

TALLER DE DESARROLLO DE APLICACIONES 1

SEMANA: 3

ENTREGADO POR: Mescua Segovia Marino

CÓDIGO: N04058C

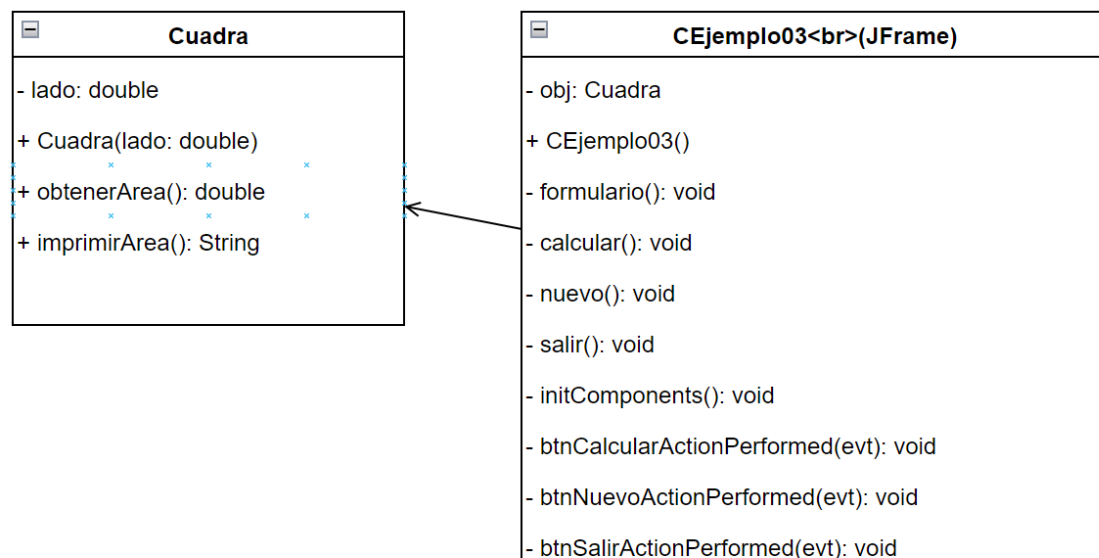
CATEDRÁTICO: Fernández Bejarano Raúl Enrique

EJEMPLO 1:

ENUNCIADO:

Realizar un programa que halle el área de cuadrado, utilizando Programación Orientada a Objetos, Lenguaje Java, el IDE NetBeans y el marco de gestión de proyectos de software Maven

DIAGRAMA DE CLASES:



DISEÑO DE LA APLICACIÓN:

Ingrese el lado:

Area :

CALCULAR **NUEVO** **SALIR**

CÓDIGO DE LA APLICACIÓN:

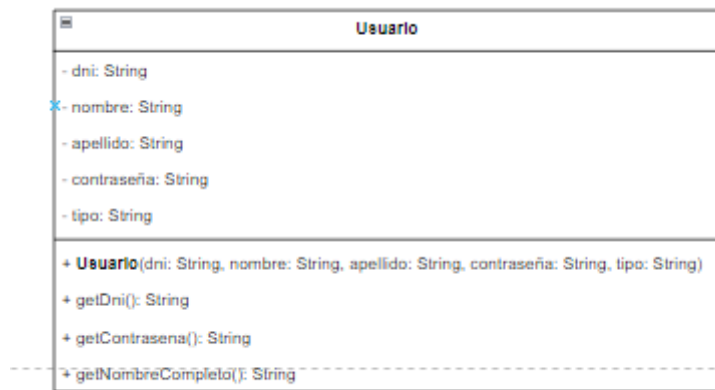
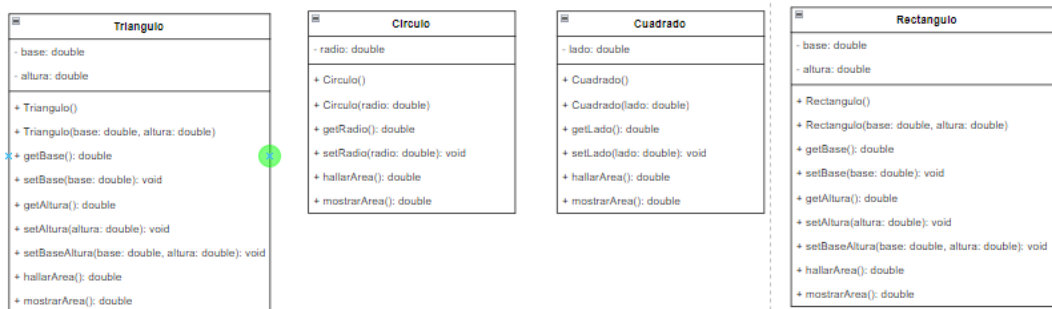
```

18 public Ejemplo3() {
19     initComponents();
20 }
21
22 /**
23  * This method is called from within the constructor to initialize the form.
24  * WARNING: Do NOT modify this code. The content of this method is always
25  * regenerated by the Form Editor.
26  */
27 @SuppressWarnings("unchecked")
28 Generated Code
137
138 private void btn_calcularActionPerformed(java.awt.event.ActionEvent evt) {
139     try {
140         double lado = Double.parseDouble(txt_lado.getText());
141         double area = lado * lado;
142
143         txt_area.setText(String.valueOf(area));
144         txt_lado.setEnabled(false);
145         btn_calcular.setEnabled(false);
146         btn_nuevo.setEnabled(true);
147
148     } catch (NumberFormatException e) {
149         JOptionPane.showMessageDialog(null, "Ingrese un número válido");
150     }
151 }
152
153 private void btn_nuevoActionPerformed(java.awt.event.ActionEvent evt) {
154     txt_lado.setText("");
155     txt_area.setText("");
156     txt_lado.setEnabled(true);
157     btn_calcular.setEnabled(false);
158     btn_nuevo.setEnabled(false);
159 }
160
161 private void btn_nuevoActionPerformed(java.awt.event.ActionEvent evt) {
162     txt_lado.setText("");
163     txt_area.setText("");
164     txt_lado.setEnabled(true);
165     btn_calcular.setEnabled(false);
166     btn_nuevo.setEnabled(false);
167 }
168
169 private void btn_salirActionPerformed(java.awt.event.ActionEvent evt) {
170     System.exit(0);
171 }
172
173 private void txt_areaActionPerformed(java.awt.event.ActionEvent evt) {
174     // TODO add your handling code here:
175 }
176
177 private void txt_ladoKeyReleased(java.awt.event.KeyEvent evt) {
178     if (!txt_lado.getText().isEmpty()) {
179         btn_calcular.setEnabled(true);
180     } else {
181         btn_calcular.setEnabled(false);
182     }
183 }

```

ACTIVIDAD 1:

DIAGRAMA DE CLASES:



DISEÑO DE LA APLICACIÓN:

LOGIN

Usuario :

Contraseña :

REGISTRARSE

ENTRAR

SALIR

CALCULAR SALIR

Ingrese Lado :

Area :

CALCULAR NUEVO

CALCULAR SALIR

Ingrese Lado : Ingrese Base :

Area :

CALCULAR NUEVO

REGISTRO

DNI :

NOMBRE :

APELLIDO :

CONTRASEÑA ;

REPETIR CONTRASEÑA ;

