



# Seminario 1: Acceso a bases de datos en el desarrollo de Sistemas de Información

Diseño y Desarrollo de Sistemas de Información  
GIIMADE - 2024/2025

## Autores

Carmen Azorín Martí  
Adrián Jaén Fuentes  
Rafael Luque Framit  
Pablo Luque Salguero  
Jaime Martínez Bravo  
Juan Pedro Moreno Ruiz

Granada, 10 de noviembre de 2024

## Índice

1. Lenguaje utilizado. ....	1
2. Tareas de instalación llevadas a cabo. ....	1
3. Fuentes de obtención del código. ....	2
4. Reparto del trabajo. ....	2

## 1. Lenguaje utilizado.

El lenguaje utilizado en este seminario ha sido Java. Con este lenguaje hemos implementado la conexión a la base de datos así como el propio trabajo que se pide en el seminario. Como API hemos usado JDBC para la implementación del código.

Para probar que el código estaba bien implementado, hemos usado SQLDeveloper conectado a la base de datos de la ETSIIT.

## 2. Tareas de instalación llevadas a cabo.

Como hemos elegido Java para este proyecto, hemos utilizado Maven como gestor de dependencias. Para instalar Maven es necesario instalar Java previamente. Para ello hemos descargado la versión de Java 21 JDK de <https://adoptium.net/es/temurin/releases/>.

Una vez instalado Java hemos instalado Maven. Para ello hemos seguido las instrucciones oficiales: <https://maven.apache.org/install.html>

Podemos comprobar que ambos están instalados con los siguientes comandos:

```
1 java -version
2 mvn -version
```

Una vez instalados, hemos querido usar VisualStudio Code para gestionar el proyecto, por lo que hemos instalado las extensiones recomendadas por Microsoft para trabajar con proyectos de Java: <https://marketplace.visualstudio.com/items?itemName=vscjava.vscode-java-pack>

Con esto, ya podíamos crear un proyecto de Maven desde Visual Studio Code y agregar las dependencias necesarias. La primera dependencia es la de Oracle JDBC, que se encuentra en los repositorios de Maven y se puede instalar agregando al fichero pom.xml las siguientes líneas:

```
1 <dependency>
2     <groupId>com.oracle.database.jdbc</groupId>
3     <artifactId>ojdbc11</artifactId>
4     <version>23.5.0.24.07</version>
5 </dependency>
```

Otra dependencia que hemos utilizado es dotenv-java. Esta permite el uso de ficheros .env para establecer variables que no queremos incluir en el código fuente de la aplicación, como puede ser el usuario y la contraseña de la base de datos. Para incluir la dependencia, de igual forma que para la anterior podemos añadir lo siguiente al fichero pom.xml:

```
1 <dependency>
2     <groupId>io.github.cdimascio</groupId>
3     <artifactId>dotenv-java</artifactId>
4     <version>3.0.2</version>
5 </dependency>
```

Para ejecutar la aplicación basta con escribir el siguiente comando dentro de la carpeta del proyecto, que descargará e incluirá las dependencias necesarias:

```
1 mvn exec:java
```

### 3. Fuentes de obtención del código.

Para saber cómo usar la API de Java para usar SQL, hemos consultado el siguiente enlace: <https://docs.oracle.com/en/java/javase/11/docs/api/java.sql/java/sql/package-summary.html>

Por otro lado, nos hemos ayudado de Chat GPT para generar parte del código en Java: <https://chatgpt.com>

### 4. Reparto del trabajo.

La forma de completar el Seminario ha sido conectándonos a un servidor de Discord y, a modo de reunión, realizar todos juntos los distintos pasos necesarios. Las decisiones necesarias se tomaron también en grupo, así como la resolución de los distintos problemas que nos encontramos mientras hacíamos el seminario.