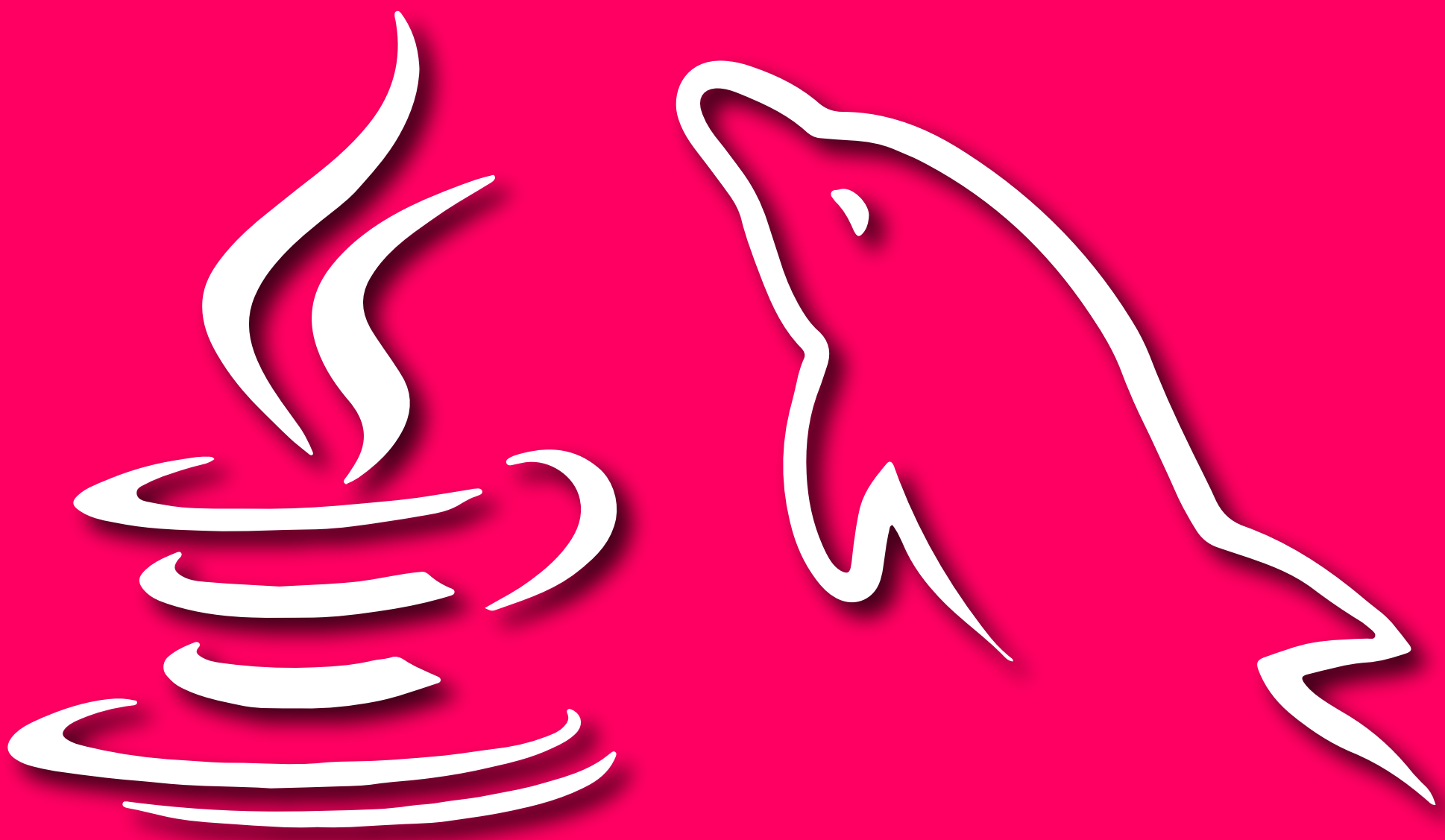
El futuro digital es de todos

MinTIC



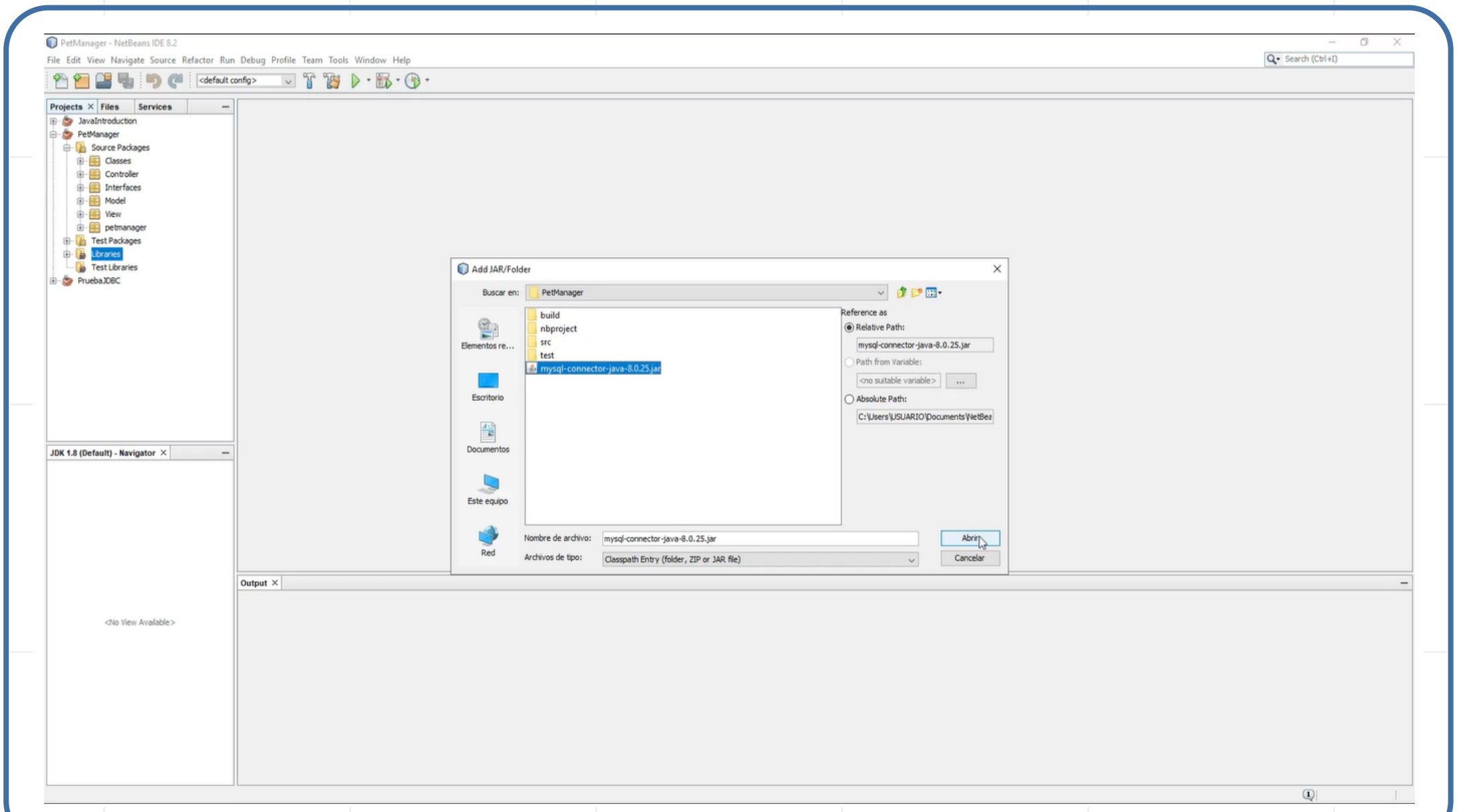
Configuración y Función Guardar con JDBC

