



INSTITUTO TECNOLOGICO DE OAXACA

PROYECTO FINAL DE TOPICOS AVANZADOS DE LA PROGRAMACION

MATERIA: TAP

EQUIPO 6: JIMENEZ LOPEZ FENANDO ARATH

JIMENEZ TORRES DIEGO ALEJANDRO

DOCUMENTACION DEL PROGRAMA

INDICE

Descripción del proyecto.....

Instrucciones de instalación.....

Explicación del funcionamiento y la estructura de la base de datos.....

Resumen de cada clase y sus responsabilidades.....

DESCRIPCION DEL PROYECTO:

Mi proyecto consiste en el desarrollo de una página web diseñada para facilitar y agilizar los trámites relacionados con la obtención, renovación o reposición de licencias. El objetivo principal es proporcionar una plataforma intuitiva y accesible que permita a los usuarios realizar estos trámites de manera eficiente, reduciendo el tiempo de espera y mejorando la experiencia del usuario.

El sistema incluye funcionalidades clave como:

- Registro y autenticación de usuarios para garantizar la seguridad de los datos.
- Formularios dinámicos para capturar la información requerida según el tipo de trámite.
- Integración con bases de datos para almacenar y gestionar los registros de los solicitantes.
- Interfaz amigable y responsive, optimizada para dispositivos móviles y de escritorio.

El desarrollo del proyecto se realizó utilizando tecnologías como Apache Netbeans. Además, se implementaron buenas prácticas de desarrollo web y seguridad para proteger los datos personales de los usuarios.

Este proyecto busca simplificar procesos administrativos tradicionales, promoviendo la digitalización y mejorando la accesibilidad a servicios esenciales.

INSTRUCCIONES DE INSTALACION

Las instrucciones de instalación son demasiado sencillas:

Se necesita un componente Javax.Mail.JAR:

Este archivo lo que hace es que puede que el programa tenga una interacción con el uso de correo y pueda adjuntar correo emisor y un correo receptor

Se necesita también el PDFBox-app-3.0.3.JAR.

Este archivo lo que otorga es la creación de pdf en el programa para que los datos que se encuentran en la tabla de usuarios se puedan extraer a un pdf y ese sea el documento que se le pueda enviar vía correo.

Explicación del funcionamiento y la estructura de la base de datos

Se crea una base de datos en mysql de nombre Sistema licencias, como siguiente se le agrega una tabla en donde se verán reflejados los datos de las licencias que se van creando por el usuario o el administrador los datos que tendrá la tabla serian los siguientes

```
id INT AUTO_INCREMENT PRIMARY KEY,  
numero_licencia VARCHAR(20) UNIQUE NOT NULL,  
nombre_apellidos VARCHAR(100) NOT NULL,  
edad INT NOT NULL,  
curp VARCHAR(18) NOT NULL,  
nacionalidad VARCHAR(50) NOT NULL,  
tipo_vehiculo VARCHAR(50) NOT NULL,  
tipo_sangre VARCHAR(3) NOT NULL,  
antiguedad VARCHAR(20) NOT NULL DEFAULT '0 años',  
fecha_expedicion DATE,  
fecha_vigencia DATE NOT NULL DEFAULT '2029-12-02'
```

CLASE CONEXIÓN

```
public static void establecerConexion() {  
    try {  
        String url = "jdbc:mysql://localhost:3306/sistema_licencias";  
        String usuario = "root"; // Cambia esto por tu usuario de MySQL  
        String contrasena = "1234"; // Cambia esto por tu contraseña de MySQL  
        conexion = DriverManager.getConnection(url, usuario, contrasena);  
        System.out.println("Conexión exitosa a la base de datos.");  
    } catch (SQLException e) {  
        e.printStackTrace();  
        JOptionPane.showMessageDialog(null, "Error al conectar con la base de datos.",  
            "Error", JOptionPane.ERROR_MESSAGE);  
        System.exit(1);  
    }  
}
```

Configura los datos de conexión:

- URL de la base de datos: "jdbc:mysql://localhost:3306/sistema_licencias".
- Usuario: "root".
- Contraseña: "1234".

Establece la conexión: Usa DriverManager.getConnection para conectarse a la base de datos y almacena la conexión en la variable conexion.

Mensajes y errores:

- Si tiene éxito, imprime: "*Conexión exitosa a la base de datos.*"
- Si falla, muestra un cuadro de diálogo con un mensaje de error y detiene el programa con System.exit(1).
-

