 2.0
Categories	Validation Frameworks
Tags	hibernate persistence validation
Date	Aug 04, 2022
Files	<a href="#">pom (14 KB)</a> <a href="#">jar (1.3 MB)</a> <a href="#">View All</a>
Repositories	Central
Ranking	#240 in MvnRepository (See Top Artifacts) #3 in Validation Frameworks
Used By	1,757 artifacts

Maven | **Gradle** | Gradle (Short) | Gradle (Kotlin) | SBT | Ivy | Grape | Leiningen | Buildr

```
<!-- https://mvnrepository.com/artifact/org.hibernate.validator/hibernate-validator -->
<dependency>
  <groupId>org.hibernate.validator</groupId>
  <artifactId>hibernate-validator</artifactId>
  <version>8.0.0.CR2</version>
</dependency>
```

3. Entre no arquivo pom e cole na área de dependências.

```
<groupId>one.digitalinnovation</groupId>
<artifactId>quick-start-maven</artifactId>
<version>version' 1.0-SNAPSHOT</version>

<name>quick-start-maven</name>
<!-- FIXME change it to the project's website -->
<url>http://www.example.com</url>

<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <maven.compiler.source>1.7</maven.compiler.source>
  <maven.compiler.target>1.7</maven.compiler.target>
</properties>

<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.11</version>
    <scope>test</scope>
  </dependency>
</dependencies>

<!-- https://mvnrepository.com/artifact/org.hibernate.validator/hibernate-validator -->
<dependency>
  <groupId>org.hibernate.validator</groupId>
  <artifactId>hibernate-validator</artifactId>
  <version>8.0.0.CR2</version>
</dependency>
```

4. No cmd executamos o comando mvn compile.