PAC 2. UF6.

Diseño de programas con lenguajes de programación orientados a objetos para la gestión de bases de datos.

Aníbal Santos Gómez

En primer lugar para realizar esta práctica con bases de datos y Java en SQL , necesitaremos crear los prefijos de conexión a través de los siguientes métodos en una nueva clase que denominaremos BBDDVetIlerna:

```
package grafico;

import java.sql.Connection;

public class BBDDVetIlerna
{
    private String host;
    private String bbdd;
    private String usuario;
    private String contrasena;
    private Connection conexion;

    public BBDDVetIlerna(String host, String puerto, String bbdd, String usuario, String contrasena) // prefijo de conexion
    {
        this.host=host;
        this.bbdd=bbdd;
        this.usuario=usuario;
        this.contrasena=contrasena;

        //prefijo de conexion

        String prefijo="jdbc:mysql://";

        //url de conexion

        String url="";
        url= prefijo + host + ":" + puerto + "/" + bbdd;
```

Esto es lo que posteriormente estableceremos en VentanaDefinitiva (que es nuestro main):

```
public VentanaDefinitiva(ArrayList<Cliente> clientes) {

    this.clientes = clientes;

    //UF6-PAC2 conexion objeto bbdd

    bbddMySQL = new BBDDVetIlerna("mysql.hostinger.es", "3306", "u508027557_ansan", "u508027557_asg", "721989");


    frame = new JFrame("Clínica veterinaria");
    frame.setSize(900, 700);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

Continuando con nuestra clase BBDDVetIlerna, registraremos el Driver de conexión JDBC a MysQL, estableciendo una confirmación automática de los cambios en la base de datos:

```
//registramos el driver de conexion JDBC a MySQL

try {
    DriverManager.registerDriver(new com.mysql.jdbc.Driver());

    conexion = DriverManager.getConnection(url, usuario, contrasena);

    //establecemos confimarcion automatica de los cambios en la base de datos

    conexion.setAutoCommit(true);

} catch (SQLException e) {

    e.printStackTrace();
}
```

Posteriormente llevaremos a cabo la creación de una serie de métodos que nos permitirán llevar a cabo las siguientes tareas:

1. Leer datos:

```
public ArrayList<Cliente> leerDatos()
{
    ArrayList<Cliente> resultado =null;

    String sqlClientes ="SELECT dni, nombre FROM CLIENTE";
    String sqlMascotas="SELECT codigo, nombre, genero, raza, color, tipo, categoria FROM Mascota WHERE dni_cliente like ?";
    String sqlTratamientos="SELECT fecha, tratamiento FROM Tratamiento WHERE codigo_mascota = ?";
```

Preparamos las consultas y dentro de Clientes las de mascotas y posteriormente las de tratamiento:

```
//preparamos las consultas

try(Statement stClientes= conexion.createStatement();
    PreparedStatement psMascotas= conexion.prepareStatement(sqlMascotas);
    PreparedStatement psTratamientos= conexion.prepareStatement(sqlTratamientos);)
```

```
ResultSet rsClientes = stClientes.executeQuery(sqlClientes);

    while(rsClientes.next())
    {
        Cliente cli= new Cliente();
        cli.setDni(rsClientes.getString(1));
        cli.setNombre(rsClientes.getString(2));

        //preparamos la consulta de mascotas

        psMascotas.setString(1,cli.getDni());

        //ejecutamos la consulta de mascotas una vez preparada

        ResultSet rsMascotas = psMascotas.executeQuery();

        while( rsMascotas.next() ) {
            Mascota mascota=null;
            if( rsMascotas.getString(7).equalsIgnoreCase("perro" )) {
                mascota = new Perro();
                ((Perro)mascota).setRaza(rsMascotas.getString(4));
                mascota.setCodigo(rsMascotas.getInt(1));
                mascota.setNombre(rsMascotas.getString(2));
                mascota.setGenero(rsMascotas.getString(3));
```

