

CURSO DE JAVA CON JDBC

EJERCICIO

PROCESO DE INGENIERIA INVERSA



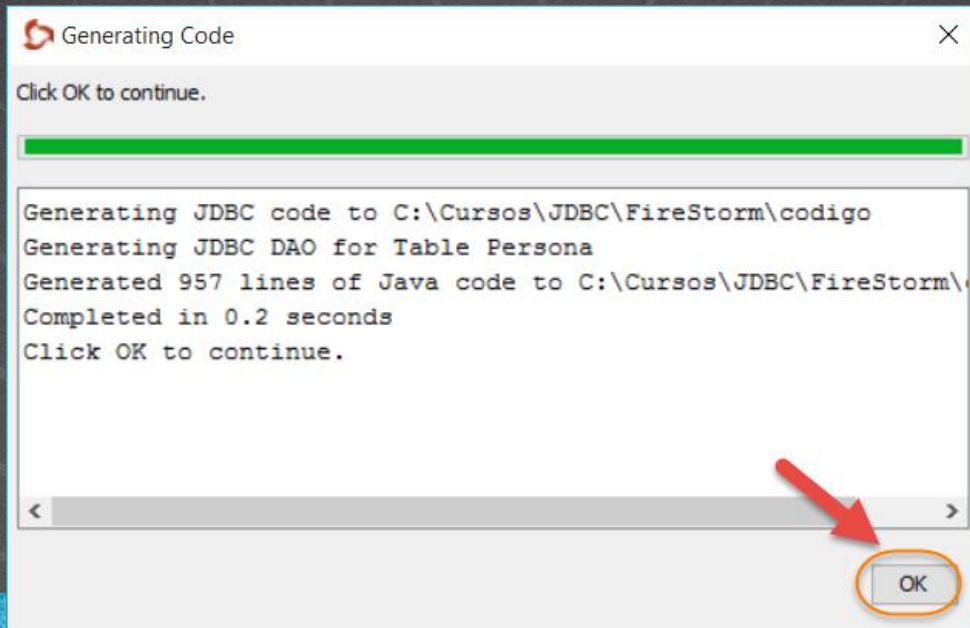
Experiencia y Conocimiento para tu vida

CURSO DE JAVA CON JDBC

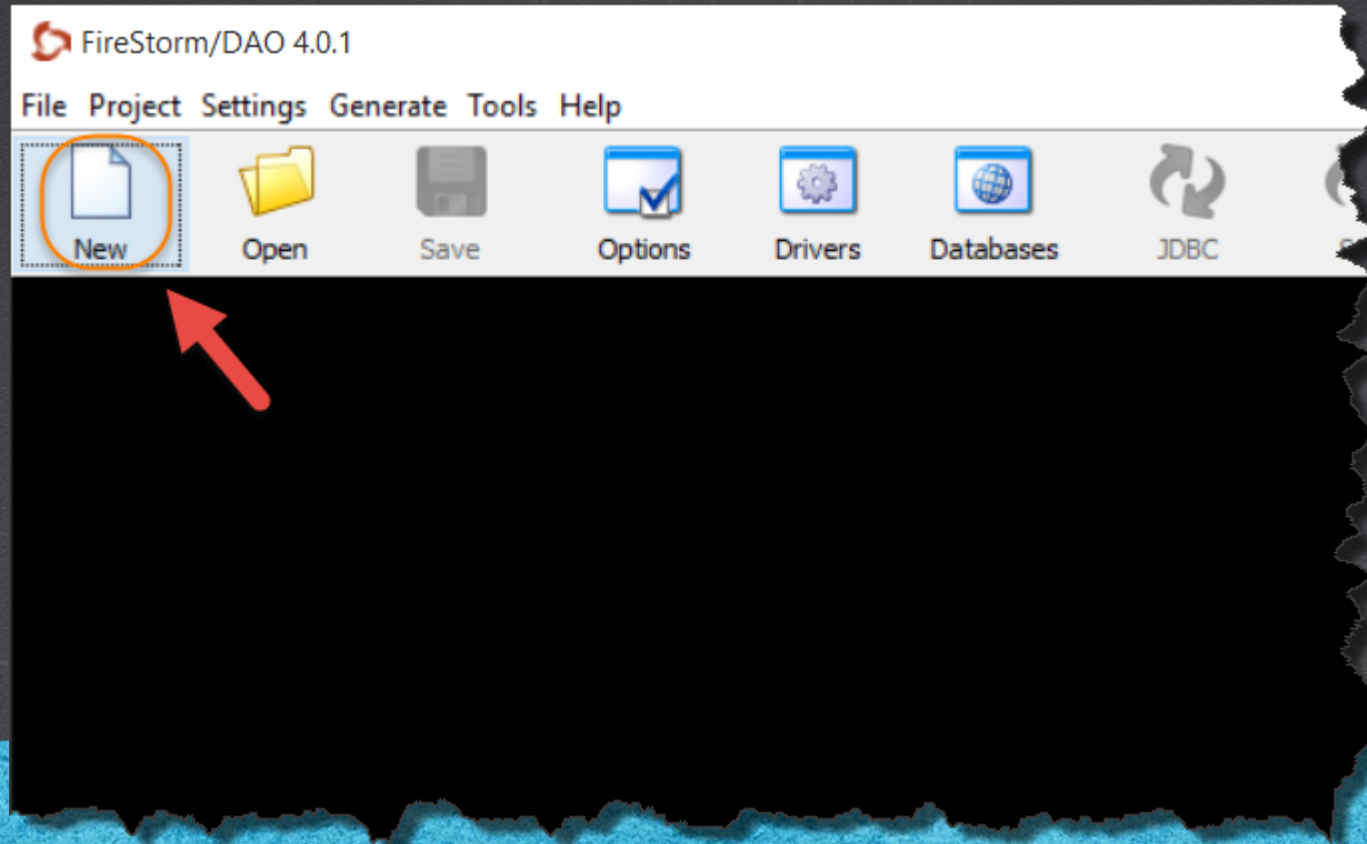
www.globalmentoring.com.mx

OBJETIVO DEL EJERCICIO

Crear un ejercicio donde obtengamos el código de la capa de datos con ayuda de la herramienta FireStorm/Dao.

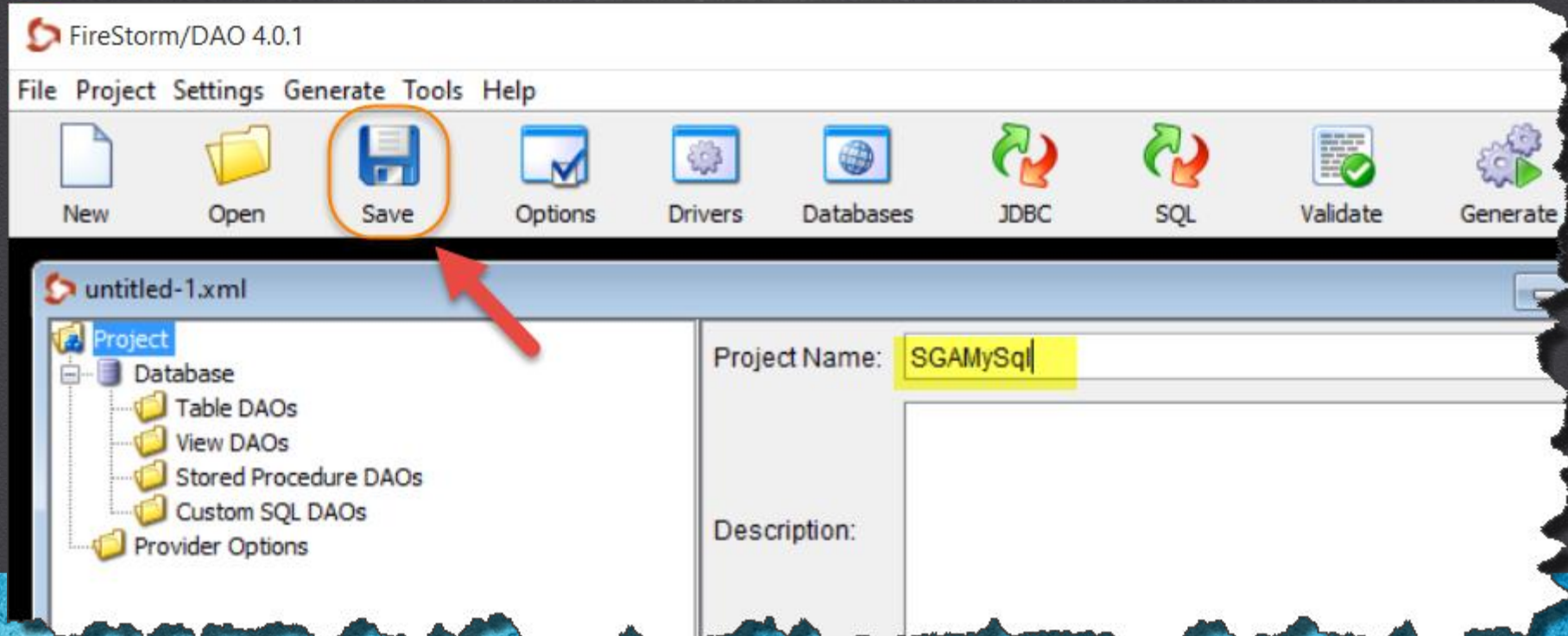


PASO 1. NUEVO PROYECTO DE FIRESTORM/DAO



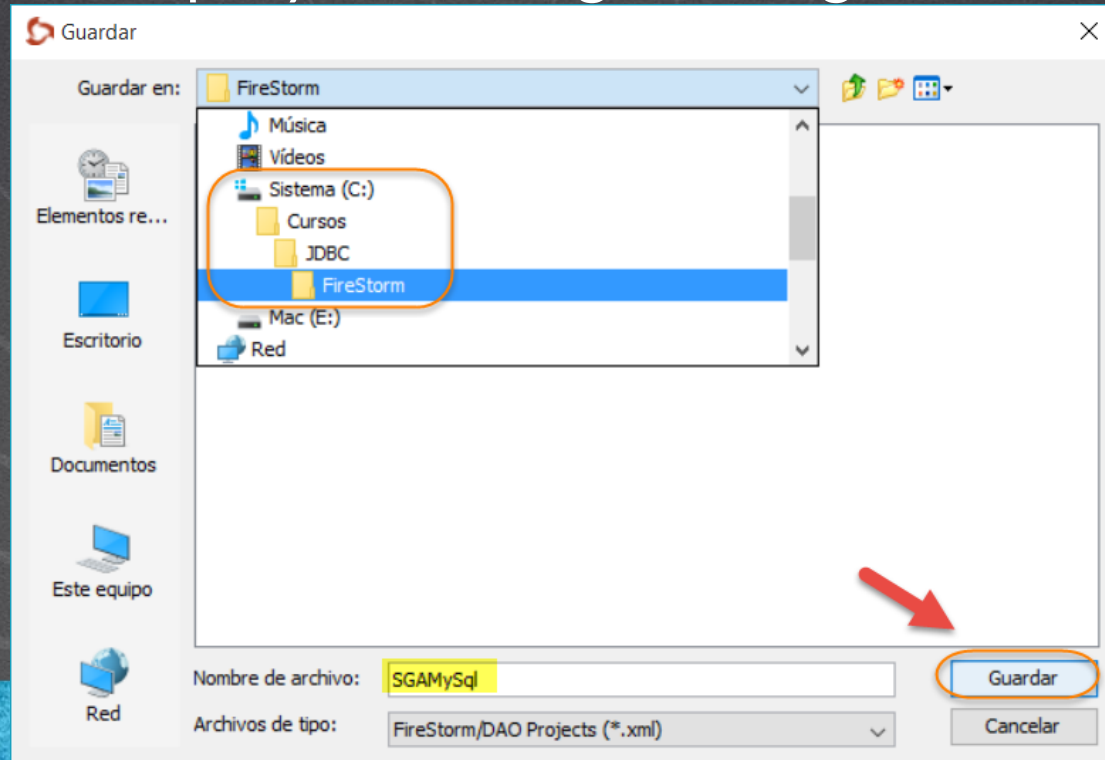
PASO 1. NUEVO PROYECTO DE FIRESTORM/DAO

Guardamos nuestro proyecto:



