



UNIVERSIDAD TECNOLÓGICA DE PANAMÁ
FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES
DEPARTAMENTO DE PROGRAMACIÓN DE COMPUTADORAS
GUÍA DE ACTIVIDAD N°. 01
(TAREA / PRÁCTICA, LABORATORIOS, ETC.)

Facilitador(a): Emilio Batista

Asignatura: Herramientas de la Programación II

Estudiantes:

- Geremi Tejeira | 9-768-42
- Terry He | 8-1021-2180
- Jhuomar Barría | 9-766-196

Fecha: 17/04/2025

Grupo: 11L-128

A. TÍTULO DE LA EXPERIENCIA: Guía de Laboratorio Tarea: Crear programa para la formula cuadrática, utilizando los conceptos de creación del objeto, y los métodos.

B. TEMAS: Confección y análisis del problema, y creación programa fuente.

C. OBJETIVO(S):

Conocer las formas básicas de construcción de pseudo códigos dentro del ámbito de la lógica de algoritmos, de manera tal que el estudiante comprenda los pasos necesarios que debe seguir para resolución de problemas.

D. METODOLOGÍA:

- 1) *De acuerdo con la cantidad de estudiantes y/o equipo, trabaje de manera individual o grupal.*
- 2) *Distribuya la actividad entre los estudiantes y/o grupos.*
- 3) *La actividad debe ser desarrollado en el tiempo establecido por el docente.*
- 4) *La solución de los problemas se realizará usando Pseudocódigo.*
- 5) *Presentar la prueba de escritorio para validar la solución.*
- 6) *Para ejecutar la solución utilizar el lenguaje C.*
- 7) *La entrega de la solución queda a discreción del docente. Sugerencia: Plataforma, dispositivo físico, correo electrónico, sustentación e impreso, entre otros.*

E. ENUNCIADO:

Confeccionar un programa en java para calcular la formula cuadrática, y que pueda realizar los siguiente:

- ✓ Utilizar las sentencias, imprimir y leer en java el if, definir las variables flotantes y enteras.
- ✓ Definir nombre a las variables del entorno, A, B,C.
- ✓ Verificar la formula cuadrática (su forma)
- ✓ Verificar si las raíces pueden ser negativas (formula general de la cuadrática)
- ✓ Calcular los valores x1, x2
- ✓ Entrada de datos A, B, C.
- ✓ Imprimir resultados.
- ✓ Subir trabajo a la plataforma de la UTP.
- ✓ Crear objeto lectura de datos, método calcular valores x1, x2,
- ✓ Crear el método de impresión de valores x1, x2

$$ax^2 + bx + c = 0$$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$



UNIVERSIDAD TECNOLÓGICA DE PANAMÁ
FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES
DEPARTAMENTO DE PROGRAMACIÓN DE COMPUTADORAS
GUÍA DE ACTIVIDAD N°. 01
(TAREA / PRÁCTICA, LABORATORIOS, ETC.)

F. PROCEDIMIENTO:

Indicar todos los pasos necesarios para realizar la experiencia. Se pueden mostrar con esquemas, dibujos, imágenes, fotos, entre otros.

- a. **Análisis y Diseño del Problema**
- b. **Codificación en Java**
- c. **Captura de la corrida del programa de java, en Visual estudio code o eclipse.**
- d. **Codificación en Java**

G. RECURSOS:

Listar los distintos recursos que se emplearán en la Asignación: La guía de la actividad, lápiz, papel, borradores, tablero, marcador, aplicaciones de Celular (CppDroid), Moodle, Videos o tutoriales de youtube, entre otros.

H. RESULTADOS (OPCIONAL, DE ACUERDO CON LA ACTIVIDAD):

Sección donde se colocan los datos obtenidos con la experiencia. Se formatea de acuerdo con la salida deseada: conjunto de preguntas, tablas, gráficos, preguntas abiertas, entre otros.

Análisis y Diseño		
Entradas	Proceso	Salida
Leer los valores a,b, c	Calcular el discriminante $D = b^2 - 4ac$ Verificar el valor del discriminante: - Si $D > 0$: calcular dos soluciones reales. - Si $D = 0$: calcular una solución real. - Si $D < 0$: calcular dos soluciones complejas.	Imprimir los resultados de x_1 y x_2 . Mostrar reporte con: <ul style="list-style-type: none">• Nombre• Cédula• Nombre de la Facultad• Universidad• Número de grupo• Fecha y año• Valores de x_1 y x_2
Algoritmo		
<pre>Algoritmo ResolucionEcuacionCuadraticaConInterfaz { // Inicializar los colores personalizados para el diseño oscuro Definir NEON_BLUE, NEON_PINK, NEON_GREEN, DARK_BG, DARKER_BG, TEXT_COLOR como Colores // Establecer apariencia FlatDarkLaf y configurar la interfaz Intentar Establecer apariencia FlatDarkLaf Configurar colores personalizados en los componentes de UI Capturar excepción Imprimir error // Crear ventana principal Crear JFrame llamado frame con título "Resolución de Ecuaciones Cuadráticas" Configurar cierre, tamaño mínimo y estado maximizado de la ventana // Crear panel principal y usar GridBagLayout Crear mainPanel con fondo DARK_BG Asignar mainPanel como contenido de frame // Agregar panel de título Llamar a función createTitlePanel y agregar al mainPanel</pre>		



