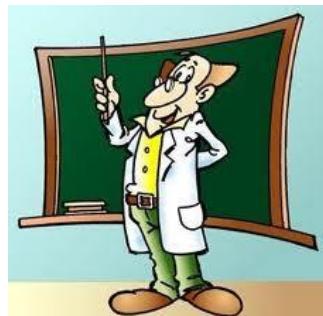




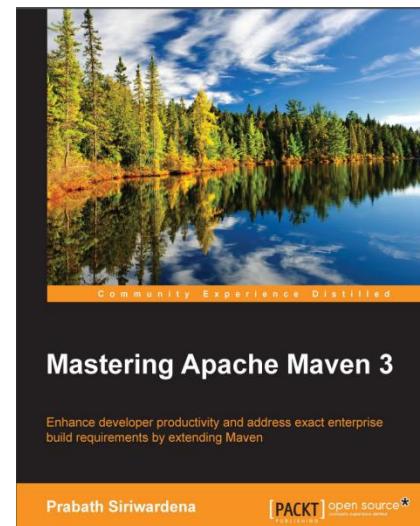
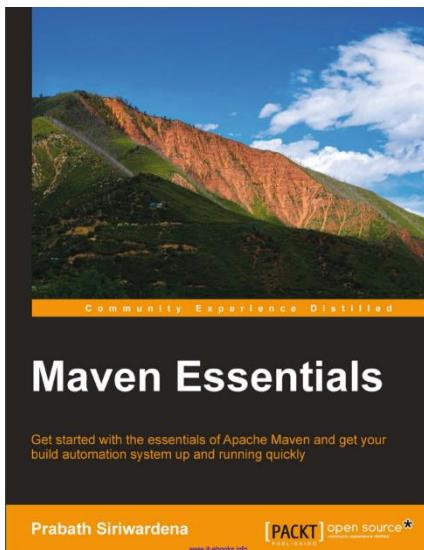
## Maven – Workshop



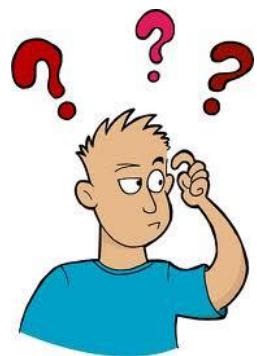
Prof. Aparecido V. de Freitas  
Doutor em Engenharia  
da Computação pela EPUSP  
[aparecidovfreitas@gmail.com](mailto:aparecidovfreitas@gmail.com)



# Bibliografia



# O que é Maven



# Maven



- Ferramenta de automação de build
- Gerencia o processo de desenvolvimento de produtos (artefatos) em Java (preferencialmente)
- Abordagem declarativa (diferentemente do Ant)
- Convenção sobre Configuração
- Desenvolvido pelo grupo Apache (suporte do projeto Avalon)



# Porque usar Maven ?

- Padronização do processo de desenvolvimento
- Gerenciamento de dependências (bibliotecas)
- Compartilhamento de componentes
  - Diferentemente de compartilhamento de código/build
- Extensível através de plugins
- Testes facilitados
- Documentação facilitada
- Fácil integração com ferramentas de integração contínua e monitoramento da qualidade do código
- Integração com IDE

# Instalação

- Baixar distribuição
  - <http://maven.apache.org/download.html>
- Descompactar
- Configurar variáveis de ambiente
  - Windows
    - M2\_HOME=C:\apache-maven-3.0.4 (opcional)
    - JAVA\_HOME=<pasta de instalação do JDK>
    - PATH=%M2\_HOME%\bin;%JAVA\_HOME%\bin;%PATH%
- Testar instalação
  - PROMPT> mvn -version

```
Apache Maven 3.0.4 (r1232337; 2012-01-17 06:44:56-0200)
Maven home: C:\Program Files\Java\apache-maven-3.0.4
Java version: 1.7.0_03, vendor: Oracle Corporation
Java home: C:\Program Files\Java\jdk1.7.0_03-x64\jre
Default locale: pt_BR, platform encoding: Cp1252
OS name: "windows 7", version: "6.1", arch: "amd64", family: "windows"
```



# <http://maven.apache.org/>

The screenshot shows a web browser window with multiple tabs open. The active tab displays the Apache Maven Project homepage at <https://maven.apache.org/index.html>. The page features the Apache logo (feather) and the text "Apache Maven Project" with the URL "http://maven.apache.org/". On the right, a large "Maven™" logo is displayed. The main content area is titled "Welcome to Apache Maven". It explains what Maven is and how it can manage a project's build, reporting, and documentation from a central piece of information. It also provides links for "Download" and "Get Sources" and indicates the last published date as "Last Published: 2018-08-01". A sidebar on the left contains links for "Welcome", "License", "ABOUT MAVEN", "What is Maven?", "Features", "Download", "Use", "Release Notes", "DOCUMENTATION", "Maven Plugins", "Index (category)", "User Centre", "Plugin Developer Centre", "Maven Central Repository", and "Maven Developers".



# Download

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "Maven – Download Apache Maven" and displays the Apache Maven Project website at <https://maven.apache.org/download.cgi>. The page content includes the Apache logo, the Maven logo, and sections for "Downloading Apache Maven 3.5.4" and "System Requirements".

**Apache Maven Project**  
<http://maven.apache.org>

Apache / Maven / Download Apache Maven

**Download** | Get Sources | Last Published: 2018-08-01

**Downloading Apache Maven 3.5.4**

Apache Maven 3.5.4 is the latest release and recommended version for all users.

The currently selected download mirror is <http://mirror.nbtelecom.com.br/apache/>. If you encounter a problem with this mirror, please select another mirror. If all mirrors are failing, there are *backup* mirrors (at the end of the mirrors list) that should be available. You may also consult the [complete list of mirrors](#).

Other mirrors: <http://ftp.unicamp.br/pub/apac> ▾ [Change](#)

**System Requirements**

<b>Java Development Kit (JDK)</b>	Maven 3.3+ require JDK 1.7 or above to execute - they still allows you to build against 1.3 and other JDK versions by <a href="#">Using Toolchains</a>
<b>Memory</b>	No minimum requirement
<b>Disk</b>	Approximately 10MB is required for the Maven installation itself. In addition to that, additional disk space will be used for your local Maven repository. The size of your local repository will vary depending on usage but expect at least 500MB.



## Processo de Instalação

- ✓ **Maven** é uma Java Tool, assim você deve ter Java instalado em seu computador;
- ✓ Assegure que a variável de ambiente **JAVA\_HOME** está definida e apontada para a instalação JDK;
- ✓ Extrair a distribuição em qualquer diretório do sistema;

```
unzip apache-maven-3.5.4-bin.zip
```

OR

```
tar xzvf apache-maven-3.5.4-bin.tar.gz
```

- ✓ Adicione o diretório bin do diretório onde o Maven foi instalado na variável de ambiente PATH;
- ✓ Adicione a variável de ambiente **MAVEN\_HOME** apontando para o diretório de instalação do MAVEN.
- ✓ Confirme com **mvn –version** no prompt de comandos. O resultado deve ser similar a:

```
Apache Maven 3.5.4 (1edded0938998edf8bf061f1ceb3cfdeccf443fe; 2018-06-17T20:33:14+02:00)
Maven home: /opt/apache-maven-3.5.4
Java version: 1.8.0_45, vendor: Oracle Corporation
Java home: /Library/Java/JavaVirtualMachines/jdk1.8.0_45.jdk/Contents/Home/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "mac os x", version: "10.8.5", arch: "x86_64", family: "mac"
```



## **mvn -version**

```
C:\> Command Prompt  
C:\Users\VB_Aparecido>mvn -version  
Apache Maven 3.5.4 (1edded0938998edf8bf061f1ceb3cfdeccf443fe; 2018-06-17T11:33:14-07:00)  
Maven home: C:\apache-maven-3.5.4\bin\..  
Java version: 1.8.0_171, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk1.8.0_171\jre  
Default locale: en_US, platform encoding: Cp1252  
OS name: "windows 7", version: "6.1", arch: "amd64", family: "windows"  
C:\Users\VB_Aparecido>
```

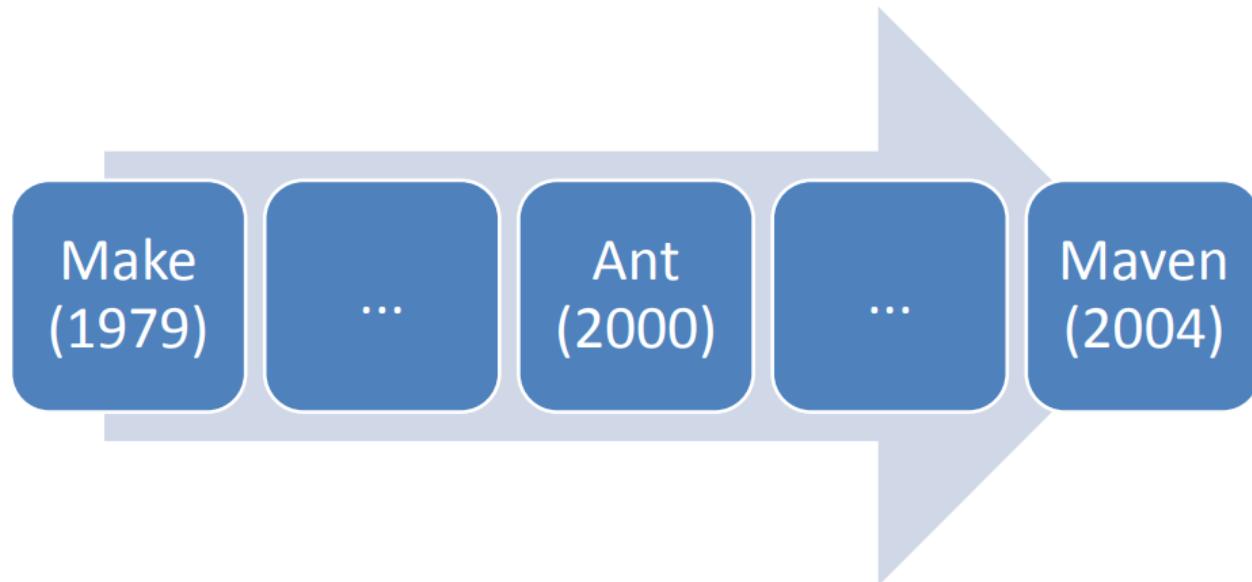


# Funcionalidades

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



# Gestão do Processo de Build



# ANT x Maven

- Ant
  - Fortemente baseada em configuração
  - Procedural
  - Processo de construção e ferramentas utilizadas ficam misturados
- Maven
  - Convenção sobre Configuração
  - Declarativo
  - Existência de um processo padrão
  - Ferramentas padrões dependentes do tipo de empacotamento



# 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



# Gestão 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





# Arquivo pom.xml

```
<project ...>
  <modelVersion>4.0.0</modelVersion>
  <groupId>uff</groupId>
  <artifactId>teste</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>
  <name>teste</name>
  <url>http://maven.apache.org</url>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```



# Geração da Estrutura do Projeto

(src/main/java/uff/App.java)

```
package uff;

/**
 * Hello world!
 *
 */
public class App
{
    public static void main( String[] args )
    {
        System.out.println( "Hello World!" );
    }
}
```



# Geração da Estrutura do Projeto

## (src/test/java/uff/AppTest.java)

```
package uff;

import junit.framework.Test;
import junit.framework.TestCase;
import junit.framework.TestSuite;

/**
 * Unit test for simple App.
 */
public class AppTest
    extends TestCase
{
    (...)

    /**
     * Rigourous Test :-)
     */
    public void testApp()
    {
        assertTrue( true );
    }
}
```



# Convenção da 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	Descriptor Maven do projeto





# Gestão do Processo de Build

- Problema: “Como posso construir o projeto?”
- Maven permite utilizar um **processo padrão** para construção do projeto
- Processo composto das seguintes fases (entre outras):
  - Compilação
  - Testes
  - Empacotamento

# Geração do Projeto Maven

```
$ mvn archetype:generate  
[80: remote -> maven-archetype-quickstart (An archetype  
which contains a sample Maven project.)]  
...  
Define value for property 'groupId': : uff  
Define value for property 'artifactId': : teste  
Define value for property 'version': 1.0-SNAPSHOT:  
Define value for property 'package': uff:  
...
```

## O que foi gerado?

```
pom.xml  
src/main/java/uff/App.java  
src/test/java/uff/AppTest.java
```



# Compilação

```
$ mvn compile
[INFO] Scanning for projects...
[INFO] -----
[INFO] Building teste
[INFO]   task-segment: [compile]
[INFO] -----
[INFO] [compiler:compile {execution: default-compile}]
[INFO] Compiling 1 source file to
  C:\Users\leomurta\workspace\teste\target\classes
[INFO] -----
[INFO] BUILD SUCCESSFUL
[INFO] -----
[INFO] Total time: 3 seconds
[INFO] Finished at: Sat Oct 16 11:47:39 BRT 2010
[INFO] Final Memory: 15M/130M
[INFO] -----
```



# Teste

```
$ mvn test  
...  
[INFO] [compiler:compile {execution: default-compile}]  
[INFO] Nothing to compile - all classes are up to date  
[INFO] [compiler:testCompile {execution: default-testCompile}]  
[INFO] Compiling 1 source file to target\test-classes  
[INFO] [surefire:test {execution: default-test}]  
[INFO] Surefire report directory:  
C:\Users\leomurta\workspace\teste\target\surefire-reports  
-----  
T E S T S  
-----  
Running uff.AppTest  
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.049  
sec  
...  
...
```



# Empacotamento

```
$ mvn package  
...  
[INFO] [compiler:compile {execution: default-compile}]  
[INFO] Nothing to compile - all classes are up to date  
[INFO] [compiler:testCompile {execution: default-testCompile}]  
[INFO] Nothing to compile - all classes are up to date  
[INFO] [surefire:test {execution: default-test}]  
[INFO] Surefire report directory: target\surefire-reports  
...  
[INFO] [jar:jar {execution: default-jar}]  
[INFO] Building jar: target\teste-1.0-SNAPSHOT.jar  
...
```



# Gestão do Processo de Build

## (fases mais comuns)

- **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





# Gestão de Dependências do Projeto

- 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



# Gestão de Dependências do Projeto

## (definindo dependência no pom.xml)

```
<project ...>
...
<dependencies>
    <dependency>
        <groupId>axis</groupId>
        <artifactId>axis</artifactId>
        <version>1.4</version>
        <scope>compile</scope>
    </dependency>
    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>3.8.1</version>
        <scope>test</scope>
    </dependency>
</dependencies>
</project>
```



# Repositório Local

- ✓ O Maven utiliza um diretório local para baixar os artefatos da internet. O diretório padrão fica dentro pasta do usuário, na pasta .m2. Um exemplo no Windows é c:\users\aparecido\.m2\repository.
- ✓ Entretanto, pode-se mudar esse diretório para junto de outros arquivos de desenvolvimento.
- ✓ Para isso, basta editar o settings.xml, movendo a tag <localRepository> para fora do comentário e adicionando o caminho, por exemplo:

```
<localRepository>c:\develop\apache-maven-3.5.4\repo</localRepository>
```

- ✓ Não se esqueça de criar o diretório especificado caso o mesmo não exista.



# Exemplo – Processo de Build



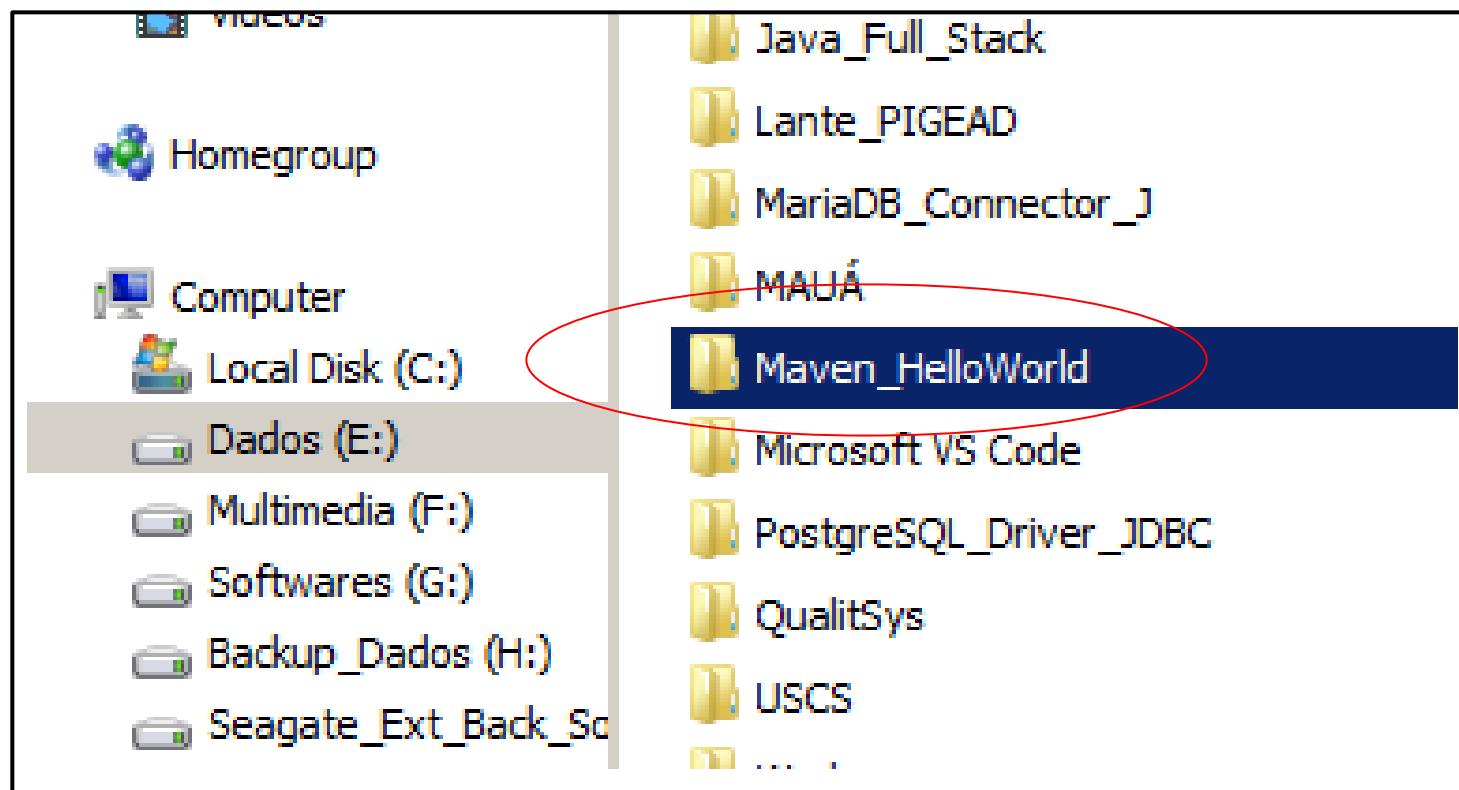
# Exemplo de Geração – Passo 1

1. Vamos criar uma aplicação “Hello World” a partir da linha de comandos do Maven;
2. A pasta do projeto será: E:\Maven\_HelloWorld
3. Estamos assumindo que o Maven está instalado e a variável de ambiente MAVEN\_HOME está definida e apontando para a pasta de instalação do Maven.



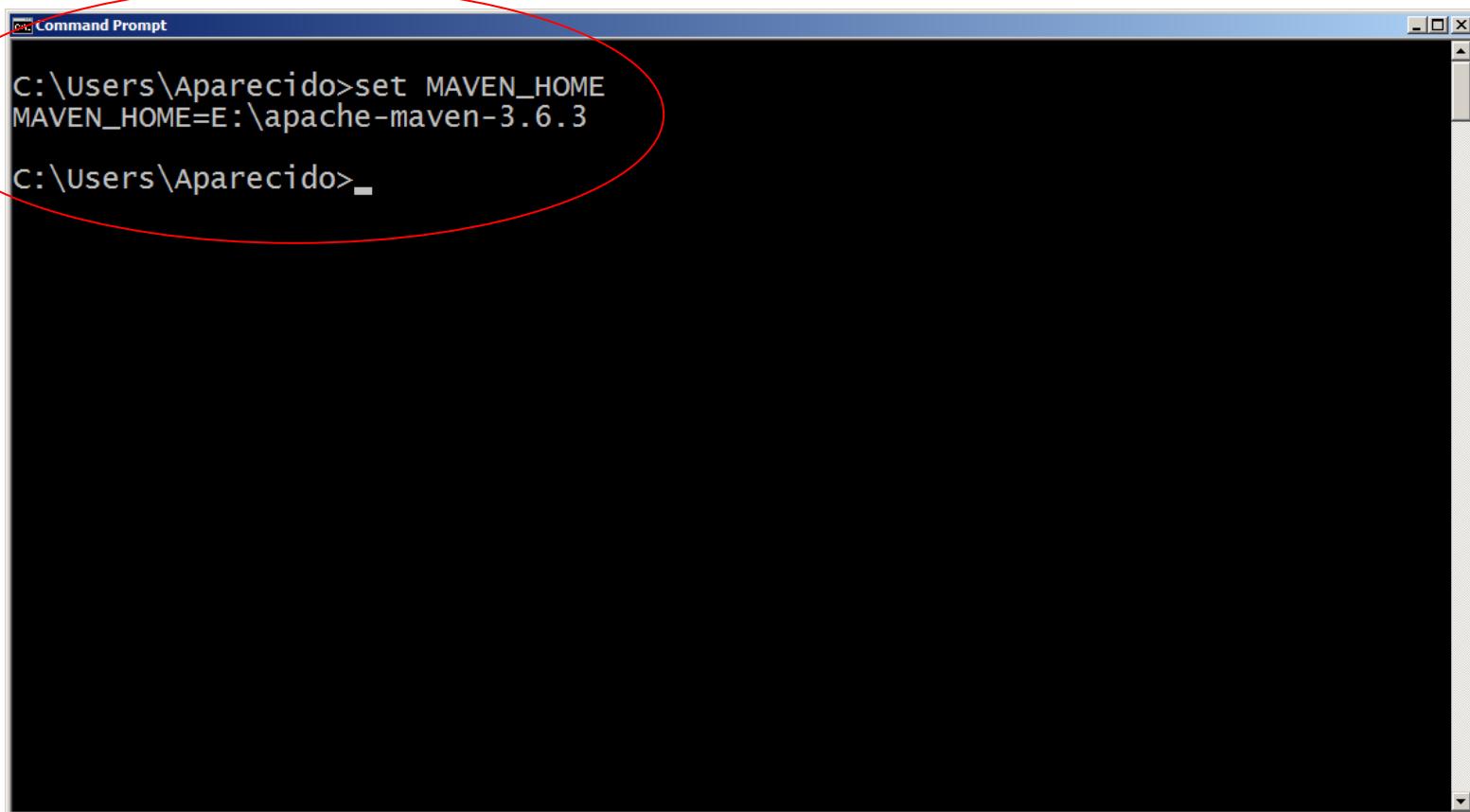
# Exemplo de Geração – passo 1

4. Verificação da pasta criada para o projeto:



# Exemplo de Geração – passo 1

5. Verificação da variável de ambiente do Maven (MAVEN\_HOME)