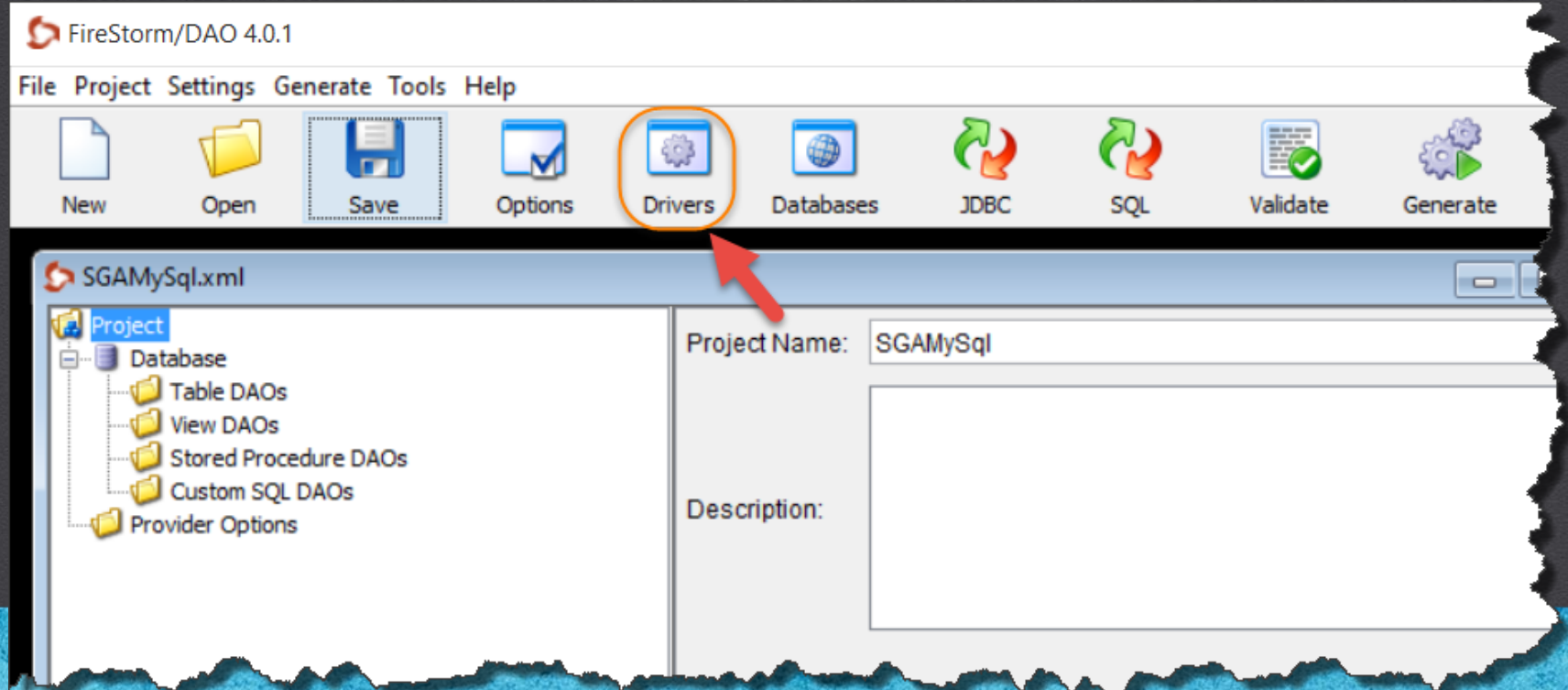
PASO 1. NUEVO PROYECTO DE FIRESTORM/DAO

Guardamos nuestro proyecto escogemos alguna ruta, ej:



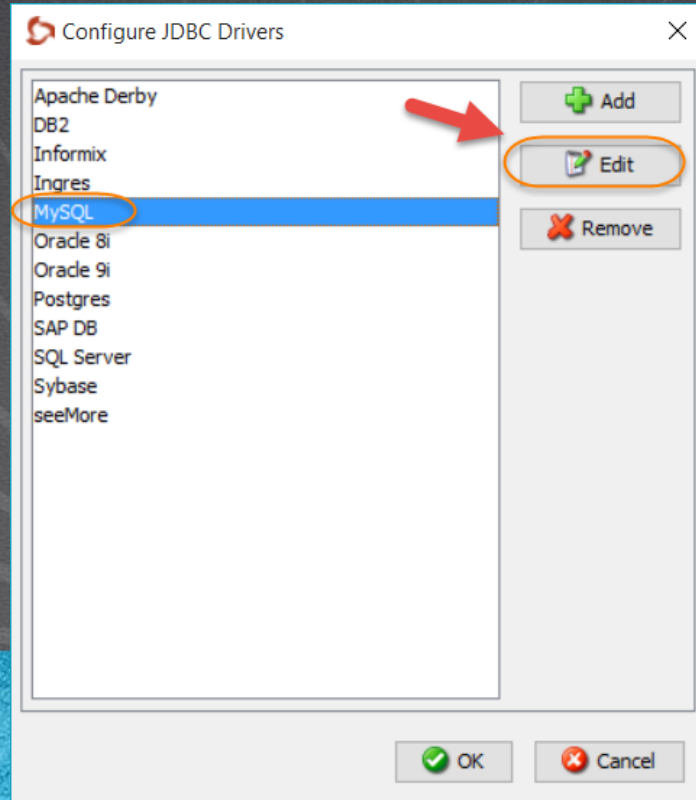
PASO 2. CONFIGURAMOS EL DRIVER DE MYSQL

Configuramos el driver de Mysql:



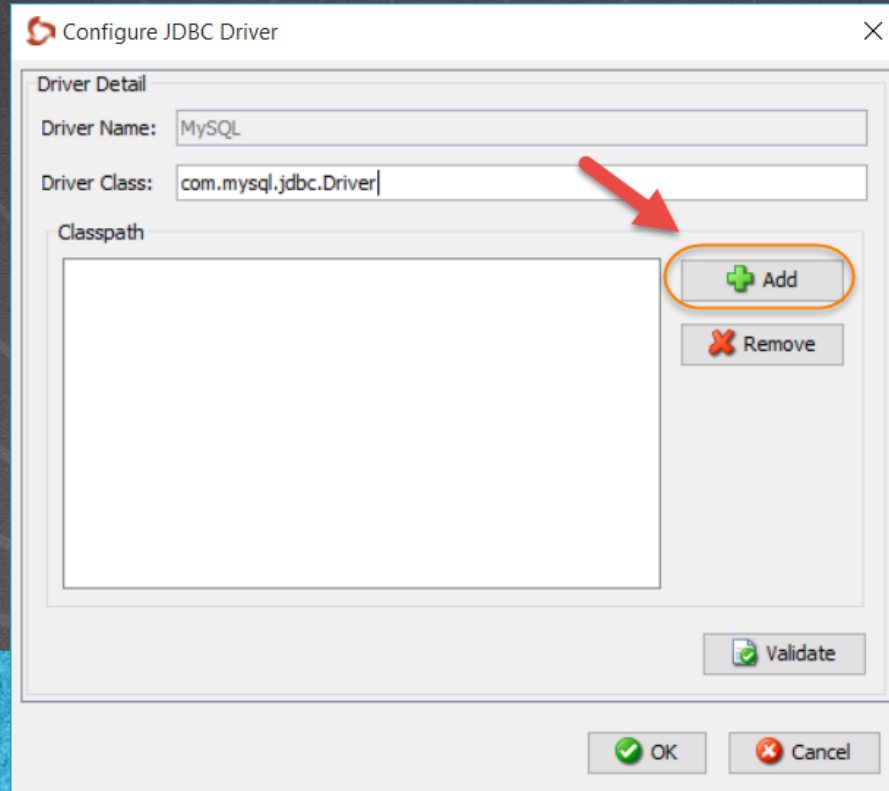
PASO 2. CONFIGURAMOS EL DRIVER DE MYSQL

Configuramos el driver de Mysql:



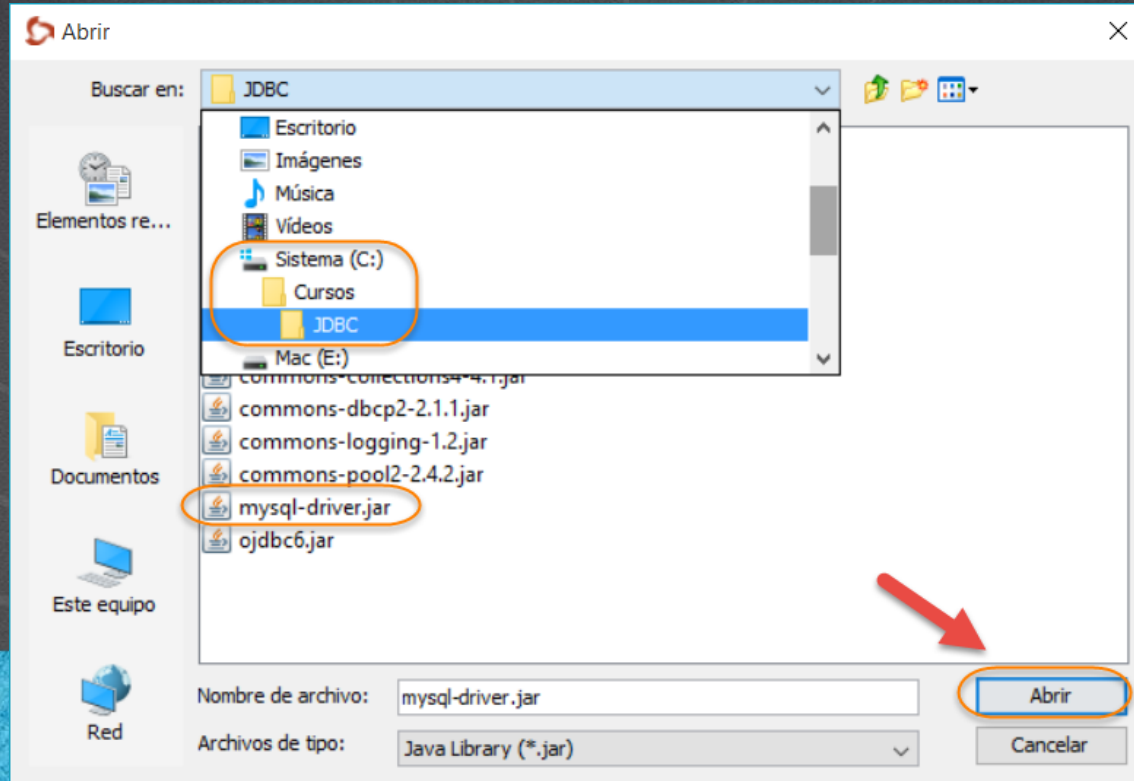
PASO 2. CONFIGURAMOS EL DRIVER DE MYSQL

Agregamos el driver de mysql:



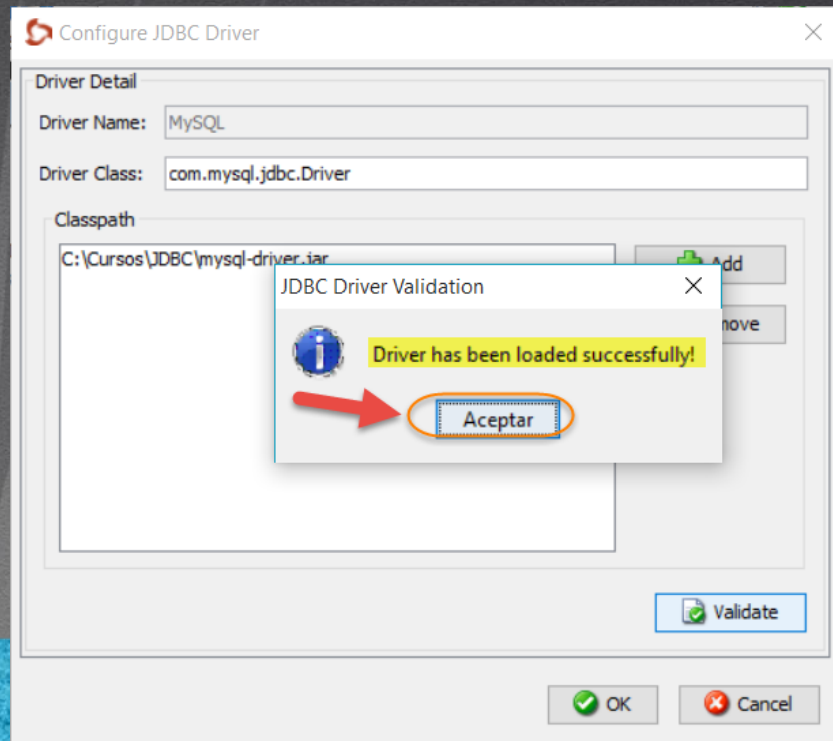
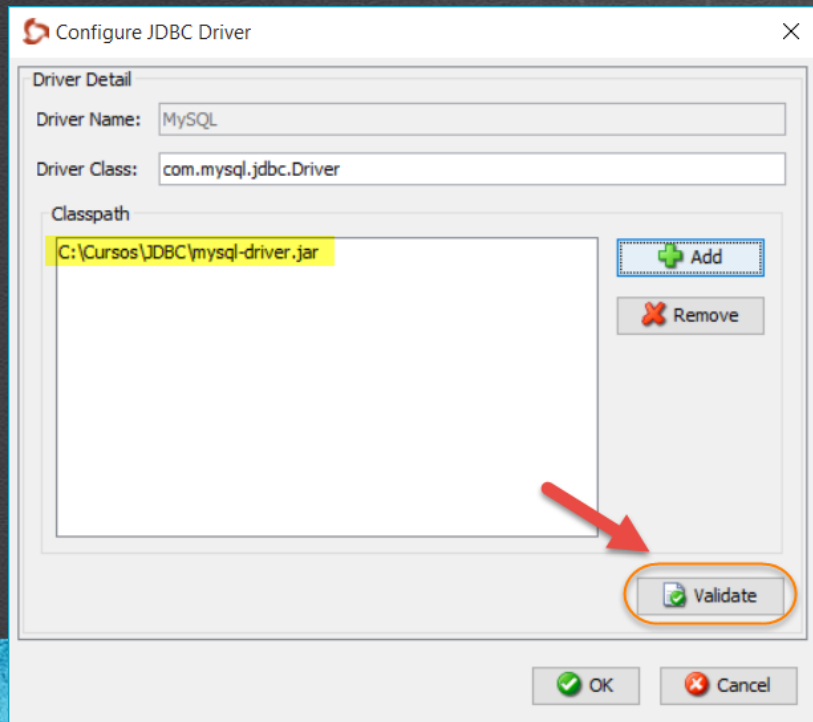
PASO 2. CONFIGURAMOS EL DRIVER DE MYSQL

Seleccionamos la carpeta donde tenemos descargado el driver:



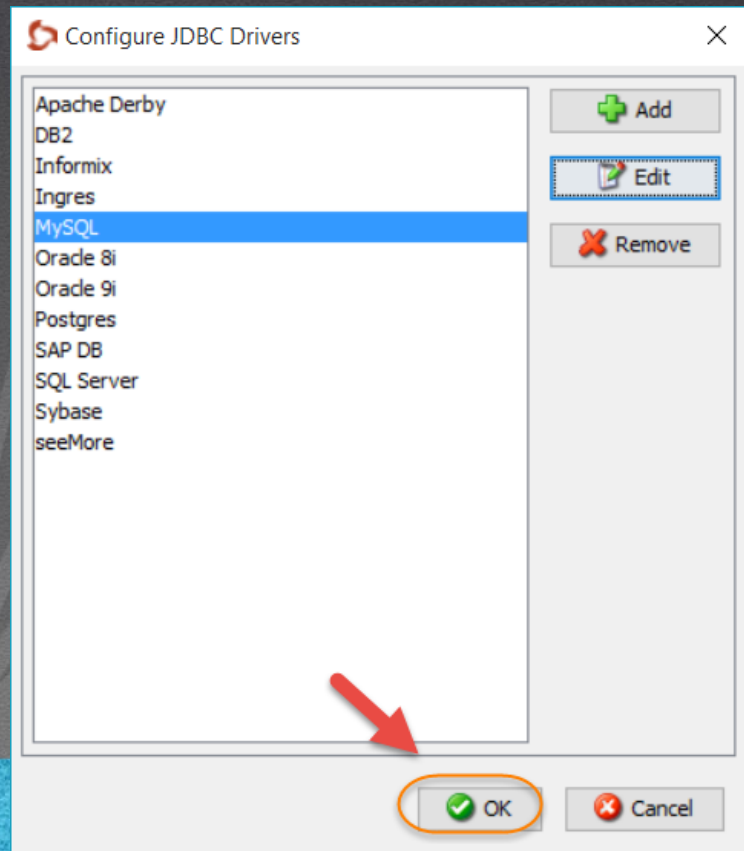
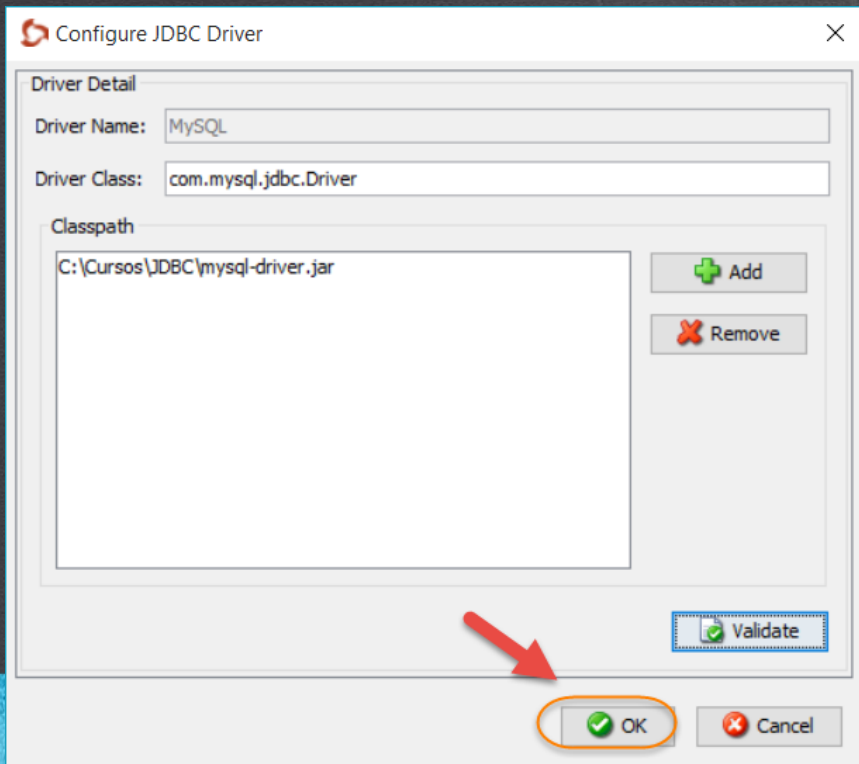
PASO 2. CONFIGURAMOS EL DRIVER DE MYSQL

Validamos el driver agregado:



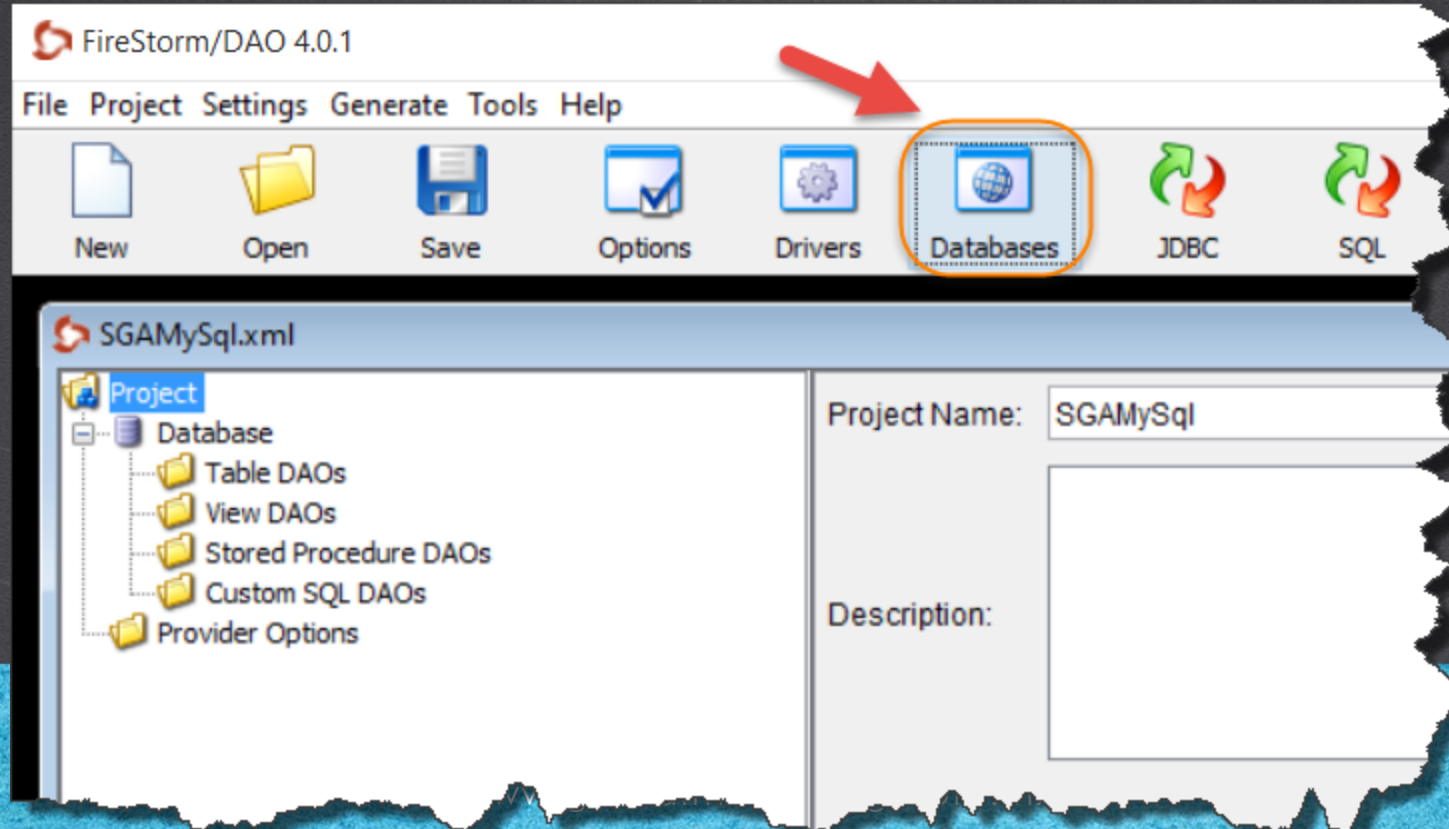
PASO 2. CONFIGURAMOS EL DRIVER DE MYSQL

Guardamos los cambios:



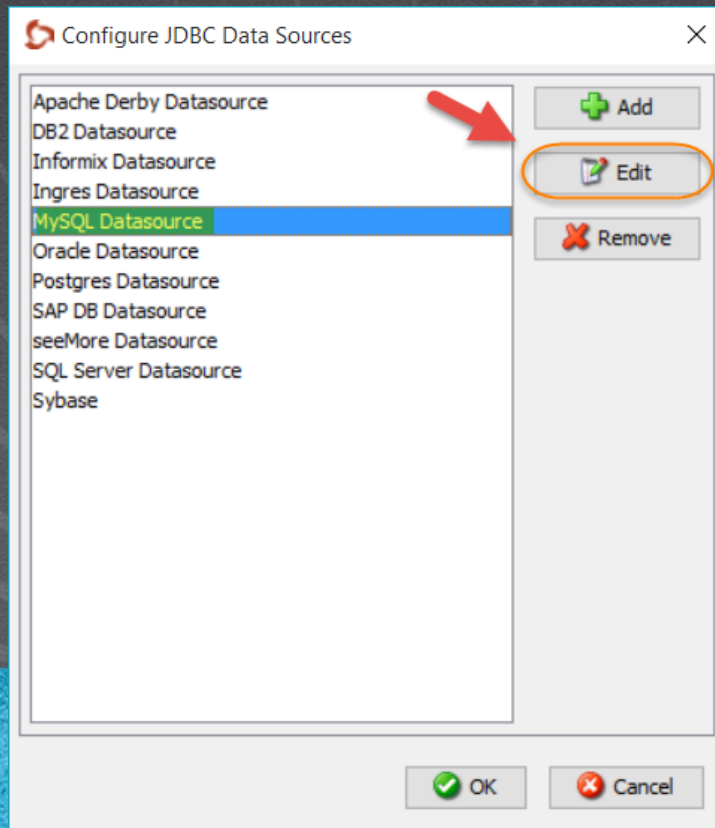
PASO 3. CONFIGURAMOS LA CONEXIÓN A MYSQL

Configuramos la conexión a la base de datos de MySql:



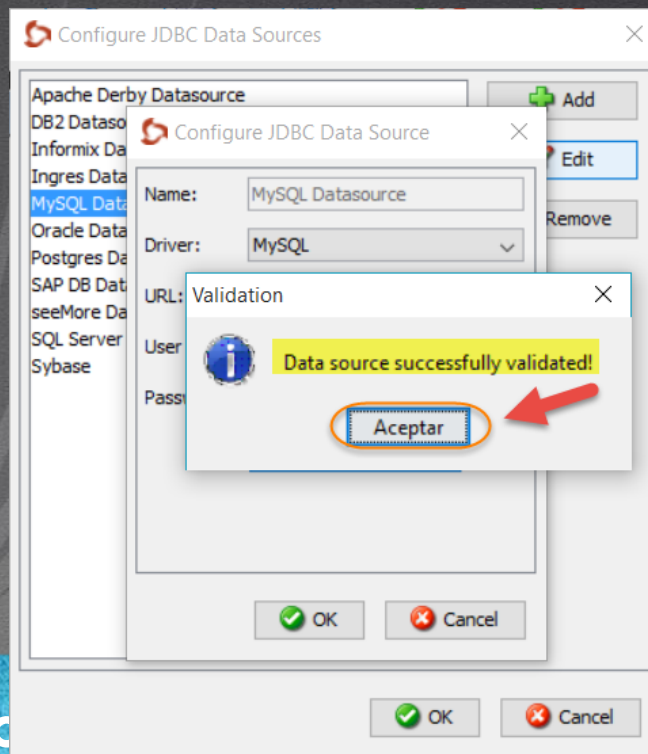
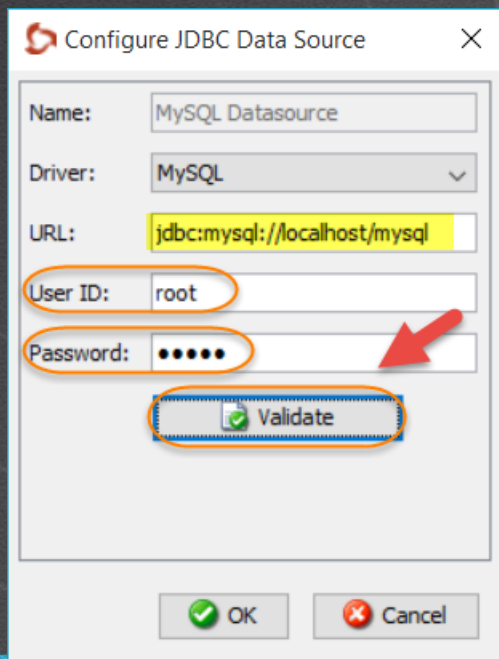
PASO 3. CONFIGURAMOS LA CONEXIÓN A MYSQL

Configuramos la conexión a la base de datos de MySql:



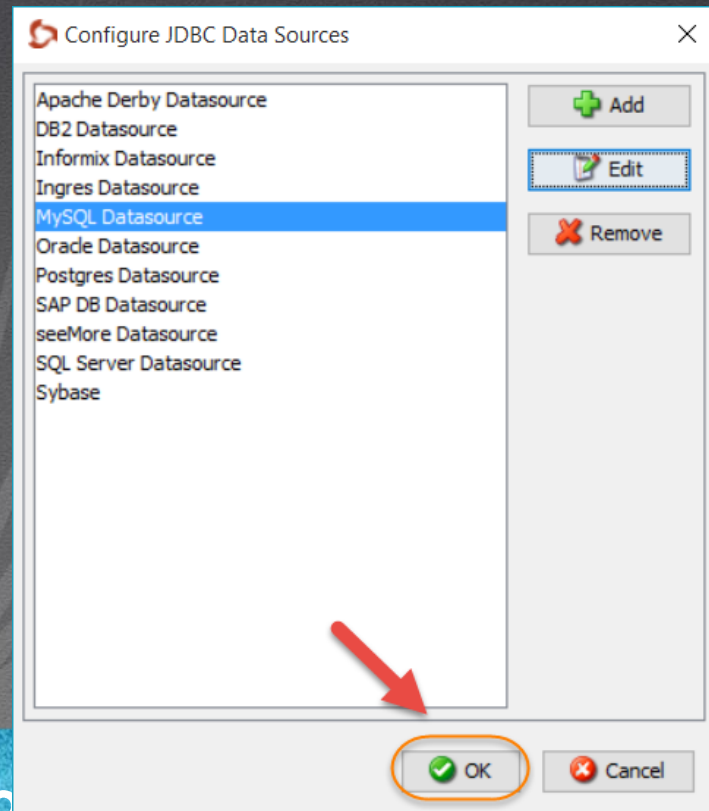
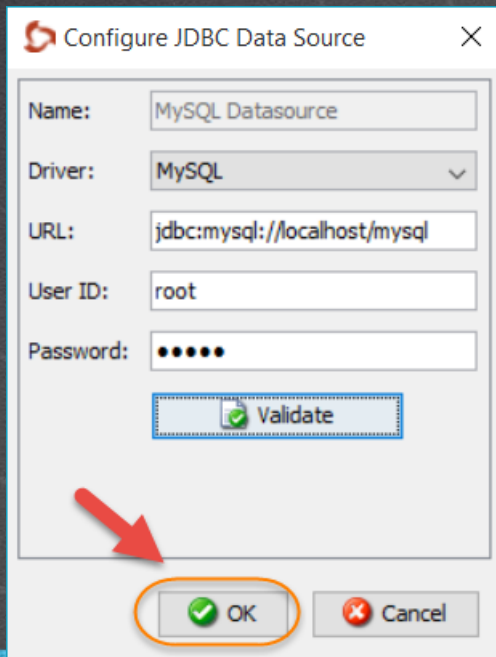
PASO 3. CONFIGURAMOS LA CONEXIÓN A MYSQL

Configuramos la conexión a la base de datos de MySql y la validamos:



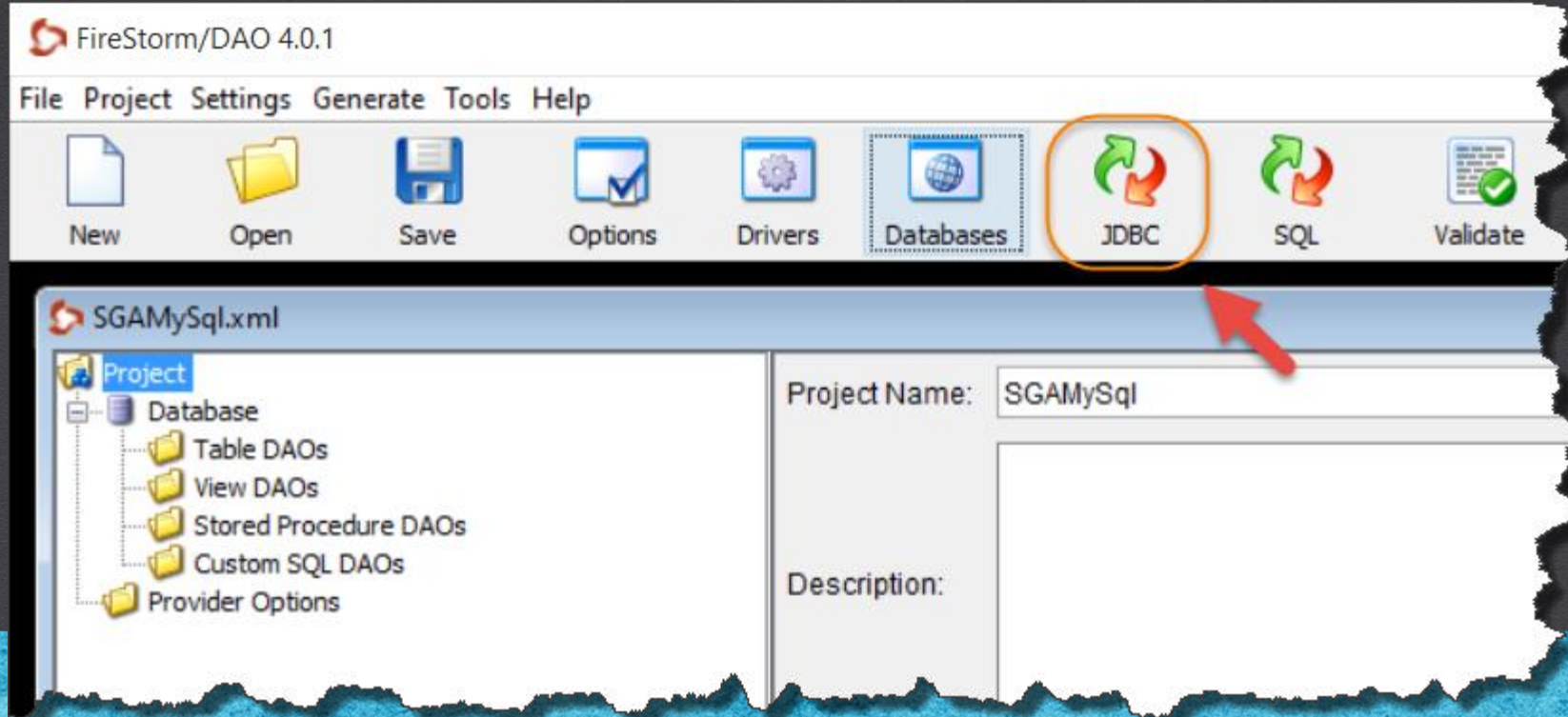
PASO 3. CONFIGURAMOS LA CONEXIÓN A MYSQL

Aceptamos los cambios:



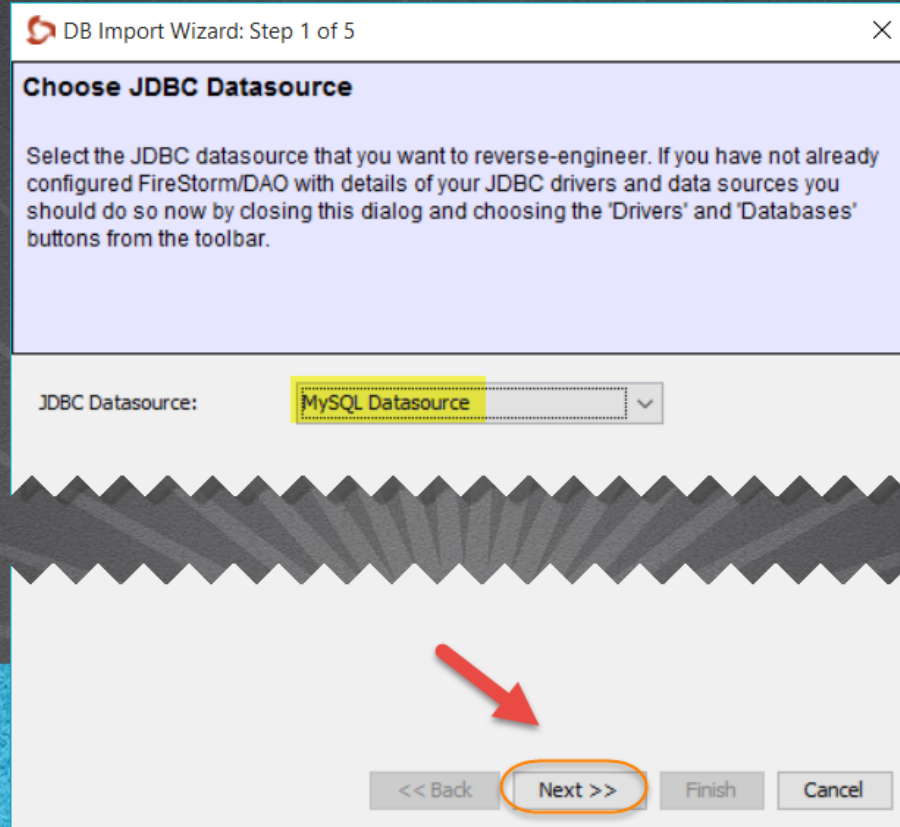
PASO 4. CONFIGURAMOS LA INGENIERÍA INVERSA

Preparamos el proyecto para generar el código Java a partir de las tabla de la base de datos:



PASO 4. CONFIGURAMOS LA INGENIERÍA INVERSA

Seleccionamos el DataSource deseado, en este caso el de MySQL:



PASO 4. CONFIGURAMOS LA INGENIERÍA INVERSA

Seleccionamos el esquema o catalogo de base de datos con el que vamos a trabajar:

The screenshot shows a window titled "DB Import Wizard: Step 2 of 5" with a close button (X) in the top right corner. The main heading is "Choose Catalog and Schema". Below this, a text instruction reads: "Please select the database catalog and schema you want to reverse-engineer". There are two dropdown menus: "Catalog:" with "sga" selected, and "Schema:" which is currently empty. Both dropdowns are highlighted with orange circles. At the bottom of the window, there are four buttons: "<< Back", "Next >>", "Finish", and "Cancel". The "Next >>" button is highlighted with an orange circle, and a red arrow points to it from below.

DB Import Wizard: Step 2 of 5

Choose Catalog and Schema

Please select the database catalog and schema you want to reverse-engineer

Catalog: sga

Schema:

<< Back Next >> Finish Cancel

PASO 4. CONFIGURAMOS LA INGENIERÍA INVERSA

Solo importaremos tablas, así seleccionamos solo las opciones mostradas:

DB Import Wizard: Step 3 of 5

Choose Object Types

Select the database object types that you want to reverse-engineer from your database

Reverse-engineer the following types:

- ☒ Import Tables
- ☐ Import Views
- ☐ Import Stored Procedures

Java Preferences:

- ☒ Map to Java primitive types where possible (e.g. long instead of java.lang.Long)
- ☐ Map large numeric types to BigInteger or BigDecimal

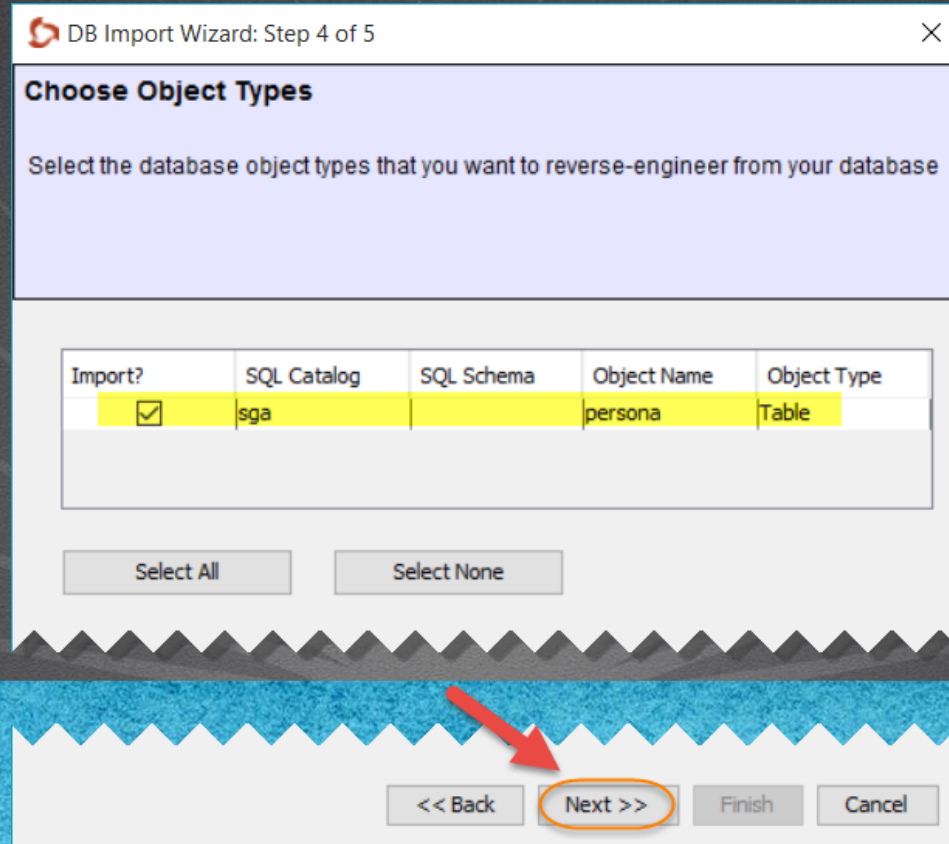
General Options:

- ☐ Write verbose output to log file

<< Back **Next >>** Finish Cancel

PASO 4. CONFIGURAMOS LA INGENIERÍA INVERSA

Seleccionamos las tablas a importar:



DB Import Wizard: Step 4 of 5

Choose Object Types

Select the database object types that you want to reverse-engineer from your database

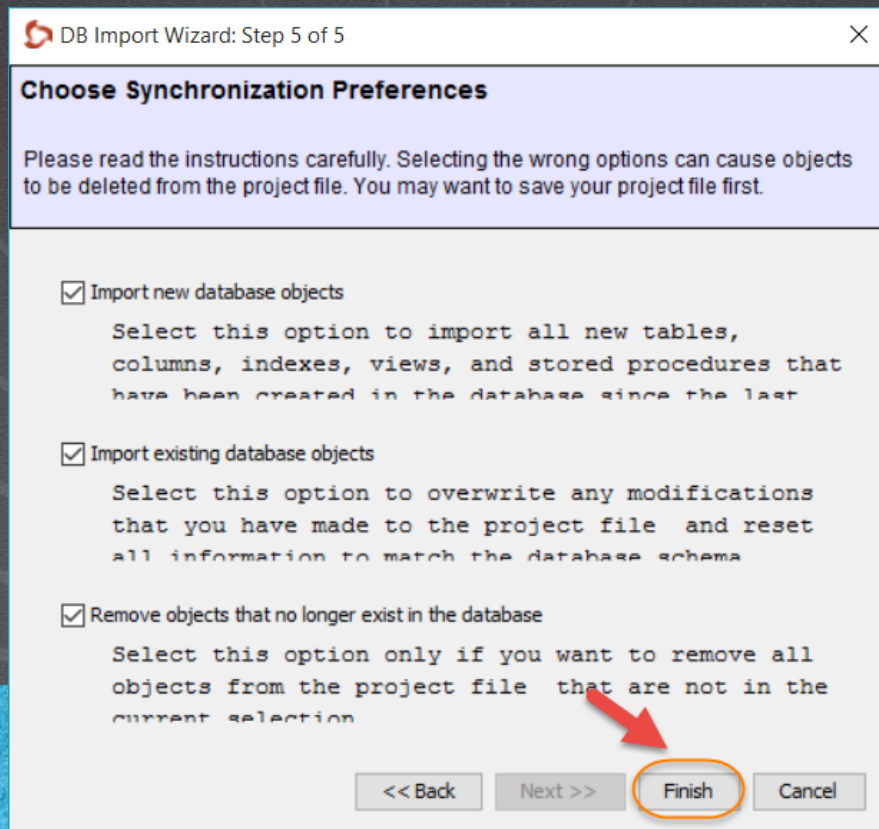
Import?	SQL Catalog	SQL Schema	Object Name	Object Type
<input checked="" type="checkbox"/>	sga		persona	Table

