



Cifras Significativas

Jenifeer Juliette O. Ruiz, Jerónimo Israel M. Quintero y Paul Emmanuel V. Vasquez

Ingeniería en Sistemas Computacionales, Universidad Politécnica de Tlaxcala Región Poniente

Física

Ing. Vanesa T. Zavala

25 de Enero del 2024



Índice.

Contenido

| | |
|---|----|
| Resumen del proyecto..... | 3 |
| Introducción..... | 4 |
| ¿Qué son las Cifras Significativas?..... | 4 |
| ¿Cuál es el método de las cifras significativas?..... | 5 |
| Entorno de desarrollo utilizado, JGrasp: | 5 |
| Lenguaje de programación utilizado, Java:..... | 6 |
| Librerías utilizadas. | 7 |
| Código..... | 8 |
| Explicación del código..... | 11 |
| Ejecución del código. | 12 |
| Resultado del Código:..... | 14 |
| Conclusión:..... | 14 |
| Bibliografía..... | 14 |



Resumen del proyecto

El proyecto es una aplicación de interfaz gráfica de usuario (GUI) desarrollada en Java utilizando la biblioteca Swing. Proporciona dos funciones principales:

Cálculo de Cifras Significativas:

- Esta función permite al usuario ingresar un número en un campo de texto.
- Al presionar un botón, la aplicación elimina los ceros a la izquierda del número ingresado, ya sea positivo o negativo.
- El resultado, es decir, el número sin los ceros a la izquierda, se muestra en un área de texto en la misma ventana.

Cálculo de Notación Científica:

- Esta función permite al usuario ingresar una base y un exponente en campos de texto separados.
- Al presionar un botón, la aplicación calcula el resultado multiplicando la base por 10 elevado al exponente.
- El resultado se muestra en una ventana emergente.

El proyecto utiliza la programación orientada a objetos en Java para organizar el código de manera modular y clara. Se siguen las prácticas estándar de desarrollo de interfaces gráficas de usuario con Swing. Además, la entrada del usuario se valida para garantizar la correcta



Introducción

Objetivo:

El objetivo del proyecto era crear una aplicación simple que proporcionara herramientas para el manejo básico de números, incluyendo la eliminación de ceros a la izquierda y el cálculo de números en notación científica. Se quería desarrollar una interfaz gráfica de usuario (GUI) intuitiva y fácil de usar para realizar estas operaciones.

Cómo se hizo:

Para lograr este objetivo, se utilizó el lenguaje de programación Java junto con la biblioteca Swing para crear la interfaz gráfica de usuario. Se dividió el proyecto en dos partes principales.

¿Qué son las Cifras Significativas?:

Las cifras significativas, también conocidas como dígitos significativos o cifras significativas, son los dígitos que se consideran relevantes en una medida. Estos dígitos reflejan la precisión con la que se puede expresar una cantidad medida.

En un número, las cifras significativas son todos los dígitos que contribuyen a su precisión, junto con el primer dígito no nulo a la izquierda y cualquier dígito a la derecha de este primer dígito no nulo que es seguido por ceros (a menos que se indique lo contrario por un punto decimal).

Por ejemplo:

En el número 345.67, hay cinco cifras significativas: 3, 4, 5, 6 y 7.

En el número 0.00245, hay tres cifras significativas: 2, 4 y 5.

Las cifras significativas son importantes porque indican la precisión de una medición. Cuando se realizan cálculos con números que tienen diferentes cantidades de cifras significativas, es importante mantener el número apropiado de cifras significativas en el resultado final para mantener la precisión adecuada.

En el contexto científico y de ingeniería, es común utilizar las cifras significativas para expresar la precisión de los datos experimentales, mediciones y cálculos.



¿Cuál es el método de las cifras significativas?

- Todos los dígitos diferentes de cero son cifras significativas.
- Los ceros entre dos cifras significativas son cifras significativas.
- Los ceros a la izquierda del primer dígito diferente de cero no son cifras significativas.
- Los ceros a la derecha del último dígito diferente de cero son cifras significativas si hay un punto decimal en el número. (Academy, 2024)

Entorno de desarrollo utilizado, JGrasp:

JGrasp es un entorno de desarrollo integrado (IDE) para Java que se utiliza para escribir, compilar y depurar programas Java. Algunas características importantes de JGrasp son:

Editor de texto avanzado: Proporciona un editor de texto con resaltado de sintaxis para Java y otras características útiles, como autocompletar y plegado de código.

