

**Universidad Mariano Gálvez de Guatemala**  
**Facultad de Ingeniería**  
**Ingeniería en Sistemas de Información y Ciencias de la Comunicación.**



**Ing. Johnny Morales.**

**Curso:** Programación II

**Integrantes del Grupo del Proyecto**

Nombre	carné
Juan Pascual Mateo Montejo.	0904-19-8614
Eduardo Andrés Gaspar Andrés.	0904-19-6482
Erick Brandon Díaz Mateo.	0904-14-12755

*Conoceréis la Verdad y la Verdad os Hará Libres.*

## Índice

<b>Introducción .....</b>	<b>1</b>
<b>Proceso de Elaboración del Software .....</b>	<b>2</b>
<b>TECNOLOGÍAS UTILIZADAS .....</b>	<b>9</b>
<b>Netbeans .....</b>	<b>9</b>
<b>XAMPP .....</b>	<b>9</b>
<b>GLIFFY .....</b>	<b>9</b>
<b>Diagrama de clase .....</b>	<b>10</b>
<b>Manuel del Programa .....</b>	<b>11</b>
<b>Conclusiones .....</b>	<b>18</b>

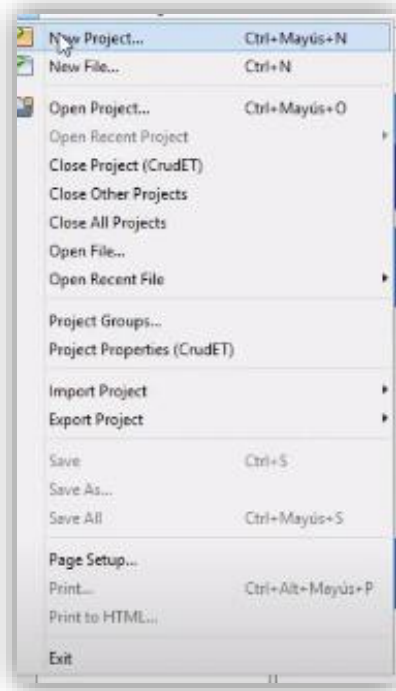
# Introducción

A continuación, la presentación de un programa hecho en el lenguaje de java para uso de una veterinaria, permite anotar los datos de los clientes y de su mascota, permitiendo guardar los registros dentro de una base de datos.

En el programa hizo uso del famoso CRUD, este permite las siguientes características, crear registros, leer registros, actualizar registros, borrar registros.

# Proceso de Elaboración del Software

Hacemos uso de Netbeans, creamos un nuevo proyecto.

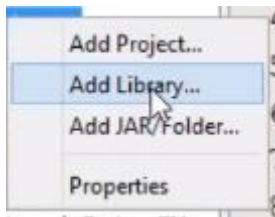


A la clase le dimos el nombre de “conectar” con una variable llamada conection y con un método público “Connection conexion ()”

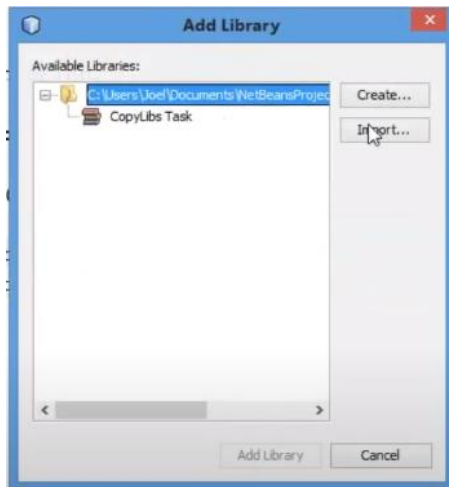
```
public class conectar {  
    Connection connect = null;  
    public Connection conexion()  
}
```

Como también hacemos uso de un try catch, dentro de este hacemos uso de “Class.forName”, donde anotaremos la dirección del Drive de MySQL JDBC Driver

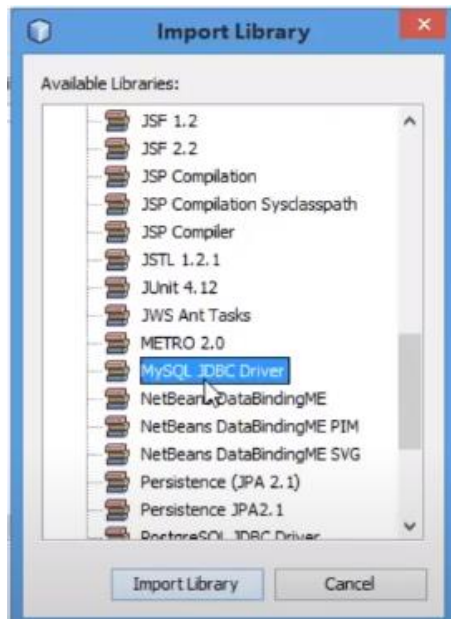
```
try {  
    |  
} catch (Exception e) {  
}
```



Nos dirigiremos a la librería.



Después le damos clic en la opción de import y se desplegará la siguiente gráfica.



