

ciclo: [DAM]

MÓDULO DE [PROGRAMACIÓN]

[Tarea Nº 09]

Alumno:

[Juan Carlos Filter Martín]

[15456141A]

**Contenido**

[1. Documentos que se adjuntan a este informe. 3](#__RefHeading___Toc572_1477578061)

[2. RA09\_b) Se han programado conexiones con bases de datos. 3](#__RefHeading___Toc574_1477578061)

[D) Crear una BD MySQL con el nombre equivalente al DNI 3](#__RefHeading___Toc576_1477578061)

[E) Contendrá una tabla llamada Departamentos con los siguientes campos: 4](#__RefHeading___Toc578_1477578061)

[Código tipo numérico entero (primary key) 5](#__RefHeading___Toc580_1477578061)

[Nombre tipo String de 50 caracteres 5](#__RefHeading___Toc582_1477578061)

[Id de localización tipo numérico entero 5](#__RefHeading___Toc584_1477578061)

[Id del mánager tipo numérico entero 5](#__RefHeading___Toc586_1477578061)

[Tabla Departamentos creada (Resultado) 5](#__RefHeading___Toc580_1477578061_Copia_1)

[F) Crear usuario para acceder a la BD: 6](#__RefHeading___Toc578_1477578061_Copia_1)

[G) Creamos el proyecto Java agregándole las librerías necesarias para trabajar con BD 7](#__RefHeading___Toc925_1477578061)

[3. RA09\_d) Se han creado programas para recuperar y mostrar información almacenada en bases de datos. 8](#__RefHeading___Toc588_1477578061)

[A) Crear la interfaz de usuario 8](#__RefHeading___Toc927_1477578061)

[Insertar 9](#__RefHeading___Toc929_1477578061)

[Clase GestorBD 9](#__RefHeading___Toc931_1477578061)

[JFrame form (Interfaz de usuario) 11](#__RefHeading___Toc931_1477578061_Copia_1)

[Resultado Interfaz Gráfica 12](#__RefHeading___Toc931_1477578061_Copia_2)

[Resultado (phpMyAdmin) 14](#__RefHeading___Toc931_1477578061_Copia_3)

[4. RA09\_e) Se han efectuado borrados y modificaciones sobre la información almacenada. 14](#__RefHeading___Toc590_1477578061)

[Borrar 14](#__RefHeading___Toc929_1477578061_Copia_1)

[Clase GestorBD 15](#__RefHeading___Toc931_1477578061_Copia_4)

[JFrame Form (Interfaz de usuario) 19](#__RefHeading___Toc931_1477578061_Copia_5)

[Resultado (Interfaz Gráfica) 20](#__RefHeading___Toc931_1477578061_Copia_d)

[Resultado (phpMyAdmin) 21](#__RefHeading___Toc931_1477578061_Copia_6)

[Modificar 22](#__RefHeading___Toc929_1477578061_Copia_2)

[Clase GestorBD 22](#__RefHeading___Toc931_1477578061_Copia_7)

[JFrame form (Interfaz de usuario) 24](#__RefHeading___Toc931_1477578061_Copia_8)

[Resultado (Interfaz Gráfica) 25](#__RefHeading___Toc931_1477578061_Copia_9)

[Resultado (phpMyAdmin) 26](#__RefHeading___Toc931_1477578061_Copia_e)

[5. RA09\_e) Se han efectuado borrados y modificaciones sobre la información almacenada. 27](#__RefHeading___Toc592_1477578061)

[Mostrar 27](#__RefHeading___Toc929_1477578061_Copia_3)

[Clase GestorBD 27](#__RefHeading___Toc931_1477578061_Copia_a)

[JFrame form (Interfaz de usuario) 29](#__RefHeading___Toc931_1477578061_Copia_b)

[Resultado (Interfaz Gráfica) 30](#__RefHeading___Toc931_1477578061_Copia_c)

[Limpiar Campos y salir 31](#__RefHeading___Toc929_1477578061_Copia_4)

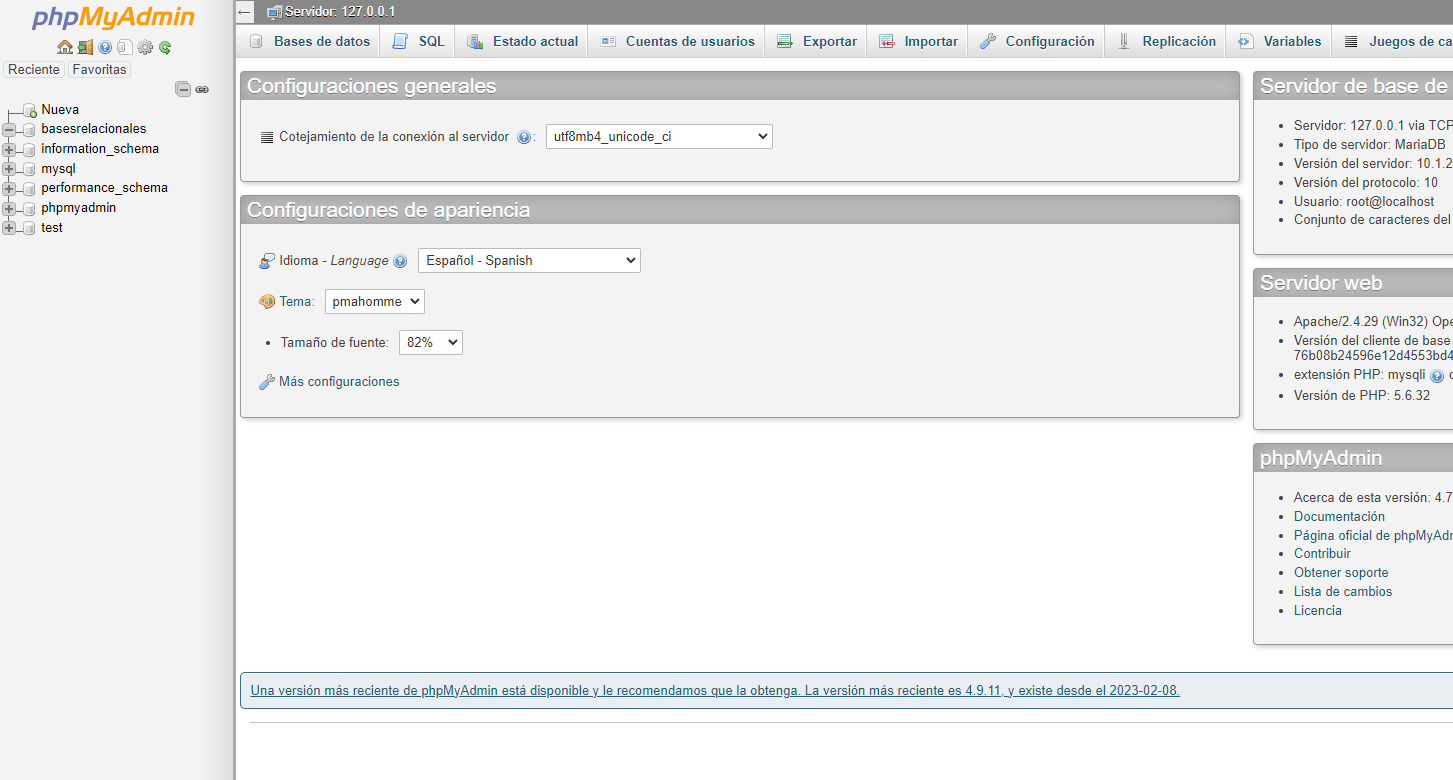
# Documentos que se adjuntan a este informe.

A continuación se detallan los documentos que componen la presente entrega de la tarea:

1. Informe de elaboración de la tarea.
2. Proyecto Java
3. Script.sql exportado de creación de la BD

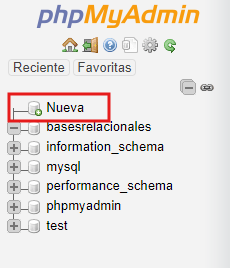
# RA09\_b) Se han programado conexiones con bases de datos.

Desde XAMPP iniciamos el servicio Apache y MySQL. Entramos a phpMyAdmin.