```
cmd: Command Prompt
C:\Users\Aparecido>set MAVEN_HOME
MAVEN_HOME=E:\apache-maven-3.6.3

C:\Users\Aparecido>_
```

A red oval highlights the command and its output.

# Exemplo de Geração – passo 2

1. Por padrão, o **Maven** baixa todos os módulos especificados no **POM.XML** na pasta: **\users\usuario\.m2\repository**;
2. Pode-se configurar o Maven para baixar as bibliotecas em alguma pasta que pode ser definida pelo usuário;
3. Para isso, basta editar o arquivo **settings.xml**, movendo a tag **<localRepository>** para fora do comentário e adicionando o caminho da nova pasta usada para o Repositório, por exemplo:

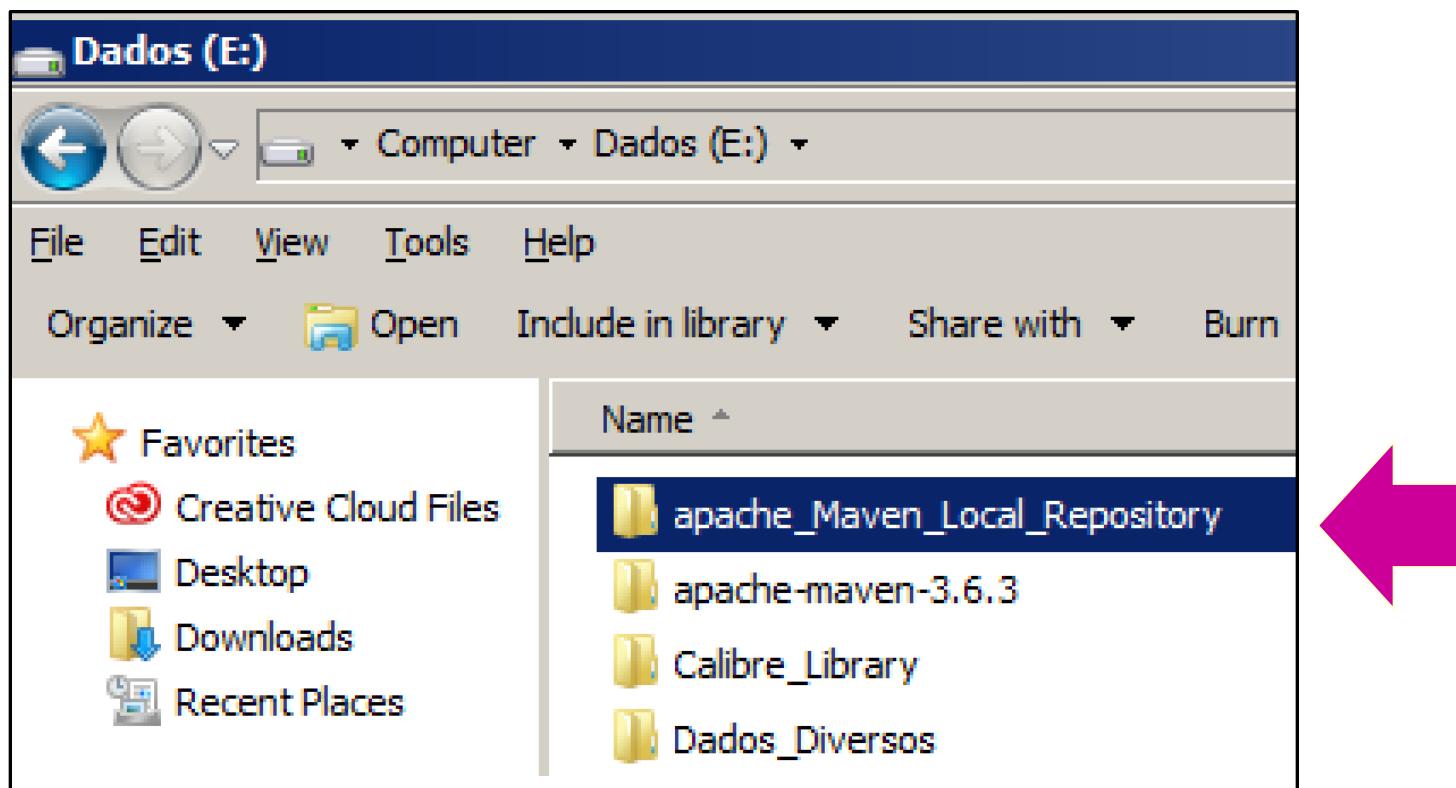


```
<E:\apache_Maven_Local_Repository</localRepository>
```



# Exemplo de Geração – passo 2

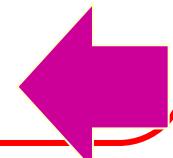
- Vamos então criar uma pasta para configurar o repositório local do Maven (C:\apache\_Maven\_Local\_Repository).



# Exemplo de Geração – passo 2

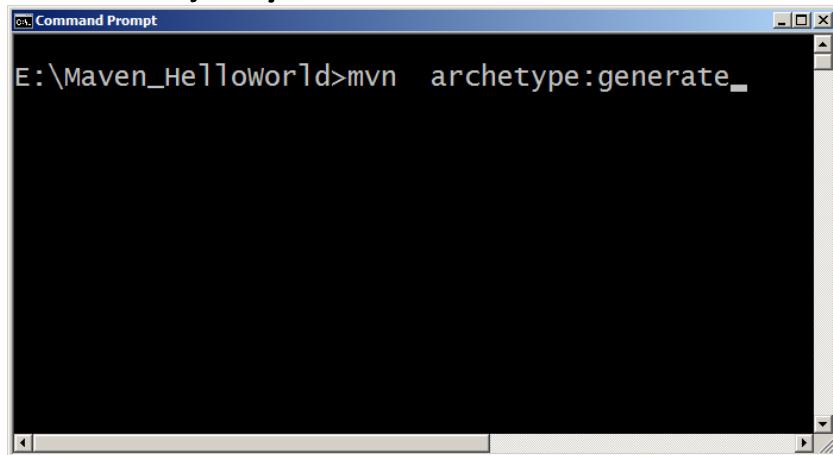
5. Na pasta **conf** do diretório de **instalação** do Maven, alterar o arquivo **settings.xml** informando a **pasta** que será usada como **Repositório Local**.

```
<!-- localRepository  
| The path to the local repository maven will use to store artifacts.  
|  
| Default: ${user.home}/.m2/repository  
<localRepository>/path/to/local/repo</localRepository>  
-->  
  
<localRepository>E:\apache_Maven_Local_Repository</localRepository>
```



# Exemplo de Geração – passo 3

1. Entre na pasta do projeto recentemente criada:



2. Vamos agora criar o nosso primeiro projeto **Maven** pelo terminal;  
Archetypes são templates de projetos pré-configurados.
3. Entre com o comando:

**\$ mvn archetype:generate**



# Exemplo de Geração – passo 3

4. Maven irá iniciar o processo de download.

```
Command Prompt - mvn archetype:generate
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/2.4/maven-install-plugin-2.4.jar (27 kB at 71 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-deploy-plugin/2.7/maven-deploy-plugin-2.7.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-deploy-plugin/2.7/maven-deploy-plugin-2.7.pom (5.6 kB at 16 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-deploy-plugin/2.7/maven-deploy-plugin-2.7.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-deploy-plugin/2.7/maven-deploy-plugin-2.7.jar (27 kB at 76 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-site-plugin/3.3/maven-site-plugin-3.3.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-site-plugin/3.3/maven-site-plugin-3.3.pom (21 kB at 54 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/24/maven-plugins-24.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/24/maven-plugins-24.pom (11 kB at 30 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/23/maven-parent-23.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/23/maven-parent-23.pom (33 kB at 91 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/apache/13/apache-13.pom
```



# Exemplo de Geração – passo 3

5. Há uma lista muito grande de archetypes previamente definidos no Maven; Teclar <enter> para aceitar o default (1593).

```
Command Prompt - mvn archetype:generate
2675: remote -> uk.ac.ebi.gxa:atlas-archetype (Archetype for generating a custom
Atlas webapp)
2676: remote -> uk.ac.gate:gate-plugin-archetype (Maven archetype to create a new
GATE plugin project.)
2677: remote -> uk.ac.gate:gate-pr-archetype (Maven archetype to create a new GA
TE plugin project including a sample PR class (an empty LanguageAnalyser).)
2678: remote -> uk.ac.nactem.argo:argo-analysis-engine-archetype (An archetype w
hich contains a sample Argo (UIMA) Analysis Engine)
2679: remote -> uk.ac.nactem.argo:argo-reader-archetype (An archetype which cont
ains a sample Argo (UIMA) Reader)
2680: remote -> uk.ac.rdg.resc:edal-ncwms-based-webapp (-)
2681: remote -> uk.co.nemstix:basic-javaee7-archetype (A basic Java EE7 Maven ar
chetype)
2682: remote -> uk.co.solong:angular-spring-archetype (So Long archetype for RES
Tful spring services with an AngularJS frontend. Includes debian deployment)
2683: remote -> us.fatehi:schemacrawler-archetype-maven-project (-)
2684: remote -> us.fatehi:schemacrawler-archetype-plugin-command (-)
2685: remote -> us.fatehi:schemacrawler-archetype-plugin-dbconnector (-)
2686: remote -> us.fatehi:schemacrawler-archetype-plugin-lint (-)
2687: remote -> ws.osiris:osiris-archetype (Maven Archetype for Osiris)
2688: remote -> xyz.luan.generator:xyz-gae-generator (-)
2689: remote -> xyz.luan.generator:xyz-generator (-)
2690: remote -> za.co.absa.hyperdrive:component-archetype (-)
Choose a number or apply filter (format: [groupId:]artifactId, case sensitive co
ntains): 1593:
```



# Exemplo de Geração – passo 3

## 6. Teclar <enter> para quickstart version (8)

```
Command Prompt - mvn archetype:generate
2680: remote -> uk.ac.rdg.resc:edal-ncwms-based-webapp (-)
2681: remote -> uk.co.nemstix:basic-javaee7-archetype (A basic Java EE7 Maven ar-
chetype)
2682: remote -> uk.co.solong:angular-spring-archetype (So Long archetype for RES-
Tful spring services with an AngularJS frontend. Includes debian deployment)
2683: remote -> us.fatehi:schemacrawler-archetype-maven-project (-)
2684: remote -> us.fatehi:schemacrawler-archetype-plugin-command (-)
2685: remote -> us.fatehi:schemacrawler-archetype-plugin-dbconnector (-)
2686: remote -> us.fatehi:schemacrawler-archetype-plugin-lint (-)
2687: remote -> ws.osiris:osiris-archetype (Maven Archetype for osiris)
2688: remote -> xyz.luan.generator:xyz-gae-generator (-)
2689: remote -> xyz.luan.generator:xyz-generator (-)
2690: remote -> za.co.absa.hyperdrive:component-archetype (-)
Choose a number or apply filter (format: [groupId:]artifactId, case sensitive co-
ntains): 1593:
Choose org.apache.maven.archetypes:maven-archetype-quickstart version:
1: 1.0-alpha-1
2: 1.0-alpha-2
3: 1.0-alpha-3
4: 1.0-alpha-4
5: 1.0
6: 1.1
7: 1.3
8: 1.4
Choose a number: 8:
```



# Exemplo de Geração – passo 3

7. Entre com o groupId, que representa a Organização (**br.uscs**)

```
Command Prompt - mvn archetype:generate
1: 1.0-alpha-1
2: 1.0-alpha-2
3: 1.0-alpha-3
4: 1.0-alpha-4
5: 1.0
6: 1.1
7: 1.3
8: 1.4
Choose a number: 8:
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.4/maven-archetype-quickstart-1.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.4/maven-archetype-quickstart-1.4.pom (1.6 kB at 2.0 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-bundles/1.4/maven-archetype-bundles-1.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-bundles/1.4/maven-archetype-bundles-1.4.pom (4.5 kB at 14 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.4/maven-archetype-quickstart-1.4.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.4/maven-archetype-quickstart-1.4.jar (7.1 kB at 21 kB/s)
Define value for property 'groupId': br.uscs
```



# Exemplo de Geração – passo 3

8. Entre com o artifactId, que representa a nossa aplicação. Entraremos com **MavenHelloWorld**.

```
C:\ Command Prompt - mvn archetype:generate
2: 1.0-alpha-2
3: 1.0-alpha-3
4: 1.0-alpha-4
5: 1.0
6: 1.1
7: 1.3
8: 1.4
Choose a number: 8:
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.4/maven-archetype-quickstart-1.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.4/maven-archetype-quickstart-1.4.pom (1.6 kB at 2.0 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-bundles/1.4/maven-archetype-bundles-1.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-bundles/1.4/maven-archetype-bundles-1.4.pom (4.5 kB at 14 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.4/maven-archetype-quickstart-1.4.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.4/maven-archetype-quickstart-1.4.jar (7.1 kB at 21 kB/s)
Define value for property 'groupId': br.uscs
Define value for property 'artifactId': MavenHelloWorld
```



# Exemplo de Geração – passo 3

9. Tecle <Enter> para aceitar a versão: 1.0-SNAPSHOT

```
Command Prompt - mvn archetype:generate
3: 1.0-alpha-3
4: 1.0-alpha-4
5: 1.0
6: 1.1
7: 1.3
8: 1.4
Choose a number: 8:
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.4/maven-archetype-quickstart-1.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.4/maven-archetype-quickstart-1.4.pom (1.6 kB at 2.0 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-bundles/1.4/maven-archetype-bundles-1.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-bundles/1.4/maven-archetype-bundles-1.4.pom (4.5 kB at 14 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.4/maven-archetype-quickstart-1.4.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.4/maven-archetype-quickstart-1.4.jar (7.1 kB at 21 kB/s)
Define value for property 'groupId': br.uscs
Define value for property 'artifactId': MavenHelloWorld
Define value for property 'version' 1.0-SNAPSHOT: :
```



# Exemplo de Geração – passo 3

10. Tecle <Enter> para aceitar o package definido (br.uscs)

```
Command Prompt - mvn archetype:generate
4: 1.0-alpha-4
5: 1.0
6: 1.1
7: 1.3
8: 1.4
Choose a number: 8:
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.4/maven-archetype-quickstart-1.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.4/maven-archetype-quickstart-1.4.pom (1.6 kB at 2.0 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-bundles/1.4/maven-archetype-bundles-1.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-bundles/1.4/maven-archetype-bundles-1.4.pom (4.5 kB at 14 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.4/maven-archetype-quickstart-1.4.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.4/maven-archetype-quickstart-1.4.jar (7.1 kB at 21 kB/s)
Define value for property 'groupId': br.uscs
Define value for property 'artifactId': MavenHelloWorld
Define value for property 'version' 1.0-SNAPSHOT: :
Define value for property 'package' br.uscs: :
```



# Exemplo de Geração – passo 3

11. Tecle <Enter> para aceitar as configurações (Y)