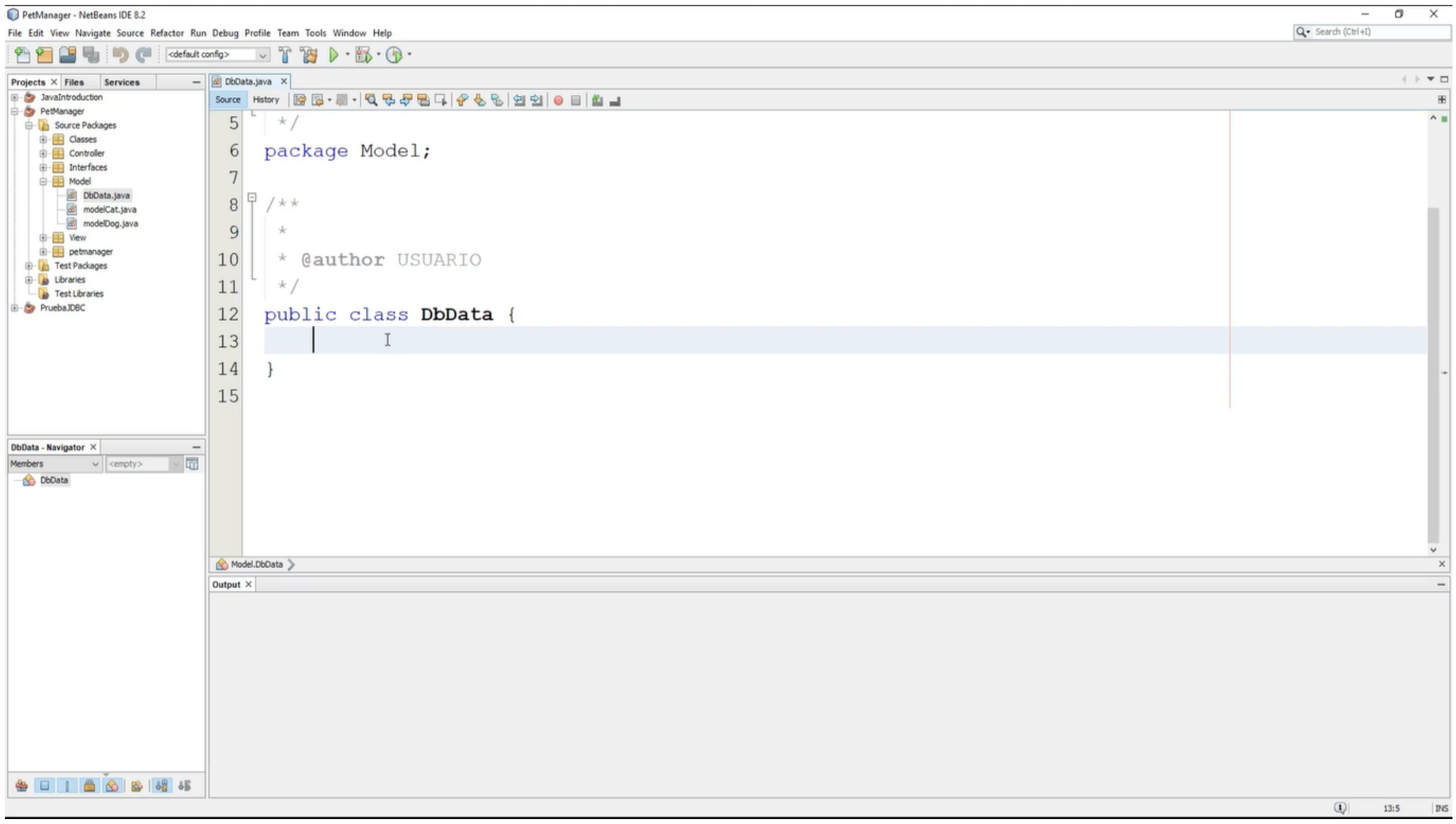
Universidad de Caldas

