

**Construcción y Pruebas de Software**

IV Ciclo

Laboratorio N° 1

**“Construcción Automática de Proyecto - Maven”**

**Laboratorio: Construcción Automática de Proyecto - Maven**

**OBJETIVOS**

1. Crear proyectos en Java con Maven en Eclipse.
2. Configurar las dependencias de librerías en Maven
3. Identificar los ciclos de vida de Maven

**EQUIPOS, MATERIALES, PROGRAMAS Y RECURSOS**

* PC Personal.
* Sistema operativo Windows XP o superior
* Material disponible desde Tecsup Virtual.
* Conexión a red.
* Eclipse IDE

**SEGURIDAD**

* Colocar las mochilas en el gabinete al final del salón para evitar caídas en caso de sismo.
* No ingresar con bebidas ni comidas.
* Apagar los equipos y los monitores al culminar la sesión.

**INTRODUCCIÓN**

En los proyectos en Java es necesario automatizar el proceso de compilación, pruebas, empaquetado, despliegue y ejecución de aplicaciones, en esta laboratorio se emplea la herramienta Maven para realizar esta tarea.

**PREPARACIÓN**

El Alumno debe revisar previamente el material del curso en Tecsup Virtual y revisar su texto.

**PROCEDIMIENTO y RESULTADOS**

**PARTE I - macOS ( Instalación)**

1. Install brew

|  |
| --- |
