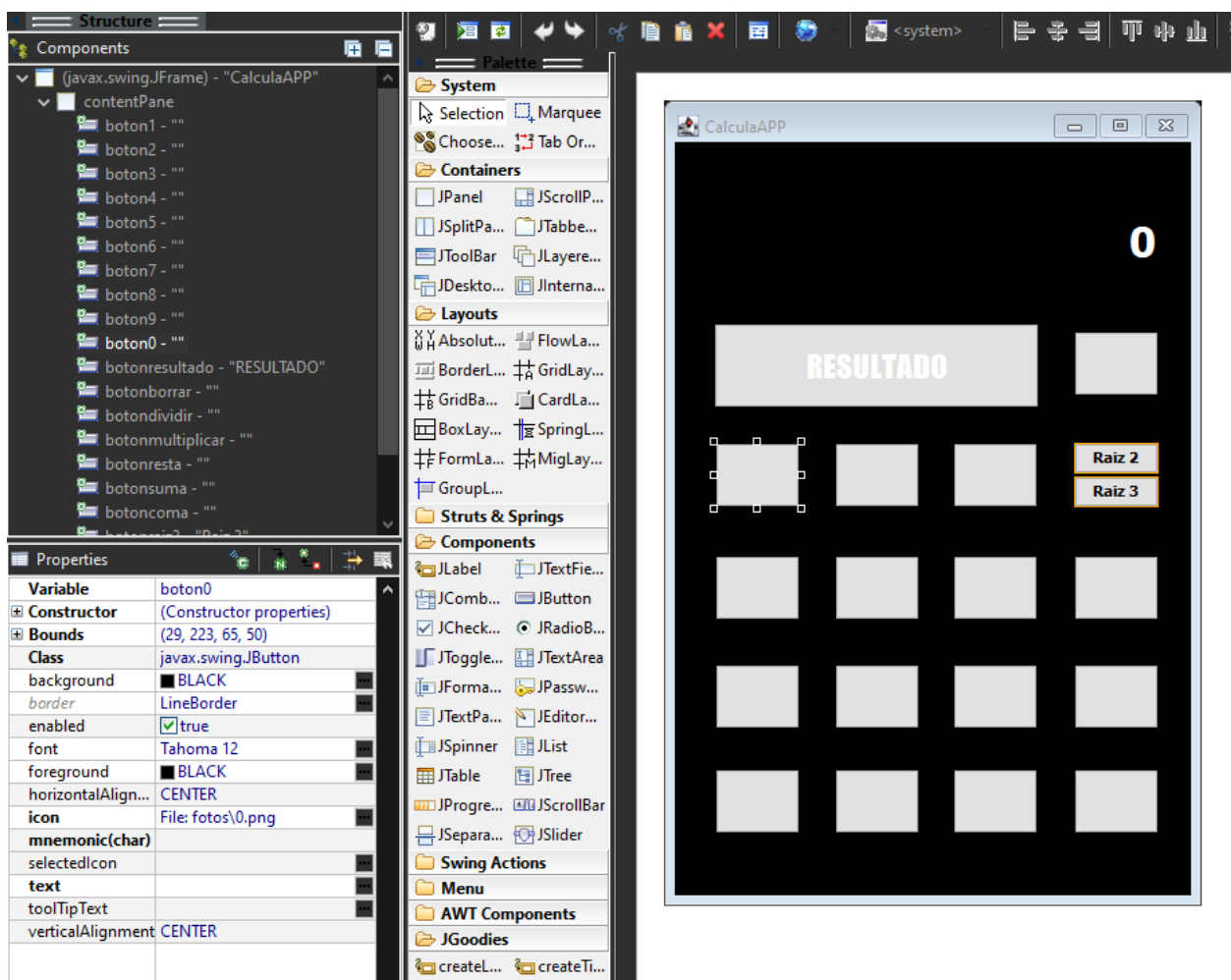


Actividad 2- Calculadora

Crea una calculadora básica en Java

Creamos nuestro proyecto que consta de un Main, una VentanaPrincipal y una VentanaLogin.

Comenzamos creando la clase VentanaPrincipal con el plugin WindowBuilder. Se crea una clase que extiende de JFrame.



Damos forma a la GUI, creamos los diferentes JButton que respesentan los dígitos del 0 al 9 y las operaciones +-*/, además de Raíz 2 que deberá sacar una ventana modal de error y Raíz 3 que deberá lanzar una ventana de login para poder utilizar su

funcionalidad. Por último, la pantalla de la calculadora es un JTextField. Utilizando la página :

Material Design Icons

Material Design Icons' growing icon collection allows designers and developers targeting various platforms to download icons in the format, color and size they need for any project. Loading... Sponsored by Icons8 Material Icon Sets