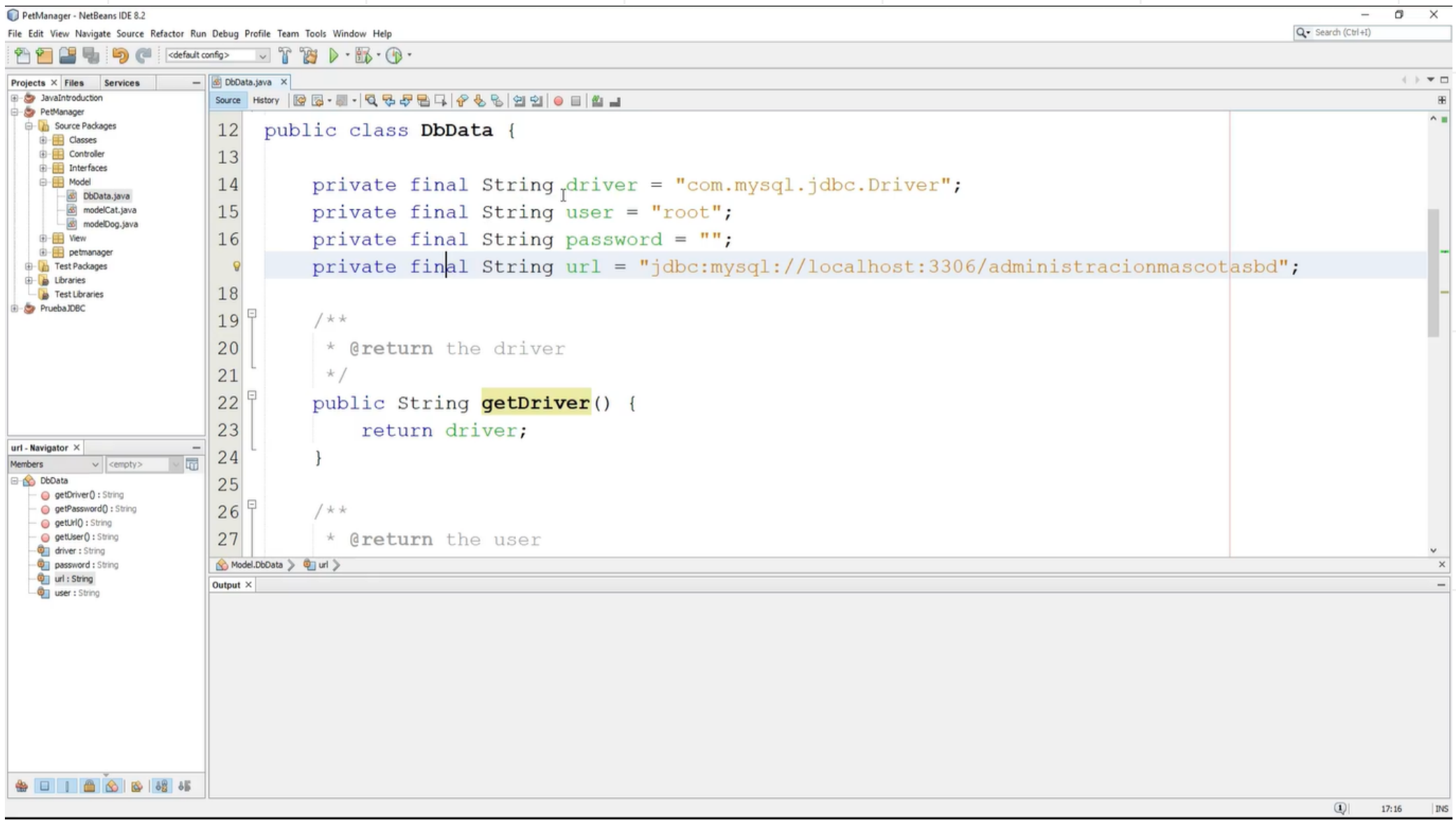
Hola:

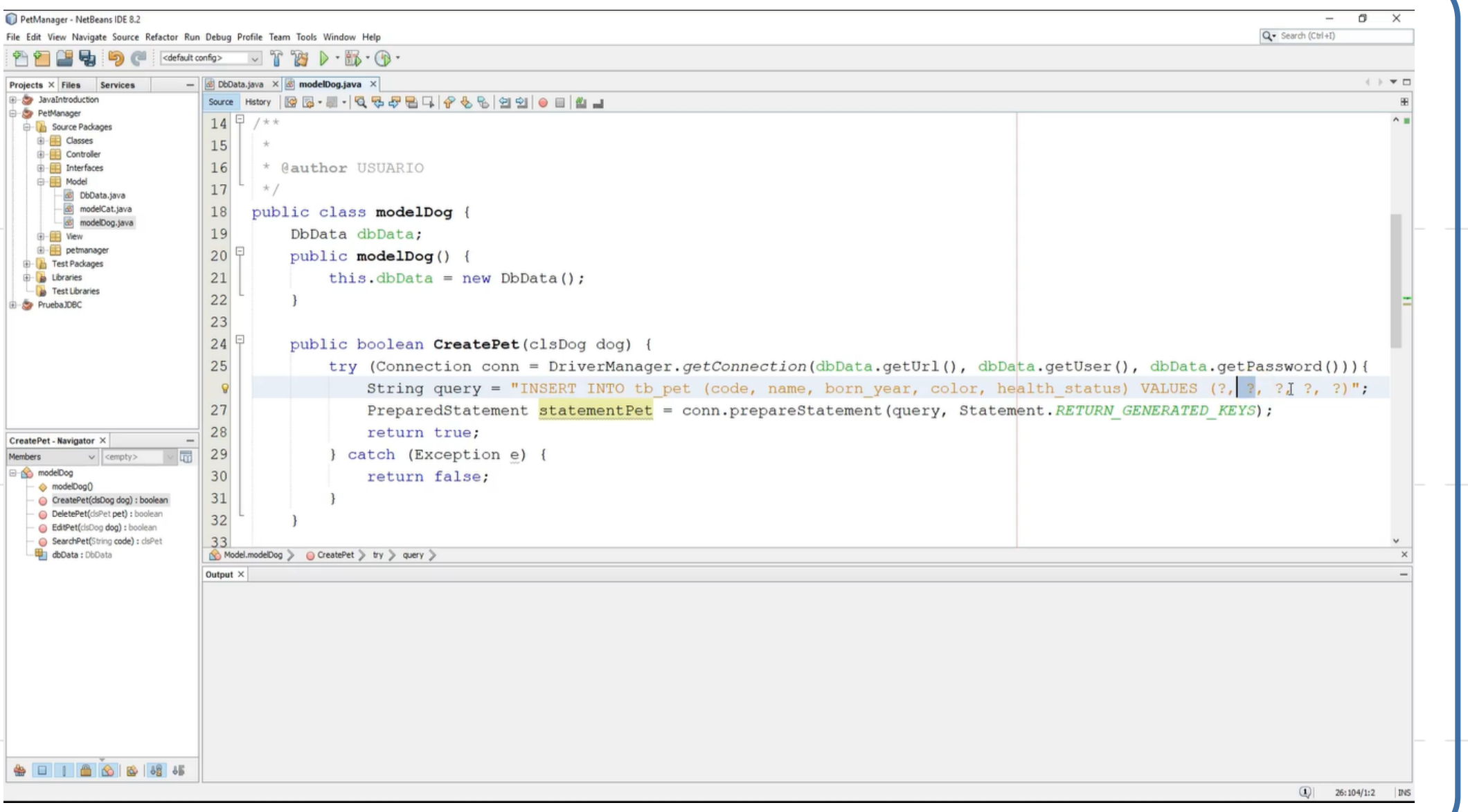
Hemos visto cómo configurar el JDBC para conectar y gestionar la base de datos en Mysql que integraremos y pondremos en funcionamiento en el proyecto de mascotas que hemos trabajado. Veamos cómo agregar el JDBC y cómo ajustar la funcionalidad de almacenamiento de datos con la ayuda de esta librería para que se vea reflejado un proceso de almacenamiento persistente en Mysql mediante la interfaz gráfica de Java.