```
Command Prompt - mvn archetypegenerate
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.4/maven-archetype-quickstart-1.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.4/maven-archetype-quickstart-1.4.pom (1.6 kB at 2.0 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-bundles/1.4/maven-archetype-bundles-1.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-bundles/1.4/maven-archetype-bundles-1.4.pom (4.5 kB at 14 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.4/maven-archetype-quickstart-1.4.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.4/maven-archetype-quickstart-1.4.jar (7.1 kB at 21 kB/s)
Define value for property 'groupId': br.uscs
Define value for property 'artifactId': MavenHelloWorld
Define value for property 'version' 1.0-SNAPSHOT: :
Define value for property 'package' br.uscs: :
Confirm properties configuration:
groupId: br.uscs
artifactId: MavenHelloWorld
version: 1.0-SNAPSHOT
package: br.uscs
Y: :
```



# Exemplo de Geração – passo 3

## 12. Processo de Build concluído (Projeto Criado)



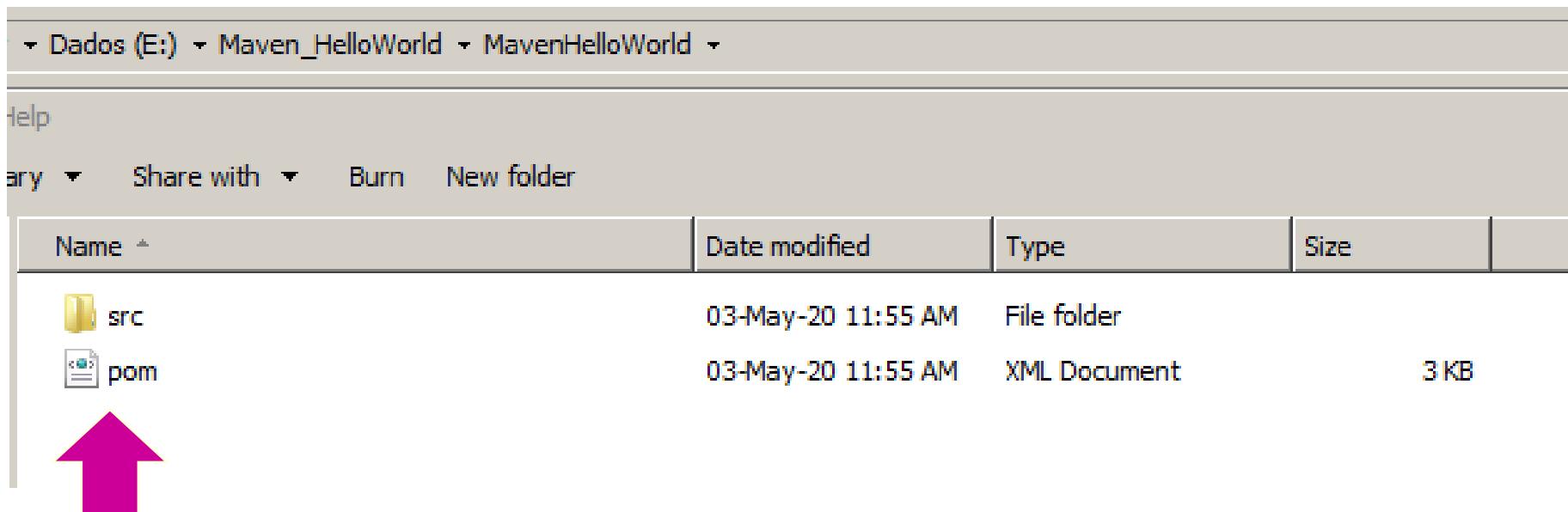
```
C:\ Command Prompt
[INFO] -----
[INFO] Using following parameters for creating project from Archetype: maven-archetype-quickstart:1.4
[INFO] -----
[INFO] Parameter: groupId, value: br.uscs
[INFO] Parameter: artifactId, value: MavenHelloWorld
[INFO] Parameter: version, value: 1.0-SNAPSHOT
[INFO] Parameter: package, value: br.uscs
[INFO] Parameter: packageInPathFormat, value: br/uscs
[INFO] Parameter: package, value: br.uscs
[INFO] Parameter: version, value: 1.0-SNAPSHOT
[INFO] Parameter: groupId, value: br.uscs
[INFO] Parameter: artifactId, value: MavenHelloWorld
[INFO] Project created from Archetype in dir: E:\Maven_Helloworld\MavenHelloWorld
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 39:41 min
[INFO] Finished at: 2020-05-03T11:55:24-07:00
[INFO] -----
```

E:\Maven\_Helloworld>



# Exemplo de Geração – passo 3

13. Verificando o projeto criado . O arquivo POM.XML também foi criado.



Name	Date modified	Type	Size
src	03-May-20 11:55 AM	File folder	
pom	03-May-20 11:55 AM	XML Document	3 KB



# Exemplo de Geração – passo 3

14. Consultando o arquivo POM.XML criado no projeto.



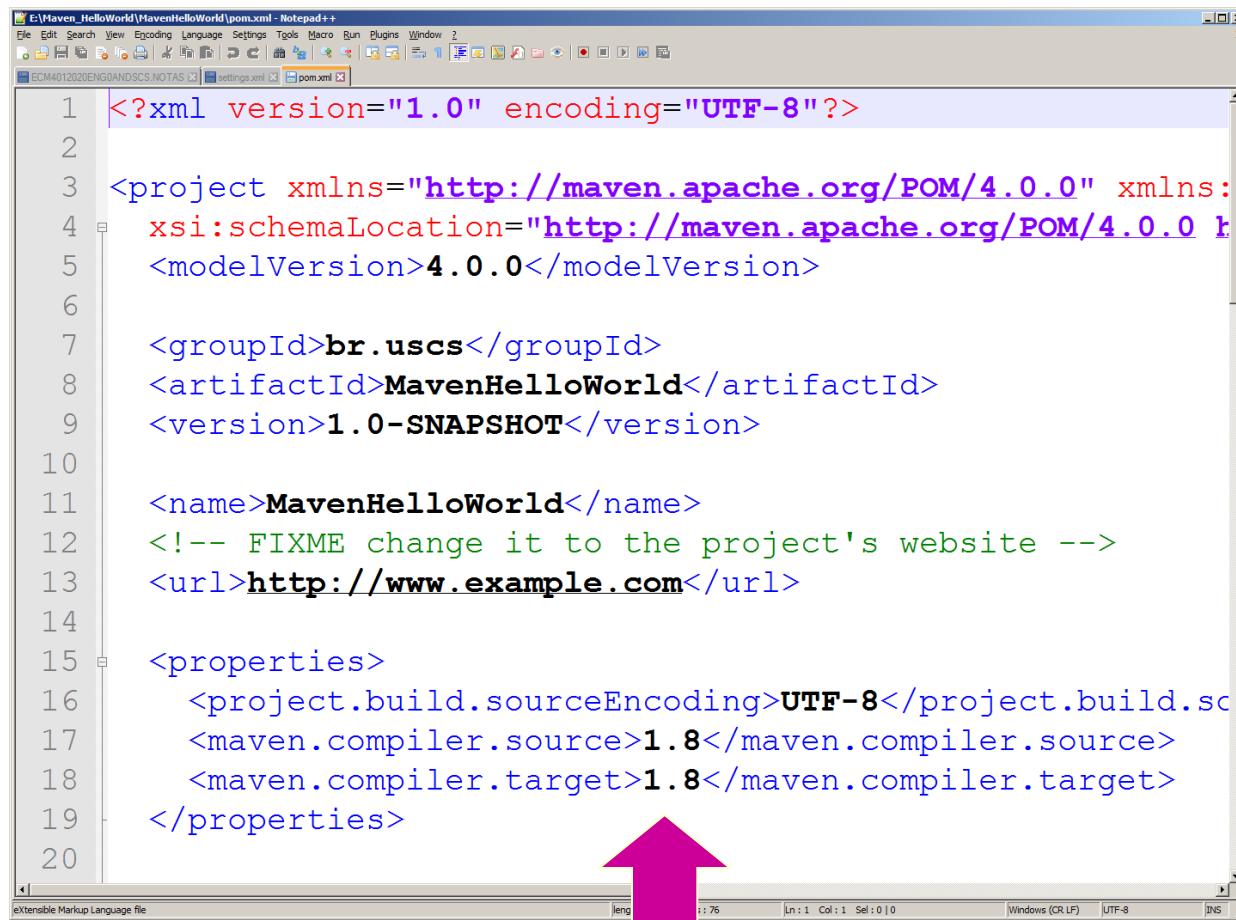
```
E: > Maven_HelloWorld > MavenHelloWorld > pom.xml
1   <?xml version="1.0" encoding="UTF-8"?>
2
3   <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"
5         modelVersion="4.0.0">
6
7     <groupId>br.uscs</groupId>
8     <artifactId>MavenHelloWorld</artifactId>
9     <version>1.0-SNAPSHOT</version>
10
11    <name>MavenHelloWorld</name>
12    <!-- FIXME change it to the project's website -->
13    <url>http://www.example.com</url>
14
15    <properties>
16        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
17        <maven.compiler.source>1.7</maven.compiler.source>
18        <maven.compiler.target>1.7</maven.compiler.target>
19    </properties>
20
21    <dependencies>
22        <dependency>
23            <groupId>junit</groupId>
```

File Edit Selection View Run Terminal Help pom.xml - Visual Studio Code



# Exemplo de Geração – passo 3

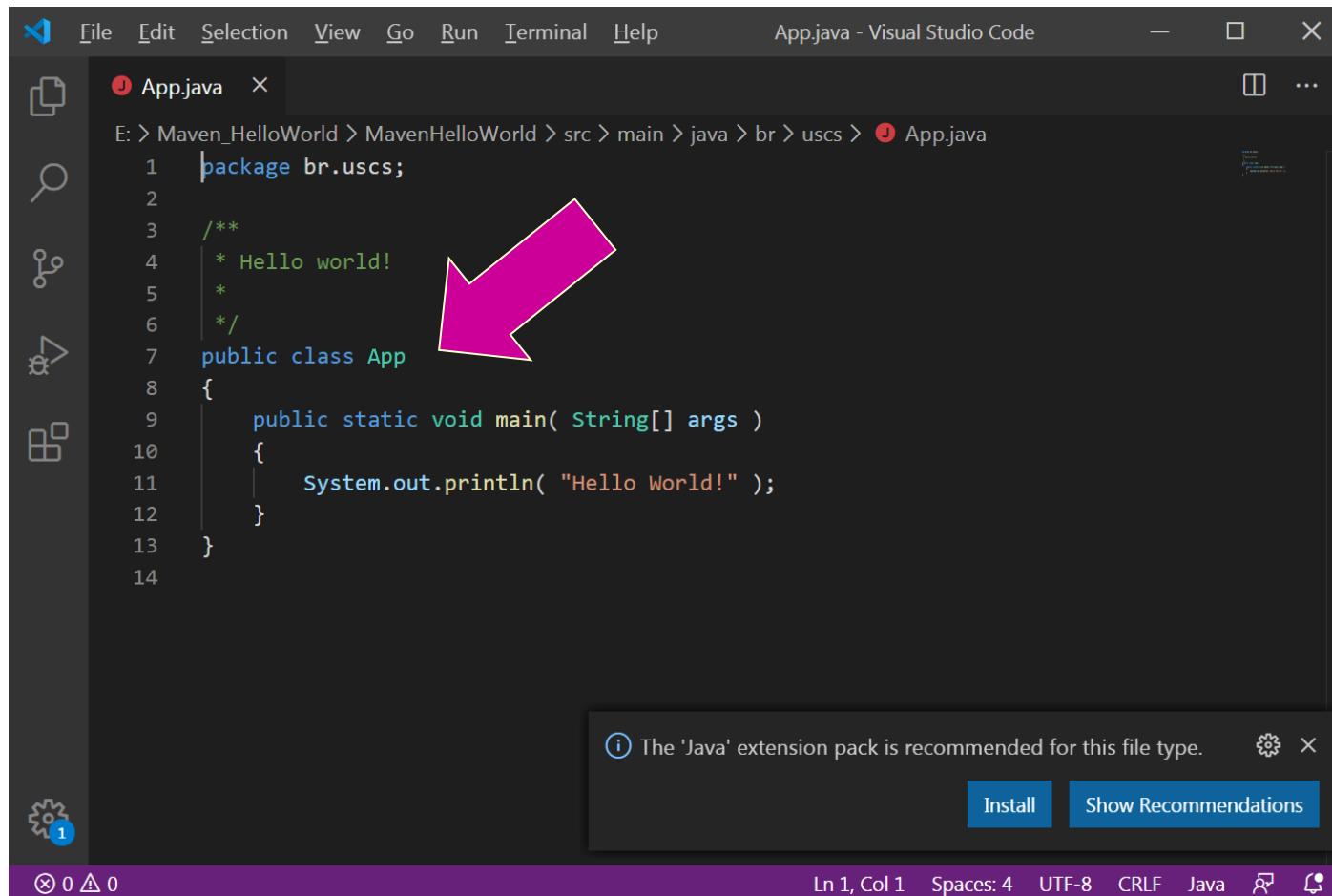
15. Estamos trabalhando com Java 1.8, Vamos então atualizar o POM.XML gerado.



```
E:\Maven_HelloWorld\MavenHelloWorld\pom.xml - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window 
ECMA4012020ENGANDSCS NOTAS settings.xml pom.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <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">
4   <modelVersion>4.0.0</modelVersion>
5
6   <groupId>br.uscs</groupId>
7   <artifactId> MavenHelloWorld</artifactId>
8   <version>1.0-SNAPSHOT</version>
9
10  <name> MavenHelloWorld</name>
11  <!-- FIXME change it to the project's website -->
12  <url>http://www.example.com</url>
13
14  <properties>
15    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
16    <maven.compiler.source>1.8</maven.compiler.source>
17    <maven.compiler.target>1.8</maven.compiler.target>
18  </properties>
19
20
extensible Markup Language file Jeng : 76 Ln : 1 Col : 1 Sel : 0 | 0 Windows (CR LF) UTF-8 INS
```

# Exemplo de Geração – passo 3

16. A classe App com o método main() foi criada.



```
File Edit Selection View Go Run Terminal Help App.java - Visual Studio Code
App.java  X
E: > Maven_HelloWorld > MavenHelloWorld > src > main > java > br > uscs > App.java
1 package br.uscs;
2
3 /**
4 * Hello world!
5 *
6 */
7 public class App
8 {
9     public static void main( String[] args )
10    {
11        System.out.println( "Hello World!" );
12    }
13 }
14
```

The 'Java' extension pack is recommended for this file type. [Install](#) [Show Recommendations](#)

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Java ⚙️ 🔍



# Exemplo de Geração – passo 4

1. Vamos compilar a aplicação: **\$ mvn compile** (na pasta da aplicação)

A screenshot of a Windows Command Prompt window titled "Command Prompt". The window has a blue title bar and a black background. In the title bar, it says "C:\ Command Prompt". Inside the window, the command "E:\Maven\_Helloworld\MavenHelloWorld>mvn compile" is visible, indicating the user is in the MavenHelloWorld directory and is about to run the "mvn compile" command.





# Exemplo de Geração – passo 4

2. Serão feitos downloads dos módulos requisitados.

```
Command Prompt
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-compiler-javac/2.8.4/plexus-compiler-javac-2.8.4.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-compiler-manager/2.8.4/plexus-compiler-manager-2.8.4.jar (4.7 kB at 5.2 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-compiler-api/2.8.4/plexus-compiler-api-2.8.4.jar (27 kB at 29 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-compiler-javac/2.8.4/plexus-compiler-javac-2.8.4.jar (21 kB at 19 kB/s)

Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/2.0.4/plexus-utils-2.0.4.jar (222 kB at 184 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/com/thoughtworks/qdox/qdox/2.0-M9/qdox-2.0-M9.jar (317 kB at 232 kB/s)
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to E:\Maven_Helloworld\MavenHelloWorld\target\classes
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  35.168 s
[INFO] Finished at: 2020-05-03T12:13:41-07:00
[INFO] -----
```

E:\Maven\_Helloworld\MavenHelloWorld>



# Exemplo de Geração – passo 4

3. Será também criada a pasta target com os classes geradas.

Name	Date modified	Type	Size
src	03-May-20 11:55 AM	File folder	
target	03-May-20 12:13 PM	File folder	
pom	03-May-20 12:08 PM	XML Document	3 KB



# Exemplo de Geração – passo 4

## 4. Visualizando também o repositório Local



Name	Date modified	Type
antr	03-May-20 11:16 AM	File folder
asm	03-May-20 11:16 AM	File folder
backport-util-concurrent	03-May-20 12:13 PM	File folder
com	03-May-20 12:13 PM	File folder
commons-codec	03-May-20 11:16 AM	File folder
commons-collections	03-May-20 11:16 AM	File folder
commons-io	03-May-20 11:16 AM	File folder
commons-lang	03-May-20 11:16 AM	File folder
jdom	03-May-20 11:16 AM	File folder
junit	03-May-20 12:13 PM	File folder
net	03-May-20 11:16 AM	File folder
org	03-May-20 12:13 PM	File folder





# Exemplo de Geração – passo 5

1. Gerando ambiente de teste: **\$ mvn test** (na pasta do projeto)

A screenshot of a Windows Command Prompt window titled "Command Prompt". The window has a blue header bar and a black body. In the top-left corner of the body, there is a small white icon. The command "mvn test" is typed in the window, with the "mvn" part in black and "test" in red. The cursor is positioned after the "test" command.



# Exemplo de Geração – passo 5

## 2. Efetuando o BUILD de teste