CREAR SALIR

CÓDIGO DE LA APLICACIÓN - OBJETOS:

```
13 public class Usuario {
14
15     String dni;
16     String nombre;
17     String apellido;
18     String contrasena;
19
20     // Constructor
21     public Usuario(String dni, String nombre, String apellido, String contrasena) {
22         this.dni = dni;
23         this.nombre = nombre;
24         this.apellido = apellido;
25         this.contrasena = contrasena;
26     }
27
28     // Lista estática de usuarios
29     public static ArrayList<Usuario> listaDeUsuarios = new ArrayList<>();
30 }
31
```

```
6
7 public class Circulo {
8
9     private double radio;
10
11     // Constructor vacío
12     public Circulo() {
13     }
14
15     // Constructor con parámetro
16     public Circulo(double radio) {
17         this.radio = radio;
18     }
19
20     public double getRadio() {
21         return radio;
22     }
23
24     public void setRadio(double radio) {
25         this.radio = radio;
26     }
27
28     public double hallarArea() {
29         double area = Math.PI * Math.pow(radio, 2);
30         return area;
31     }
32
33     public double mostrarArea() {
34         return hallarArea();
35     }
36 }
37
```

```
7 public class Cuadrado {
8
9     private double lado;
10
11     // Constructor vacío
12     public Cuadrado() {
13     }
14
15     // Constructor con parámetro
16     public Cuadrado(double lado) {
17         this.lado = lado;
18     }
19
20     public double getLado() {
21         return lado;
22     }
23
24     public void setLado(double lado) {
25         this.lado = lado;
26     }
27
28     public double hallarArea() {
29         double area = Math.pow(lado, 2);
30         return area;
31     }
32
33     public double mostrarArea() {
34         return hallarArea();
35     }
36 }
37
```



```
7 public class Rectangulo {
8
9     private double base;
10    private double altura;
11
12    // Constructor vacío
13    public Rectangulo() {
14    }
15
16    // Constructor con parámetros
17    public Rectangulo(double base, double altura) {
18        this.base = base;
19        this.altura = altura;
20    }
21
22    public double getBase() {
23        return base;
24    }
25
26    public void setBase(double base) {
27        this.base = base;
28    }
29
30    public double getAltura() {
31        return altura;
32    }
33
34    public void setAltura(double altura) {
35        this.altura = altura;
36    }
37
38    public void setBaseAltura(double base, double altura) {
39        this.base = base;
40        this.altura = altura;
41    }
42
43    public double hallarArea() {
44        double area = base * altura;
45        return area;
46    }
47
48    public double mostrarArea() {
49        return hallarArea();
50    }
51 }
52
```

```
7 public class Triangulo {
8
9     private double base;
10    private double altura;
11
12    // Constructor vacío
13    public Triangulo() {
14    }
15
16    // Constructor con parámetros
17    public Triangulo(double base, double altura) {
18        this.base = base;
19        this.altura = altura;
20    }
21
22    public double getBase() {
23        return base;
24    }
25
26    public void setBase(double base) {
27        this.base = base;
28    }
29
30    public double getAltura() {
31        return altura;
32    }
33
34    public void setAltura(double altura) {
35        this.altura = altura;
36    }
37
38    public void setBaseAltura(double base, double altura) {
39        this.base = base;
40        this.altura = altura;
41    }
42
43    public double hallarArea() {
44        double area = (base * altura) / 2;
45        return area;
46    }
47
48    public double mostrarArea() {
49        return hallarArea();
50    }
51 }
52
```

CÓDIGO DE LA APLICACIÓN - LOGIN:

```
129 private void btn_entrarActionPerformed(java.awt.event.ActionEvent evt) {
130
131     String textoUsuario = txt_usuario.getText();
132     String textoContrasena = new String(txt_contraseña.getPassword());
133
134     // Validar si es admin
135     if (textoUsuario.equals("admin") && textoContrasena.equals("admin")) {
136         // Abrir MenuAdmin
137         MenuAdmin menuAdmin = new MenuAdmin();
138         menuAdmin.setVisible(true);
139         this.dispose();
140     } else {
141         // Buscar usuario en la lista
142         boolean usuarioEncontrado = false;
143
144         for (Usuario usuario : Usuario.listaDeUsuarios) {
145             if (usuario.dni.equals(textoUsuario) && usuario.contrasena.equals(textoContrasena)) {
146                 // Abrir menú normal
147                 Menu menuUsuario = new Menu();
148                 menuUsuario.setVisible(true);
149                 this.dispose();
150                 usuarioEncontrado = true;
151                 break;
152             }
153         }
154
155         if (!usuarioEncontrado) {
156             JOptionPane.showMessageDialog(null, "Usuario o contraseña incorrectos");
157         }
158     }
159 }
160
161 private void btn_registrarActionPerformed(java.awt.event.ActionEvent evt) {
162     Registrar formulario = new Registrar();
163     formulario.setVisible(true);
164     this.dispose();
165 }
166
167 private void btn_salirActionPerformed(java.awt.event.ActionEvent evt) {
168     System.exit(0);
169 }
170
```

