Proyecto Calculadora: Estructura de Datos I

Primavera 2024

Maestra: Silvia Guardati Buemo

Instituto Tecnológico Autónomo de México

Luciano Moctezuma: 209116

Alejandro Salmón Meehan: 213252

Lorenzo Pazos García: 212858

Sebastián Velasco Galindo: 213592

Luca Boschetti: 211222

Índice:

1. Descripción

1.1 Objetivo:

El objetivo del proyecto es construir un programa de calculadora que reúna las características de un software de calidad, el cual sea claro y fácil de utilizar. Se aplicarán pilas y arreglos, estructuras de datos necesarias para el desarrollo de los métodos y funciones necesarias para obtener el resultado esperado. Se pondrá en práctica estas estructuras de datos para familiarizarse con la forma de aplicación para proyectos y tareas

futuras. Además, se pondrán a prueba y trabajarán las habilidades de comunicación y trabajo en equipo.

1.2 Requisitos:

La calculadora debe tener una GUI (Graphic User Interface); la cual adopte la funcionalidad de dos clases: La clase de revisores y la clase de calculadora. La calculadora debe hacer operaciones con números enteros, positivos y negativos; con números con punto decimal y seguir la jerarquía de operaciones para regresar un resultado correcto.

2. Restricciones:

La calculadora a se debe desarrollar en NetBeans, y debe ser compartido a través de GitHub, lugar donde deben subirse constantemente avances de las clases para implementar la funcionalidad a la interfaz gráfica, reutilizar métodos en los algoritmos necesarios y realizar las pruebas correspondientes.

3. Solución Diseñada

3.1 Clases UML:

Revisores
- operaciones: String
+ Revisores() + Revisores(operaciones: String) + setOperaciones(operaciones: String): void + getOperaciones(): String - esOperador(ch: Character): boolean - esParentesis(ch: Character): boolean + revParentesisVacios(): boolean + revisadorPuntos(): boolean + revisadorSignos(): boolean + revisadorParentesis(): boolean + revisadorDivisionCero(): boolean + revSintaxis(): boolean
PilaADT<T>
+ push(dato: T): void + pop(): T + isEmpty(): boolean + peek(): T

PilaA<T>
- pila: T[] - tope: int - MAX: int
+ PilaA() + PilaA(max: int) + push(dato: T): void + expande(): void + pop(): T + isEmpty(): boolean + peek(): T + toString(): String

ExcepcionColeccionVacia
+ ExcepcionColeccionVacia(message: String)

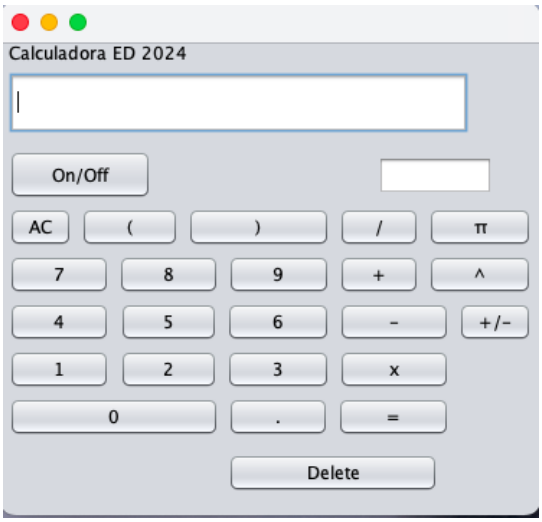
GUICalculadora
- prendido: boolean
+ GUICalculadora() + Prendido(): boolean + Apagado(): boolean + resultado(String cadena): String - initComponents() - Pantalla: JTextField - BotonPrendeApaga: JButton - BotonReset: JButton - AbreParentesis: JButton - CierraParentesis: JButton - Boton0: JButton - Boton1: JButton - Boton2: JButton - Boton3: JButton - Boton4: JButton - Boton5: JButton - Boton6: JButton - Boton7: JButton - Boton8: JButton - Boton9: JButton - BotonSuma: JButton - BotonResta: JButton - BotonMultiplicacion: JButton

- BotonDivision: JButton - BotonExponente: JButton - BotonPi: JButton - BotonNegativo: JButton - BotonPuntoDecimal: JButton - BotonResultado: JButton - BotonBack: JButton - OnOff: JTextField
Calculadora
+ Calculadora() + prioridades(c: char): int + creadorPosfija(cadena: String): ArrayList<String> + evaluaPosFija(a: ArrayList): double

3.2 Algoritmos Principales:

Nuestros métodos principales son creadorPosfija y evaluaPosFija. El se encarga de recibir un string con el problema matemático escrito por el usuario y convertirlo en un arraylist donde venga en orden cada número y símbolo de la operación matemática. El segundo se encarga de evaluar las operaciones según su orden en el arraylist regresado por el método mencionado anteriormente, así obteniendo el resultado final de la operación escrita por el usuario. Estos son los algoritmos más importantes, ya que en ellos se encuentra el funcionamiento de la calculadora que se ve representado visualmente en el GUI.

