

**INSTITUTO SUPERIOR TECNOLÓGICO MAYOR “PEDRO  
TRAVERSARI”**



**CARRERA DE DESARROLLO DE SOFTWARE**

**NIVEL: PRIMERO**

**MEMORIA DEL PROYECTO ACADÉMICO DE INNOVACIÓN TECNOLÓGICA**

**TÍTULO:**

Sistema de Resolución de Problemas Matemáticos con Interfaz Gráfica en  
Java

**AUTORES:**

- Ronald Subía
- Shirley Pizarro
- Manuel Collantes
- Sahid Changoluisa

**TUTORA: Msgtr: Estefani Sanchez**

**QUITO-ECUADOR**

**2025**

## Tabla de contenido

Sistema de Resolución de Problemas Matemáticos con Interfaz Gráfica en Java .....	1
Marco Teórico .....	3
Fundamentos de Programación y Algoritmos .....	3
Características principales de un algoritmo .....	4
Pseudocódigo y Diagramas de Flujo .....	4
Pseudocódigo .....	4
Diagramas de Flujo .....	4
Su utilidad radica en .....	4
Introducción a Java y su Ecosistema .....	4
El ecosistema de Java comprende varios elementos: .....	5
1. JDK (Java Development Kit) .....	5
2. JVM (Java Virtual Machine) .....	5
3. JRE (Java Runtime Environment) .....	5
4. IDEs (Entornos de Desarrollo) .....	5
Estructura básica de un programa Java .....	5
¿Qué significa cada parte? .....	5
Operadores Aritméticos .....	6
Operador (+) .....	6
Operador (-) .....	6
Operador (*) .....	6
División (/) .....	6
Módulo (%) .....	6
Orden general de precedencia (de mayor a menor) .....	6
Paréntesis .....	6
Operadores unarios .....	6
Exponenciación .....	6
Suma y resta .....	6
Operadores de comparación .....	6
Operadores lógicos .....	6
Estructura if-else y if-else anidados: Conceptos, Usos y Desafíos .....	7
Factores y Productos Notables .....	7
Tipos y Estrategias de Factorización .....	7

Factorización de Polinomios y Matrices.....	7
Enseñanza y Dificultades Comunes .....	7
Errores Frecuentes en Estudiantes .....	7
Trinomios cuadrados perfectos.....	8
Propiedades matemáticas y direcciones de investigación.....	8
Ecuaciones de primer grado .....	8
Función y sus características .....	8
¿Qué es una función?.....	8
Dominio de una función .....	9
Rango de una función .....	9
¿Qué es una interfaz gráfica?.....	9
¿Qué es Windowbuilder?.....	9
Concepto básico matemáticos .....	9
Regla de Cramer .....	9
Una interfaz se crea de la siguiente manera: .....	10
Elementos claves en el desarrollo de una interfaz.....	10
Palabras claves/reservadas usadas en edición de interfaz .....	10
Palabra this.....	10
Acción set .....	11
Palabra getText().....	11
this.setResizable(false) .....	11
Clase.setVisible(true).....	11
Clase.this.dispose() .....	11
this.setLocationRelativeTo(null) .....	11
Implementación de la regla de Cramer.....	11
Implementación en la interfaz .....	12

## Marco Teórico

### **Fundamentos de Programación y Algoritmos**

- La programación es el proceso mediante el cual se diseñan y construyen soluciones para ser ejecutadas por un computador. Estas

soluciones se expresan mediante instrucciones lógicas y ordenadas (Joyanes, 2016).

- Un algoritmo es un conjunto de pasos precisos, ordenados y finitos que permiten resolver un problema específico (Cormen et al., 2009).

### **Características principales de un algoritmo**

- Preciso y claro
- Ordenado
- Finito
- Determinado
- Orientado a la solución de un problema

Los algoritmos constituyen la base para el desarrollo de cualquier programa informático.