```
Command Prompt
B/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/
surefire/surefire-junit4/2.22.1/surefire-junit4-2.22.1.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/s
urefire/surefire-junit4/2.22.1/surefire-junit4-2.22.1.jar (85 kB at 221 kB/s)
[INFO] -----
[INFO] T E S T S
[INFO] -----
[INFO] Running br.uscs.AppTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.038 s -
in br.uscs.AppTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 22.914 s
[INFO] Finished at: 2020-05-03T12:21:03-07:00
[INFO] -----
```

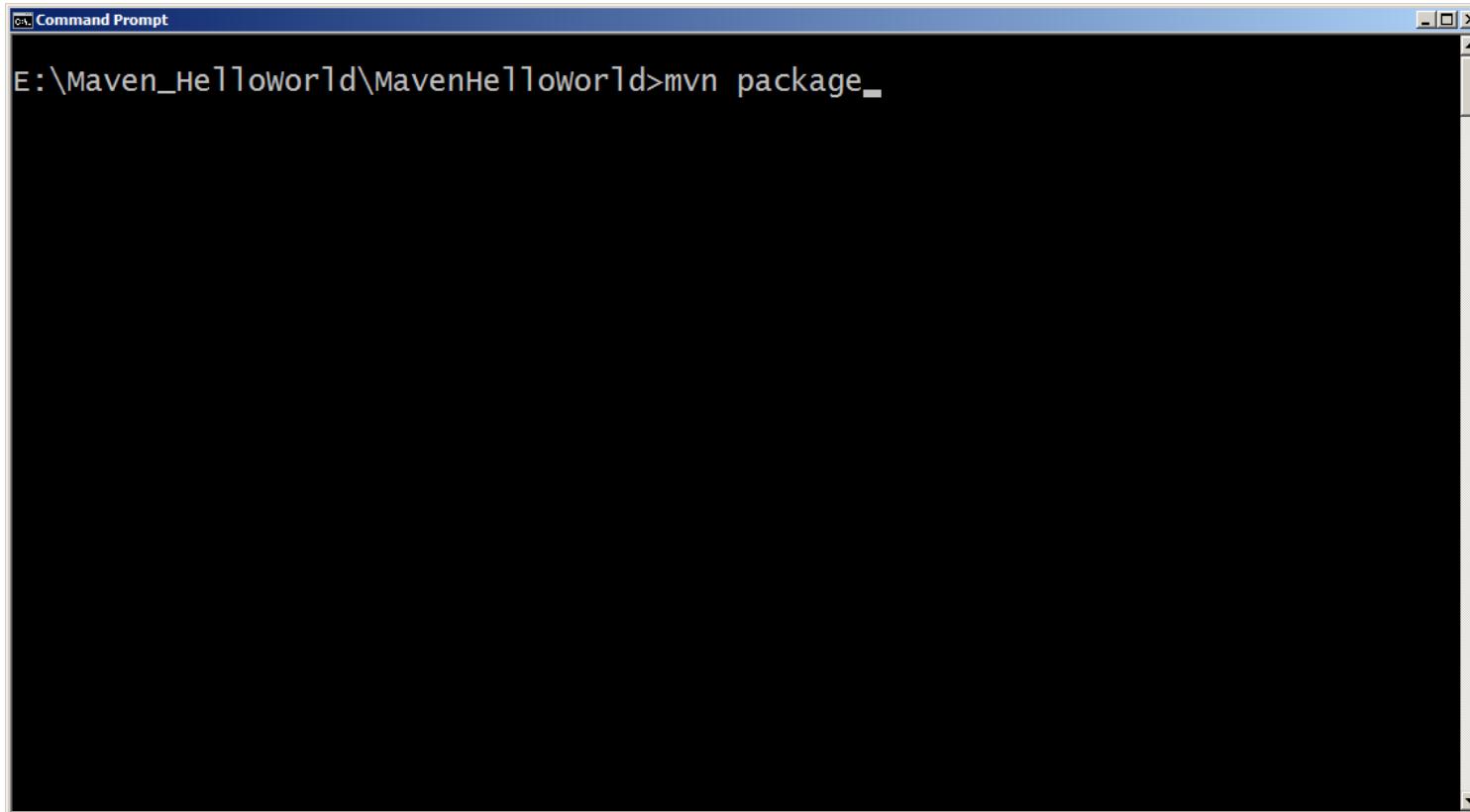
E:\Maven\_Helloworld\MavenHelloWorld>





# Exemplo de Geração – passo 6

1. Empacotando a aplicação: **mvn package** (geração .jar)



A screenshot of a Windows Command Prompt window titled "Command Prompt". The window shows the path "E:\Maven\_Helloworld\MavenHelloWorld>" followed by the command "mvn package". The rest of the window is blank, indicating that the command has not yet been executed.





# Exemplo de Geração – passo 6

## 2. .jar gerado

```
Command Prompt
nappy/0.4/snappy-0.4.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/iq80/snappy/snappy/0.4/snappy-0.4.jar (58 kB at 46 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/tukaani/xz/1.5/xz-1.5.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-io/2.7.1/plexus-io-2.7.1.jar (86 kB at 57 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-shared-utils/3.0.1/maven-shared-utils-3.0.1.jar (154 kB at 81 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/tukaani/xz/1.5/xz-1.5.jar (100 kB at 47 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-archiver/3.4/plexus-archiver-3.4.jar (187 kB at 78 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/commons/commons-compress/1.11/commons-compress-1.11.jar (426 kB at 164 kB/s)
[INFO] Building jar: E:\Maven_Helloworld\MavenHelloWorld\target\MavenHelloWorld-1.0-SNAPSHOT.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 10.666 s
[INFO] Finished at: 2020-05-03T12:25:24-07:00
[INFO] -----
```

E:\Maven\_Helloworld\MavenHelloWorld>





# Exemplo de Geração – passo 6

## 3. Verificando na pasta target o .jar criado

```
Windows Command Prompt
1 File(s)          2,736 bytes
4 Dir(s)   365,494,022,144 bytes free

E:\Maven_Helloworld\MavenHelloWorld>cd target

E:\Maven_Helloworld\MavenHelloWorld\target>dir
Volume in drive E is Dados
Volume Serial Number is 14D9-25F0

Directory of E:\Maven_Helloworld\MavenHelloWorld\target

03-May-20  12:25 PM    <DIR>      .
03-May-20  12:25 PM    <DIR>      ..
03-May-20  12:13 PM    <DIR>      classes
03-May-20  12:13 PM    <DIR>      generated-sources
03-May-20  12:20 PM    <DIR>      generated-test-sources
03-May-20  12:25 PM    <DIR>      maven-archiver
03-May-20  12:13 PM    <DIR>      maven-status
03-May-20  12:25 PM          2,585 MavenHelloWorld-1.0-SNAPSHOT.jar
03-May-20  12:21 PM    <DIR>      surefire-reports
03-May-20  12:20 PM    <DIR>      test-classes
                           1 File(s)          2,585 bytes
                           9 Dir(s)   365,494,022,144 bytes free

E:\Maven_Helloworld\MavenHelloWorld\target>
```

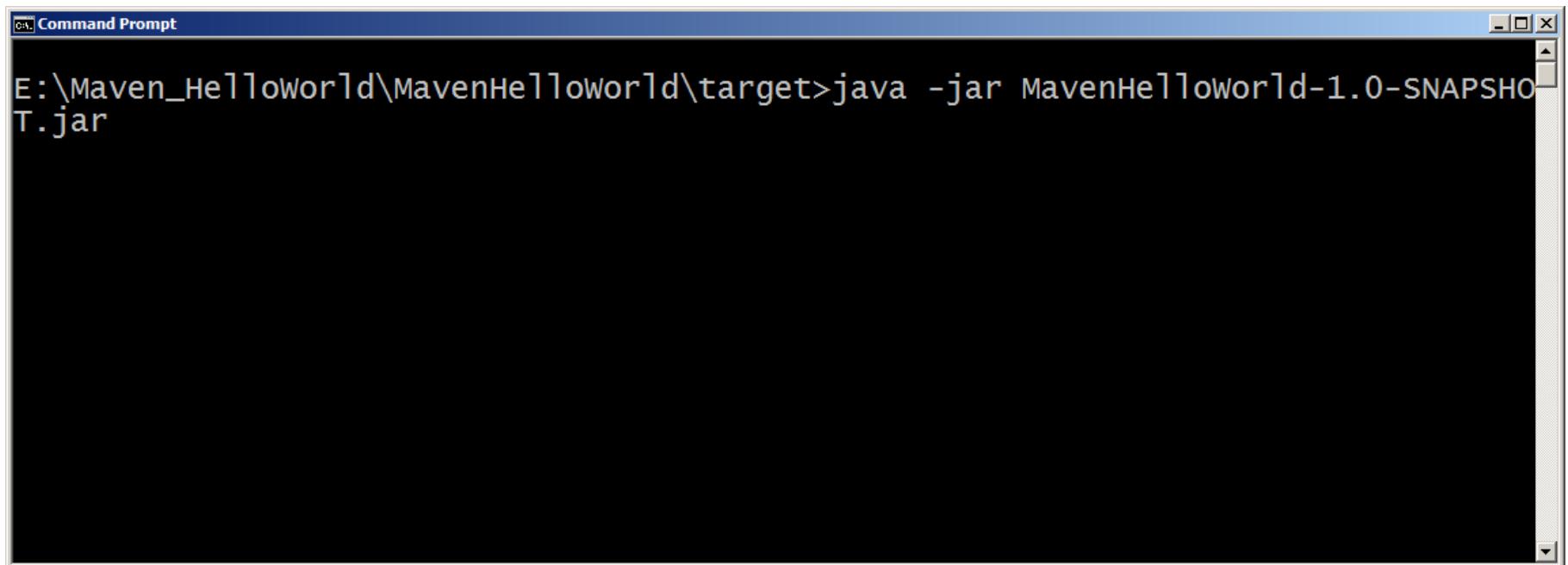




# Exemplo de Geração – passo 7

1. Na pasta target do projeto vamos executar o .jar

Comando : **java -jar MavenHelloWorld-1.0-SNAPSHOT.jar**



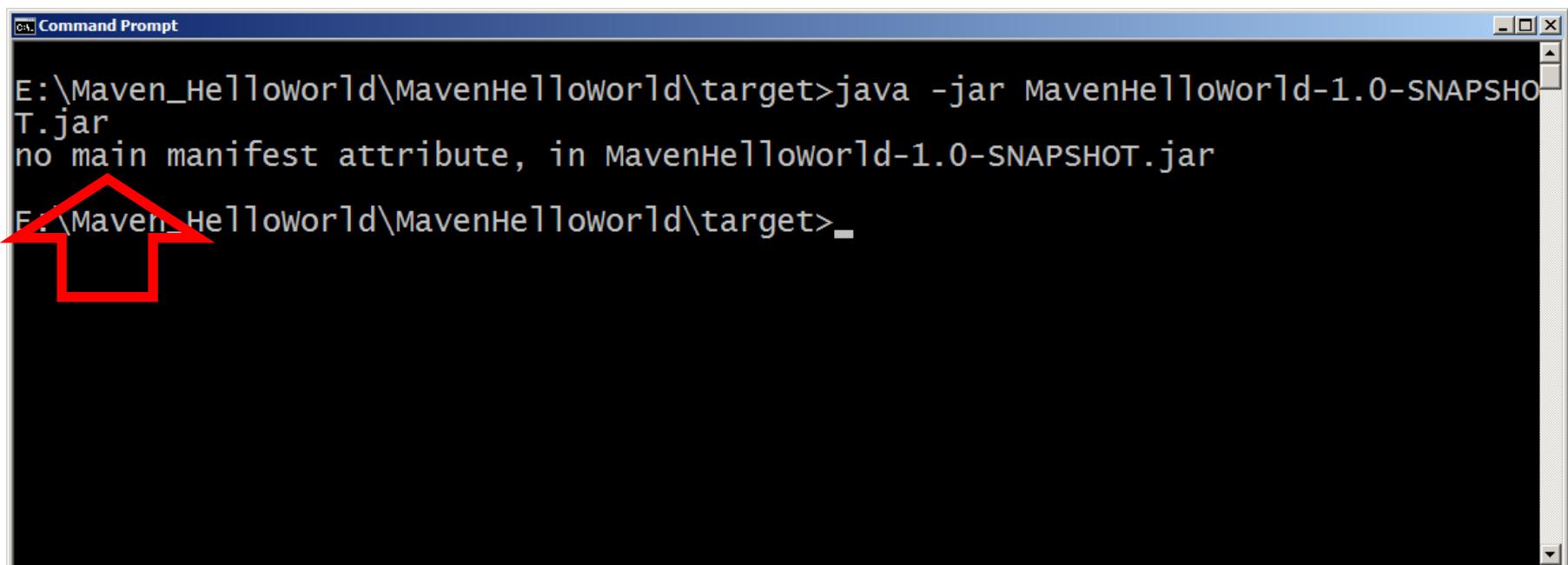
```
Command Prompt
E:\Maven_Helloworld\MavenHelloWorld\target>java -jar MavenHelloWorld-1.0-SNAPSHOT.jar
```





# Exemplo de Geração – passo 7

2. Na hora da execução o interpretador Java retorna erro informando que **não** encontrou o arquivo com atributos **Manifest**



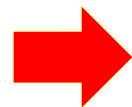
```
CD Command Prompt
E:\Maven_Helloworld\MavenHelloWorld\target>java -jar MavenHelloWorld-1.0-SNAPSHOT.jar
no main manifest attribute, in MavenHelloWorld-1.0-SNAPSHOT.jar
E:\Maven_Helloworld\MavenHelloWorld\target>_
```

A red house-shaped arrow points to the word "Manifest" in the error message.



# Exemplo de Geração – passo 8

1. Vamos incluir no POM.XML o código XML para gerar o Manifest.



```
<plugin>
    <!-- Build an executable JAR -->
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-jar-plugin</artifactId>
    <version>2.4</version>
    <configuration>
        <archive>
            <manifest>
                <mainClass>br.uscs.App</mainClass>
            </manifest>
        </archive>
    </configuration>
</plugin>
```





# Exemplo de Geração – passo 9

1. Vamos gerar um novo build da aplicação : mvn clean package

The screenshot shows a Windows Command Prompt window titled "Command Prompt". The path E:\Maven\_Helloworld\MavenHelloWorld\target is displayed. Two commands are run:  
1. java -jar MavenHelloWorld-1.0-SNAPSHOT.jar  
This command results in an error message: "no main manifest attribute, in MavenHelloWorld-1.0-SNAPSHOT.jar".  
2. mvn clean package

```
E:\Maven_Helloworld\MavenHelloWorld\target>java -jar MavenHelloWorld-1.0-SNAPSHOT.jar
no main manifest attribute, in MavenHelloWorld-1.0-SNAPSHOT.jar
E:\Maven_Helloworld\MavenHelloWorld\target>mvn clean package
```



# Exemplo de Geração – passo 9

## 2. Build recriação...

```
Command Prompt
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, skipped: 0
[INFO]
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ MavenHelloWorld ---
[INFO] Building jar: E:\Maven_Helloworld\MavenHelloWorld\target\MavenHelloWorld-
1.0-SNAPSHOT.jar
[INFO] -
[INFO] BUILD SUCCESS
[INFO] -
[INFO] Total time: 3.207 s
[INFO] Finished at: 2020-05-03T12:44:35-07:00
[INFO] -
```

E:\Maven\_Helloworld\MavenHelloWorld>



# Exemplo de Geração – passo 9

3. Executando o arquivo .jar criado, na pasta target :

Comando : `java -jar MavenHelloWorld-1.0-SNAPSHOT.jar`



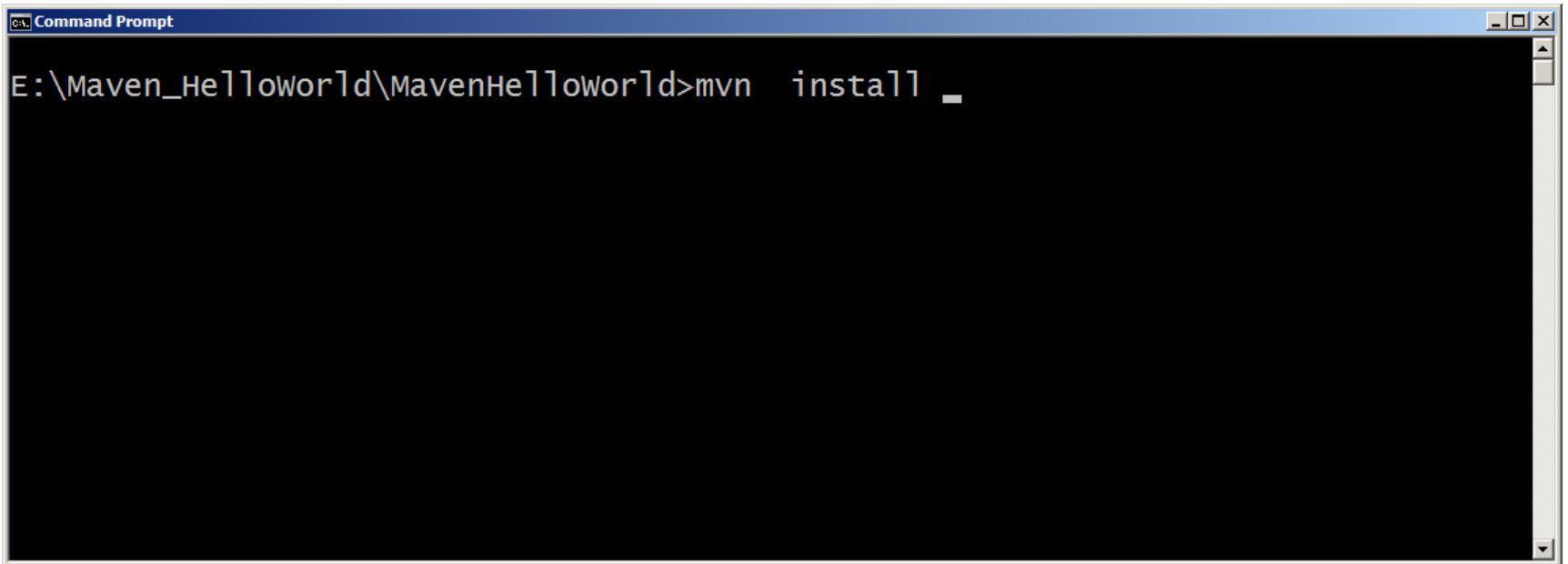
```
Command Prompt  
E:\Maven_Helloworld\MavenHelloWorld\target>java -jar MavenHelloWorld-1.0-SNAPSHOT.jar  
Hello World!  
E:\Maven_Helloworld\MavenHelloWorld\target>
```



# Exemplo de Geração – passo 10

1. Gerando a aplicação (instalando) no Repositório Local

Comando: mvn install (na pasta do projeto)



A screenshot of a Windows Command Prompt window titled "Command Prompt". The window shows the command "E:\Maven\_Helloworld\MavenHelloWorld>mvn install -" entered at the prompt. The window has a standard blue title bar and a black background.





# Exemplo de Geração – passo 10

## 2. Build concluído

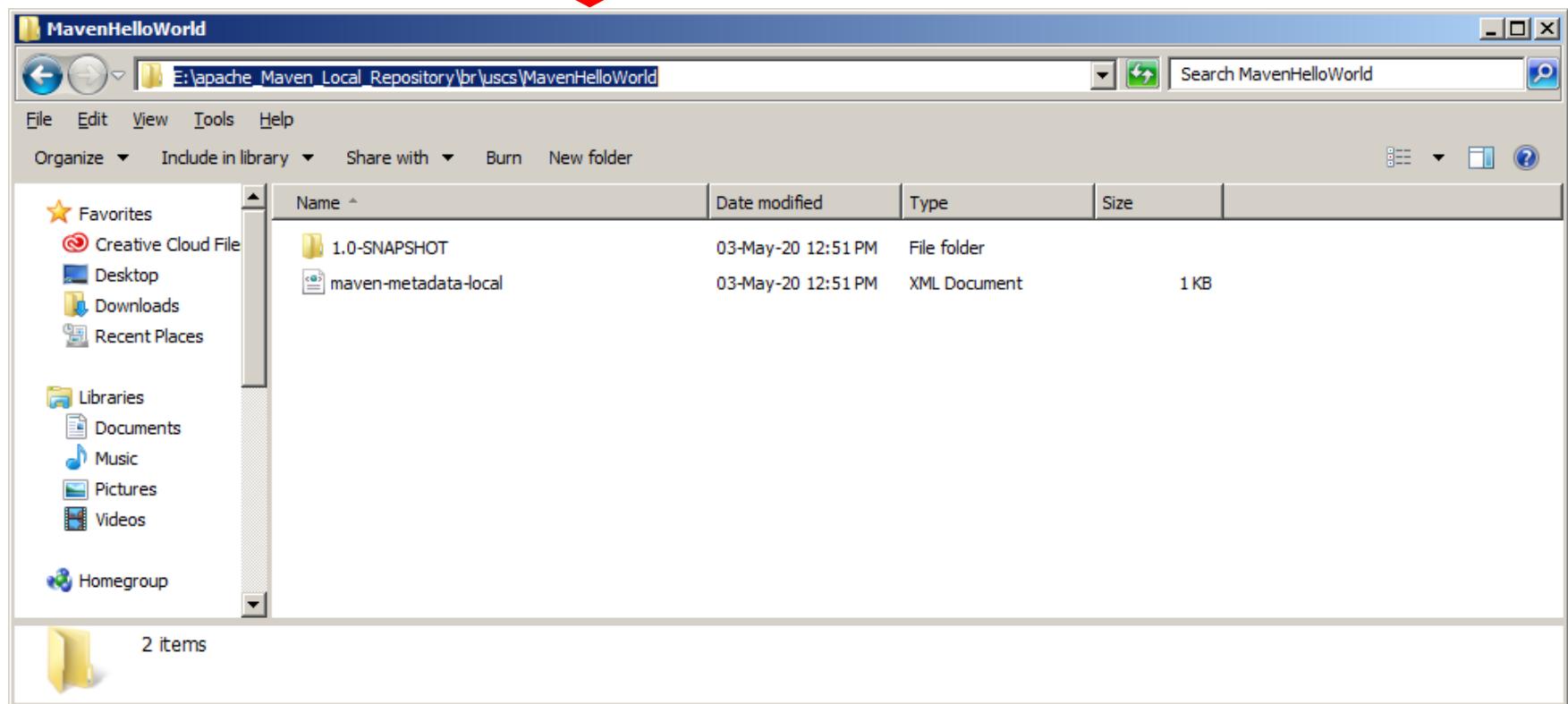
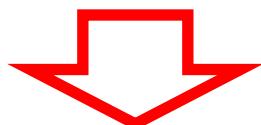
```
Command Prompt
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0.15/plexus-utils-3.0.15.jar (239 kB at 184 kB/s)
[INFO] Installing E:\Maven_Helloworld\MavenHelloWorld\target\MavenHelloWorld-1.0-SNAPSHOT.jar to E:\apache_Maven_Local_Repository\br\uscs\MavenHelloWorld\1.0-SNAPSHOT\MavenHelloWorld-1.0-SNAPSHOT.jar
[INFO] Installing E:\Maven_Helloworld\MavenHelloWorld\pom.xml to E:\apache_Maven_Local_Repository\br\uscs\MavenHelloWorld\1.0-SNAPSHOT\MavenHelloWorld-1.0-SNAPSHOT.pom
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 8.682 s
[INFO] Finished at: 2020-05-03T12:51:48-07:00
[INFO] -----
```

E:\Maven\_Helloworld\MavenHelloWorld>



# Exemplo de Geração – passo 10

## 3. Checando a aplicação no Repositório Local



# Exemplo de Geração – passo 10

4. Processando a aplicação a partir do Repositório Local

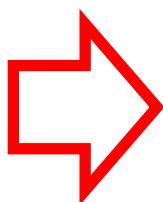