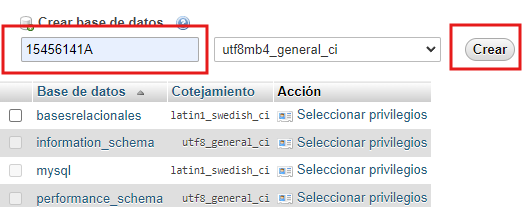


## Crear una BD MySQL con el nombre equivalente al DNI

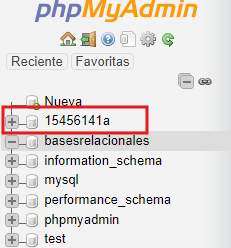
Desde el panel lateral izquierdo creamos una BD nueva.



Con el nombre del DNI: 15456141A y pulsamos en crear

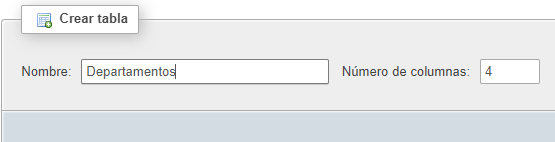


Y tendríamos la BD creada

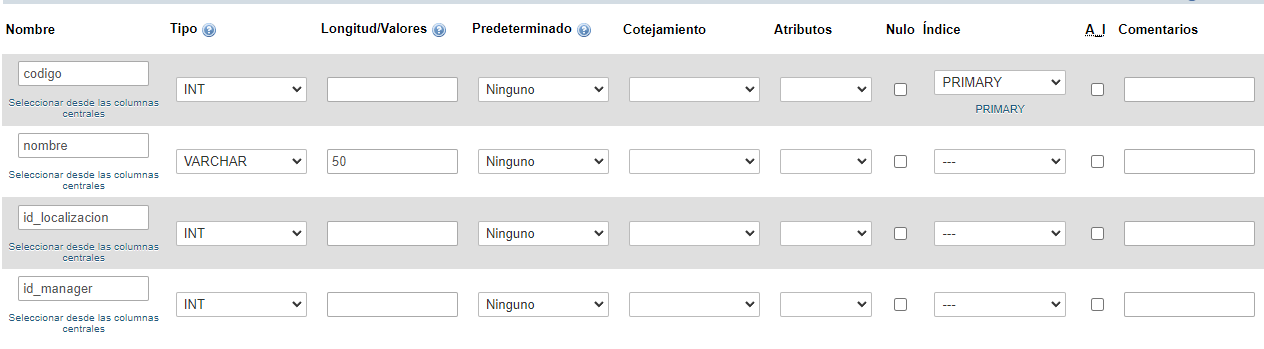


## Contendrá una tabla llamada Departamentos con los siguientes campos:

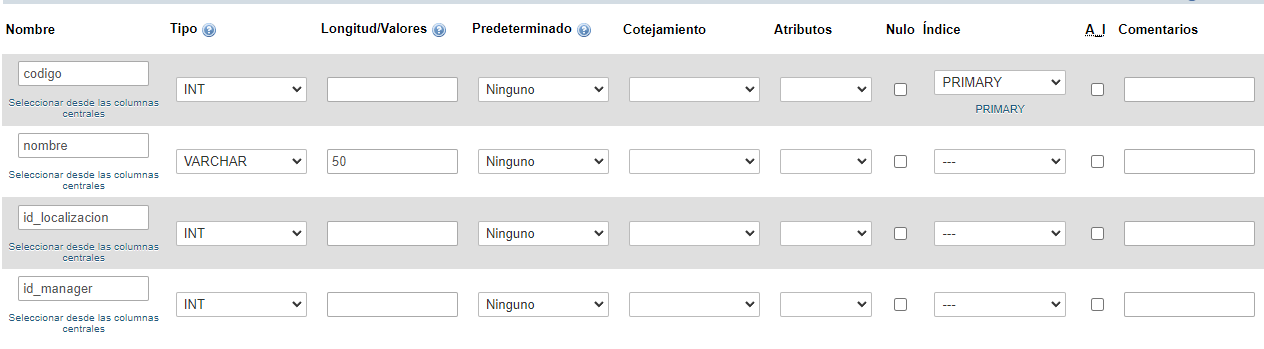
Creamos la tabla Departamentos con 4 columnas ya va a contener 4 campos



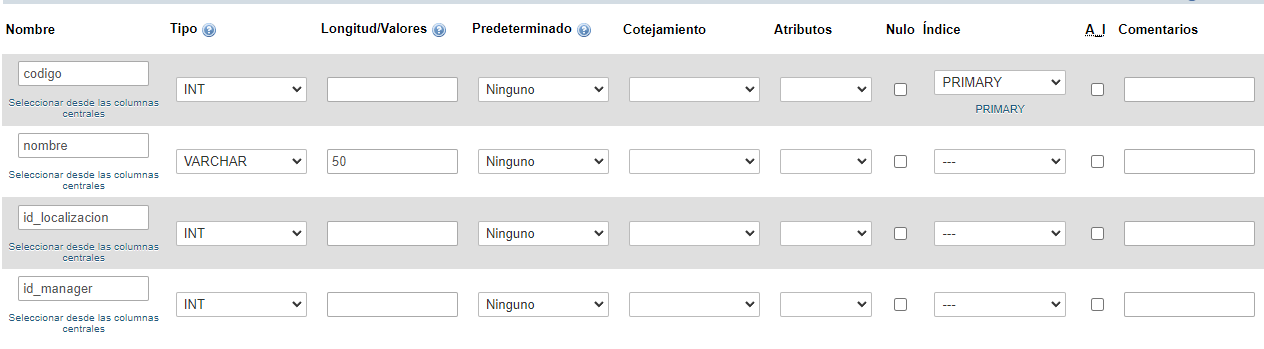
### Código tipo numérico entero (primary key)



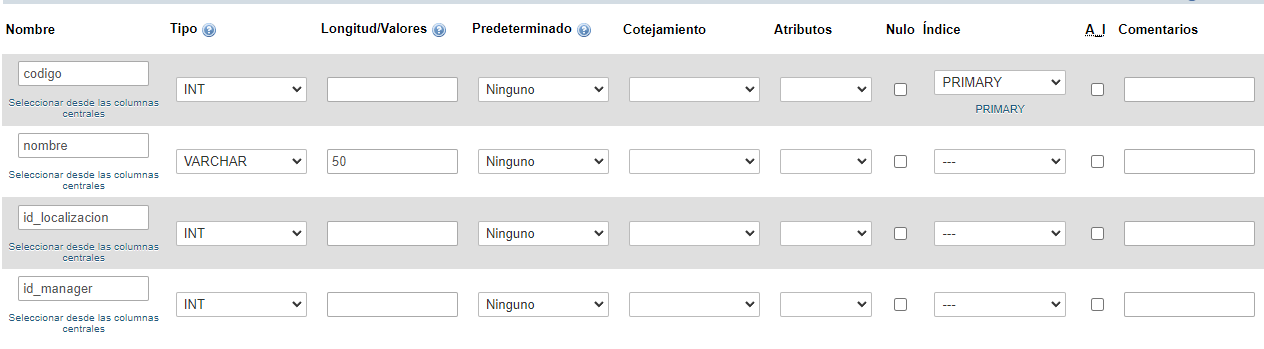
### Nombre tipo String de 50 caracteres



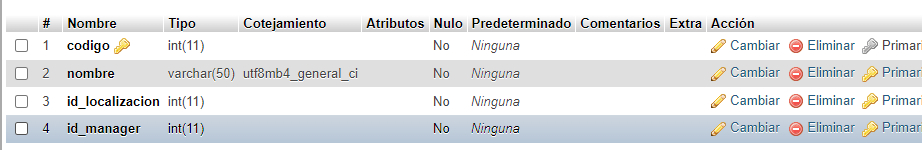
### Id de localización tipo numérico entero