## **Pseudocódigo y Diagramas de Flujo**

### **Pseudocódigo**

El pseudocódigo es una representación textual de un algoritmo utilizando un lenguaje cercano al natural y sin la rigurosidad sintáctica de un lenguaje de programación formal (Joyanes, 2016). Su finalidad es facilitar la planificación y comprensión de las soluciones antes de escribir código.

### Diagramas de Flujo

Los diagramas de flujo son representaciones visuales de los pasos de un algoritmo mediante símbolos estandarizados como óvalos (inicio/fin), rectángulos (procesos), rombos (decisiones) y flechas que indican la secuencia lógica (Sierra, 2014).

### **Su utilidad radica en**

- Comprender el proceso completo
- Identificar errores lógicos
- Explicar procesos de manera gráfica
- Lenguaje de programación. - El lenguaje de programación permite a los desarrolladores de software crear programas/aplicaciones o sitios web con algoritmos finitos que tienen con finalidad solucionar problemas de los usuarios.

## **Introducción a Java y su Ecosistema**

- Java es un lenguaje de programación orientado a objetos, robusto y multiplataforma, diseñado bajo el principio “Write once, run anywhere” (Oracle, 2023). Esto significa que un programa Java puede ejecutarse en cualquier sistema operativo que cuente con una Máquina Virtual de Java (JVM).

- Para los colaboradores IBM (s.f), “Java es un lenguaje de desarrollo basado en el lenguaje C/C++, razón por la cual su sintaxis es similar. La tecnología de Java se ejecuta en conjunto entre el lenguaje y el software. Java es muy beneficioso en el mercado laboral debido a su buen control en la base de datos en el ámbito web”.

## **El ecosistema de Java comprende varios elementos:**

### **1. JDK (Java Development Kit)**

- Incluye herramientas para desarrollar aplicaciones Java como el compilador javac, la JVM, librerías estándar y otras utilidades.

### **2. JVM (Java Virtual Machine)**

- Es la encargada de ejecutar el bytecode generado por el compilador, actuando como intermediaria entre el programa y el sistema operativo (Oracle, 2023).

### **3. JRE (Java Runtime Environment)**

- Proporciona el entorno necesario para ejecutar aplicaciones Java, aunque no permite desarrollarlas, ya que no incluye el compilador.

### **4. IDEs (Entornos de Desarrollo)**

Son herramientas que facilitan la escritura, análisis y depuración del código. Entre los más comunes se encuentran IntelliJ IDEA, Eclipse y NetBeans (Deitel & Deitel, 2017).

## **Estructura básica de un programa Java**

Todo programa en Java tiene una estructura mínima.

Ejemplo:

```
public class Main {
    public static void main(String[] args) {
        // Código del programa
    }
}
```

## **¿Qué significa cada parte?**

- class Main: define una clase
- public static void main: es el método principal
- main: es el punto donde empieza el programa
- { }: delimitan bloques de código

Sintaxis y convenciones de código

Sintaxis básica

- Cada instrucción termina con;
- Java distingue mayúsculas y minúsculas
- Main ≠ main

- El código se escribe dentro de { }

## Operadores Aritméticos

Los operadores aritméticos son símbolos que representan operaciones matemáticas básicas como la suma, resta, multiplicación y división.

### Operador (+)

El operador + suma los números presentes a la izquierda y derecha del operador. Recalcamos lo de números porque no tendría sentido sumar dos cadenas de texto, o dos listas, pero en Python es posible hacer este tipo de cosas.

### Operador (-)

El operador - resta los números presentes a la izquierda y derecha del operador. A diferencia el operador + en este caso no podemos restar cadenas o listas.

### Operador (\*)

El operador \* multiplica los números presentes a la izquierda y derecha del operador.

### División (/)

Este operador divide un valor por otro. Es importante notar que la división entre enteros devuelve un entero, truncando cualquier parte decimal.

### Módulo (%)

Este operador devuelve el resto de una división entera. Es útil para determinar si un número es divisible por otro o para realizar operaciones cíclicas.

