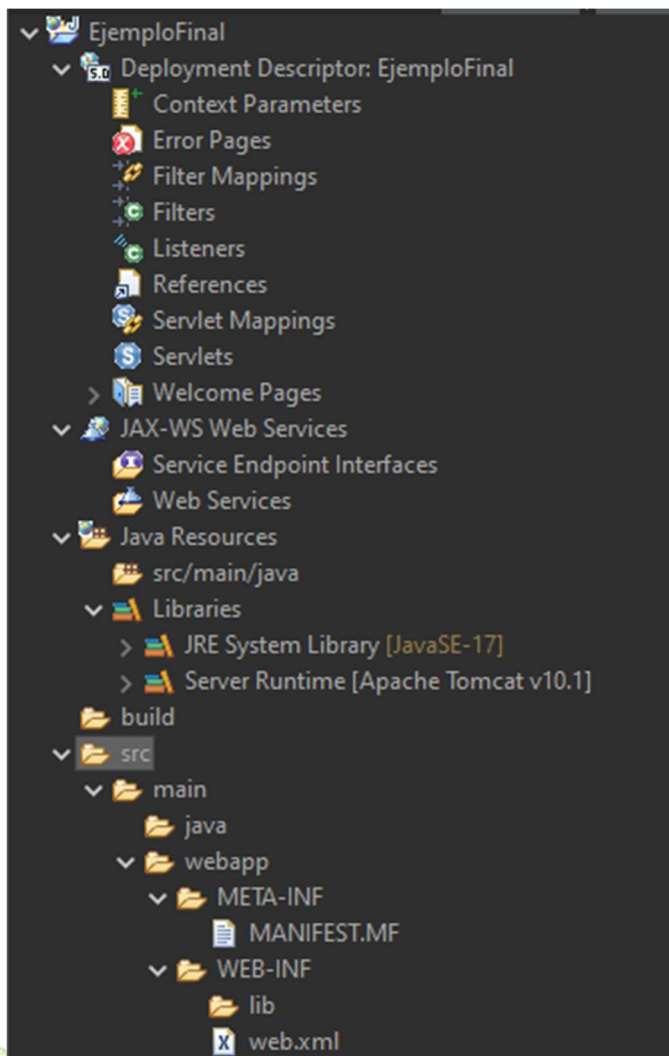


## AD\_T1\_ACT3: Proyecto Maven en Eclipse

- **¿Qué es Maven?**

Apache Maven es una herramienta de gestión y construcción de proyectos ampliamente utilizada en el desarrollo de software. Su principal objetivo es simplificar y automatizar el proceso de construcción, compilación, prueba y distribución de aplicaciones y bibliotecas de software.

- **Crea un proyecto Maven sin arquetipo y muestra que estructura tiene.**



- **¿Qué contiene el archivo pom.xml? Muestra que estructura tiene.**

```
<?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>

  <!-- Información del proyecto -->

  <groupId>com.example</groupId>

  <artifactId>mi-proyecto</artifactId>

  <version>1.0.0</version>

  <name>Mi Proyecto</name>

  <description>Descripción de mi proyecto</description>

  <!-- Configuración del proyecto -->

  <properties>

    <maven.compiler.source>1.8</maven.compiler.source>

    <maven.compiler.target>1.8</maven.compiler.target>

  </properties>

  <!-- Dependencias del proyecto -->

  <dependencies>

    <!-- Ejemplo de dependencia -->

    <dependency>

      <groupId>org.springframework</groupId>

      <artifactId>spring-core</artifactId>

      <version>5.3.10</version>
```

```
</dependency>  
  
</dependencies>  
  
</project>
```

- **¿Qué es una dependencia en Maven? ¿Qué contiene? ¿Dónde la puedes descargar?**

En el contexto de Maven, una "dependencia" se refiere a una biblioteca de código o un conjunto de clases que tu proyecto necesita para compilar, ejecutar o funcionar correctamente. Las dependencias pueden ser componentes de código abierto o bibliotecas de terceros que tu aplicación utiliza para funcionar. Maven utiliza un sistema de gestión de dependencias para administrar estas bibliotecas y asegurarse de que estén disponibles en tu proyecto cuando sean necesarias.