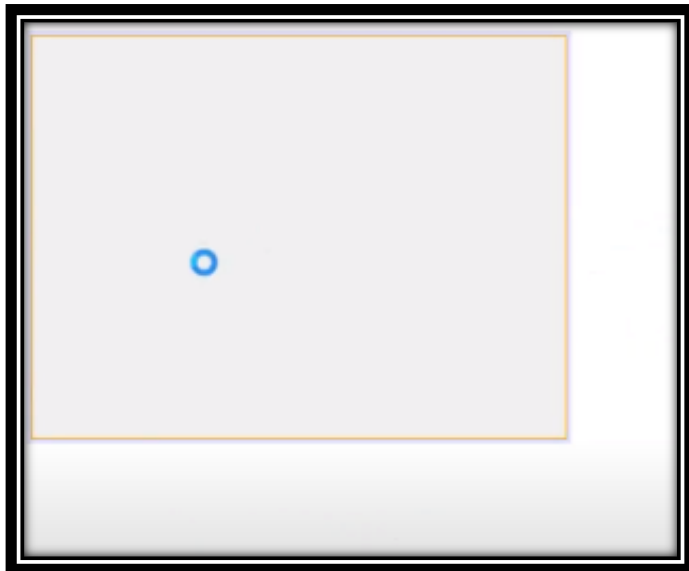
Dentro de todas las opciones buscaremos el Drive de MySQL JDBC Driver

```
Source Refactor Run Debug Profile Team Tools Window Help
<default.config>
Start Page X conectar.java X
1 package clientes;
2
3
4 import java.sql.*;
5 import javax.swing.*;
6
7 /**
8  * @author Charlie
9  */
10 public class conectar {
11     Connection conect = null;
12     public Connection conexion()
13     {
14         try {
15             //Cargamos el Driver MySQL
16             Class.forName("com.mysql.jdbc.Driver");
17             //Cargamos el Driver MySQL
18             conect = DriverManager.getConnection("jdbc:mysql://localhost/mic", "root", "");
19             System.out.println("conexion establecida");
20             //OptionPane.showMessageDialog(null, "Conectado exitosamente");
21             //Cargamos el Driver Access
22             //Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
23             //Conectar en red base
24             //String strConect = "jdbc:odbc:Driver=Microsoft Access Driver (*.mdb);DBQ=C:\\Programas\\Microsoft Access\\Microsoft Access Database Engine\\*.*";
25             //String strConect = "jdbc:odbc:Driver=Microsoft Access Driver (*.mdb);DBQ=C:\\Programas\\Microsoft Access\\Microsoft Access Database Engine\\*.*";
26         } catch (ClassNotFoundException | SQLException e) {
27             System.out.println("Error de conexion");
28             JOptionPane.showMessageDialog(null, "Error de conexion");
29         }
30     }
31 }
```

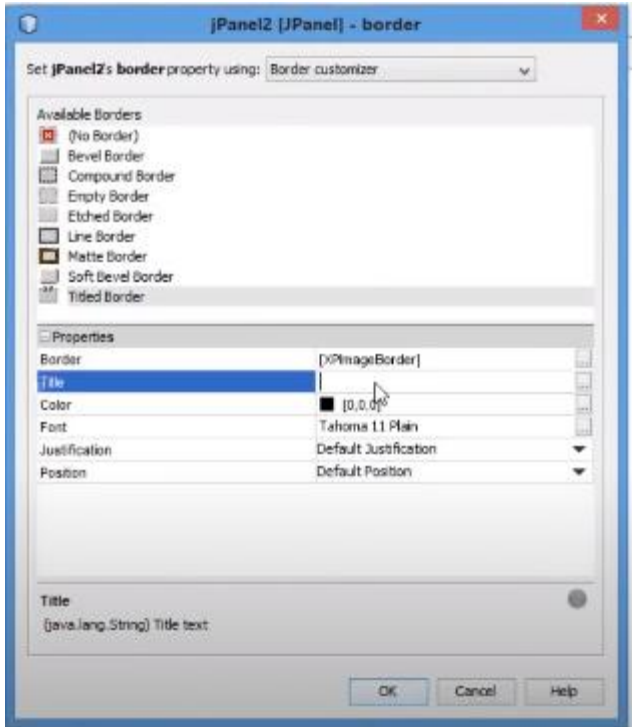
Después de agregar el Drive de MySQL, dentro de Class. forName agregamos la dirección, toda la información dentro de comillas.

Dentro de DriverManager.getConnection agregamos la URL en este va incorporado el puerto y el nombre de la base de datos, de igual forma necesitaremos el usuario el cual es "root" como también podemos agregar la contraseña o simplemente acceso libre.

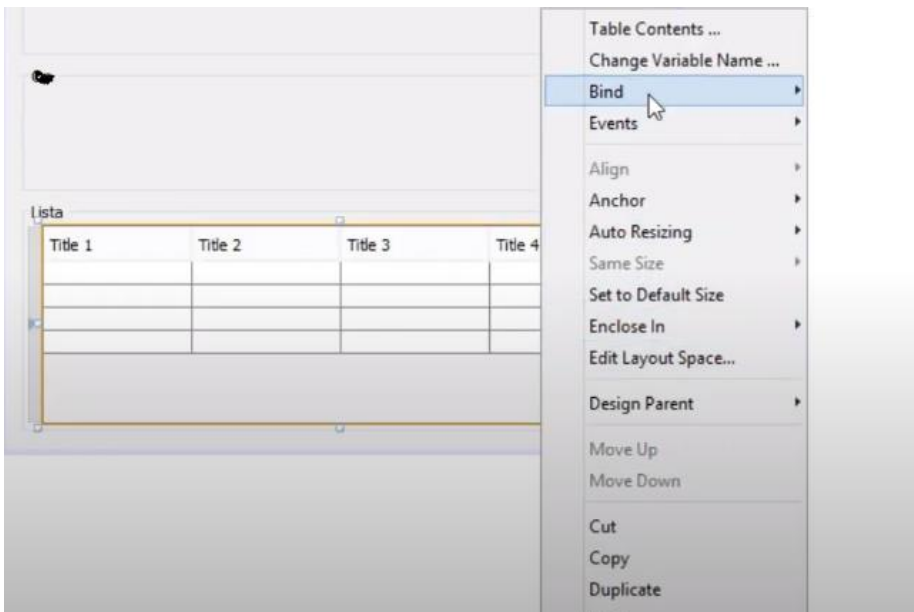
Seguimos rellenando el try catch marcando las excepciones y creando un mensaje para verificar si la conexión ha sido exitosa.