UNIVERSIDAD TECNOLÓGICA DE PANAMÁ
FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES
DEPARTAMENTO DE PROGRAMACIÓN DE COMPUTADORAS
GUÍA DE ACTIVIDAD N°. 01
(TAREA / PRÁCTICA, LABORATORIOS, ETC.)

```
// Crear campos de entrada para a, b y c
Crear aField, bField, cField como JTextField
Aplicar estilo a cada campo con styleTextField

// Crear panel de entrada con los campos a, b, c y agregarlo al mainPanel
Llamar a createInputPanel(aField, bField, cField) y agregarlo al mainPanel

// Crear botón "Resolver Ecuación"
Llamar a createButtonPanel y obtener el botón resolverBtn
Agregar el panel del botón al mainPanel

// Crear panel de resultados y agregarlo al mainPanel
Llamar a createResultPanel y agregarlo al mainPanel

// Definir acción al presionar el botón "Resolver"
Al hacer clic en resolverBtn hacer:
{
    Limpiar resultadoPanel
    Leer valores de a, b, c desde los campos de texto

    Si a es igual a 0 entonces
        Mostrar error "A no puede ser cero"
        Terminar acción
    FinSi

    // Mostrar ecuación original usando LaTeX
    Crear cadena ecuacionOriginal con formato de  $ax^2 + bx + c = 0$ 
    Mostrar ecuación usando TeXFormula y TeXIcon dentro de un panel

    // Calcular discriminante
    Calcular discriminante =  $b^2 - 4ac$ 
    Calcular dosA =  $2a$ 
    Definir variables latex, colorResultado

    Si discriminante > 0 entonces
        Calcular raíces reales distintas x1 y x2
        Crear cadena latex con resultado
        Asignar colorResultado = NEON_BLUE
    Sino si discriminante == 0 entonces
        Calcular raíz única x
        Crear cadena latex con resultado
        Asignar colorResultado = NEON_GREEN
    Sino
        Calcular parte real e imaginaria de raíces complejas
        Crear cadena latex con resultado
        Asignar colorResultado = NEON_PINK
    FinSi

    // Mostrar desarrollo del resultado
    Crear panel discriminante con valor mostrado
    Crear panel desarrollo con fórmula LaTeX usando TeXFormula y TeXIcon
    Agregar ambos paneles al resultadoPanel
```



UNIVERSIDAD TECNOLÓGICA DE PANAMÁ
FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES
DEPARTAMENTO DE PROGRAMACIÓN DE COMPUTADORAS
GUÍA DE ACTIVIDAD N°. 01
(TAREA / PRÁCTICA, LABORATORIOS, ETC.)

```
// Imprimir datos del estudiante y resultados
Crear cadena de texto estudianteInfo con los siguientes datos:
    "Nombre: [Nombre del estudiante]"
    "Cédula: [Número de cédula]"
    "Nombre de la Facultad: [Nombre de la Facultad]"
    "Universidad: [Nombre de la Universidad]"
    "Número de grupo: [Número de grupo]"
    "Fecha y año: [Fecha y año actuales]"
    "Valor de  $x_1$ : [Valor de x1]"
    "Valor de  $x_2$ : [Valor de x2]"

Mostrar cadena estudianteInfo en un panel de información o consola
}
Capturar error de formato numérico
Mostrar mensaje de error "Ingrese valores válidos"

// Finalizar configuración y mostrar ventana
Ajustar frame y hacerlo visible
}
```

Programa en lenguaje java

```
● ● ●
Fórmula:  $x = \frac{-B \pm \sqrt{B^2 - 4A}}{2A}$ 
Ingrese el valor de A:
1
Ingrese el valor de B:
-3
Ingrese el valor de C:
2

Resultado:
x1 = 2.0
x2 = 1.0
```

Este proyecto en Java resuelve ecuaciones cuadráticas de la forma $ax^2 + bx + c = 0$, integrando la lógica de cálculo y la interfaz gráfica en una sola clase principal llamada App. Utiliza principios de Programación Orientada a Objetos como encapsulamiento, reutilización y modularidad. La clase incluye métodos para capturar datos, calcular raíces (reales e imaginarias) y mostrar resultados. También emplea un ActionListener para manejar eventos al presionar el botón de cálculo. Su estructura clara y modular facilita futuras mejoras y demuestra una implementación efectiva de la POO en aplicaciones con interfaz gráfica.

UNIVERSIDAD TECNOLÓGICA DE PANAMÁ
FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES
DEPARTAMENTO DE PROGRAMACIÓN DE COMPUTADORAS
GUÍA DE ACTIVIDAD N°. 01
(TAREA / PRÁCTICA, LABORATORIOS, ETC.)

```
import java.util.Scanner;

public class FormulaCuadratica {

    // Método para leer los valores A, B y C desde consola
    public static float[] leerValores() {
        Scanner scanner = new Scanner(System.in);
        float[] valores = new float[3];

        System.out.println("Ingrese el valor de A: ");
        valores[0] = scanner.nextFloat();

        System.out.println("Ingrese el valor de B: ");
        valores[1] = scanner.nextFloat();

        System.out.println("Ingrese el valor de C: ");
        valores[2] = scanner.nextFloat();

        return valores;
    }

    // Método para calcular las raíces de la fórmula cuadrática
    public static void calcularRaices(float A, float B, float C) {
        double discriminante = (B * B) - (4 * A * C);

        // Si el discriminante es positivo, tenemos raíces reales
        if (discriminante > 0) {
            double x1 = (-B + Math.sqrt(discriminante)) / (2 * A);
            double x2 = (-B - Math.sqrt(discriminante)) / (2 * A);
            imprimirResultados(x1, x2);
        }
        // Si el discriminante es cero, tenemos una única raíz real
        else if (discriminante == 0) {
            double x = -B / (2 * A);
            imprimirResultados(x, x);
        }
        // Si el discriminante es negativo, las raíces son complejas (imaginarias)
        else {
            double parteReal = -B / (2 * A);
            double parteImaginaria = Math.sqrt(-discriminante) / (2 * A);

            // Imprimimos las raíces complejas
            System.out.println("Las raíces son complejas:");
            System.out.println("x1 = " + parteReal + " + " + parteImaginaria + "i");
            System.out.println("x2 = " + parteReal + " - " + parteImaginaria + "i");
        }
    }

    // Método para imprimir las raíces x1 y x2 (en caso de raíces reales)
    public static void imprimirResultados(double x1, double x2) {
        System.out.println("Resultado:");
        System.out.println("x1 = " + x1);
        System.out.println("x2 = " + x2);
    }

    // Método principal del programa
    public static void main(String[] args) {
        System.out.println("=== Cálculo de Fórmula Cuadrática ===");
        System.out.println("Fórmula:  $x = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}$ ");

        float[] valores = leerValores();
        float A = valores[0];
        float B = valores[1];
        float C = valores[2];

        // Comprobamos que A no sea 0, ya que no sería una ecuación cuadrática válida
        if (A == 0) {
            System.out.println("Error: A no puede ser cero.");
        } else {
            // Calculamos las raíces con los valores ingresados
            calcularRaices(A, B, C);
        }
    }
}
```