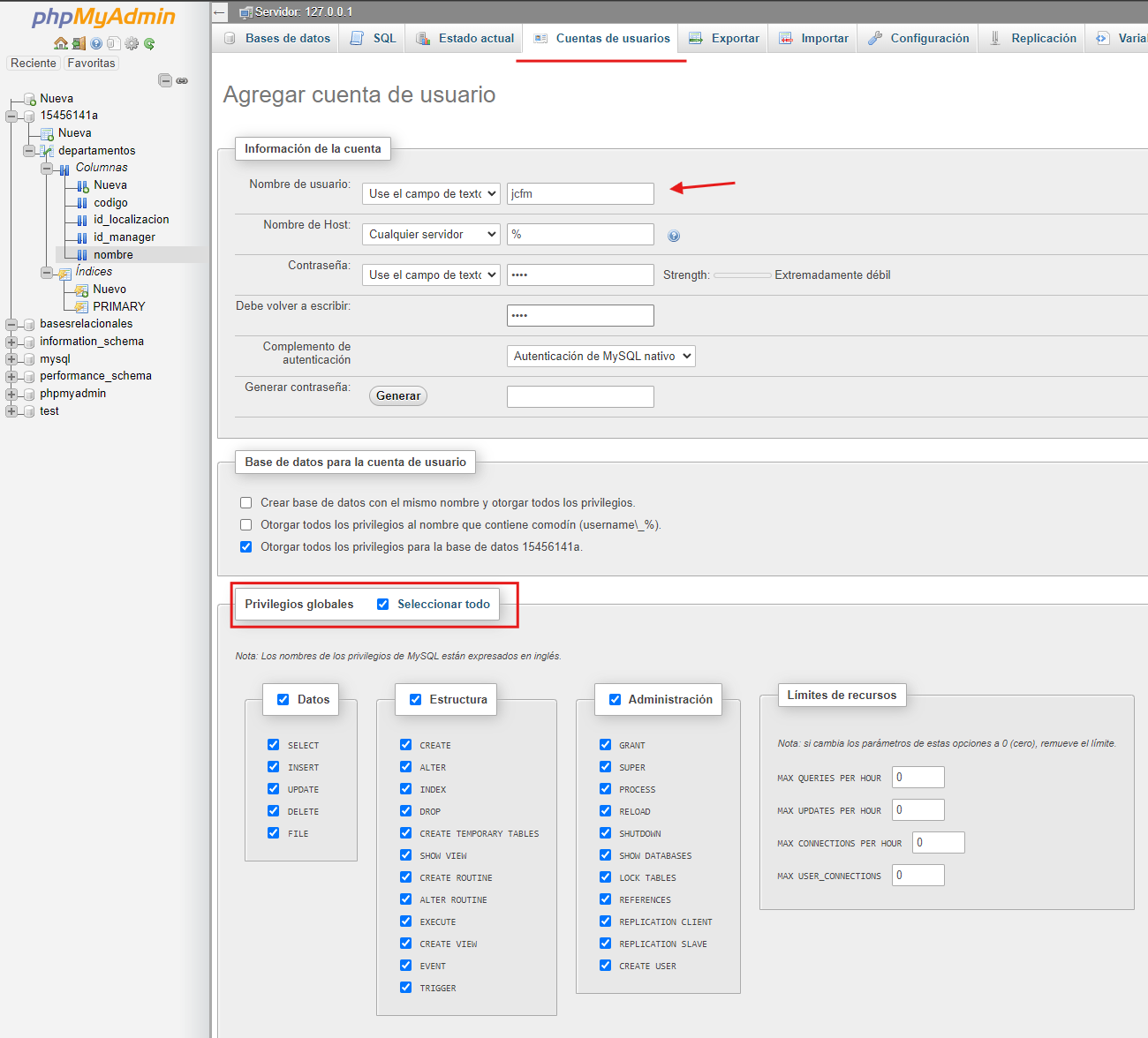
### Id del mánager tipo numérico entero

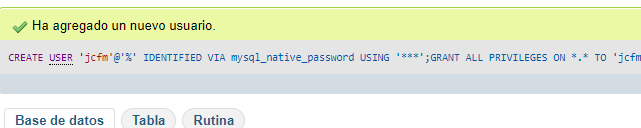


### Tabla Departamentos creada (Resultado)



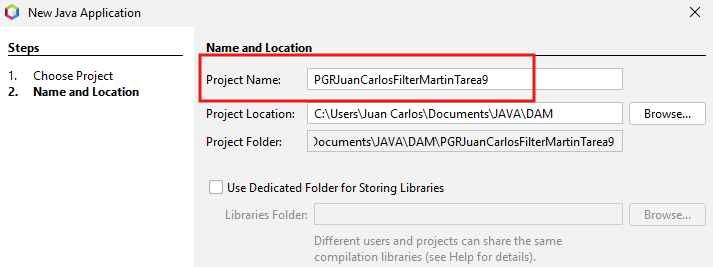
## Crear usuario para acceder a la BD:

Crearemos un usuario para poder acceder a la BD y asignamos todos los privilegios para poder realizar las diferentes consultas, insert, etc.

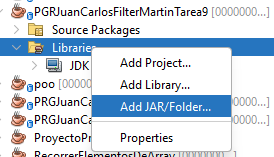


## Creamos el proyecto Java agregándole las librerías necesarias para trabajar con BD

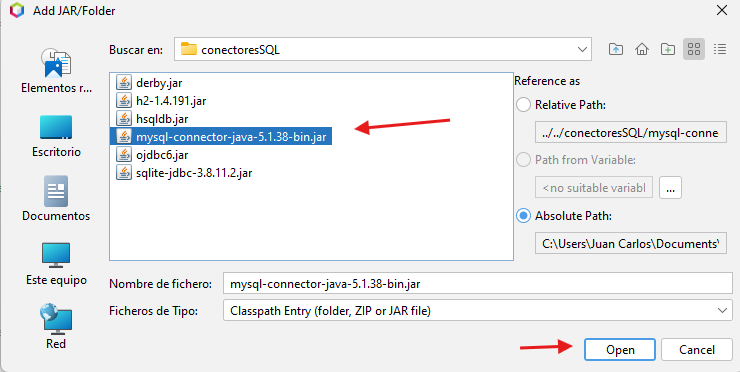
1. Creamos el proyecto



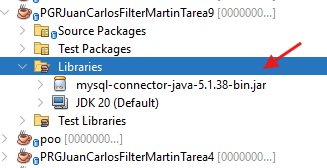
1. Agregamos librerías pulsando botón derecho sobre libraries > **Add JAR/Folder**



1. Vamos al directorio donde se encuentre la librería **“conectoresSQL.zip”** y pulsamos en **Open**



1. Y tendríamos las librerías agregadas al proyecto.



# RA09\_d) Se han creado programas para recuperar y mostrar información almacenada en bases de datos.

## Crear la interfaz de usuario

Tendremos que crear una clase JFrame Form y empezar a diseñar la interfaz gráfica.



(Con la interfaz construida vamos a crear los deferentes eventos asociados a cada botón)

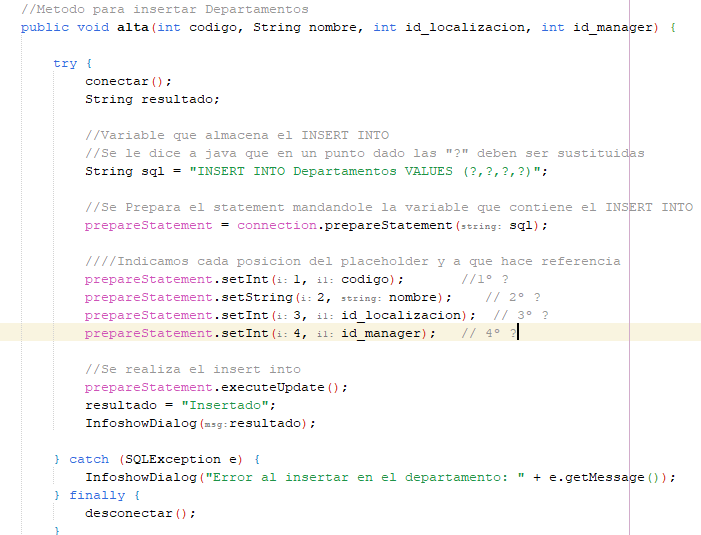
## ****Insertar****

Se va a recoge la información introducida por el usuario en (código, nombre, id\_localizacion, id\_manager).

