

# Manual Técnico.

Hemos desarrollado un CRUD en NetBeans la cual tiene una conexión a una base de datos (MySQL Workbench).

Primero creamos una clase llamada LoginF, en donde se pide usuario y contraseña y si esto es correcto nos permita dirigirnos a la siguiente interfaz que es ConexiónBD, pero si los datos ingresados en el LoginF son incorrectos nos muestre un mensaje de error.

```
1  package SParcial;
2
3  import javax.swing.*;
4  import java.awt.*;
5  import java.awt.event.ActionEvent;
6  import java.awt.event.ActionListener;
7  import java.io.File;
8  import javax.imageio.ImageIO;
9
10 public class LoginF {
11
12     public static void main(String[] args) {
13         SwingUtilities.invokeLater(LoginF::new);
14     }
15
16     public LoginF() {
17         // Crear el marco
18         JFrame frame = new JFrame("Inicio");
19         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
20         frame.setSize(400, 350);
21         frame.setLocationRelativeTo(null);
22
23         // Cambiar el icono
24         ImageIcon icon = new ImageIcon("C:\\Users\\DELL\\Desktop\\PFFP\\ProFinal\\src\\login.jpg"); // Ajusta la ruta
25         frame.setIconImage(icon.getImage());
26
27         // Crear y añadir el panel con imagen de fondo
28         BackgroundPanel panel = new BackgroundPanel();
29         panel.setLayout(null);
30     }
```

```
32     JLabel iniLabel = new JLabel("Bienvenido");
33     iniLabel.setForeground(Color.WHITE);
34     iniLabel.setFont(new Font("Elephant", Font.BOLD, 30));
35     iniLabel.setBounds(100, 18, 200, 25);
36     panel.add(iniLabel);
37
38     JLabel usuarioLabel = new JLabel("Usuario:");
39     usuarioLabel.setForeground(Color.WHITE);
40     usuarioLabel.setFont(new Font("Elephant", Font.BOLD, 17));
41     usuarioLabel.setBounds(35, 150, 150, 25);
42     panel.add(usuarioLabel);
43
44     JLabel contraseñaLabel = new JLabel("Contraseña:");
45     contraseñaLabel.setForeground(Color.WHITE);
46     contraseñaLabel.setFont(new Font("Elephant", Font.BOLD, 17));
47     contraseñaLabel.setBounds(35, 200, 155, 25);
48     panel.add(contraseñaLabel);
49
50     JTextField usuarioText = new JTextField();
51     usuarioText.setBounds(155, 150, 200, 25);
52     panel.add(usuarioText);
53
54     JPasswordField contraseñaText = new JPasswordField();
55     contraseñaText.setBounds(155, 200, 200, 25);
56     panel.add(contraseñaText);
57
58     // Crear el botón "Iniciar Sesión"
59     JButton iniciarBTN = new JButton("Iniciar Sesión");
60     iniciarBTN.setForeground(Color.WHITE);
61     iniciarBTN.setBackground(Color.BLUE);
```

```
Source History
62      iniciarBTN.setFont(new Font("Arial", Font.BOLD, 14));
63      iniciarBTN.setBounds(155, 270, 130, 25);
64      panel.add(iniciarBTN);
65
66      // Crear el botón "Salir"
67      JButton salirBTN = new JButton("Salir");
68      salirBTN.setForeground(Color.WHITE);
69      salirBTN.setBackground(Color.BLUE);
70      salirBTN.setFont(new Font("Arial", Font.BOLD, 14));
71      salirBTN.setBounds(295, 270, 75, 25);
72      panel.add(salirBTN);
73
74      // Acción al hacer clic en "Iniciar Sesión"
75      iniciarBTN.addActionListener(e -> {
76          String usuario = usuarioText.getText();
77          String contraseña = new String(contraseñaText.getPassword());
78
79          if (usuario.equals("elicia") && contraseña.equals("mas")) {
80              frame.dispose(); // Cierra la ventana de inicio de sesión
81              SwingUtilities.invokeLater(() -> {
82                  // Inicia la ventana de la aplicación JavaFX
83                  ConexionBD.launch(ConexionBD.class); // Llama a la clase ConexionBD que es una aplicación JavaFX
84              });
85          } else {
86              JOptionPane.showMessageDialog(frame, "Usuario o contraseña incorrectos.", "Error", JOptionPane.ERROR_MESSAGE);
87          }
88      });
89
90      // Acción al hacer clic en "Salir"
91      salirBTN.addActionListener(e -> System.exit(0));
92
93      // Añadir el panel al marco
94      frame.add(panel);
95      frame.setVisible(true);
96  }
97
98      // Clase para el panel con imagen de fondo
99      static class BackgroundPanel extends JPanel {
100          private Image backgroundImage;
101
102          public BackgroundPanel() {
103              try {
104                  backgroundImage = ImageIO.read(new File("C:\\Users\\DELL\\Desktop\\Progra II\\Vacacion\\Form Vacacion\\background.jpg"));
105              } catch (Exception e) {
106                  e.printStackTrace();
107              }
108          }
109
110          @Override
111          protected void paintComponent(Graphics g) {
112              super.paintComponent(g);
113              if (backgroundImage != null) {
114                  g.drawImage(backgroundImage, 0, 0, getWidth(), getHeight(), this);
115              }
116          }
117      }
118  }
119  }
```