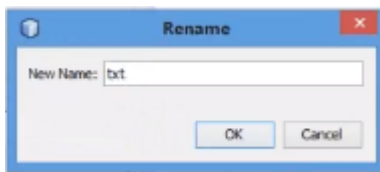
Creamos un formulario, dentro de este creamos lo que vendría siendo los paneles



Este apartado sirve para agregarle el nombre al panel.



agregamos una tabla y la editamos



Cambiamos los nombres de las variables, esto tiene que ver con el tema del orden y identificar las variables.

```

// Statement st:
try {
    Statement st = cn.createStatement();
    ResultSet rs = st.executeQuery(sql);
    //st = cn.createStatement();

    while (rs.next()){
        registros [0] = rs.getString("cod_cli");
        registros [1] = rs.getString("nom_cli");
        registros [2] = rs.getString("direc_cli");
        registros [3] = rs.getString("tel_cli");
        registros [4] = rs.getString("nomMascota");
        registros [5] = rs.getString("Observaciones");// nombre de las tablas en la bd

        model.addRow(registros);
    }
    t_datos.setModel(model);

} catch (SQLException ex) {
    JOptionPane.showMessageDialog(null,ex);
}

```

Después hacemos uso de un while(rs.next())

Dentro de este colocaremos el listado de lo que existe dentro de nuestra tabla.

```

import java.sql.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
public class ing_cli extends javax.swing.JFrame {

```



```

conectar con = new conectar();
Connection cn = con.conexion();

DefaultTableModel model ;
public ing_cli() {
    initComponents();
    limpiar ();
    bloquear();
    cargar();
}
void cargar () {
String [] titulos = {"cod_cli","nom_cli","direc_cli","tel_cli","nomMascota","Observaciones"};

```

Este sirve para los títulos de la tabla.

```
String [] registros = new String [6];/
```

Este vector sirve para cambiar los registros

```
model = new DefaultTableModel (null,titulos);
```

Permite agregar los títulos.

```

conectar cc = new conectar ();//
Connection cn = cc.conexion();

try {
    Statement st = cn.createStatement();
    ResultSet rs = st.executeQuery(sql);
    //st = cn.createStatement();

    while (rs.next()){
        registros [0] = rs.getString("cod_cli");
        registros [1] = rs.getString("nom_cli");
        registros [2] = rs.getString("direc_cli");
        registros [3] = rs.getString("tel_cli");
        registros [4] = rs.getString("nomMascota");
        registros [5] = rs.getString("Observaciones");/

```

Nombre de las tablas de la base de datos.

```

void limpiar () {
//cod_cli.setText("");
nom_cli.setText("");
direc_cli.setText("");
cel_cli.setText("");
nom_mascota.setText("");
observaciones.setText("");
buscar.setText("");

```

Para limpiar las cajas

```

}
void bloquear () {
//cod_cli.setEnabled(false);
nom_cli.setEnabled(false);
direc_cli.setEnabled(false);
cel_cli.setEnabled(false);
nom_mascota.setEnabled(false);
observaciones.setEnabled(false); ///
btn_nuevo.setEnabled(true);
btn_guardar.setEnabled(false);
btn_cancelar.setEnabled(false);
,

```

Sirve para bloquear las cajas

```

/
void des_bloquear () {
//cod_cli.setEnabled(true);
nom_cli.setEnabled(true);
direc_cli.setEnabled(true);
cel_cli.setEnabled(true);
nom_mascota.setEnabled(true);
observaciones.setEnabled(true);
btn_nuevo.setEnabled(false);
btn_guardar.setEnabled(true);
btn_cancelar.setEnabled(true);

// }

```

## **TECNOLOGÍAS UTILIZADAS**

### **Netbeans**

Netbeans IDE es un entorno de desarrollo - una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación.

Es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. Netbeans IDE es un producto libre y gratuito sin restricciones de uso.

Un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento. Sun Microsystems fundó el proyecto de código abierto Netbeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos (Actualmente Sun Microsystems es administrado por Oracle Corporation).

