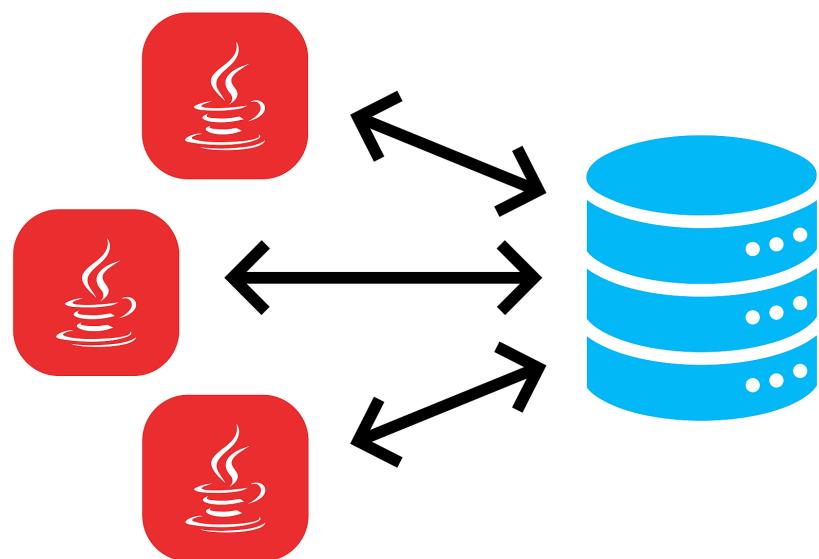


# Taller UD10\_1: Conectores



## 1. Gestión de librerías con **Maven** vs Manualmente (Jar)

- 1. 1. ¿Qué es Maven?
- 1. 2. Beneficios de Usar Maven
- 1. 3. Diferencias con agregar las librerías directamente al proyecto

## 2. Controlador MySql (Manualmente - Jar)

## 3. Controlador MySql (IntelliJ - Maven)

## 4. Actividades

## 5. Fuentes de información

# 1. Gestión de librerías con Maven vs Manualmente (Jar)

## 1.1. ¿Qué es Maven?

Maven es una herramienta de gestión y comprensión de proyectos, principalmente utilizada en proyectos Java. Su principal función es gestionar las dependencias (librerías y otros componentes que tu proyecto necesita), construir el proyecto y gestionar la configuración del ciclo de vida del desarrollo. Maven utiliza un archivo de configuración denominado `pom.xml` (Project Object Model), donde se especifican las dependencias y otros detalles del proyecto.

## 1.2. Beneficios de Usar Maven

1. **Gestión Automática de Dependencias:** Maven descarga automáticamente las dependencias del proyecto desde repositorios remotos y las incluye en el proyecto, evitando la necesidad de descargar y agregar manualmente los archivos JAR.
2. **Estandarización del Proyecto:** Proporciona una estructura estándar para los proyectos, lo que facilita la organización y el mantenimiento.
3. **Reproducibilidad:** El uso de `pom.xml` permite que cualquier desarrollador que clone el repositorio tenga exactamente las mismas versiones de las dependencias, asegurando que el proyecto se construya de manera consistente en diferentes entornos.
4. **Integración con IDEs:** Herramientas como IntelliJ IDEA tienen soporte nativo para Maven, lo que facilita la configuración y gestión del proyecto dentro del IDE.
5. **Gestión del Ciclo de Vida del Proyecto:** Maven puede automatizar tareas como compilación, pruebas, empaquetado y despliegue, facilitando la integración continua.

## 1.3. Diferencias con agregar las librerías directamente al proyecto

**1. Gestión de Dependencias:** en **Maven** las dependencias se especifican en el archivo `pom.xml`. Maven descarga las dependencias automáticamente desde los repositorios configurados. Mientras que en **Manual** las dependencias se descargan manualmente y se agregan al proyecto, lo que puede llevar a inconsistencias y errores si diferentes desarrolladores utilizan versiones distintas de las librerías.

**2. Actualización de Dependencias:** en **Maven** actualizar una dependencia es tan simple como cambiar la versión en el `pom.xml`. Maven se encarga de descargar la nueva versión. Por contra, en **Manual** requiere descargar la nueva versión manualmente y reemplazar los archivos JAR antiguos en el proyecto.

**3. Estructura del Proyecto:** en **Maven** proporciona una estructura de proyecto estandarizada, lo que facilita la comprensión y la navegación del proyecto. Mientras que en **Manual** la estructura del proyecto puede variar según el desarrollador, lo que puede llevar a confusión y desorganización.