| /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)" |

1. Install Maven

|  |
| --- |
| brew install maven |

1. Verificar la instalación de Maven

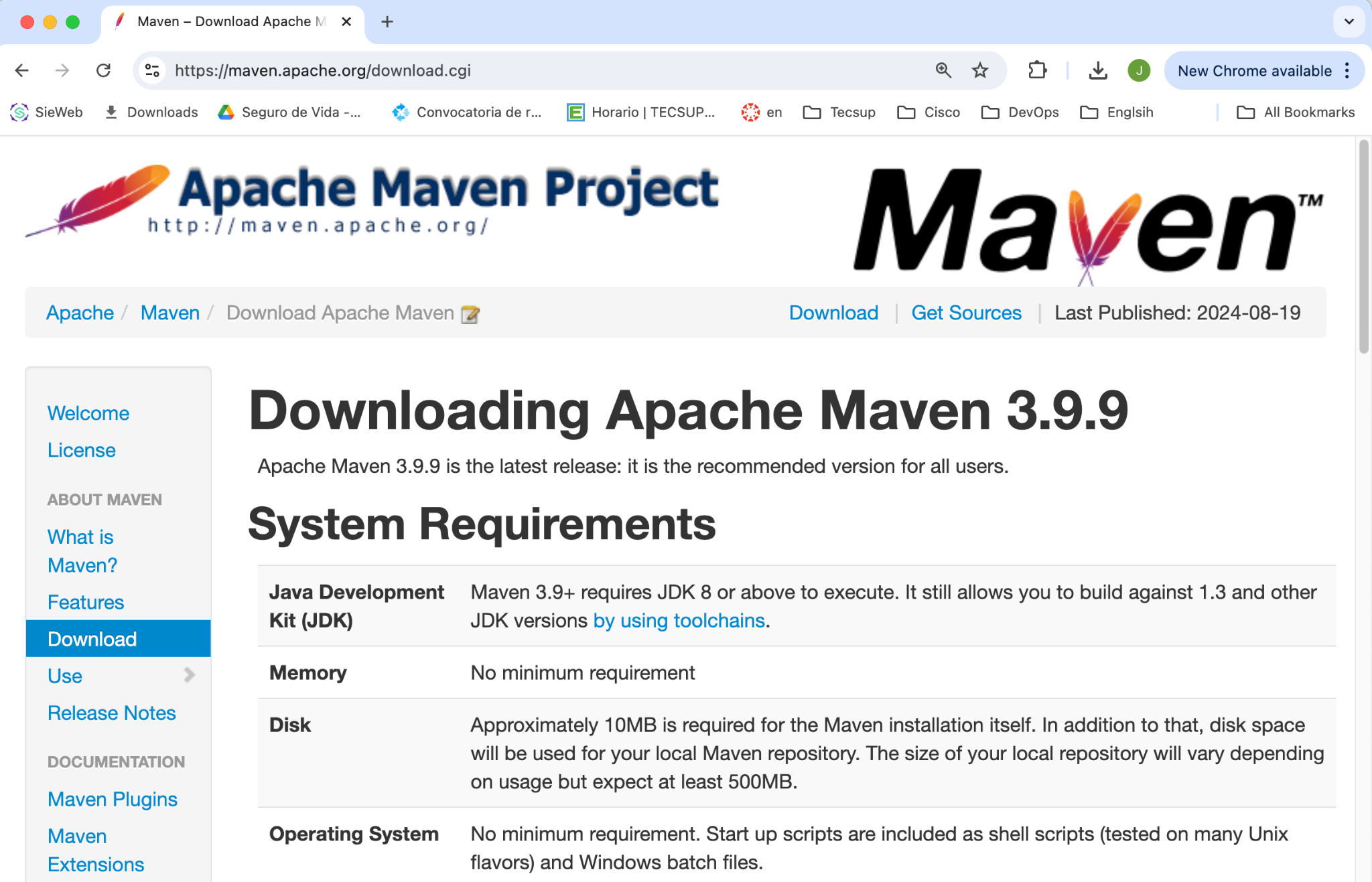
|  |
| --- |
| mvn -version |

**PARTE I - WINDOWS ( Instalación)**

1. Crear un proyecto en Maven desde consola ( el JDK debe esta instalado y la variable JAVA\_HOME creada y la variable PATH debe hacer referencia a la ruta %JAVA\_HOME%\bin)
   1. Instalar Maven

Bajar el instalador

<https://maven.apache.org/download.cgi>



Descomprimirlo en la carpeta D:\bin ( se puede usar otra carpeta)



* + - Configurar la variable de entorno MAVEN\_HOME que tenga como valor la ruta de la carpeta donde se ha descomprimido los instaladores de Maven

Ejemplo

MAVEN\_HOME = D:\bin\apache-maven-3.9.9

* + - Incluir en la variable de entorno PATH la ruta %MAVEN\_HOME%\bin

Una vez finalizada los pasos anteriores abrir una consola de comandos y ejecutar el comando

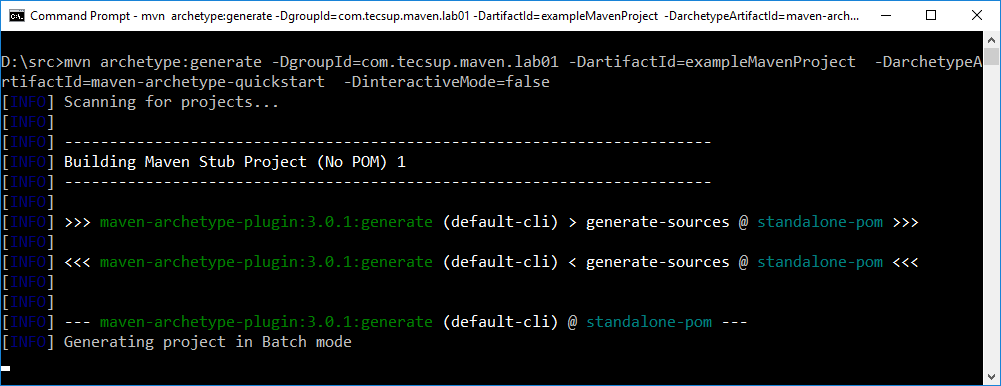
mvn -version

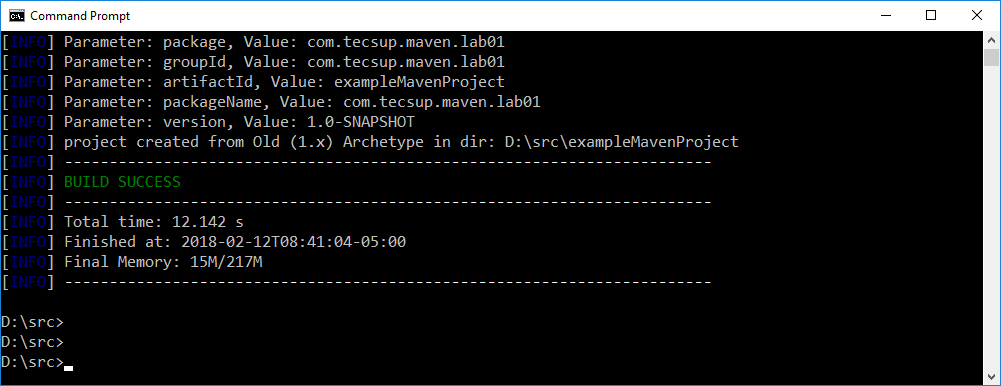


<https://maven.apache.org/download.cgi>

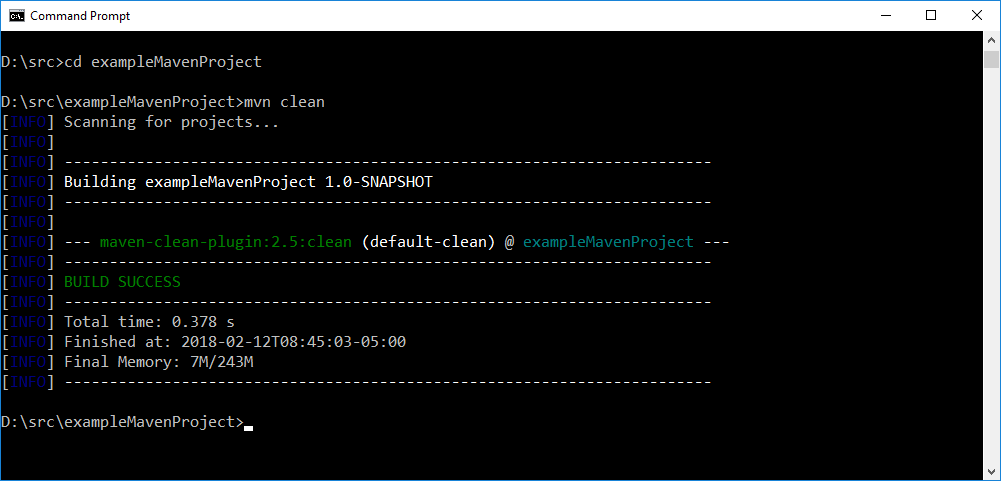
* 1. Crear un proyecto desde línea de comandos usando un artifact en la ruta d:\src

