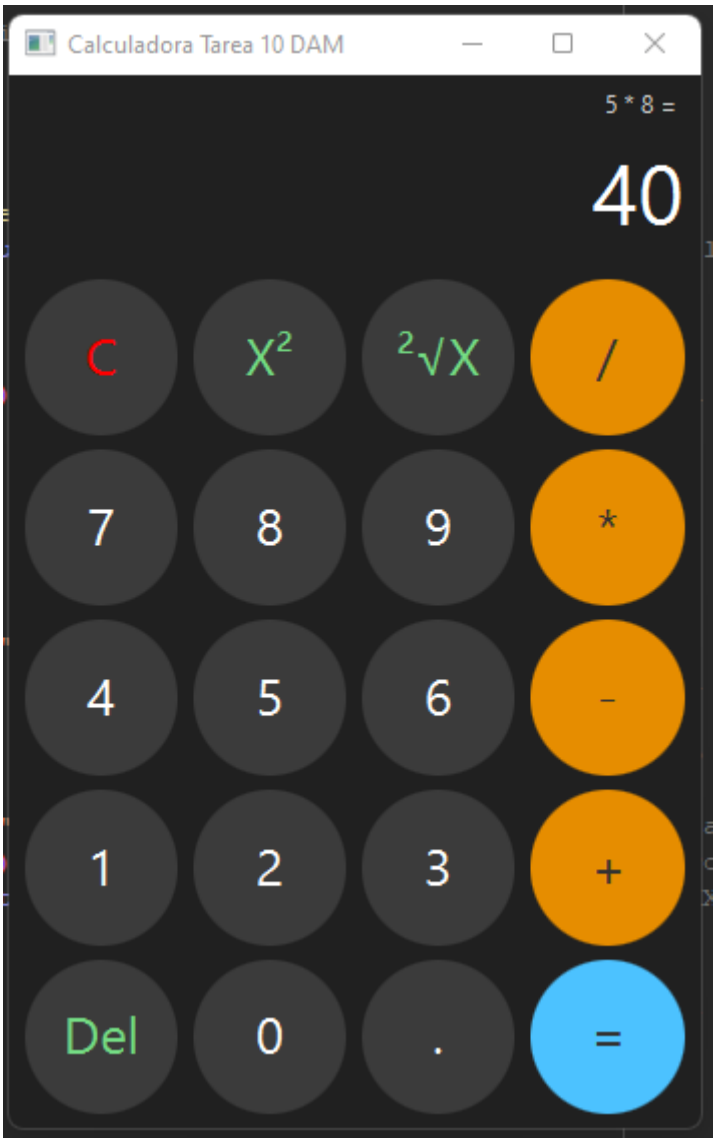
Informe de la PROG10

Con los conocimientos adquiridos durante la unidad se propone la implementación de una calculadora, parecida a la que se implementa en uno de los vídeos referenciados en los contenidos, pero utilizando la librería JavaFX. El nombre de la aplicación será PROG10_Tarea. Ten en cuenta que:

- La calculadora podrá realizar solo operaciones sumar, restar, multiplicacion y división de números enteros.
- Será obligatorio como mínimo realizar operaciones con dos operandos, valorándose positivamente la posibilidad de realizar operaciones con mas operandos.



Todos los botones de los números 0,1,2,3,4,5,6,7,8,9 y el punto están vinculados el evento On Action con el método de clickBtnNum

Que es la que utilizo para crear los números. Una funcionalidad que he he añadido es que no solo sean números enteros sino que también números decimales.

```
@FXML
private void clickBtnNum(ActionEvent event) {
    Button btn = (Button) event.getSource(); //En la variable btn copiamos todos los datos del botón que ha generado el evento
    String aux = getPantalla();

    if (this.checkOperacion){
        if (btn.getText().equals(".")) // esto es para evitar que solo muestre un punto
            aux = "0.";
        else
            aux = btn.getText();
        checkOperacion = false;
    } else {
        if (aux.equals("0"))                    //Si en pantalla hay un 0
            if (btn.getText().equals("."))      //Si se ha pulsado el punto
                aux += btn.getText();           //Pues añadimos ese punto y queda así -> 0.
            else
                aux = btn.getText();            //Cambia el 0 por el numero nuevo del 1 al 9
        else
            if (btn.getText().equals(".")) {    //Si hay un solo "0" y se ha pulsado un punto
                if (!aux.contains("."))         //Si no se tiene un punto ya introducido
                    aux += btn.getText();       //Añade a que haya un punto -> XXX.
            }
            else
                aux += btn.getText();           //Añade cualquier numero del 0 al 9
    }
    setPantalla(aux);
}
```

Los botones de sumar, restar, multiplicar, dividir y el igual están vinculado con el método btnOperador.

En dicho método no solo gestiono si tiene que realizar las operaciones indicadas entre dos operandos, sino que ademas se le añade la funcionalidad de ser con más operandos.