```
Windows Command Prompt  
E:\apache_Maven_Local_Repository\br\uscs\MavenHelloWorld\1.0-SNAPSHOT>java -jar  
MavenHelloWorld-1.0-SNAPSHOT.jar
```



# Exemplo de Geração – passo 10

5. Processando a aplicação a partir do Repositório Local

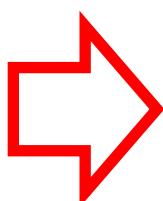


```
Command Prompt  
E:\apache_Maven_Local_Repository\br\uscs\MavenHelloWorld\1.0-SNAPSHOT>java -jar  
MavenHelloWorld-1.0-SNAPSHOT.jar  
Hello World!  
E:\apache_Maven_Local_Repository\br\uscs\MavenHelloWorld\1.0-SNAPSHOT>
```



# Exemplo de Geração – passo 10

5. Processando a aplicação a partir do Repositório Local



```
Command Prompt  
E:\apache_Maven_Local_Repository\br\uscs\MavenHelloWorld\1.0-SNAPSHOT>java -jar  
MavenHelloWorld-1.0-SNAPSHOT.jar  
Hello World!  
E:\apache_Maven_Local_Repository\br\uscs\MavenHelloWorld\1.0-SNAPSHOT>
```



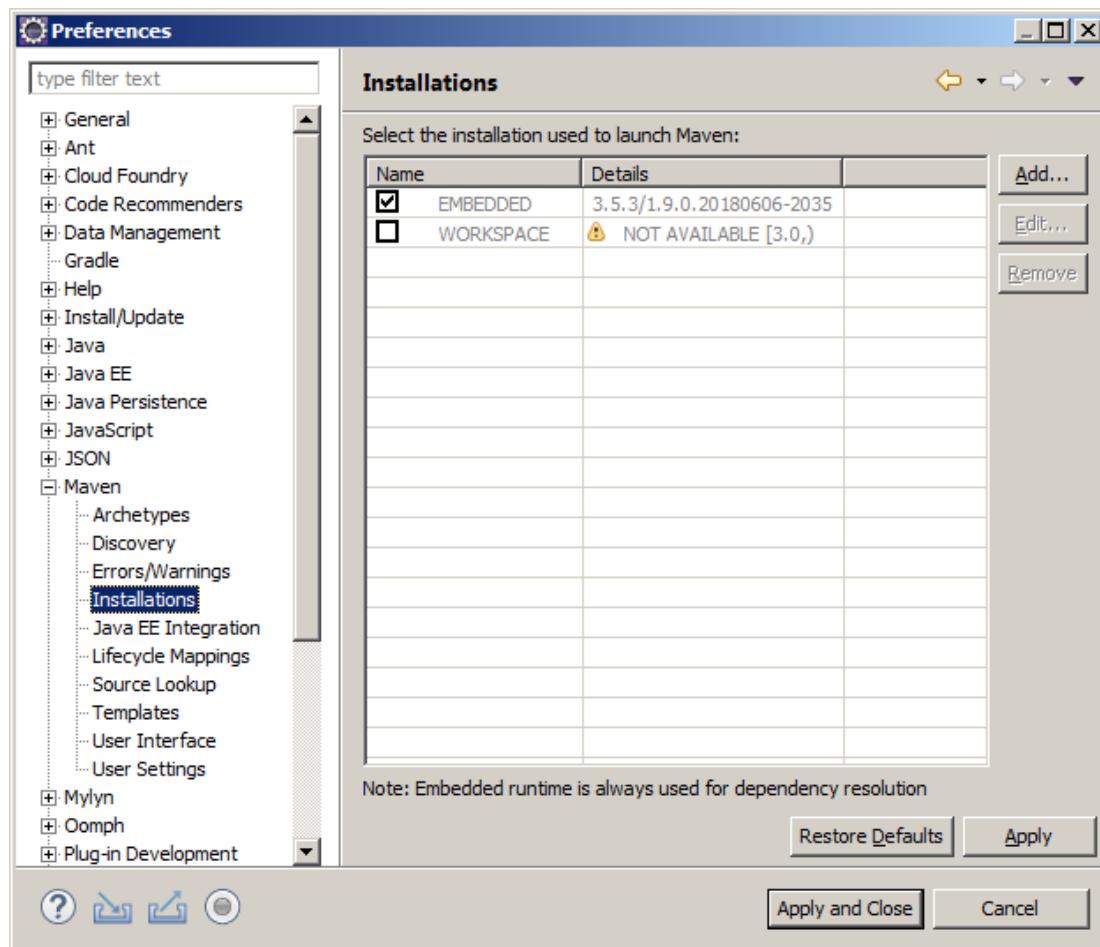
# Integração do Maven com Eclipse

- ✓ Para a integração do Maven com o Eclipse recomenda-se utilizar o plugin M2E;
- ✓ M2E é um plugin do Eclipse que faz a integração com o Maven;
- ✓ Existe também o Maven Eclipse Plugin que gera um projeto para o Eclipse. Este é um plugin do Maven que simplesmente gera os arquivos de configuração necessários para a IDE; Portanto, são plugins diferentes;
- ✓ A distribuição Eclipse for JEE Developers já vem com o plugin M2E e uma instalação interna do Maven;
- ✓ Caso você esteja com uma versão diferente do Eclipse, use o menu Help > Eclipse Marketplace ... , pesquise por M2E e instale o plugin;



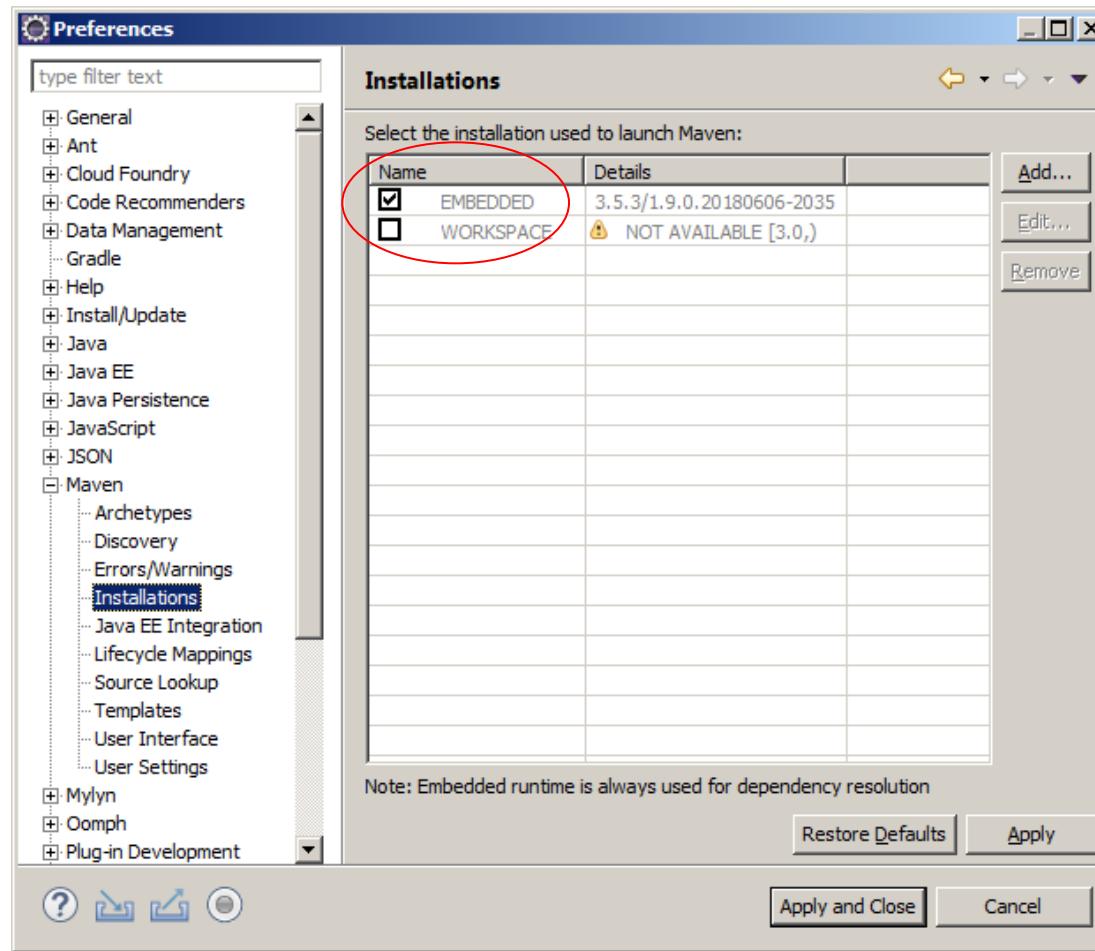
# Integração do Maven com Eclipse

- ✓ Com o plugin **M2E** instalado e o **Eclipse** aberto, acesse o menu Window > Preferences;
- ✓ Vá para a opção **Maven > Installations**.



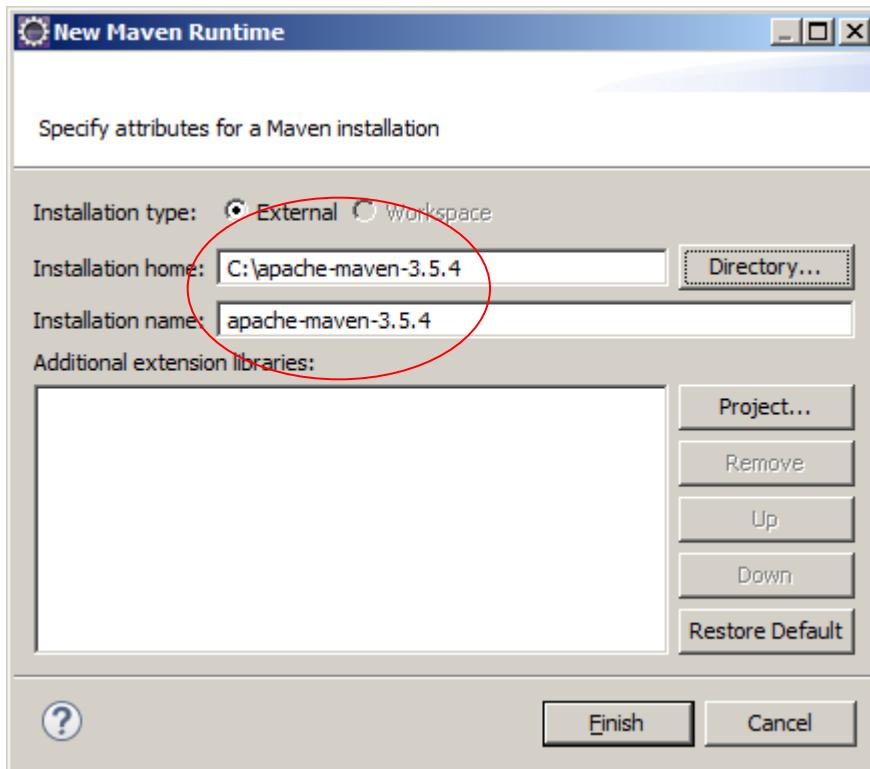
# Integração do Maven com Eclipse

- ✓ Observe que já existe uma instalação “embarcada”, mas com uma versão desatualizada;
- ✓ Vamos adicionar o nosso **Maven** recentemente instalado;
- ✓ Clique em Add...



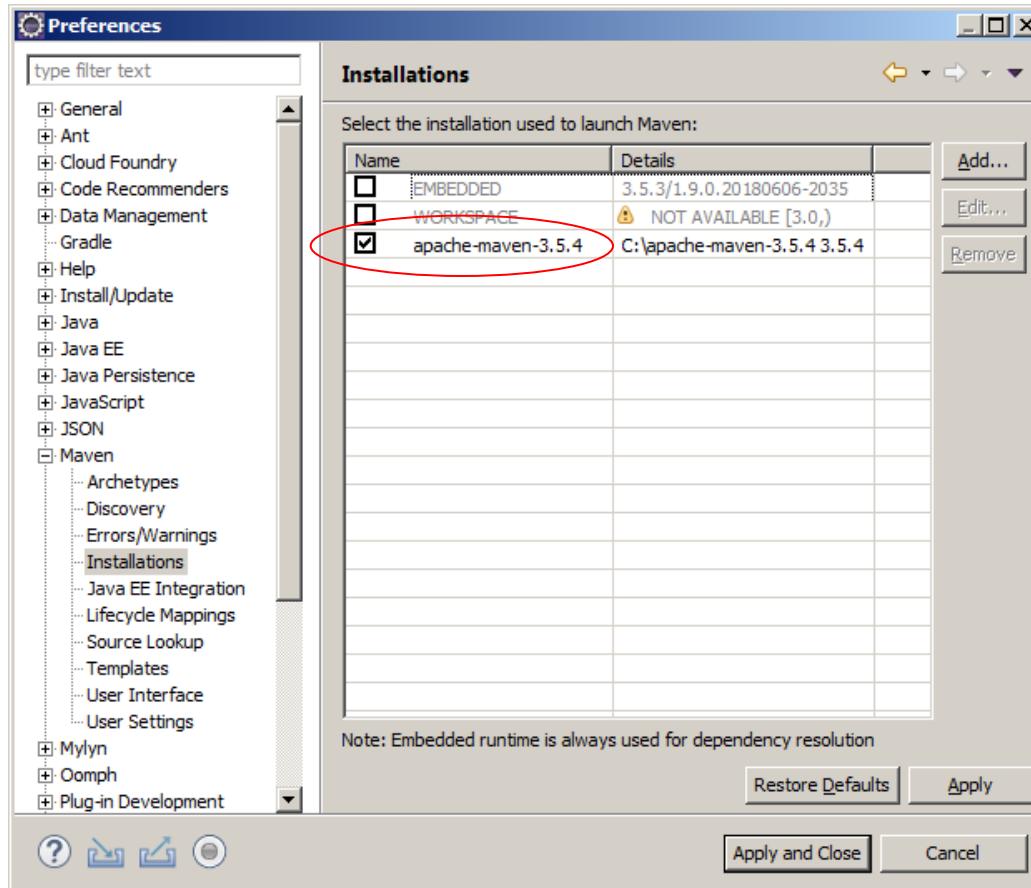
# Integração do Maven com Eclipse

- ✓ Selecione a pasta com a nossa instalação e tecle Finish.



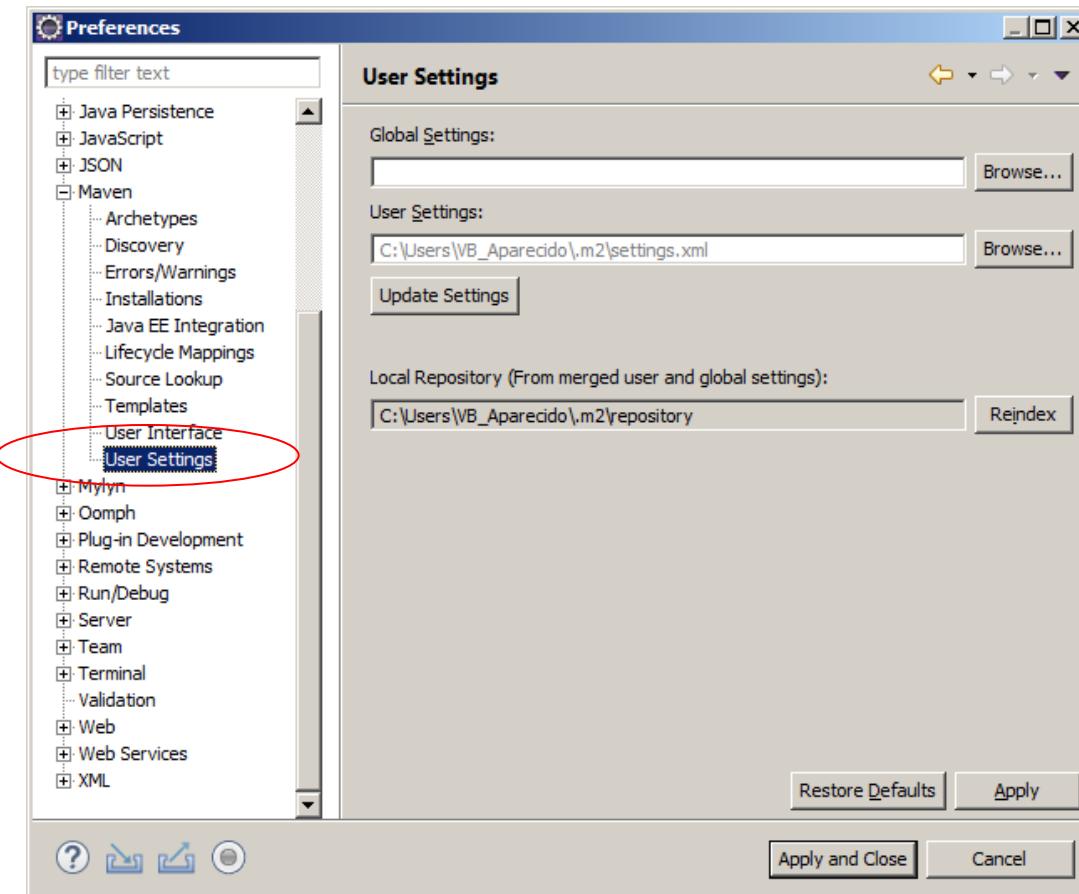
# Integração do Maven com Eclipse

- ✓ Marque a nossa instalação e tecle Apply and Close.



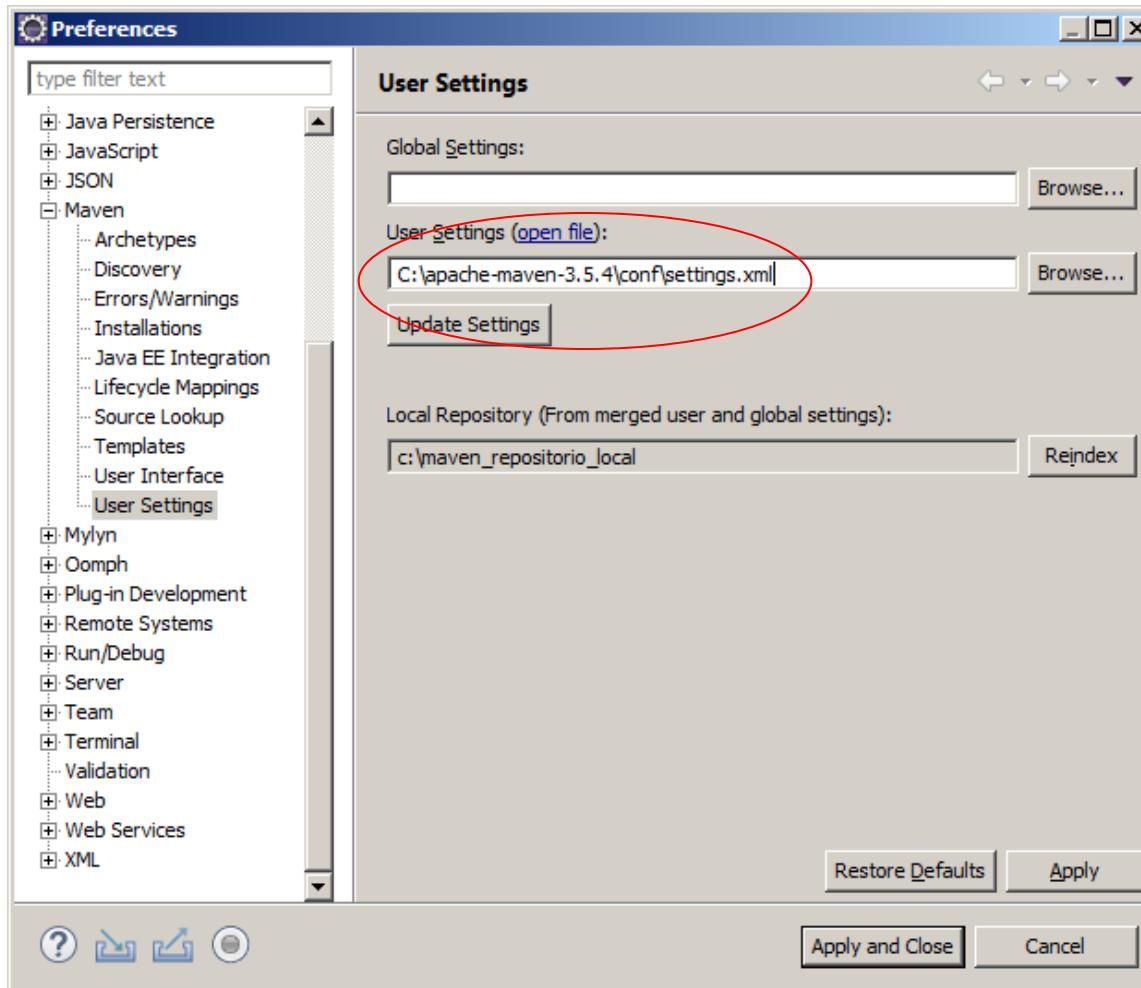
# Integração do Maven com Eclipse

- ✓ Agora vamos atualizar o arquivo de configuração do **Maven**;
- ✓ Vá ao menu **User Settings** e defina o arquivo de configuração existente no diretório onde o **Maven** foi instalado.



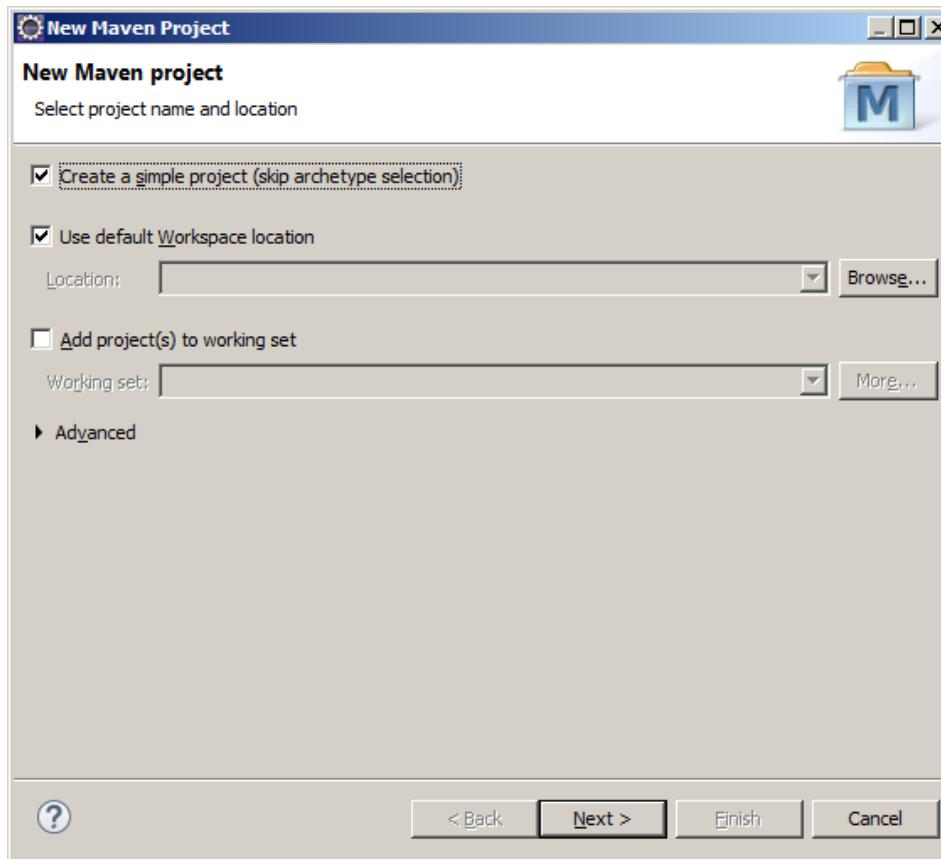
# Integração do Maven com Eclipse

- ✓ Após a definição do arquivo de configuração, tecle em **Update Settings**;
- ✓ Em seguida, tecle em **Apply and Close**.



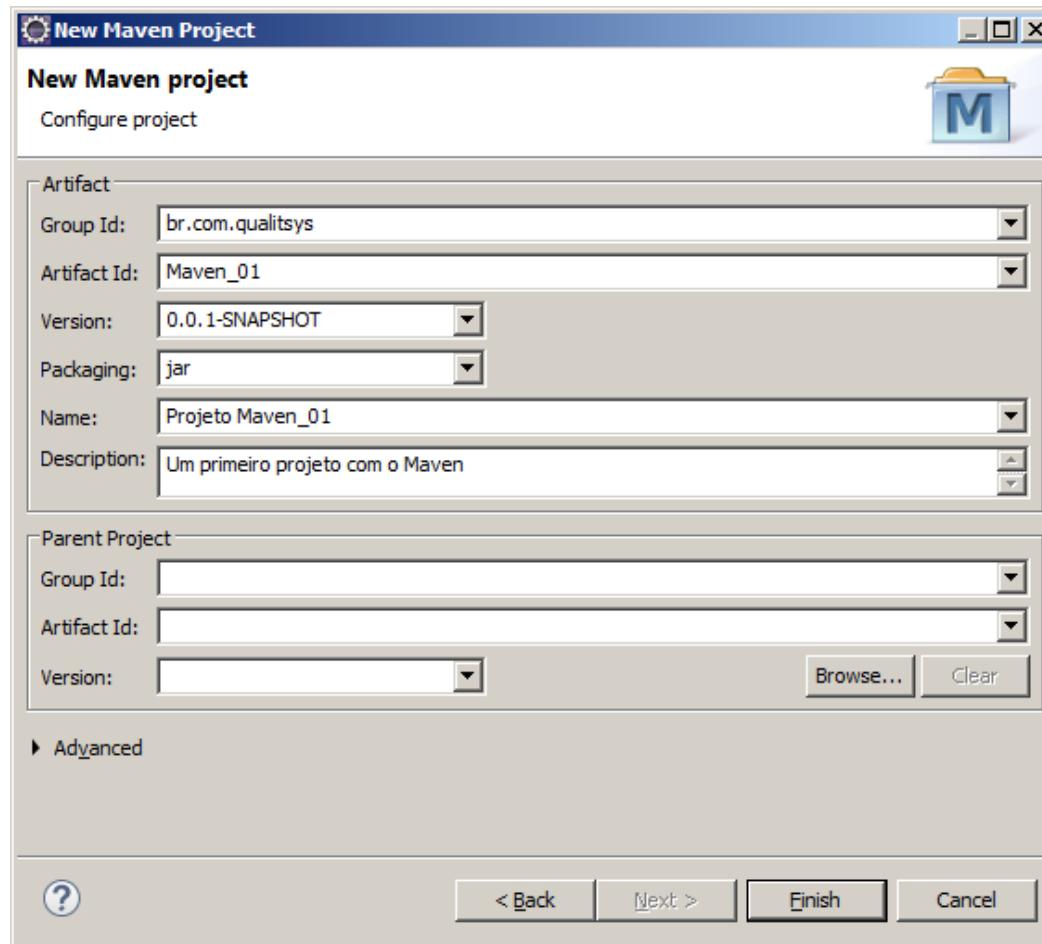
# Criação de um Projeto Maven no Eclipse

- ✓ Com o Eclipse configurado, pode-se criar um novo projeto no **Eclipse**;
- ✓ Acesse o menu **File > New > Maven Project**.
- ✓ Selecione a opção **Create a simple Project (skip archetype selection)**.
- ✓ Clique em **Next >**.



# Criação de um Projeto Maven no Eclipse

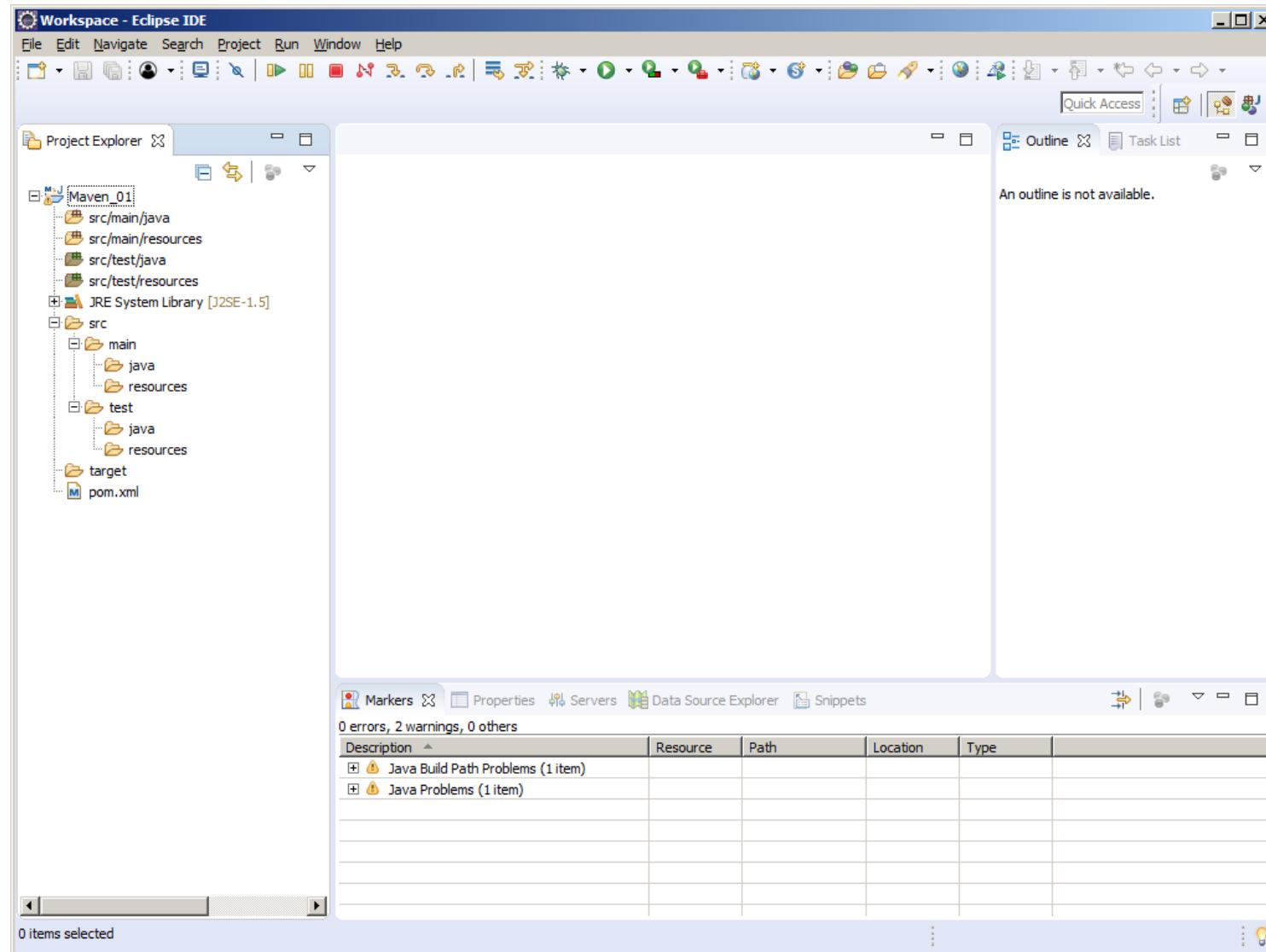
- ✓ **Group Id** representa a organização onde o projeto será desenvolvido;
- ✓ **Artifact Id** corresponde ao projeto que será desenvolvido;
- ✓ **Version** e **Packaging** já deverão estar preenchidos;
- ✓ Clique em **Finish**.





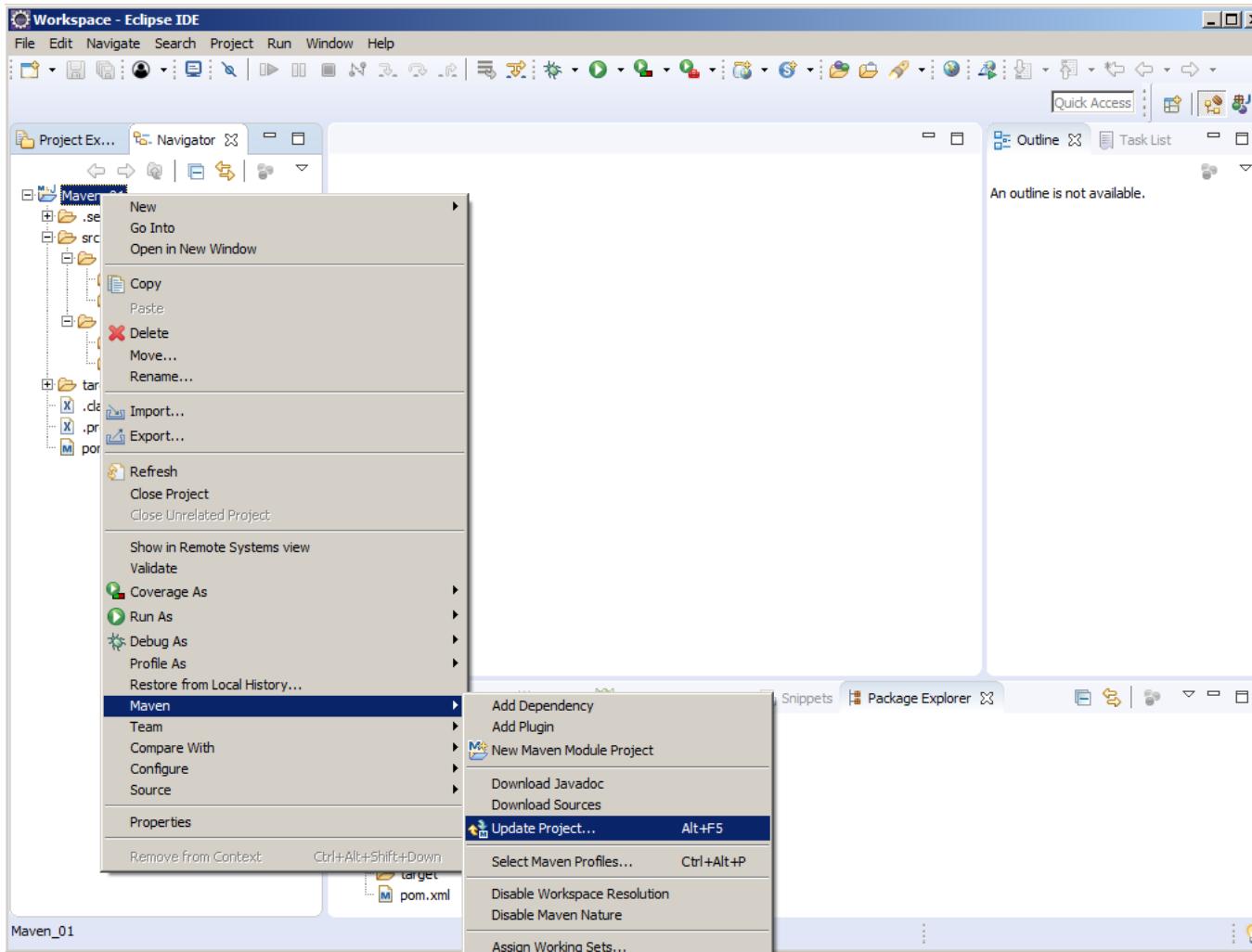
# Criação de um Projeto Maven no Eclipse

✓ Projeto Maven criado!