```
        //añadimos todos los tratamientos de la mascota
        mascota.addTratamiento(this.getTratamientos(mascota.getCodigo()));
    }
    else if( rsMascotas.getString(7).equalsIgnoreCase("gato" )) {
        mascota = new Gato();
        ((Gato)mascota).setColor(rsMascotas.getString(5));
        mascota.setCodigo(rsMascotas.getInt(1));
        mascota.setNombre(rsMascotas.getString(2));
        mascota.setGenero(rsMascotas.getString(3));

        //añadimos todos los tratamientos de la mascota
        mascota.addTratamiento(this.getTratamientos(mascota.getCodigo()));
    }
    else if( rsMascotas.getString(7).equalsIgnoreCase("roedor")) {
        mascota = new Roedor();
        ((Roedor)mascota).setTipo(rsMascotas.getString(6));
        mascota.setCodigo(rsMascotas.getInt(1));
        mascota.setNombre(rsMascotas.getString(2));
        mascota.setGenero(rsMascotas.getString(3));

        //añadimos todos los tratamientos de la mascota
        mascota.addTratamiento(this.getTratamientos(mascota.getCodigo()));
    }
```

```
    catch (SQLException e)
    {
        e.printStackTrace();
    }

    finally {return resultado;}
```

2. Tratamientos: para ello declaramos un map de tipo HashMap, creamos una consulta con un parámetro, un try con recursos y vamos preparando el parámetro, si existen errores de consultas saltaría un catch.

```
private Map<String,String> getTratamientos(int codMascota)
{
    //Declaramos el mapa de tipo HashMap
    Map<String, String>resultado= new HashMap<>();

    //Creamos la consulta con un parametro
    String sqlTratamientos="SELECT fecha, tratamiento FROM Tratamiento WHERE codigo_mascota = ?";

    //Creamos un try con recursos, en este caso con el preparedStatement de Tratamientos que obten
    try(PreparedStatement psTratamientos = conexion.prepareStatement(sqlTratamientos))
    {
        //preparamos el parámetro recibido para completar la consulta
        psTratamientos.setInt(1, codMascota);

        //ejecutamos la consulta
        ResultSet rsTratamientos = psTratamientos.executeQuery();

        //mientras la consulta tenga resultados
        while (rsTratamientos.next()) {

            //incorporamos una tupla clave-valor al mapa con fecha y tramiento correspondiente
            resultado.put(rsTratamientos.getString(1), rsTratamientos.getString(2));

        }
    }
```

```
    //si ha error de ejecucion de consultas, slata el catch
    catch (SQLException e)
    {
        e.printStackTrace();
    }
    //en cualquier caso (haya error o vaya todo bien devolveremos
    finally {return resultado;}
}
```

3. Escritura de nuevo cliente, nueva mascota, nuevo tratamiento: aquí iremos metiendo las conocidas consultas de INSERT INTO (SQL).

```
public void nuevoClienteBBDD (Cliente cliente)
{
    String consulta = "INSERT INTO CLIENTE dni, nombre VALUES ?, ?";

    try (PreparedStatement ps = conexion.prepareStatement(consulta))
    {
        ps.setString(1, cliente.getDni().toUpperCase());
        ps.setString(2, cliente.getNombre());

        if(ps.executeUpdate() ==0 )
        {
            //System.out.println("Error en la inserción cliente");
            throw new Exception("Error en la inserción cliente");
        }

        System.out.println("El cliente ha sido insertado correctamente");

    }

    catch (Exception e)
    {
        System.err.println(e.getMessage());
    }

}
```

En nueva mascota tendremos que verificar que pertenece a un cliente.

