

Documentación

La idea principal para la ficha médica era crear una interfaz con una serie de secciones, en la principal habría una tabla donde aparecen todas las fichas creadas y 3 botones (Agregar ficha, Modificar y Eliminar), al pulsar en agregar ficha, la interfaz cambia. Se puede introducir la fecha, el veterinario y la mascota (ambos con una lista desplegable), y 3 apartados con textareas para los síntomas, medicación y tratamiento. También hay 2 botones para guardar y volver a la interfaz anterior. En el caso de modificar, la idea era poder seleccionar una ficha de la tabla y modificarla, y para eliminar, lo mismo.

Se comenzó haciendo una plantilla base bastante básica e introduciendo características como la conexión con la base de datos para ello es necesario descargar **mysql-connector-j-9.3.0.jar** e importar lo siguiente

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
```

Atributos + getters y setters

```
public class Fichamedica extends javax.swing.JFrame {
    private Date date;
    private String tratamientos;
    private String medication;
    private String symptom;
    private int id_vet;
    private int id_mascota;

    public Fichamedica(Date date, String tratamientos, String medication, String symptom, int id_vet, int id_mascota){
        this.date=date;
        this.tratamientos=tratamientos;
        this.medication=medication;
        this.symptom=symptom;
        this.id_vet=id_vet;
        this.id_mascota=id_mascota;
    }

    public Date getDate() {
        return date;
    }

    public void setDate(Date date) {
        this.date = date;
    }

    public String getTratamientos() {
        return tratamientos;
    }

    public void setTratamientos(String tratamientos) {
        this.tratamientos = tratamientos;
    }

    public String getMedication() {
        return medication;
    }

    public void setMedication(String medication) {
        this.medication = medication;
    }
}
```

a la hora de realizar conexiones en los diferentes métodos hace falta introducir lo siguiente:

```

public void cargarFichas() {
    String url = "jdbc:mysql://localhost:3306/clinica_veterinaria";
    String usuario = "root";
    String contraseña = "";

    String sql = "SELECT ID_ficha, fecha, ID_mascota, ID_veterinario FROM fichamedica";

    try (Connection conn = DriverManager.getConnection(url, usuario, contraseña); Statement stmt = conn.createStatement(); Re

```

(la línea que no sale entera es la siguiente: **try (Connection conn = DriverManager.getConnection(url, usuario, contraseña); Statement stmt = conn.createStatement(); ResultSet rs = stmt.executeQuery(sql)) {**
)

En el apartado principal se crea una un modelo de tabla default con las columnas que queremos y se asigna al jTable creado en la interfaz:

```

DefaultTableModel tabla = new DefaultTableModel();
tabla.addColumn("ID Ficha");
tabla.addColumn("Fecha");
tabla.addColumn("Mascota");
tabla.addColumn("Veterinario");

tablaDatos.setModel(tabla); // Asignar el Jtable a la tabla

```

para cargar los datos de la base de datos en la tabla, añadimos filas de la siguiente manera:

```

while (rs.next()) {

    tabla.addRow(new Object[]{
        rs.getInt("ID_ficha"),
        rs.getDate("fecha"),
        rs.getInt("ID_mascota"),
        rs.getInt("ID_veterinario") // Mostrar el veterinario asociado
    });
}

```

Alex ha creado un método para limpiar los campos de la interfaz a la hora de crear nuevas fichas con el siguiente código:

```

public void limpiarCampos(JPanel panel) {
    for (Component comp : panel.getComponents()) {
        if (comp instanceof JTextField) {
            ((JTextField) comp).setText("");
        } else if (comp instanceof JScrollPane) {
            Component contenido = ((JScrollPane) comp).getViewport().getView();
            if (contenido instanceof JTextArea) {
                ((JTextArea) contenido).setText("");
            }
        } else if (comp instanceof JPanel) {
            limpiarCampos((JPanel) comp);
        }
    }
}

```

el cual recorre los componentes, tanto los **JTextField**, como los **JScrollPane** y los **JTextArea** para limpiar su contenido y que no haya información insertada anteriormente.

Para poder seleccionar a los veterinarios/mascotas disponibles, se hace uso del siguiente método (uno para veterinarios y otro para mascotas)

```
private void cargarVeterinarios() {
    String url = "jdbc:mysql://localhost:3306/clinica_veterinaria";
    String usuario = "root";
    String contraseña = "";

    String sql = "SELECT ID_VET, NumeroColegiado FROM veterinario";

    try (Connection conn = DriverManager.getConnection(url, usuario, contraseña); Statement stmt = conn.createStatement(); ResultSet rs = stmt.executeQuery(sql)) {

        comboVeterinarios.removeAllItems();

        while (rs.next()) {
            String item = (rs.getInt("ID_VET") + " - " + rs.getString("NumeroColegiado")); // Agregar ID y nombre
            comboVeterinarios.addItem(item);
        }

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

el cual hace una conexión con la base de datos y mediante un select pide el id y el número de colegiado/nombre de la mascota. Una vez realizada la conexión, quita todos los items que puede haber en los **comboBox** (componente desplegable con diferentes items/datos) y hace un bucle en el que va pasando de fila en fila de la tabla en busca de la información, la cual guarda en un string que coge la id y el numero/nombre para guardarlos en el mismo espacio

```
while (rs.next()) {
    String item = (rs.getInt("ID_VET") + " - " + rs.getString("NumeroColegiado")); // Agregar ID y nombre
    comboVeterinarios.addItem(item);
}
```

y lo añade al hueco(item) del comboBox.

