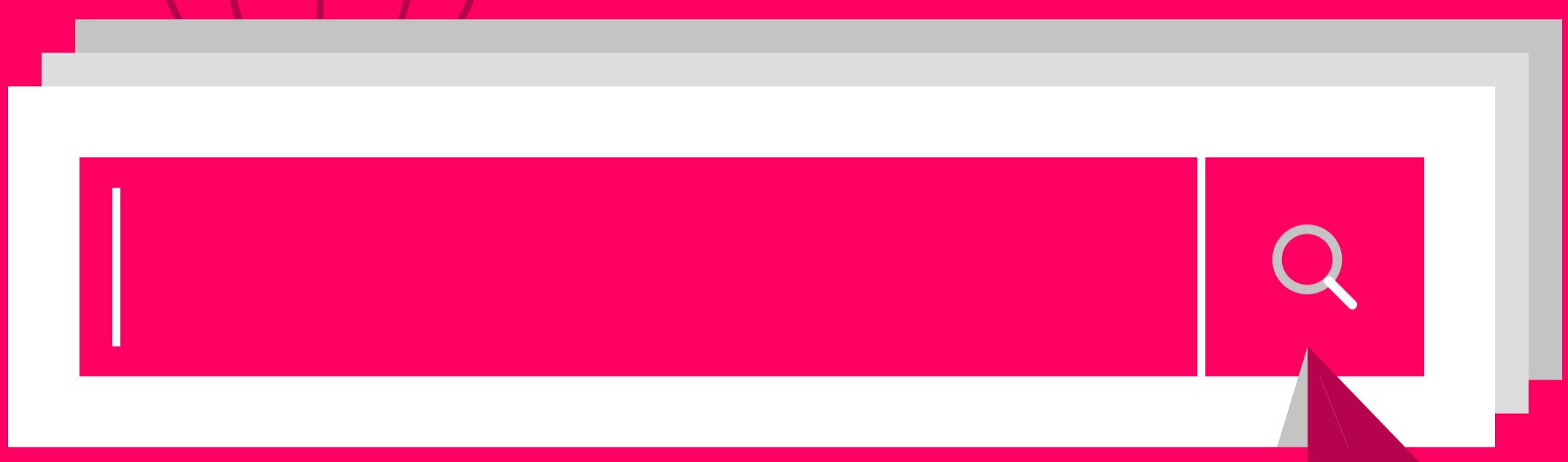
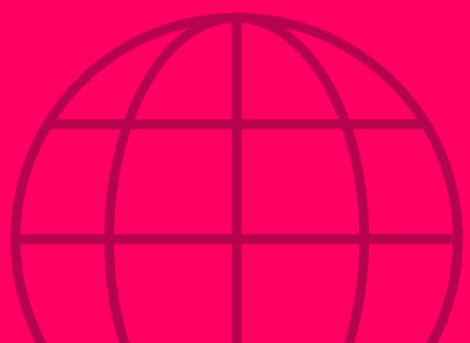