```
public void nuevaMascotaBBDD(Cliente cliente, Mascota mascota) //conocemos la información del cliente
{

    if(buscoClienteBBDD(cliente.getDni()) != null)
    {
        if(buscoMascotaBBDD(mascota.getCodigo()) == null )
        {
            String consulta = "INSERT INTO Mascota codigo, nombre, genero, raza, color, edad, categoria, dni_cliente VALUES ?, ?, ?, ?, ?, ?, ?, ?";
```

```
try (PreparedStatement ps = conexion.prepareStatement(consulta))
{
    ps.setInt(1, mascota.getCodigo());
    ps.setString(2, mascota.getNombre());
    ps.setString(3, mascota.getGenero());

    /*
     * perro raza
     * gato color
     * roedor tipo
     *
     */

    if (mascota instanceof Perro)
    {
        ps.setString(4,((Perro) mascota).getRaza());
        ps.setString(5, "");
        ps.setString(6, "");
        ps.setString(7, "PERRO");
    }
    else if(mascota instanceof Gato)
    {
        ps.setString(4, "");
        ps.setString(5,((Gato) mascota).getColor());
        ps.setString(6, "");
        ps.setString(7, "GATO");

    }
    else if(mascota instanceof Roedor)
    {
        ps.setString(4, "");
        ps.setString(5, "");
        ps.setString(6,((Roedor) mascota).getTipo());
        ps.setString(7, "ROEDOR");
    }
    else
    {
        throw new Exception ("La BBDD no permite este tipo de mascotas");
    }
```

```
                ps.setString(8, cliente.getDni().toUpperCase());

                if(ps.executeUpdate() == 0)
                {
                    throw new Exception ("ERROR durante la insercion de la mascota");
                }

                else
                {
                    System.err.println("MAScota insertada correctamente");
                }
            }

            catch (Exception e)
            {
                System.err.println(e.getMessage());
            }
        }

    }else
    {
        System.err.println("El cliente no existe en la BBDD");
    }
```

Al igual que en mascota comprobamos la pertenencia a un cliente, en la creación de un nuevo tratamiento comprobaremos la pertenencia a una mascota.

```java
public void nuevoTratamiento (Mascota mascota, String fecha, String tratamiento)
{
    if(buscaMascotaBBDD(mascota.getCodigo()) != null)
    {
        String consulta = "INSERT INTO fecha, tratamiento codigo_mascota VALUES ?, ?, ? ";

        try (PreparedStatement ps= conexion.prepareStatement(consulta))
        {
            ps.setString(1, fecha);
            ps.setString(2, tratamiento);
            ps.setInt(3, mascota.getCodigo());

            if(ps.executeUpdate() == 0)
            {
                throw new Exception("Error durante la insercion del tratamiento");
            }
            else
            {
                System.out.println("Tratamiento insertado correctamente");
            }
        }

        catch (Exception e)
        {
            System.err.println(e.getMessage());
        }
    }
    else
    {
        System.err.println("La mascota no exite en la BBDD");
    }
}
```

4. Búsqueda de cliente, búsqueda de mascota:

```
private Cliente buscaClienteBBDD (String dni)
{
    Cliente resultado = null;

    String consulta ="SELECT dni, nombre FROM Cliente WHERE dni like ?";

    try (PreparedStatement ps = conexion.prepareStatement(consulta))
    {
        ps.setString(1, dni.toUpperCase());

        ResultSet rs = ps.executeQuery();

        if(rs.next())
        {
            resultado=new Cliente();
            resultado.setDni(rs.getString(1));
            resultado.setNombre(rs.getString(2));
        }

    }

    catch (Exception e)
    {
        System.err.println(e.getMessage());
    }
    finally
    {
.....
.....return resultado;
.....
.....}
```

```
private Mascota buscaMascotaBBDD (int codigo)
{
    Mascota resultado = null;

    String consulta ="SELECT codigo, nombre, genero, raza, color, tipo, categoria, dni_cliente FROM Mascota WHERE codigo like ?";
```

```
try (PreparedStatement ps = conexion.prepareStatement(consulta))
{
    ps.setInt(1, codigo);