 <https://materialdesignicons.com/>

Descargamos, damos color y tamaño a los diferentes iconos de la calculadora. Una vez hecho implementamos en cada JButton su imagen correspondiente. Ejemplo del botón 1:

```
//BOTON 1
//CREAMOS EL OBJETO JButton
JButton boton1 = new JButton("");
//LE DAMOS COLOR NEGRO DE BACKGROUND
boton1.setBackground(new Color(0, 0, 0));
//LE PONEMOS EL BORDE NEGRO PARA QUITARLO
boton1.setBorder(new LineBorder(Color.BLACK));
//IMPLEMENTAMOS LA IMAGEN QUE HARÁ DE BOTÓN
boton1.setIcon(new ImageIcon("fotos\\1.png"));
//COLOR BLANCO DE LA LETRA
boton1.setForeground(new Color(0, 0, 0));
//TIPO DE LETRA Y TAMAÑO (NO ES NECESARIO AL USAR UNA IMAGEN
boton1.setFont(new Font("Tahoma", Font.PLAIN, 12));
//LE DAMOS UNA POSICIÓN Y TAMAÑO AL BOTÓN
boton1.setBounds(29, 466, 65, 50);
//SE AÑADE EL BOTÓN
contentPane.add(boton1);
```

Resultado:



Le damos diseño a todos los botones. Y a continuación le damos funcionalidad a cada uno:

```
//BOTON DIGITO 1
//SE CREA ActionListener
boton1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        //UN IF QUE CONTROLA SI HAY ALGO ESCRITO O NO
        //SI NO LO HAY PONE UN 1
        if (panelresultados.getText().equals("0")) {
            panelresultados.setText("1");
        }else {
            //SI YA HAY ALGO AÑADE EL 1 + LO QUE YA ESTÉ
        }
    }
});
```

```

        panelresultados.setText(panelresultados.getText() + "1");

    }
}
});

```

Todos los dígitos son iguales exceptuando que cada uno aplica su número.

Continuamos con la funcionalidad de las operaciones. Tenemos los siguientes atributos de clase:

```

private JPanel contentPane;
private JTextField panelresultados;
//GUARDA EL RESULTADO DE CADA OPERACIÓN
private double resultado;
//GUARDA QUÉ OPERACIÓN SE HA HECHO
private String operador;
//CONTROLA LA CANTIDAD DE DECIMALES (4)
private DecimalFormat formato = new DecimalFormat("#.####");

```

```

//LA SUMA SE LE AÑADE UN ACTIONLISTENER
botonsuma.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        //AL SER UNA SUMA EL ATRIBUTO OPERADOR RECIBE UN "+"
        operador = "+";
        //RESULTADO RECOGE PARSEADO A DOUBLE LO QUE ESTÉ ESCRITO
        //EN LA PANTALLA
        resultado = Double.parseDouble(panelresultados.getText());
        //BORRA LA PANTALLA PARA RECIBIR LIMPIO EL SIGUIENTE DIGITO
        panelresultados.setText("");

    }
});

```

Todos los operadores son iguales cambiando lo que añaden al atributo operador.

Los diferentes son:

```

//PUNTO PARA DECIMALES
botoncoma.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        //SI LA PANTALLA CONTIENE UN PUNTO YA NO PONDRÁ MÁS
        if(!panelresultados.getText().contains(".")) {
            //SI CONTIENE UN CERO PONDRÁ .
            if (panelresultados.getText().equals("0")) {

```

```

        panelresultados.setText(".");
        //SI NO PONE LO QUE HAYA + .
    } else {
        panelresultados.setText(panelresultados.getText() + ".");
    }
}
}
});

```

```

//BORRAR
botonborrar.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        //REINICIA LAS VARIABLES Y BORRA LA PANTALLA
        operador = "";
        resultado = 0;
        panelresultados.setText("");

    }
});

```

```

//SACA UN AVISO DE NO DISPONIBILIDAD
botonraiz2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        //SACA UNA VENTANA MODAL AL PULSAR EL BOTON
        JOptionPane.showMessageDialog(botonraiz2, "Funcionalidad no disponible");

    }
});

```

```

//LLAMA A UN MÉTODO QUE CALCULA EL RESULTADO
botonresultado.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        calcularResultado();

    }
});

```

```

private void calcularResultado() {
    try {
        //EL METODO COMPARA LO QUE CONTIENE LA VARIABLE OPERADOR
        //QUE HA SIDO CAMBIADA EN FUNCION DEL BOTON DE CADA OPERACION
        if (operador.equals("+")) {
            //SUMA RESULTADO A LO QUE HAYA EN ESTE MOMENTO EN PANTALLA

```

```

        //ES DECIR, EL SEGUNDO DIGITO
        resultado += new Double(panelresultados.getText());
    } else if (operador.equals("-")) {
        resultado -= new Double(panelresultados.getText());
    } else if (operador.equals("/")) {
        resultado /= new Double(panelresultados.getText());
    } else if (operador.equals("*")) {
        resultado *= new Double(panelresultados.getText());
    }
    //UNA VEZ PASA POR LA CONDICIÓN
    //PANTALLA MUESTRA EL RESULTADO LIMITADO A 4 DECIMALES
    panelresultados.setText("" + formato.format(resultado));
    //REINICIA EL OPERADOR PARA PODER CONTINUAR OPERANDO
    operador = "";
}
//TRY-CATCH PARA CONTROLAR QUE ALGUIEN INTENTE OPERAR
//SIN HABER PUESTO NUMEROS
catch (Exception e1) {
    JOptionPane.showMessageDialog(panelresultados, "Introduce un número");
}
}
}