RESUMEN DE CADA CLASE Y SUS RESPONSABILIDADES

MOSTRAR LOGIN USUARIO

```
public static void mostrarLoginUsuario(String Usuario) {  
    JFrame ventanaMenuUsu = new JFrame("Menú Principal");  
    ventanaMenuUsu.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    ventanaMenuUsu.setSize(400, 300); // Tamaño ajustado  
  
    // Usar un BorderLayout para mayor flexibilidad  
    ventanaMenuUsu.setLayout(new BorderLayout());  
  
    // Panel para los botones con un GridLayout ajustado  
    JPanel panelBotones = new JPanel(new GridLayout(2, 1, 10, 10));  
    panelBotones.setOpaque(false); // Para mantener el fondo del marco  
  
    // Crear botones con texto e imágenes  
    JButton botonRegistrar = new JButton("Registrar Nueva Licencia", new  
        ImageIcon("src/iconos/registrar.png"));  
    JButton botonSalir = new JButton("Salir", new ImageIcon("src/iconos/salir.png"));  
  
    ImageIcon iconoOriginal = new ImageIcon("src/iconos/registrar.png");  
    ImageIcon iconoOriginal2 = new ImageIcon("src/iconos/salir.png");  
    Image imagenEscalada = iconoOriginal.getImage().getScaledInstance(100, 100,  
        Image.SCALE_SMOOTH); // Cambia 30x30 al tamaño deseado  
    ImageIcon iconoRedimensionado = new ImageIcon(imagenEscalada);  
    botonRegistrar.setIcon(iconoRedimensionado);  
  
    Image imagenEscalada2 = iconoOriginal2.getImage().getScaledInstance(100, 100,  
        Image.SCALE_SMOOTH); // Cambia 30x30 al tamaño deseado
```

```

ImagenIconoRedimensionado2 = new ImageIcon(imagenEscalada2);

botonSalir.setIcon(iconoRedimensionado2);

// Personalizar el estilo de los botones

botonRegistrar.setHorizontalAlignment(SwingConstants.LEFT);

botonRegistrar.setFont(new Font("Arial", Font.PLAIN, 14));

botonRegistrar.setBackground(new Color(240, 248, 255)); // Color de fondo azul claro

botonRegistrar.setFocusPainted(false);

botonSalir.setHorizontalAlignment(SwingConstants.LEFT);

botonSalir.setFont(new Font("Arial", Font.PLAIN, 14));

botonSalir.setBackground(new Color(240, 248, 255));

botonSalir.setFocusPainted(false);

// Agregar una imagen decorativa en la parte superior

JLabel imagenDecorativa = new JLabel(new ImageIcon("src/imagenes2/.jpg"));

imagenDecorativa.setHorizontalAlignment(SwingConstants.CENTER);

// Agregar los botones al panel

panelBotones.add(botonRegistrar);

panelBotones.add(botonSalir);

// Agregar los componentes a la ventana

ventanaMenuUsu.add(imagenDecorativa, BorderLayout.NORTH);

ventanaMenuUsu.add(panelBotones, BorderLayout.CENTER);

```

```

botonRegistrar.setPreferredSize(new Dimension(100, 20)); // Ancho 200px, Alto 50px

botonSalir.setPreferredSize(new Dimension(100, 20));

// Configurar fondo de la ventana

ventanaMenuUsu.getContentPane().setBackground(new Color(173, 216, 230)); //
Fondo verde claro

ventanaMenuUsu.setLocationRelativeTo(null); // Centrar ventana

ventanaMenuUsu.setVisible(true);

// Acciones de los botones

botonRegistrar.addActionListener(e -> mostrarFormularioRegistro(ventanaMenuUsu));

botonSalir.addActionListener(e -> {
    ventanaMenuUsu.dispose();
    JOptionPane.showMessageDialog(null, "Saliendo del sistema. ¡Hasta luego!");
});

}

```

Ventana principal (JFrame):

- Se llama "*Menú Principal*" y tiene un tamaño de 400x300 píxeles.
- Usa un diseño BorderLayout para organizar sus componentes.

Componentes principales:

- **Botones:**
 - **Registrar Nueva Licencia:** Con una imagen (registrar.png) y texto. Al hacer clic, abre un formulario de registro.
 - **Salir:** Con una imagen (salir.png) y texto. Al hacer clic, cierra la ventana y muestra un mensaje de despedida.
- **Imagen decorativa:** Se muestra en la parte superior como un banner.

Estilo y diseño:

- Los botones se personalizan con fuentes, colores de fondo (azul claro), e imágenes redimensionadas a un tamaño específico (100x100 píxeles).
- Los componentes están organizados:
 - Imagen decorativa en la parte superior (NORTH del BorderLayout).
 - Panel con los botones en el centro (CENTER), usando un GridLayout de 2 filas y 1 columna.

Acciones de los botones:

- **botonRegistrar**: Llama a un método externo (`mostrarFormularioRegistro`) para abrir un formulario de registro.
- **botonSalir**: Cierra la ventana principal y muestra un mensaje de salida con `JOptionPane`.

Extras:

- Centra la ventana en la pantalla (`setLocationRelativeTo(null)`).
- Establece un fondo azul claro para la ventana completa.

CLASE PARA ENVIAR CORREO

```
private static void enviarCorreoConAdjunto(String destinatario, File archivoAdjunto) {  
  
    String remitente = "fl9840856@gmail.com"; // Tu correo  
  
    String contraseña = "tacato14"; // Tu contraseña  
  
    // Configurar propiedades SMTP  
  
    Properties props = new Properties();  
  
    props.put("mail.smtp.host", "smtp.gmail.com");  
  
    props.put("mail.smtp.port", "587");  
  
    props.put("mail.smtp.auth", "true");  
  
    props.put("mail.smtp.starttls.enable", "true");  
  
    // Crear sesión  
  
    Session session = Session.getInstance(props, new Authenticator() {
```

```

protected PasswordAuthentication getPasswordAuthentication() {
    return new PasswordAuthentication(remitente, contraseña);
}

});

try {
    // Crear el mensaje

    Message message = new MimeMessage(session);
    message.setFrom(new InternetAddress(remitente));
    message.setRecipients(Message.RecipientType.TO, InternetAddress.parse(destinatario));
    message.setSubject("Archivo adjunto");

    // Parte del mensaje de texto

    MimeBodyPart parteTexto = new MimeBodyPart();
    parteTexto.setText("Adjunto el archivo solicitado.");

    // Parte del archivo adjunto

    MimeBodyPart parteAdjunto = new MimeBodyPart();
    parteAdjunto.attachFile(archivoAdjunto);

    // Combinar ambas partes

    Multipart multipart = new MimeMultipart();
    multipart.addBodyPart(parteTexto);
    multipart.addBodyPart(parteAdjunto);
    message.setContent(multipart);

    // Enviar el mensaje

    Transport.send(message);
}