CÓDIGO DE LA APLICACIÓN - REGISTER:

```
private void btnSalirActionPerformed(java.awt.event.ActionEvent evt) {  
    this.dispose();  
}  
  
private void btnCrearActionPerformed(java.awt.event.ActionEvent evt) {  
    String dniIngresado = txtDni.getText();  
    String nombreIngresado = txtNombre.getText();  
    String apellidoIngresado = txtApellido.getText();  
    String contrasenaIngresada = new String(txtContrasena.getPassword());  
    String repetirContrasena = new String(txtRepetirContrasena.getPassword());  
  
    // Validar que los campos no estén vacíos  
    if (dniIngresado.isEmpty() || nombreIngresado.isEmpty() || apellidoIngresado.isEmpty() ||  
        contrasenaIngresada.isEmpty() || repetirContrasena.isEmpty()) {  
        JOptionPane.showMessageDialog(null, "Por favor, complete todos los campos");  
        return;  
    }  
  
    // Validar que las contraseñas coincidan  
    if (!contrasenaIngresada.equals(repetirContrasena)) {  
        JOptionPane.showMessageDialog(null, "Las contraseñas no coinciden");  
        return;  
    }  
  
    // Verificar si el DNI ya existe  
    for (Usuario usuario : Usuario.listaDeUsuarios) {  
        if (usuario.dni.equals(dniIngresado)) {  
            JOptionPane.showMessageDialog(null, "El DNI ya está registrado");  
            return;  
        }  
    }  
  
    // Si todo está correcto, crear el usuario  
    Usuario nuevoUsuario = new Usuario(dniIngresado, nombreIngresado, apellidoIngresado, contrasenaIngresada);  
    Usuario.listaDeUsuarios.add(nuevoUsuario);  
  
    JOptionPane.showMessageDialog(null, "Usuario registrado exitosamente");  
  
    // Volver al login  
    Login login = new Login();  
    login.setVisible(true);  
    this.dispose();  
}
```

CÓDIGO DE LA APLICACIÓN - MENU:

```
18 public Menu() {
19     initComponents();
20     btn_calcular.setEnabled(false);
21 }
22
23 String figuraSeleccionada = "";
24
25 public void seleccionarFigura(String figura) {
26     figuraSeleccionada = figura;
27     txt_lado.setText("");
28     txt_area.setText("");
29     txt_lado.setEnabled(true);
30     btn_calcular.setEnabled(false);
31     btnNuevo.setEnabled(false);
32 }
33
34 /**
35  * This method is called from within the constructor to initialize the form.
36  * WARNING: Do NOT modify this code. The content of this method is always
37  * regenerated by the Form Editor.
38  */
39 @SuppressWarnings("unchecked")
40 Generated Code
41
42 private void txt_ladoKeyTyped(java.awt.event.KeyEvent evt) {
43     btn_calcular.setEnabled(true);
44 }
45
46
47
48
49
50
51 private void btn_calcularActionPerformed(java.awt.event.ActionEvent evt) {
52     try {
53         double valor = Double.parseDouble(txt_lado.getText());
54         double resultado = 0;
55
56         if (figuraSeleccionada.equals("cuadrado")) {
57             resultado = valor * valor;
58         } else if (figuraSeleccionada.equals("circulo")) {
59             resultado = Math.PI * valor * valor;
60         }
61
62         txt_area.setText(String.valueOf(resultado));
63         txt_lado.setEnabled(false);
64         btn_calcular.setEnabled(false);
65         btnNuevo.setEnabled(true);
66
67     } catch (NumberFormatException e) {
68         JOptionPane.showMessageDialog(null, "Ingrese un valor numérico válido");
69     }
70 }
71
72 private void btnNuevoActionPerformed(java.awt.event.ActionEvent evt) {
73     txt_lado.setText("");
74     txt_area.setText("");
75     txt_lado.setEnabled(true);
76     btn_calcular.setEnabled(false);
77     btnNuevo.setEnabled(false);
78 }
```