### **XAMPP**

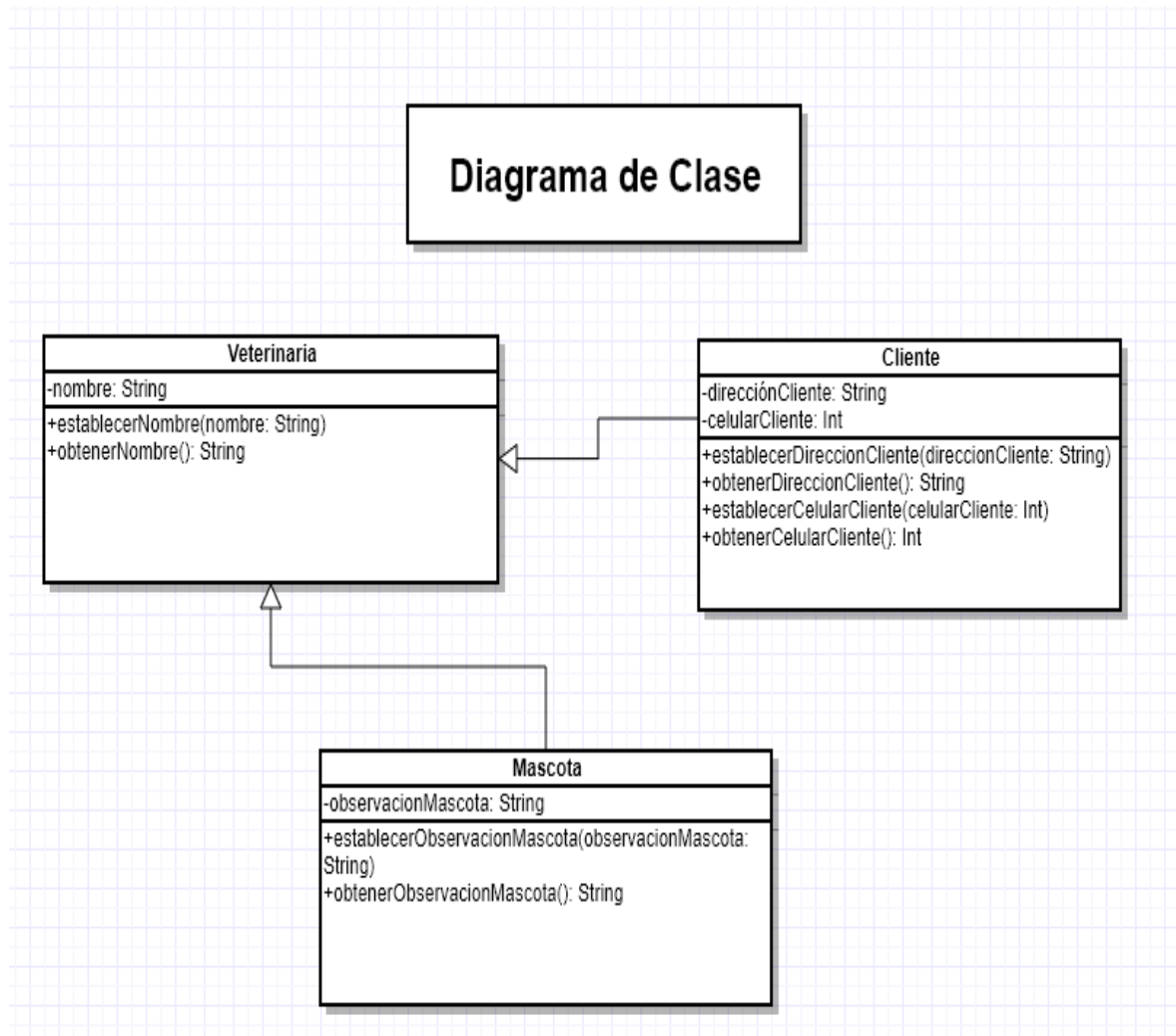
Es un paquete de software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl. El nombre es en realidad un acrónimo: X (para cualquiera de los diferentes sistemas operativos), Apache, MariaDB/MySQL, PHP, Perl. A partir de la versión 5.6.15, XAMPP cambió la base de datos MySQL por MariaDB, un fork de MySQL con licencia GPL.

### **GLIFFY**

Es un sitio web que da las herramientas necesarias para poder diseñar diversos tipos de diagramas en forma de mapas conceptuales con un toque profesional.

Gliffy es una aplicación online para crear, compartir diagramas y colaborar. Funciona a través de un navegador web en la concepción cloud computing, sin la necesidad de instalar el software en el ordenador. Puedes almacenar y realizar copias de tus documentos y crear una variedad de diagramas, dibujos, flujos de proceso e interfaces.

## Diagrama de clase



## Manuel del Programa

### Programas de IDE que necesita tener para abrir y utilizar el programa.

Asegúrese que en su ordenador tenga instalado el IDE de Netbeans 8.2 la última versión, sin en este caso aún no se preocupe observa un enlaces que esta de color azul de clic a ese enlace ya que te llevara a un video donde te explicara paso a paso de como descargar e instalar este IDE.



[https://www.youtube.com/watch?v=BEeYPg\\_WQyl](https://www.youtube.com/watch?v=BEeYPg_WQyl)

También necesitara instalar uno que se llama XAMPP la cual nos permite trabajar con base de datos de Mysql ya que el programa estará conecta a una base de datos la cual permitirá ingresar, borrar, modificar datos.