Herramientas de compilación y depuración: JGrasp permite compilar y ejecutar programas Java directamente desde el IDE. Además, ofrece herramientas de depuración para detectar y corregir errores en el código.

Visualización de estructuras de datos: Proporciona herramientas visuales para la visualización de estructuras de datos, como árboles y grafos, lo que puede ser útil para comprender y depurar programas complejos.

Soporte multiplataforma: JGrasp está disponible para Windows, macOS y Linux, lo que permite a los usuarios desarrollar aplicaciones Java en diferentes sistemas operativos.



Leguaje de programación utilizado, Java:

Java es un lenguaje de programación de alto nivel y orientado a objetos que se utiliza ampliamente en el desarrollo de software. Algunas características importantes de Java son:

Portabilidad: Java es un lenguaje portátil, lo que significa que el código Java puede ejecutarse en cualquier plataforma que tenga una máquina virtual Java (JVM) instalada. Esto permite que las aplicaciones Java funcionen en una amplia variedad de dispositivos y sistemas operativos.

Orientado a objetos: Java es un lenguaje orientado a objetos, lo que significa que todo en Java es un objeto. Esto facilita la creación de programas modulares y extensibles.

Seguridad: Java tiene características de seguridad integradas, como el modelo de seguridad de sandbox y la verificación de bytecodes, que ayudan a proteger los sistemas contra programas maliciosos.

Rendimiento: Aunque Java es un lenguaje interpretado, las implementaciones modernas de Java (como OpenJDK y Oracle JDK) utilizan técnicas avanzadas de optimización y compilación JIT (Just-In-Time) para ofrecer un rendimiento cercano al de lenguajes compilados.

En resumen, JGrasp es un IDE utilizado para desarrollar programas Java, mientras que Java es el lenguaje de programación utilizado para escribir esos programas. Juntos, proporcionan un entorno completo para escribir, compilar y depurar aplicaciones Java.



Librerías utilizadas.

javax.swing.*: Esta biblioteca proporciona clases y componentes para crear interfaces gráficas de usuario (GUI) en Java. En este código, se utiliza para crear ventanas, botones, campos de texto, etiquetas, áreas de texto y cuadros de diálogo.

java.awt.*: Este paquete proporciona clases para crear y manipular componentes de la interfaz de usuario y gráficos básicos. En este código, se utiliza para organizar y diseñar los componentes en la interfaz de usuario, mediante layouts como GridLayout.

java.awt.event.ActionEvent y java.awt.event.ActionListener: Estas clases y la interfaz se utilizan para manejar eventos de acción, como hacer clic en un botón. En el código, se emplean para ejecutar ciertas acciones cuando se hace clic en los botones.

javax.swing.JOptionPane: Esta clase proporciona métodos estáticos para mostrar ventanas emergentes, como cuadros de diálogo de mensaje, entrada y confirmación. En el código, se usa para mostrar mensajes al usuario.

javax.swing.SwingUtilities: Esta clase proporciona métodos estáticos para realizar operaciones relacionadas con Swing en el hilo de despacho de eventos (EDT). En el código, se utiliza para ejecutar la creación de la interfaz de usuario Swing en el hilo EDT.

```
import javax.swing.*;  
import java.awt.*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;
```

Imagen1librerías



Código

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class MainFrame1 extends JFrame {

    public MainFrame1() {
        super("Selección de Operación");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(400, 200);
        setLocationRelativeTo(null);

        JPanel mainPanel = new JPanel(new GridLayout(2, 1));

        JButton button1 = new JButton("Números Normales");
        button1.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                dispose();
                new CifrasSignificativasGUI();
            }
        });
        mainPanel.add(button1);

        JButton button2 = new JButton("Notación Científica");
        button2.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                dispose();
                new NotacionCientificaFrame();
            }
        });
        mainPanel.add(button2);

        add(mainPanel);
        setVisible(true);
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            @Override
            public void run() {
                new MainFrame1();
            }
        });
    }
}
```




```
    }  
    });  
}  
  
class CifrasSignificativasGUI extends JFrame {  
  
    private JTextField inputField;  
    private JTextArea resultArea;  
  
    public CifrasSignificativasGUI() {  
        super("Cifras Significativas");  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        setSize(400, 200);  
        setLocationRelativeTo(null);  
  
        JPanel mainPanel = new JPanel(new GridLayout(3, 1));  
  
        inputField = new JTextField();  
        mainPanel.add(inputField);  
  
        JButton calculateButton = new JButton("Calcular Cifras Significativas");  
        calculateButton.addActionListener(new ActionListener() {  
            @Override  
            public void actionPerformed(ActionEvent e) {  
                mostrarCifrasSignificativas();  
            }  
        });  
        mainPanel.add(calculateButton);  
  
        resultArea = new JTextArea();  
        resultArea.setEditable(false);  
        mainPanel.add(resultArea);  
  
        add(mainPanel);  
        setVisible(true);  
    }  
  
    private void mostrarCifrasSignificativas() {  
        try {  
            String inputText = inputField.getText();  
            String numeroSinCeros = inputText.replaceFirst("^0+(?!$)", "");  
            resultArea.setText("Cifras Significativas: " + numeroSinCeros);  
        } catch (NumberFormatException ex) {  
            resultArea.setText("Error: Ingresa un número válido.");  
        }  
    }  
}
```



```
}  
}  
}  
  
class NotacionCientificaFrame extends JFrame {  
  
    public NotacionCientificaFrame() {  
        super("Notación Científica");  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        setSize(400, 200);  
        setLocationRelativeTo(null);  
  
        JPanel mainPanel = new JPanel(new GridLayout(3, 2));  
  
        JTextField baseField = new JTextField();  
        mainPanel.add(new JLabel("Base:"));  
        mainPanel.add(baseField);  
  
        JTextField exponentField = new JTextField();  
        mainPanel.add(new JLabel("Exponente:"));  
        mainPanel.add(exponentField);  
  
        JButton calculateButton = new JButton("Calcular");  
        calculateButton.addActionListener(new ActionListener() {  
            @Override  
            public void actionPerformed(ActionEvent e) {  
                try {  
                    double base = Double.parseDouble(baseField.getText());  
                    int exponente = Integer.parseInt(exponentField.getText());  
                    double resultado = base * Math.pow(10, exponente);  
                    JOptionPane.showMessageDialog(null, "Resultado: " +  
resultado, "Resultado", JOptionPane.PLAIN_MESSAGE);  
                } catch (NumberFormatException ex) {  
                    JOptionPane.showMessageDialog(null, "Error: Ingresa un número  
válido.", "Error", JOptionPane.ERROR_MESSAGE);  
                }  
            }  
        });  
        mainPanel.add(calculateButton);  
  