```

Por último, tenemos el botón Raíz 3 que lanza una ventana de login:

```

//ABRE OTRA VENTANA DE LA CLASE VentanaLogin en la que introducir una contraseña
//Una vez acertada se podrá utilizar la operación
botonraiz3.addActionListener(new ActionListener(){

    public void actionPerformed(ActionEvent e) {
        // CREAMOS UN OBJETO VENTANALOGIN QUE RECIBE COMO PARAMETRO
        //LA PANTALLA DE LA CALCULADORA. ESTO NOS SIRVE PARA QUE PUEDA
        //OPERAR CON LO QUE HAYA EN PANTALLA
        VentanaLogin ventanaLogin = new VentanaLogin(panelresultados);
    }
});
}

```

Creamos una clase VentanaLogin que extiende de JDialog. Usando el WindowBuilder le damos forma y estilo.

```

public class VentanaLogin extends JDialog {

    private final JPanel contentPanel = new JPanel();
    private JPasswordField password;
    private JButton botonOK, botonCancelar;
    private JTextField pantallaResultado;
}

```

```

DecimalFormat formato = new DecimalFormat("#.####");

//EL CONSTRUCTOR RECIBE LA PANTALLA DE VENTANAPRINCIPAL
public VentanaLogin(JTextField num1) {
    this.pantallaResultado = num1;

    setBounds(100, 100, 450, 300);
    getContentPane().setLayout(new BorderLayout());
    contentPanel.setForeground(new Color(255, 255, 255));
    contentPanel.setBackground(new Color(0, 0, 0));
    contentPanel.setBorder(new EmptyBorder(5, 5, 5, 5));
    getContentPane().add(contentPanel, BorderLayout.CENTER);
    contentPanel.setLayout(null);
    setTitle("CalculaAPP");
    setResizable(false);
    inicializarComponentes();
    setVisible(true);
}

```

El resultado es el siguiente:



A continuación inicializamos los componentes y le damos funcionalidad:

```

//INICIALIZAMOS
private void inicializarComponentes() {
    //CAMPO DE LA CONTRASEÑA
    password = new JPasswordField();
    password.setBounds(80, 82, 268, 62);
    contentPanel.add(password);
}

```

```

//BOTON CANCELAR
botonCancelar = new JButton("BORRAR");
botonCancelar.setForeground(new Color(255, 140, 0));
botonCancelar.setFont(new Font("Tahoma", Font.BOLD, 13));
botonCancelar.setBounds(223, 155, 125, 23);
contentPanel.add(botonCancelar);

//BOTON OK PARA VALIDAR PASSWORD
botonOK = new JButton("INTRODUCIR");
botonOK.setFont(new Font("Tahoma", Font.BOLD, 13));
botonOK.setForeground(new Color(255, 140, 0));
botonOK.setBounds(80, 155, 125, 23);
contentPanel.add(botonOK);

```

```

//FUNCIONALIDAD BOTON OK
botonOK.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        //Guardamos en un String el valor de la contraseña introducida
        String contrasena = String.valueOf(password.getPassword());

        // Establecemos que la contraseña correcta sea avengers. En caso de serlo,
        //se dará paso a la operación
        if(contrasena.equals("avengers")) {

            try {
                // Esta ventana será la que realice la operación con el número que
                //le ha llegado por pantalla desde la ventana principal
                // Convertimos a Double el valor en string de la caja de texto del número
                double numero = Double.parseDouble(pantallaResultado.getText());
                // Calculamos la raíz cuadrada
                double raízCuadrada = Math.cbrt(numero);

                // Asignamos a la JLabel que contendrá el resultado del valor
                // de la operación, que saldrá en la etiqueta de la VentanaPrincipal
                pantallaResultado.setText("Resultado: " + formato.format(raízCuadrada));

                // Hacemos que la ventana desaparezca (se cierra) para dar
                //paso a la operación de la calculadora
                dispose();

            }
            //UN TRY-CATCH QUE CONTROLA QUE NO SE HAYA INTRODUCIDO NINGUN NUMERO
            //EN PANTALLA Y SE HAYA PULSADO RAIZ3
            catch (Exception e1) {
                JOptionPane.showMessageDialog(botonOK, "Introduce un número");
                dispose();
            }

        } else {

```



```

        setTitle("Contraseña incorrecta, vuelva a introducirla:");

        password.setText("");
        password.requestFocus();

    }
}
});

```

```

//FUNCIONALIDAD BOTON CANCELAR
botonCancelar.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        password.setText("");
        password.requestFocus();

    }
});

```

Una vez terminadas las ventanas y que estas sean funcionales creamos el Main:

```

public class Main {

    public static void main(String[] args) {
        VentanaPrincipal calculadora = new VentanaPrincipal();
        calculadora.setVisible(true);

    }

}

```