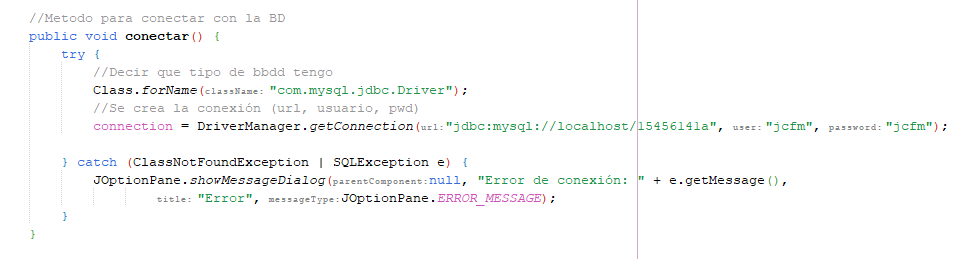
Con los datos recogidos se va a acceder a la BD e insertar el nuevo departamento en la tabla departamentos.

Una vez insertado el departamento se mostrarán todos los registros en la Tabla.

### Clase GestorBD

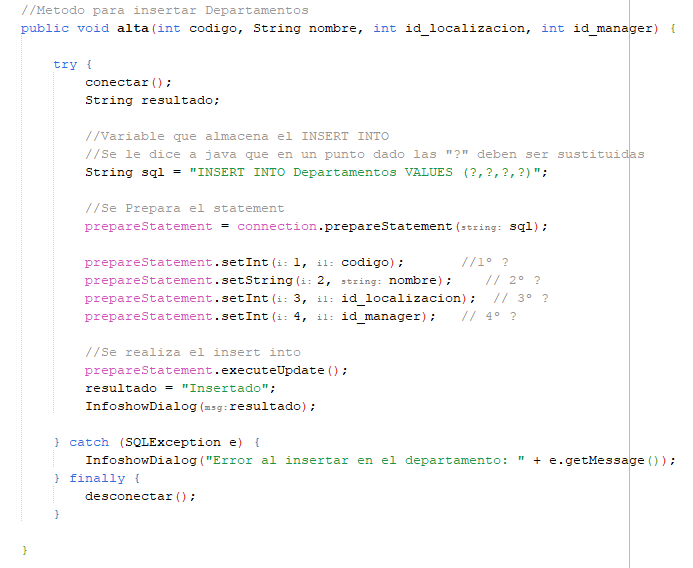


* **Se conectará a la BD** con el método **conectar();**

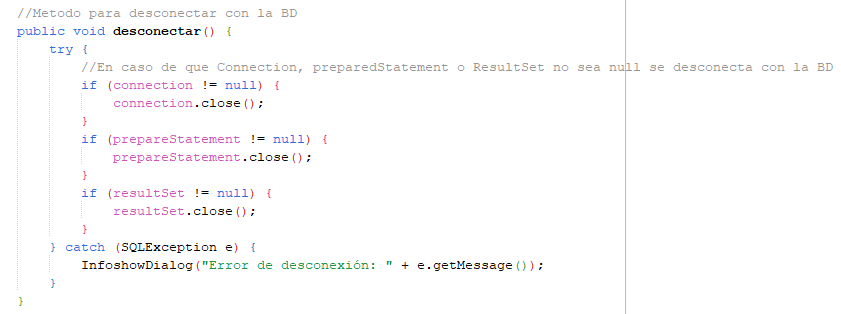


**Carga el driver JDBC.**

**Y se crea una conexión en el que se va a indicar donde se encuentra la BD y el login**



* **Almacenamos el INSERT INTO en una variable**
* **Preparamos el statement** en el que le pasamos el INSERT INTO y más adelante insertamos en los placeholder *(indicándolo con la posición “?” + el parámetro código, nombre, id\_localización, id\_manager)*
* **Ejecutamos la sentencia** mediante la variable preparedStatement junto al método executeUpdate()
* **Desconectamos la BD** al pasar al finally.



**Si connection no es null > cerramos la conexión**

**Si preparedStatement no es null > cerramos la conexión**

**Si resultSet no es null > cerramos la conexión**

### **JFrame form (Interfaz de usuario)**

* Ahora recogeremos los datos que introduzca el usuario y llamaremos a los diferentes métodos ya creados para realizar las diferentes funciones



* Creamos un **objeto de la clase GestorBD**.
* **Llamada al método estático controlarCamposVacios** de la clase CamposVaciosJFrame (controlará que no se deje los campos vacíos a la ahora de introducir un departamento)
* **Si devuelve null** significa que no hay campos TextField en blanco entonces **pasamos a llamar al método de altas.**
* **Llamamos al método mostrar** y este mostrará todos los departamentos que contiene la BD en la tabla de la interfaz gráfica.

### **Resultado Interfaz Gráfica**

* Introducimos los datos, pulsamos en INSERTAR



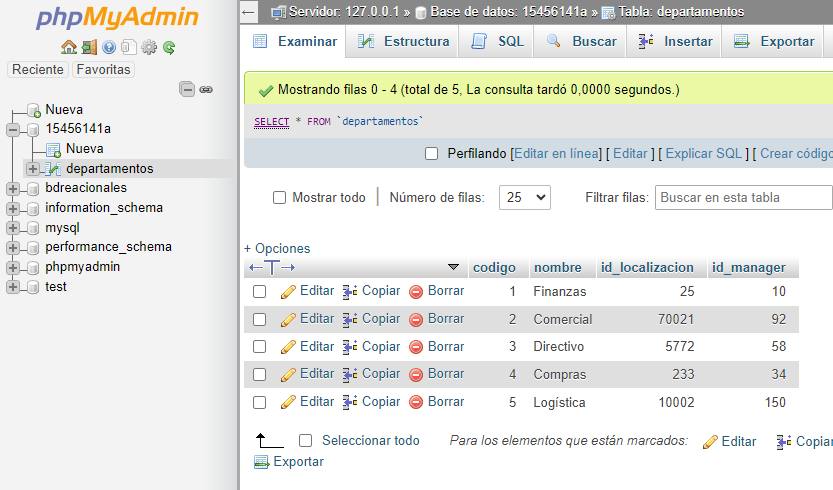
* Aparecerá una ventana emergente indicando que se ha insertado.



* Aparecerá la lista de los departamentos que hay en la BD con el nuevo departamento insertado en la tabla.

****

### Resultado (phpMyAdmin)



# RA09\_e) Se han efectuado borrados y modificaciones sobre la información almacenada.

## Borrar

Se va a recoger el código introducido en el TextField Código que es la Primary Key.

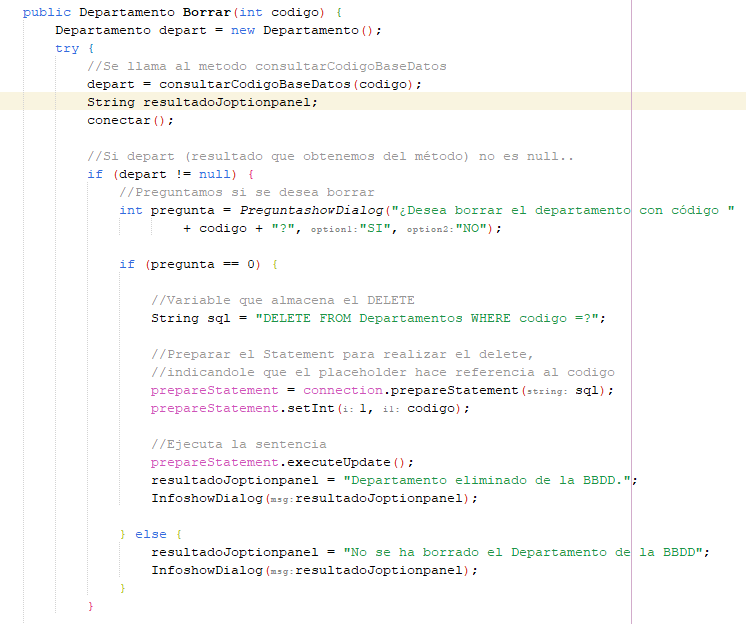
Posteriormente el programa preguntará si está seguro de borrar ese departamento.