3.3 Pantalla:



4. <u>Pruebas diseñadas:</u>

Pruebas para el lector pos-fija: En esta prueba se utilizó un método estático que recibía un ArrayList ya acomodado en la forma pos-fija, ese mismo método se copió y utilizó en

la clase calculadora. La prueba consistía en escribir los métodos y que realizará las operaciones matemáticas de forma correcta, el ArrayList era proporcionado ya considerando la jerarquía de operaciones, la cual se ordena en el método creador pos-fija.

Pruebas para el GUI: Para lograr que el GUI tenga un diseño tanto agradable para el usuario como efectivo para realizar las tareas deseadas, seguimos una serie de pasos. Primero anotamos que botones queríamos tener en esta, después diseñamos el GUI de manera manual agregando botones y TextBoxes principalmente. Después de tener la interfaz diseñada y con un aspecto entendible y fácil de usar, comenzamos a darle funcionalidad a cada botón, la mayoría siendo solo agregar dicho número/signo.

- Botón Apagado/Prendido: Hemos decidido agregar un botón de apagado y prendido, que activa y desactiva los botones como si la calculadora estuviera apagada o prendida.
- Botón Resultado: Para el botón de resultado fue para el que más pruebas necesitamos, diseñamos una funcionalidad que organizara los métodos y los corriera de forma apropiada, para esto necesitamos varias pruebas, especialmente para correr y probar si el revisor de errores servía correctamente.

Pruebas de los Revisores:

Para determinar e identificar cada revisor de posibles errores se desarrollaron algoritmos separados, los cuales se utilizaron en el metodo final para que checara cada error por separado y regresara un mismo resultado. En una clase con main se corrió cada método individualmente y por último uno que englobara todos los algoritmos.

5. Limitaciones de la solución:

Encontramos bastantes errores después de terminar el proyecto, lo que nos hizo cambiar nuestros métodos ligeramente hasta que funcionara correctamente...

6. Posibles Mejoras y conclusiones:

La mayor restricción que encontramos fue el uso de GitHub, a pesar de poder subir y descargar nuestro proyecto para la división de tareas, es algo que nos costó un buen porcentaje del tiempo. El trabajo en equipo se llevó a cabo de manera efectiva, todos contribuyendo de sus habilidades para terminar el proyecto a tiempo y sin mayores complicaciones.

7. Apéndice:

```
import java.util.ArrayList;
import java.util.Arrays;

/*
* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
*/

/**
*
* @author Alejandro Salmón, Lorenzo Pazos, Sebastián Velasco, Luciano Moctezuma, Luca Boschetti
*/
public class Calculadora {


/**
*Constuctor vacio de Calculadora
*
*/
   public Calculadora() {
   }

   /**
    * Metod que recibe un caracter, es decir signo, y determina la prioridad para las operaciones
    * @param c
    * @return <ul>
    *         <il> -1 si no cumple ninguna condicion
    *         <il> 3 si '^'
    *         <il> 2 si '*', '/'
    *         <il> 1 si '+', '-'
    *         <il> 0 si '(', ')'
    *         <ul>
    */
   public  int prioridades(char c) {
      int ans = -1;
      //Regresa respuesta -1 si no cumple ninguno de los casos,
      switch (c) {
         case '^' -> ans = 3;
         case '*', '/' -> ans = 2;
         case '+', '-' -> ans = 1;
         case '(', ')' -> ans = 0;
      }

      return ans;
   }

   /**
    * Metodo que recibe una Cadena y la traduce a posfija en forma de ArrayList
    * @param cadena
    * @return double ArrayList de tipo String
    */
   public ArrayList<String> creadorPosfija(String cadena) {
      ArrayList<String> aux = new ArrayList<>();
      ArrayList<String> resp = new ArrayList<>();
      PilaA<Character> pila = new PilaA<>(); // Cambiado a pila genérica
      ArrayList<Character> num= new ArrayList<>(Arrays.asList('1','2','3','4','5','6','7','8','9','0','.','-'));
      StringBuilder sb;
      int i;
      sb = new StringBuilder();
      //bucle para extraer los numeros y signos de una cadena y agregarlos en ese mismo orden a un ArrayList
      for (i = 0; i < cadena.length(); i++) {
            if(i==cadena.length()-1 && num.contains(cadena.charAt(i))){
               sb.append(cadena.charAt(i));
               aux.add(sb.toString());
            }
            else{
```

```
                if(i==0 && cadena.charAt(i)=='-'){
                    sb.append(cadena.charAt(i));
                    if (num.contains(cadena.charAt(i+1))){
                        sb.append(cadena.charAt(i+1));
                        i++;
                        aux.add(sb.toString());
                        sb = new StringBuilder();
                        continue;
                    }
                }
                if(num.contains(cadena.charAt(i))){
                    if(cadena.charAt(i)=='-'){
                        if (num.contains(cadena.charAt(i+1))){
                            sb.append(cadena.charAt(i));
                            sb.append(cadena.charAt(i+1));
                            if (cadena.charAt(i-1)!='('){
                                aux.add("+");
                            }
                            i++;
                            aux.add(sb.toString());
                            sb = new StringBuilder();
                        }else {
                            if(aux.size()>1 && aux.get(aux.size() - 1).equals("(")){
                                sb.append(cadena.charAt(i));
                            }
                        }

                    }
                    else{
                        if(sb.isEmpty())
                            aux.add(cadena.charAt(i) + "");
                        else{
                            aux.add(sb.toString());
                            sb = new StringBuilder();
                            aux.add(cadena.charAt(i)+"");
                        }
                    }
                }
                else{
                    if(sb.isEmpty())
                        aux.add(cadena.charAt(i) + "");
                    else{
                        aux.add(sb.toString());
                        sb=new StringBuilder();
                        aux.add(cadena.charAt(i) + "");
                    }

                }
            }
        }
        // bucle para agregar los valores del ArrayList auxiliar pero ya en forma posfija
        for (i = 0; i < aux.size(); i++) {
            //si el tamaño de la cadena que se encuentra en determinada posicion del array es 1 agrega a la pila el signo que se ecuentra
si es ')' y el signo que esta en el tope de la pila no es '(' agrega al arreglo y quita el numero en el tope
            if (aux.get(i).length() == 1) {
                if (aux.get(i).equals("(")) {
                    pila.push('(');
                }
                else{
                    if (aux.get(i).equals(")")) {
                        while (!pila.isEmpty() && pila.peek() != '(') {
                            if (pila.peek() != '(') {
                                resp.add(pila.pop() + "");
                            } else {
                                pila.push(aux.get(i).charAt(0));
                            }
                        }
                        if (!pila.isEmpty()) {
                            if (pila.peek() == '(')
                                pila.pop();
```

```
                }
            }
            else {
                //bucle para determinar si el signo que se va a meter al Array tenga mayor prioridad que el que se encuentra en la pila
                while (!pila.isEmpty() && prioridades(aux.get(i).charAt(0)) > prioridades(pila.peek())+1) {
                    if(pila.peek()=='('){
                        pila.pop();
                    }
                    else{
                        //cualquier otro signo lo quita de la pila y lo agrega al Array de respuesta
                        resp.add(pila.pop() + "");
                    }
                }
                pila.push(aux.get(i).charAt(0));
            }
        }
    } else{
        resp.add(aux.get(i));
    }
  }
  while (!pila.isEmpty()) {
    resp.add(pila.pop() + "");
  }
  return resp;
}



 /**
  * Metodo que devuelve un double
  * @param a ArrayList de cadenas que el metodo analisa para convertirlo a numeros con decimal y realizar las respectivas
operaciones interpretando la forma pos-fija.
  * @return double Devuelve el resultado de la operación
  */
 public double evaluaPosFija(ArrayList a){
    double resp, x, y;
    PilaA pila= new PilaA();
    int i;
    char d;

    System.out.println(pila);
    for(i=0; i<a.size(); i++){
        //Ciclo para convertir la cadena en numero o en caracter
        try{
            System.out.println(a.get(i));
            pila.push(Double.parseDouble((String) a.get(i)));
        }catch(Exception error){
            d= (Character) a.get(i).toString().charAt(0);

            switch(d){
                //Si recibe el caracter '+' suma los dos valores anteriormente guardados en la pila
                case '+':
                    x=(double) pila.pop();
                    y=(double) pila.pop();
                    pila.push(x+y);
                    break;
                //Si recibe el caracter '-' resta los dos valores anteriormente guardados en la pila
                case '-':
                    x=(double) pila.pop();
                    y=(double) pila.pop();
                    pila.push(y-x);
                    break;
                case '*':
                    //Si recibe el caracter '*' multiplica los dos valores anteriormente guardados en la pila
                    x=(double) pila.pop();
                    y=(double) pila.pop();
                    pila.push(x*y);
                    break;
                case '/':
```

```
                    //Si recibe el caracter '/' divide los dos valores anteriormente guardados en la pila como denominador el valor guardado en el tope de la pila
                    x=(double) pila.pop();
                    y=(double) pila.pop();
                    pila.push(y/x);
                    break;
                case '^':
                    //Si recibe el caracter '^' eleva a la potencia del numero que este guardado en el tope al otro numero que se encuentre en la pila
                    x=(double) pila.pop();
                    y=(double) pila.pop();
                    pila.push(Math.pow(y,x));
                    break;
            }
        }

    }
    resp=(double) pila.pop();

    return resp;
  }

}
```

```
/*
* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
*/

/**
*
* @author lucianomoctezuma
*/
public class Revisores {
  private String operaciones; //Vamos a crear una clase calculadora y su único atributo va a ser la cadena.

  public Revisores(){

  }

  public Revisores(String operaciones){
    this.operaciones=operaciones;
  }

  public void setOperaciones(String operaciones){
    this.operaciones=operaciones;
  }

  public String getOperaciones(){
    return operaciones;
  }

  //Metodo para determinar si el char es un operador

  private boolean esOperador (Character ch){
```

```
        boolean resp=false;

        if(ch == '+' || ch == '*' || ch == '/' || ch == '^' || ch == '-')
            resp=true;
        return resp;
    }

    //Metodo para determinar si el char es un parentesis

    private boolean esParentesis (Character ch){
        boolean resp = false;
        if(ch == '(' || ch == ')')
            resp = true;
        return resp;
    }

    //Los siguientes métodos son para la revisión de sintáxis
    //Metodo que revise que no hayan parentesis vacios
    public boolean revParentesisVacios(){
        boolean bandera = true;
        int i = 0;
        while(i < operaciones.length() && bandera){
            if(operaciones.charAt(i)== '('){
                if(operaciones.charAt(i + 1) == ')')
                    bandera = false;
            }
            i++;
        }
        return bandera;
    }
   //Este  método va a revisar la puntuación, es decir, evitar que el usuario ponga 1..0, ya que esto no es un número.
    public boolean revisadorPuntos(){
        boolean bandera = true;
        int i=0;

        while(i < operaciones.length() - 1 && bandera){ //Se utiliza un while porque hay una bandera.
            //Si hay puntos, se van a contar todos los puntos que existen antes de un operador.
            if (operaciones.charAt(i) == '.'){
                if(operaciones.charAt(i + 1) == '.'){
                    bandera = false;
                }
            }
            i++;
        }
        return bandera;
    }

    public boolean revisadorSignos(){ //Metodo que checa si hay dos signos seguidos, regresa true si no los hay, y regresa false si hay
dos seguidos
        boolean bandera = true;        ///Inicializamos una bandera en true
        int i = 0;
        Character ch = operaciones.charAt(i);

        if(ch == '+' || ch == '*' || ch == '/' || ch == '^')  //Tenemos que poner esta primera condicion porque el primer signo no puede ser
un operando excepto ('!' o '-')
            bandera = false;
        else {
            i++;
            if(operaciones.length() > 0){     //Para que se haga una lectura de signos, por lo menos tiene que haber dos chars
                while(i < operaciones.length() - 1 && bandera){  //el while se condiciona con el tamano de la cadena y que la bandera
siga siendo true
                    if(esOperador(operaciones.charAt(i))){   //checa para ver si el char es operando
                        if(esOperador(operaciones.charAt(i + 1))){  //si el siguiente char a el es operando, cambia la bandera
                            bandera = false;
                        }
                    }
                    if(operaciones.charAt(i) == '!'){  //Lo mismo pero ahora con la posibilidad de un numero negativo para eviatar dos
dignos negativos seguidos
                        if(operaciones.charAt(i + 1) == '!'){  //solo lo compara con otro negativo por que si puede haber un negativo y
despues un operando ejemplo: 4 * !8
```

```
                    bandera = false;
                }
            }
            i++;
        }
      }
      else
        bandera = false;
    }
     if(esOperador(operaciones.charAt(i)) || operaciones.charAt(i) == '!'){ //Esto significa que el ultimo char acabo siendo un
operador o un '!', por lo que tampoco es valido
        bandera = false;