CÓDIGO DE LA APLICACIÓN – MENU ADMIN:

```
18  //
19  String figuraSeleccionada = "";
20
21  public MenuAdmin() {
22      initComponents();
23      btnCalcular.setEnabled(false);
24  }
25
26  public void seleccionarFigura(String figura) {
27      figuraSeleccionada = figura;
28      txt_area.setText("");
29      txt_lado.setText("");
30      txt_base.setText("");
31      txt_lado.setEnabled(false);
32      txt_base.setEnabled(false);
33      btnCalcular.setEnabled(false);
34      btnNuevo.setEnabled(false);
35
36      switch (figura) {
37          case "cuadrado":
38              txt_lado.setEnabled(true);
39              break;
40          case "triangulo":
41          case "rectangulo":
42              txt_lado.setEnabled(true);
43              txt_base.setEnabled(true);
44              break;
45          case "circulo":
46              txt_lado.setEnabled(true);
47              break;
48      }
49  }
50
51  /**
52   * This method is called from within the constructor to initialize the form.
53   * WARNING: Do NOT modify this code. The content of this method is always
54   * regenerated by the Form Editor.
55   */
56  @SuppressWarnings("unchecked")
57  Generated Code
211
212  private void txt_ladoKeyTyped(java.awt.event.KeyEvent evt) {
213      btnCalcular.setEnabled(true);
214  }
215
216  private void txt_baseKeyTyped(java.awt.event.KeyEvent evt) {
217      btnCalcular.setEnabled(true);
218  }
219
220  private void btnNuevoActionPerformed(java.awt.event.ActionEvent evt) {
221      txt_lado.setText("");
222      txt_base.setText("");
223      txt_area.setText("");
224      btnNuevo.setEnabled(false);
225      btnCalcular.setEnabled(false);
226
227      if (figuraSeleccionada.equals("cuadrado") || figuraSeleccionada.equals("circulo")) {
228          txt_lado.setEnabled(true);
229      } else {
230          txt_lado.setEnabled(true);
231          txt_base.setEnabled(true);
232      }
233  }
234
```

```

private void btnCalcularActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        double resultado = 0;

        double lado = Double.parseDouble(txt_lado.getText());
        double base = txt_base.isEnabled() ? Double.parseDouble(txt_base.getText()) : 0;

        switch (figuraSeleccionada) {
            case "cuadrado":
                resultado = lado * lado;
                break;
            case "rectangulo":
                resultado = lado * base;
                break;
            case "triangulo":
                resultado = (lado * base) / 2;
                break;
            case "circulo":
                resultado = Math.PI * lado * lado;
                break;
        }

        txt_area.setText(String.valueOf(resultado));
        txt_lado.setEnabled(false);
        txt_base.setEnabled(false);
        btnCalcular.setEnabled(false);
        btnNuevo.setEnabled(true);

    } catch (NumberFormatException e) {
        JOptionPane.showMessageDialog(null, "Ingresa valores válidos");
    }
}

private void ItemCuadradoActionPerformed(java.awt.event.ActionEvent evt) {
    seleccionarFigura("cuadrado");
}

private void itemTrianguloActionPerformed(java.awt.event.ActionEvent evt) {
    seleccionarFigura("rectangulo");
}

private void itemRectanguloActionPerformed(java.awt.event.ActionEvent evt) {
    seleccionarFigura("triangulo");
}

private void itemCirculoActionPerformed(java.awt.event.ActionEvent evt) {
    seleccionarFigura("circulo");
}

```