**4. Repositorios de Dependencias:** en **Maven** utiliza repositorios remotos (como Maven Central) para buscar y descargar dependencias. También permite configurar repositorios internos. Pero en **Manual** las dependencias deben ser gestionadas y almacenadas manualmente, lo que puede ser engorroso y propenso a errores.

**5. Integración con Herramientas de Construcción:** **Maven** se integra fácilmente con herramientas de construcción y CI/CD, permitiendo la automatización de tareas como compilación, pruebas y despliegue. Por otro lado en **Manual** se requieren scripts personalizados y configuración adicional para integrar tareas de construcción y despliegue.

## 2. Controlador MySql (Manualmente - Jar)

1. El primer paso es descargar desde <https://www.mysql.com/products/connector/> el conector apropiado.

The screenshot shows the MySQL Connectors page. At the top, it says "MySQL provides standards-based drivers for JDBC, ODBC, and .Net enabling developers to build direct connections to MySQL databases from their applications." Below this, there's a section titled "Developed by MySQL" containing links to various drivers:

- ADO.NET Driver for MySQL (Connector/.NET) [Download](#)
- ODBC Driver for MySQL (Connector/ODBC) [Download](#)
- JDBC Driver for MySQL (Connector/J)** [Download](#) (This link is highlighted with a red box.)
- Node.js Driver for MySQL (Connector/Node.js) [Download](#)
- Python Driver for MySQL (Connector/Python) [Download](#)
- C++ Driver for MySQL (Connector/C++) [Download](#)

2. Elegiremos la opción independiente de la Plataforma, que nos provee del archivo jar necesario (comprimido en tar.gz o zip, según nos interese):

### MySQL Community Downloads

◀ Connector/

The screenshot shows the MySQL Community Downloads page for the Connector/J 9.0.0 release. At the top, there are tabs: "General Availability (GA) Releases" (which is selected), "Archives", and a help icon. Below the tabs, it says "Connector/J 9.0.0".

Select Operating System: [Platform Independent](#)

Platform Independent (Architecture Independent), Compressed TAR Archive (mysql-connector-j-9.0.0.tar.gz)	9.0.0	4.3M	<a href="#">Download</a>
Platform Independent (Architecture Independent), ZIP Archive (mysql-connector-j-9.0.0.zip)	9.0.0	5.1M	<a href="#">Download</a>

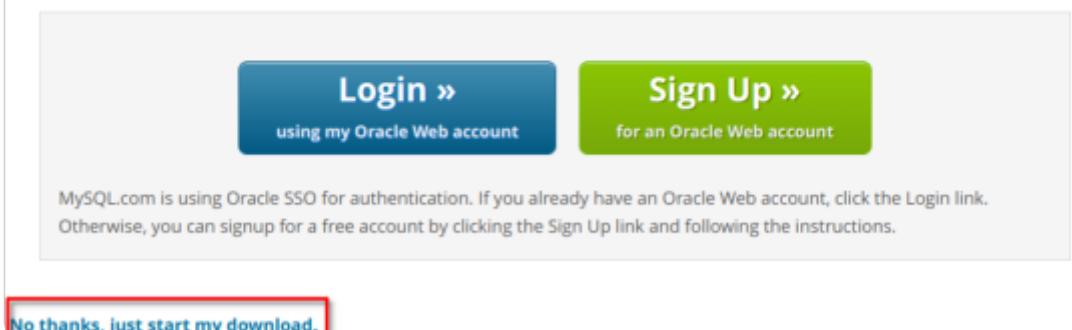
**!** We suggest that you use the [MD5 checksums and GnuPG signatures](#) to verify the integrity of the packages you download.

3. Haz clic en **Download** y selecciona la opción: **No thanks, just start download**

[Login Now](#) or [Sign Up](#) for a free account.