El futuro digital  
es de todos

MinTIC



# Función Buscar con JDBC

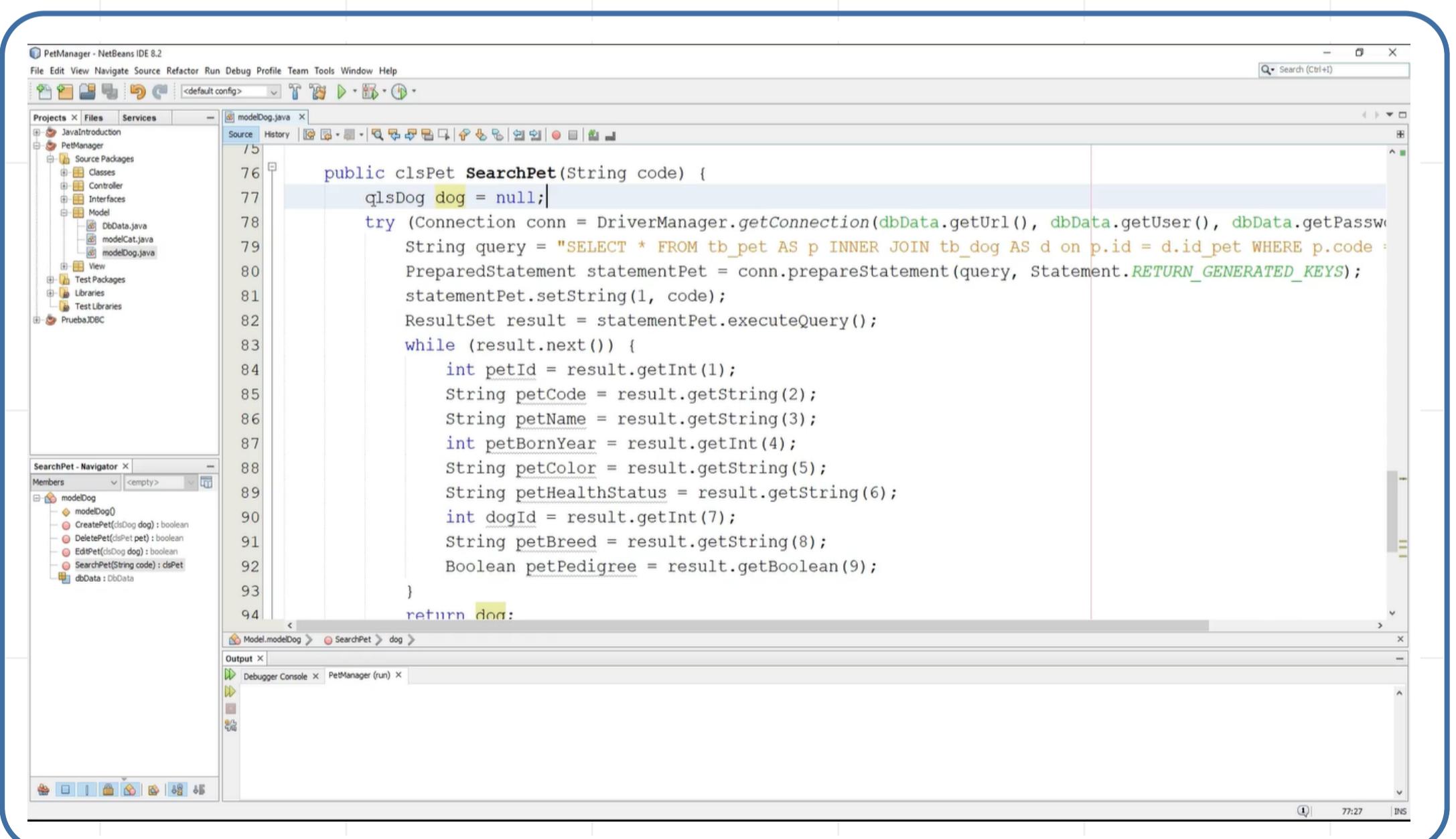


Universidad de Caldas

# Hola:

Una vez configurado el JDBC para conectar y gestionar la base de datos en Mysql con la funcionalidad de creación de registros se debe integrar las demás operaciones del CRUD en la capa de modelo con el apoyo del JDBC para así completar la gestión de mascotas en nuestra aplicación y para ello buscaremos información para actualizarla posteriormente y listarla como un reporte.

Video de implementación de las funcionalidades de búsqueda de información.



The screenshot shows the NetBeans IDE interface with the following details:

- Title Bar:** PetManager - NetBeans IDE 8.2
- Menu Bar:** File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
- Toolbar:** Includes icons for file operations like Open, Save, and Build.
- Project Explorer:** Shows a project named "PetManager" with packages like "Source Packages" containing "Controller", "Interfaces", and "Model" which includes "modelDog.java". Other packages like "View", "Test Packages", "Libraries", and "Test Libraries" are also listed.
- Code Editor:** The "modelDog.java" file is open, showing Java code for a "SearchPet" method. The code uses JDBC to query a MySQL database for a dog based on its code. It handles the connection, prepares the statement, executes the query, and loops through the results to build a "clsPet" object named "dog".

```
public clsPet SearchPet(String code) {
    clsDog dog = null;
    try (Connection conn = DriverManager.getConnection(dbData.getUrl(), dbData.getUser(), dbData.getPassword())) {
        String query = "SELECT * FROM tb_pet AS p INNER JOIN tb_dog AS d on p.id = d.id_pet WHERE p.code = ?";
        PreparedStatement statementPet = conn.prepareStatement(query, Statement.RETURN_GENERATED_KEYS);
        statementPet.setString(1, code);
        ResultSet result = statementPet.executeQuery();
        while (result.next()) {
            int petId = result.getInt(1);
            String petCode = result.getString(2);
            String petName = result.getString(3);
            int petBornYear = result.getInt(4);
            String petColor = result.getString(5);
            String petHealthStatus = result.getString(6);
            int dogId = result.getInt(7);
            String petBreed = result.getString(8);
            Boolean petPedigree = result.getBoolean(9);
        }
    }
    return dog;
}
```
- Navigator:** Shows the members of the "modelDog" class, including the "SearchPet" method and other methods like "CreatePet", "DeletePet", "EditPet", and "SearchPet".
- Output:** Shows the "Debugger Console" and "PetManager (run)" tabs.

The screenshot shows the phpMyAdmin interface for the database 'administracionmascotasbd'. In the main window, a SQL query is being run:

```
1 SELECT * FROM tb_pet AS p INNER JOIN tb_dog AS d on p.id = d.id_pet WHERE p.code = '005'
```

The results of the query are displayed in a table:

	id	code	name	born_year	color	health_status	id	breed	pedigree	id_pet
4	005	Pepe	2019	Blanco	Healthy		3	Schnauzer	1	4

The screenshot shows the NetBeans IDE version 8.2 with a Java project named 'PetManager'. The 'modelDog.java' file is open, and a refactoring dialog titled 'Encapsulate Fields' is displayed over the code. The code in 'modelDog.java' includes methods like Eat(), Move(), Sound(), and getCode(). The refactoring dialog lists fields to encapsulate, including 'petId', 'code', 'name', 'born\_year', 'color', and 'health\_status'. The 'Refactor' button is highlighted.

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

View

Test Packages

Libraries

Test Libraries

PruebaJDBC

dsDog - Navigator X

Members

dsDog :: clsPet

- Sound0 : clsPet
- WalkAround0
- WalkAround(n km)
- WalkAround(boolean dogLeash)
- getAnimalType() : String ↗ clsPet
- getBreed() : String
- isPedigree() : boolean
- setBreed(String breed)
- setPedigree(boolean pedigree)

breed : String

dogId : int

pedigree : boolean

dsDog.java saved.

Source History

```
1 /*  
2  * To change this license header, choose License Headers in Project Properties.  
3  * To change this template file, choose Tools | Templates  
4  * and open the template in the editor.  
5 */  
6 package Classes;  
7  
8 /**  
9  *  
10 * @author USUARIO  
11 */  
12 public class clsDog extends clsPet {  
13     int dogId;  
14     private String breed;  
15     private boolean pedigree;  
16  
17     public void WalkAround(){  
18         System.out.println("El perro " + super.getName() + " está caminando.");  
19     }  
20 }
```

Output

Debugger Console X PetManager (run) X

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

View

Test Packages

Libraries

Test Libraries

PruebaJDBC

modelDog - Navigator X

Members

modelDog

- modelDog0
- CreatePet(dsDog dog) : boolean
- DeletePet(dsPet pet) : boolean
- EditPet(dsDog dog) : boolean
- SearchPet(String code) : dsPet

dbData : DbData

modelDog.java saved.

Source History

```
80 IStatement statementPet = conn.prepareStatement(query);  
81     pet.setString(1, code);  
82     result = statementPet.executeQuery();  
83     result.next()); {  
84         petId = result.getInt(1);  
85         petCode = result.getString(2);  
86         petName = result.getString(3);  
87         petBornYear = result.getInt(4);  
88         petColor = result.getString(5);  
89         petHealthStatus = result.getString(6);  
90         dogId = result.getInt(7);  
91         petBreed = result.getString(8);  
92         petPedigree = result.getBoolean(9);  
93         = new clsDog(dogId, petBreed, petPedigree, petId, petCode, petName, petBornYear, petColor, petHealthStatus);  
94  
95     log;  
96     :option e) {  
97     log;  
98 }
```