Para la hora de modificar una ficha añadimos que si el id de la ficha seleccionada es igual al id del veterinario, recoja del **comboBox** el item con dicho id

```
if (rs.getInt("ID_VET") == idSeleccionado) {
    veterinarioSeleccionado = item;
}

if (veterinarioSeleccionado != null) {
    comboVeterinarios.setSelectedItem(veterinarioSeleccionado);
}
```

Para guardar datos a la hora de generar fichas se ha creado un método personalizado llamado **guardarDatos()** el cual empieza recogiendo en variables la información introducida en los campos de texto, en la fecha y los comboBox

```
public void guardarDatos() {
    String sintoma = sintomas.getText();
    String medica = medicacion.getText();
    Date fechaFicha = fecha.getDate();
    java.sql.Date fechaSQL = new java.sql.Date(fechaFicha.getTime());
    String trata = tratamiento.getText();
    String seleccionadol = (String) comboVeterinarios.getSelectedItem();
    int idV = Integer.parseInt(seleccionadol.split(" - ")[0]); // Extraer el ID
    String seleccionado2 = (String) comboMascotas.getSelectedItem();
    int idM = Integer.parseInt(seleccionado2.split(" - ")[0]); // Extraer el ID
```

para los comboBox, como anteriormente los guardaba como **Id + " - " + nombre**, le pedimos que seleccione lo que se encuentra en el primer espacio antes del separador " - "

Una vez que se han recogido los datos, se hace conexión con la base de datos y realizamos la consulta correspondiente

```
String sql = "INSERT INTO fichamedica (fecha, sintomas, tratamiento, medicacion, ID_mascota, ID_veterinario) VALUES (?, ?, ?, ?, ?, ?)";

try (Connection conn = DriverManager.getConnection(url, usuario, contraseña); PreparedStatement stmt = conn.prepareStatement(sql)) {

    stmt.setDate(1, fechaSQL);
    stmt.setString(2, sintoma);
    stmt.setString(3, trata);
    stmt.setString(4, medica);
    stmt.setInt(5, idM);
    stmt.setInt(6, idV);
    stmt.executeUpdate();
}
```

en este caso es un insert into, donde las variables que vamos a introducir las representamos con ? ya que no se puede introducir ahí la variable creada anteriormente. Hacemos uso de **stmt** el cual es un statement que sirve para trabajar con las consultas de sql. Ponemos el **stmt**, seguido del tipo de set que vamos a utilizar ya sea Int o Date o String y dentro del paréntesis, primero la posición de la columna a la que nos referimos, seguido de la variable que queremos ingresar.

stmt.setTipo(posicion,variable);

Para borrar dichas en la tabla de la aplicación y de la base de datos, se hace unos del método personalizado **borrarDatos()**

```
public void borrarDatos() {
    int filaSeleccionada = jTable2.getSelectedRow();
    if (filaSeleccionada < 0) {
        JOptionPane.showMessageDialog(null, "Seleccione una ficha");
        return;
    }
    DefaultTableModel tabla = (DefaultTableModel) jTable2.getModel();
    int idFicha = (int) tabla.getValueAt(filaSeleccionada, 0); // Obtener el ID de la ficha médica

    String url = "jdbc:mysql://localhost:3306/clinica_veterinaria";
    String usuario = "root";
    String contraseña = "";

    String sql = "DELETE FROM fichamedica WHERE ID_ficha = ?";

    try (Connection conn = DriverManager.getConnection(url, usuario, contraseña); PreparedStatement stmt = conn.prepareStatement(sql)) {

        stmt.setInt(1, idFicha); // Establecer el ID en la consulta
        int filasEliminadas = stmt.executeUpdate();

        if (filasEliminadas > 0) {
            tabla.removeRow(filaSeleccionada); // Eliminar la fila de la tabla en la interfaz
        } else {
            // No se eliminó nada
        }
    }
}
```

El cual funciona seleccionando una ficha de la tabla, en caso de no seleccionar ninguna, aparece un mensaje solicitando que lo haga, una vez seleccionada, se obtiene el id de la ficha y se hace una consulta de **delete** basándose en dicho id.

por un lado se elimina la ficha en la base de datos ejecutando la consulta y por otro lado si se ha cumplido lo anterior, se procede a eliminar la fila en nuestra tabla de la aplicación.