An Oracle Web Account provides you with the following advantages:

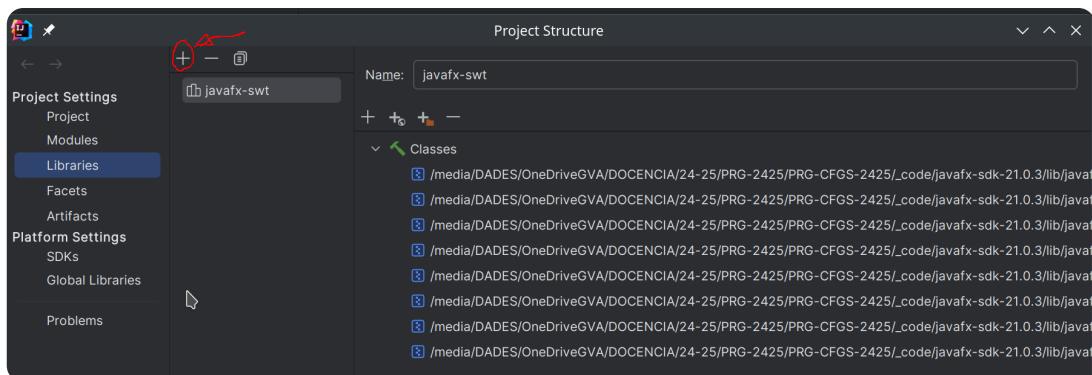
- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system



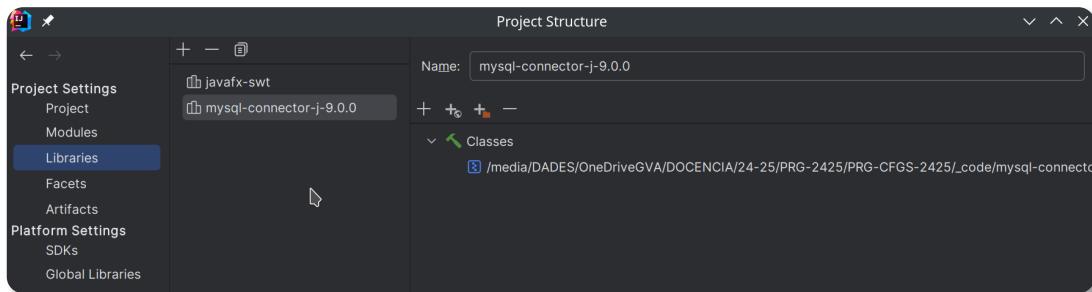
4. Ahora descomprimimos el archivo descargado, que contendrá el archivo jar que nos interesa:

```
mysql-connector-j-9.0.0.jar
```

5. Ahora deberemos añadir la librería JDBC a nuestro proyecto. Este punto dependerá del IDE que estemos usando, en mi caso con IntelliJ seria en [File/Project Structure \(Ctrl+Alt+Shift+S\)](#), y pulsar el botón para añadir la librería.



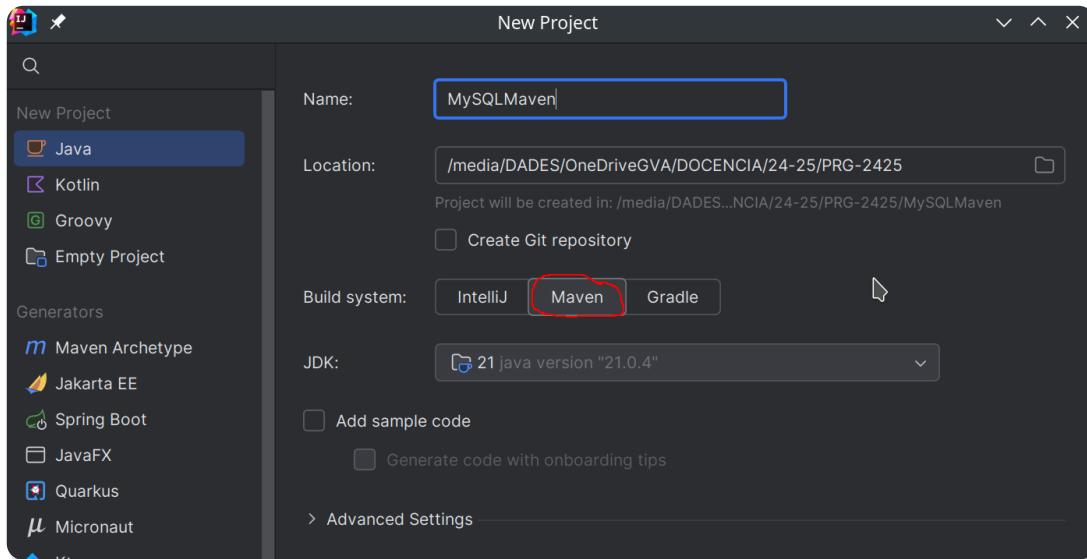
6. Busca en la carpeta descomprimida el archivo y selecciónalo, te informará de que la librería se añadirá a tu proyecto (pulsa OK) y debería quedar algo así:



### 3. Controlador MySQL (IntelliJ - Maven)

#### 1. Crear un Nuevo Proyecto con Maven:

- Al crear un nuevo proyecto en IntelliJ, selecciona "Maven" como sistema de construcción.