## Orden general de precedencia (de mayor a menor)

### Paréntesis

Las operaciones dentro de los paréntesis se resuelven primero.

### Operadores unarios

Operadores como la negación (-) y el incremento/decremento (++, --).

### Exponenciación

Operaciones de potencia (ej. \*\*).

Multiplicación, división y módulo: Se evalúan de izquierda a derecha.

### Suma y resta

Se evalúan de izquierda a derecha.

### Operadores de comparación

Mayor que (>), menor que (<), etc.

### Operadores lógicos

AND y OR. AND generalmente tiene mayor precedencia que OR.

## Estructura if-else y if-else anidados: Conceptos, Usos y Desafíos

Las estructuras if-else y los if-else anidados son fundamentales en la programación, permitiendo tomar decisiones y controlar el flujo de ejecución según condiciones lógicas. Estas estructuras se encuentran en casi todos los lenguajes de programación y son esenciales para resolver problemas complejos y modelar lógica condicional.

### Conceptos Básicos y Sintaxis

La estructura if-else evalúa una condición: si es verdadera, ejecuta un bloque de código; si es falsa, ejecuta otro bloque alternativo. Los if-else anidados consisten en colocar una estructura if-else dentro de otra, permitiendo múltiples niveles de decisión y mayor complejidad en la lógica.

## Factores y Productos Notables

### Perspectivas desde la Investigación Matemática y Educativa

La factorización y los productos notables son conceptos fundamentales en álgebra, esenciales tanto en la teoría matemática como en la enseñanza escolar. La investigación reciente abarca desde enfoques avanzados de factorización en álgebra abstracta y matrices, hasta estrategias didácticas para mejorar la comprensión de estos temas en estudiantes de secundaria.

## Tipos y Estrategias de Factorización

### Factorización de Polinomios y Matrices

Se han desarrollado algoritmos avanzados para la factorización de polinomios usando matrices, mejorando la eficiencia y reduciendo el tamaño de los factores, especialmente en polinomios sumando-reducibles 35. Además, se exploran factorizaciones en espacios de productos escalares y en contextos de álgebra abstracta, como módulos de Lie y álgebras de Banach 12416.

Productos Notables y Representaciones: Los productos notables, como el cuadrado de un binomio o la diferencia de cuadrados, se abordan tanto desde la perspectiva algebraica clásica como mediante representaciones matriciales y tensoriales.

## Enseñanza y Dificultades Comunes

### Errores Frecuentes en Estudiantes

Los estudiantes suelen confundir la factorización de polinomios con la resolución de ecuaciones, agrupar incorrectamente monomios y aplicar mal las reglas prácticas de productos notables 11.

Estrategias Didácticas Innovadoras: El uso de materiales concretos, representaciones geométricas y software interactivo ha mostrado ser efectivo para mejorar la comprensión y reducir errores conceptuales en la enseñanza de factorización y productos notables.

## Trinomios cuadrados perfectos

Propiedades matemáticas y direcciones de investigación

Un trinomio cuadrado perfecto es una expresión cuadrática de la forma  $(a^2 + 2ab + b^2 = (a + b)^2)$ , fundamental en álgebra para factorizar y resolver ecuaciones cuadráticas. La investigación extiende este concepto a la teoría de números, la combinatoria y las estructuras algebraicas, explorando cuándo los trinomios o expresiones relacionadas producen cuadrados perfectos y sus implicaciones matemáticas más amplias.

## Ecuaciones de primer grado

El lenguaje algebraico utiliza letras, números y símbolos matemáticos para representar situaciones, relaciones y problemas de la vida real.

Las letras representan variables (valores desconocidos) y los números representan constantes.

Reglas básicas para despejar una variable

Para despejar una variable se aplican las operaciones inversas:

Sumar ↔ Restar

Multiplicar ↔ Dividir

Métodos de resolución de sistemas de ecuaciones

Método por determinantes (Regla de Cramer)

Se utiliza en sistemas de dos ecuaciones con dos incógnitas.