Otra funcionalidad que añado o control de errores es controlar la división por cero.

```
@FXML
private void btnOperador(ActionEvent event) {

    Button btn = (Button) event.getSource();  //Obtenemos el objeto que ha generado el evento para saber que operación vamos a realizar

    BigDecimal entradaOperador = new BigDecimal(getPantalla());      //Leemos el digito que tenemos en pantalla

    if (this.operador == null) {            //Si es NULL el atributo donde lo almacenamos el primer operador
        this.operador = entradaOperador;    //pues le iniciamos en el valor de pantalla
    } else if (this.operacion != null) {    //si hay una operación en marcha
        switch (this.operacion){            //pregunta cul es la operación y en función de la que sea la realiza.
            case "-" -> this.operador = this.operador.add(entradaOperador);
            case "-" -> this.operador = this.operador.subtract(entradaOperador);
            case "*" -> this.operador = this.operador.multiply(entradaOperador);
            case "/" -> {
                try {
                    this.operador = this.operador.divide(entradaOperador,MathContext.DECIMAL32);  //MathContext es la precisión del coma flotante
                } catch (ArithmeticException ex) {      //Controlamos la división entre cero
                    Alert alert = new Alert(Alert.AlertType.ERROR);
                    alert.setHeaderText(null);
                    alert.setTitle("Error" );
                    alert.setContentText("No se puede dividir entre cero.\n" + ex.getMessage());
                    alert.showAndWait();
                }
            }
        }
        setPantalla(this.operador.toString());  //Muestra el resultado en pantalla.
    }
    if (btn.getText().equals("=")) { //Si la tecla pulsada es la de = pues inicia todo, sin borrar la pantalla
       // this.operacion = null;
        this.operador = null;
        lbOperacion.setText(lbOperacion.getText() + entradaOperador.toString() + " = ");  //Operación completa realiza
    } else {
        this.operacion = btn.getText();    //almacena la operación.
        lbOperacion.setText(this.operador.toString() + " " + this.operacion + " ");  //Muestra cual es la operación
    }

    this.checkOperacion =true;  //la activa para cambiar el valor de pantalla

}
```

El botón de borrar “C” lo que hace es reiniciar todos los atributos y objetos que intervienen en una operación.

```
 */
@FXML
private void ClickBtnReset(ActionEvent event) {
    setPantalla("0"); //Restablece la pantalla a cero
    lbOperacion.setText("");
    this.checkOperacion = false;
    this.operador = null;
    this.operacion = null;
}
```

Enl botón DEL es para borrar el último carácter que tenga el digito en pantalla y si se borran todos pone el cero.

```
 */
@FXML
private void ClickBtnBorrar(ActionEvent event) {
    String aux = getPantalla().substring(0,getPantalla().length()-1);

    if (aux.length()<=0)
        setPantalla("0");
    else
        setPantalla(aux);

}
```

El botón de Raíz Cuadrada pues realiza la raíz cuadrada de numero en pantalla y controla el error de hacer una raíz a un número negativo.

```
@FXML
private void ClickBtnRaiz(ActionEvent event) {
    BigDecimal valor = new BigDecimal(getPantalla());

    try {
       setPantalla(valor.sqrt(MathContext.DECIMAL32).toString());
    } catch (Exception ex) {
       Alert alert = new Alert(Alert.AlertType.ERROR);
       alert.setHeaderText(null);
       alert.setTitle("Error" );
       alert.setContentText("Intento de raiz cuadrada de un número negativo\n" + ex.getMessage());
       alert.showAndWait();
    }
}
```

Otra de las funcionalidades que he añadido es la de calcular el cuadrado de un número

```
/**
 * Método ClickBtnPotencia
 * Calcula el cuadrado de un número
 * @param event
 */
@FXML
private void ClickBtnPotencia(ActionEvent event) {
    BigDecimal valor = new BigDecimal(getPantalla());

    setPantalla(valor.pow(2).toString());
}
```

Y aunque no sea funcionalidad he incorporado en la parte superior al igual que hace la calculadora de Windows que muestre la operación que se está realizando. Esto lo realizo en el método btnOperador.

```
if (btn.getText().equals("=")) { //Si la tecla pulsada es la de = pues inicia todo, sin borrar la pantalla
   // this.operacion = null;
    this.operador = null;
    lbOperacion.setText(lbOperacion.getText() + entradaOperador.toString() + " = ");  //Operación completa realiza
} else {
    this.operacion = btn.getText();    //almacena la operación.
    lbOperacion.setText(this.operador.toString() + " " + this.operacion + " ");  //Muestra cual es la operación
}
```

Para dar formas y colores a los botones de la calculadora he utilizado un fichero CSS, dicho fichero contiene 3 clases (Definición de clase CSS) que son para los diferentes tipos de botones.

No solo para cambiar de formas sino que también he añadido el electo “hover” es decir que al pasar con el ratón cambie de color y lo mismo al hacer click.

```
/*---Botones---*/
.botonesNegros {
    -fx-background-color: #3b3b3b;
    -fx-background-radius: 100px;
}
.botonesNegros:hover {
    -fx-background-color: #606060;
    -fx-background-radius: 100px;
}
.botonesNegros:pressed {
    -fx-background-color: #909090;
    -fx-background-radius: 100px;
}

/*---Botones de las Operaciones---*/
.botonesNaranjas {
    -fx-background-color: #e68d00;
    -fx-background-radius: 100px;
}
.botonesNaranjas:hover {
    -fx-background-color:  #d89e41;
    -fx-background-radius: 100px;
}
```

```
.botonesNaranjas:pressed {
    -fx-background-color:  #dbaf69;
    -fx-background-radius: 100px;
}

/*---Botón del igual---*/
.botonesAzules {
    -fx-background-color:  #4cc2ff;
    -fx-background-radius: 100px;
}
.botonesAzules:hover {
    -fx-background-color:  #0087cc;
    -fx-background-radius: 100px;
}
.botonesAzules:pressed {
    -fx-background-color:  #5da4c8;
    -fx-background-radius: 100px;
}
```