|  |
| --- |
| mvn archetype:generate -DgroupId=pe.edu.tecsup.lab01 -DartifactId=cps\_term\_lab01 -DarchetypeArtifactId=maven-archetype-quickstart -DarchetypeVersion=1.4 -DinteractiveMode=false |





Ingresar al directorio del proyecto e ingresar el comando mvn clean



Deberá configurar la versión de java en **pom.xml** para compilar

|  |
| --- |
| <properties>  <maven.compiler.source>17</maven.compiler.source>  <maven.compiler.target>17</maven.compiler.target>  </properties> |

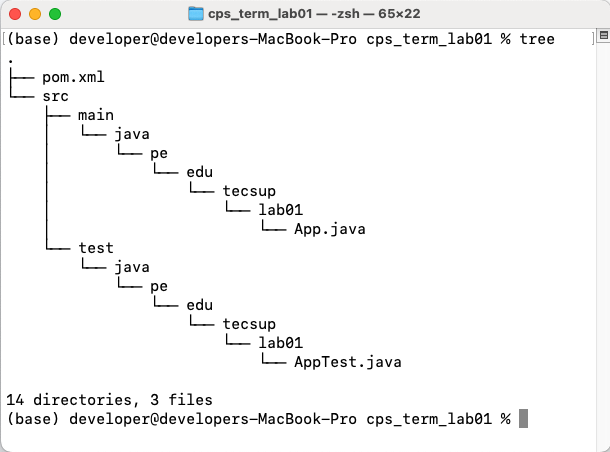
* 1. Ejecute el con el comando “**mvn compile**” , ¿Qué cambios puede observar en la estructura del proyecto?

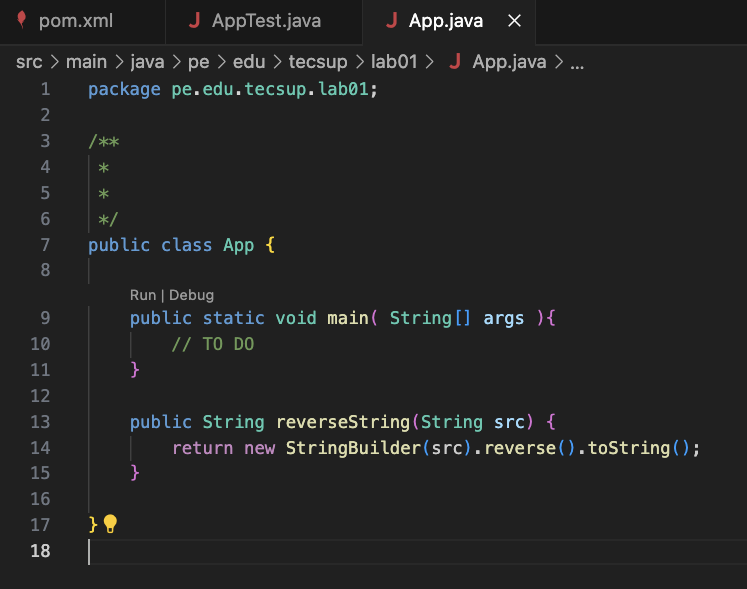
* 1. Ejecute el con el comando “**mvn clean**” , ¿Qué cambios puede observar en la estructura del proyecto?
  2. Ejecute el con el comando “**mvn test**” , ¿Qué cambios puede observar en la estructura del proyecto?
  3. Ejecute el con el comando “**mvn clean**” , ¿Qué cambios puede observar en la estructura del proyecto?