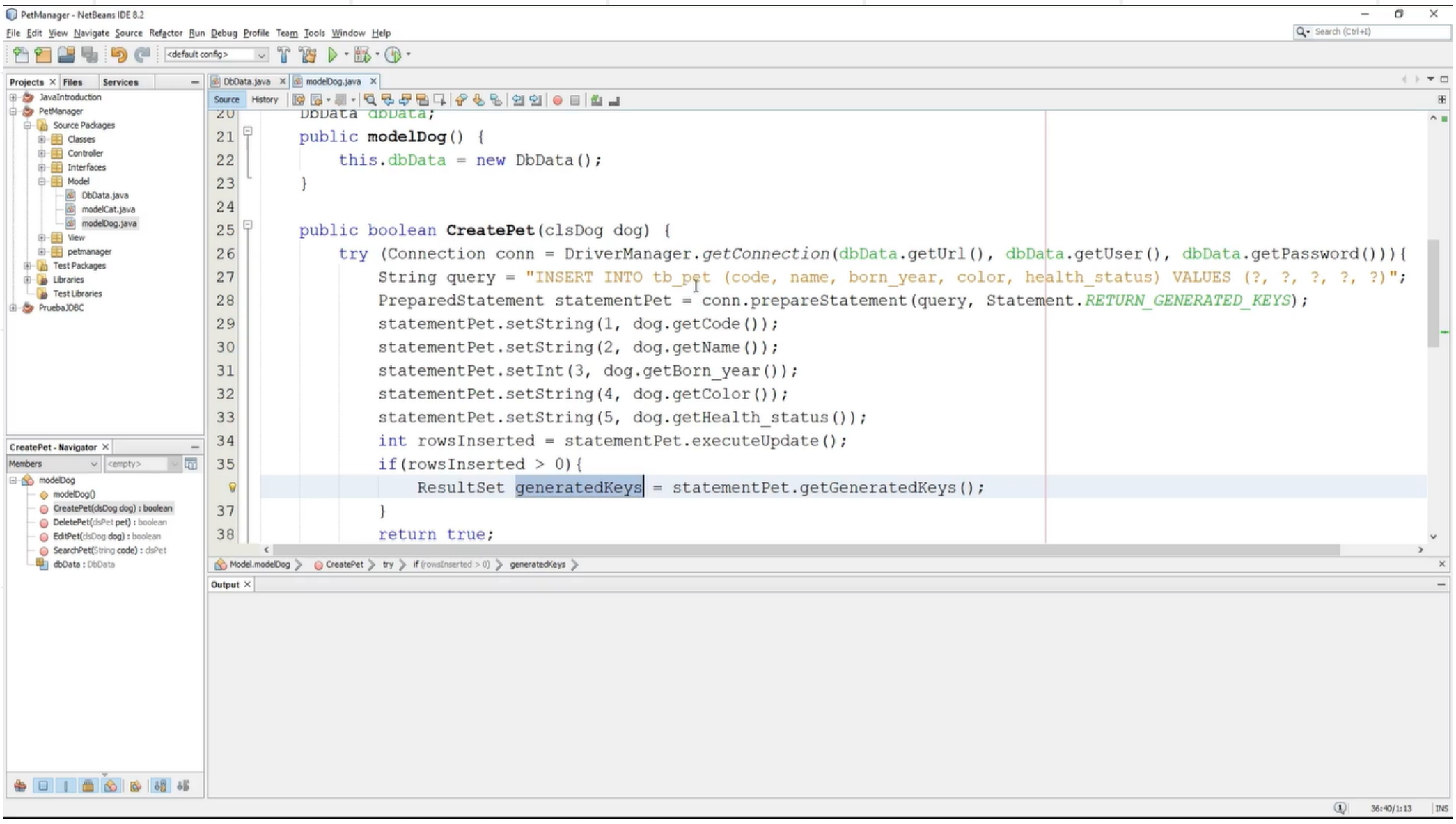
Video de integración de JDBC en el proyecto MVC y funcionalidad de almacenamiento de datos.




PetManager - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Projects Files Services
JavaIntroduction
PetManager
Source Packages
Classes
Controller
Interfaces
Model
DbData.java
modelCat.java
modelDog.java
View
petmanager
Test Packages
Libraries
Test Libraries
PruebaJDBC
DbData.java
Source History
5 */
6 package Model;
7
8 /**
9 *
10 * @author USUARIO
11 */
12 public class DbData {
13
14 }
15
DbData - Navigator
Members
<empty>
DbData
Model.DbData
Output
13:5
INS


PetManager - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Projects Files Services
JavaIntroduction
PetManager
Source Packages
Classes
Controller
Interfaces
Model
DbData.java
modelCat.java
modelDog.java
View
petmanager
Test Packages
Libraries
Test Libraries
PruebaJDBC
DbData.java
Source History
12 public class DbData {
13
14 private final String driver = "com.mysql.jdbc.Driver";
15 private final String user = "root";
16 private final String password = "";
private final String url = "jdbc:mysql://localhost:3306/administracionmascotasbd";
18
19 /**
20 * @return the driver
21 */
22 public String getDriver() {
23 return driver;
24 }
25
26 /**
27 * @return the user
url - Navigator
Members
<empty>
DbData
getDriver() : String
getPassword() : String
getUrl() : String
getUser() : String
driver : String
password : String
url : String
user : String
Model.DbData
url
Output
17:16
INS


PetManager - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects Files Services
JavaIntroduction
PetManager
Source Packages
Classes
Controller
Interfaces
Model
DbData.java
modelCat.java
modelDog.java
View
petmanager
Test Packages
Libraries
Test Libraries
PruebaJDBC
DbData.java
modelDog.java
Source History
14 /**
15 *
16 * @author USUARIO
17 */
18 public class modelDog {
19 DbData dbData;
20 public modelDog() {
21 this.dbData = new DbData();
22 }
24 public boolean CreatePet(clsDog dog) {
25 try (Connection conn = DriverManager.getConnection(dbData.getUrl(), dbData.getUser(), dbData.getPassword())){
26 String query = "INSERT INTO tb_pet (code, name, born_year, color, health_status) VALUES (?, ?, ?, ?, ?)";
27 PreparedStatement statementPet = conn.prepareStatement(query, Statement.RETURN_GENERATED_KEYS);
28 return true;
29 } catch (Exception e) {
30 return false;
31 }
32 }
33
CreatePet - Navigator
Members
modelDog
modelDog()
CreatePet(clsDog dog) : boolean
DeletePet(clsPet pet) : boolean
EditPet(clsDog dog) : boolean
SearchPet(String code) : clsPet
dbData : DbData
Model.modelDog > CreatePet > try > query >
Output
26:104/1:2
INS