Si la respuesta es sí se borrará el registro correspondiente de la tabla departamentos de la BD.

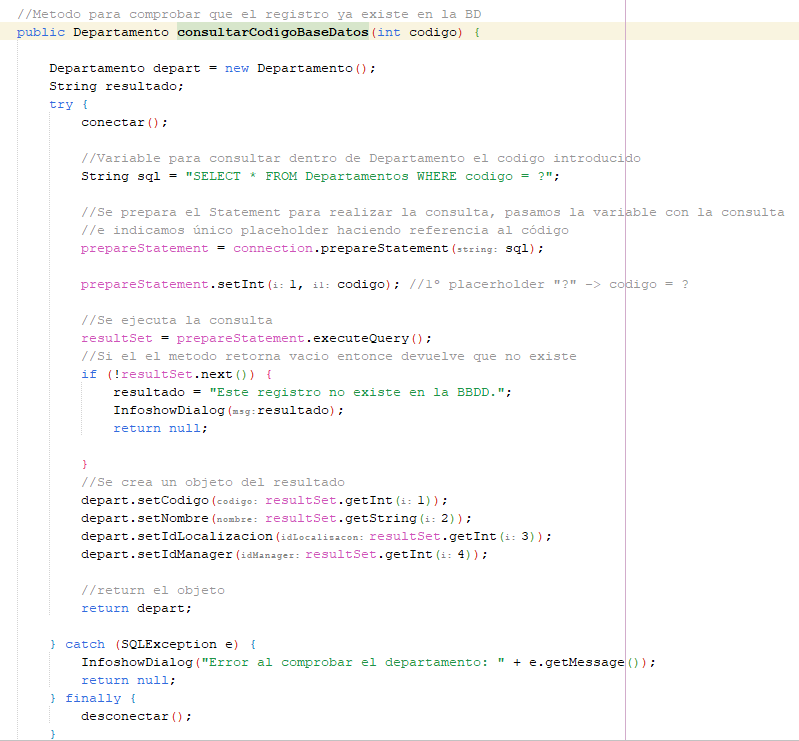
Se rellenará los datos del departamento en los textField.

Por último se mostrarán todos los departamentos que aún quedan en la Tabla.

### Clase GestorBD



* Se llama al método **consultarCodigoBaseDatos** que será almacenado en el objeto de la clase departamento.



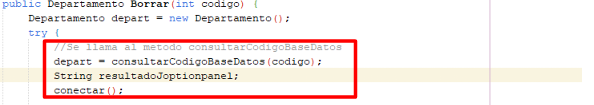
**Conectamos con la BD**

**Preparamos la consulta, indicamos el placeholder y ejecutamos la consulta**

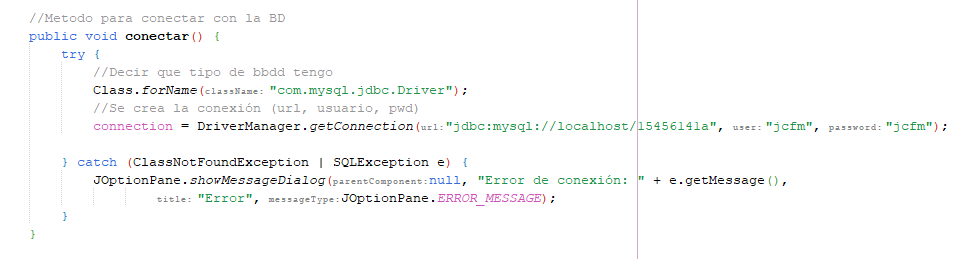
**Una vez ejecutada realizamos un if que si no existe dicho código en la BD mandar una ventana emergente mostrando que no existe ese registro.**

**En cambio si la consulta devuelve un registro, almacenamos dicho registro al objeto depart.**

**Retornamos el depart y cerramos la conexión.**

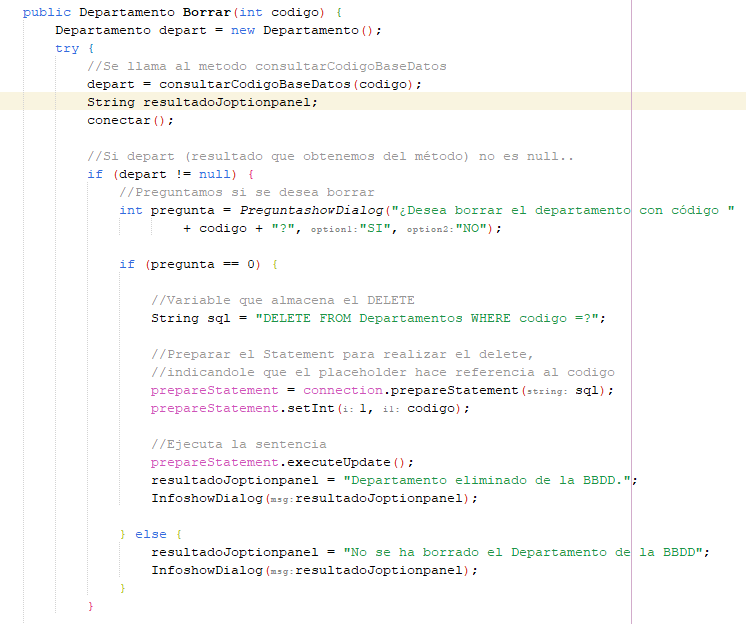


* Una vez realizada la consulta **conectamos con la BD** ya que anteriormente fue desconectada.

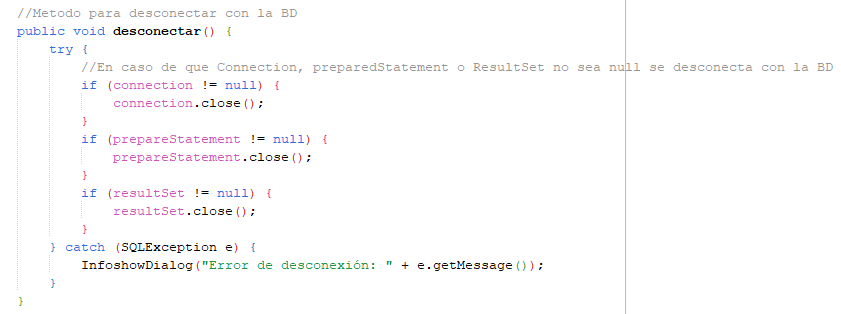


**Carga el driver JDBC.**

**Y se crea una conexión en el que se va a indicar donde se encuentra la BD y el login**



* **Si el objeto depart no es null** entonces **preguntamos si se desea borrar** ese departamento que se ha indicado.
* **Si la respuesta de la pregunta es Si** entonces entra al siguiente if, guardamos la sentencia del DELETE
* **Preparamos el statement** con la variable que almacena la sentencia.
* **Indicamos que el placeholder es el código** que se indica *(código que va a ser recogido desde el textField)*
* **Ejecutamos la sentencia del DELETE** y mostramos por ventaja emergente que se ha eliminado.
* **Si la respuesta de la pregunta es No** entonces se indica mediante una ventaja emergete que no se ha eliminado.
* Por ultimo se **desconecta de la BD** al entrar en el finally.