Ilustración 1: Este es el código fuente, usado de referencia para crear el programa final completo con GUI

En esta sección capture las pantallas de la corrida de su programa.

UNIVERSIDAD TECNOLÓGICA DE PANAMÁ
FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES
DEPARTAMENTO DE PROGRAMACIÓN DE COMPUTADORAS
GUÍA DE ACTIVIDAD N°. 01
(TAREA / PRÁCTICA, LABORATORIOS, ETC.)

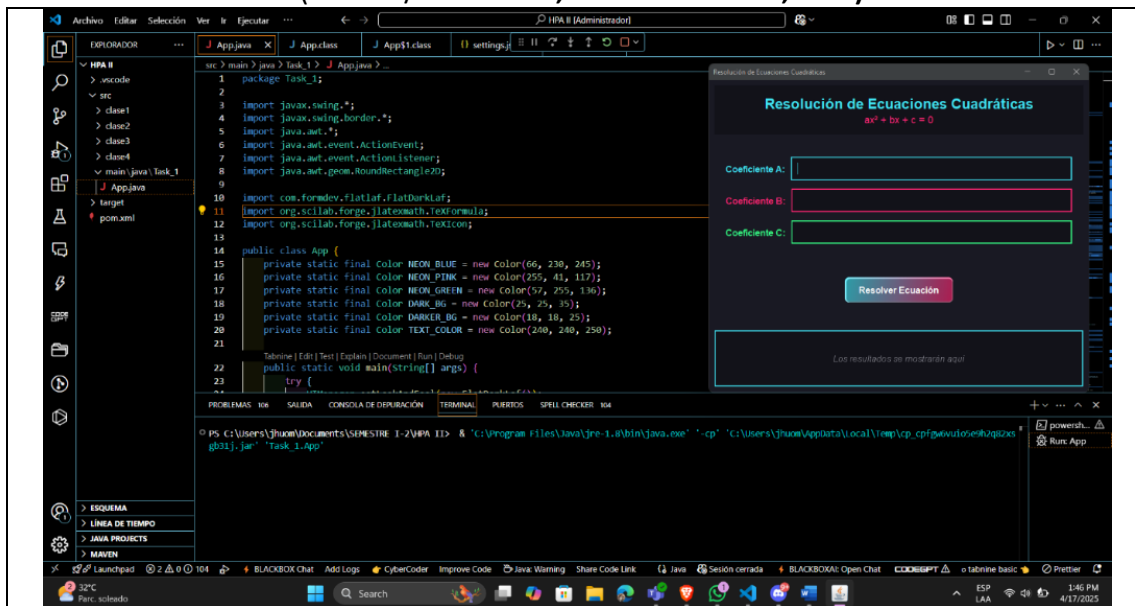


Ilustración 2: Ejecución inicial del programa en JAVA

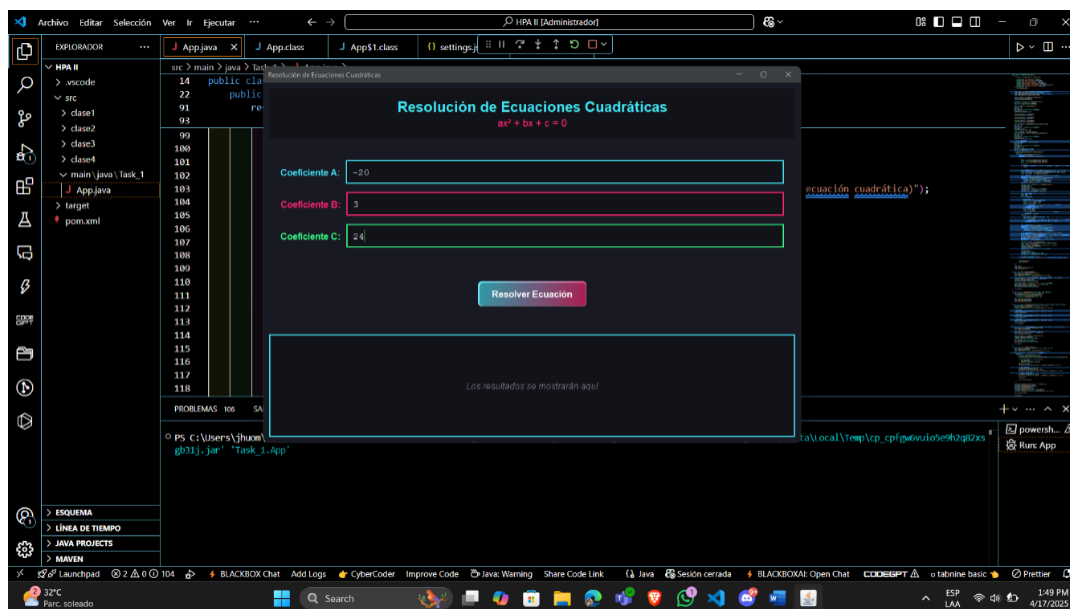


Ilustración 3: Inserción de valores a "A, B, C"

UNIVERSIDAD TECNOLÓGICA DE PANAMÁ
FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES
DEPARTAMENTO DE PROGRAMACIÓN DE COMPUTADORAS
GUÍA DE ACTIVIDAD N°. 01
(TAREA / PRÁCTICA, LABORATORIOS, ETC.)