```

```

        JOptionPane.showMessageDialog(null, "Correo enviado exitosamente con el archivo adjunto.");
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Error al enviar el correo: " + e.getMessage());
    }
}

```

Datos básicos del remitente

- **Correo remitente:** "f19840856@gmail.com" (puedes reemplazarlo con otro correo).
 - **Contraseña:** "tacato14" (deberías usar un *App Password* de Gmail en lugar de la contraseña real por motivos de seguridad).
-

2. Configuración del servidor SMTP

- Usa el servidor de Gmail (`smtp.gmail.com`) para enviar el correo.
 - Configura propiedades como:
 - **Puerto:** 587 (para SMTP con TLS).
 - **Autenticación:** Habilitada (`mail.smtp.auth`).
 - **TLS:** Activado (`mail.smtp.starttls.enable`).
-

3. Autenticación y sesión

- Crea una sesión `Session` con las propiedades configuradas.
 - Implementa un autenticador (`Authenticator`) que usa el correo y la contraseña del remitente.
-

4. Creación del correo

- Usa la clase `MimeMessage` para construir el mensaje.
- **Remitente y destinatario:**
 - El remitente se configura con `setFrom`.
 - El destinatario se pasa como parámetro al método.
- **Asunto:** "Archivo adjunto".

- **Contenido del correo:**
 - **Cuerpo del mensaje (texto):** "Adjunto el archivo solicitado.".
 - **Archivo adjunto:** Se incluye usando un `MimeBodyPart`.
-

5. Envío del correo

- Une el texto y el archivo adjunto en un `Multipart` (estructura para manejar contenido mixto).
 - Configura el mensaje con `setContent` y usa `Transport.send` para enviarlo.
 - Si el envío es exitoso, muestra un cuadro de diálogo indicando éxito.
-

6. Manejo de errores

- Si ocurre algún problema durante el proceso (credenciales inválidas, archivo no encontrado, problemas de red), muestra un cuadro de diálogo con el mensaje de error.

MOSTRAR LICENCIAS EN PDF

```
public static void mostrarLicenciasParaPDF(JFrame ventanaAnterior) {  
  
    JFrame ventanaLicencias = new JFrame("Seleccionar Licencia para Descargar PDF");  
  
    ventanaLicencias.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);  
  
    ventanaLicencias.setSize(800, 400);  
  
    ventanaLicencias.setLocationRelativeTo(null);  
  
    try {  
  
        String sql = "SELECT * FROM licencias";  
  
        Statement statement = conexion.createStatement();  
  
        ResultSet resultSet = statement.executeQuery(sql);  
  
  
        String[] columnas = {"No. Licencia", "Nombre", "Edad", "CURP", "Nacionalidad", "Tipo Vehículo",  
"Tipo Sangre", "Expedición", "Vigencia"};
```

```

ArrayList<String[]> datos = new ArrayList<>();

while (resultSet.next()) {

    String numeroLicencia = String.valueOf(resultSet.getInt("numero_licencia"));

    String nombre = resultSet.getString("nombre_apellidos");

    String edad = String.valueOf(resultSet.getInt("edad"));

    String curp = resultSet.getString("curp");

    String nacionalidad = resultSet.getString("nacionalidad");

    String tipoVehiculo = resultSet.getString("tipo_vehiculo");

    String tipoSangre = resultSet.getString("tipo_sangre");

    String fechaExpedicion = resultSet.getString("fecha_expedicion");

    String fechaVigencia = resultSet.getString("fecha_vigencia");

    datos.add(new String[]{numeroLicencia, nombre, edad, curp, nacionalidad, tipoVehiculo,
        tipoSangre, fechaExpedicion, fechaVigencia});

}

String[][] datosArray = datos.toArray(new String[0][]);

JTable tabla = new JTable(datosArray, columnas);

JScrollPane scrollPane = new JScrollPane(tabla);

JPanel panelBotones = new JPanel();

 JButton botonDescargar = new JButton("Descargar PDF");

 JButton botonRegresar = new JButton("Regresar");

```

```

botonDescargar.addActionListener(e -> {

    int filaSeleccionada = tabla.getSelectedRow();

    if (filaSeleccionada != -1) {

        String nombreLicencia = tabla.getValueAt(filaSeleccionada, 1).toString(); // Nombre del
titular

        String numeroLicencia = tabla.getValueAt(filaSeleccionada, 0).toString(); // Número de
licencia

        String edad = tabla.getValueAt(filaSeleccionada, 2).toString();

        String curp = tabla.getValueAt(filaSeleccionada, 3).toString();

        String nacionalidad = tabla.getValueAt(filaSeleccionada, 4).toString();

        String tipoVehiculo = tabla.getValueAt(filaSeleccionada, 5).toString();

        String tipoSangre = tabla.getValueAt(filaSeleccionada, 6).toString();

        String fechaExpedicion = tabla.getValueAt(filaSeleccionada, 7).toString();

        String fechaVigencia = tabla.getValueAt(filaSeleccionada, 8).toString();

        // Generar PDF para la licencia seleccionada

        generarPDFLicencia(numeroLicencia, nombreLicencia, edad, curp, nacionalidad,
tipoVehiculo, tipoSangre, fechaExpedicion, fechaVigencia);

    } else {

        JOptionPane.showMessageDialog(ventanaLicencias, "Por favor, selecciona una licencia.");
    }

});

botonRegresar.addActionListener(e -> {

    ventanaLicencias.dispose();
}