PetManager - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects Files Services
JavaIntroduction
PetManager
Source Packages
Classes
Controller
Interfaces
Model
DbData.java
modelCat.java
modelDog.java
View
petmanager
Test Packages
Libraries
Test Libraries
PruebaJDBC
DbData.java
modelDog.java
Source History
DbData dbData;
21 public modelDog() {
22 this.dbData = new DbData();
23 }
24
25 public boolean CreatePet(clsDog dog) {
26 try (Connection conn = DriverManager.getConnection(dbData.getUrl(), dbData.getUser(), dbData.getPassword())){
27 String query = "INSERT INTO tb_pet (code, name, born_year, color, health_status) VALUES (?, ?, ?, ?, ?)";
28 PreparedStatement statementPet = conn.prepareStatement(query, Statement.RETURN_GENERATED_KEYS);
29 statementPet.setString(1, dog.getCode());
30 statementPet.setString(2, dog.getName());
31 statementPet.setInt(3, dog.getBorn_year());
32 statementPet.setString(4, dog.getColor());
33 statementPet.setString(5, dog.getHealth_status());
34 int rowsInserted = statementPet.executeUpdate();
35 if(rowsInserted > 0){
ResultSet generatedKeys = statementPet.getGeneratedKeys();
37 }
38 return true;
CreatePet - Navigator
Members
modelDog
modelDog()
CreatePet(clsDog dog) : boolean
DeletePet(clsPet pet) : boolean
EditPet(clsDog dog) : boolean
SearchPet(String code) : clsPet
dbData : DbData
Model.modelDog > CreatePet > try > if (rowsInserted > 0) > generatedKeys
Output
36:40/1:13
INS

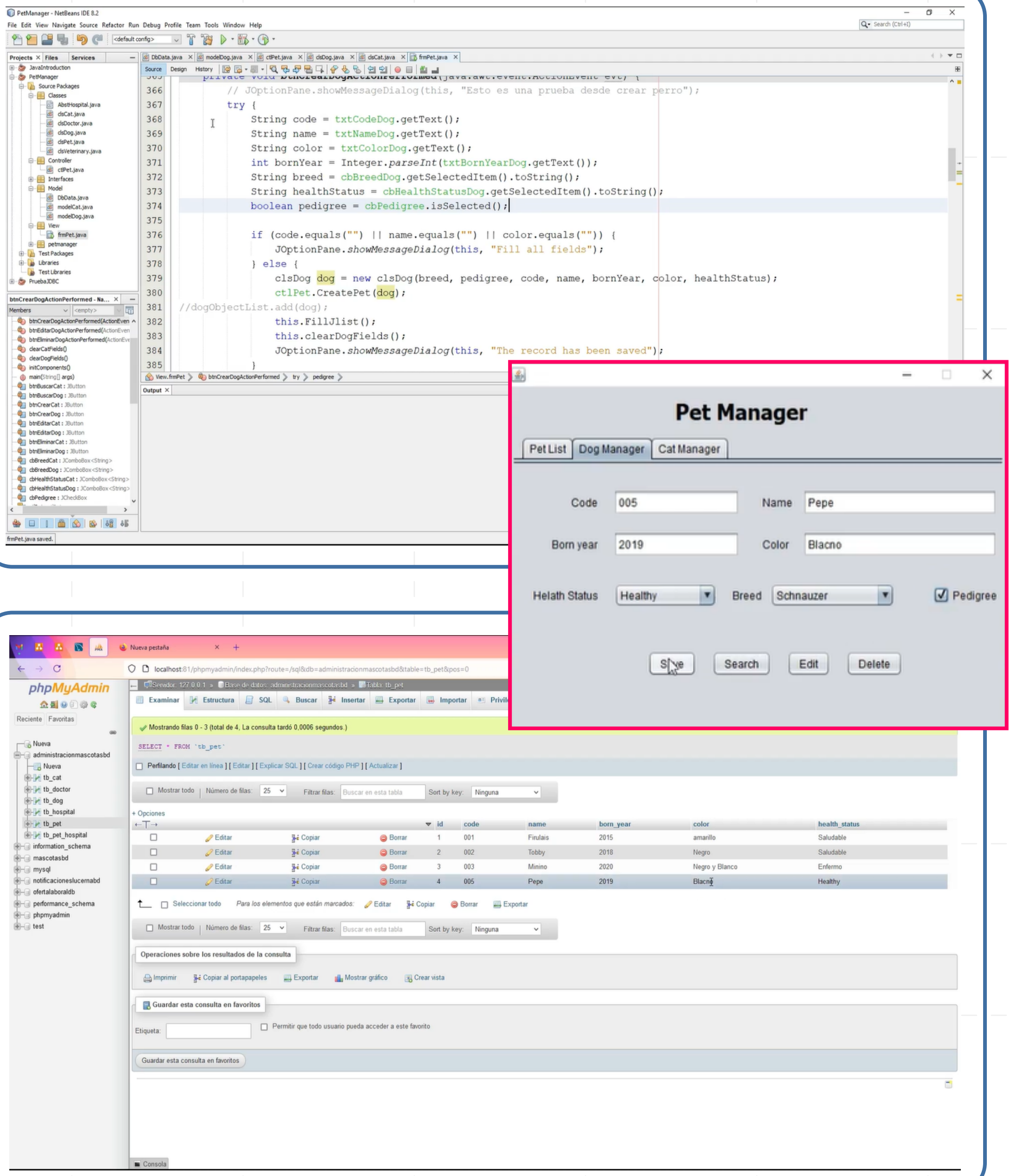
```
ResultSet generatedKeys = statementPet.getGeneratedKeys();
if(generatedKeys.next()){
    int idPet = generatedKeys.getInt(1);
    query = "INSERT INTO tb_dog (breed, pedigree, id_pet) VALUES (?, ?, ?)";
    PreparedStatement statementDog = conn.prepareStatement(query);
    statementDog.setString(1, dog.getBreed());
    statementDog.setBoolean(2, dog.isPedigree());
    statementDog.setInt(3, idPet);
    rowsInserted = statementDog.executeUpdate();
    if(rowsInserted > 0){
        return true;
    }
}
}
return false;
} catch (SQLException e) {
    return false;
}
```