    }
    return bandera;
  }

  public boolean revisadorParentesis() {
    PilaA<Character> pila = new PilaA<>();
    boolean bandera = true;
    int i = 0;
    while (i < operaciones.length() && bandera) {
      char aux = operaciones.charAt(i);
      if (aux == '(') {
        pila.push(aux);
      } else if (aux == ')') {
        if (pila.isEmpty()) {
          return false; // No hay un '(' correspondiente, paréntesis no balanceados.
        } else {
          pila.pop(); // Sacamos el '(' correspondiente.
        }
      }
      i++;
    }
  return pila.isEmpty() && bandera; // Si la pila está vacía, los paréntesis están balanceados.

  }

  //Metodo que revisa que nunca haya una division entre cero
  public boolean revisadorDivisionCero(){
    boolean resp = true;
    int i = 0;

    if(operaciones.length() > 0){
      while (i < operaciones.length() - 1 && resp){
        if(operaciones.charAt(i) == '/') {
          if(operaciones.charAt(i + 1) == '0')
          resp = false;
        }
        i++;
      }
    }
    return resp;
  }

  //Metodo booleano que revise que todos los metodos regresen true y asi determinar si la cadena de operaciones esta bien escrita

  public boolean revSintaxis(){
    boolean resp = true;
    boolean Vacios = revParentesisVacios();
    boolean Puntos = revisadorPuntos();
    boolean Signos = revisadorSignos();
    boolean Parentesis = revisadorParentesis();
    boolean Cero = revisadorDivisionCero();

    if (!Puntos || !Signos || !Parentesis || !Cero || !Vacios){
      resp = false;
    }
    return resp;
  }
}
```

```
import java.util.*;

/*
* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
* Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
*/

/**
* Esta es una clase para la interfaz grafica con la que va a interactuar el usuario
* @author Lorenzo Pazos
*/
public class GUICalculadora extends javax.swing.JFrame {
  boolean prendido=false;


    public GUICalculadora() {
        initComponents();
    }

   /**
    * Metodo para encender la calculadora y habilitar los botones
    * @return boolean
    */

    public boolean Prendido(){
        BotonReset.setEnabled(true);
        AbreParentesis.setEnabled(true);
        CierraParentesis.setEnabled(true);
        Boton0.setEnabled(true);
        Boton1.setEnabled(true);
        Boton2.setEnabled(true);
        Boton3.setEnabled(true);
        Boton4.setEnabled(true);
        Boton5.setEnabled(true);
        Boton6.setEnabled(true);
        Boton7.setEnabled(true);
        Boton8.setEnabled(true);
        Boton9.setEnabled(true);
        BotonSuma.setEnabled(true);
        BotonResta.setEnabled(true);
        BotonMultiplicacion.setEnabled(true);
        BotonDivision.setEnabled(true);
        BotonExponente.setEnabled(true);
        BotonPi.setEnabled(true);
        BotonNegativo.setEnabled(true);
        BotonPuntoDecimal.setEnabled(true);
        BotonResultado.setEnabled(true);
        BotonBack.setEnabled(true);
        Pantalla.setText("");
        prendido=true;
        OnOff.setText("On");

         return prendido;
    }

    /**
     * Metodo para deshabilitar los botones y "apagar" la calculadora
     * @return boolean
     */

    public boolean Apagado(){

        BotonReset.setEnabled(false);
        AbreParentesis.setEnabled(false);
        CierraParentesis.setEnabled(false);
        Boton0.setEnabled(false);
        Boton1.setEnabled(false);
        Boton2.setEnabled(false);
```

```
    Boton3.setEnabled(false);
    Boton4.setEnabled(false);
    Boton5.setEnabled(false);
    Boton6.setEnabled(false);
    Boton7.setEnabled(false);
    Boton8.setEnabled(false);
    Boton9.setEnabled(false);
    BotonSuma.setEnabled(false);
    BotonResta.setEnabled(false);
    BotonMultiplicacion.setEnabled(false);
    BotonDivision.setEnabled(false);
    BotonExponente.setEnabled(false);
    BotonPi.setEnabled(false);
    BotonNegativo.setEnabled(false);
    BotonPuntoDecimal.setEnabled(false);
    BotonResultado.setEnabled(false);
    BotonBack.setEnabled(false);
    Pantalla.setText("");
    prendido=false;
    OnOff.setText("Off");

    return prendido;
}

/**
 * Metodo que reune todos los metodos hechos, los utiliza, y arroja un resultado
 * para la operacion dada por el usuario
 * @param cadena
 * @return String resultado
 */

  public String resultado(String cadena) {
    StringBuilder sb = new StringBuilder();
    Calculadora calculadora = new Calculadora();
    Revisores revisor = new Revisores(cadena);
    ArrayList<String> a= new ArrayList<>(Arrays.asList("+","-","/","*","^"));
    boolean resp;
    resp= false;

    int i=0;
    //bucle solamente para determinar si es un numero solo y que no cambie el textArea
    while(i<a.size() && !resp){
        resp=cadena.contains(a.get(i));
         i++;
    }

    if(resp){
        if(revisor.revSintaxis()){
            sb.append(calculadora.evaluaPosFija(calculadora.creadorPosfija(cadena)));
        }
        else
            sb.append("Math ERROR");
    }
    else
        sb.append(cadena);
    return sb.toString();
}

/**
 * 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
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents
private void initComponents() {

    TituloCalc = new javax.swing.JLabel();
    Pantalla = new javax.swing.JTextField();
    BotonPrendeApaga = new javax.swing.JButton();
```

```
BotonReset = new javax.swing.JButton();
Boton7 = new javax.swing.JButton();
Boton0 = new javax.swing.JButton();
Boton4 = new javax.swing.JButton();
Boton1 = new javax.swing.JButton();
AbreParentesis = new javax.swing.JButton();
Boton3 = new javax.swing.JButton();
Boton8 = new javax.swing.JButton();
Boton5 = new javax.swing.JButton();
Boton2 = new javax.swing.JButton();
CierraParentesis = new javax.swing.JButton();
Boton9 = new javax.swing.JButton();
Boton6 = new javax.swing.JButton();
BotonMultiplicacion = new javax.swing.JButton();
BotonPuntoDecimal = new javax.swing.JButton();
BotonResultado = new javax.swing.JButton();
BotonDivision = new javax.swing.JButton();