Una dependencia en Maven está definida en el archivo pom.xml (Project Object Model) de tu proyecto. Aquí hay un ejemplo de cómo se define una dependencia en un archivo pom.xml:

xmlCopy code

```
<dependency>                                <groupId>org.springframework</groupId>  
<artifactId>spring-core</artifactId>        <version>5.3.10</version>  
</dependency>
```

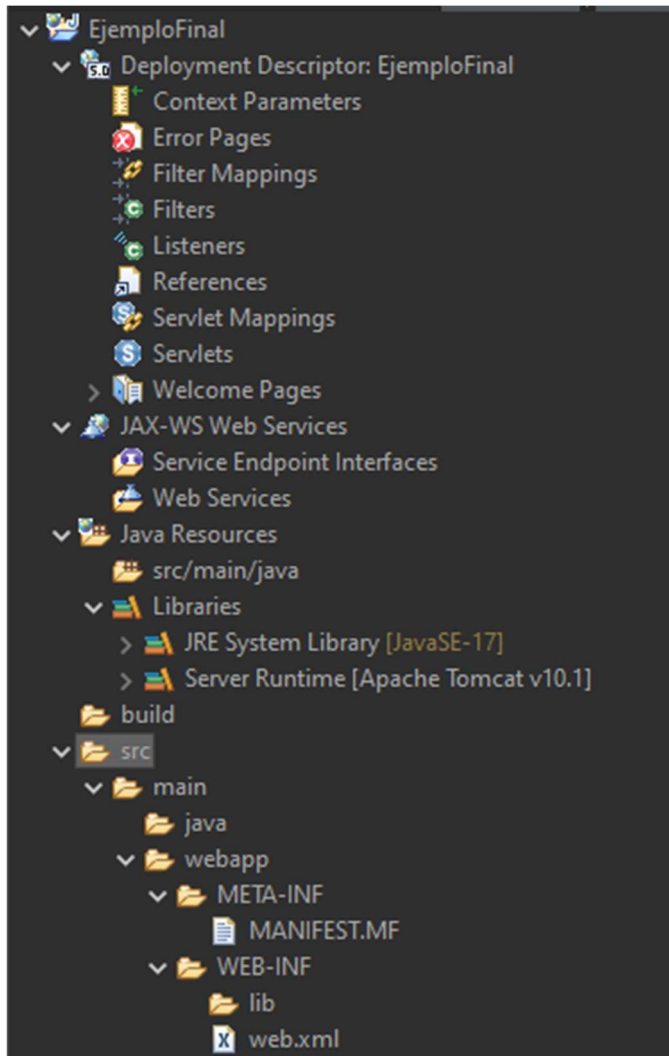
groupId: Identifica al grupo o la organización que proporciona la dependencia.

artifactId: Es el nombre de la dependencia o biblioteca.

version: Indica la versión específica de la dependencia que deseas utilizar.

Los repositorios Maven contienen una amplia gama de dependencias de código abierto y bibliotecas populares que puedes utilizar en tus proyectos. Maven Central Repository es el repositorio central más conocido y utilizado, y muchas bibliotecas de código abierto están disponibles en él. Sin embargo, también puedes configurar repositorios personalizados si necesitas acceder a dependencias específicas que no se encuentran en el repositorio central.

- **Crea un proyecto Maven utilizando el arquetipo para webapp de Maven (1.4) desde Eclipse e incluye la dependencia para trabajar con servlets.**



- He puesto en el pom :
  - <dependencies>
    - <!-- Otras dependencias -->
    - <dependency>
      - <groupId>javax.servlet</groupId>
      - <artifactId>javax.servlet-api</artifactId>
      - <version>4.0.1</version> <!-- O la versión que desees -->
      - <scope>provided</scope>
    - </dependency>
  - </dependencies>
- **Indica y explica que hace referencia cada parte de la estructura del proyecto creado.**



- *Directorio raíz del proyecto*: Este es el directorio principal del proyecto y suele tener el mismo nombre que el proyecto. Contiene archivos de configuración como pom.xml y puede incluir otros recursos de nivel superior.
  - *src/main/java*: Este directorio contiene todo el código fuente de tu proyecto Java, incluyendo tus servlets y clases Java personalizadas. En el caso de una aplicación web, es donde se encuentra la lógica de negocio.
  - *src/main/resources*: Aquí se colocan los archivos de recursos como archivos de configuración, propiedades, plantillas, y otros recursos no Java que tu aplicación pueda necesitar.
  - *src/main/webapp*: Este directorio es crucial para las aplicaciones web. Contiene todos los recursos web estáticos como HTML, CSS, JavaScript, imágenes, y también el archivo WEB-INF/web.xml, que es el descriptor de despliegue de la aplicación web. Dentro de WEB-INF, se encuentra el directorio lib, que es donde se colocan las bibliotecas JAR (dependencias) específicas de tu proyecto.
  - *src/test/java y src/test/resources*: Estos directorios contienen los archivos de código de prueba y recursos para realizar pruebas unitarias y de integración de tu aplicación.
  - *target*: Maven coloca aquí los resultados de la construcción, como el archivo WAR (Web Application Archive) después de compilar el proyecto.
  - *pom.xml*: Este archivo es el núcleo de Maven. Define la configuración del proyecto, incluyendo las dependencias, plugins, y otra información esencial. En este archivo es donde incluirías las dependencias necesarias para trabajar con servlets, como Servlet API.
- 
- **Crea una JSP utilizando las librerías añadidas y prueba el programa creado.**