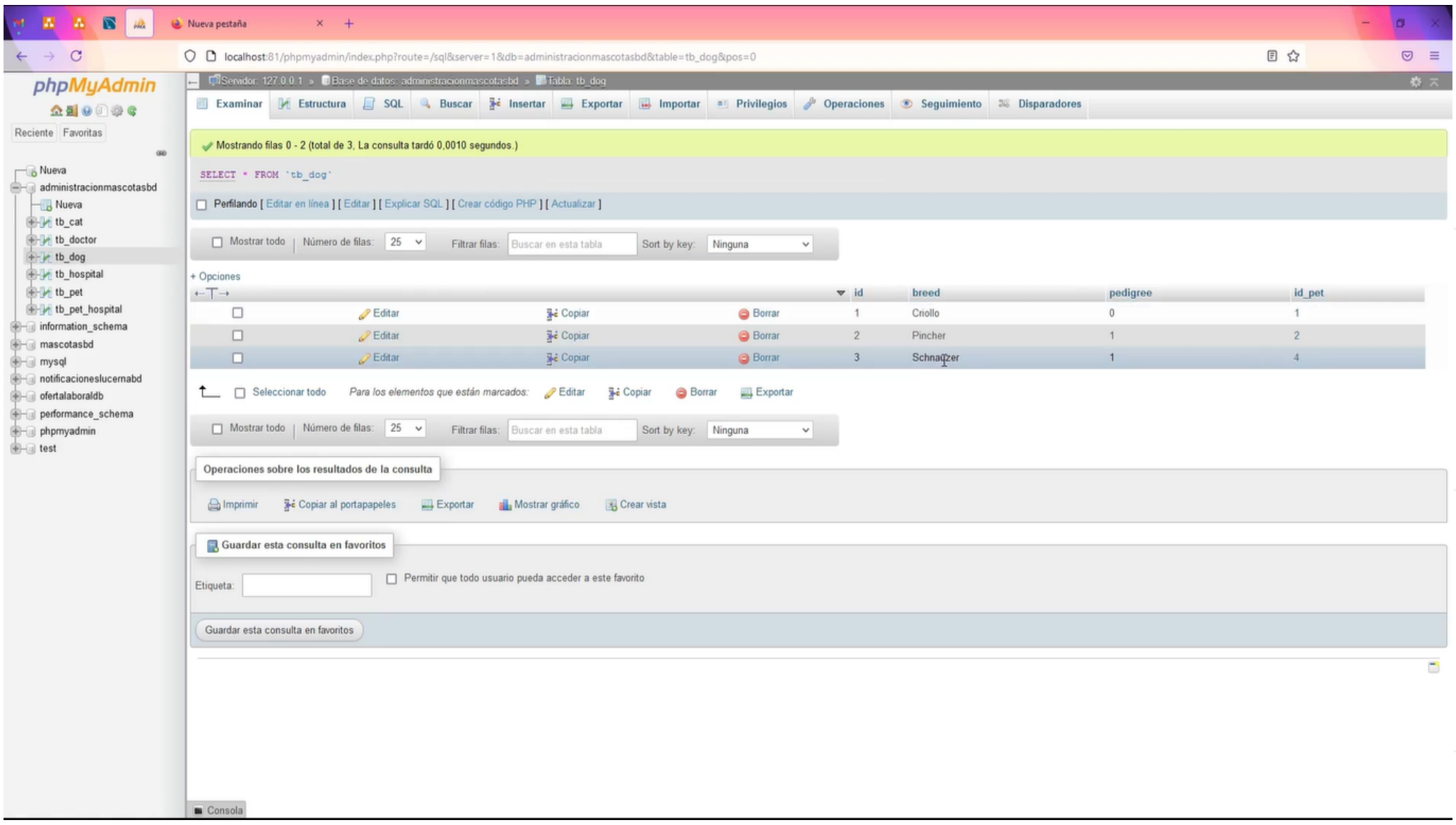
```
@Override
public void Sound(){
    System.out.println("El perro " + super.getName() +" hace Guuuuaaaauuu");
}

/**
 * @return the breed
 */
public String getBreed() {
    return breed;
}


@Override
public String getAnimalType(){
    return "Dog";
}

/**
 * @param breed the breed to set
 */
public void setBreed(String breed) {
```