Output

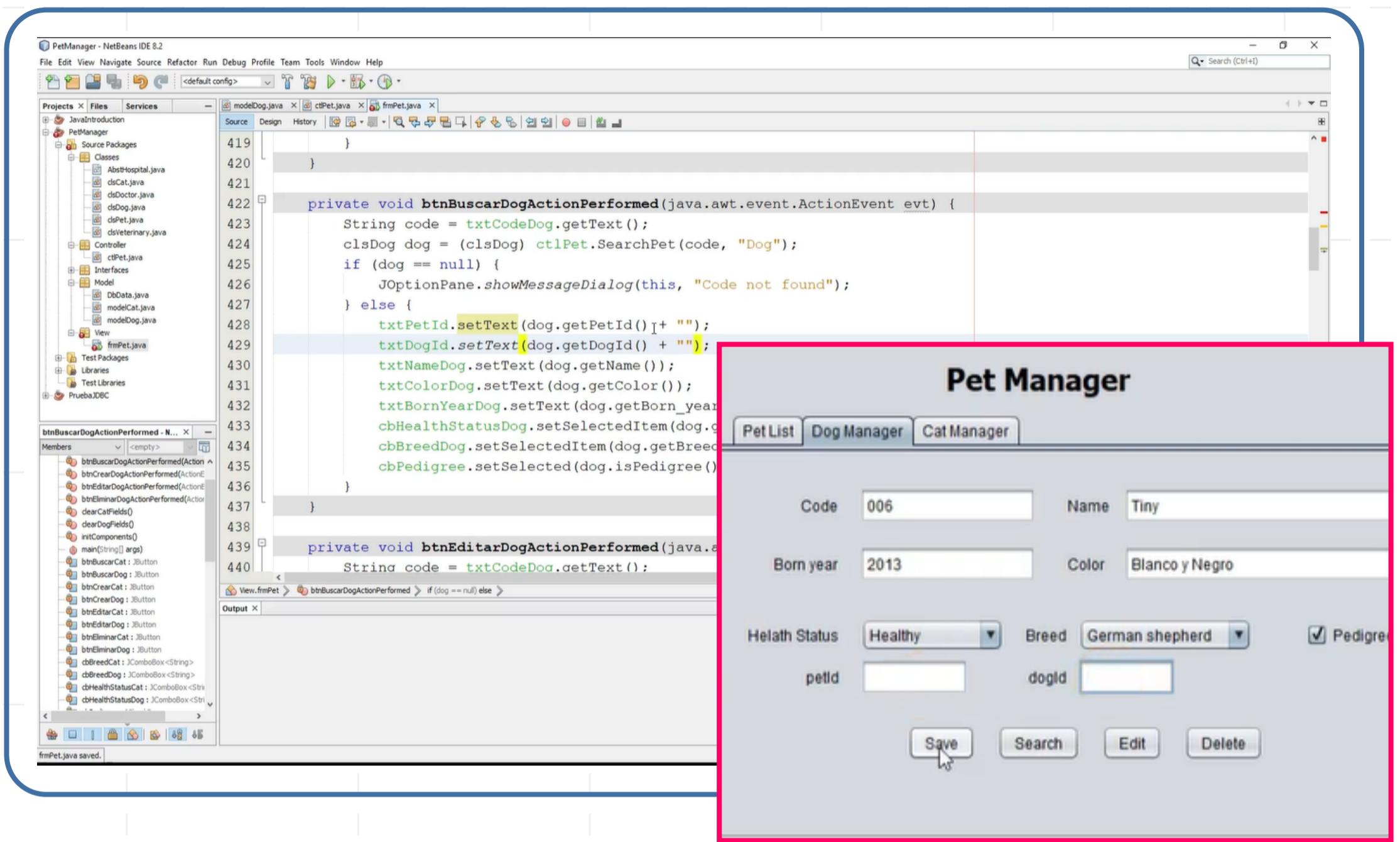
Debugger Console X PetManager (run) X

The screenshot shows the NetBeans IDE interface with the following details:

- Title Bar:** PetManager - NetBeans IDE 8.2
- Menu Bar:** File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
- Search Bar:** Search (Ctrl+I)
- Toolbar:** Standard NetBeans toolbar icons.
- Projects Tab:** Projects, Files, Services (selected).
- Project Explorer:** Shows the PetManager project structure:
  - JavaIntroduction
  - PetManager
    - Source Packages
      - Classes
        - AbstHospital.java
        - dsCat.java
        - dsDoctor.java
        - dsDog.java
        - dsPet.java
        - dsVeterinary.java
      - Controller
        - ctlPet.java
      - Interfaces
      - Model
        - DbData.java
        - modelCat.java
        - modelDog.java
      - View
    - Test Packages
    - Libraries
    - Test Libraries
    - PruebaJDBC
  - Code Editor:** Controller.ctlPet.java (SearchPet method). The line `pet = this.modelDog.SearchPet(code);` is highlighted.
  - Navigator:** SearchPet - Navigator X (Members tab, empty).
  - Output:** Debugger Console X, PetManager (run) X.
  - Bottom:** Standard NetBeans status bar with icons and text (82:44, INS).

The screenshot shows the NetBeans IDE interface with the following details:

- Title Bar:** PetManager - NetBeans IDE 8.2
- Menu Bar:** File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
- Toolbar:** Includes icons for New, Open, Save, Cut, Copy, Paste, Find, Replace, and others.
- Project Explorer:** Shows the project structure under "JavaSource".
  - 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).
  - Test Packages: Libraries, Test Libraries, PruebaJDBC.
- Code Editor:** The main window displays Java code for `frmPet.java`. The code handles button actions for searching and editing dogs. It uses `ActionEvent`, `clsDog`, and `JOptionPane` classes.
- Members List:** Shows the members of the `frmPet` class, including methods like `btnBuscarDogActionPerformed` and `btnEditarDogActionPerformed`.
- Output:** Shows the stack trace of the current run: `frmPet > btnBuscarDogActionPerformed > if (dog == null) else >`.
- Bottom Bar:** Includes icons for Run, Stop, Build, and others.



The screenshot shows the phpMyAdmin interface connected to the "administracionmascotasbd" database, specifically the "tb\_pet" table. The table structure is as follows:

	color	health_status
amarillo	Saludable	
Negro	Saludable	
Negro y Blanco	Enfermo	
Blanco	Healthy	
Blanco y Negro	Healthy	

A modal dialog from the "Pet Manager" application is overlaid on the phpMyAdmin interface. The modal has the title "Pet Manager" and contains fields for "Code" (006), "Name" (empty), "Born year" (empty), "Color" (empty), "Health Status" (Sick), "Breed" (Criollo), and checkboxes for "petId" and "dogId". The "Search" button is highlighted with a red box. The modal also includes "Save", "Edit", and "Delete" buttons.

The screenshot shows the phpMyAdmin interface for a MySQL database named 'administracionmascotasbd'. The current table is 'tb\_pet'. A modal dialog titled 'Pet Manager' is open, showing a form to edit a pet record. The form fields include: Code (006), Name (Tiny), Born year (2013), Color (Blanco y Negro), Health Status (Healthy), Breed (German shepherd), and Pedigree (checkbox checked). Below the form is a table listing five pet records:

	petId	dogId	color	health_status		
<input type="checkbox"/>	1	001	Firulais	2015	amarillo	Saludable
<input type="checkbox"/>	2	002	Tobby	2018	Negro	Saludable
<input type="checkbox"/>	3	003	Minino	2020	Negro y Blanco	Enfermo
<input type="checkbox"/>	4	005	Pepe	2019	Blacno	Healthy
<input type="checkbox"/>	5	006	Tiny	2013	Blanco y Negro	Healthy

At the bottom of the modal, there are buttons for Save, Search, Edit, and Delete. The main phpMyAdmin interface shows the database structure and other tables like tb\_cat, tb\_dog, and tb\_hospital.

Hemos aprendido cómo realizar la función de búsqueda en una base de datos Mysql con el apoyo de JDBC para implementarlo desde Java. Ahora practiquemos con la solución del reto de la semana para profundizar en esta temática con el material adicional disponible.



Universidad de Caldas