Después de haber ingresado los datos correctos nos dirigirá a la interfaz ConexionBD, en donde tenemos un menú para nuestro Crud, al nosotros seleccionar una opción nos dirigirá al proceso que hace cada botón, que está conectado a una base de datos.

## Menú, botones

```
147
148 // Ruta de la imagen de fondo
149 String imagePath = "file://C:/Users/DELL/Desktop/Progra%20II/Vacacion/Form%20Vacaciones/src/fr.jpg"; // Aseg
150
151 // Cargar la imagen
152 Image image = new Image(imagePath);
153 BackgroundImage backgroundImage = new BackgroundImage(image,
154     BackgroundRepeat.NO_REPEAT,
155     BackgroundRepeat.NO_REPEAT,
156     BackgroundPosition.CENTER,
157     new BackgroundSize(BackgroundSize.AUTO, BackgroundSize.AUTO, true, true, false, false));
158
159 // Layout principal
160 VBox root = new VBox(10);
161 root.setAlignment(Pos.CENTER);
162
163 // Asignar el fondo a la VBox
164 root.setBackground(new Background(backgroundImage));
165
166 // Crear un título personalizado con fuente y color
167 Button btnC = new Button("CRUD");
168 btnC.setStyle(
169     "-fx-font-size: 75px; " +
170     "-fx-padding: 3 23; " + // Ajuste de padding (márgenes internos)
171     "-fx-text-fill: #bdf44a; " + // Color personalizado para el texto (en este caso)
172     "-fx-font-weight: bold; " + // Texto en negrita
173     "-fx-font-family: 'Georgia'; " + // Fuente personalizada
174     "-fx-border-width: 0; " + // Eliminar borde alrededor del botón
175     "-fx-focus-color: transparent; " + // Eliminar color de borde cuando esté en foco
176     "-fx-background-color: transparent; " + // Fondo transparente
177
```

```
181 // Botones para cada acción
182 Button btnIngresar = new Button("Ingresar Producto");
183 btnIngresar.setStyle("-fx-font-size: 16px; -fx-padding: 3 23; -fx-background-color: #DAF7A6; -fx-text-fill: b
184 btnIngresar.setOnAction(e -> showIngresarProducto(stage));
185
186 Button btnMostrar = new Button("Mostrar Productos");
187 btnMostrar.setStyle("-fx-font-size: 16px; -fx-padding: 3 21; -fx-background-color: #DAF7A6; -fx-text-fill: bl
188 btnMostrar.setOnAction(e -> showMostrarProductos(stage));
189
190 Button btnActualizar = new Button("Actualizar Producto");
191 btnActualizar.setStyle("-fx-font-size: 16px; -fx-padding: 3 18; -fx-background-color: #DAF7A6; -fx-text-fill:
192 btnActualizar.setOnAction(e -> showActualizarProducto(stage));
193
194 Button btnBuscar = new Button("Buscar por Código ");
195 btnBuscar.setStyle("-fx-font-size: 16px; -fx-padding: 3 23; -fx-background-color: #DAF7A6; -fx-text-fill: bla
196 btnBuscar.setOnAction(e -> showBuscarProducto(stage));
197
198 Button btnBuscarNom = new Button("Buscar por Nombre ");
199 btnBuscarNom.setStyle("-fx-font-size: 16px; -fx-padding: 3 18; -fx-background-color: #DAF7A6; -fx-text-fill:
200 btnBuscarNom.setOnAction(e -> showBuscarNom(stage));
201
202 Button btnEliminar = new Button("Eliminar Producto");
203 btnEliminar.setStyle("-fx-font-size: 16px; -fx-padding: 3 23; -fx-background-color: #DAF7A6; -fx-text-fill: b
204 btnEliminar.setOnAction(e -> showEliminarProducto(stage));
205
206 Button btnSalir = new Button("Salir");
207 btnSalir.setStyle("-fx-font-size: 16px; -fx-padding: 3 23; -fx-background-color: #DAF7A6; -fx-text-fill: bla