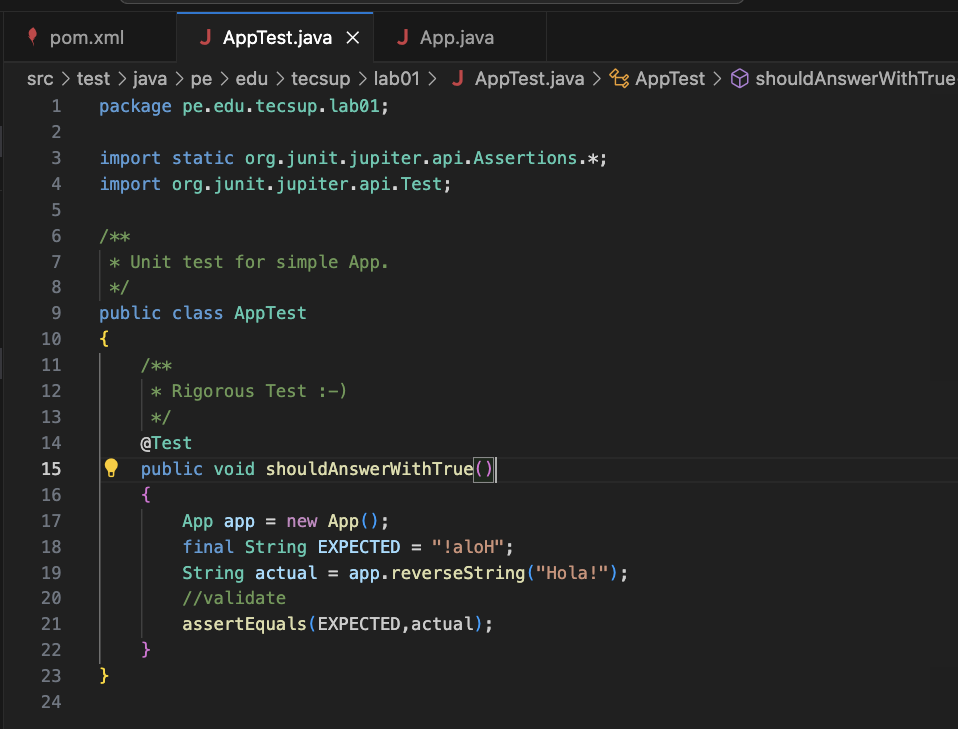
* 1. Coloque la ruta de su repositorio local de Maven
  2. Ejecute el con el comando “**mvn install**” , ¿Qué cambios puede observar en la estructura del proyecto y en el repositorio local de Maven?
  3. Ejecute el con el comando “**mvn clea**n” , ¿Qué cambios puede observar en la estructura del proyecto y en el repositorio local de Maven?
  4. Crear un nuevo proyecto con el archetype usado en la parte b