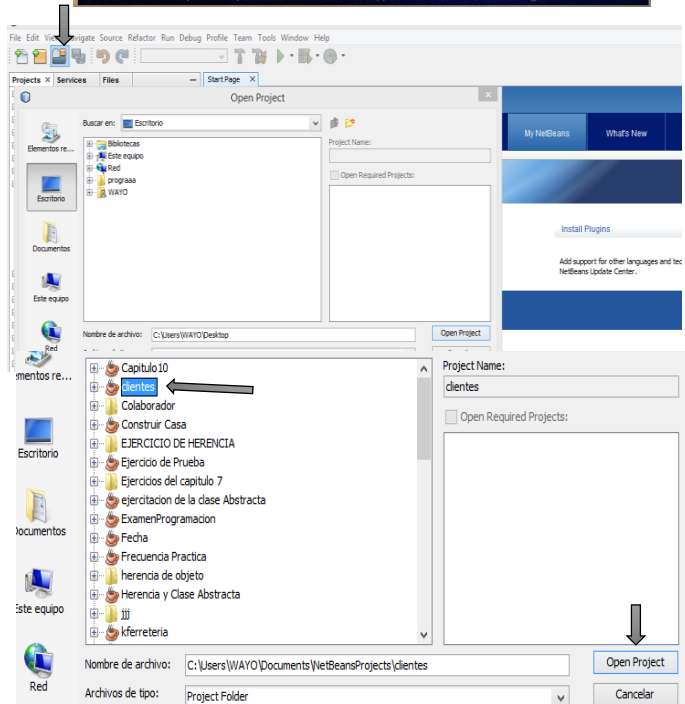
Aquí le dejo el enlace para que pueda descargar e instalar XAMPP.

<https://www.youtube.com/watch?v=nDd2d98Zet8>

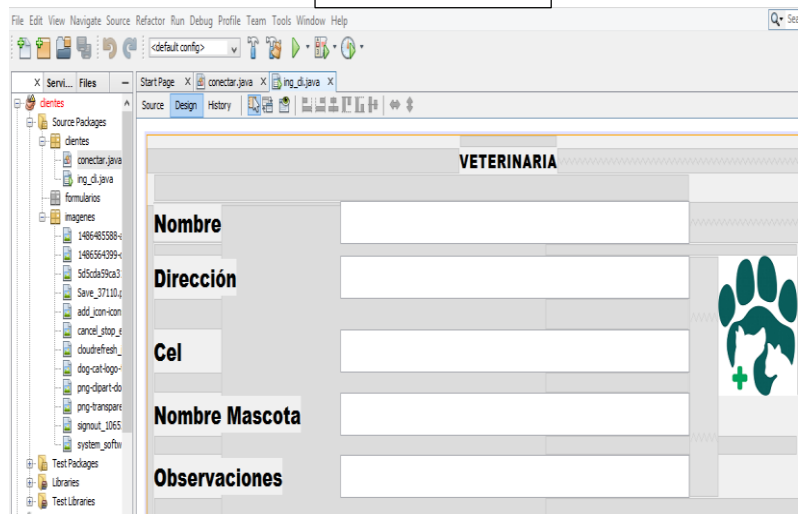
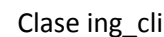
Una vez teniendo los dos IDE mencionados proceda a abrir Netbeans.



Cuando el IDE de Netbeans ya se está ejecutando vamos a darle clic sobre la carpeta azul la que esta señala con una flecha la cual nos abrirá una venta de lo que hay en nuestra máquina, en la cual buscaremos la ubicación en donde esta guardada nuestra carpeta de nuestro programa, una vez encontrada la carpeta del programa la seleccionamos como se muestra en la segunda imagen y le damos en open Project para añadirlo al IDE de Netbeans



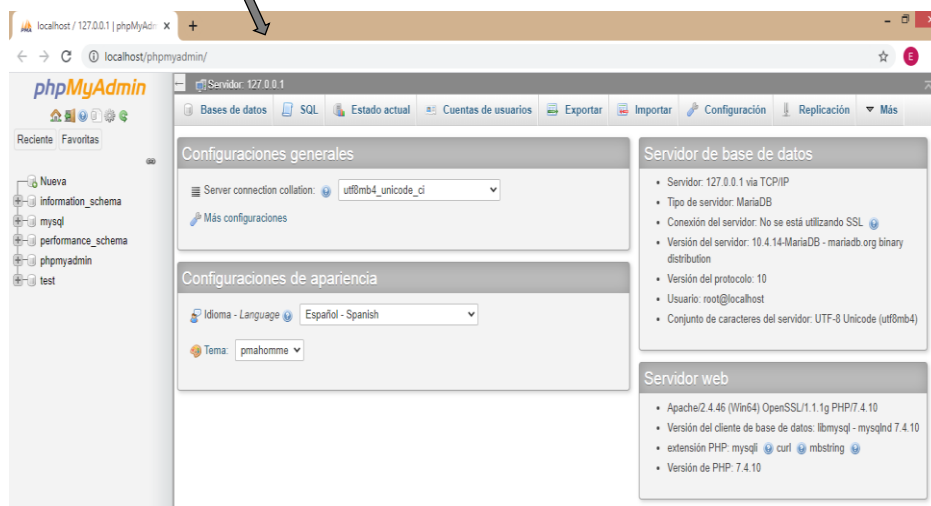
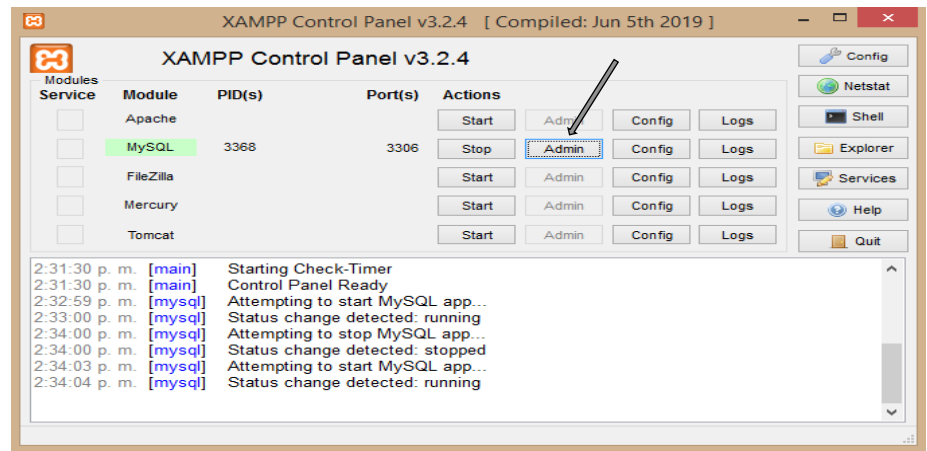
En la clase conector se mostrara el código para podernos conectarnos a la base de datos y la otra es el panel ejecutable en la cual podemos modificar la información con los botones que queremos.