        add(mainPanel);  
        setVisible(true);  
    }  
}
```



Explicación del código.

El código proporciona una interfaz gráfica de usuario (GUI) básica en Java utilizando la biblioteca Swing. Consiste en dos partes principales:

Selección de Operación (MainFrame):

Cuando se ejecuta el programa, se muestra una ventana que presenta dos botones: "Números Normales" y "Notación Científica".

Al hacer clic en el botón "Números Normales", se abre una nueva ventana donde el usuario puede ingresar un número. Este número puede ser positivo o negativo y puede contener ceros a la izquierda. Al hacer clic en un botón, el programa elimina los ceros a la izquierda del número ingresado y muestra el resultado en la misma ventana.

Al hacer clic en el botón "Notación Científica", se abre otra ventana donde el usuario puede ingresar una base y un exponente. Luego, el programa calcula el resultado multiplicando la base por 10 elevado al exponente y muestra el resultado en una ventana emergente.

Cifras Significativas (CifrasSignificativasGUI):

Esta clase define la ventana para la operación de "Números Normales".

El usuario puede ingresar un número en un campo de texto y hacer clic en un botón para calcular las cifras significativas del número.

El programa elimina los ceros a la izquierda del número ingresado y muestra el resultado (el número sin ceros a la izquierda) en un área de texto en la misma ventana.

Notación Científica (NotacionCientificaFrame):

Esta clase define la ventana para la operación de "Notación Científica".

El usuario puede ingresar una base y un exponente en campos de texto y hacer clic en un botón para calcular el resultado.

El programa realiza el cálculo de la notación científica y muestra el resultado en una ventana emergente.

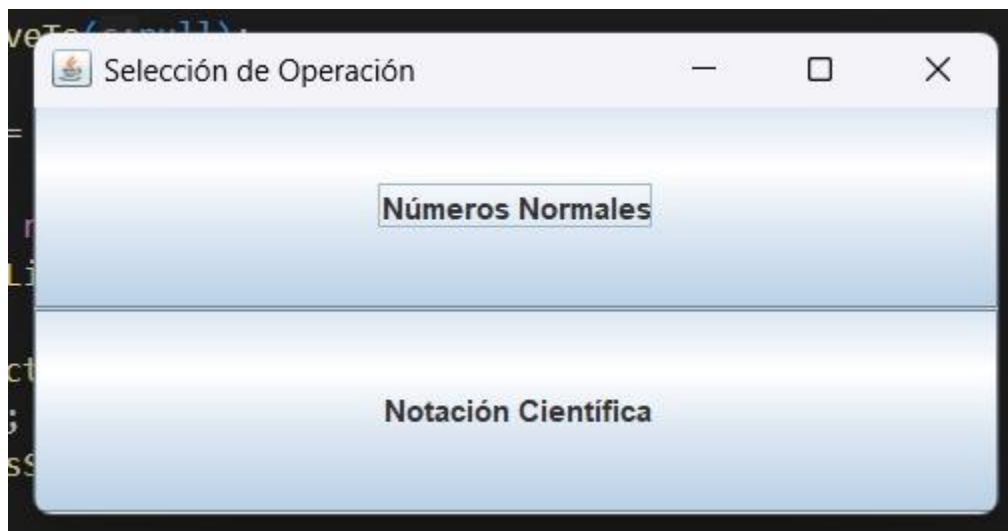


En resumen, el código proporciona una interfaz simple donde el usuario puede realizar operaciones básicas con números, eliminando ceros a la izquierda y calculando la notación científica.

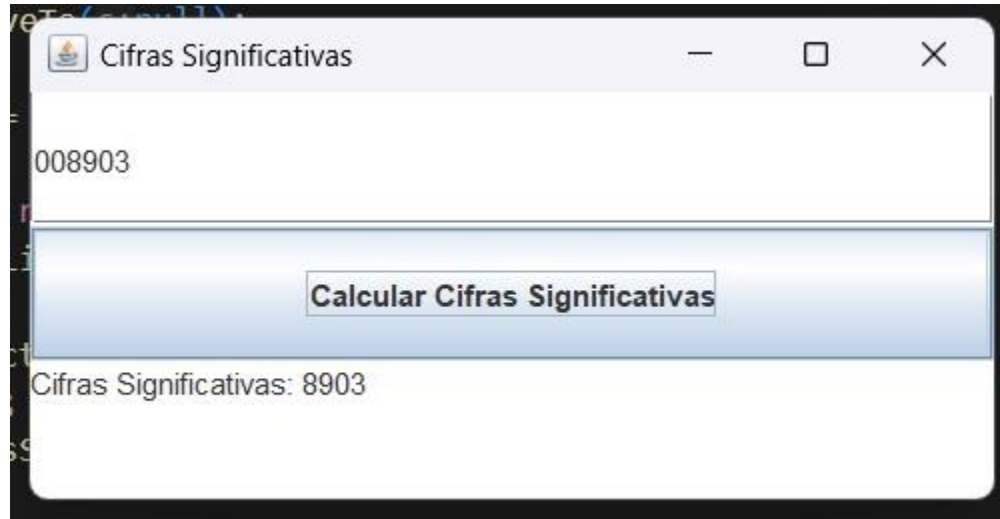
```
J SignificantFiguresCalculator.java  J CifrasSignificativasGUI.java  J CifrasSignificativasGUI2.java  J MainFrame1.java X  J MainFrame2.java 2
J MainFrame1.java > NotacionCientificaFrame
1  import javax.swing.*;
2  import java.awt.*;
3  import java.awt.event.ActionEvent;
4  import java.awt.event.ActionListener;
5
6  public class MainFrame1 extends JFrame {
7
8      public MainFrame1() {