    ResultSet rs = ps.executeQuery();

    if(rs.next())
    {
        if( rs.getString(7).equalsIgnoreCase("perro" )) {
            resultado = new Perro();
            ((Perro)resultado).setRaza(rs.getString(4));
            resultado.setCodigo(rs.getInt(1));
            resultado.setNombre(rs.getString(2));
            resultado.setGenero(rs.getString(3));


        }
        else if( rs.getString(7).equalsIgnoreCase("gato" )) {
            resultado = new Gato();
            ((Gato)resultado).setColor(rs.getString(5));
            resultado.setCodigo(rs.getInt(1));
            resultado.setNombre(rs.getString(2));
            resultado.setGenero(rs.getString(3));


        }
        else if( rs.getString(7).equalsIgnoreCase("roedor")) {
            resultado = new Roedor();
            ((Roedor)resultado).setTipo(rs.getString(6));
            resultado.setCodigo(rs.getInt(1));
            resultado.setNombre(rs.getString(2));
            resultado.setGenero(rs.getString(3));


        }
    }

}
```

```
  catch (Exception e)
  {
      System.err.println(e.getMessage());
  }
  finally
  {

    return resultado;

  }
```

Realizado esto nos vamos a nuestra clase VentanaDefinitiva, donde como hemos visto antes estableceremos los parámetros de conexión a nuestra base de datos:

```
bbddMySQL = new BBDDVetIlerna("mysql.hostinger.es", "3306", "u508027557_ansan", "u508027557_asg", "721989");


frame = new JFrame("Clinica veterinaria");
frame.setSize(900, 700);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

Ahora tendremos que ejecutar el añadir el cliente, en la base de datos, lo haremos dentro del punto anterior referente al DNI:

```
// añadimos el cliente a la "base de datos"
clientes.add(cliente);

/////PAC UF 6 -- 02 !!

/* Guradamos el cliente en la bddd mysql
 *
 *
 *
 */

bbddMySQL.nuevoClienteBBDD(cliente); /// UF6 PAC 2

/*
 *
 *
 * Fin de BBDD MySQL
 */


System.out.println("Datos cliente: " + cliente);
```

```
if (validaDNI(dni) == true) {

    for (Cliente cli: clientes) {
        if (cli.getDni().equalsIgnoreCase(dni)) {
            for (Mascota m : cli.getMascotas()) {
                String nombreMascota = m.getNombre();
                //trat_mascotas.add(nombreMascota);
                String fecha = trat_fechaText.getText();
                String tratamiento = trat_tratamientoText.getText();
                m.addTratamiento(fecha, tratamiento);

                /* Guradamos el cliente en la bddd mysql
                 *
                 *
                 *
                 */

                bbddMySQL.nuevoTratamiento(m, fecha, tratamiento); //// UF 6 PAC 2

                /*
                 *
                 *
                 * Fin de BBDD MySQL
                 */

                System.out.println(m.getNombre() + " :: fecha: " + fecha + " - tratamiento: " + tratamiento);
            }
            trat_mascotasComboBox.setEnabled(true);
            break;
        }
```

```
}
String nombre = nue_nombreMascotaText.getText();
mascota.setNombre(nombre);
//mascota.setNombre(nue_nombreMascotaText.getText());
mascota.setGenero(nue_machoRB.isSelected()?"macho":"hembra");


for (Cliente cli : clientes) {
    if( cli.getDni().equalsIgnoreCase(dni)) {
        cli.addMascota(mascota);

        /* Guradamos el cliente en la bddd mysql
         *
         *
         *
         */

        bbddMySQL.nuevaMascotaBBDD(cli, mascota); ////// UF6 PAC 2

        /*
         *
         *
         * Fin de BBDD MySQL
         */

        System.out.println("NUEVA MASCOTA:: Datos cliente: " + cli);
        break;
        }
```