**PARTE II ( Usando un terminal)**

1. Realizar los siguientes cambios en el código ( pueden usar cualquier editor)



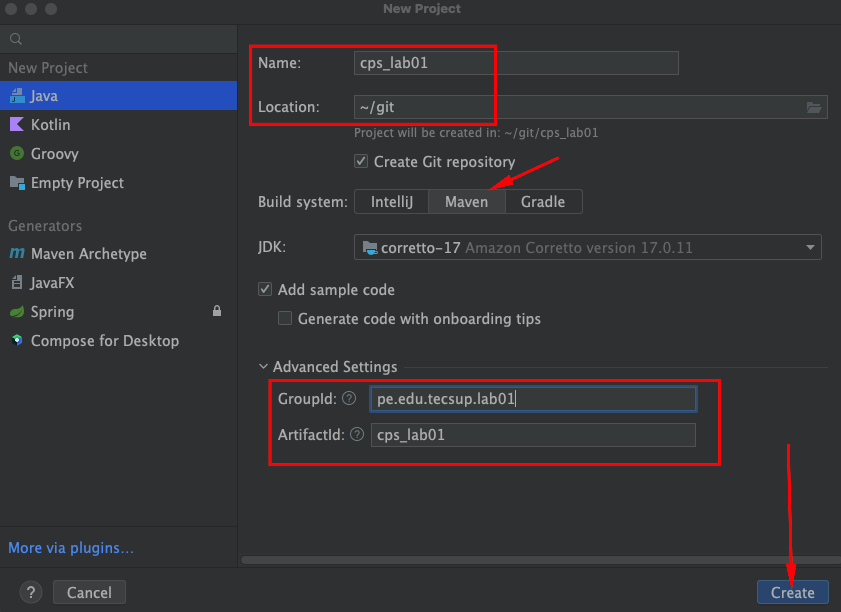




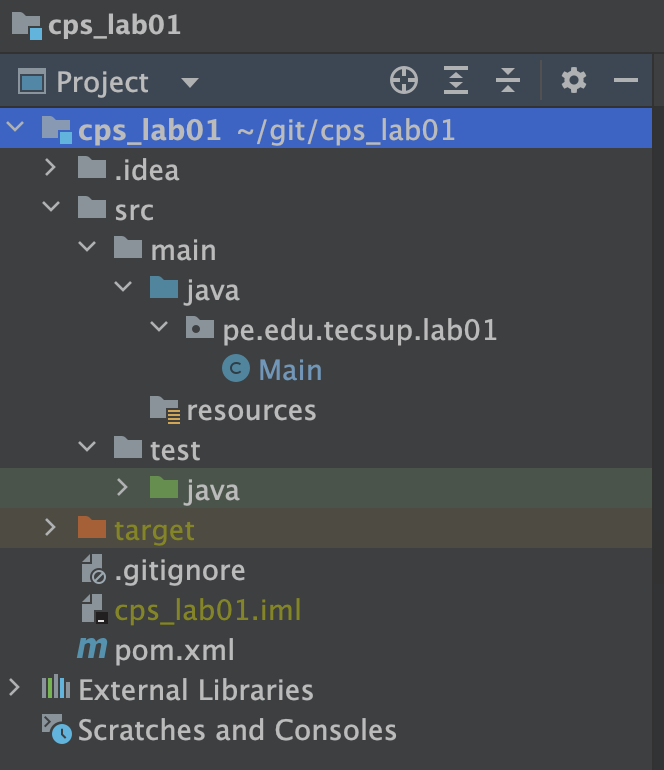
1. Compilar, testear, empaquetar e instalar el proyecto. Adjuntar evidencias
2. Realice cambios en la funcionalidad del método ***reverseString***, verifique si la prueba detecta el cambio de funcionalidad, adjuntar evidencias.

**PARTE III (Usando un IDE)**

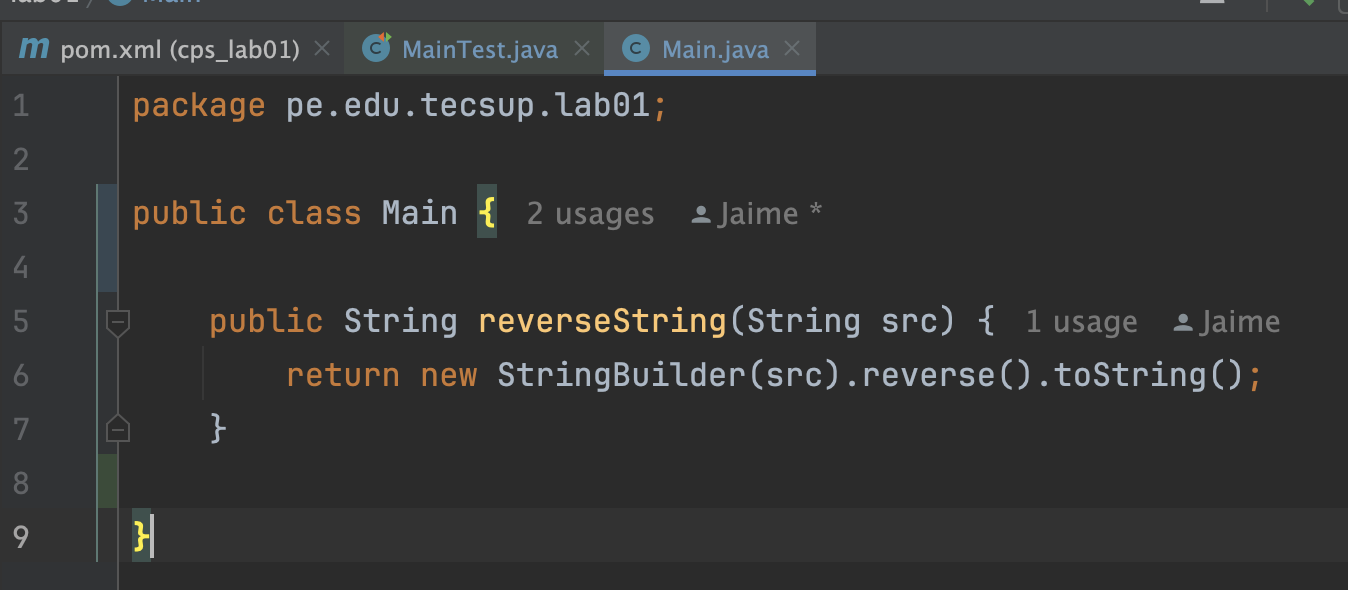
1. Crear el siguiente proyecto en Intellij IDEA



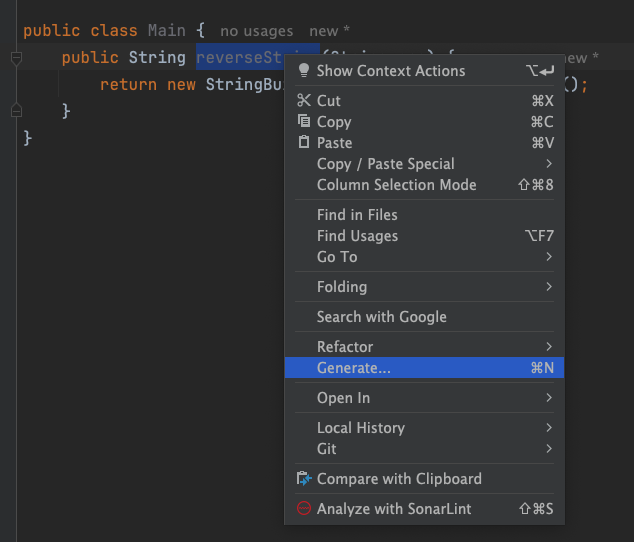
1. La estructura creada es la siguiente

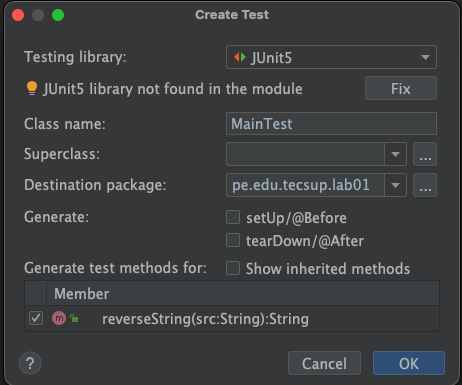


1. Modificar la clase Main.java



1. Generar la clase MainTest.java



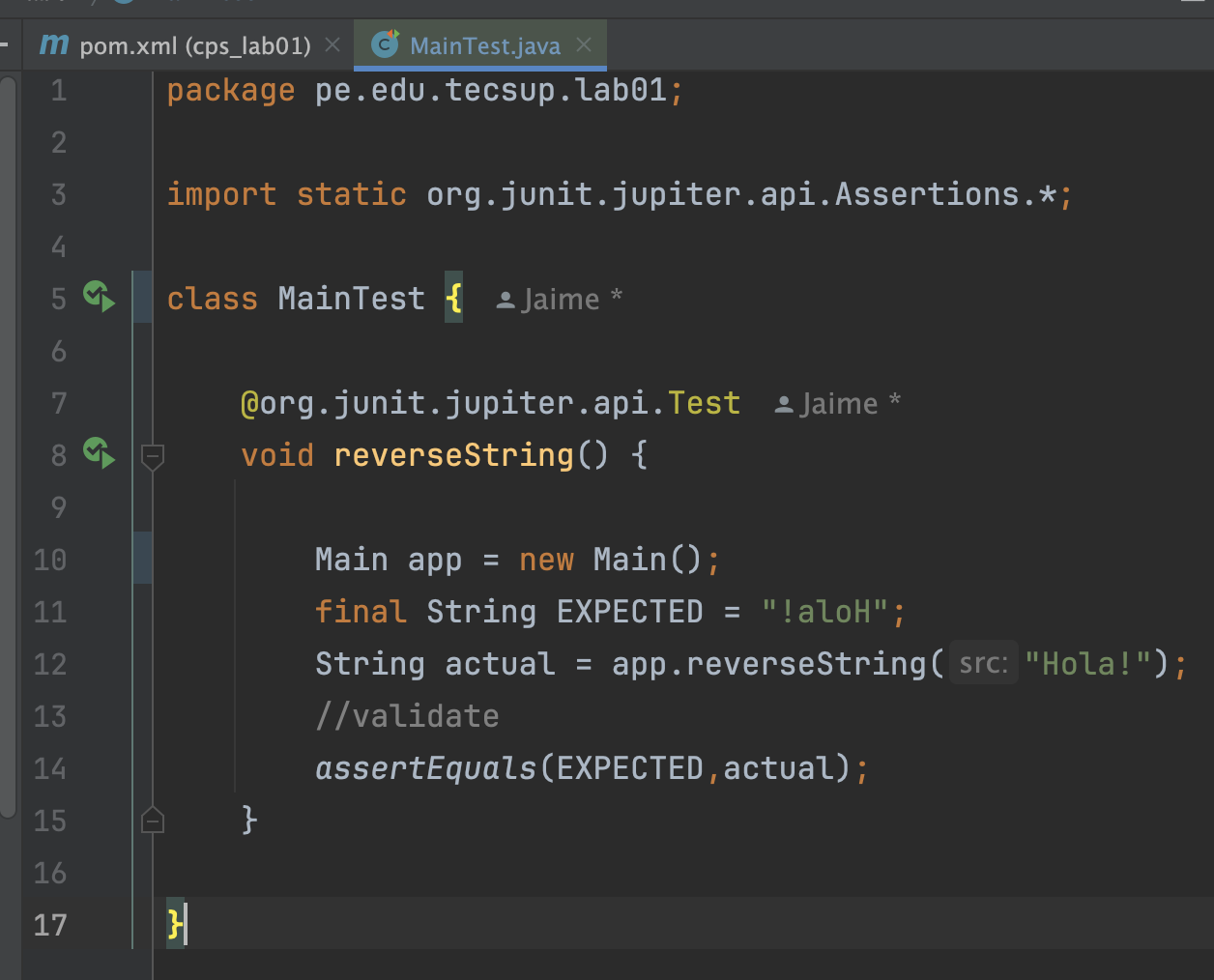


1. Realizar los siguientes cambios