**Si connection no es null > cerramos la conexión**

**Si preparedStatement no es null > cerramos la conexión**

**Si resultSet no es null > cerramos la conexión**

### JFrame Form (Interfaz de usuario)

* Ahora recogeremos los datos que introduzca el usuario y llamaremos a los diferentes métodos ya creados para realizar las diferentes funciones



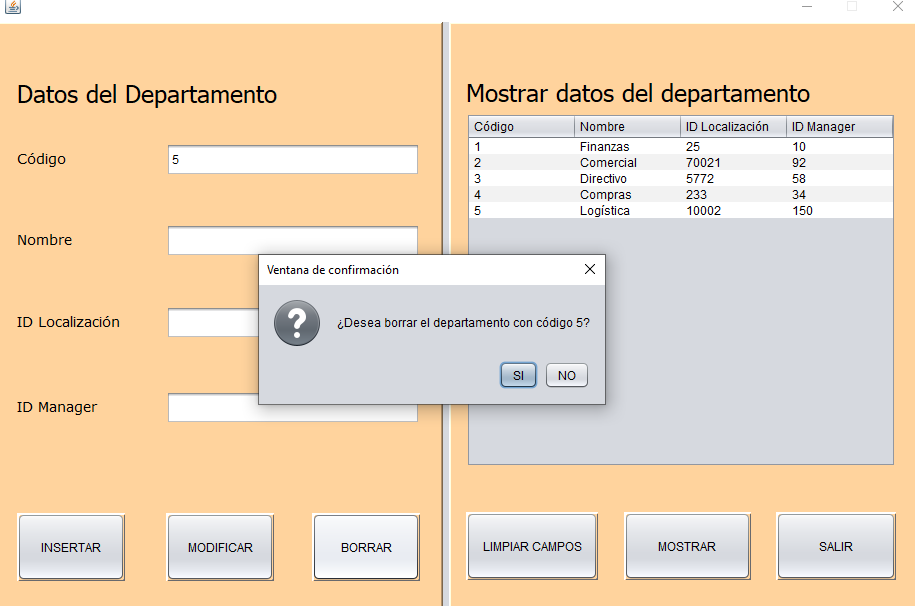
* Creamos un **objeto de la clase GestorBD**.
* **Llamamos al método estático controlarCampoCodigo** de la clase CamposVaciosJFrame, para controlar que no se deje el campo código vacío para poder eliminar el departamento.
* **Si devuelve null** significa que el campo TextField código no está en blanco entonces **pasamos a llamar al método de borrar** almacenándolo en el objeto de la clase departamento.
* **Introducimos los campos** código, nombre, id localización y id manager **en los TextField** que serán recogidos del objeto depart anteriormente almacenado del método borrar.
* Por ultimo llamamos al método mostrar() para mostrar los departamentos en la Tabla con la actualización del borrado.

### Resultado **(Interfaz Gráfica)**

* Introducimos un código ya registrado en la BD para realizar el borrado del departamento.



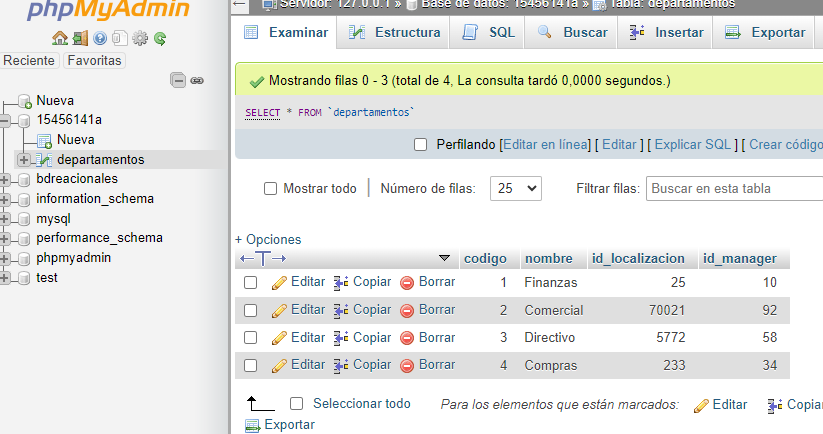
* Al pulsar en **BORRAR**  nos preguntará si se desea borrar el departamento.

****

Al indicar que si entonces **se eliminará el departamento, se auto completará los textFields** **y se mostrará la modificación en la tabla** en este caso se eliminará el departamento indicado

### 

### **Resultado (phpMyAdmin)**



## Modificar

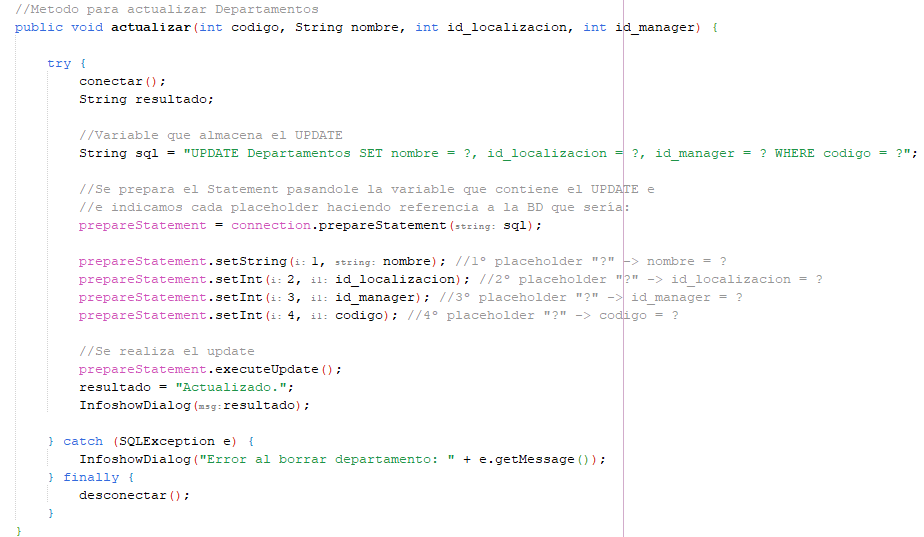
Se va a modificar un departamento introduciendo el código registrado en el textField y la modificación del nombre, id\_localización e id\_manager que se necesite.

Una vez el usuario haya realizado los cambios deseados, al seleccionar la opción de modificar actualizarán los datos en la base de datos.

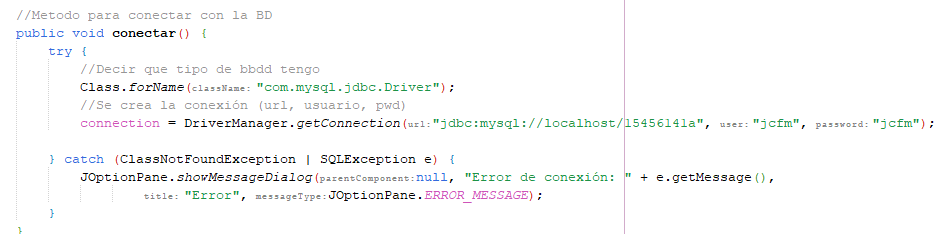
Se mostrará la modificación rellenando los TextFields *(Si se indica el código y se modifica por ejemplo el nombre entonces el resto de campos recogerá los datos del anterior departamento)*

Se mostrarán todos los departamentos en la Tabla, de modo que se pueda comprobar que efectivamente se ha realizado la modificación.