208 btnSalir.setOnAction(e -> System.exit(0));
209
210 root.getChildren().addAll(btnC, btnIngresar, btnMostrar, btnActualizar, btnBuscar, btnBuscarNom, btnEliminar, b
```

```
1 package SParcial;
2
3 import java.awt.Font;
4 import javafx.application.Application;
5 import javafx.collections.FXCollections;
6 import javafx.collections.ObservableList;
7 import javafx.geometry.Pos;
8 import javafx.scene.Scene;
9 import javafx.scene.control.*;
10 import javafx.scene.image.Image;
11 import javafx.scene.layout.*;
12 import javafx.scene.paint.Color;
13 import javafx.stage.Stage;
14
15 import java.sql.*;
16 import java.util.ArrayList;
17 import javafx.geometry.Insets;
18 import javafx.swing.JLabel;
19
20 public class ConexionBD extends Application {
21
22     private static final String URL = "jdbc:mysql://localhost:3306/BDFloreriaLaElegancia";
23     private static final String USER = "root";
24     private static final String PASSWORD = "ma213027%ca";
25
26     // Conexión a la base de datos
27     public static Connection conectar() {
28         Connection conexion = null;
29         try {
30             conexion = DriverManager.getConnection(URL, USER, PASSWORD);
31
```

```

Start Page X SQL 1 X ConexionBD.java X LoginF.java X
Source History
32 } catch (SQLException e) {
33     System.out.println("Error al conectar: " + e.getMessage());
34 }
35 return conexion;
36 }
37
38 // Insertar un producto en la base de datos
39 public static void insertarProducto(int codigo, String nombre, double precio, int cantidad, String fecha) {
40     String query = "INSERT INTO producto (codigoProducto, nombreProducto, precioUnitario, cantidadProducto, fechaVencimiento) VALUES (?, ?, ?, ?, ?)";
41     try (Connection con = ConexionBD.conectar(); PreparedStatement pst = con.prepareStatement(query)) {
42         pst.setInt(1, codigo);
43         pst.setString(2, nombre);
44         pst.setDouble(3, precio);
45         pst.setInt(4, cantidad);
46         pst.setDate(5, java.sql.Date.valueOf(fecha));
47         pst.executeUpdate();
48         System.out.println("Producto insertado correctamente");
49     } catch (SQLException e) {
50         System.out.println("Error al insertar el producto: " + e.getMessage());
51     }
52 }
53
54 // Listar los productos de la base de datos
55 public static String[] listarProductos() {
56     String query = "SELECT * FROM producto";
57     ArrayList<String> productos = new ArrayList<>();
58     try (Connection con = ConexionBD.conectar(); Statement st = con.createStatement(); ResultSet rs = st.executeQuery()) {
59         while (rs.next()) {
60             String producto = "Código: " + rs.getInt("codigoProducto") + ", Nombre: " + rs.getString("nombreProducto") + ", Precio: " + rs.getDouble("precioUnitario") + ", Cantidad: " + rs.getInt("cantidadProducto") + ", Fecha Vencimiento: " + rs.getDate("fechaVencimiento");
61             productos.add(producto);
62         }
63     } catch (SQLException e) {
64         System.out.println("Error al listar los productos: " + e.getMessage());
65     }
66     return productos.toArray(new String[0]);
67 }
68
69 // Actualizar un producto
70 public static void actualizarProducto(int codigo, String nombre, double precio, int cantidad, String fecha) {