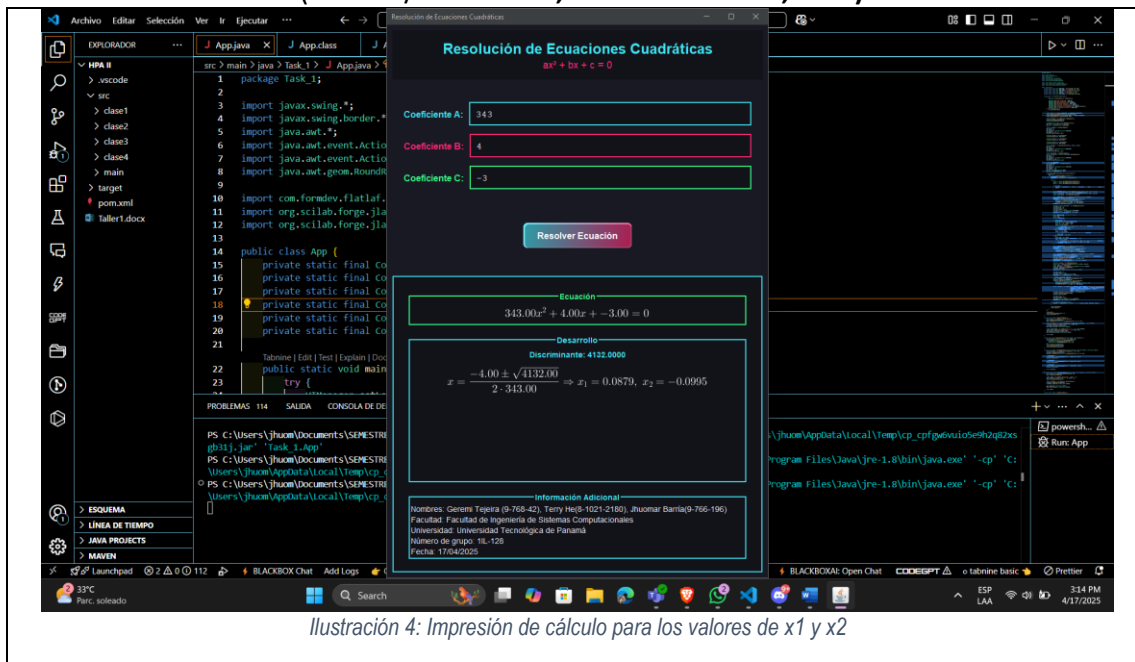


Ilustración 4: Impresión de cálculo para los valores de x1 y x2

Codificación en lenguaje java

Enlace del Código en GitHub: <https://github.com/Quantumquirkz/ScholarScript>

```
package Task_1;

import javax.swing.*;
import javax.swing.border.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.geom.RoundRectangle2D;

import com.formdev.flatlaf.FlatDarkLaf;
import org.scilab.forge.jlatexmath.TeXFormula;
import org.scilab.forge.jlatexmath.TeXIcon;

public class App {
    private static final Color NEON_BLUE = new Color(66, 230, 245);
    private static final Color NEON_PINK = new Color(255, 41, 117);
    private static final Color NEON_GREEN = new Color(57, 255, 136);
    private static final Color DARK_BG = new Color(25, 25, 35);
    private static final Color DARKER_BG = new Color(18, 18, 25);
    private static final Color TEXT_COLOR = new Color(240, 240, 250);

    public static void main(String[] args) {
```




UNIVERSIDAD TECNOLÓGICA DE PANAMÁ
FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES
DEPARTAMENTO DE PROGRAMACIÓN DE COMPUTADORAS
GUÍA DE ACTIVIDAD N°. 01
(TAREA / PRÁCTICA, LABORATORIOS, ETC.)

```
try {
    UIManager.setLookAndFeel(new FlatDarkLaf());

    UIManager.put("Panel.background", DARK_BG);
    UIManager.put("Label.foreground", TEXT_COLOR);
    UIManager.put("TextField.background", DARKER_BG);
    UIManager.put("TextField.foreground", TEXT_COLOR);
    UIManager.put("TextField.caretForeground", NEON_BLUE);
    UIManager.put("TextField.selectionBackground", NEON_BLUE.darker());
    UIManager.put("TitledBorder.titleColor", NEON_BLUE);
} catch (Exception ex) {
    ex.printStackTrace();
}

JFrame frame = new JFrame("Resolución de Ecuaciones Cuadráticas");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setExtendedState(JFrame.MAXIMIZED_BOTH); // Iniciar maximizado
frame.setMinimumSize(new Dimension(700, 500));

JPanel mainPanel = new JPanel(new GridBagLayout());
mainPanel.setBackground(DARK_BG);
frame.setContentPane(mainPanel);

GridBagConstraints gbc = new GridBagConstraints();
gbc.insets = new Insets(10, 10, 10, 10);
gbc.fill = GridBagConstraints.HORIZONTAL;

JPanel titlePanel = createTitlePanel();
gbc.gridx = 0;
gbc.gridy = 0;
gbc.gridwidth = GridBagConstraints.REMAINDER;
gbc.weightx = 1.0;
mainPanel.add(titlePanel, gbc);

JTextField aField = new JTextField();
styleTextField(aField, NEON_BLUE);

JTextField bField = new JTextField();
styleTextField(bField, NEON_PINK);

JTextField cField = new JTextField();
styleTextField(cField, NEON_GREEN);

JPanel inputPanel = createInputPanel(aField, bField, cField);
gbc.gridx = 0;
gbc.gridy = 1;
gbc.gridwidth = GridBagConstraints.REMAINDER;
gbc.weightx = 1.0;
```