- IntelliJ generará la estructura del proyecto y el archivo `pom.xml`.

#### 2. Importar un Proyecto Existente con Maven:

- Si ya tienes un proyecto con un `pom.xml`, simplemente importa el proyecto en IntelliJ y selecciona el archivo `pom.xml`. IntelliJ configurará el proyecto automáticamente.

#### 3. Agregar Dependencias:

- Abre el archivo `pom.xml` y agrega las dependencias necesarias dentro de la sección `<dependencies>`. IntelliJ descargará y configurará automáticamente las dependencias.

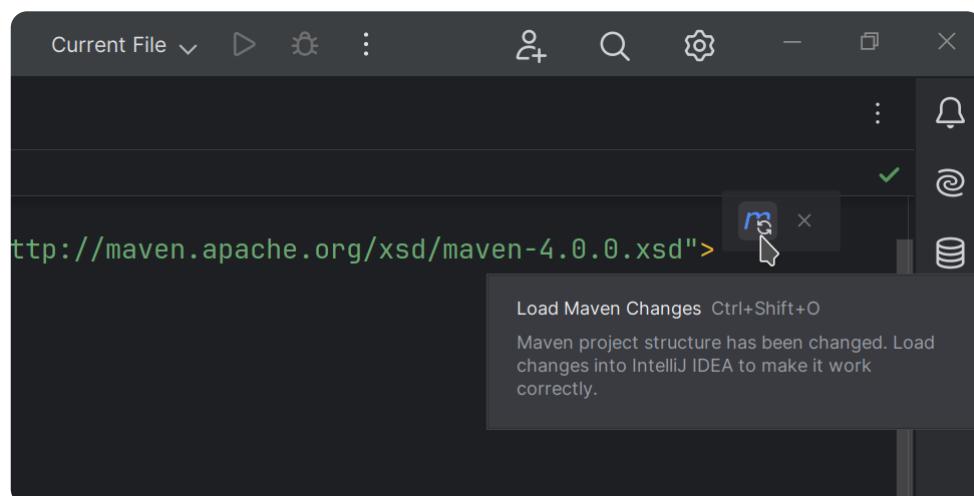
```

1 [...]
2 <dependencies>
3     <!-- Dependencia de MySQL -->
4     <dependency>
5         <groupId>mysql</groupId>
6         <artifactId>mysql-connector-java</artifactId>
7         <version>8.0.33</version>
8     </dependency>
9 </dependencies>
10 [...]

```

#### 4. Actualizar Dependencias:

- Si cambias el archivo pom, IntelliJ mostrará un botón para cargar los cambios:



## 4. Actividades

---

Ahora que conoces las diferencias entre agregar una librería manualmente o mediante Maven, genera un documento pdf con el siguiente contenido:

1. Después de haber probado las dos maneras de agregar el conector MySQL a tu proyecto, explica qué método te ha resultado más difícil. Luego, investiga si en el repositorio de central de Maven, siempre está disponible la última versión del controlador MySQL, crees que esto puede llegar a ser un problema?
2. Explica en tu documento como agregar los controladores (mediante Maven) para bases de datos Oracle (explica las diferencias entre el Ojdbc8 y el Ojdbc11) y SQLite. (Capturas de pantalla y explicaciones, y capturas que demuestren que el proyecto tiene las librerías agregadas).

Envia tu fichero pdf a la tarea de Aules.

## 5. Fuentes de información

---

- [Wikipedia](#)
- [Programación \(Grado Superior\) - Juan Carlos Moreno Pérez \(Ed. Ra-ma\)](#)
- Apuntes IES Henri Matisse (Javi García Jimenez?)
- Apuntes AulaCampus
- [Apuntes José Luis Comesaña](#)
- [Apuntes IOC Programació bàsica \(Joan Arnedo Moreno\)](#)
- [Apuntes IOC Programació Orientada a Objectes \(Joan Arnedo Moreno\)](#)
- [FXDocs](#)
- <https://openjfx.io/openjfx-docs/>
- <https://arturoblasco.github.io/pr>