

Ejercicio

Ejercicio3. Calculadora

Botones de memoria

MC (Memory Clear): Elimina cualquier número almacenado en memoria.

MR (Memory Recall): Recupera el número almacenado en memoria. El número permanece en memoria.

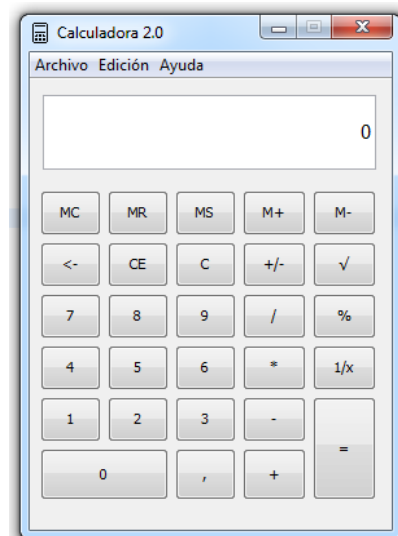
MS (Memory Storage): Almacena en memoria el número mostrado.

M+: Suma el número mostrado a otro número que se encuentre en memoria pero no muestra la suma de estos números.

M-: Resta el número mostrado a otro número que se encuentre en memoria pero no muestra la resta de estos números.

CE (Clear error): Elimina el número mostrado. Se utiliza para cuando se comete un error en el ingreso de datos pero sin eliminar todo el cálculo anterior.

C (Clear): Elimina todo el cálculo actual.



Programa Principal

@Override

```
public void start(Stage primaryStage) {  
    try {  
        String fxml = "Calculadora.fxml";  
  
        // Cargar la ventana  
        Parent root = FXMLLoader.load(getClass().getResource(fxml));  
  
        // Cargar la Scene  
        Scene scene = new Scene(root);  
  
        // Asignar propiedades al Stage  
        primaryStage.setTitle("Calculadora 2.0");  
        primaryStage.setResizable(false);  
  
        // Asignar la scene y mostrar  
        primaryStage.setScene(scene);  
        primaryStage.show();  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

Clase Calculadora

```
//Atributos
private double num1;
private double num2;

private String operacion;
private String numActual;
private String numMemoria;

//Constructor
public Calculadora() {...}

//Getters and Setters

//Operaciones aritméticas
public double sumar(){...}
public double restar(){...}
public double multiplicar(){...}
public double dividir() throws DivisionPorCeroExcepcion{...}
public void porcentaje(){...}
public double raiz() throws RaizNegativaExcepcion{...}
public double inversa() throws DivisionPorCeroExcepcion{...}

//Cambiar signo al valor actual
public void cambiarSigno(){...}

//Resetear todas las variables
public void reset(){...}

//Concatenar número al numActual
public String concatenar(String numero){...}
public void retroceder(){...}

//Concatenar número al numActual
public static String concatenar(String numero){...}
public void retroceder(){...}

//etc...
```

Controlador Calculadora

```
public class CalculadoraController implements Initializable {

    // Atributos graficos FXML
    @FXML private Button sumar;
    @FXML private Button restar;
    // ...
    @FXML private Button boton_0;
    // ...

    @Override
    public void initialize(URL url, ResourceBundle resourceBundle) {
        // Evento botones, apertura de ventanas
        sumar.setOnMouseClicked((event) -> asignarOperacion("+"));
        boton_0.setOnMouseClicked((event) -> pulsarNumero(0));
    }

    //*****
    // Métodos de la clase CalculadoraController
    //*****
    private void asignarOperacion(String operacion){...}
    private void pulsarNumero(String numero){...}
    private void retroceder(){...}
    private void cambiarSigno(){...}
    private void calcular() {...}
    private void raiz() {...}
    private void inversa() {...}
    private void clearError() {...}
    private void clear() {...}
    private void refrescarTextoResultado() {...}

    //*****
    // BOTONES DE MEMORIA
    //*****
    private void memoryClear() { calculadora.memoryClear();}
    private void memoryStorage() { calculadora.memoryStorage();}
    private void memoryRecall() {
        calculadora.memoryRecall();
        mostrarInfo();
    }
    private void memorySumar() { calculadora.memorySumar(); }
    private void memoryRestar() { calculadora.memoryRestar(); }
}
```


Menú ejemplo

```
<MenuBar layoutX="0.0" layoutY="0.0" prefHeight="25.0" prefWidth="250.0">
  <menus>
    <Menu mnemonicParsing="false" text="Ver">
      <items>
        <MenuItem fx:id="menu_Historial" mnemonicParsing="false" text="Historial" />
        <MenuItem fx:id="menu_Salir" mnemonicParsing="false" text="Salir" />
      </items>
    </Menu>
    <Menu mnemonicParsing="false" text="Edición">
      <items>
        <MenuItem fx:id="menu_Copiar" mnemonicParsing="false" text="Copiar" />
        <MenuItem fx:id="menu_Pegar" mnemonicParsing="false" text="Pegar" />
      </items>
    </Menu>
    <Menu mnemonicParsing="false" text="Ayuda">
      <items>
        <MenuItem fx:id="menu_Ayuda" mnemonicParsing="false" text="Ver la Ayuda" />
        <MenuItem fx:id="menu_AcercaDe" mnemonicParsing="false" text="Acerca de
Calculadora" />
      </items>
    </Menu>
  </menus>
</MenuBar>
```

//Mostrar otra ventana

```
private void mostrarVentanaAyuda(String rutaFXML, String titulo) {

    try{

        FXMLLoader fxmlLoader = new FXMLLoader(getClass().getResource(rutaFXML));
        Parent root = (Parent) fxmlLoader.load();

        Stage stage = new Stage();

        //Asignar al Stage la escena
        stage.setTitle(titulo);
        stage.setResizable(false);
        stage.initModality(Modality.APPLICATION_MODAL);
        stage.setScene(new Scene(root));

        //Mostrar el Stage (ventana)
        stage.show();

    }
    catch (Exception e){
        e.printStackTrace();
    }
}
```

Excepciones

Crear al menos dos excepciones

[DivisionPorCeroException.java](#)

[RaizNegativaException.java](#)

Estilos CSS

Main.java

```
@Override
public void start(Stage primaryStage) {

    ...

    // Asignar hoja de estilos
    scene.getStylesheets().add("/vista/css/estilos.css");
}
```

Estilos.css

```
.button {
    -fx-background-color: #FBFCFC;
    -fx-text-fill: #979A9A;
}
```