BotonSuma = new javax.swing.JButton();
BotonResta = new javax.swing.JButton();
BotonPi = new javax.swing.JButton();
BotonExponente = new javax.swing.JButton();
BotonNegativo = new javax.swing.JButton();
BotonBack = new javax.swing.JButton();
OnOff = new javax.swing.JTextField();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

TituloCalc.setText("Calculadora ED 2024");

Pantalla.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        PantallaActionPerformed(evt);
    }
});

BotonPrendeApaga.setText("On/Off");
BotonPrendeApaga.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        BotonPrendeApagaActionPerformed(evt);
    }
});

BotonReset.setText("AC");
BotonReset.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        BotonResetActionPerformed(evt);
    }
});

Boton7.setText("7");
Boton7.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        Boton7ActionPerformed(evt);
    }
});

Boton0.setText("0");
Boton0.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        Boton0ActionPerformed(evt);
    }
});

Boton4.setText("4");
Boton4.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        Boton4ActionPerformed(evt);
    }
});
```

```
Boton1.setText("1");
Boton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        Boton1ActionPerformed(evt);
    }
});

AbreParentesis.setText("(");
AbreParentesis.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        AbreParentesisActionPerformed(evt);
    }
});

Boton3.setText("3");
Boton3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        Boton3ActionPerformed(evt);
    }
});

Boton8.setText("8");
Boton8.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        Boton8ActionPerformed(evt);
    }
});

Boton5.setText("5");
Boton5.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        Boton5ActionPerformed(evt);
    }
});

Boton2.setText("2");
Boton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        Boton2ActionPerformed(evt);
    }
});

CierraParentesis.setText(")");
CierraParentesis.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        CierraParentesisActionPerformed(evt);
    }
});

Boton9.setText("9");
Boton9.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        Boton9ActionPerformed(evt);
    }
});

Boton6.setText("6");
Boton6.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        Boton6ActionPerformed(evt);
    }
});

BotonMultiplicacion.setText("x");
BotonMultiplicacion.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        BotonMultiplicacionActionPerformed(evt);
    }
});
```

```
BotonPuntoDecimal.setText(".");
BotonPuntoDecimal.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        BotonPuntoDecimalActionPerformed(evt);
    }
});

BotonResultado.setText("=");
BotonResultado.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        BotonResultadoActionPerformed(evt);
    }
});

BotonDivision.setText("/");
BotonDivision.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        BotonDivisionActionPerformed(evt);
    }
});

BotonSuma.setText("+");
BotonSuma.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        BotonSumaActionPerformed(evt);
    }
});

BotonResta.setText("-");
BotonResta.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        BotonRestaActionPerformed(evt);
    }
});

BotonPi.setText("π");
BotonPi.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        BotonPiActionPerformed(evt);
    }
});

BotonExponente.setText("^");
BotonExponente.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        BotonExponenteActionPerformed(evt);
    }
});

BotonNegativo.setText("+/-");
BotonNegativo.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        BotonNegativoActionPerformed(evt);
    }
});

BotonBack.setText("Delete");
BotonBack.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        BotonBackActionPerformed(evt);
    }
});

OnOff.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        OnOffActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
```

```
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addComponent(TituloCalc, javax.swing.GroupLayout.PREFERRED_SIZE, 250,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addGroup(layout.createSequentialGroup()
                                .addComponent(Boton0, javax.swing.GroupLayout.PREFERRED_SIZE, 146,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                    .addComponent(BotonBack, javax.swing.GroupLayout.PREFERRED_SIZE, 146,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                    .addGroup(layout.createSequentialGroup()
                                        .addComponent(BotonPuntoDecimal, javax.swing.GroupLayout.PREFERRED_SIZE, 70,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                        .addComponent(BotonResultado, javax.swing.GroupLayout.PREFERRED_SIZE, 78,
javax.swing.GroupLayout.PREFERRED_SIZE))))
                            .addGroup(layout.createSequentialGroup()
                                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
                                    .addGroup(javax.swing.GroupLayout.Alignment.LEADING, layout.createSequentialGroup()
                                        .addComponent(BotonReset)
                                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                        .addComponent(AbreParentesis, javax.swing.GroupLayout.PREFERRED_SIZE, 68,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                        .addComponent(CierraParentesis, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                        .addGroup(layout.createSequentialGroup()
                                            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                                                .addComponent(Boton4, javax.swing.GroupLayout.PREFERRED_SIZE, 70,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                                .addComponent(Boton1, javax.swing.GroupLayout.PREFERRED_SIZE, 70,
javax.swing.GroupLayout.PREFERRED_SIZE))
                                            .addGap(7, 7, 7)
                                            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                                                .addComponent(Boton5, javax.swing.GroupLayout.PREFERRED_SIZE, 68,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                                .addComponent(Boton2, javax.swing.GroupLayout.PREFERRED_SIZE, 68,
javax.swing.GroupLayout.PREFERRED_SIZE))
                                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                                                .addComponent(Boton3, javax.swing.GroupLayout.PREFERRED_SIZE, 72,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                                .addComponent(Boton6, javax.swing.GroupLayout.PREFERRED_SIZE, 72,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                                        .addGroup(layout.createSequentialGroup()
                                            .addComponent(Boton7, javax.swing.GroupLayout.PREFERRED_SIZE, 70,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                            .addComponent(Boton8, javax.swing.GroupLayout.PREFERRED_SIZE, 70,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                            .addComponent(Boton9, javax.swing.GroupLayout.PREFERRED_SIZE, 71,
javax.swing.GroupLayout.PREFERRED_SIZE))))
                                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                                    .addGroup(layout.createSequentialGroup()
                                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                                            .addComponent(BotonResta, javax.swing.GroupLayout.DEFAULT_SIZE, 77, Short.MAX_VALUE)
                                            .addComponent(BotonMultiplicacion, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                        .addComponent(BotonNegativo))
```

```
                        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
                            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                .addComponent(BotonSuma, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                .addComponent(BotonDivision, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                            .addGap(6, 6, 6)
                            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                                .addComponent(BotonExponente, javax.swing.GroupLayout.DEFAULT_SIZE, 72,
Short.MAX_VALUE)
                                .addComponent(BotonPi, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))))
                    .addComponent(Pantalla, javax.swing.GroupLayout.PREFERRED_SIZE, 319,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addGroup(layout.createSequentialGroup()
                .addComponent(BotonPrendeApaga, javax.swing.GroupLayout.PREFERRED_SIZE, 99,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(OnOff, javax.swing.GroupLayout.PREFERRED_SIZE, 80,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(33, 33, 33))))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(TituloCalc)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(Pantalla, javax.swing.GroupLayout.PREFERRED_SIZE, 42,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(BotonPrendeApaga, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(OnOff, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(BotonReset)
                .addComponent(AbreParentesis)
                .addComponent(CierraParentesis)
                .addComponent(BotonDivision)
                .addComponent(BotonPi))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(Boton7)
                .addComponent(Boton8)
                .addComponent(Boton9)
                .addComponent(BotonSuma)
                .addComponent(BotonExponente))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(Boton4)
                .addComponent(Boton5)
                .addComponent(Boton6)
                .addComponent(BotonResta)
                .addComponent(BotonNegativo))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(Boton1)
                .addComponent(Boton2)
                .addComponent(Boton3)
                .addComponent(BotonMultiplicacion))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(Boton0)
                .addComponent(BotonPuntoDecimal)
                .addComponent(BotonResultado))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
```

```
                .addComponent(BotonBack)
                .addGap(0, 16, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>//GEN-END:initComponents

    private void PantallaActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_PantallaActionPerformed

    }//GEN-LAST:event_PantallaActionPerformed

    private void BotonPrendeApagaActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_BotonPrendeApagaActionPerformed
if(prendido==false){
            Prendido();}
        else
            Apagado();
    }//GEN-LAST:event_BotonPrendeApagaActionPerformed

    private void BotonBackActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_BotonBackActionPerformed
  int length = Pantalla.getText().length();
        int newLength = Pantalla.getText().length() - 1;
        String op;

        if(length >= 1){
            StringBuilder sb = new StringBuilder(Pantalla.getText());
            sb.deleteCharAt(newLength);
            op = sb.toString();
            Pantalla.setText(op);
        }
    }//GEN-LAST:event_BotonBackActionPerformed

    private void Boton5ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_Boton5ActionPerformed
        Pantalla.setText(Pantalla.getText()+ "5");
    }//GEN-LAST:event_Boton5ActionPerformed

    private void BotonPuntoDecimalActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_BotonPuntoDecimalActionPerformed
        Pantalla.setText(Pantalla.getText()+ ".");
    }//GEN-LAST:event_BotonPuntoDecimalActionPerformed

    private void AbreParentesisActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_AbreParentesisActionPerformed
        Pantalla.setText(Pantalla.getText()+ "(");
    }//GEN-LAST:event_AbreParentesisActionPerformed

    private void CierraParentesisActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_CierraParentesisActionPerformed
      Pantalla.setText(Pantalla.getText()+ ")");
    }//GEN-LAST:event_CierraParentesisActionPerformed

    private void Boton7ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_Boton7ActionPerformed
       Pantalla.setText(Pantalla.getText()+ "7");
    }//GEN-LAST:event_Boton7ActionPerformed

    private void Boton8ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_Boton8ActionPerformed
        Pantalla.setText(Pantalla.getText()+ "8");
    }//GEN-LAST:event_Boton8ActionPerformed

    private void Boton9ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_Boton9ActionPerformed
        Pantalla.setText(Pantalla.getText()+ "9");
    }//GEN-LAST:event_Boton9ActionPerformed

    private void Boton4ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_Boton4ActionPerformed
        Pantalla.setText(Pantalla.getText()+ "4");
    }//GEN-LAST:event_Boton4ActionPerformed

    private void Boton6ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_Boton6ActionPerformed
       Pantalla.setText(Pantalla.getText()+ "6");
    }//GEN-LAST:event_Boton6ActionPerformed
```

```
    private void Boton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_Boton1ActionPerformed
        Pantalla.setText(Pantalla.getText()+ "1");
    }//GEN-LAST:event_Boton1ActionPerformed

    private void Boton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_Boton2ActionPerformed
        Pantalla.setText(Pantalla.getText()+ "2");
    }//GEN-LAST:event_Boton2ActionPerformed

    private void Boton3ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_Boton3ActionPerformed
       Pantalla.setText(Pantalla.getText()+ "3");
    }//GEN-LAST:event_Boton3ActionPerformed

    private void Boton0ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_Boton0ActionPerformed
       Pantalla.setText(Pantalla.getText()+ "0");
    }//GEN-LAST:event_Boton0ActionPerformed

    private void BotonDivisionActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_BotonDivisionActionPerformed
        Pantalla.setText(Pantalla.getText()+ "/");
    }//GEN-LAST:event_BotonDivisionActionPerformed

    private void BotonSumaActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_BotonSumaActionPerformed
        Pantalla.setText(Pantalla.getText()+ "+");
    }//GEN-LAST:event_BotonSumaActionPerformed

    private void BotonRestaActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_BotonRestaActionPerformed
        Pantalla.setText(Pantalla.getText()+ "-");
    }//GEN-LAST:event_BotonRestaActionPerformed

    private void BotonMultiplicacionActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_BotonMultiplicacionActionPerformed
       Pantalla.setText(Pantalla.getText()+ "*");
    }//GEN-LAST:event_BotonMultiplicacionActionPerformed

    private void BotonResultadoActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_BotonResultadoActionPerformed

        Pantalla.setText(resultado(Pantalla.getText()));
    }//GEN-LAST:event_BotonResultadoActionPerformed

    private void BotonPiActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_BotonPiActionPerformed
       Pantalla.setText(Pantalla.getText()+ "3.141592");
    }//GEN-LAST:event_BotonPiActionPerformed

    private void BotonExponenteActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_BotonExponenteActionPerformed
        Pantalla.setText(Pantalla.getText()+ "^");
    }//GEN-LAST:event_BotonExponenteActionPerformed

    private void BotonNegativoActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_BotonNegativoActionPerformed
        Pantalla.setText(Pantalla.getText()+"(-");
    }//GEN-LAST:event_BotonNegativoActionPerformed

    private void BotonResetActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_BotonResetActionPerformed
        Pantalla.setText("");
    }//GEN-LAST:event_BotonResetActionPerformed

    private void OnOffActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_OnOffActionPerformed

    }//GEN-LAST:event_OnOffActionPerformed

    /**
     * Main que corre la interfaz grafica
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
```

```
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(GUICalculadora.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(GUICalculadora.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(GUICalculadora.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(GUICalculadora.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new GUICalculadora().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton AbreParentesis;
    private javax.swing.JButton Boton0;
    private javax.swing.JButton Boton1;
    private javax.swing.JButton Boton2;
    private javax.swing.JButton Boton3;
    private javax.swing.JButton Boton4;
    private javax.swing.JButton Boton5;
    private javax.swing.JButton Boton6;
    private javax.swing.JButton Boton7;
    private javax.swing.JButton Boton8;
    private javax.swing.JButton Boton9;
    private javax.swing.JButton BotonBack;
    private javax.swing.JButton BotonDivision;
    private javax.swing.JButton BotonExponente;
    private javax.swing.JButton BotonMultiplicacion;
    private javax.swing.JButton BotonNegativo;
    private javax.swing.JButton BotonPi;
    private javax.swing.JButton BotonPrendeApaga;
    private javax.swing.JButton BotonPuntoDecimal;
    private javax.swing.JButton BotonReset;
    private javax.swing.JButton BotonResta;
    private javax.swing.JButton BotonResultado;
    private javax.swing.JButton BotonSuma;
    private javax.swing.JButton CierraParentesis;
    private javax.swing.JTextField OnOff;
    private javax.swing.JTextField Pantalla;
    private javax.swing.JLabel TituloCalc;
    // End of variables declaration//GEN-END:variables
}
```