### Clase GestorBD

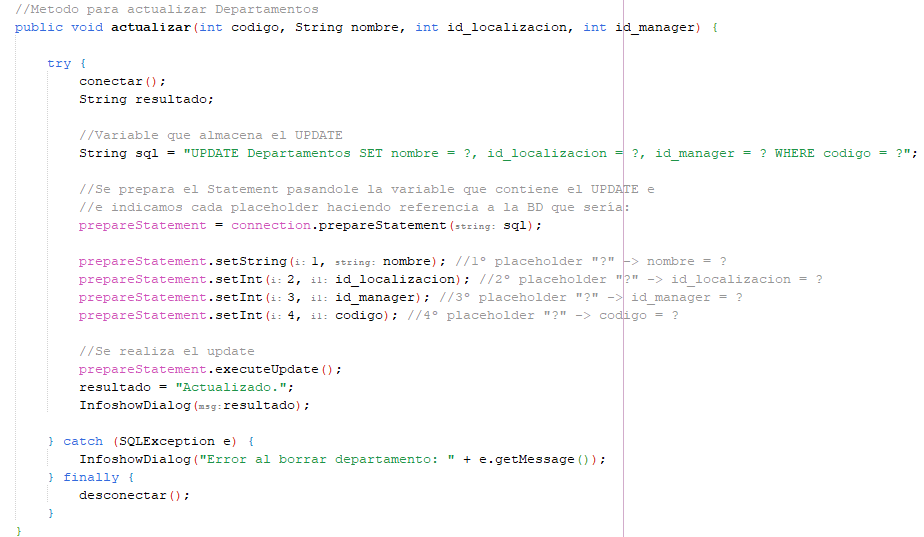


* **Se conectará a la BD** con el método **conectar();**

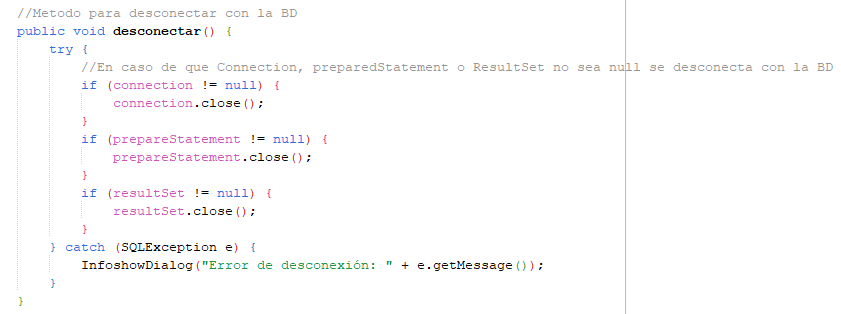


**Carga el driver JDBC.**

**Y se crea una conexión en el que se va a indicar donde se encuentra la BD y el login**



* **Almacenamos** en la variable sql **el UPDATE** que se va a realizar.
* **Preparamos el Statement** pasándole la variable sql
* **Indicamos cada placeholder** donde hace referencia al nombre, id\_localizacion, id\_manager y el código que es controlado con el WHERE de la sentencia.
* **Se ejecuta el Update** y mostramos mediante una ventana emergente que se ha actualizados
* **Desconectamos la BD** al entrar en el finally



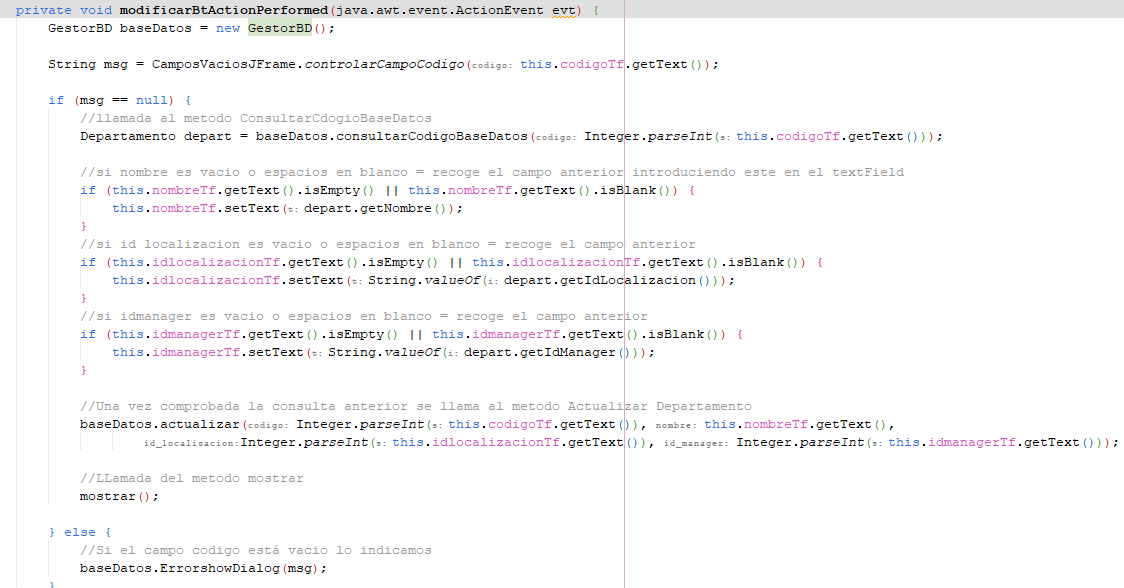
**Si connection no es null > cerramos la conexión**

**Si preparedStatement no es null > cerramos la conexión**

**Si resultSet no es null > cerramos la conexión**

### JFrame form (Interfaz de usuario)

* Ahora recogeremos los datos que introduzca el usuario y llamaremos a los diferentes métodos ya creados para realizar las diferentes funciones



* **Creamos un objeto de la clase GestorBD.**
* **Llamamos al método estático controlarCampoCodigo** de la clase CamposVaciosJFrame, para controlar que no se deje el campo código vacío para poder modificar el departamento.
* **Si devuelve null** significa que el campo TextField código no está en blanco
* **Entonces pasamos a llamar al método de consultarCodigoBaseDatos**, que será almacenado en el objeto depart de la clase departamento
* Mediante los if **se va a controlar que si los textFields nombre, id\_localizacion e id\_manager se encuentra en blanco o con espacios.**
* **Se va rellenar los textFields** con la consulta obtenida con el método consultarCodigoBaseDatos y que se han almacenado en el objeto departamento.
* Una vez comprobada la consulta y rellenando los textFields en blanco si fuera el caso **pasaremos a llamar al método actualizar.**
* Por ultimo llamamos al método mostrar. Mostrando todos los departamentos actualizados de la BD.

### Resultado **(Interfaz Gráfica)**

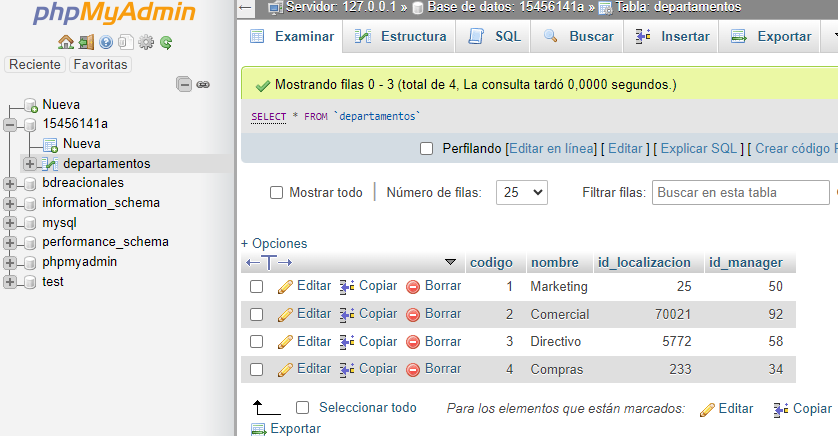
* **Introducimos un código** ya registrado en la BD para actualizar ese registro modificando el departamento.
* Solamente vamos a indicarle un nombre y el ID\_Manager



* Al pulsar en **MODIFICAR** el departamento **1** será actualizado a Marketing pero conservando el ID\_Localización anterior.
* **Se auto completará los TextField** con el departamento modificado recogiendo los datos faltantes y se mostrará la modificación en la Tabla.



### Resultado **(phpMyAdmin)**

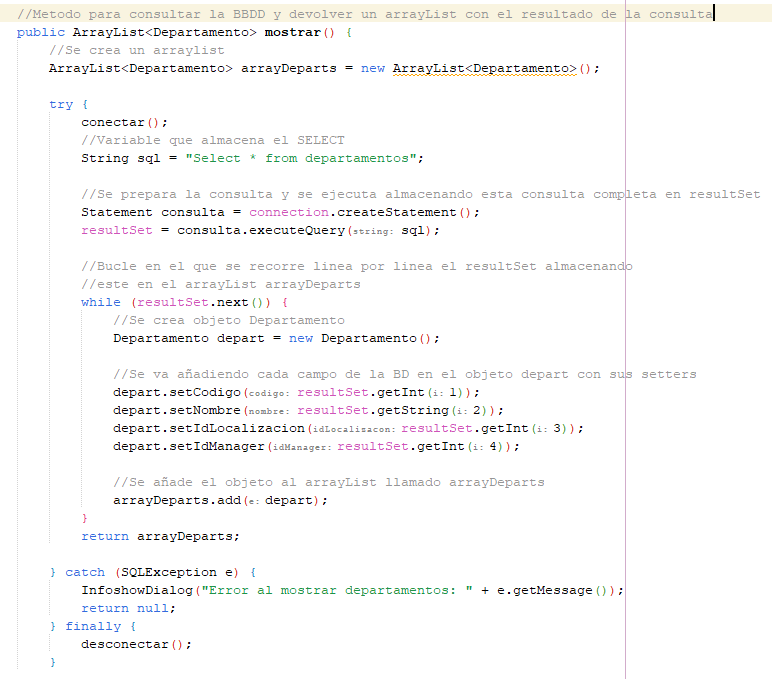


# RA09\_e) Se han efectuado borrados y modificaciones sobre la información almacenada.

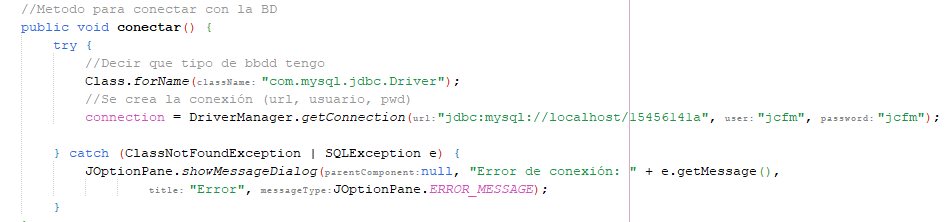
## Mostrar

Para la función mostrar se va a realizar una consulta sobre la tabla departamentos de la base de datos y mostrar toda la información en la Tabla.