Ecuaciones de Segundo Grado

Definición

Una ecuación de segundo grado es una ecuación donde la variable tiene exponente 2.

## Función y sus características

### ¿Qué es una función?

Una función es una relación entre dos conjuntos donde a cada valor de x (variable independiente) le corresponde un único valor de y (variable dependiente).

## Dominio de una función

El dominio es el conjunto de todos los valores de  $x$  para los cuales la función está definida.

## Rango de una función

El rango (o recorrido) es el conjunto de valores que puede tomar  $y$  al evaluar la función.

## ¿Qué es una interfaz gráfica?

Es la forma visual en que interactuamos con un programa o sistema operativo, usando elementos como **iconos, botones, menús y ventanas**, en lugar de comandos de texto, para realizar tareas de forma más intuitiva y amigable mediante el ratón o el tacto.

## ¿Qué es Windowbuilder?

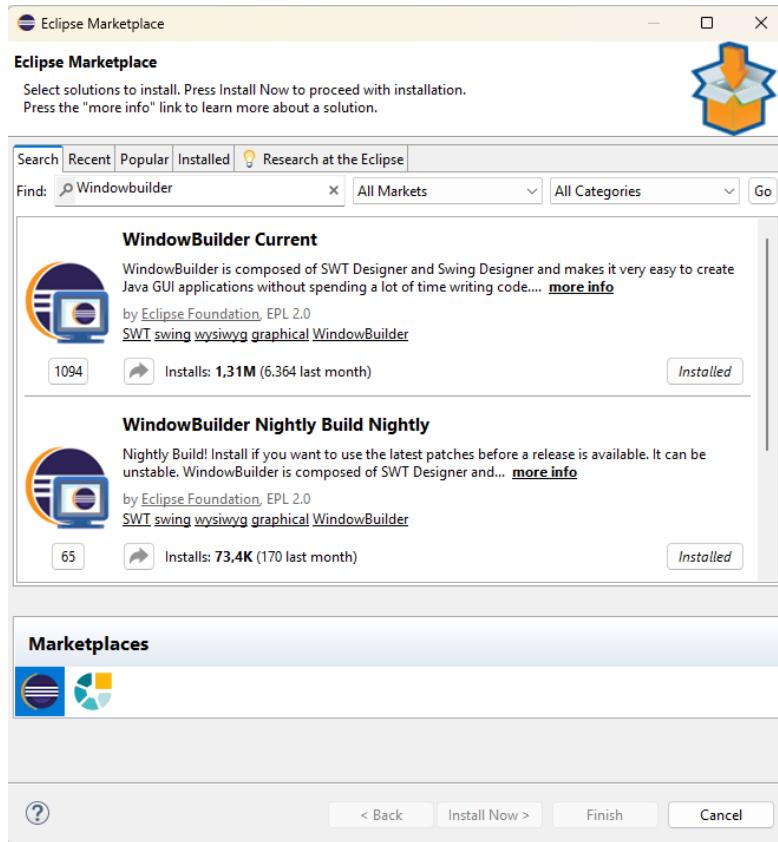
WindowBuilder es un **plugin de código abierto para el IDE Eclipse que permite diseñar interfaces gráficas de usuario (GUI) en Java** de forma visual y rápida, utilizando un método de **arrastrar y soltar (drag-and-drop)** en una vista WYSIWYG, y generando automáticamente el código Java necesario.

## Concepto básico matemáticos

### Regla de Cramer

Creación de interfaz con **WindowBuilder**

Para crear la interfaz gráfica de la aplicación se utilizó la herramienta del IDE eclipse **Windowbuilder** que se descarga de forma gratuita en “eclipse Marketplace”



### Una interfaz se crea de la siguiente manera:

Se debe crear un proyecto de java en donde se alojará un paquete y dentro del mismo se creará un Jframe.