PetManager - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects Files Services
JavaIntroduction
PetManager
Source Packages
Classes
AbstHospital.java
clsCat.java
clsDoctor.java
clsDog.java
clsPet.java
clsVeterinary.java
Controller
ctlPet.java
Interfaces
Model
DbData.java
modelCat.java
modelDog.java
View
frmPet.java
petmanager
Test Packages
Libraries
Test Libraries
PruebaJDBC
DbData.java
modelDog.java
ctlPet.java
clsDog.java
clsCat.java
frmPet.java
Source Design History
// JOptionPane.showMessageDialog(this, "Esto es una prueba desde crear perro");
try {
String code = txtCodeDog.getText();
String name = txtNameDog.getText();
String color = txtColorDog.getText();
int bornYear = Integer.parseInt(txtBornYearDog.getText());
String breed = cbBreedDog.getSelectedItem().toString();
String healthStatus = cbHealthStatusDog.getSelectedItem().toString();
boolean pedigree = cbPedigree.isSelected();
if (code.equals("") || name.equals("") || color.equals("")) {
JOptionPane.showMessageDialog(this, "Fill all fields");
} else {
clsDog dog = new clsDog(breed, pedigree, code, name, bornYear, color, healthStatus);
ctlPet.CreatePet(dog);
//dogObjectList.add(dog);
this.FillJlist();
this.clearDogFields();
JOptionPane.showMessageDialog(this, "The record has been saved");
}
View.frmPet > btnCrearDogActionPerformed > try > pedigree >
Output
btnCrearDogActionPerformed - Na...
Members
btnCrearDogActionPerformed(ActionEven
btnEditarDogActionPerformed(ActionEven
btnEliminarDogActionPerformed(ActionEv
clearCatFields()
clearDogFields()
initComponents()
main(String[] args)
btnBuscarCat : JButton
btnBuscarDog : JButton
btnCrearCat : JButton
btnCrearDog : JButton
btnEditarCat : JButton
btnEditarDog : JButton
btnEliminarCat : JButton
btnEliminarDog : JButton
cbBreedCat : JComboBox<String>
cbBreedDog : JComboBox<String>
cbHealthStatusCat : JComboBox<String>
cbHealthStatusDog : JComboBox<String>
cbPedigree : JCheckBox
frmPet.java saved.
Pet Manager
Pet List
Dog Manager
Cat Manager
Code
005
Name
Pepe
Born year
2019
Color
Blacno
Helath Status
Healthy
Breed
Schnauzer
Pedigree
Save
Search
Edit
Delete
Nueva pestaña
localhost:81/phpmyadmin/index.php?route=/sql&db=administracionmascotasbd&table=tb_pet&pos=0
phpMyAdmin
Reciente
Favoritas
Nueva
administracionmascotasbd
Nueva
tb_cat
tb_doctor
tb_dog
tb_hospital
tb_pet
tb_pet_hospital
information_schema
mascotasbd
mysql
notificacioneslucernabd
ofertalaboraldb
performance_schema
phpmyadmin
test
Servidor: 127.0.0.1 » Base de datos: administracionmascotasbd » Tabla: tb_pet
Examinar
Estructura
SQL
Buscar
Insertar
Exportar
Importar
Mostrando filas 0 - 3 (total de 4, La consulta tardó 0,0006 segundos.)
SELECT * FROM `tb_pet`
Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]
Mostrar todo
Número de filas: 25
Filtrar filas: Buscar en esta tabla
Sort by key: Ninguna
+ Opciones
id code name born_year color health_status
Editar Copiar Borrar 1 001 Firulais 2015 amarillo Saludable
Editar Copiar Borrar 2 002 Tobby 2018 Negro Saludable
Editar Copiar Borrar 3 003 Minino 2020 Negro y Blanco Enfermo
Editar Copiar Borrar 4 005 Pepe 2019 Blacno Healthy
Seleccionar todo
Para los elementos que están marcados:
Editar
Copiar
Borrar
Exportar
Mostrar todo
Número de filas: 25
Filtrar filas: Buscar en esta tabla
Sort by key: Ninguna
Operaciones sobre los resultados de la consulta
Imprimir
Copiar al portapapeles
Exportar
Mostrar gráfico
Crear vista
Guardar esta consulta en favoritos
Etiqueta:
Permitir que todo usuario pueda acceder a este favorito
Guardar esta consulta en favoritos
Consola



Hemos aprendido cómo integrar el JDBC en el proyecto MVC, además, de utilizarlo para almacenar información persistentemente en la base de datos de Mysql. Practiquemos con la solución del reto de la semana y profundicemos en esta temática con el material adicional.

Universidad de Caldas