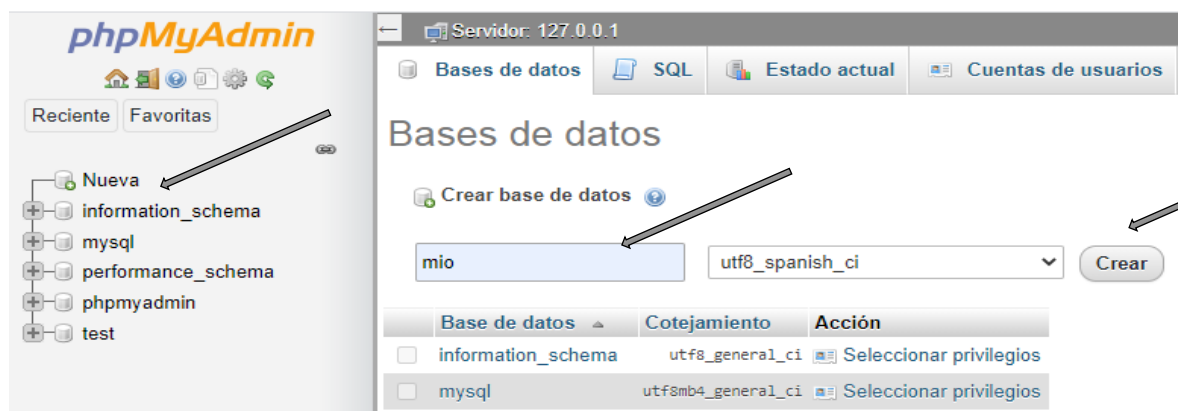
En el panel de Xampp daremos clic sobre el módulo de Apache y MySql para que la página local de xampp funcione de lo contrario no funcionara.



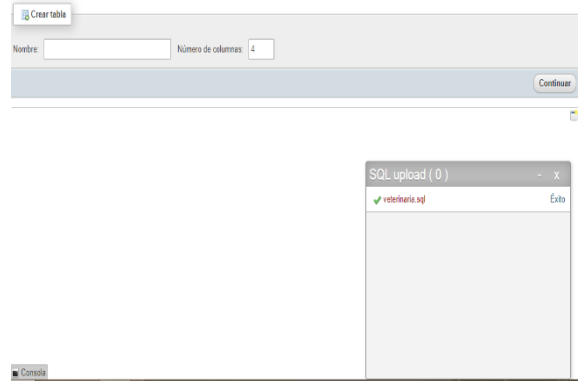
Luego de activar los módulos antes mencionado ahora le damos clic en el apartado de MySQL en donde dice admin para entrar a la página de Xampp de base de datos.



En la cual crearemos una nueva base de datos que se llamara mio o con el nombre que usted quiera ponerle dar clic en crear para que se guarde los cambios y se cree la base de datos mio.