Para modificar la ficha, es más de lo mismo, hacemos uso de **modificarFicha()** el cual comienza obteniendo los **id** de la ficha, veterinario y mascota y realizando una consulta para obtener el resto de datos basándose en el **id** de la ficha médica

```

public void modificarFicha() {
    int filaSeleccionada = jTable2.getSelectedRow();

    DefaultTableModel tabla = (DefaultTableModel) jTable2.getModel();
    int idFicha = (int) tabla.getValueAt(filaSeleccionada, 0); // Obtener el ID de la ficha médica
    int idVeterinario = Integer.parseInt(tabla.getValueAt(filaSeleccionada, 3).toString());
    int idMascota = Integer.parseInt(tabla.getValueAt(filaSeleccionada, 2).toString());

    String url = "jdbc:mysql://localhost:3306/clinica_veterinaria";
    String usuario = "root";
    String contraseña = "";

    String sql = "SELECT fecha, ID_mascota, ID_veterinario, sintomas, tratamiento, medicacion FROM fichamedica WHERE ID_ficha = ?";
}

```

con esto realizamos la consulta y todos los datos que nos proporciona, los introducimos en variables y después en sus campos correspondientes

```

stmt.setInt(1, idFicha);
ResultSet rs = stmt.executeQuery();

if (rs.next()) {
    Date fechaFicha = rs.getDate("fecha");
    //int idMascota = rs.getInt("ID_mascota");
    //int idVeterinario = rs.getInt("ID_veterinario");
    String sintoma = rs.getString("sintomas");
    String trata = rs.getString("tratamiento");
    String medica = rs.getString("medicacion");

    sintomas.setText(sintoma);
    fecha.setDate(fechaFicha);
    tratamiento.setText(trata);
    medicacion.setText(medica);
    cargarVeterinarios(idVeterinario);
    cargarMascotas(idMascota);
}

```

El método de **actualizarDatos()** es muy similar al de guardar datos, donde obtenemos los datos de los campos en variables y luego mediante una consulta de **update** los introducimos en la base de datos

```

public void actualizarDatos() {
    int filaSeleccionada = jTable2.getSelectedRow();
    DefaultTableModel tabla = (DefaultTableModel) jTable2.getModel();
    String sintoma = sintomas.getText();
    String medica = medicacion.getText();
    Date fechaFicha = fecha.getDate();
    java.sql.Date fechaSQL = new java.sql.Date(fechaFicha.getTime());
    String trata = tratamiento.getText();
    String seleccionadot1 = (String) comboVeterinarios.getSelectedItem();
    int idV = Integer.parseInt(seleccionadot1.split(" - ")[0]); // Extraer el ID
    String seleccionadot2 = (String) comboMascotas.getSelectedItem();
    int idM = Integer.parseInt(seleccionadot2.split(" - ")[0]); // Extraer el ID
    /*int idM = Integer.parseInt(idMascota.getText());
    int idV = Integer.parseInt(idVeterinario.getText());*/
    int idFicha = (int) tabla.getValueAt(filaSeleccionada, 0);
    System.out.println("ID de ficha a actualizar: " + idFicha);
    String url = "jdbc:mysql://localhost:3306/clinica_veterinaria";
    String usuario = "root";
    String contraseña = "";

    String sql = "UPDATE fichamedica SET fecha = ?, sintomas = ?, tratamiento = ?, medicacion = ?, ID_mascota = ?, ID_veterinario = ? WHERE ID_ficha = ?";

    try (Connection conn = DriverManager.getConnection(url, usuario, contraseña); PreparedStatement stmt = conn.prepareStatement(sql)) {

        stmt.setDate(1, fechaSQL);
        stmt.setString(2, sintoma);
        stmt.setString(3, trata);
        stmt.setString(4, medica);
        stmt.setInt(5, idM);
        stmt.setInt(6, idV);
        stmt.setInt(7, idFicha);
        stmt.executeUpdate();
    }
}

```

Y finalmente se introducen los métodos en los botones junto a algunos comandos para ocultar los paneles que no se usan y poner visibles los que se usan

```
private void guardarActionPerformed(java.awt.event.ActionEvent evt) {
    guardarDatos();
    jPanel1.setVisible(true);
    jPanel2.setVisible(false);
    limpiarCampos(jPanel2);
    cargarFichas();
}
```

```
private void volverActionPerformed(java.awt.event.ActionEvent evt) {
    jPanel1.setVisible(true);
    jPanel2.setVisible(false);
    limpiarCampos(jPanel2);
    cargarFichas();
}
```

El tema de la interfaz gráfica, una vez que era completamente funcional, se mejoró su aspecto quedando de la siguiente manera:

La página principal de las fichas:

Centro de control de Fichas Médicas

ID Ficha	Fecha	Mascota	Veterinario
13	2025-05-16	2	1
15	2025-05-19	1	1

Agregar

Modificar

Eliminar

El apartado de agregar:

Centro de control de Fichas Médicas

Agregar ficha

Número de colegiado

1 - COLVET123

Fecha

Mascota

1 - Firulais

Sintomas

Tratamiento

Medicación

Guardar

Volver

El apartado de modificar:

Centro de control de Fichas Médicas

Modificar ficha

Número de colegiado

1 - COLVET123

Mascota

1 - Firulais

Sintomas

das

Medicación

dsa

Fecha

19 may 2025

Tratamiento

vcx

Guardar

Volver

Para eliminar no hay un interfaz diferente ya que consiste únicamente en seleccionar una ficha de la tabla y clicar en eliminar