Argentina programa 4.0

Gestión de dependencias

"Desarrollador Java Inicial"

Dependencias de Software

Argentina programa 4.0

¿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

Argentina programa 4.0

¿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

Argentina programa 4.0

¿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

Argentina programa 4.0

Maven - ¿Qué nos resuelve?

Argentina programa 4.0

- 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?

Argentina programa 4.0

Un Project Object Model o POM es la unidad fundamental de trabajo en Maven.

Es un archivo XML que contiene información sobre el proyecto y los detalles de configuración utilizados por Maven para construir el proyecto.

Contiene valores por defecto para la mayoría de los proyectos. Algunos ejemplos son el directorio de construcción, que es target; el directorio fuente, que es src/main/java; el directorio fuente de prueba, que es src/test/java; y así sucesivamente.

Cuando se ejecuta una task o un goal, Maven busca el POM en el directorio actual. Lee el POM, obtiene la información de configuración necesaria, y luego ejecuta el goal.

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

En su versión más básica se parecerá a:

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

Maven - ¿Cómo funciona?

Argentina programa 4.0

Elementos de un archivo POM

project: este es el elemento de nivel superior en todos los archivos pom.xml de Maven.

modelVersion: indica qué versión del modelo de objetos está utilizando este POM. La versión del modelo en sí cambia con muy poca frecuencia, pero es obligatorio para garantizar la estabilidad de uso en caso de que los desarrolladores de Maven consideren necesario cambiar el modelo.

groupId: este elemento indica el identificador único de la organización o grupo que ha creado el proyecto. El groupId es uno de los identificadores clave de un proyecto y suele basarse en el nombre de dominio completo de la organización. Por ejemplo org.apache.maven.plugins es el groupId designado para todos los plugins de Maven.

artifactId: nombre base único del artefacto primario que está siendo generado por este proyecto. El artefacto primario para un proyecto es típicamente un archivo JAR. Los artefactos secundarios, como los paquetes de fuentes, también utilizan el artifactId como parte de su nombre final. Un artefacto típico producido por Maven tendría la forma. (por ejemplo, myapp-1.0.jar).

version: este elemento indica la versión del artefacto generado por el proyecto. Maven te ayuda mucho con la gestión de versiones y a menudo verás el designador SNAPSHOT en una versión, que indica que un proyecto está en estado de desarrollo.

name: nombre utilizado para mostrar el proyecto. Se utiliza a menudo en la documentación generada por Maven.

url: este elemento indica dónde se puede encontrar el sitio del proyecto. Se utiliza a menudo en la documentación generada por Maven.

properties: este elemento contiene marcadores de posición de valores accesibles en cualquier lugar dentro de un POM.

dependencies: los hijos de este elemento listan las dependencias. La piedra angular del POM.

build: Este elemento maneja cosas como declarar la estructura de directorios de tu proyecto y manejar plugins.

Maven - ¿Cómo usarlo?

Argentina programa 4.0

Si contamos con Maven instalado en nuestro sistema podemos ejecutar:

```
mvn archetype:generate -DgroupId=com.mycompany.app -DartifactId=my-app
-DarchetypeArtifactId=maven-archetype-quickstart -DarchetypeVersion=1.4 -DinteractiveMode=false
```

```
my-app
|-- pom.xml
`-- src
    |-- main
    |   `-- java
    |       `-- com
    |           `-- mycompany
    |               `-- app
    |                   `-- App.java
    `-- test
        `-- java
            `-- com
                `-- mycompany
                    `-- app
                        `-- AppTest.java
```

Maven - ¿Cómo usarlo?

Argentina programa 4.0

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

Maven - ¿Cómo usarlo?

Argentina programa 4.0

Para Eclipse: **mvn eclipse:clean**

Para IntelliJ: **mvn idea:clean**

Sirve para borrar cualquier configuración del proyecto anterior.

Maven - ¿Cómo usarlo?

Argentina programa 4.0

Para Eclipse: **mvn eclipse:eclipse**

Para IntelliJ: **mvn idea:idea**

Crea la configuración del proyecto.

Maven - ¿Cómo usarlo?

Argentina programa 4.0

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

Argentina programa 4.0

```
<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?

Argentina programa 4.0

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

Maven - ¿Cómo usarlo?

Argentina programa 4.0

- ***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.

- ***mvn package:*** Toma el código compilado y lo empaqueta en su formato distribuible, como un jar.

Maven - ¿Cómo agrego dependencias?

Argentina programa 4.0

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

Maven - ¿Cómo agrego dependencias?

Argentina programa 4.0

```
<project>
 ……….
 <!-- Dependencias -->
 <dependencies>

    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.12</version>
    </dependency>

  </dependencies>
 ……….
</project>
```

Maven - ¿Cómo agrego dependencias?

Argentina programa 4.0

- 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!