9          super(title:"Selección de Operación");
10         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
11         setSize(width:400, height:200);
12         setLocationRelativeTo(c:null);
13
14         JPanel mainPanel = new JPanel(new GridLayout(rows:2, cols:1));
15
16         JButton button1 = new JButton(text:"Números Normales");
17         button1.addActionListener(new ActionListener() {
18             @Override
19             public void actionPerformed(ActionEvent e) {
20                 dispose();
21                 new CifrasSignificativasGUI();
22             }
23         });
24     }
25 }
```

2código para notación científica

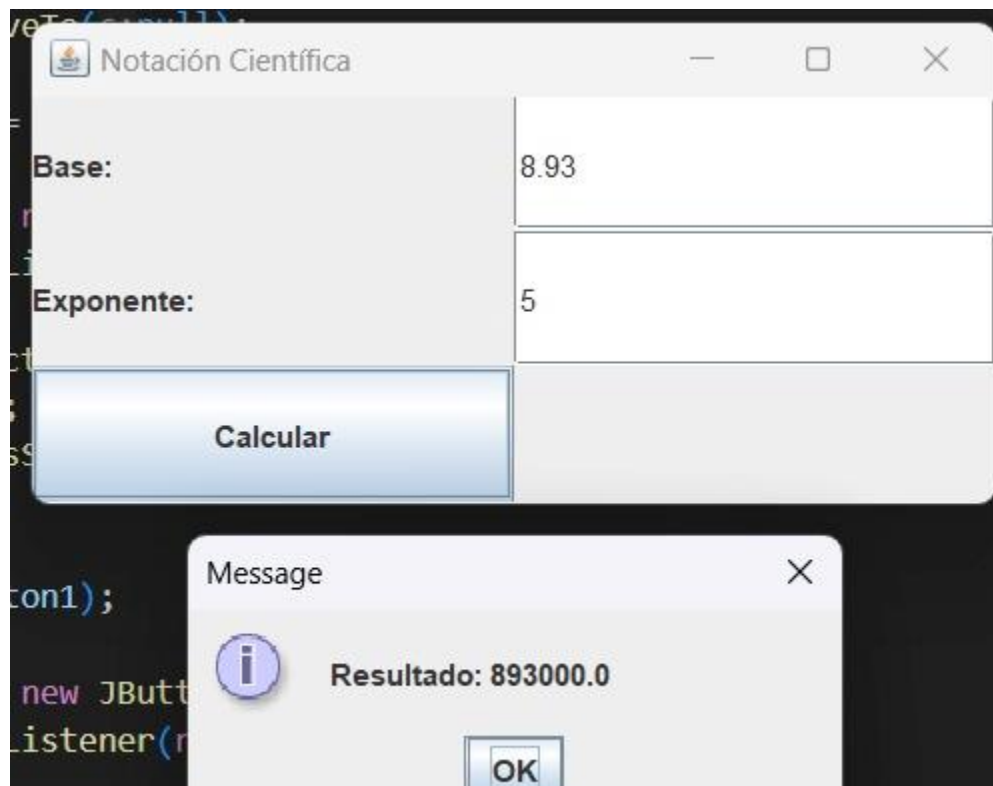
Ejecución del código.



3ejemplo de ejecución del código



4ejemplo de ejecución 2



5ejecución del programa finalizada



Resultado del Código:

La aplicación desarrollada proporciona una interfaz gráfica de usuario (GUI) en Java utilizando la biblioteca Swing.

Ofrece dos operaciones principales:

Cálculo de Cifras Significativas: Permite al usuario ingresar un número y elimina los ceros a la izquierda, mostrando el resultado en la misma ventana.

Cálculo de Notación Científica: Permite al usuario ingresar una base y un exponente, calculando la notación científica del número ingresado y mostrando el resultado en una ventana emergente.

Conclusión:

El proyecto logra su objetivo de proporcionar herramientas básicas para el manejo de números a través de una interfaz gráfica intuitiva y fácil de usar.

La aplicación demuestra el uso de programación orientada a objetos en Java y el desarrollo de interfaces gráficas con Swing.

Se utilizó JGrasp como el entorno de desarrollo integrado (IDE) para escribir, compilar y depurar el código Java.

El proyecto sirve como un ejemplo simple pero funcional de cómo desarrollar una aplicación GUI en Java para realizar operaciones básicas con números.

Además, este proyecto podría ser ampliado en el futuro para agregar más funcionalidades y mejorar la experiencia del usuario, como la validación de entrada de datos y la mejora de la interfaz de usuario.

Bibliografía

Academy, K. (24 de 01 de 2024). *Khan Academy*. Obtenido de Khan Academy:

<https://www.bing.com/search?q=introduccion+a+las+cifras+significativas&form=ANNT H1&refig=de7b73f77aa84b0aadc5940d65712b3a&sp=1&ghc=1&lq=0&qs=NWT&pq=i ntroduccion+a+las+cifras+sig&sc=6- 29&cvid=de7b73f77aa84b0aadc5940d65712b3a&showconv=1>

(s.f.). *Cifras Significativas*.