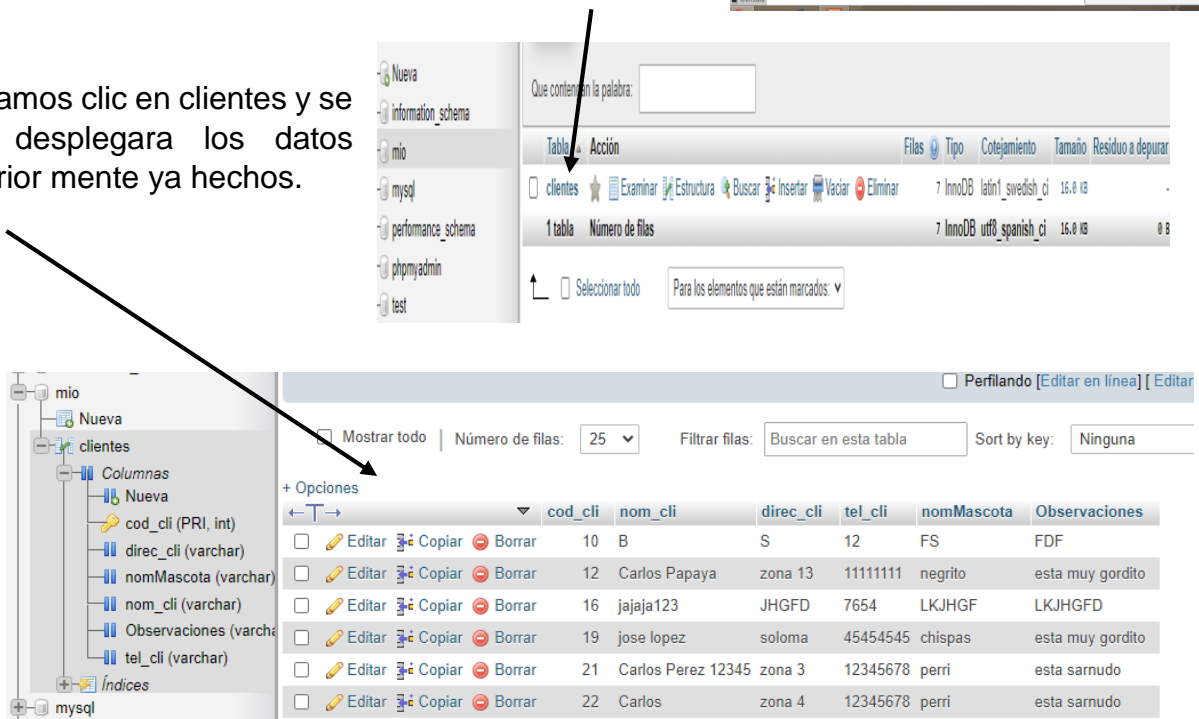


Una vez creado tu base de datos arrastra la carpeta veterinaria en ella y te saldrá una ventanita que dirá veterinaria añadida.



Una vez añadida la carpeta le damos clic en la base de datos mio en la cual ya nos mostrara que la carpeta de los datos cliente ya está añadido.

Le damos clic en clientes y se nos desplegara los datos anterior mente ya hechos.




Nos vamos al código de Netbeans de la clase conectar y asegurarse de que el nombre de la base de datos sea la misma si no, debemos de cambiarla y ponerla como esta en la base de datos asi nos podremos conectar sin ningún problema al momento de ejecutarlo.

```

15 null;
16 n conexion()
17
18 try {
19     //Cargamos el Driver MySQL
20     Class.forName("com.mysql.jdbc.Driver");
21     //Class.forName("org.gjt.mm.mysql.Driver");
22     connect = DriverManager.getConnection("jdbc:mysql://localhost/mio","root","");
23     System.out.println("conexion establecida");
24     //JOptionPane.showMessageDialog(null,"Conectado exitosamente");
25     //Cargamos el Driver Access
26     //Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

```



Para ejecutar el programa le damos clic en  y saldrá una ventanita diciendo que ya estamos conectados y se mostrara el menú principal del programa.

```

16: Class.forName("com.mysql.jdbc.Driver");
17: //Class.forName("org.gjt.mm.mysql.Driver");
18: connect = DriverManager.getConnection("jdbc:mysql://localhost/mio","root","");
19: System.out.println("conexion establecida");
20: //JOptionPane.showMessageDialog(null,"Conectado exitosamente");
21: //Cargamos el Driver Access
22: //Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
23: //Conectar en red base
24: //String strConect = "jdbc:odbc:Driver=Microsoft Access Driver (*.mdb);DBQ=//servi
25: //Conectar Localmente

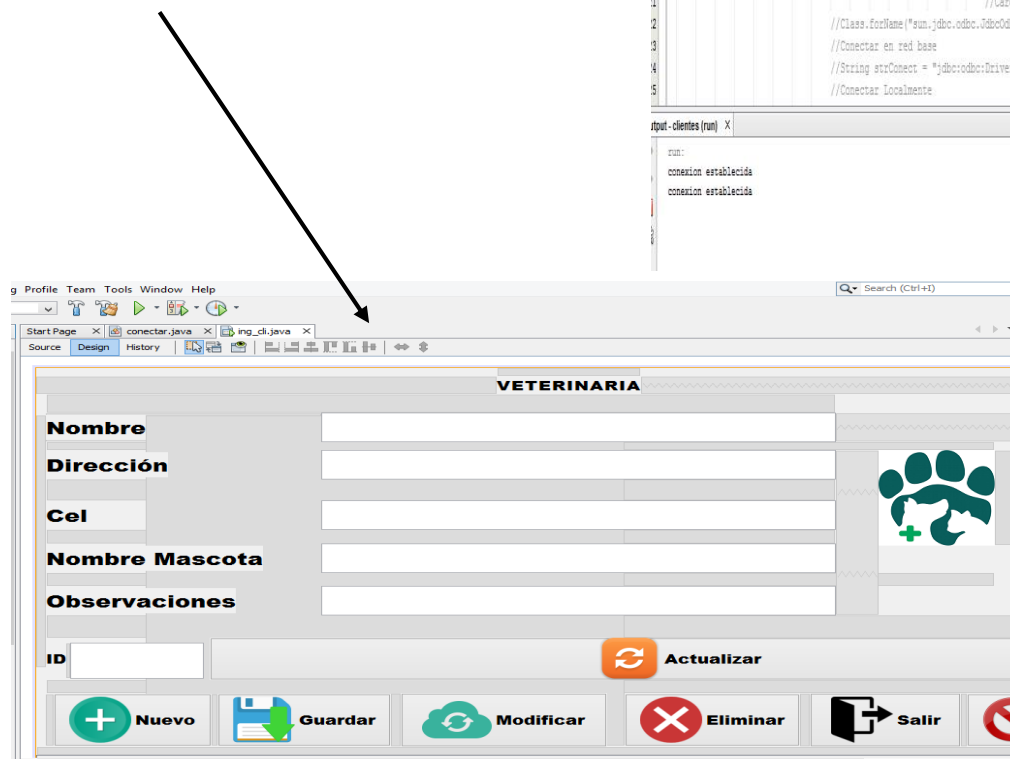
```

Output-clients (run) X

```

run:
conexion establecida
conexion establecida

```



Aremos algunas pruebas la primera será modificar un dato donde dice Id colocaremos el número del dato que queremos modificar y le damos en modificar y aparecerá en pantalla luego le damos en actualizar y se guardara automáticamente con los datos modificados.