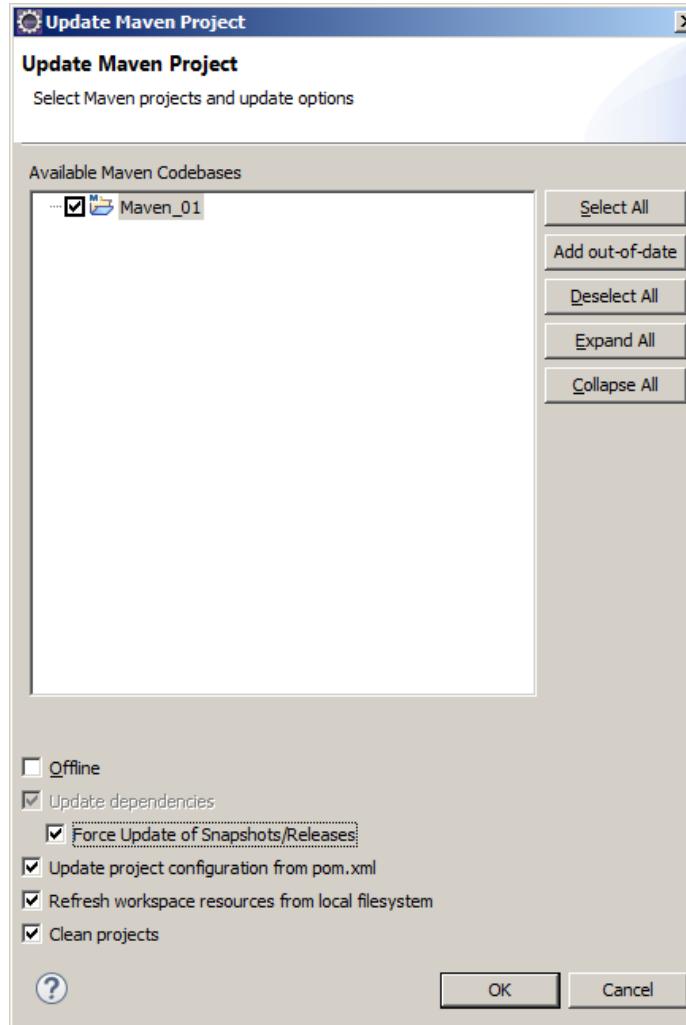
# Criação de um Projeto Maven no Eclipse

- ✓ Para configurar o projeto como um projeto Java, clique no botão direito sobre o projeto, acesse o menu Maven >Update Project ...



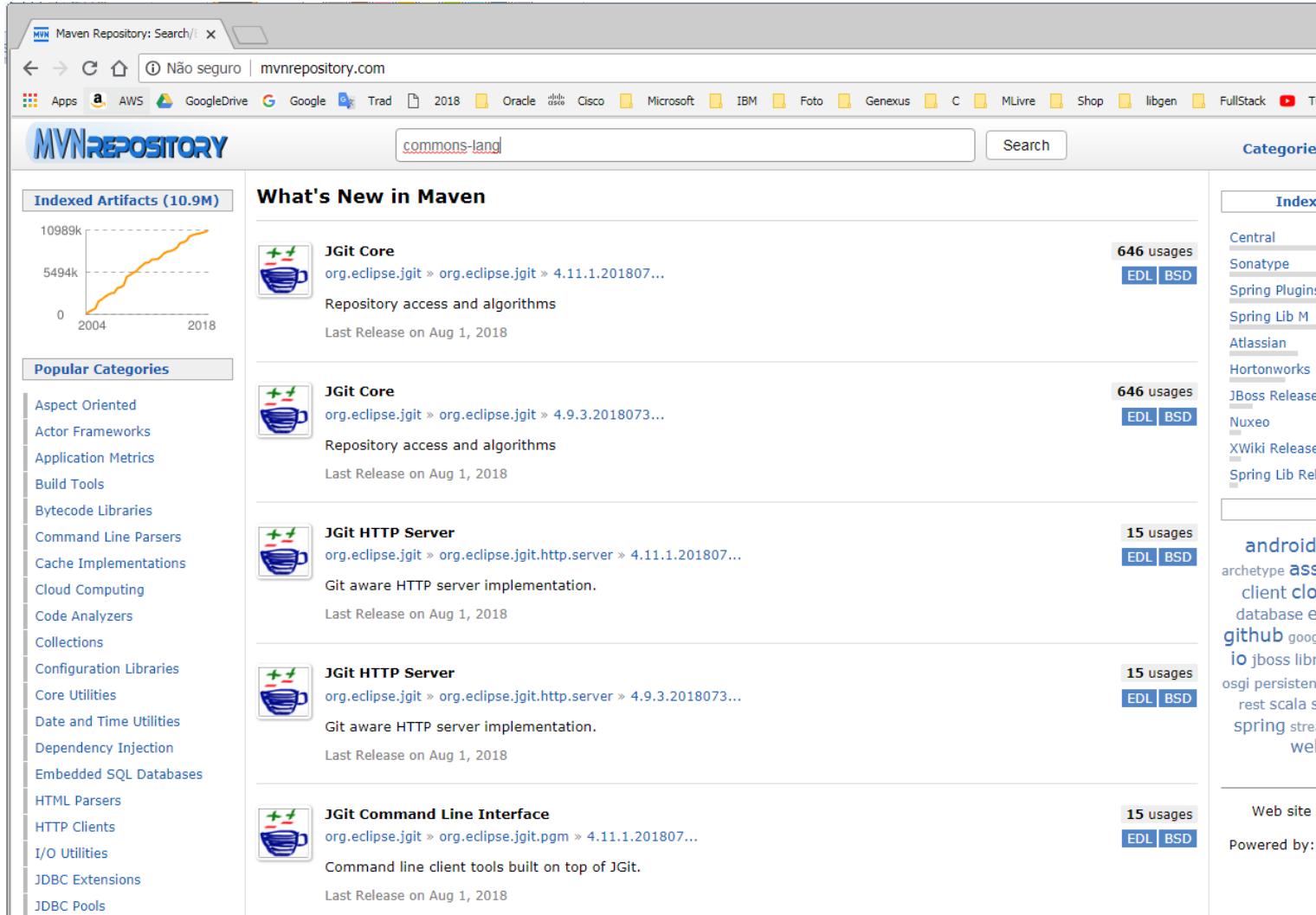
# Criação de um Projeto Maven no Eclipse

- ✓ Selecione a opção **Force Update of Snapshots/Releases**;
- ✓ Clique em **OK**.



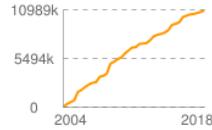
# Adicionando uma dependência

- ✓ Acesse o site <http://mvnrepository.com> que contém uma lista das dependências disponíveis no repositório do Maven.
- ✓ Vamos pesquisar por Commons-Lang.



The screenshot shows the Maven Repository search interface. The search term 'commons-lang' is entered in the search bar. The results page displays several Maven artifacts related to JGit, such as 'JGit Core' and 'JGit HTTP Server', which are used for Git access and algorithms. The sidebar on the left lists popular categories like Aspect Oriented, Actor Frameworks, and Build Tools. The right sidebar shows links to various Maven repositories like Central, Sonatype, and Spring Plugins.

**Indexed Artifacts (10.9M)**



A line chart showing the growth of indexed artifacts over time. The Y-axis represents the number of artifacts (0 to 10989k) and the X-axis represents the year (2004 to 2018). The chart shows a steady increase from approximately 5494k in 2004 to 10989k in 2018.

**Popular Categories**

- Aspect Oriented
- Actor Frameworks
- Application Metrics
- Build Tools
- Bytecode Libraries
- Command Line Parsers
- Cache Implementations
- Cloud Computing
- Code Analyzers
- Collections
- Configuration Libraries
- Core Utilities
- Date and Time Utilities
- Dependency Injection
- Embedded SQL Databases
- HTML Parsers
- HTTP Clients
- I/O Utilities
- JDBC Extensions
- JDBC Pools

**What's New in Maven**

Artifact	Description	Usages	Licenses
<a href="#">JGit Core</a> org.eclipse.jgit > org.eclipse.jgit > 4.11.1.201807...	Repository access and algorithms Last Release on Aug 1, 2018	646 usages	<a href="#">EDL</a> <a href="#">BSD</a>
<a href="#">JGit Core</a> org.eclipse.jgit > org.eclipse.jgit > 4.9.3.2018073...	Repository access and algorithms Last Release on Aug 1, 2018	646 usages	<a href="#">EDL</a> <a href="#">BSD</a>
<a href="#">JGit HTTP Server</a> org.eclipse.jgit > org.eclipse.jgit.http.server > 4.11.1.201807...	Git aware HTTP server implementation. Last Release on Aug 1, 2018	15 usages	<a href="#">EDL</a> <a href="#">BSD</a>
<a href="#">JGit HTTP Server</a> org.eclipse.jgit > org.eclipse.jgit.http.server > 4.9.3.2018073...	Git aware HTTP server implementation. Last Release on Aug 1, 2018	15 usages	<a href="#">EDL</a> <a href="#">BSD</a>
<a href="#">JGit Command Line Interface</a> org.eclipse.jgit > org.eclipse.jgit.pgm > 4.11.1.201807...	Command line client tools built on top of JGit. Last Release on Aug 1, 2018	15 usages	<a href="#">EDL</a> <a href="#">BSD</a>

**Categories**

- Central
- Sonatype
- Spring Plugins
- Spring Lib M
- Atlassian
- Hortonworks
- JBoss Releases
- Nuxeo
- XWiki Releases
- Spring Lib Rel

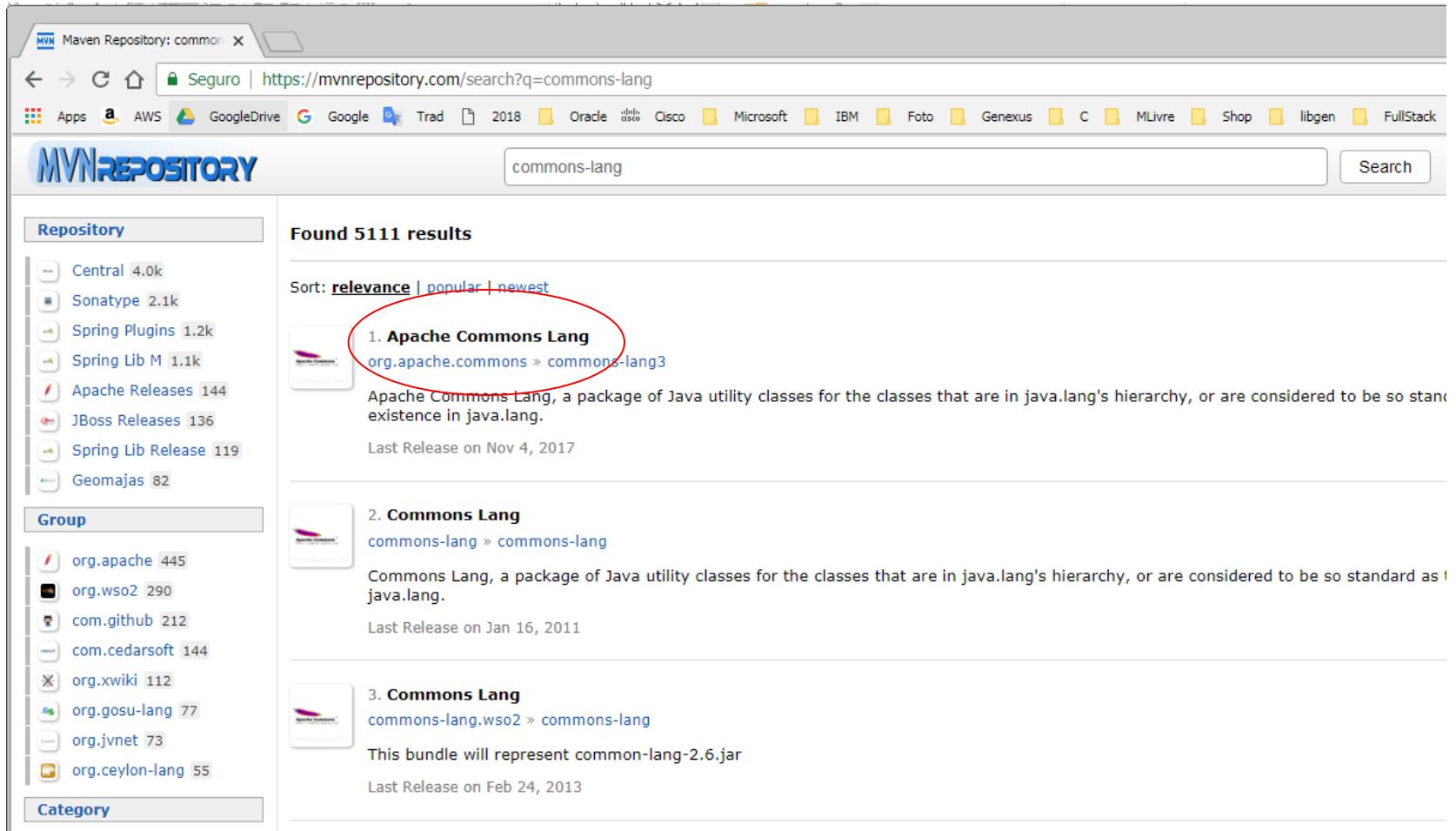
**Index**

- android
- archetype ass
- client cloj
- database ed
- github googl
- io jboss libr
- osgi persistenc
- rest scala si
- spring strea
- web

**Web site c**  
Powered by:

# Adicionando uma dependência

- ✓ Selecione Apache Commons Lang, conforme indicação abaixo.



Maven Repository: common Seguro | https://mvnrepository.com/search?q=commons-lang

Found 5111 results