71     String query = "UPDATE producto SET nombreProducto = ?, precioUnitario = ?, cantidadProducto = ?, fechaVencimiento = ? WHERE codigoProducto = ?";
72     try (Connection con = ConexionBD.conectar(); PreparedStatement pst = con.prepareStatement(query)) {
73         pst.setString(1, nombre);
74         pst.setDouble(2, precio);
75         pst.setInt(3, cantidad);
76         pst.setDate(4, java.sql.Date.valueOf(fecha));
77         pst.setInt(5, codigo);
78         pst.executeUpdate();
79         System.out.println("Producto actualizado correctamente");
80     } catch (SQLException e) {
81         System.out.println("Error al actualizar el producto: " + e.getMessage());
82     }
83 }
84
85 // Buscar un producto
86 public static String buscarProducto(int codigo) {
87     String query = "SELECT * FROM producto WHERE codigoProducto = ?";
88     try (Connection con = ConexionBD.conectar(); PreparedStatement pst = con.prepareStatement(query)) {
89         pst.setInt(1, codigo);
90         ResultSet rs = pst.executeQuery();
91         if (rs.next()) {
92             return "Código: " + rs.getInt("codigoProducto") + ", Nombre: " + rs.getString("nombreProducto") + ", Precio: " + rs.getDouble("precioUnitario") + ", Cantidad: " + rs.getInt("cantidadProducto") + ", Fecha Vencimiento: " + rs.getDate("fechaVencimiento");
93         } else {
94             return "Producto no encontrado";
95         }
96     } catch (SQLException e) {
97         System.out.println("Error al buscar el producto: " + e.getMessage());
98         return "Error al buscar el producto";
99     }
100 }
101
102 public static String buscarProductoPorNombre(String buscarNom) {
103     String query = "SELECT * FROM producto WHERE nombreProducto LIKE ?";
104     try (Connection con = ConexionBD.conectar(); PreparedStatement pst = con.prepareStatement(query)) {
105         // Usamos LIKE para permitir coincidencias parciales en el nombre
106         pst.setString(1, "%" + buscarNom + "%");
107         ResultSet rs = pst.executeQuery();
108         // Si encontramos productos que coinciden
109         if (rs.next()) {
110             return "Código: " + rs.getInt("codigoProducto") + ", Nombre: " + rs.getString("nombreProducto") + ", Precio: " + rs.getDouble("precioUnitario") + ", Cantidad: " + rs.getInt("cantidadProducto") + ", Fecha Vencimiento: " + rs.getDate("fechaVencimiento");
111         } else {
112             return "Producto no encontrado";
113         }
114     } catch (SQLException e) {
115         System.out.println("Error al buscar el producto: " + e.getMessage());
116         return "Error al buscar el producto";
117     }
118 }
119
120 public static void eliminarProducto(int codigo) {
121     String query = "DELETE FROM producto WHERE codigoProducto = ?";
122     try (Connection con = ConexionBD.conectar(); PreparedStatement pst = con.prepareStatement(query)) {
123         pst.setInt(1, codigo);
124         pst.executeUpdate();
125         System.out.println("Producto eliminado correctamente");
126     } catch (SQLException e) {
127         System.out.println("Error al eliminar el producto: " + e.getMessage());
128     }
129 }
130
131 }
132
133
134
135
136

