



Argentina
programa
4.0

Gestión de dependencias

“Desarrollador Java Inicial”

Dependencias de Software

¿Qué es una dependencia?

No siempre queremos desarrollar desde cero toda lógica de nuestro Sistema, sino que intentamos reutilizar componentes ya creados, probados y utilizados por gran parte de la comunidad.

Claro está que esto nos ahorra tiempos y costos.

Dependencias de Software

¿Qué es una dependencia?

- Podríamos decir que una dependencia es una biblioteca hecha por terceros que voy a tener disponible para utilizar en mi proyecto.

Gestores de Dependencias

¿Qué es un Gestor de Dependencias?

- Un Gestor de Dependencias es un Sistema que se encarga, entre otras cosas, de gestionar todas las dependencias de un proyecto de Software para que no tengamos que hacerlo nosotros mismos de forma manual.
- Si lo hiciéramos de forma manual tendríamos que encargarnos de mantener actualizadas todas las bibliotecas y las bibliotecas de las cuales éstas dependen.

Gestores de Dependencias



Maven - ¿Qué nos resuelve?

- Creación automática de los proyectos para poder importarlos desde un IDE.
- Automatización de tareas de compilación/ejecución de test/cobertura, etc.
- Manejo de las dependencias de nuestro proyecto.

Maven - ¿Cómo funciona?

En el directorio raíz del proyecto se deberá crear un archivo con el nombre
pom.xml

En su versión más básica se parecerá a esto:

```
<project>  
  <modelVersion>4.0.0</modelVersion>  
  <groupId>nombre-proyecto</groupId>  
  <artifactId>nombre-proyecto</artifactId>  
  <version>1</version>  
</project>
```

Maven - ¿Cómo usarlo?

¿Cómo uso Maven para importar mi proyecto desde Eclipse o IntelliJ?

Maven - ¿Cómo usarlo?

Para Eclipse: **mvn eclipse:clean**

Para IntelliJ: **mvn idea:clean**

Sirve para borrar cualquier configuración del proyecto anterior.

Maven - ¿Cómo usarlo?

Para Eclipse: **mvn eclipse:eclipse**

Para IntelliJ: **mvn idea:idea**

Crea la configuración del proyecto.

Maven - ¿Cómo usarlo?

Para Eclipse: `mvn eclipse:eclipse -DdownloadSources=true
-DdownloadJavadocs=true`

Para IntelliJ: `mvn idea:idea -DdownloadSources=true
-DdownloadJavadocs=true`

Crea la configuración del proyecto y baja el código fuente de las bibliotecas utilizadas.

Maven – pom.xml

```
<project>
  <modelVersion>4.0.0</modelVersion>
  <groupId>proyecto</groupId>
  <artifactId>proyecto</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <build>
    <plugins>
      <plugin>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.1</version>
        <configuration>
          <source>1.8</source>
          <target>1.8</target>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>
```

Maven - ¿Cómo usarlo?

- `mvn clean`: Borra todos los archivos compilados que existan.
- `mvn compile`: Compila todo el código fuente.

Maven - ¿Cómo usarlo?

- ***mvn test***: Primero compila todo el código fuente si no estuviera compilado y luego ejecuta los tests.
- ***mvn install***: Ejecuta todos los pasos anteriores (compile, test, etc) y además junta el código compilado en un archivo .jar y deja la biblioteca disponible en mi sistema para que la pueda utilizar algún otro proyecto.

Maven - ¿Cómo agrego dependencias?

Las dependencias en Maven se agregan en el archivo pom.xml

Maven - ¿Cómo agrego dependencias?

```
<project>
```

```
.....
```

```
<!-- Dependencias -->
```

```
<dependencies>
```

```
    <dependency>
```

```
        <groupId>junit</groupId>
```

```
        <artifactId>junit</artifactId>
```

```
        <version>4.12</version>
```

```
    </dependency>
```

```
</dependencies>
```

```
.....
```

```
</project>
```


Maven - ¿Cómo agrego dependencias?

- Si ya habías importado el proyecto en Eclipse, acordate de hacer ***mvn eclipse:eclipse*** y refrescar el proyecto para que tome los cambios.
- Si ya habías importado el proyecto en IntelliJ, acordate de hacer ***mvn idea:idea*** y refrescar el proyecto para que tome los cambios.



Argentina
programa
4.0

¡Gracias!