### Clase GestorBD



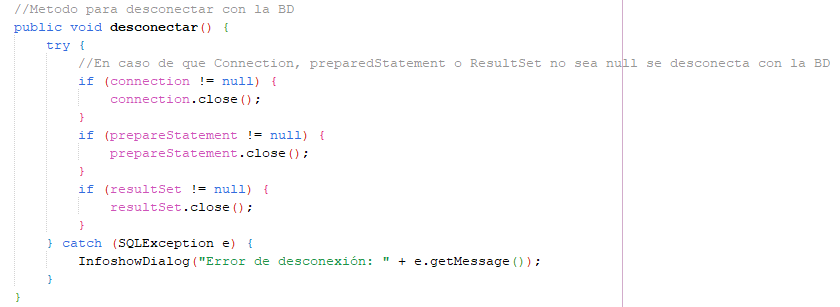
* Se **crea un arrayList** objeto de la clase Departamento
* Conectamos con la BD con el método **conectar();**

****

**Carga el driver JDBC.**

**Y se crea una conexión en el que se va a indicar donde se encuentra la BD y el login**

* Una vez conectada la BD se almacenará en la variable **sql** la consulta SELECT de la tabla departamentos.
* Se prepara la consulta creando el **Statement** y realizamos la consulta con el método **executeQuery()** pasandole la variabla **sql** por parámetros.
* Entraremos en el **bucle while se recorre linea por linea el resultSet para almacenar este en el arrayList arrayDeparts.**
* En cada vuelta del bucle **se añade cada campo de la BD en el objeto depart** y **se añade el objeto en el arrayDeparts.**
* Entrará al finally que desconectará la BD



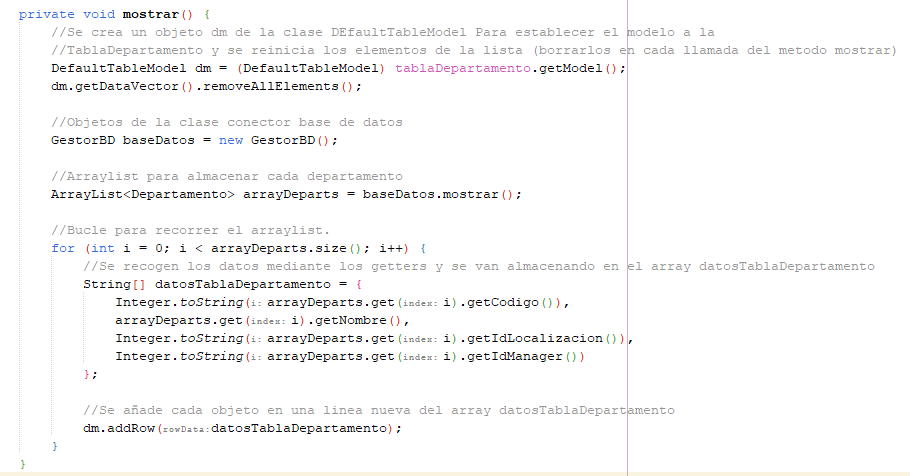
**Si connection no es null > cerramos la conexión**

**Si preparedStatement no es null > cerramos la conexión**

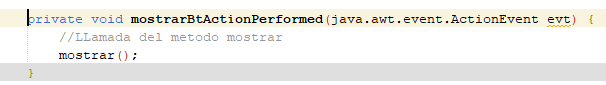
**Si resultSet no es null > cerramos la conexión**

### JFrame form (Interfaz de usuario)

En el JFrame Form se ha creado un método mostrar. Este método va a llamar al otro método mostrar que será almacenado en un arraylist.

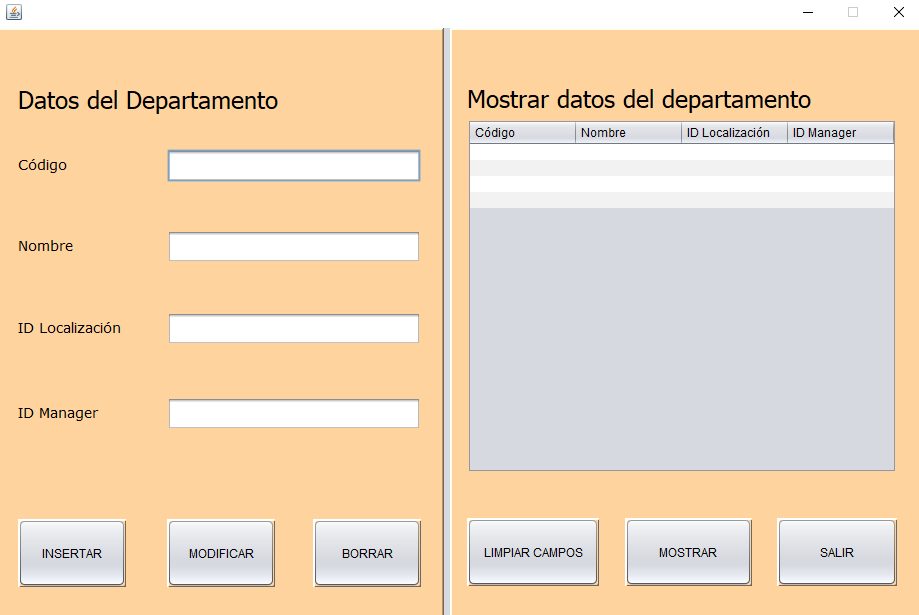


* **Se crea un objeto de la clase DefaulTableModel** con esto establecemos el modelo en tablaDepartamento
* **Reiniciamos el dm** (objeto de la clase DefaulTableModel) para que cada vez que se ejecute este método la tabla se reinicie a 0 y no se dupliquen los campos.
* **Creamos un objeto de la clase baseDatos para poder llamar al método mostrar** de la misma y posteriormente **llamamos a ese método almacenándolo en un arrayList.**
* **Recorremos el arrayList mediante el bucle for**. **Este arrayList será almacenado en un array llamado datosTablaDepartamento** y en **cada vuelta del bucle se irán recogiendo los departamentos.**
* Por último **añadimos el array que se ha obtenido en el bucle for en el dm.**

****

* En el botón mostrar **se llamá al método mostrar()** que se encuentra en el propio JFrame Form.

### Resultado **(Interfaz Gráfica)**



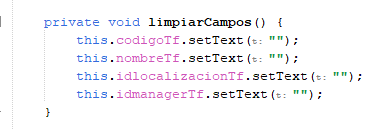
* Pulsamos en el botón **MOSTRAR** y obtendremos la lista de departamentos que hay en la BD

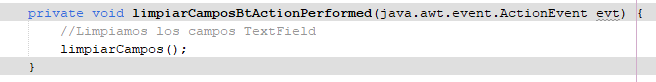


## **Limpiar Campos y salir**

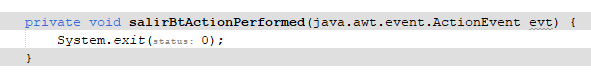
Se han creado dos botones más:

**1 - Función de limpiar los campos textField:**

****

****

**2 - Función para salir de la aplicación:**

****