El “design mode”, o modo de diseño, permite colocar elementos en un panel, el cual debe estar definido como capa absoluta o “absolute layout”.

### Elementos claves en el desarrollo de una interfaz

**Botones/JButton:** Ejecutan acciones al presionarlos con el ratón, al darles doble click en el modo diseño estos abrirán un “listener” que permitirá ejecutar acciones logias implementadas dentro de su clase.

**Etiquetas/JLabels:** Permiten colocar textos dentro de las ventanas y no realizan acción alguna cuando se las selecciona.

**Campos de texto/JTextfield:** Áreas que la entrada y salida de datos que se presentaran a los usuarios, se puede habilitar que reciban o no datos desde el editor, asi como asignarles una variable para que realicen diferentes acciones.

### Palabras claves/reservadas usadas en edición de interfaz

#### Palabra this

Se traduce como “esto” y se utiliza para llamar constructores o la clase/instancias en la que se esta desarrollando la ventana, permite realizar acciones específicas.

### **Acción set**

Se traduce como “colocar/poner” y se antepone a acciones que va a realizar la ventana y como se ejecutan.

### **Palabra getText()**

Permite tomar datos especificando la variable de un campo de texto en específico.

### **this.setResizable(false)**

Impide que se aumente o reduzca la ventana y quede del mismo tamaño.

### **Clase.setVisible(true)**

Permite que una ventana en otra clase se abra.

### **Clase.this.dispose()**

Hace que la ventana actual se cierre al abrir otra.

### **this.setLocationRelativeTo(null)**

Abre al programa desde el centro de la pantalla.

## **Implementación de la regla de Cramer**

La regla de Cramer es un método algebraico que permite la resolución de sistema de ecuaciones tanto 2x2 como 3x3 mediante la resolución de determinantes de matrices.

### **Resolución de un sistema de ecuaciones 2x2**

Ejemplo:

$$\text{Para resolver el sistema de ecuaciones } A = \begin{cases} 5x + 2y = 3 \\ -3 + 3 = 15 \end{cases}$$

Se crea una matriz 2x2 con las incógnitas x e y, luego se resuelve su determinante.

$$\begin{pmatrix} x_1 & y_1 \\ x_2 & y_2 \end{pmatrix} = (x_1 * y_2) - (y_1 * x_2)$$

$$\begin{pmatrix} 5 & 2 \\ -3 & 3 \end{pmatrix} = (5 * 3) - (2 * (-3)) = 15 + 6 = 21$$

Luego se reemplaza x por los términos independientes para encontrar la determinante de x.

$$\begin{pmatrix} t_1 & y_1 \\ t_2 & y_2 \end{pmatrix} = (t_1 * y_2) - (y_1 * t_2)$$

$$\begin{pmatrix} 3 & 2 \\ 15 & 3 \end{pmatrix} = (3 * 3) - (2 * 15) = 9 - 30 = -21$$

Se repite con la incógnita y para el determinante de y.

$$\begin{pmatrix} x_1 & t_1 \\ x_2 & t_2 \end{pmatrix} = (x_1 * t_2) - (t_1 * x_2)$$

$$\begin{pmatrix} 5 & 3 \\ -3 & 15 \end{pmatrix} = (5 * 15) - (3 * (-3)) = 75 + 9 = 84$$

D=21 (determinante de la matriz x, y)

Dx=-21(determinante de la incógnita x)

Dy=84(determinante de la incógnita y)

Para encontrar los valores de x e y se debe dividir la matriz de cada incógnita sobre la matriz formada por x, y.

$$x = \frac{dx}{d} \quad y = \frac{dy}{d}$$

$$x = \frac{84}{21} \quad y = \frac{-21}{21}$$

$$\mathbf{x = 4} \quad \mathbf{y = -1}$$

### **Implementación en la interfaz**

Para realizar el método se debe crear de forma ordenada campos de textos con variables que indiquen su función para que puedan tomar los valores por teclado y se pueda imprimir el resultado.