UNIVERSIDAD TECNOLÓGICA DE PANAMÁ
FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES
DEPARTAMENTO DE PROGRAMACIÓN DE COMPUTADORAS
GUÍA DE ACTIVIDAD N°. 01
(TAREA / PRÁCTICA, LABORATORIOS, ETC.)

```
mainPanel.add(inputPanel, gbc);

JPanel buttonPanel = createButtonPanel();
JButton resolverBtn = (JButton) buttonPanel.getComponent(0);
gbc.gridx = 0;
gbc.gridy = 2;
gbc.gridwidth = GridBagConstraints.REMAINDER;
gbc.weightx = 1.0;
mainPanel.add(buttonPanel, gbc);

// Panel de resultados
JPanel resultadoPanel = createResultPanel();
gbc.gridx = 0;
gbc.gridy = 3;
gbc.gridwidth = GridBagConstraints.REMAINDER;
gbc.weightx = 1.0;
gbc.weighty = 1.0;
gbc.fill = GridBagConstraints.BOTH;
mainPanel.add(resultadoPanel, gbc);

resolverBtn.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        resultadoPanel.removeAll();
        resultadoPanel.setLayout(new GridBagLayout());

        try {
            double a = Double.parseDouble(aField.getText());
            double b = Double.parseDouble(bField.getText());
            double c = Double.parseDouble(cField.getText());

            if (a == 0) {
                mostrarError(resultadoPanel, "El coeficiente A no puede ser cero (no sería una ecuación cuadrática)");
                return;
            }

            String ecuacionOriginal = String.format("%.2fx^2 + %.2fx + %.2f = 0", a, b, c);
            TeXFormula formulaOriginal = new TeXFormula(ecuacionOriginal);
            TeXIcon iconOriginal = formulaOriginal.createTeXIcon(TeXFormula.SERIF, 20);
            JLabel labelOriginal = new JLabel(iconOriginal);

            JPanel ecuacionPanel = new JPanel(new FlowLayout(FlowLayout.CENTER));
            ecuacionPanel.setBackground(DARKER_BG);
            ecuacionPanel.setBorder(BorderFactory.createTitledBorder(
                BorderFactory.createLineBorder(NEON_GREEN, 2),
                "Ecuación", TitledBorder.CENTER, TitledBorder.TOP,
                new Font("SansSerif", Font.BOLD, 14), NEON_GREEN));
```

UNIVERSIDAD TECNOLÓGICA DE PANAMÁ
FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES
DEPARTAMENTO DE PROGRAMACIÓN DE COMPUTADORAS
GUÍA DE ACTIVIDAD N°. 01
(TAREA / PRÁCTICA, LABORATORIOS, ETC.)

```
ecuacionPanel.add(labelOriginal);

GridBagConstraints resultGbc = new GridBagConstraints();
resultGbc.gridx = 0;
resultGbc.gridy = 0;
resultGbc.weightx = 1.0;
resultGbc.fill = GridBagConstraints.HORIZONTAL;
resultGbc.insets = new Insets(10, 10, 5, 10);
resultadoPanel.add(ecuacionPanel, resultGbc);

double discriminante = b * b - 4 * a * c;
double dosA = 2 * a;
String latex;
Color colorResultado;

if (discriminante > 0) {
    double sqrtD = Math.sqrt(discriminante);
    double x1 = (-b + sqrtD) / dosA;
    double x2 = (-b - sqrtD) / dosA;
    latex = String.format(
        "x = \\frac{%2f}{%2f} \\pm \\sqrt{%2f}{2} \\cdot %2f \\Rightarrow x_1 = %4f, x_2 = %4f",
        -b, discriminante, a, x1, x2);
    colorResultado = NEON_BLUE;
} else if (discriminante == 0) {
    double x = -b / dosA;
    latex = String.format(
        "x = \\frac{%2f}{%2f} \\pm \\sqrt{0}{2} \\cdot %2f \\Rightarrow x_1 = x_2 = %4f",
        -b, a, x);
    colorResultado = NEON_GREEN;
} else {
    double real = -b / dosA;
    double imag = Math.sqrt(-discriminante) / dosA;
    latex = String.format(
        "x = \\frac{%2f}{%2f} \\pm \\sqrt{%2f}{2} \\cdot %2f = %2f \\pm %2fi \\Rightarrow x_1 = %4f"
        + "%4fi, x_2 = %4f - %4fi",
        -b, discriminante, a, real, imag, real, imag, real, imag);
    colorResultado = NEON_PINK;
}

JPanel desarrolloPanel = new JPanel(new BorderLayout());
desarrolloPanel.setBackground(DARKER_BG);
desarrolloPanel.setBorder(BorderFactory.createTitledBorder(
    BorderFactory.createLineBorder(colorResultado, 2),
    "Desarrollo", TitledBorder.CENTER, TitledBorder.TOP,
    new Font("SansSerif", Font.BOLD, 14), colorResultado));

JPanel discriminantePanel = new JPanel(new FlowLayout(FlowLayout.CENTER));
discriminantePanel.setBackground(DARKER_BG);
```