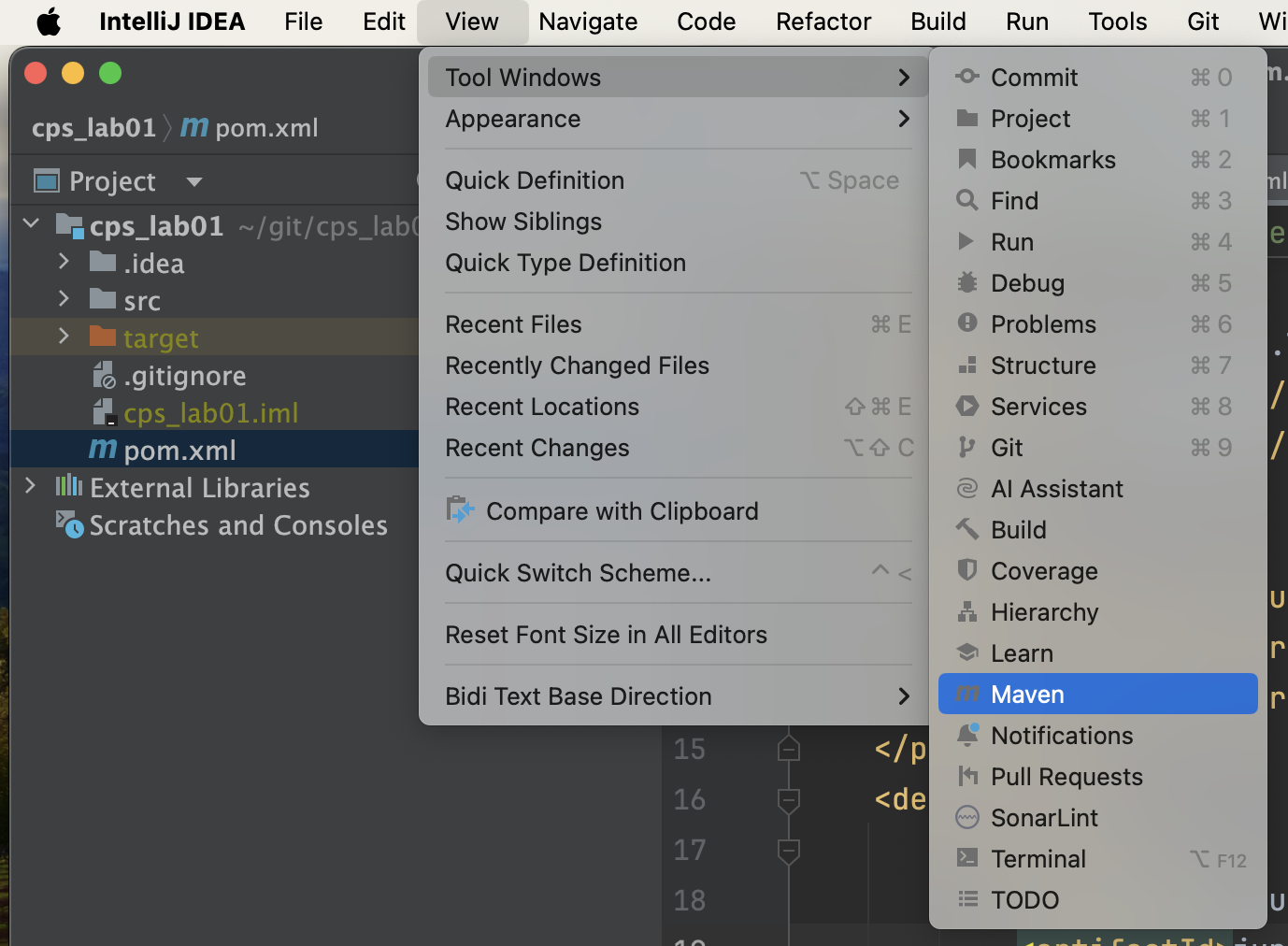
* Verificar el archivo pom.xml ( verificar si las dependencias se han generado correctamente ), en caso de ser necesario, copiarlo las dependencias en el archivo pom.xml de su proyecto. esta marcado de color amarillo.