Se debe generar un botón que calcule el resultado y lo imprima, en el se ejecutara la regla de Cramer.

Para que el botón pueda generar el resultado deben los campos de texto asignarse con los datos x1, x2, y1, y2, t1 y t2. Para que el botón tome el dato de los números escritos en los campos se debe declarar una nueva variable de tipo float o double, ya que solo se puede realizar el ejercicio si son datos numéricos. Debido a que los campos solo toman valores de tipo string o cadena, se los debe transformar al tipo que se va a utilizar para el proceso, en este caso se usa la sentencia “Parse” junto al tipo de dato a almacenar.

En este caso para tomar el valor del campo x1 en una nueva variable X1 de tipo flotante seria la siguiente sentencia: float X1= Float.parseFloat(x1.getText()) y asi con las que siguen.

Para la resolución se crean variables que almacenen las determinantes de la matriz formada por x e y, la determinante de x y la de y.

$$d = (x1 * y2) - (y1 * x2)$$

$$dx = (t1 * y2) - (y1 * t2)$$

$$dy = (x1 * t2) - (t1 * x2)$$

Nuevas variables para el resultado de x e y

$$x = \frac{dx}{d} \quad y = \frac{dy}{d}$$

Para imprimir un resultado se debe tener en cuenta que un campo de texto solo toma e imprime datos de tipo cadena o string, para realizar la acción se debe hacer:

Una variable tipo String que concatene los resultados y lo que se va a imprimir

```
String respuesta = "x= "+x+"\n"+ "y= "+y
```

Luego se debe llamar al campo de texto que servirá para imprimir el resultado, en este caso el campo Res.

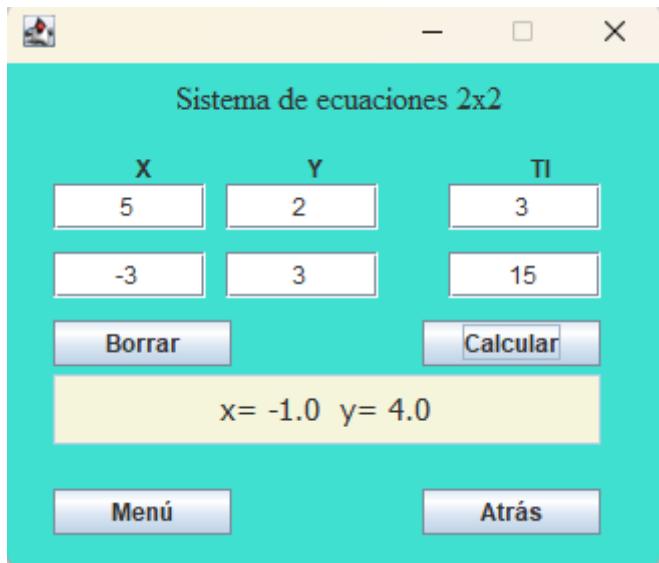
Res.setText(respuesta) e imprimirá el resultado.

```
JButton btnNewButton_3 = new JButton("Calcular");
btnNewButton_3.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        //x
        float X1= Float.parseFloat(x1.getText());
        float X2= Float.parseFloat(x2.getText());
        //y
        float Y1= Float.parseFloat(y1.getText());
        float Y2= Float.parseFloat(y2.getText());
        //terminos independientes
        float T1= Float.parseFloat(t1.getText());
        float T2= Float.parseFloat(t2.getText());
        //Determinante de la matriz, x e y
        float d = (X1*Y2)-(X2*Y1);
        float dx = (T1*Y2)-(T2*Y1);
        float dy = (X1*T2)-(X2*T1);
        //Resultados
        float x = dx/d;
        float y = dy/d;
        //Transforma los resultados a String para que puedan
        //imprimirse en el campo de texto "Res"
        String respuesta = "x= "+x+"\n"+ "y= "+y;
        Res.setText(respuesta);
    }
}
```

Para borrar los datos almacenados como impresos se debe crear un botón que ejecute el proceso de set.Textfield("") en cada campo

```
JButton btnNewButton_2 = new JButton("Borrar");
btnNewButton_2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        x1.setText("");
        x2.setText("");
        y1.setText("");
        y2.setText("");
        t1.setText("");
        t2.setText("");
        Res.setText("");
    }
..
```

**Ejecución en proceso:**



Para un sistema de ecuaciones 3x3 es implementar una incógnita z donde se crearía una matriz 3x3 para resolver el ejercicio.