UNIVERSIDAD TECNOLÓGICA DE PANAMÁ
FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES
DEPARTAMENTO DE PROGRAMACIÓN DE COMPUTADORAS
GUÍA DE ACTIVIDAD N°. 01
(TAREA / PRÁCTICA, LABORATORIOS, ETC.)

```
JLabel discLabel = new JLabel("Discriminante: " + String.format("%.4f", discriminante));  
discLabel.setForeground(colorResultado);  
discLabel.setFont(new Font("SansSerif", Font.BOLD, 14));  
discriminantePanel.add(discLabel);  
desarrolloPanel.add(discriminantePanel, BorderLayout.NORTH);
```

```
TeXFormula formula = new TeXFormula(latex);  
TeXIcon icon = formula.createTeXIcon(TeXFormula.SERIF, 20);  
JLabel label = new JLabel(icon);  
JPanel latexPanel = new JPanel(new FlowLayout(FlowLayout.CENTER));  
latexPanel.setBackground(DARKER_BG);  
latexPanel.add(label);  
desarrolloPanel.add(latexPanel, BorderLayout.CENTER);
```

```
resultGbc.gridy = 1;  
resultGbc.weighty = 1.0;  
resultGbc.fill = GridBagConstraints.BOTH;  
resultadoPanel.add(desarrolloPanel, resultGbc);
```

```
// Agregar información adicional  
JPanel infoPanel = new JPanel(new GridLayout(0, 1));  
infoPanel.setBackground(DARKER_BG);  
infoPanel.setBorder(BorderFactory.createTitledBorder(  
    BorderFactory.createLineBorder(NEON_BLUE, 2),  
    "Información Adicional", TitledBorder.CENTER, TitledBorder.TOP,  
    new Font("SansSerif", Font.BOLD, 14), NEON_BLUE));
```

```
JLabel infoLabel = new JLabel("<html>Nombres: Geremi Tejeira (9-768-42), Terry He(8-1021-  
2180), Jhuomar Barria(9-766-196)<br>" +  
    "Facultad: Facultad de Ingeniería de Sistemas Computacionales<br>" +  
    "Universidad: Universidad Tecnológica de Panamá<br>" +  
    "Número de grupo: 1IL-128<br>" +  
    "Fecha: 17/04/2025</html>");  
infoLabel.setFont(new Font("SansSerif", Font.PLAIN, 14));  
infoLabel.setForeground(TEXT_COLOR);  
infoPanel.add(infoLabel);
```

```
resultGbc.gridy = 2;  
resultGbc.weighty = 0.0;  
resultGbc.fill = GridBagConstraints.HORIZONTAL;  
resultadoPanel.add(infoPanel, resultGbc);
```

```
} catch (NumberFormatException ex) {  
    mostrarError(resultadoPanel, "Por favor ingrese valores numéricos válidos");  
}
```

```
frame.revalidate();  
frame.repaint();
```



UNIVERSIDAD TECNOLÓGICA DE PANAMÁ
FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES
DEPARTAMENTO DE PROGRAMACIÓN DE COMPUTADORAS
GUÍA DE ACTIVIDAD N°. 01
(TAREA / PRÁCTICA, LABORATORIOS, ETC.)

```
}  
});  
  
frame.pack();  
frame.setLocationRelativeTo(null);  
frame.setVisible(true);  
}  
  
private static JPanel createTitlePanel() {  
    JPanel panel = new JPanel(new BorderLayout());  
    panel.setBackground(DARKER_BG);  
    panel.setBorder(BorderFactory.createEmptyBorder(15, 15, 15, 15));  
  
    JLabel titleLabel = new JLabel("Resolución de Ecuaciones Cuadráticas", JLabel.CENTER);  
    titleLabel.setFont(new Font("SansSerif", Font.BOLD, 26));  
    titleLabel.setForeground(NEON_BLUE);  
    panel.add(titleLabel, BorderLayout.CENTER);  
  
    JLabel subtitleLabel = new JLabel("ax2 + bx + c = 0", JLabel.CENTER);  
    subtitleLabel.setFont(new Font("SansSerif", Font.PLAIN, 18));  
    subtitleLabel.setForeground(NEON_PINK);  
    panel.add(subtitleLabel, BorderLayout.SOUTH);  
  
    return panel;  
}  
  
private static void styleTextField(JTextField field, Color neonColor) {  
    field.setFont(new Font("Monospaced", Font.PLAIN, 16));  
    field.setBorder(BorderFactory.createCompoundBorder(  
        BorderFactory.createLineBorder(neonColor, 2),  
        BorderFactory.createEmptyBorder(8, 10, 8, 10)  
    ));  
    field.setBackground(DARKER_BG);  
    field.setForeground(TEXT_COLOR);  
    field.setCaretColor(neonColor);  
}  
  
private static JPanel createInputPanel(JTextField aField, JTextField bField, JTextField cField) {  
    JPanel mainPanel = new JPanel(new GridLayout(3, 1, 10, 15));  
    mainPanel.setBackground(DARK_BG);  
    mainPanel.setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20));  
  
    // Coeficiente A  
    JPanel aPanel = createCoefficientPanel("A", NEON_BLUE, aField);  
    mainPanel.add(aPanel);  
  
    // Coeficiente B  
    JPanel bPanel = createCoefficientPanel("B", NEON_PINK, bField);
```



UNIVERSIDAD TECNOLÓGICA DE PANAMÁ
FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES
DEPARTAMENTO DE PROGRAMACIÓN DE COMPUTADORAS
GUÍA DE ACTIVIDAD N°. 01
(TAREA / PRÁCTICA, LABORATORIOS, ETC.)

```
mainPanel.add(bPanel);

// Coeficiente C
JPanel cPanel = createCoefficientPanel("C", NEON_GREEN, cField);
mainPanel.add(cPanel);

return mainPanel;
}

private static JPanel createCoefficientPanel(String coef, Color neonColor, JTextField field) {
    JPanel panel = new JPanel(new BorderLayout(10, 0));
    panel.setBackground(DARK_BG);

    JLabel label = new JLabel("Coeficiente " + coef + " :");
    label.setFont(new Font("SansSerif", Font.BOLD, 16));
    label.setForeground(neonColor);

    JPanel fieldPanel = new JPanel(new BorderLayout());
    fieldPanel.setBackground(DARK_BG);
    fieldPanel.add(field, BorderLayout.CENTER);

    panel.add(label, BorderLayout.WEST);
    panel.add(fieldPanel, BorderLayout.CENTER);

    return panel;
}

private static JPanel createButtonPanel() {
    JPanel panel = new JPanel(new FlowLayout(FlowLayout.CENTER, 20, 10));
    panel.setBackground(DARK_BG);
    panel.setBorder(BorderFactory.createEmptyBorder(10, 0, 20, 0));

    JButton resolverBtn = new JButton("Resolver Ecuación") {
        @Override
        protected void paintComponent(Graphics g) {
            Graphics2D g2 = (Graphics2D) g.create();
            g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
                RenderingHints.VALUE_ANTIALIAS_ON);

            GradientPaint gradient = new GradientPaint(
                0, 0, NEON_BLUE,
                getWidth(), getHeight(), NEON_PINK
            );
            g2.setPaint(gradient);
            g2.fill(new RoundRectangle2D.Float(0, 0, getWidth(), getHeight(), 15, 15));

            // Borde brillante
            g2.setStroke(new BasicStroke(2f));
        }
    };
}
```