Select All Select None

<< Back Next >> Finish Cancel

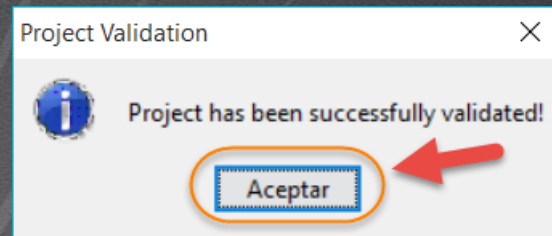
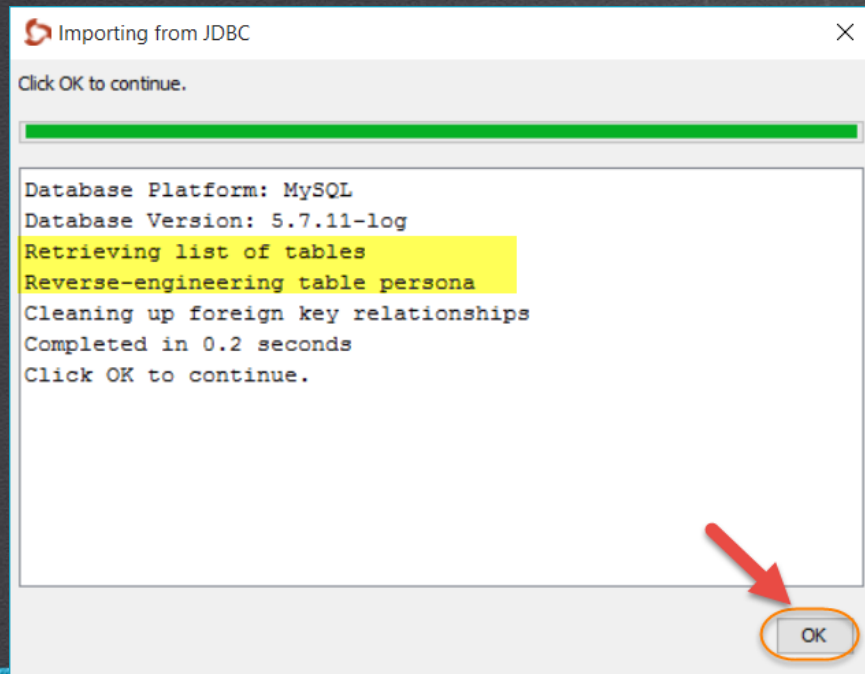
PASO 4. CONFIGURAMOS LA INGENIERÍA INVERSA

Seleccionamos las tablas a importar:



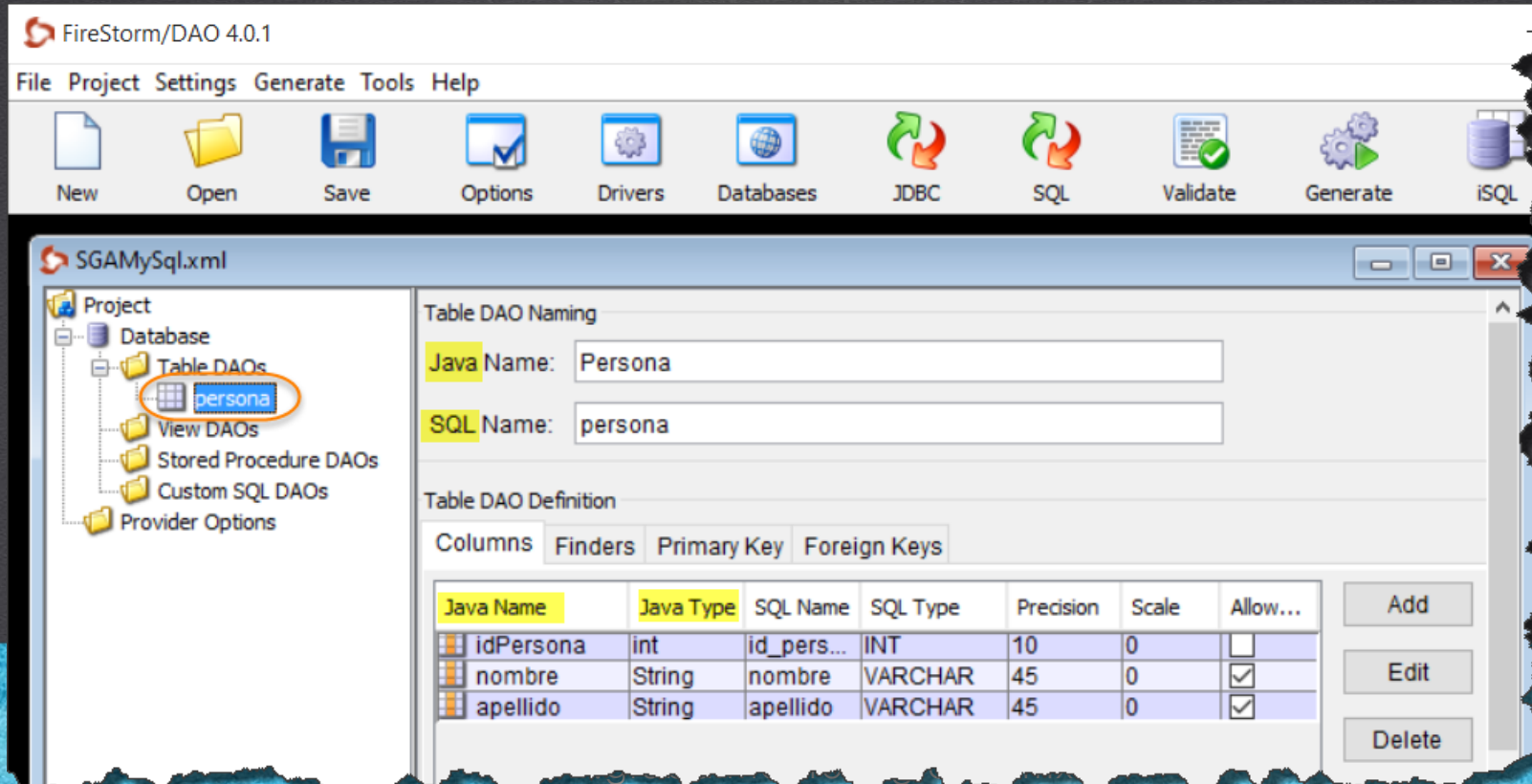
PASO 4. CONFIGURAMOS LA INGENIERÍA INVERSA

Seleccionamos las tablas a importar:



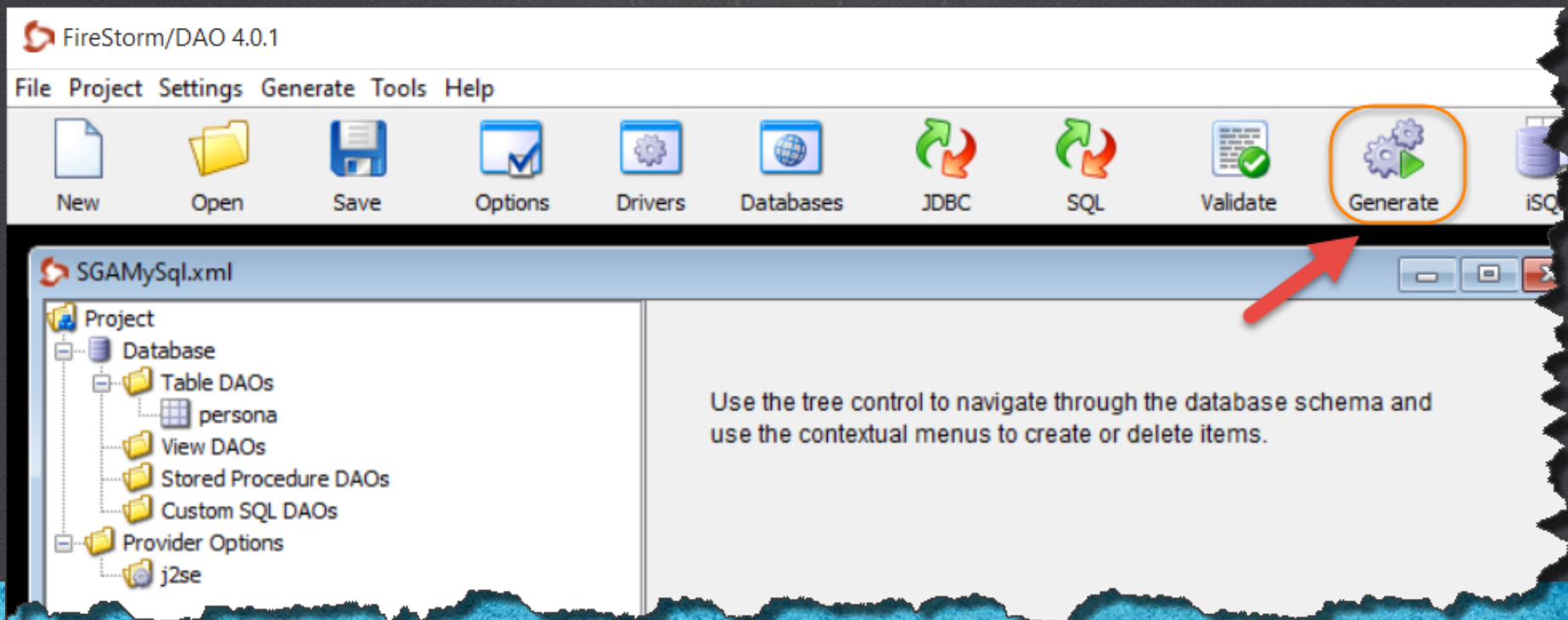
PASO 4. CONFIGURAMOS LA INGENIERÍA INVERSA

Aquí podemos configurar el código Java que se va a generar a partir de la tabla a importar:



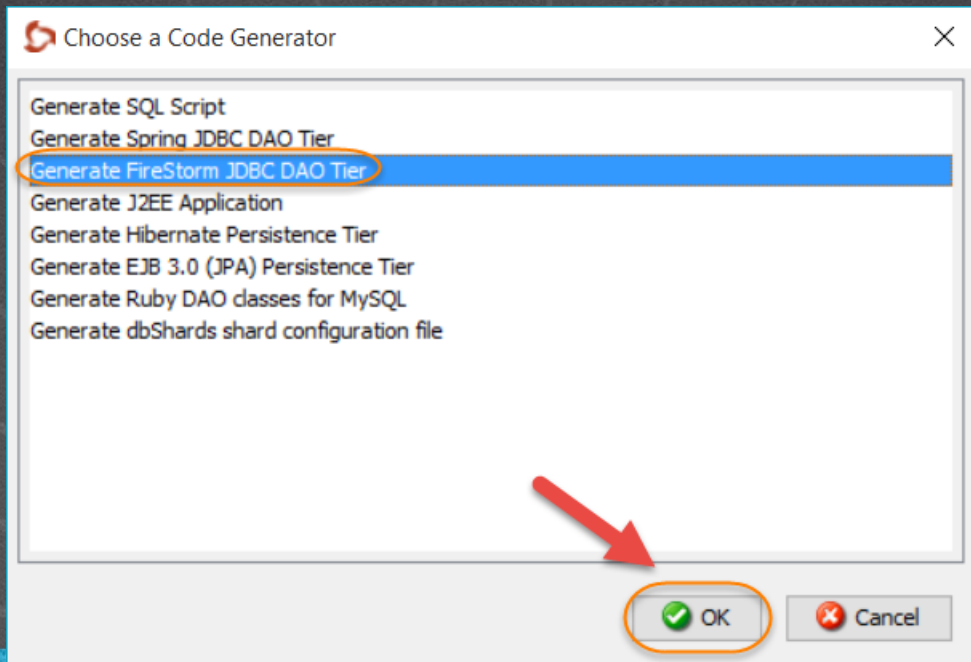
PASO 5. HACEMOS LA INGENIERÍA INVERSA

Generamos el código Java a partir de la configuración realizada:



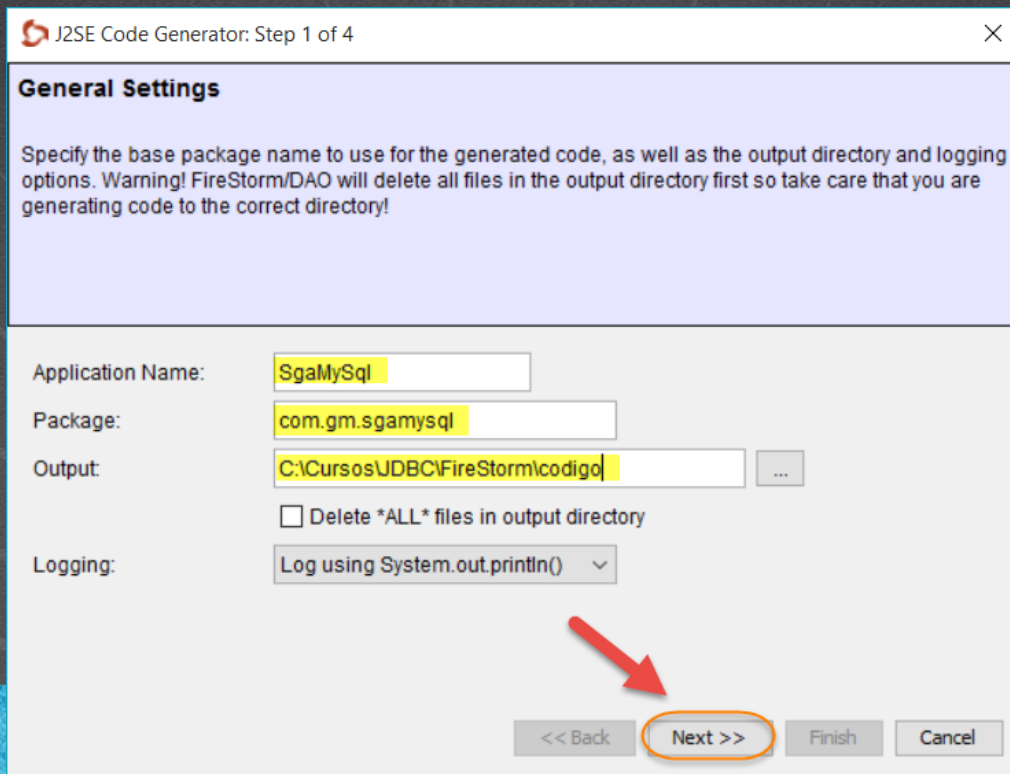
PASO 5. HACEMOS LA INGENIERÍA INVERSA

Generamos el código Java a partir de la configuración realizada:



PASO 5. HACEMOS LA INGENIERÍA INVERSA

Generamos el código Java a partir de la configuración realizada:



J2SE Code Generator: Step 1 of 4

General Settings

Specify the base package name to use for the generated code, as well as the output directory and logging options. Warning! FireStorm/DAO will delete all files in the output directory first so take care that you are generating code to the correct directory!

Application Name:

Package:

Output: ...

☐ Delete *ALL* files in output directory

Logging: ▾

<< Back **Next >>** Finish Cancel

PASO 5. HACEMOS LA INGENIERÍA INVERSA

Generamos el código Java a partir de la configuración realizada:



The screenshot shows a dialog box titled "J2SE Code Generator: Step 2 of 4". The main heading is "Choose Persistence Strategy". Below this, a text block explains that FireStorm/DAO generates a Data Access Object (DAO) interface and supports various persistence technologies. It asks the user to select a technology, recommending 'JDBC' if unsure. Under the heading "Choose a persistence framework", there are three radio button options: "JDBC" (which is selected and circled in orange), "Enterprise JavaBeans (EJB)", and "Hibernate". At the bottom, there are four buttons: "<< Back", "Next >>" (circled in orange with a red arrow pointing to it), "Finish", and "Cancel".

Choose Persistence Strategy

FireStorm/DAO generates a Data Access Object (DAO) interface but supports a number of different persistence technologies that can be used to implement that DAO interface. Please select the technology that you would like to use. If you are not sure then we recommend that you choose 'JDBC'

Choose a persistence framework

☒ JDBC

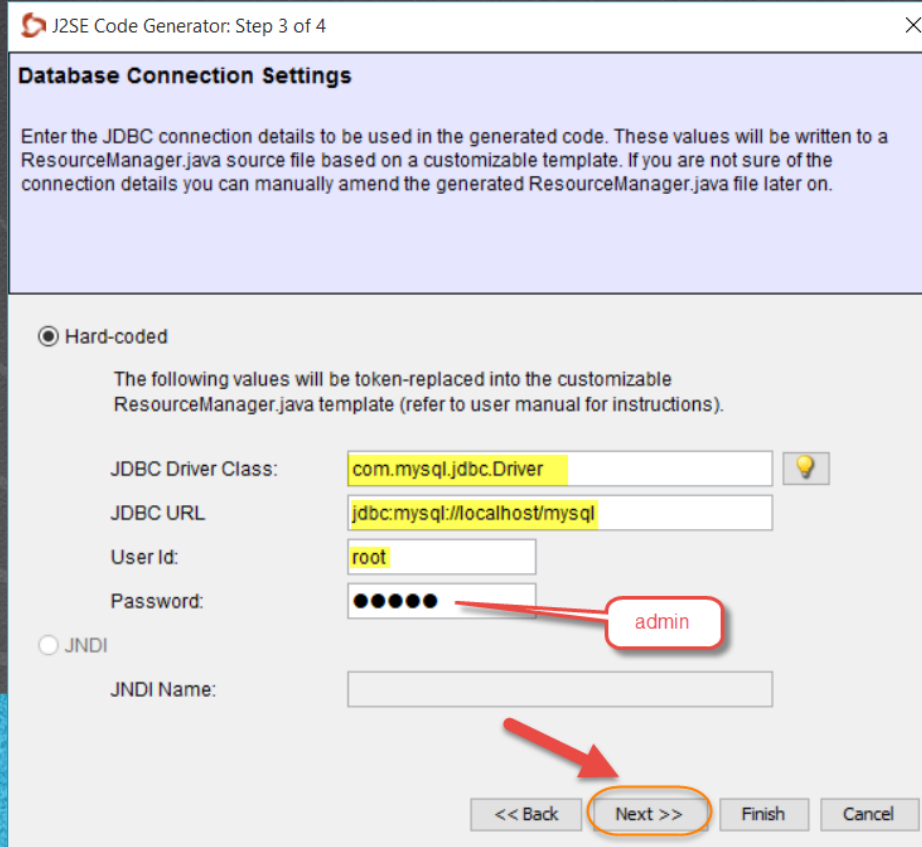
☐ Enterprise JavaBeans (EJB)

☐ Hibernate

<< Back **Next >>** Finish Cancel

PASO 5. HACEMOS LA INGENIERÍA INVERSA

Generamos el código Java a partir de la configuración realizada:



J2SE Code Generator: Step 3 of 4

Database Connection Settings

Enter the JDBC connection details to be used in the generated code. These values will be written to a ResourceManager.java source file based on a customizable template. If you are not sure of the connection details you can manually amend the generated ResourceManager.java file later on.

☒ **Hard-coded**

The following values will be token-replaced into the customizable ResourceManager.java template (refer to user manual for instructions).

JDBC Driver Class:

JDBC URL:

User Id:

Password:

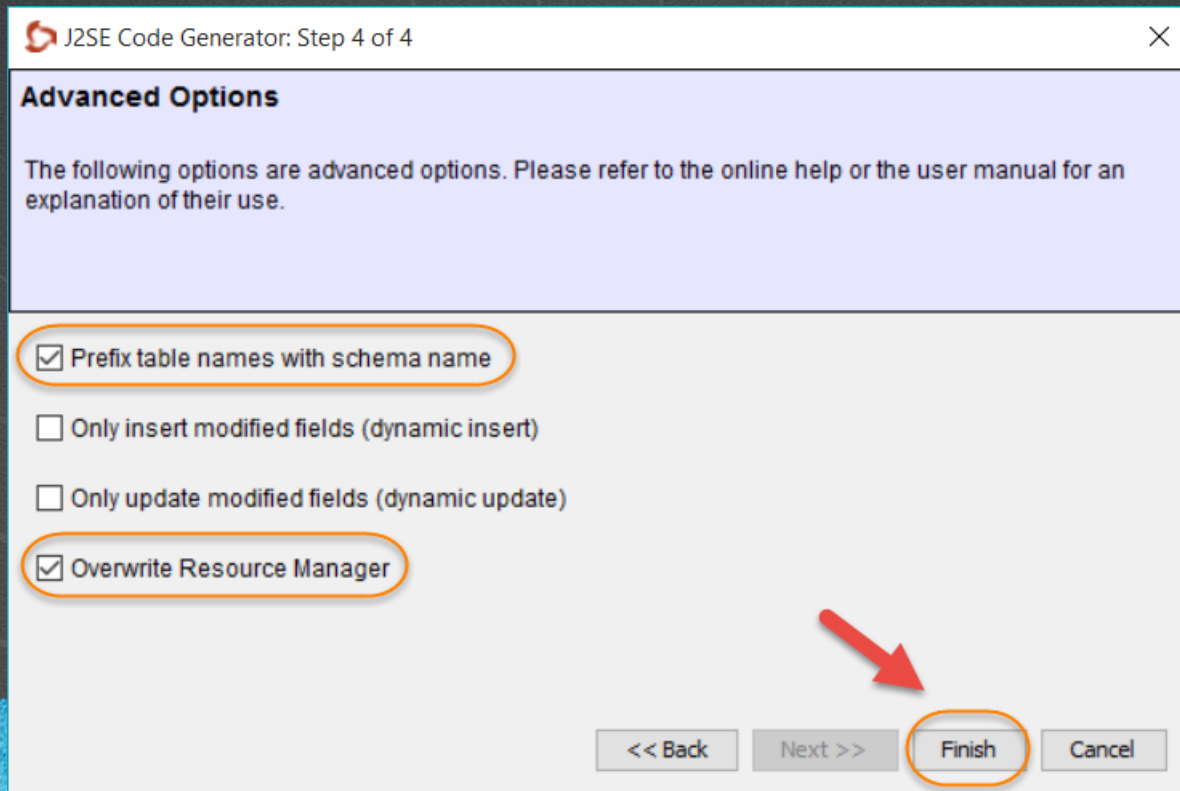
☐ **JNDI**

JNDI Name:

The screenshot shows a dialog box titled "J2SE Code Generator: Step 3 of 4". It has a close button (X) in the top right. The main section is titled "Database Connection Settings" and contains instructions: "Enter the JDBC connection details to be used in the generated code. These values will be written to a ResourceManager.java source file based on a customizable template. If you are not sure of the connection details you can manually amend the generated ResourceManager.java file later on." Below this, there are two radio buttons: "Hard-coded" (selected) and "JNDI". Under "Hard-coded", it says "The following values will be token-replaced into the customizable ResourceManager.java template (refer to user manual for instructions)." There are four input fields: "JDBC Driver Class" with value "com.mysql.jdbc.Driver", "JDBC URL" with value "jdbc:mysql://localhost/mysql", "User Id" with value "root", and "Password" with masked characters "•••••". A red arrow points from the "admin" text in a red box to the password field. Below the "JNDI" section is an empty "JNDI Name" field. At the bottom, there are four buttons: "<< Back", "Next >>" (highlighted with a red circle and a red arrow pointing to it), "Finish", and "Cancel".

PASO 5. HACEMOS LA INGENIERÍA INVERSA

Generamos el código Java a partir de la configuración realizada:



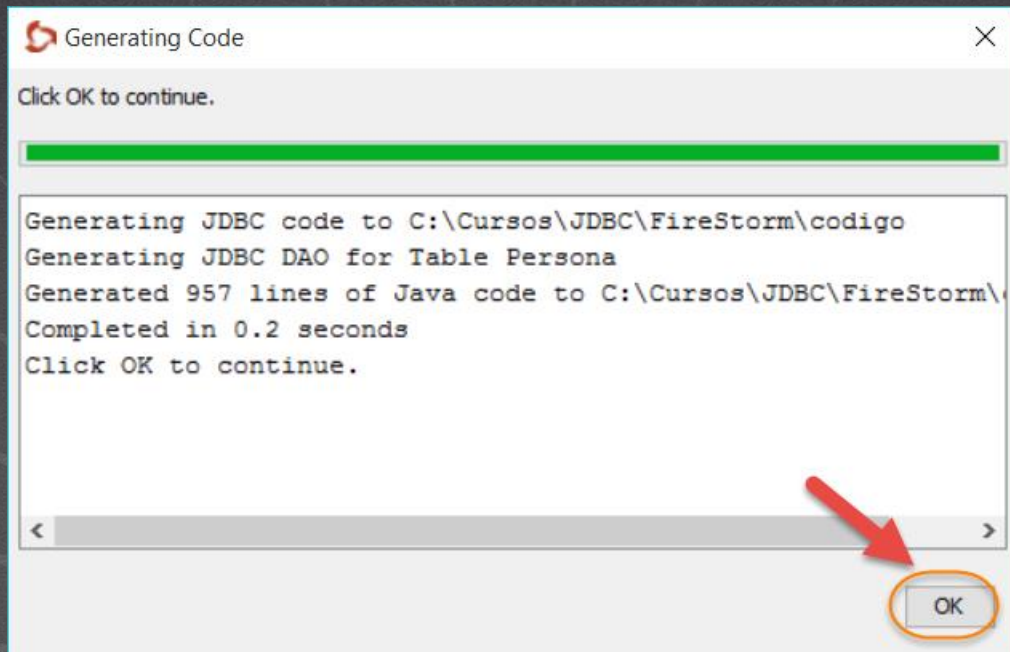
The screenshot shows a dialog box titled "J2SE Code Generator: Step 4 of 4". It has a close button (X) in the top right corner. The main content area is titled "Advanced Options" and contains the text: "The following options are advanced options. Please refer to the online help or the user manual for an explanation of their use." Below this text are four checkboxes, each with a label:

- ☒ Prefix table names with schema name
- ☐ Only insert modified fields (dynamic insert)
- ☐ Only update modified fields (dynamic update)
- ☒ Overwrite Resource Manager

At the bottom of the dialog box are four buttons: "<< Back", "Next >>", "Finish", and "Cancel". A red arrow points to the "Finish" button, which is also circled in orange. The "Prefix table names with schema name" and "Overwrite Resource Manager" checkboxes are also circled in orange.

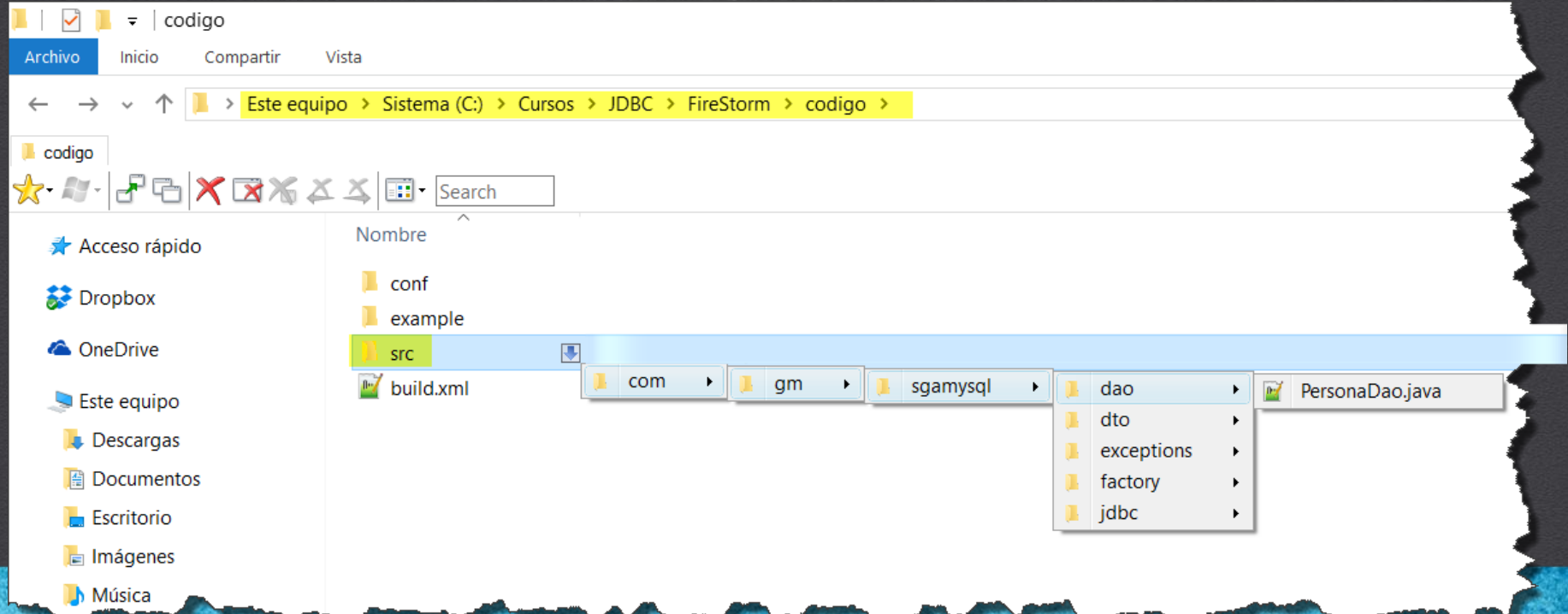
PASO 5. HACEMOS LA INGENIERÍA INVERSA

Generamos el código Java a partir de la configuración realizada:



PASO 6. REVISION DE LA INGENIERÍA INVERSA

Revisamos el código generado en la carpeta seleccionada. Las carpetas importantes son src:

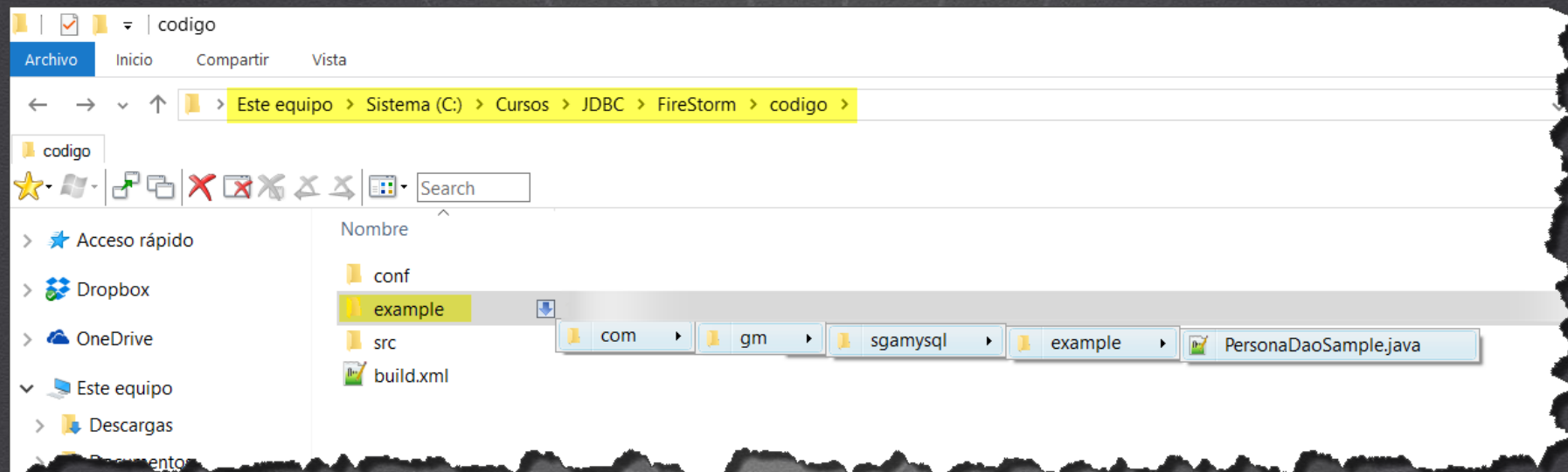


CURSO DE JAVA CON JDBC

www.globalmentoring.com.mx

PASO 6. REVISION DE LA INGENIERÍA INVERSA

La otra carpeta importante es la de example:



CONCLUSIÓN DEL EJERCICIO

- Con este ejercicio hemos puesto en práctica el proceso de ingeniería inversa.
- Este proceso nos ayuda a automatizar el proceso de generación de la capa de datos, y así ocuparnos más por las reglas de negocio de nuestra aplicación que por el acceso a datos, además de que todo el código generado es personalizable según nuestras necesidades.



CURSO DE JAVA CON JDBC

www.globalmentoring.com.mx

CURSO ONLINE

JAVA CON JDBC

Por: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida

CURSO DE JAVA CON JDBC

www.globalmentoring.com.mx