```

```

        ventanaAnterior.setVisible(true);

    });

panelBotones.add(botonDescargar);
panelBotones.add(botonRegresar);

ventanaLicencias.setLayout(new BorderLayout());
ventanaLicencias.add(scrollPane, BorderLayout.CENTER);
ventanaLicencias.add(panelBotones, BorderLayout.SOUTH);
ventanaLicencias.setVisible(true);

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

    JOptionPane.showMessageDialog(null, "Error al obtener licencias.", "Error",
JOptionPane.ERROR_MESSAGE);
}

}
}

```

1. Ventana principal (JFrame):

- **Nombre:** "*Seleccionar Licencia para Descargar PDF*".
- **Tamaño:** 800x400 píxeles.
- **Ubicación centrada en la pantalla.**
- **Usa un diseño** BorderLayout.

2. Consulta a la base de datos:

- **Realiza una consulta SQL (SELECT * FROM licencias) para recuperar todas las licencias almacenadas.**
- **Extrae los datos relevantes:**
 - Número de licencia, nombre, edad, CURP, nacionalidad, tipo de vehículo, tipo de sangre, fecha de expedición y vigencia.
- **Los resultados se almacenan en una lista de arreglos (ArrayList<String[]>) para alimentar la tabla.**

3. Tabla de datos (JTable):

- Contiene los resultados obtenidos de la base de datos.
- Columnas: *No. Licencia, Nombre, Edad, CURP, Nacionalidad, Tipo Vehículo, Tipo Sangre, Expedición, Vigencia.*
- Incluye un JScrollPane para que sea navegable si hay muchos registros.

4. Botones:

- Descargar PDF:
 - Valida que el usuario seleccione una fila de la tabla.
 - Obtiene los datos de la fila seleccionada.
 - Llama al método generarPDFLicencia, pasándole los datos de la licencia para generar el archivo PDF correspondiente.
 - Si no hay una selección, muestra un mensaje de error con JOptionPane.
- Regresar:
 - Cierra la ventana actual y vuelve a la ventana anterior (ventanaAnterior).

CLASE PARA GENERAR PDF

```
public static void generarPDFLicencia(String numeroLicencia, String  
nombreLicencia, String edad,  
  
String curp, String nacionalidad, String tipoVehiculo, String tipoSangre,  
String fechaExpedicion, String fechaVencimiento) {
```

```
try {
```

Crear un documento PDF

```
PDDocument documento = new PDDocument();
```

Crear una página de tamaño A4

```
PDPage pagina = new PDPage(PDRectangle.A4);  
  
documento.addPage(pagina);
```

Cargar la fuente personalizada (Arial.ttf)

```
File archivoFuente = new File("C:\Windows\Fonts\Arial.ttf");  
  
if (!archivoFuente.exists()) {  
  
    JOptionPane.showMessageDialog(null, "No se encontró la fuente Arial.ttf  
en la ruta especificada.", "Error", JOptionPane.ERROR_MESSAGE);  
  
    return;  
  
}  
  
PDFont fuentePersonalizada = PDType0Font.load(documento,  
archivoFuente);
```

Crear un objeto de texto en la página

```
try (PDPageContentStream contenido = new  
PDPageContentStream(documento, pagina)) {  
  
    contenido.beginText();
```

```
contenido.setFont(fuentePersonalizada, 12); // Usar la fuente cargada

contenido.setLeading(14.5f);

contenido.newLineAtOffset(50, 750);

contenido.showText("Licencia No. " + numeroLicencia);

contenido.newLine();

contenido.showText("Nombre: " + nombreLicencia);

contenido.newLine();

contenido.showText("Edad: " + edad);

contenido.newLine();

contenido.showText("CURP: " + curp);

contenido.newLine();

contenido.showText("Nacionalidad:" + nacionalidad);

contenido.newLine();

contenido.showText("Vehiculo:" + tipoVehiculo);

contenido.newLine();

contenido.showText("Sangre: " + tipoSangre);

contenido.newLine();

contenido.showText("Fecha del trámite: " + fechaExpedicion);
```

```
contenido.newLine();

contenido.showText("Fecha Vencimiento: " + fechaVencimiento);

contenido.newLine();
```

Añadir más datos de la licencia si es necesario

```
contenido.endText();

} catch (IOException e) {

    JOptionPane.showMessageDialog(null, "Error al escribir en el PDF",
"Error", JOptionPane.ERROR_MESSAGE);

    e.printStackTrace();

    return;

}
```

Pedir al usuario dónde guardar el PDF

```
JFileChooser fileChooser = new JFileChooser();

fileChooser.setDialogTitle("Guardar Licencia como PDF");

fileChooser.setAcceptAllFileFilterUsed(false);
```

```
    fileChooser.addChoosableFileFilter(new FileNameExtensionFilter("PDF Files",
"pdf"));

    int resultado = fileChooser.showSaveDialog(null);

    if (resultado == JFileChooser.APPROVE_OPTION) {

        File archivo = fileChooser.getSelectedFile();

        if (!archivo.getName().endsWith(".pdf")) {

            archivo = new File(archivo.getAbsolutePath() + ".pdf");

        }

        documento.save(archivo);

        JOptionPane.showMessageDialog(null, "PDF guardado con éxito en:" + 
archivo.getAbsolutePath());

    }

}
```

Cerrar el documento

```
documento.close();

} catch (IOException e) {

    e.printStackTrace();
```

```
        JOptionPane.showMessageDialog(null, "Error al generar el PDF.", "Error",
JOptionPane.ERROR_MESSAGE);

    }

}
```

MOSTRAR FORMULARIO REGISTRO

```
public static void mostrarFormularioRegistro(JFrame ventanaAnterior) {

    JFrame ventanaRegistro = new JFrame("Registrar Nueva Licencia");

    ventanaRegistro.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

    Tamaño ajustado

    ventanaRegistro.setSize(400, 500);

    ventanaRegistro.setLayout(new GridLayout(8, 2, 10, 10));



    JLabel etiquetaNombre = new JLabel("Nombre:");

    JTextField campoNombre = new JTextField();

    JLabel etiquetaEdad = new JLabel("Edad:");

    JTextField campoEdad = new JTextField();

    JLabel etiquetaCURP = new JLabel("CURP:");

    JTextField campoCURP = new JTextField();
```

```
JLabel etiquetaNombre = new JLabel("Nombre:");
JTextField campoNombre = new JTextField();
JLabel etiquetaEdad = new JLabel("Edad:");
JTextField campoEdad = new JTextField();
JLabel etiquetaCURP = new JLabel("CURP:");
JTextField campoCURP = new JTextField();
JLabel etiquetaNacionalidad = new JLabel("Nacionalidad:");
JTextField campoNacionalidad = new JTextField();
JLabel etiquetaTipoVehiculo = new JLabel("Tipo Vehículo:");
JTextField campoTipoVehiculo = new JTextField();
JLabel etiquetaTipoSangre = new JLabel("Tipo Sangre:");
JTextField campoTipoSangre = new JTextField();
JButton botonGuardar = new JButton("Guardar");
JButton botonRegresar = new JButton("Regresar");

ventanaRegistro.add(etiquetaNombre);
ventanaRegistro.add(campoNombre);
ventanaRegistro.add(etiquetaEdad);
ventanaRegistro.add(campoEdad);
ventanaRegistro.add(etiquetaCURP);
ventanaRegistro.add(campoCURP);
ventanaRegistro.add(etiquetaNacionalidad);
ventanaRegistro.add(campoNacionalidad);
ventanaRegistro.add(etiquetaTipoVehiculo);
ventanaRegistro.add(campoTipoVehiculo);
```

```
ventanaRegistro.add(etiquetaTipoSangre);

ventanaRegistro.add(campoTipoSangre);

ventanaRegistro.add(botonRegresar);

ventanaRegistro.add(botonGuardar);

ventanaRegistro.getContentPane().setBackground(Color.cyan); // Fondo negro

ventanaRegistro.setVisible(true);

ventanaRegistro.setLocationRelativeTo(null);

botonGuardar.addActionListener(e -> {

    // Tu lógica para guardar la licencia

    String nombre = campoNombre.getText();

    String edadTexto = campoEdad.getText();

    String curp = campoCURP.getText();

    String nacionalidad = campoNacionalidad.getText();

    String tipoVehiculo = campoTipoVehiculo.getText();

    String tipoSangre = campoTipoSangre.getText();

    Validación de campos vacíos
```

```

    if (nombre.isEmpty() || edadTexto.isEmpty() || curp.isEmpty() ||
nacionalidad.isEmpty() || tipoVehiculo.isEmpty() || tipoSangre.isEmpty()) {

        JOptionPane.showMessageDialog(ventanaRegistro, "Por favor, complete
todos los campos.", "Error", JOptionPane.ERROR_MESSAGE);

        return;
    }

```

Validación de la edad

```

int edad;

try {
    edad = Integer.parseInt(edadTexto);

    if (edad < 17) {

        JOptionPane.showMessageDialog(ventanaRegistro, "No se pueden
registrar menores de 17 años.", "Error", JOptionPane.ERROR_MESSAGE);

        return;
    }

} catch (NumberFormatException ex) {

    JOptionPane.showMessageDialog(ventanaRegistro, "La edad debe ser un
número.", "Error", JOptionPane.ERROR_MESSAGE);

    return;
}

```

```
}
```

Validación del tipo de sangre

```
if (!tipoSangre.matches("[A-Za-z+-]{1,3}")) {  
  
    JOptionPane.showMessageDialog(ventanaRegistro, "El tipo de sangre debe  
ser válido (ej. O+, AB-, etc.).", "Error", JOptionPane.ERROR_MESSAGE);  
  
    return;  
  
}
```

```
try {
```

Inserción de los datos en la base de datos

```
String sql = "INSERT INTO licencias (nombre_apellidos, edad, curp,  
nacionalidad, tipo_vehiculo, tipo_sangre, fecha_expedicion, fecha_vigencia) " +  
  
"VALUES (?, ?, ?, ?, ?, ?, CURDATE(), '2029-12-02')";
```

```
PreparedStatement statement = conexion.prepareStatement(sql);
```

```
statement.setString(1, nombre);
```

```
statement.setInt(2, edad);
```

```
statement.setString(3, curp);
```

```
statement.setString(4, nacionalidad);
```

```
statement.setString(5, tipoVehiculo);

statement.setString(6, tipoSangre);

statement.executeUpdate();

JOptionPane.showMessageDialog(ventanaRegistro, "Licencia registrada con éxito.");

ventanaRegistro.dispose();

ventanaAnterior.setVisible(true);

} catch (SQLException ex) {

    ex.printStackTrace();

    JOptionPane.showMessageDialog(ventanaRegistro, "Error al guardar la licencia.", "Error", JOptionPane.ERROR_MESSAGE);

}

});

botonRegresar.addActionListener(e -> {

    ventanaRegistro.dispose();

    ventanaAnterior.setVisible(true);

});
```

```
}
```

MOSTRAR LICENSIAS

```
public static void mostrarLicenciasExistentes(JFrame ventanaAnterior) {  
  
    JFrame ventanaLicencias = new JFrame("Licencias Existentes");  
  
    ventanaLicencias.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);  
  
    ventanaLicencias.setSize(800, 400);  
  
    ventanaLicencias.setLocationRelativeTo(null);  
  
    try {  
  
        String sql = "SELECT * FROM licencias";  
  
        Statement statement = conexion.createStatement();  
  
        ResultSet resultSet = statement.executeQuery(sql);  
  
        String[] columnas = {"No. Licencia", "Nombre", "Edad", "CURP",  
"Nacionalidad", "Tipo Vehículo", "Tipo Sangre", "Expedición", "Vigencia"};  
  
        ArrayList<String[]> datos = new ArrayList<>();  
  
        while (resultSet.next()) {  
  
            String numeroLicencia =  
String.valueOf(resultSet.getInt("numero_licencia"));  
    }  
}
```

```
String nombre = resultSet.getString("nombre_apellidos");

String edad = String.valueOf(resultSet.getInt("edad"));

String curp = resultSet.getString("curp");

String nacionalidad = resultSet.getString("nacionalidad");

String tipoVehiculo = resultSet.getString("tipo_vehiculo");

String tipoSangre = resultSet.getString("tipo_sangre");

String fechaExpedicion = resultSet.getString("fecha_expedicion");

String fechaVigencia = resultSet.getString("fecha_vigencia");

datos.add(new String[]{numeroLicencia, nombre, edad, curp,
nacionalidad, tipoVehiculo, tipoSangre, fechaExpedicion, fechaVigencia});

}

String[][] datosArray = datos.toArray(new String[0][]{});

JTable tabla = new JTable(datosArray, columnas);

JScrollPane scrollPane = new JScrollPane(tabla);

JPanel panelBotones = new JPanel();

JButton botonEliminar = new JButton("Eliminar");
```

```

JButton botonModificar = new JButton("Modificar");

JButton botonRegresar = new JButton("Regresar");

botonEliminar.addActionListener(e -> {

    int filaSeleccionada = tabla.getSelectedRow();

    if (filaSeleccionada != -1) {

        String numeroLicencia = tabla.getValueAt(filaSeleccionada,
0).toString();

        int confirmacion = JOptionPane.showConfirmDialog(ventanaLicencias,
"¿Está seguro de eliminar esta licencia?", "Confirmación",
JOptionPane.YES_NO_OPTION);

        if (confirmacion == JOptionPane.YES_OPTION) {

            eliminarLicencia(numeroLicencia);

            ((DefaultTableModel)
tabla.getModel()).removeRow(filaSeleccionada);

            JOptionPane.showMessageDialog(ventanaLicencias, "Licencia
eliminada exitosamente.");
        }
    } else {

        JOptionPane.showMessageDialog(ventanaLicencias, "Por favor,
seleccione una licencia para eliminar.");
    }
});

```

```
}

});

botonModificar.addActionListener(e -> {

    int filaSeleccionada = tabla.getSelectedRow();

    if (filaSeleccionada != -1) {

        String numeroLicencia = tabla.getValueAt(filaSeleccionada,
0).toString();

        mostrarFormularioModificar(numeroLicencia, ventanaLicencias);

    } else {

        JOptionPane.showMessageDialog(ventanaLicencias, "Por favor,
seleccione una licencia para modificar.");

    }

});

botonRegresar.addActionListener(e -> {

    ventanaLicencias.dispose();

    ventanaAnterior.setVisible(true);

});
```

```

panelBotones.add(botonEliminar);

panelBotones.add(botonModificar);

panelBotones.add(botonRegresar);

ventanaLicencias.setLayout(new BorderLayout());

ventanaLicencias.add(scrollPane, BorderLayout.CENTER);

ventanaLicencias.add(panelBotones, BorderLayout.SOUTH);

ventanaLicencias.setVisible(true);

} catch (SQLException ex) {

    ex.printStackTrace();

    JOptionPane.showMessageDialog(null, "Error al obtener licencias.", "Error",
JOptionPane.ERROR_MESSAGE);

}

}

```

□ **Ventana (JFrame):**

- Nombre: "*Licencias Existentes*".
- Tamaño: 800x400 píxeles.
- Ubicación centrada en la pantalla.
- Disposición general: BorderLayout.

Consulta SQL:

- Recupera todos los registros de la tabla licencias usando la consulta:
- □ Extrae los datos:
 - Número de licencia, nombre, edad, CURP, nacionalidad, tipo de vehículo, tipo de sangre, fecha de expedición y fecha de vigencia.
- Los datos se almacenan en una lista (ArrayList) para ser utilizados en la tabla.

Tabla de licencias (JTable):

- Contiene los datos obtenidos de la base de datos.
- Encabezados de columna: *No. Licencia, Nombre, Edad, CURP, Nacionalidad, Tipo Vehículo, Tipo Sangre, Expedición, Vigencia*.
- Incluye un JScrollPane para desplazarse si hay muchos registros.

Botones disponibles:

- **Eliminar:**
 - Valida que se seleccione una fila.
 - Muestra un cuadro de confirmación (JOptionPane) para confirmar la eliminación.
 - Llama al método eliminarLicencia pasando el número de licencia seleccionado.
 - Elimina la fila seleccionada de la tabla con DefaultTableModel.
 - Muestra un mensaje de éxito tras la eliminación.
- **Modificar:**
 - Valida que se seleccione una fila.
 - Obtiene el número de licencia de la fila seleccionada.
 - Llama al método mostrarFormularioModificar, pasando la información de la licencia y la ventana actual.
- **Regresar:**
 - Cierra la ventana actual y vuelve a la ventana anterior (ventanaAnterior).

Diseño:

- La tabla ocupa el centro de la ventana.
- Los botones se colocan en un panel en la parte inferior de la ventana

MOSTRAR FORMULARIO PARA MODIFICAR

```
private static void mostrarFormularioModificar(String numeroLicencia, JFrame ventanaAnterior) {
```

```
    JFrame ventanaModificar = new JFrame("Modificar Licencia");
```

```
    ventanaModificar.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
```

```

ventanaModificar.setSize(400, 500);

ventanaModificar.setLayout(new GridLayout(8, 2, 10, 10));

JLabel etiquetaNombre = new JLabel("Nombre:");
JTextField campoNombre = new JTextField();

JLabel etiquetaEdad = new JLabel("Edad:");
JTextField campoEdad = new JTextField();

JLabel etiquetaCURP = new JLabel("CURP:");
JTextField campoCURP = new JTextField();

JLabel etiquetaNacionalidad = new JLabel("Nacionalidad:");
JTextField campoNacionalidad = new JTextField();

JLabel etiquetaTipoVehiculo = new JLabel("Tipo Vehículo:");
JTextField campoTipoVehiculo = new JTextField();

JLabel etiquetaTipoSangre = new JLabel("Tipo Sangre:");
JTextField campoTipoSangre = new JTextField();

JButton botonGuardar = new JButton("Guardar");
JButton botonCancelar = new JButton("Cancelar");

// Cargar datos actuales

try {

    String sql = "SELECT * FROM licencias WHERE numero_licencia = ?";

    PreparedStatement statement = conexion.prepareStatement(sql);

    statement.setString(1, numeroLicencia);

    ResultSet resultSet = statement.executeQuery();
}

```

```

        if (resultSet.next()) {

            campoNombre.setText(resultSet.getString("nombre_apellidos"));

            campoEdad.setText(String.valueOf(resultSet.getInt("edad")));

            campoCURP.setText(resultSet.getString("curp"));

            campoNacionalidad.setText(resultSet.getString("nacionalidad"));

            campoTipoVehiculo.setText(resultSet.getString("tipo_vehiculo"));

            campoTipoSangre.setText(resultSet.getString("tipo_sangre"));

        } else {

            JOptionPane.showMessageDialog(ventanaModificar, "Licencia no
encontrada.", "Error", JOptionPane.ERROR_MESSAGE);

            return;

        }

    } catch (SQLException ex) {

        ex.printStackTrace();

        JOptionPane.showMessageDialog(ventanaModificar, "Error al cargar datos de la
licencia.", "Error", JOptionPane.ERROR_MESSAGE);

        return;

    }

// Agregar campos al formulario

ventanaModificar.add(etiquetaNombre);

ventanaModificar.add(campoNombre);

ventanaModificar.add(etiquetaEdad);

ventanaModificar.add(campoEdad);

```

```

ventanaModificar.add(etiquetaCURP);

ventanaModificar.add(campoCURP);

ventanaModificar.add(etiquetaNacionalidad);

ventanaModificar.add(campoNacionalidad);

ventanaModificar.add(etiquetaTipoVehiculo);

ventanaModificar.add(campoTipoVehiculo);

ventanaModificar.add(etiquetaTipoSangre);

ventanaModificar.add(campoTipoSangre);

ventanaModificar.add(botonCancelar);

ventanaModificar.add(botonGuardar);

// Acción para guardar los cambios

botonGuardar.addActionListener(e -> {

    try {

        String sql = "UPDATE licencias SET nombre_apellidos = ?, edad = ?, curp = ? , nacionalidad = ?, tipo_vehiculo = ?, tipo_sangre = ? WHERE numero_licencia = ?";

        PreparedStatement statement = conexion.prepareStatement(sql);

        statement.setString(1, campoNombre.getText());

        statement.setInt(2, Integer.parseInt(campoEdad.getText()));

        statement.setString(3, campoCURP.getText());

        statement.setString(4, campoNacionalidad.getText());

        statement.setString(5, campoTipoVehiculo.getText());

        statement.setString(6, campoTipoSangre.getText());

        statement.setString(7, numeroLicencia);
    }
})

```

```

        statement.executeUpdate();

        JOptionPane.showMessageDialog(ventanaModificar, "Licencia modificada
exitosamente.");

        ventanaModificar.dispose();

        ventanaAnterior.setVisible(true);

    } catch (SQLException ex) {

        ex.printStackTrace();

        JOptionPane.showMessageDialog(ventanaModificar, "Error al modificar la
licencia.", "Error", JOptionPane.ERROR_MESSAGE);

    } catch (NumberFormatException ex) {

        JOptionPane.showMessageDialog(ventanaModificar, "Por favor, ingrese datos
válidos.", "Error", JOptionPane.ERROR_MESSAGE);

    }

});

// Acción para cancelar la modificación

botonCancelar.addActionListener(e -> {

    ventanaModificar.dispose();

    ventanaAnterior.setVisible(true);

});

ventanaModificar.setVisible(true);

}

```

- **Creación de la ventana:**

- Tamaño: 400x500.
- Diseño: Formulario con etiquetas y campos (GridLayout).

- **Carga de datos:**

- Consulta SQL

```
SELECT * FROM licencias WHERE numero_licencia = ?
```

Botones:

- **Guardar:** Actualiza la licencia en la base de datos con

```
UPDATE licencias SET ... WHERE numero_licencia = ?
```

- Muestra un mensaje de éxito o error.
- **Cancelar:** Cierra la ventana sin guardar cambios.

- **Validación y errores:**

- Valida que los datos sean correctos (edad como número).
- Maneja errores de SQL o entrada inválida con mensajes al usuario.

(MENU DE INICIO)

```
public static void mostrarVentanaInicioSesion() {  
    JFrame ventanalInicio = new JFrame("Seleccione Tipo de Usuario");  
    ventanalInicio.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```

ventanainicio.setSize(400, 300); // Ajustar tamaño para más espacio

// Usar BorderLayout para mayor flexibilidad
ventanainicio.setLayout(new BorderLayout());

// Panel para los botones
JPanel panelBotones = new JPanel(new GridLayout(2, 1, 10, 10));
panelBotones.setOpaque(false); // Fondo transparente para mostrar el fondo principal

// Crear botones con imágenes al lado
JButton botonAdmin = new JButton("Administrador", new
ImageIcon("src/iconos/admin_icon.jpg"));

JButton botonUsuario = new JButton("Usuario", new
ImageIcon("src/iconos/user_icon.jfif"));

Image iconoOriginal = new ImageIcon("src/iconos/admin_icon.jpg");
Image iconoOriginal2 = new ImageIcon("src/iconos/iconos.jpg");

Image imagenEscalada = iconoOriginal.getImage().getScaledInstance(100, 100,
Image.SCALE_SMOOTH); // Cambia 30x30 al tamaño deseado

ImageIcon iconoRedimensionado = new ImageIcon(imagenEscalada);
botonAdmin.setIcon(iconoRedimensionado);

Image imagenEscalada2 = iconoOriginal2.getImage().getScaledInstance(100, 100,
Image.SCALE_SMOOTH); // Cambia 30x30 al tamaño deseado

ImageIcon iconoRedimensionado2 = new ImageIcon(imagenEscalada2);
botonUsuario.setIcon(iconoRedimensionado2);

```

```
// Personalizar botones

botonAdmin.setHorizontalAlignment(SwingConstants.LEFT); // Texto e imagen alineados
botonAdmin.setFont(new Font("Arial", Font.PLAIN, 14));
botonAdmin.setBackground(new Color(240, 248, 255)); // Fondo azul claro
botonAdmin.setFocusPainted(false);

botonUsuario.setHorizontalAlignment(SwingConstants.LEFT);
botonUsuario.setFont(new Font("Arial", Font.PLAIN, 14));
botonUsuario.setBackground(new Color(240, 248, 255));
botonUsuario.setFocusPainted(false);

// Agregar botones al panel

panelBotones.add(botonAdmin);
panelBotones.add(botonUsuario);

// Agregar una imagen decorativa en la parte inferior

JLabel imagenInferior = new JLabel(new ImageIcon("src/imagenes/footer_icon.png"));
imagenInferior.setHorizontalAlignment(SwingConstants.CENTER);

// Fondo principal de la ventana

JLabel fondo = new JLabel(new ImageIcon("src/imagenes/fondo_inicio.png"));
fondo.setLayout(new BorderLayout()); // Para que los componentes se posic和平n
correctamente
```

```
// Agregar componentes al fondo
fondo.add(panelBotones, BorderLayout.CENTER); // Panel de botones al centro
fondo.add(imagenInferior, BorderLayout.SOUTH); // Imagen decorativa al pie de la
ventana

// Agregar fondo a la ventana
ventanainicio.setContentPane(fondo);

ventanainicio.setLocationRelativeTo(null); // Centrar ventana
ventanainicio.setVisible(true);

// Acciones de los botones
botonAdmin.addActionListener(e -> {
    ventanainicio.dispose(); // Cierra la ventana de selección
    mostrarLogin("administrador"); // Llama a login con administrador
});

botonUsuario.addActionListener(e -> {
    ventanainicio.dispose(); // Cierra la ventana de selección
    mostrarLoginusuario("usuario"); // Llama a login con usuario
});

}
```