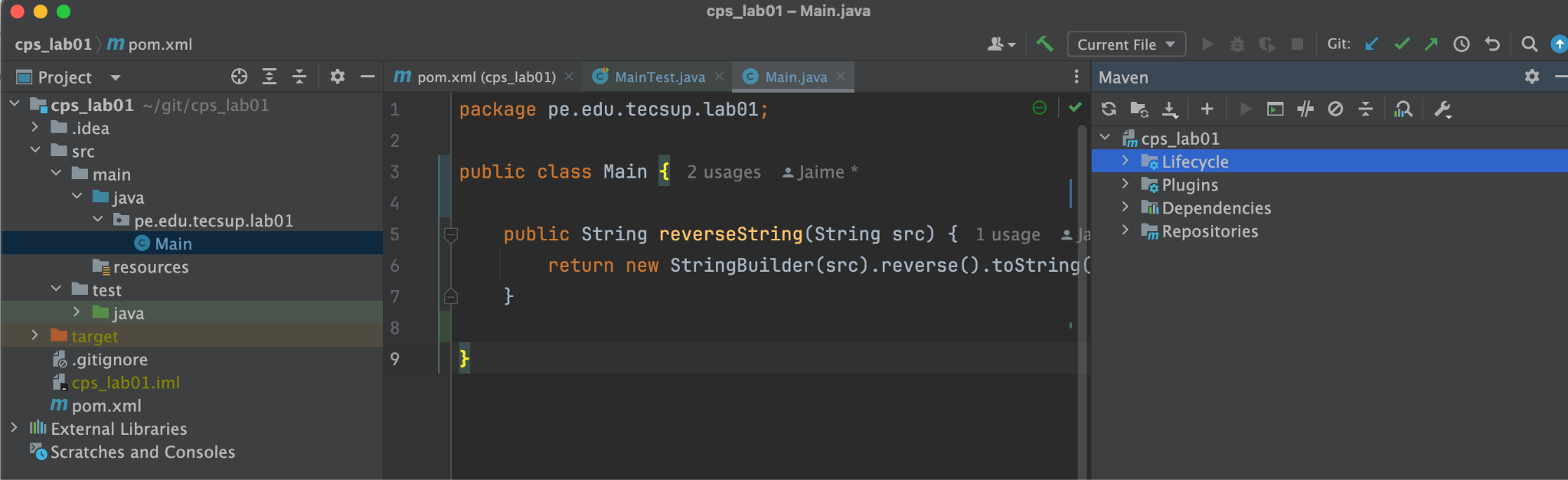
|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <project xmlns="http://maven.apache.org/POM/4.0.0"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  <modelVersion>4.0.0</modelVersion>  <groupId>pe.edu.tecsup.lab01</groupId>  <artifactId>cps\_lab01</artifactId>  <version>1.0-SNAPSHOT</version>  <properties>  <maven.compiler.source>17</maven.compiler.source>  <maven.compiler.target>17</maven.compiler.target>  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  </properties>  <dependencies>  <dependency>  <groupId>org.junit.jupiter</groupId>  <artifactId>junit-jupiter</artifactId>  <version>RELEASE</version>  <scope>test</scope>  </dependency>  </dependencies>  </project> |

* Modificar el archivo MainTest.java



1. Compilar, testear, empaquetar e instalar el proyecto., adjuntar evidencias





1. Realice cambios en la funcionalidad del método ***reverseString***, verifique si la prueba detecta el cambio de funcionalidad, adjuntar evidencias.

**CONCLUSIONES**

Indicar las conclusiones que ha llegado después de desarrollar el laboratorio.

1.

2.

3.

***Criterios de Evaluación***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **logotipo** | | ***Construcción y Pruebas de Software*** | | | | | | |
| **Rúbrica** | | | | | | |
| **Resultado** |  | | | | | | | |
| **Criterio de desempeño** |  | | | | | | | |
| **Curso** | **Construcción Automática de Proyecto - Maven** | | | | | **Periodo** | |  |
| **Actividad** | **Realizar un proyecto Java con Maven** | | | | | **Semestre** | | IV |
| **Nombre del Alumno** |  | | | | | **Semana** | | 1 |
| **Docente** | **Edwin Córdova** | **Fecha** |  | | | **Sección** | |  |
| **Criterios a Evaluar** | | **Excelente** | **Bueno** | **Requiere Mejora** | | **No Aceptable** | | **Puntaje Logrado** |
| Instalación de Maven | | 5 | 4 | 3 | | | 1 |  |
| Creación de un proyecto con Maven - Usando Terminal | | 5 | 4 | 2 | | | 1 |  |
| Creación de un proyecto con Maven - Usando IDE | | 5 | 4 | 3 | | | 1 |  |
| Redacción de conclusiones y observaciones | | 5 | 3 | 2 | | | 1 |  |
| **Total** | | **20** | **15** | | **10** | **04** | |  |

|  |  |
| --- | --- |
| ***Acciones a cumplir*** | ***Menos*** |
| * Puntualidad y dedicación | 1 |
| * Cumplimiento de tiempos establecidos | 1 |
| * Conclusiones: ortografía y redacción. | 1 |
| ***Puntaje Total*** |  |

|  |  |
| --- | --- |
| Comentarios respecto del desempeño del alumno |  |

|  |  |
| --- | --- |
|  | ***Descripción*** |
| Excelente | Demuestra un completo entendimiento del problema o realiza la actividad cumpliendo todos los requerimientos especificados. |
| Bueno | Demuestra un considerable entendimiento del problema o realiza la actividad cumpliendo con la mayoría de los requerimientos especificados. |
| Requiere Mejora | Demuestra un bajo entendimiento del problema o realiza la actividad con pocos de los requerimientos especificados. |
| No aceptable | No demuestra entendimiento del problema o actividad. |