**VETERINARIA**


**Nombre** Eduardo Pablo






**Dirección** San Juan Ixcoy

**Cel** 57348541

**Nombre Mascota** Manchas

**Observaciones** Tiene Pulgas

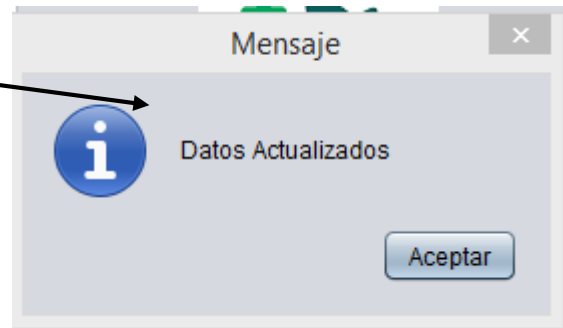
ID 10  Actualizar

 Nuevo  Guardar  Modificar  Eliminar  Salir  Cancelar

cod_cli	nom_cli	direc_cli	tel_cli	nomMascota	Observaciones
10	B	S	12	ES	EDF
12	Juan Pedro	San Pedro Soloma	48852514	Escubi	Rabia
16	jajaja123	JHGFD	7654	LKJHGF	LKJHGF
19	jose lopez	soloma	45454545	chispas	esta muy gordito
21	Carlos Perez 12345	zona 3	12345678	perri	esta sarnudo

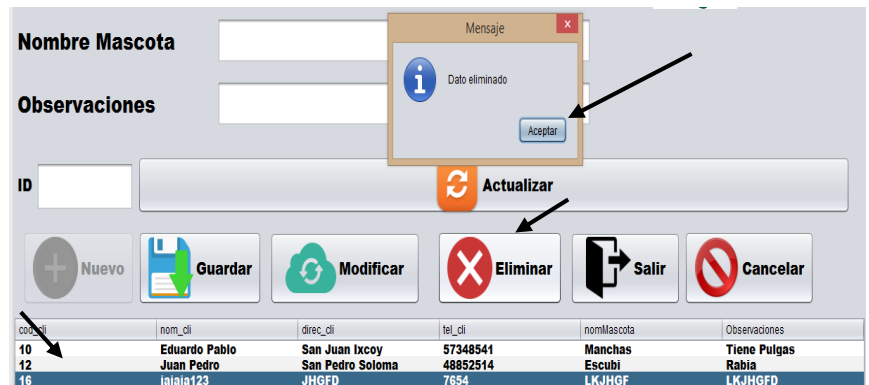
Al momento de dar en actualizar saldrá esta ventanita donde dirá datos actualizados.

Y como vemos en la parte inferior se añadió el nuevo dato modificado.



cod_cli	nom_cli	direc_cli	tel_cli	nomMascota	Observaciones
10	Eduardo Pablo	San Juan Ixcay	57348541	Manchas	Tiene Pulgas
12	Juan Pedro	San Pedro Soloma	48852514	Escubi	Rabia

El segunda prueba eliminaremos un dato, seleccionamos el dato que queremos eliminar una vez seleccionada le damos en el botón de eliminar y saldrá una ventanita diciendo que el dato fue eliminado le damos en aceptar y como vemos en la tabla de abajo ya no se encuentra.



Una tercera prueba, crearemos un nuevo dato, damos clic en el botón nuevo, luego rellene cada casilla respecto a lo que pida, le damos clic en el botón guardar y nos saldrá una ventanita que nos indicara que el registro a si guardado con éxito le damos en aceptar inmediatamente el dato ya está en la base de datos como se puede ver en la tabla de abajo. Para cerrar el programa le damos en el botón de salir y se cerrara.

**VETERINARIA**

Nombre

Dirección

Cel

Nombre Mascota

Observaciones

ID



 Actualizar

 Nuevo  Guardar  Modificar  Eliminar  Salir  Cancelar

**Mensaje**

 Registro Guardado con éxito



 **Actualizar**

cod_cli	nom_cli	direc_cli	tel_cli	nomMascota	Observaciones
10	Eduardo Pablo	San Juan Ixcoy	57348541	Manchas	Tiene Pulgas
12	Juan Pedro	San Pedro Soloma	48852514	Escubi	Rabia
22	Carlos	zona 4	12345678	perri	esta sarnudo
23	francis	ixtenam	12341234	rex	esta sarnudo
24	Micaela Paiz	Huehuetenango	32383064	Pelusa	Rabia

## **Conclusiones**

- El programa nos permite guardar nuevos registros y ordenarlos de manera ordenada conforme se va añadiendo nuevos en la base de datos de tal manera permitiendo buscar de una manera rápida un registro existente solo debemos añadir el número de ID del registro y listo.
- Al momento de guardar o modificar un registro inmediatamente se muestra en pantalla de la parte superior del menú de control.
- Se programó este programa para que una veterinaria pueda saber cuántos clientes junto con su mascota llegan a ellos, y poder ordenar sus registros conforme van llegando y que problema tiene su mascota.