TALLER DE DESARROLLO DE APLICACIONES 1

SEMANA: 4

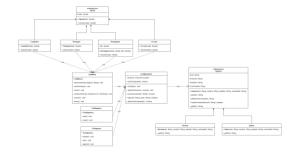
ENTREGADO POR: Mescua Segovia Marino

CÓDIGO: N04058C

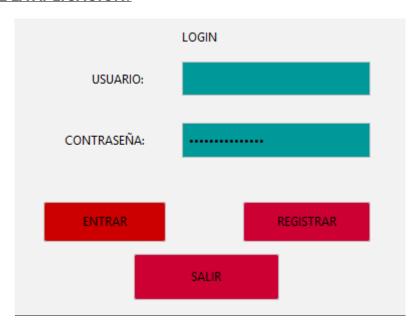
CATEDRÁTICO: Fernández Bejarano Raúl Enrique

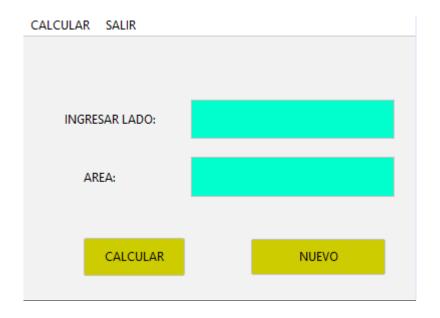
ACTIVIDAD 1:

DIAGRAMA DE CLASES:



DISEÑO DE LA APLICACIÓN:







CÓDIGO DE LA APLICACIÓN:

```
public class LoginControlador {
.6
.7
         LoginVista vista:
.9
         public LoginControlador(LoginVista vista) {
            this.vista = vista;
1
2
             vista.btnEntrar.addActionListener(e -> ingresar());
:3
             vista.btnSalir.addActionListener(e -> salir());
4
             vista.btnRegistrar.addActionListener(e -> abrirRegistro());
:5
!7
  阜
         public void ingresar() {
:8
             String usuarioIngresado = vista.txt_usuario.getText();
19
             String contrasenaIngresada = new String(vista.txtContrasena.getPassword());
10
1
             if (usuarioIngresado.equals("admin") && contrasenaIngresada.equals("admin")) {
2
                 JOptionPane.showMessageDialog(null, "Bienvenido Admin");
13
                 // Abrir MenuAdmin aquí
4
             } else {
5
                 boolean encontrado = false;
17
                 for (ModeloUsuario u : ModeloUsuario.listaUsuarios) {
8
                     if (u.dni.equals(usuarioIngresado) && u.contrasena.equals(contrasenaIngresada)) {
                         JOptionPane.showMessageDialog(null, "Bienvenido" + u.nombre);
0
                         // Abrir Menu Usuario aquí
                         encontrado = true;
1
2
                         break;
3
5
                 if (!encontrado) {
7
                     JOptionPane.showMessageDialog(null, "Usuario o contraseña incorrectos");
9
0
1
2
  早
         public void salir() {
             vista.dispose();
4
5
  口
         public void abrirRegistro() {
7
             JOptionPane.showMessageDialog(null, "Aquí abrirías el formulario de registro");
8
             // Puedes hacer new RegistroVista() y ControladorRegistro
0
     }
```

```
12
     public class MenuAdminControlador {
13
14
         MenuAdmin vista;
15
         String figuraSeleccionada = "";
16
17
          public MenuAdminControlador(MenuAdmin vista) {
18
             this.vista = vista;
19
20
21
             vista.itemCuadrado.addActionListener(e -> seleccionarFigura("cuadrado"));
22
             vista.itemCirculo.addActionListener(e -> seleccionarFigura("circulo"));
             vista.itemRectangulo.addActionListener(e -> seleccionarFigura("rectangulo"));
23
24
             vista.itemTriangulo.addActionListener(e -> seleccionarFigura("triangulo"));
25
26
              // Botones
27
             vista.btn calcular.addActionListener(e -> calcular());
28
             vista.btn nuevo.addActionListener(e -> nuevo());
29
30
             vista.btn_salir.addActionListener(e -> vista.dispose());
31
32
              // Escritura en campos
33
              vista.txt_lado.addKeyListener(new KeyAdapter() {
λŧ
                 public void keyReleased(KeyEvent e) {
35
                     validarInputs();
36
                 1
37
             });
38
39
              vista.txt_base.addKeyListener(new KeyAdapter() {
<u>}</u>↓
                 public void keyReleased(KeyEvent e) {
41
                      validarInputs();
42
43
             });
44
45
46
  口
         public void selectionarFigura(String figura) {
             figuraSeleccionada = figura;
48
             vista.txt lado.setText("");
             vista.txt_base.setText("");
49
50
             vista.txt_area.setText("");
51
52
             vista.txt_lado.setEnabled(false);
53
             vista.txt_base.setEnabled(false);
54
              vista.btn_calcular.setEnabled(false);
55
              vista.btn_nuevo.setEnabled(false);
56
57
              // Habilitar solo lo necesario
              if (figura.equals("cuadrado") || figura.equals("circulo")) {
58
```

```
if (figura.equals("cuadrado") || figura.equals("circulo")) {
58 [-
59
                  vista.txt lado.setEnabled(true);
60
61
                  vista.txt_lado.setEnabled(true);
62
                  vista.txt_base.setEnabled(true);
63
64
65
66 📮
          public void validarInputs() {
67
              boolean campoLleno = !vista.txt_lado.getText().isEmpty();
              boolean campoBase = !vista.txt_base.getText().isEmpty();
68
69
   卓
70
              if (figuraSeleccionada.equals("cuadrado") || figuraSeleccionada.equals("circulo")) {
71
                  vista.btn_calcular.setEnabled(campoLleno);
72
              } else {
73
                  vista.btn calcular.setEnabled(campoLleno && campoBase);
74
75
76
77 F
          public void calcular() {
              try {
79
                  double lado = Double.parseDouble(vista.txt_lado.getText());
                  double base = vista.txt_base.isEnabled() ? Double.parseDouble(vista.txt_base.getText()) : 0;
80
81
                  double area = 0;
82
                  switch (figuraSeleccionada) {
 <u>@</u>
84
                      case "cuadrado":
                         area = lado * lado;
85
86
                          break:
87
                      case "circulo":
88
                         area = Math.PI * lado * lado;
89
90
                      case "rectangulo":
91
                         area = lado * base;
92
                          break:
93
                      case "triangulo":
                        area = (lado * base) / 2;
94
95
                          break;
96
97
98
                  vista.txt_area.setText(String.valueOf(area));
99
                  vista.txt_lado.setEnabled(false);
100
                  vista.txt_base.setEnabled(false);
101
                  vista.btn_calcular.setEnabled(false);
102
                  vista.btn nuevo.setEnabled(true);
103
104
              } catch (NumberFormatException e) {
105
                  JOptionPane.showMessageDialog(null, "Ingrese números válidos");
```

```
108
109
           public void nuevo() {
110
              vista.txt_lado.setText("");
111
               vista.txt_base.setText("");
112
               vista.txt_area.setText("");
113
              vista.btn_calcular.setEnabled(false);
114
              vista.btn_nuevo.setEnabled(false);
115
              if (figuraSeleccionada.equals("cuadrado") || figuraSeleccionada.equals("circulo")) {
116
117
                   vista.txt lado.setEnabled(true);
               } else {
118
119
                  vista.txt_lado.setEnabled(true);
120
                   vista.txt_base.setEnabled(true);
121
122
123
124
```

```
12
        public class MenuControlador {
13
14
            Menu vista:
            String figuraSelectionada = "";
15
16
17
            public MenuControlador(Menu vista) {
    18
               this.vista = vista;
19
20
                // Asociar eventos
21
                vista.itemCuadrado.addActionListener(e -> seleccionarFigura("cuadrado"));
22
                vista.itemCirculo.addActionListener(e -> seleccionarFigura("circulo"));
23
24
                vista.btnCalcular.addActionListener(e -> calcular());
25
                vista.btnNuevo.addActionListener(e -> nuevo());
26
27
    中中中
                vista.txt_lado.addKeyListener(new KeyAdapter() {
29
29
                   public void keyReleased(KeyEvent e) {
                        if (!vista.txt_lado.getText().isEmpty()) {
30
                            vista.btnCalcular.setEnabled(true);
31
    32
                            vista.btnCalcular.setEnabled(false);
22
34
35
                });
36
37
28
    public void selectionarFigura(String figura) {
39
                figuraSeleccionada = figura;
40
                vista.txt_lado.setText("");
41
                vista.txt_area.setText("");
42
                vista.txt_lado.setEnabled(true);
42
                vista.btnCalcular.setEnabled(false);
                vista.btnNuevo.setEnabled(false);
44
45
46
    47
            public void calcular() {
48
                try {
49
                    double valor = Double.parseDouble(vista.txt lado.qetText());
50
                    double area = 0;
51
52
                    if (figuraSeleccionada.equals("cuadrado")) {
                        area = valor * valor;
53
54
    占
                    } else if (figuraSeleccionada.equals("circulo")) {
55
                        area = Math.PI * valor * valor;
56
57
                    vista.txt_area.setText(String.valueOf(area));
58
59
                    vista.txt_lado.setEnabled(false);
60
                    vista.btnCalcular.setEnabled(false);
61
                    vista.btnNuevo.setEnabled(true);
62
    白
                } catch (NumberFormatException e) {
                    JOptionPane.showMessageDialog(null, "Ingrese un valor válido");
63
64
65
66
```

```
66
67
           public void nuevo() {
   68
              vista.txt_lado.setText("");
69
               vista.txt_area.setText("");
70
               vista.txt_lado.setEnabled(true);
71
               vista.btnCalcular.setEnabled(false);
72
               vista.btnNuevo.setEnabled(false);
73
74
75
```

```
public class RegistroControlador {
17
          RegistroVista vista;
18
19 📮
          public RegistroControlador(RegistroVista vista) {
20
21
22
               vista.btnCrear.addActionListener(e -> registrarUsuario());
23
24
25 📮
          public void registrarUsuario() {
26
              String dni = vista.txtDni.getText();
               String nombre = vista.txtNombre.getText();
28
               String apellido = vista.txtApellido.getText();
29
30
               String contrasena = new String(vista.txtContrasena.getPassword());
              String repetir = new String(vista.txtRepetirContrasena.getPassword()):
31
32
               // Validaciones
33
              if (dni.isEmpty() || nombre.isEmpty() || apellido.isEmpty() || contrasena.isEmpty() || repetir.isEmpty()) {
    JOptionPane.showMessageDialog(null, "Complete todos los campos");
34
                  return;
36
37
38 =
              if (!contrasena.equals(repetir)) {
                  JOptionPane.showMessageDialog(null, "Las contraseñas no coinciden");
40
41
42
43
               for (ModeloUsuario u : ModeloUsuario.listaUsuarios) {
44
                  if (u.dni.equals(dni)) {
45
                      JOptionPane.showMessageDialog(null, "DNI ya registrado");
46
                       return;
47
48
49
50