UNIVERSIDAD TECNOLÓGICA DE PANAMÁ
FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES
DEPARTAMENTO DE PROGRAMACIÓN DE COMPUTADORAS
GUÍA DE ACTIVIDAD N°. 01
(TAREA / PRÁCTICA, LABORATORIOS, ETC.)

```
g2.setColor(new Color(255, 255, 255, 100));
g2.draw(new RoundRectangle2D.Float(1, 1, getWidth()-2, getHeight()-2, 14, 14));

// Efecto de sombra sutil
g2.setColor(new Color(0, 0, 0, 80));
g2.fill(new RoundRectangle2D.Float(3, 3, getWidth()-3, getHeight()-3, 14, 14));

g2.dispose();

super.paintComponent(g);
}
};

resolverBtn.setFont(new Font("SansSerif", Font.BOLD, 16));
resolverBtn.setForeground(Color.WHITE);
resolverBtn.setContentAreaFilled(false);
resolverBtn.setBorderPainted(false);
resolverBtn.setFocusPainted(false);
resolverBtn.setCursor(new Cursor(Cursor.HAND_CURSOR));
resolverBtn.setBorder(BorderFactory.createEmptyBorder(12, 25, 12, 25));

panel.add(resolverBtn);

return panel;
}

private static JPanel createResultPanel() {
    JPanel panel = new JPanel(new BorderLayout());
    panel.setBackground(DARKER_BG);
    panel.setBorder(BorderFactory.createCompoundBorder(
        BorderFactory.createLineBorder(NEON_BLUE, 2),
        BorderFactory.createEmptyBorder(15, 15, 15, 15)
    ));

    JLabel placeholder = new JLabel("Los resultados se mostrarán aquí", JLabel.CENTER);
    placeholder.setFont(new Font("SansSerif", Font.ITALIC, 16));
    placeholder.setForeground(new Color(TEXT_COLOR.getRed(), TEXT_COLOR.getGreen(),
    TEXT_COLOR.getBlue(), 150));
    panel.add(placeholder, BorderLayout.CENTER);

    return panel;
}

private static void mostrarError(JPanel panel, String mensaje) {
    panel.setLayout(new BorderLayout());

    JPanel errorPanel = new JPanel();
    errorPanel.setBackground(DARKER_BG);
```



UNIVERSIDAD TECNOLÓGICA DE PANAMÁ
FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES
DEPARTAMENTO DE PROGRAMACIÓN DE COMPUTADORAS
GUÍA DE ACTIVIDAD N°. 01
(TAREA / PRÁCTICA, LABORATORIOS, ETC.)

```
errorPanel.setBorder(BorderFactory.createLineBorder(NEON_PINK, 2));  
errorPanel.setLayout(new FlowLayout(FlowLayout.CENTER));
```

```
JLabel errorLabel = new JLabel(mensaje);  
errorLabel.setFont(new Font("SansSerif", Font.BOLD, 16));  
errorLabel.setForeground(NEON_PINK);  
errorPanel.add(errorLabel);
```

```
panel.add(errorPanel, BorderLayout.CENTER);
```

```
}  
}
```

I. BIBLIOGRAFIA:

Libros

1. **Deitel, P., & Deitel, H. (2018).** Java: How to Program (Early Objects), 11th Edition. Pearson.
 - Este libro proporciona una introducción completa a la programación en Java, cubriendo desde los conceptos básicos hasta temas avanzados. Fue fundamental para entender la estructura del lenguaje y la creación de interfaces gráficas con Swing.
2. **Bloch, J. (2018).** Effective Java (3rd Edition). Addison-Wesley Professional.
 - Este libro ofrece mejores prácticas y patrones de diseño en Java, lo cual fue útil para escribir código limpio y eficiente.

Documentación Oficial

1. **Oracle. (2023).** Java SE Documentation. Oracle Corporation.
 - La documentación oficial de Java fue crucial para entender el uso de clases y métodos específicos, así como para resolver dudas sobre la API de Swing.
 - [Java SE Documentation](#)
2. **FormDev Software GmbH. (2023).** FlatLaf Documentation. FormDev Software GmbH.
 - La documentación de FlatLaf me ayudó a integrar y personalizar el tema oscuro en la aplicación.
 - [FlatLaf Documentation](#)
3. **Scilab Enterprises. (2023).** JLaTeXMath Documentation. Scilab Enterprises.
 - La documentación de JLaTeXMath fue esencial para renderizar fórmulas matemáticas en la interfaz gráfica.
 - [JLaTeXMath Documentation](#)

Páginas Web y Tutoriales

1. **Baeldung. (2023).** Java Swing Tutorials. Baeldung.
 - Este sitio web ofrece tutoriales detallados sobre cómo trabajar con Swing en Java, lo cual fue muy útil para crear y personalizar los componentes de la interfaz gráfica.
 - [Baeldung Java Swing Tutorials](#)
2. **GeeksforGeeks. (2023).** Java Swing Tutorial. GeeksforGeeks.
 - GeeksforGeeks proporciona ejemplos prácticos y explicaciones claras sobre el uso de Swing, lo cual facilitó la comprensión de conceptos como JFrame, JPanel, y JButton.
 - [GeeksforGeeks Java Swing Tutorial](#)
3. **Stack Overflow. (2023).** Java Swing Questions and Answers. Stack Overflow.
 - Stack Overflow fue una herramienta invaluable para resolver problemas específicos y obtener soluciones de la comunidad de desarrolladores.
 - [Stack Overflow Java Swing](#)

Videos de YouTube

1. **TheNetNinja. (2023).** Java Swing Tutorial for Beginners. YouTube.