Sort: relevance | popular | newest

1. Apache Commons Lang  
org.apache.commons > commons-lang3

Apache Commons Lang, a package of Java utility classes for the classes that are in java.lang's hierarchy, or are considered to be so standard as to exist in java.lang.

Last Release on Nov 4, 2017

2. Commons Lang  
commons-lang > commons-lang

Commons Lang, a package of Java utility classes for the classes that are in java.lang's hierarchy, or are considered to be so standard as to exist in java.lang.

Last Release on Jan 16, 2011

3. Commons Lang  
commons-lang.wso2 > commons-lang

This bundle will represent common-lang-2.6.jar

Last Release on Feb 24, 2013

**Repository**

- Central 4.0k
- Sonatype 2.1k
- Spring Plugins 1.2k
- Spring Lib M 1.1k
- Apache Releases 144
- JBoss Releases 136
- Spring Lib Release 119
- Geomajas 82

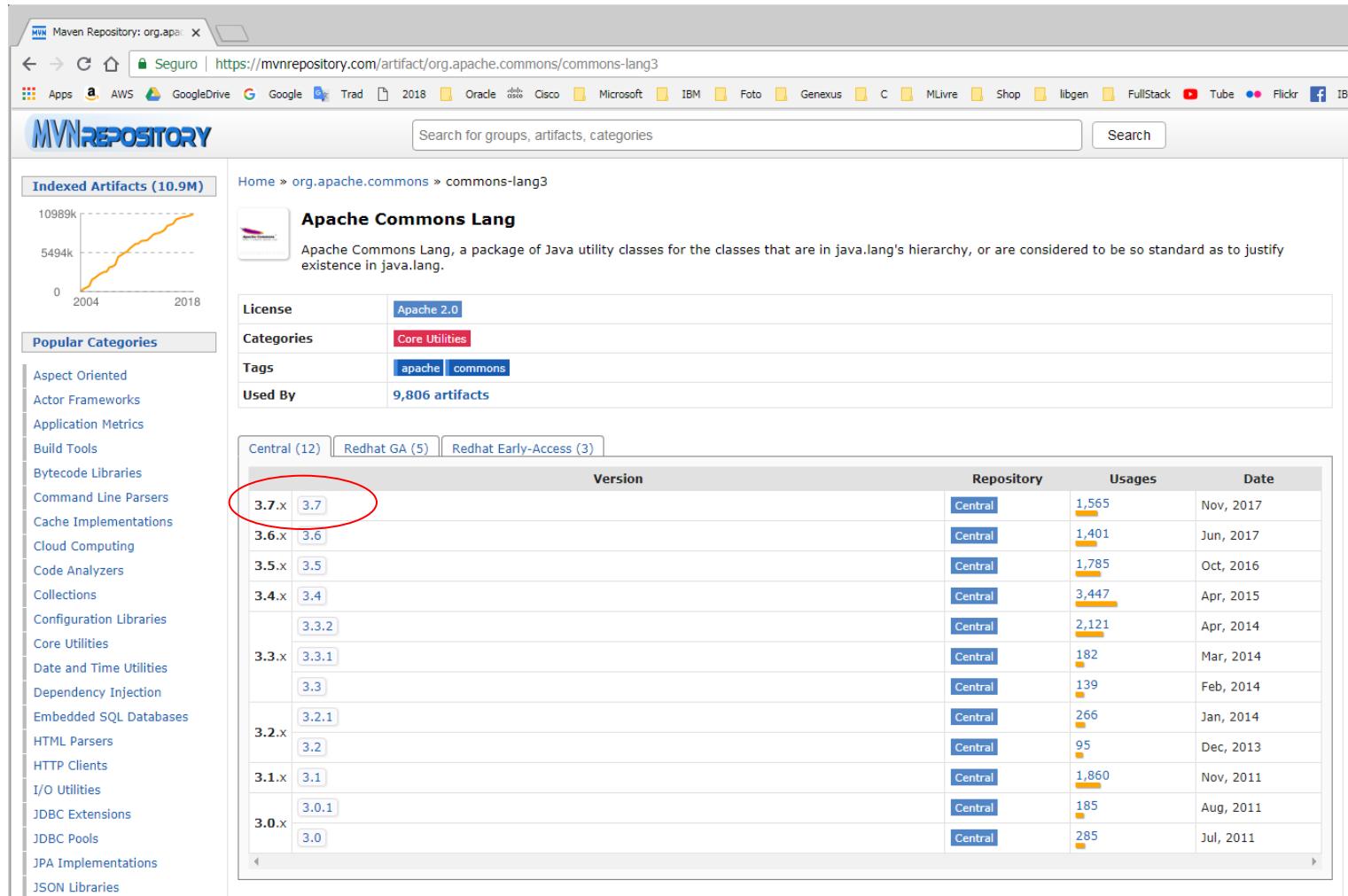
**Group**

- org.apache 445
- org.wso2 290
- com.github 212
- com.cedarsoft 144
- org.xwiki 112
- org.gosu-lang 77
- org.jvnet 73
- org.ceylon-lang 55

**Category**

# Adicionando uma dependência

- ✓ Selecione a mais nova versão, conforme ilustrado na imagem abaixo:



The screenshot shows the Maven Repository page for the Apache Commons Lang3 artifact. The page includes a search bar, navigation links, and a sidebar with popular categories like Aspect Oriented, Actor Frameworks, and Application Metrics. The main content area displays the Apache Commons Lang package details, including its license (Apache 2.0), categories (Core Utilities), and tags (apache, commons). Below this, a table lists various versions of the library, with the '3.7.x' entry circled in red. The table columns include Version, Repository, Usages, and Date.

Version	Repository	Usages	Date
3.7.x	Central	1,565	Nov, 2017
3.6.x	Central	1,401	Jun, 2017
3.5.x	Central	1,785	Oct, 2016
3.4.x	Central	3,447	Apr, 2015
3.3.2	Central	2,121	Apr, 2014
3.3.x	Central	182	Mar, 2014
3.3	Central	139	Feb, 2014
3.2.1	Central	266	Jan, 2014
3.2.x	Central	95	Dec, 2013
3.1.x	Central	1,860	Nov, 2011
3.0.1	Central	185	Aug, 2011
3.0.x	Central	285	Jul, 2011



# Adicionando uma dependência

- ✓ Copie o XML que corresponde à biblioteca **Commons-Lang**

Maven Repository: org.apache.commons/commons-lang3/3.7

Indexed Artifacts (10.9M)

Popular Categories

Apache Commons Lang > 3.7

Apache Commons Lang, a package of Java utility classes for the classes that are in java.lang's hierarchy, or are considered to be so standardly useful that they exist in java.lang.

License	Apache 2.0
Categories	Core Utilities
HomePage	<a href="http://commons.apache.org/proper/commons-lang/">http://commons.apache.org/proper/commons-lang/</a>
Date	(Nov 04, 2017)
Files	<a href="#">pom (26 KB)</a> <a href="#">jar (487 KB)</a> <a href="#">View All</a>
Repositories	Central
Used By	9,806 artifacts

Maven Gradle SBT Ivy Grape Leiningen Buildr

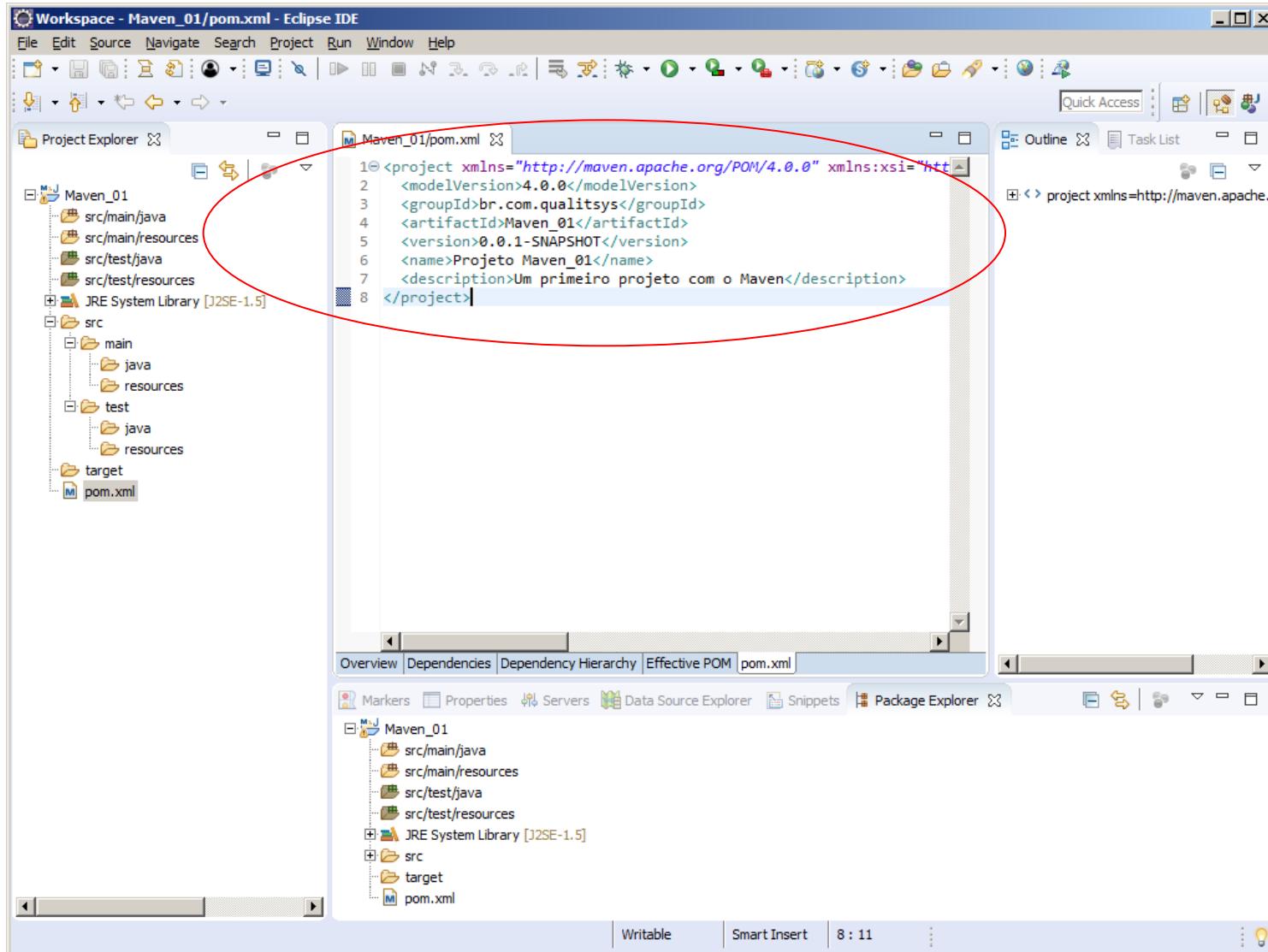
```
<!-- https://mvnrepository.com/artifact/org.apache.commons/commons-lang3 -->
<dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-lang3</artifactId>
    <version>3.7</version>
</dependency>
```

Include comment with link to declaration



# Adicionando uma dependência

- ✓ De volta ao **Eclipse**, clique duas vezes no arquivo **pom.xml** para editá-lo.
- ✓ Pressione **Ctrl+A** para selecionar todo o conteúdo do arquivo e depois **Ctrl+I** para indentar o arquivo.



# Adicionando uma dependência

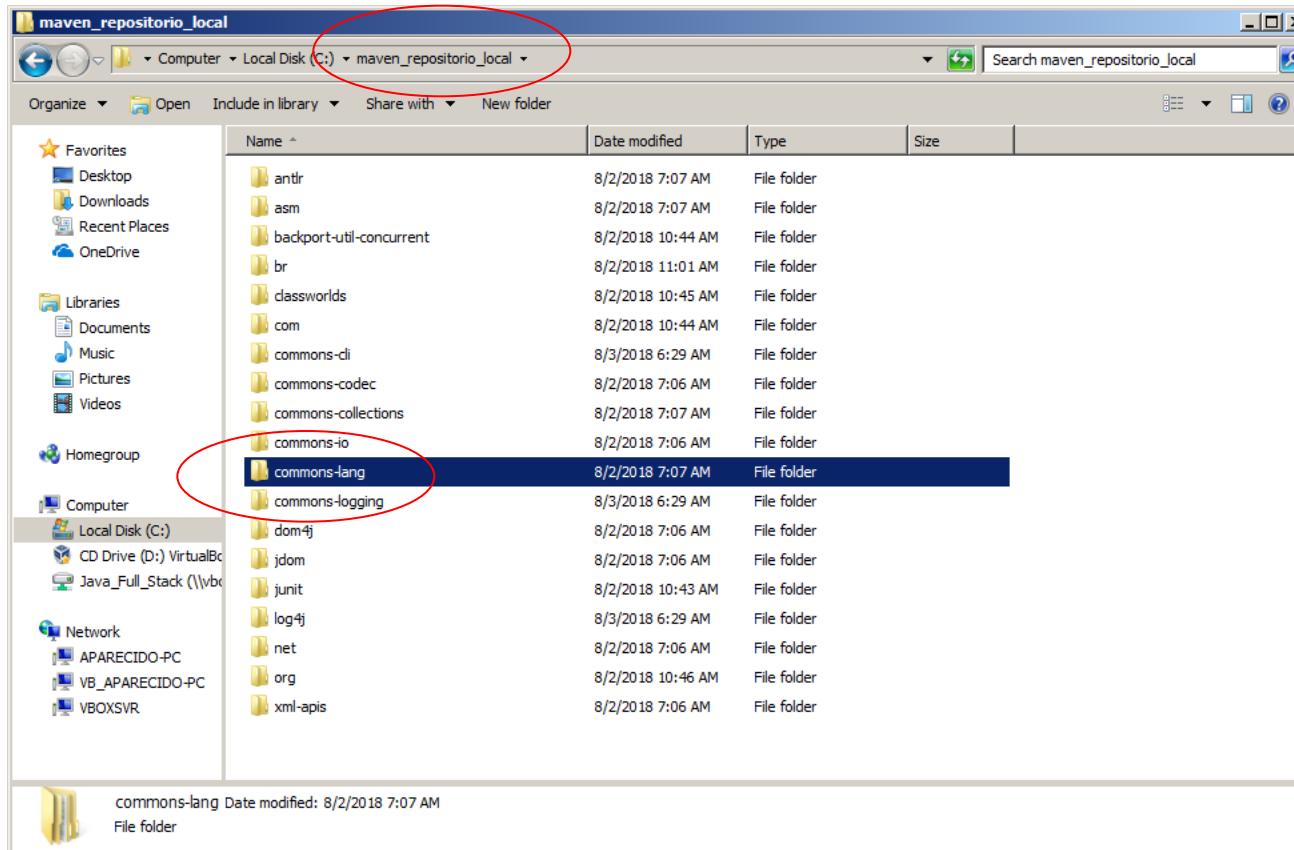
- ✓ Incluir a dependência no arquivo **pom.xml**