               ModeloUsuario nuevo = new ModeloUsuario(dni, nombre, apellido, contrasena);
              ModeloUsuario.listaUsuarios.add(nuevo);
51
52
               JOptionPane.showMessageDialog(null, "Usuario registrado correctamente");
53
55
14
13
           public class Main {
```

```
public class Main {
    public static void main(String[] args) {
        LoginVista vista = new LoginVista();
        vista.setVisible(true);
}
```

```
public class ModeloUsuario {
   public String dni;
   public String nombre;
   public String apellido;
   public String contrasena;

public ModeloUsuario(String dni, String nombre, String apellido, String contrasena) {
      this.dni = dni;
      this.nombre = nombre;
      this.apellido = apellido;
      this.contrasena = contrasena;
   }

   public static java.util.ArrayList<ModeloUsuario> listaUsuarios = new java.util.ArrayList<>();
}
```

```
13
      public class LoginVista extends javax.swing.JFrame {
14
15 🖃
           * Creates new form Login
16
17
18 📮
          public LoginVista() {
           initComponents();
 19
 <u>Q.</u>
              new LoginControlador(this);
 21
22
 23 📮
24
           * This method is called from within the constructor to initialize the form.
           \ensuremath{^{\star}} WARNING: Do NOT modify this code. The content of this method is always
 25
26
           * regenerated by the Form Editor.
27
           @SuppressWarnings("unchecked")
28
29 +
       Generated Code
115
116 📮
           * @param args the command line arguments
117
118
119 🖃
           public static void main(String args[]) {
               /\! Set the Nimbus look and feel */
120
121 🛨
              Look and feel setting code (optional)
142
              //</editor-fold>
143
144
             /st Create and display the form st/
<u>₩</u> 🖨
              java.awt.EventQueue.invokeLater(new Runnable() {
₩ 🛱
                public void run() {
147
                      new LoginVista().setVisible(true);
148
149
              });
150
```

```
* @author marino
   public class Menu extends javax.swing.JFrame {
口
      * Creates new form FMenu
*/
口
      public Menu() {
       initComponents();
         new MenuControlador(this);
戸
      ^{\star} This method is called from within the constructor to initialize the form.
       * WARNING: Do NOT modify this code. The content of this method is always
       ^{\star} regenerated by the Form Editor.
      @SuppressWarnings("unchecked")
+ Generated Code
private void txt_areaActionPerformed(java.awt.event.ActionEvent evt) {
          // TODO add your handling code here:
private void btnCalcularActionPerformed(java.awt.event.ActionEvent evt) {
         // TODO add your handling code here:
private void btnNuevoActionPerformed(java.awt.event.ActionEvent evt) {
          // TODO add your handling code here:
早 /**
```

```
public class MenuAdmin extends javax.swing.JFrame {
口
      * Creates new form MenuAdmin
*/
口
      public MenuAdmin() {
      initComponents();
new MenuAdminControlador(this);
* This method is called from within the constructor to initialize the form.
       * WARNING: Do NOT modify this code. The content of this method is always
       \ast regenerated by the Form Editor.
      */
      @SuppressWarnings("unchecked")
+ Generated Code
private void txt_ladoActionPerformed(java.awt.event.ActionEvent evt) {
          // TODO add your handling code here:
private void btn_calcularActionPerformed(java.awt.event.ActionEvent evt) {
          // TODO add your handling code here:
口
      * @param args the command line arguments */
口
      public static void main(String args[]) {
          /* Set the Nimbus look and feel */
#
          Look and feel setting code (optional)
          /* Create and display the form */
中中
         java.awt.EventQueue.invokeLater(new Runnable() {
             public void run() {
              new MenuAdmin().setVisible(true);
          });
```

```
13
       public class RegistroVista extends javax.swing.JFrame {
14
15 📮
           * Creates new form Registro
16
17
18 🖃
           public RegistroVista() {
           initComponents();
new RegistroControlador(this);
19
 <u>Q</u>
21
22
23 🖃
           * This method is called from within the constructor to initialize the form.

* WARNING: Do NOT modify this code. The content of this method is always
24
25
            * regenerated by the Form Editor.
26
            */
27
28
            @SuppressWarnings("unchecked")
29 🛨
           Generated Code
145
146 📮
           * @param args the command line arguments */
147
148
149 🖃
           public static void main(String args[]) {
150
               /st Set the Nimbus look and feel st/
              Look and feel setting code (optional)
151 🛨
172
               //</editor-fold>
173
174
                /* Create and display the form */
<u>Q</u>
               java.awt.EventQueue.invokeLater(new Runnable() {
<u>Q</u>.↓
                    public void run() {
177
                    new RegistroVista().setVisible(true);
178
179
                });
180
           }
```