```

## ➤ Ingresar.

```
private void showIngresarProducto(Stage stage) {
    VBox ingresarLayout = new VBox(15); // 15px de espacio entre los elementos verticales
    ingresarLayout.setAlignment(Pos.CENTER);

    // Fondo de la ventana
    String imagenRuta = "file:/C:/Users/DELL/Desktop/Progra II/Vacacion/Form Vacaciones/src/fo.jpg"; // Ruta de t
    Image imagenFondo = new Image(imagenRuta);
    BackgroundImage backgroundImage = new BackgroundImage(
        imagenFondo,
        BackgroundRepeat.NO_REPEAT, // No repetir la imagen
        BackgroundRepeat.NO_REPEAT, // No repetir la imagen
        BackgroundPosition.CENTER, // Centrar la imagen
        BackgroundSize.DEFAULT // Usar tamaño por defecto
    );
    Background fondo = new Background(backgroundImage);
    ingresarLayout.setBackground(fondo); // Establecer el fondo de la VBox

    // Titulo
    Label labelIngresar = new Label("Ingresar Producto");
    labelIngresar.setStyle(
        "-fx-font-size: 30px; " + // Tamaño de la fuente
        "-fx-padding: 3 18; " + // Espaciado alrededor del texto (top, right, bottom, left)
        "-fx-text-fill: #0d486a; " + // Color del texto
        "-fx-font-weight: bold; " + // Peso de la fuente (negrita)
        "-fx-font-family: 'Georgia';" // Familia de la fuente
    );

    labelIngresar.setPadding(new javafx.geometry.Insets(0, 0, 10, 0)); // Espacio debajo del titulo

    codigoField.setStyle("-fx-pref-width: 175px; -fx-border-radius: 5px; -fx-padding: 5px; -fx-border-color: #cc");
    hboxCodigo.getChildren().addAll(labelCodigo, codigoField);
    HBox.setMargin(labelCodigo, new Insets(0, 0, 0, 10)); // Margen a la izquierda

    HBox hboxNombre = new HBox(10);
    Label labelNombre = new Label("Nombre:");
    TextField nombreField = new TextField();
    nombreField.setPromptText("Nombre del Producto");
    nombreField.setStyle("-fx-pref-width: 175px; -fx-border-radius: 5px; -fx-padding: 5px; -fx-border-color: #cc");
    hboxNombre.getChildren().addAll(labelNombre, nombreField);
    HBox.setMargin(labelNombre, new Insets(0, 0, 0, 10)); // Margen a la izquierda

    HBox hboxPrecio = new HBox(10);
    Label labelPrecio = new Label("Precio:");
    TextField precioField = new TextField();
    precioField.setPromptText("Precio");
    precioField.setStyle("-fx-pref-width: 175px; -fx-border-radius: 5px; -fx-padding: 5px; -fx-border-color: #cc");
    hboxPrecio.getChildren().addAll(labelPrecio, precioField);
    HBox.setMargin(labelPrecio, new Insets(0, 0, 0, 10)); // Margen a la izquierda

    HBox hboxCantidad = new HBox(10);
    Label labelCantidad = new Label("Cantidad:");
    TextField cantidadField = new TextField();
    cantidadField.setPromptText("Cantidad");
    cantidadField.setStyle("-fx-pref-width: 175px; -fx-border-radius: 5px; -fx-padding: 5px; -fx-border-color: #cc");
    hboxCantidad.getChildren().addAll(labelCantidad, cantidadField);
    HBox.setMargin(labelCantidad, new Insets(0, 0, 0, 10)); // Margen a la izquierda

    HBox hboxFecha = new HBox(10);
    Label labelFecha = new Label("Fecha de Vencimiento:");

    HBox.setMargin(labelFecha, new Insets(0, 0, 0, 10)); // Margen a la izquierda

    // Botón para ingresar el producto
    Button btnIngresar = new Button("Ingresar");
    btnIngresar.setStyle("-fx-background-color: #0d486a; -fx-text-fill: white; -fx-border-radius: 5px; -fx-padding: 10 20");
    btnIngresar.setOnAction(e -> {
        try {
            int codigo = Integer.parseInt(codigoField.getText());
            String nombre = nombreField.getText();
            double precio = Double.parseDouble(precioField.getText());
            int cantidad = Integer.parseInt(cantidadField.getText());
            String fecha = fechaField.getText();
            insertarProducto(codigo, nombre, precio, cantidad, fecha);
            showAlert("Producto ingresado correctamente");
            showMostrarProductos(stage);
        } catch (NumberFormatException ex) {
            showAlert("Por favor ingrese valores válidos");
        }
    });

    // Botón para volver al menú principal
    Button btnVolver = new Button("Volver al Menú");
    btnVolver.setStyle("-fx-background-color: #ff5555; -fx-text-fill: white; -fx-border-radius: 5px; -fx-padding: 10 20");
    btnVolver.setOnAction(e -> start(stage));

    // Agregar todos los elementos al layout
    ingresarLayout.getChildren().addAll(labelIngresar, hboxCodigo, hboxNombre, hboxPrecio, hboxCantidad, hboxFecha, btnIngresar, btnVolver);

    // Establecer un margen en el VBox para asegurarse de que haya espacio entre los elementos y los bordes de la v
    ingresarLayout.setPadding(new Insets(20, 20, 20, 20)); // 20px de margen alrededor del VBox
}
```

## ➤ Mostrar

```
private void showMostrarProductos(Stage stage) {
    VBox mostrarLayout = new VBox(8);
    mostrarLayout.setAlignment(Pos.CENTER);

    ListView<String> productList = new ListView<>();
    ObservableList<String> products = FXCollections.observableArrayList(listarProductos());
    productList.setItems(products);

    // Botón para volver al menú
    Button btnVolver = new Button("Volver al Menú");
    btnVolver.setOnAction(e -> start(stage));

    Label labelListado = new Label("Listado de Productos");

    // Aplicar el estilo (color de texto, fuente, tamaño, etc.)
    labelListado.setStyle(
        "-fx-font-size: 24px; " + // Tamaño de fuente
        "-fx-text-fill: #2C3E50; " + // Color del texto (puedes cambiar el color por el que desees:
        "-fx-font-weight: bold; " + // Peso de la fuente
        "-fx-font-family: 'Georgia';" // Tipo de fuente (puedes cambiar 'Arial' por cualquier otro:
    );
    mostrarLayout.getChildren().addAll(labelListado, productList, btnVolver);
    Scene mostrarScene = new Scene(mostrarLayout, 350, 300);
    stage.setScene(mostrarScene);
}
```

## ➤ Actualizar.

```
350 private void showActualizarProducto(Stage stage) {
351     VBox actualizarLayout = new VBox(15); // Espaciado de 15px entre los elementos
352     actualizarLayout.setAlignment(Pos.CENTER);
353
354     // Establecer el fondo de la ventana
355     String imagenRuta = "file:/C:/Users/DELL/Desktop/Progra II/Vacacion/Form Vacaciones/src/fo.jpg";
356     Image imagenFondo = new Image(imagenRuta);
357     BackgroundImage backgroundImage = new BackgroundImage(
358         imagenFondo,
359         BackgroundRepeat.NO_REPEAT,
360         BackgroundRepeat.NO_REPEAT,
361         BackgroundPosition.CENTER,
362         BackgroundSize.DEFAULT
363     );
364     Background fondo = new Background(backgroundImage);
365     actualizarLayout.setBackground(fondo);
366
367     // Título
368     Label labelActualizar = new Label("Actualizar Producto");
369     labelActualizar.setStyle(
370         "-fx-font-size: 30px; " +
371         "-fx-padding: 3 18; " +
372         "-fx-text-fill: #0d486a; " +
373         "-fx-font-weight: bold; " +
374         "-fx-font-family: 'Georgia';"
375     );
376
377     Label labelCodigo = new Label("Codigo");
378     codigoField.setPromptText("Código del Producto");
379     codigoField.setStyle("-fx-pref-width: 150px; -fx-border-radius: 5px; -fx-padding: 5px;");
380     hboxCodigo.getChildren().addAll(labelCodigo, codigoField);
381
382     HBox hboxNombre = new HBox(10);
383     Label labelNombre = new Label("Nombre:");
384     TextField nombreField = new TextField();
385     nombreField.setPromptText("Nombre del Producto");
386     nombreField.setStyle("-fx-pref-width: 150px; -fx-border-radius: 5px; -fx-padding: 5px;");
387     hboxNombre.getChildren().addAll(labelNombre, nombreField);
388
389     HBox hboxPrecio = new HBox(10);
390     Label labelPrecio = new Label("Precio:");
391     TextField precioField = new TextField();
392     precioField.setPromptText("Precio");
393     precioField.setStyle("-fx-pref-width: 150px; -fx-border-radius: 5px; -fx-padding: 5px;");
394     hboxPrecio.getChildren().addAll(labelPrecio, precioField);
395
396     HBox hboxCantidad = new HBox(10);
397     Label labelCantidad = new Label("Cantidad:");
398     TextField cantidadField = new TextField();
399     cantidadField.setPromptText("Cantidad");
400     cantidadField.setStyle("-fx-pref-width: 150px; -fx-border-radius: 5px; -fx-padding: 5px;");
401     hboxCantidad.getChildren().addAll(labelCantidad, cantidadField);
402
403     HBox hboxFecha = new HBox(10);
404     Label labelFecha = new Label("Fecha de Vencimiento:");
405     TextField fechaField = new TextField();
406     fechaField.setPromptText("Vec. (yyyy-MM-dd)");
407 }
```

```

        int codigo = Integer.parseInt(codigoField.getText());
        String nombre = nombreField.getText();
        double precio = Double.parseDouble(precioField.getText());
        int cantidad = Integer.parseInt(cantidadField.getText());
        String fecha = fechaField.getText();
        actualizarProducto(codigo, nombre, precio, cantidad, fecha);
        showAlert("Producto actualizado correctamente");
        showMostrarProductos(stage);
    } catch (NumberFormatException ex) {
        showAlert("Por favor ingrese valores válidos");
    }
});

// Botón para volver al menú
Button btnVolver = new Button("Volver al Menú");
btnVolver.setStyle("-fx-background-color: #ff5555; -fx-text-fill: white; -fx-border-radius: 5px; -fx-padding: 10");
btnVolver.setOnAction(e -> start(stage));

// Agregar todos los elementos al VBox
actualizarLayout.getChildren().addAll(
    labelActualizar, hboxCodigo, hboxNombre, hboxPrecio, hboxCantidad, hboxFecha, btnActualizar, btnVolver
);

// Establecer el padding del VBox (alrededor de todos los elementos)
actualizarLayout.setPadding(new Insets(20)); // Un padding de 20px para asegurar el margen adecuado desde el bc

// Crear la escena y establecerla en el escenario
Scene actualizarScene = new Scene(actualizarLayout, 390, 370);
stage.setScene(actualizarScene);
}

```

## ➤ Buscar por código

```

private void showBuscarProducto(Stage stage) {
    VBox buscarLayout = new VBox(15);
    buscarLayout.setAlignment(Pos.CENTER);

    // Fondo de la ventana
    String imagenRuta = "file:/C:/Users/DELL/Desktop/Progra II/Vacacion/Form Vacaciones/src/fo.jpg";
    Image imagenFondo = new Image(imagenRuta);
    BackgroundImage backgroundImage = new BackgroundImage(
        imagenFondo,
        BackgroundRepeat.NO_REPEAT,
        BackgroundRepeat.NO_REPEAT,
        BackgroundPosition.CENTER,
        BackgroundSize.DEFAULT
    );
    Background fondo = new Background(backgroundImage);
    buscarLayout.setBackground(fondo);

    // Título
    Label labelBuscar = new Label("Buscar Producto");
    labelBuscar.setStyle(
        "-fx-font-size: 30px; " +
        "-fx-padding: 3 18; " +
        "-fx-text-fill: #0d486a; " +
        "-fx-font-weight: bold; " +
        "-fx-font-family: 'Georgia';"
    );

    // Campo para el código
    HBox hboxCodigo = new HBox(10);
    Label labelCodigo = new Label("Código a buscar:");

    codigoField.setPromptText("Código del Producto");
    codigoField.setStyle("-fx-pref-width: 150px; -fx-border-radius: 5px; -fx-padding: 5px;");
    hboxCodigo.getChildren().addAll(labelCodigo, codigoField);

    // Botón de búsqueda
    Button btnBuscar = new Button("Buscar");
    btnBuscar.setStyle("-fx-background-color: #0d486a; -fx-text-fill: white; -fx-border-radius: 5px; -fx-paddi");
    btnBuscar.setOnAction(e -> {
        try {
            int codigo = Integer.parseInt(codigoField.getText());
            String result = buscarProducto(codigo);
            showAlert(result);
        } catch (NumberFormatException ex) {
            showAlert("Por favor ingrese un código válido");
        }
    });

    // Botón para volver
    Button btnVolver = new Button("Volver al Menú");
    btnVolver.setStyle("-fx-background-color: #ff5555; -fx-text-fill: white; -fx-border-radius: 5px; -fx-paddi");
    btnVolver.setOnAction(e -> start(stage));

    // Agregar todo al layout
    buscarLayout.getChildren().addAll(labelBuscar, hboxCodigo, btnBuscar, btnVolver);
    buscarLayout.setPadding(new Insets(20)); // Añadir margen a la ventana

    // Crear la escena
    Scene buscarScene = new Scene(buscarLayout, 380, 250);
    stage.setScene(buscarScene);
}

```



## ➤ Buscar por nombre.

```
private void showBuscarNom(Stage stage) {
    VBox buscarLayout = new VBox(15); // 15px de espacio entre los elementos
    buscarLayout.setAlignment(Pos.CENTER);

    // Fondo de la ventana
    String imagenRuta = "file:/C:/Users/DELL/Desktop/Progra II/Vacacion/Form Vacaciones/src/fo.jpg";
    Image imagenFondo = new Image(imagenRuta);
    BackgroundImage backgroundImage = new BackgroundImage(
        imagenFondo,
        BackgroundRepeat.NO_REPEAT,
        BackgroundRepeat.NO_REPEAT,
        BackgroundPosition.CENTER,
        BackgroundSize.DEFAULT
    );
    Background fondo = new Background(backgroundImage);
    buscarLayout.setBackground(fondo);

    // Titulo
    Label labelBuscar = new Label("Buscar por Nombre");
    labelBuscar.setStyle(
        "-fx-font-size: 30px; " +
        "-fx-padding: 3 18; " +
        "-fx-text-fill: #0d486a; " +
        "-fx-font-weight: bold; " +
        "-fx-font-family: 'Georgia';"
    );

    // Campo para el nombre
    HBox hboxNombre = new HBox(10);

    Button btnBuscar = new Button("Buscar");
    btnBuscar.setStyle("-fx-background-color: #0d486a; -fx-text-fill: white; -fx-border-radius: 5px; -fx-pa");
    btnBuscar.setOnAction(e -> {
        String buscarNom = nombreField.getText().trim();
        if (!buscarNom.isEmpty()) {
            String result = buscarProductoPorNombre(buscarNom);
            showAlert(result);
        } else {
            showAlert("Por favor ingrese un nombre válido");
        }
    });

    // Botón para volver al menú
    Button btnVolver = new Button("Volver al Menú");
    btnVolver.setStyle("-fx-background-color: #ff5555; -fx-text-fill: white; -fx-border-radius: 5px; -fx-pa");
    btnVolver.setOnAction(e -> start(stage));

    // Agregar todos los elementos al VBox
    buscarLayout.getChildren().addAll(
        labelBuscar, hboxNombre, btnBuscar, btnVolver
    );

    // Establecer un margen alrededor del VBox
    buscarLayout.setPadding(new Insets(20)); // 20px de margen

    // Crear la escena y establecerla en el escenario
    Scene buscarScene = new Scene(buscarLayout, 390, 250);
    stage.setScene(buscarScene);
}
```

## ➤ Eliminar

```
private void showEliminarProducto(Stage stage) {
    VBox eliminarLayout = new VBox(15); // Espaciado de 15px entre los elementos
    eliminarLayout.setAlignment(Pos.CENTER);

    // Fondo de la ventana
    String imagenRuta = "file:/C:/Users/DELL/Desktop/Progra II/Vacacion/Form Vacaciones/src/fo.jpg";
    Image imagenFondo = new Image(imagenRuta);
    BackgroundImage backgroundImage = new BackgroundImage(
        imagenFondo,
        BackgroundRepeat.NO_REPEAT,
        BackgroundRepeat.NO_REPEAT,
        BackgroundPosition.CENTER,
        BackgroundSize.DEFAULT
    );
    Background fondo = new Background(backgroundImage);
    eliminarLayout.setBackground(fondo);

    // Titulo
    Label labelEliminar = new Label("Eliminar Producto");
    labelEliminar.setStyle(
        "-fx-font-size: 30px; " +
        "-fx-padding: 3 18; " +
        "-fx-text-fill: #0d486a; " +
        "-fx-font-weight: bold; " +
        "-fx-font-family: 'Georgia';"
    );

    // Campo para el código del producto
    HBox hboxCodigo = new HBox(10);
```



```

btnEliminar.setStyle("-fx-background-color: #0d486a; -fx-text-fill: white; -fx-border-radius: !
btnEliminar.setOnAction(e -> {
    try {
        int codigo = Integer.parseInt(codigoField.getText());
        eliminarProducto(codigo);
        showAlert("Producto eliminado correctamente");
        showMostrarProductos(stage);
    } catch (NumberFormatException ex) {
        showAlert("Por favor ingrese un código válido");
    }
});

```

```

// Botón para volver al menú
Button btnVolver = new Button("Volver al Menú");
btnVolver.setStyle("-fx-background-color: #ff5555; -fx-text-fill: white; -fx-border-radius: 5p
btnVolver.setOnAction(e -> start(stage));

```

```

// Agregar todos los elementos al VBox
eliminarLayout.getChildren().addAll(
    labelEliminar, hboxCodigo, btnEliminar, btnVolver
);

```

```

// Establecer un margen alrededor del VBox
eliminarLayout.setPadding(new Insets(20)); // 20px de margen

```

```

// Crear la escena y establecerla en el escenario
Scene eliminarScene = new Scene(eliminarLayout, 370, 250);
stage.setScene(eliminarScene);

```

```

private void showAlert(String message) {
    Alert alert = new Alert(Alert.AlertType.INFORMATION);
    alert.setTitle("Mensaje");
    alert.setHeaderText(null);
    alert.setContentText(message);
    alert.showAndWait();
}

```

```

// Ejecutar la aplicación
public static void main(String[] args) {
    launch(args);
}

```