Este código corresponde a una interfaz gráfica en Java usando Swing para calcular el **área y perímetro de distintas figuras geométricas**. De forma breve, su funcionamiento es el siguiente:

- **Ventana principal (figuras)**  
Crea un `JFrame` con tamaño fijo (800x550), centrado en pantalla, con fondo azul y título “Figuras”.
- **Navegación**  
Incluye un botón “Menu” que abre otra ventana (`ventana1`) y cierra la actual.
- **Figuras mostradas**  
Usa `JLabel` con `Imagen Icon` para mostrar imágenes de:
  - Cuadrado
  - Rectángulo
  - Círculo
  - Triángulo
- **Entrada de datos**  
Emplea varios `JTextField` para ingresar medidas:
  - Lados, base, altura y radio según la figura.
- **Cálculos**  
Cada figura tiene un botón “Calcular” con su `ActionListener`:
  - **Cuadrado:** área  $L \times LL \times LL \times L$  y perímetro  $4 \times L \times L \times L$
  - **Rectángulo:** área  $base \times altura$   $base \times altura$   $base \times altura$  y perímetro  $2 \times (base + altura)$   $2 \times (base + altura)$   $2 \times (base + altura)$
  - **Círculo:** área  $\pi \times R^2$   $\pi \times R^2$  y perímetro  $2 \times \pi \times R$   $2 \times \pi \times R$   $2 \times \pi \times R$
  - **Triángulo:** área  $base \times altura$   $base \times altura$   $base \times altura$  y perímetro  $3 \times lado$   $3 \times lado$   $3 \times lado$

- **Resultados**

Los resultados se muestran en campos de texto (`JTextField`) específicos para cada figura.

## RUBRICA DE EVALUACIÓN

**Coloque 1 punto en cada criterio si considera que cumple con la descripción.**

Criterio	Descripción	Puntuación
<b>Innovación Tecnológica</b>	Evaluación del grado de originalidad y avance tecnológico del proyecto.	3 puntos
- Creatividad en la solución propuesta	¿El proyecto presenta una solución novedosa y original para un problema específico?	
- Utilización de tecnologías actuales	¿Se emplean tecnologías modernas y relevantes para la resolución del problema?	
- Impacto potencial en la sociedad	¿El proyecto tiene el potencial de generar un impacto positivo en la sociedad o en un sector específico?	
<b>Total obtenido en el criterio</b>		
<b>Interdisciplinariedad</b>	Evaluación de la integración de conocimientos y enfoques de diferentes disciplinas en el proyecto.	3 puntos
- Colaboración entre áreas de conocimiento	¿El proyecto integra conocimientos y enfoques de diferentes disciplinas de manera efectiva?	
- Complementariedad de habilidades	¿Los resultados de aprendizaje de las asignaturas se complementan entre sí para abordar el problema de manera integral?	
- Investigación interdisciplinaria	¿Se considera teorías, conceptos y metodologías de las diferentes disciplinas (asignaturas) para generar nuevas perspectivas y soluciones?	
<b>Total obtenido en el criterio</b>		
<b>Desarrollo de Habilidades Blandas</b>	Evaluación de las habilidades personales y sociales desarrolladas durante la realización del proyecto.	4 puntos
- Trabajo en equipo	¿El equipo demuestra habilidades para colaborar efectivamente en la consecución de objetivos comunes?	