UNIVERSIDAD TECNOLÓGICA DE PANAMÁ
FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES
DEPARTAMENTO DE PROGRAMACIÓN DE COMPUTADORAS
GUÍA DE ACTIVIDAD N°. 01
(TAREA / PRÁCTICA, LABORATORIOS, ETC.)

- Este canal ofrece una serie de videos que explican paso a paso cómo crear aplicaciones gráficas en Java utilizando Swing. Fue muy útil para entender la estructura básica de una aplicación Swing.
- [TheNetNinja Java Swing Tutorial](#)
- 2. **ProgrammingKnowledge. (2023).** Java GUI Tutorial. YouTube.
 - Este canal proporciona tutoriales detallados sobre cómo trabajar con componentes gráficos en Java, incluyendo ejemplos prácticos y explicaciones claras.
 - [ProgrammingKnowledge Java GUI Tutorial](#)

Otros Recursos

1. **GitHub Repositories. (2023).** Open-Source Java Projects. GitHub.
 - Explorar repositorios de código abierto en GitHub me permitió ver ejemplos reales de aplicaciones Java y entender cómo se estructuran proyectos más grandes.
 - [GitHub Java Projects](#)

BIBLIOGRAFÍA.

AUTOR	NOMBRE DEL LIBRO	EDITORIAL
Luis Joyanes Aguilar	Programación en JAVA 2, algoritmos, estructuras de datos y POO	Mc Graw Hill, 2002, Primera edición. *****
Luis Joyanes Aguilar, Ignacio Zahonero Martinez	Programación en C, C++, JAVA y UML	Mc Graw Hill, 2010, Primera edición.
Douglas Bell, Mike Parr	Java para Estudiantes	Prentice Hall, 2011, Septima Edición
Deitel y Deitel	Cómo Programar en Java	Prentice Hall, 2012, Novena edición
Olinda de Barraza, Felicitad de Krol, Ludia de Meléndez, Mitzi de Velázquez	Introducción a la programación Orientada a Objetos	Imprenta Universidad Tecnológica de Panamá, 2013
David J. Barnes, Michael Kolling	Programación Orientada a Objetos con Java usando BlueJ	Pearson, 2013

BIBLIOTECA VIRTUAL UTP : -EN EL LIBRERÍA DIGITAL E-LIBRO

AUTOR	NOMBRE DEL LIBRO	EDITORIAL
Héctor Flórez Fernández	Programación Orientada a Objetos usando Java	Ecoe Ediciones (2012)
José Vélez, Alberto Peña y Gortazar, Patxi	Diseñar y programar, todo es empezar: una introducción a la Programación Orientada a Objetos usando UML y Java	Dykinson (2011)
Casanova, Assumpcio; Marqués, Francisco; Prieto, Natividad	Empezar a Programar usando Java	Editorial de la Universidad Politécnica de Valencia; 2012



UNIVERSIDAD TECNOLÓGICA DE PANAMÁ
FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES
DEPARTAMENTO DE PROGRAMACIÓN DE COMPUTADORAS
GUÍA DE ACTIVIDAD N°. 01
(TAREA / PRÁCTICA, LABORATORIOS, ETC.)

J. RÚBRICAS:

<u>Aspectos a Evaluar</u>	<u>Puntaje Máximo 100%</u>	<u>Puntaje Obtenido</u>
Estilo de escritura	5	
Indentar	5	
*Claridad en la Escritura	5	
Breve documentación a través de comentarios	5	
Secuencia lógica	35	
Prueba de Escritorio	25	
*Código en Lenguaje Java	10	
*Optimización de instrucciones	5	
*Puntualidad en la entrega	5	
Total	100	

Nota: *
Queda a discreción del docente, de acuerdo con el tema, la utilización de este criterio

Conclusión

El proyecto sobre el uso de la fórmula cuadrática en Java está estructurado en dos clases principales: `OperadorCuadratico.java` y `App.java` (programa principal). Esta organización modular permite una mejor estructuración del código, facilitando la reutilización, la organización y la legibilidad del programa. En la clase `App.java`, se utiliza la librería `import java.util.Scanner;` para permitir al usuario insertar los valores de los coeficientes `a`, `b` y `c`. Estos valores son luego utilizados para calcular las soluciones esperadas de la ecuación cuadrática mediante la fórmula cuadrática, implementada en la clase `OperadorCuadratico.java`. La clase `OperadorCuadratico.java` se encarga de calcular el discriminante y determinar las raíces de la ecuación.

Además, se ha agregado una estructura repetitiva `do-while` en la clase `App.java` para permitir al usuario decidir si desea repetir el proceso del programa o no. Esta característica mejora la interacción del usuario, permitiendo múltiples cálculos sin necesidad de reiniciar la aplicación. Este proyecto no solo resuelve correctamente ecuaciones cuadráticas reales, sino que también sienta una base sólida para futuras extensiones. Por ejemplo, se podría ampliar el programa para manejar raíces complejas, implementar validaciones avanzadas de datos, o incluso integrar una interfaz gráfica de usuario (GUI) para mejorar la experiencia del usuario. La estructura modular y la claridad del código facilitan estas posibles mejoras y extensiones, demostrando la robustez y flexibilidad del diseño del proyecto.