```
<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>br.com.qualitsys</groupId>
    <artifactId>Maven_01</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>Projeto Maven_01</name>
    <description>Um primeiro projeto com o Maven</description>

    <dependencies>
        <dependency>
            <groupId>org.apache.commons</groupId>
            <artifactId>commons-lang3</artifactId>
            <version>3.7</version>
        </dependency>
    </dependencies>
</project>
```

# Adicionando uma dependência

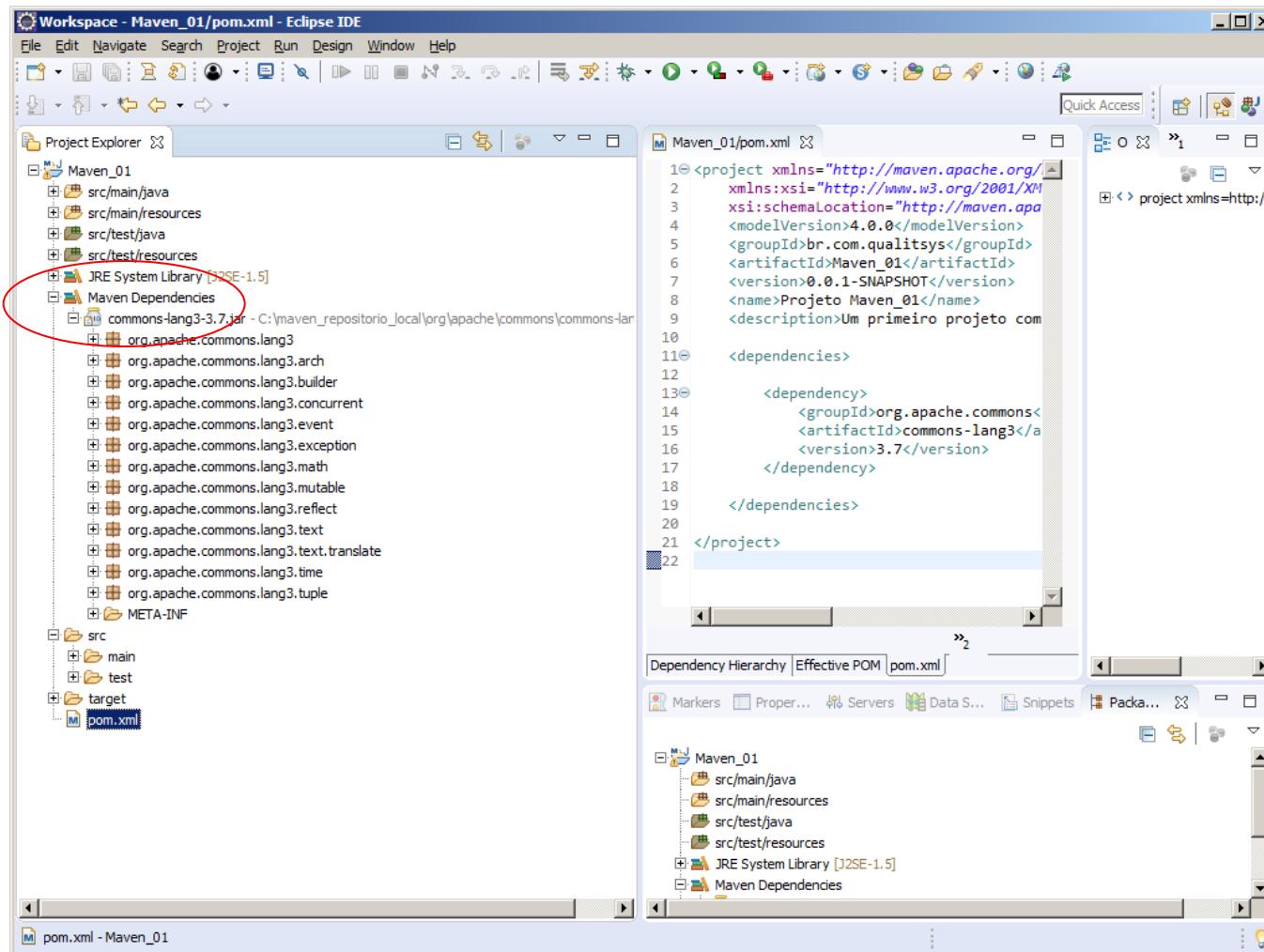
- ✓ Salvar o arquivo **pom.xml**;
- ✓ O plugin **M2E** irá identificar a alteração, baixar automaticamente a dependência do repositório central para o seu repositório local e adicioná-la ao **classpath** do projeto





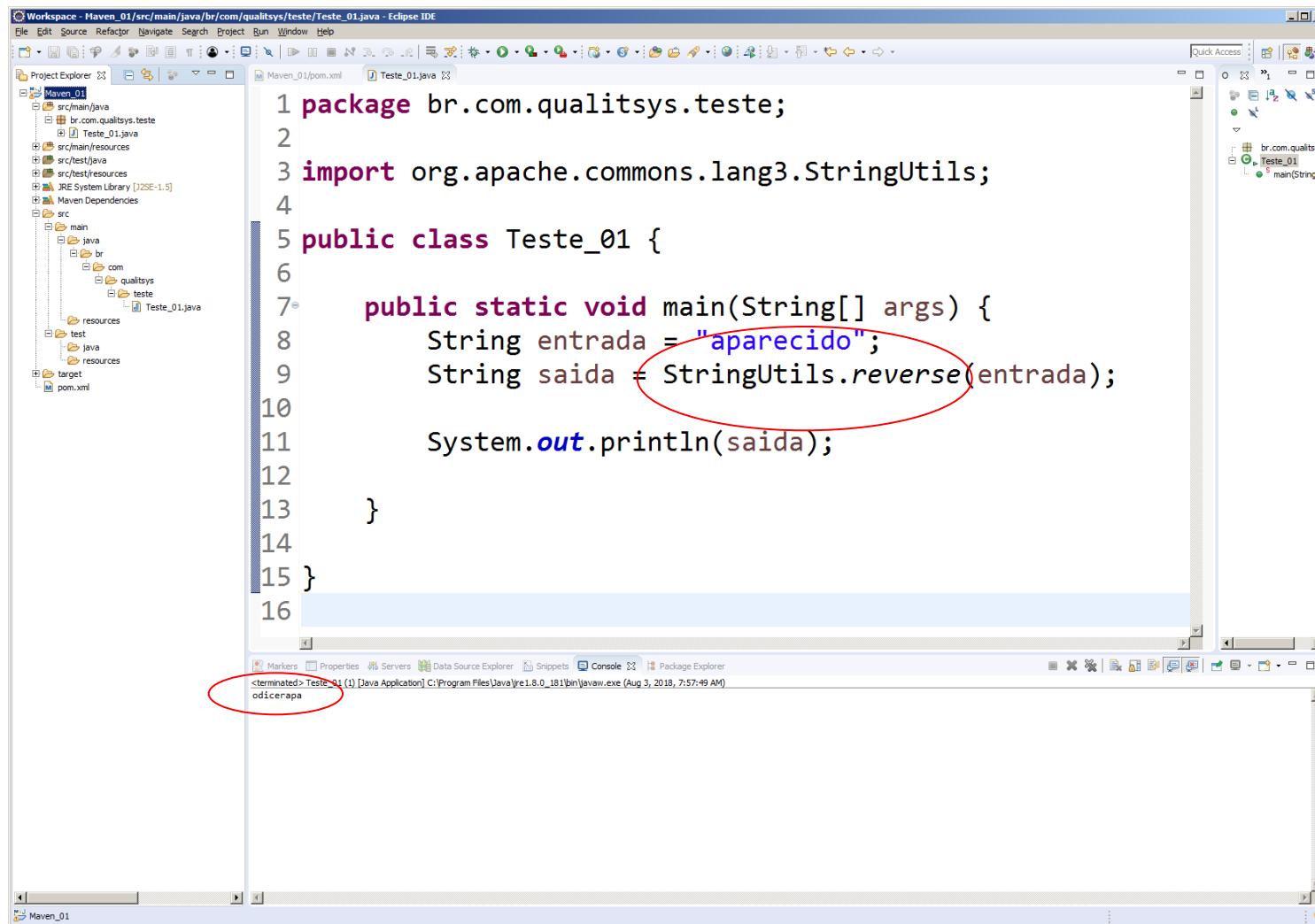
# Adicionando uma dependência

- ✓ O plugin M2E também irá adicionar a biblioteca baixada ao **classpath** do projeto



# Adicionando uma dependência

- ✓ Pronto! Agora podemos utilizar a biblioteca **Commons-Lang**;
- ✓ No projeto, criaremos o package **br.com.qualitsys.teste**;
- ✓ No package, criaremos a classe **Teste\_01** a qual utilizará o método **reverse()** da classe **StringUtils**.



The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "Maven\_01". It includes the "src/main/java" folder containing "br.com.qualitsys.teste" (with "Teste\_01.java") and "src/test/java", "src/test/resources", and "src/main/resources".
- Maven Dependencies:** Shows the dependency tree, including "org.apache.commons:commons-lang3:3.6" under "Maven Dependencies".
- Code Editor:** Displays the Java code for "Teste\_01.java". The code imports "org.apache.commons.lang3.StringUtils" and defines a main method that prints the reverse of the string "aparecido".

```
1 package br.com.qualitsys.teste;
2
3 import org.apache.commons.lang3.StringUtils;
4
5 public class Teste_01 {
6
7     public static void main(String[] args) {
8         String entrada = "aparecido";
9         String saida = StringUtils.reverse(entrada);
10
11         System.out.println(saida);
12     }
13
14
15 }
```
- Console:** Shows the output of the application's execution. The string "aparecido" is reversed to "odicerapa".

```
<terminated> Teste_01 [1] [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe [Aug 3, 2018, 7:57:49 AM]
odicerapa
```