- Comunicación efectiva	¿Los miembros del equipo se comunican de manera clara y efectiva tanto oralmente como por escrito?	
- Resolución de problemas	¿El equipo muestra habilidades para identificar, analizar y resolver problemas de manera eficiente?	
- Adaptabilidad y flexibilidad	¿El equipo demuestra capacidad para adaptarse a cambios y ajustar su enfoque según sea necesario?	
<b>Total obtenido en el criterio</b>		
<b>Calidad del Proyecto</b>	Evaluación del nivel de detalle, precisión y eficacia en la ejecución del proyecto.	4 puntos
- Metodología de trabajo	¿Se sigue una metodología clara y coherente en el desarrollo del proyecto?	
- Documentación y registro	¿Se mantiene una adecuada documentación y registro de los procesos y resultados del proyecto?	
- Calidad técnica del producto/servicio	¿El producto o servicio desarrollado cumple con estándares técnicos y de calidad adecuados?	
- Cumplimiento de objetivos y plazos	¿El equipo logra cumplir con los objetivos establecidos y los plazos previamente definidos en el esquema del proyecto?	
<b>Total obtenido en el criterio</b>		
<b>Presentación y Defensa del Proyecto</b>	Evaluación de la presentación del proyecto y la capacidad para argumentar y defender sus resultados.	3 puntos
- Claridad y estructura de la presentación	¿La presentación del proyecto es clara, bien estructurada y fácil de seguir?	
- Argumentación y justificación de decisiones	¿El equipo es capaz de argumentar y justificar las decisiones tomadas durante el desarrollo del proyecto, que trascienden los límites de las disciplinas individuales, buscando soluciones holísticas (interdisciplinariedad entre las asignaturas del proyecto).?	
- Respuestas a preguntas y críticas	¿El equipo responde de manera fundamentada y con base en la investigación realizada a las preguntas y críticas planteadas por el jurado?	
Documentación del proyecto (Memoria del Proyecto)	La presentación y documentación del proyecto son excepcionales, con claridad, organización, y detalle, incluyendo todos los aspectos relevantes.	3 puntos
Descripciones técnicas	La documentación contiene descripciones técnicas precisas para la solución del problema o desafío. Referencia trabajos previos o fuentes	

	externas que están bien integradas para justificar la solución.	
Estructura	La información presentada contiene secciones bien organizadas, con secuencia lógica y clara. La documentación cumple con la estructura institucional.	
Redacción académica	No existen errores gramaticales y ortográficos que pueden distraer al lector o afectar la claridad de los argumentos, cita y referencia correctamente acorde a las Normas APA.	
<b>Total obtenido en el criterio</b>		
<b>Total obtenido:</b>		
<b>Calificación obtenida sobre 10 Puntos</b>		

#### RUBRICA PARA EVALUACION INDIVIDUAL

Coloque 2.5 punto en cada criterio si considera que cumple con la descripción.

Criterio	Descripción	Puntuación (0-2.5)
Conocimientos Interdisciplinarios para dar solución al problema en el proyecto	Capacidad del estudiante para comprender y relacionar los conceptos de diferentes disciplinas relevantes para el proyecto.	
Colaboración y Trabajo en Equipo	Grado de participación activa del estudiante en el trabajo en equipo, incluyendo la comunicación efectiva, la cooperación, adquisición de recursos y la contribución al logro de objetivos comunes del proyecto.	
Creatividad y Originalidad	Nivel de creatividad y originalidad demostrado por el estudiante en la generación de ideas y enfoques para abordar los desafíos del proyecto, producto de la investigación interdisciplinaria.	
Aplicación Efectiva de Conocimientos Interdisciplinarios	Habilidad del estudiante para aplicar de manera efectiva los conocimientos adquiridos de diferentes disciplinas en la resolución de problemas y justificar de manera argumentada las preguntas del tribunal.	
Total/10 puntos		

ELABORADO	REVISADO	APROBADO
NOMBRE:	NOMBRE:	NOMBRE:
Firma:		Firma:
Fecha:		Fecha:

