

MIXQUIAHUALA DE JUÁREZ HGO.27 DE NOVIEMBRE DEL 2023.

**INSTITUTO TECNOLÓGICO SUPERIOR DEL OCCIDENTE DEL ESTADO DE
HIDALGO.**

TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES.

SOFTWARE DE RESOLUCIÓN DE PROBLEMAS DE INGENIERÍA

DIEGO ANTONIO BADILLO MORALES. 230110025

JOSE MANUEL CORTES CERON. 230110688

SEBASTIÁN HERNÁNDEZ ÁNGELES. 230110313

CARMEN ANAHI CORNEJO LÓPEZ. 230110084

OSWALDO GABRIEL VILLAVERDE MENDOZA. 230110856

ATRIBUTO DE EGRESO.

Identificar y formular problemas de ingeniería aplicando los principios y teorías de las ciencias básicas, así como técnicas, métodos y herramientas de Ingeniería en tecnologías de la información y comunicaciones, conduciendo experimentaciones adecuadas.



CRITERIO DE DESEMPEÑO.

**CD2: Aplicar satisfactoriamente metodologías,
herramientas y estándares en su área de
especialidad y en el desarrollo de proyectos.**



OBJETIVO.

Implementar funciones matemáticas relevantes dentro de la ingeniería en tecnologías de la información y comunicaciones para desarrollar en los estudiantes su capacidad de comprensión, análisis y solución de problemas ingenieriles utilizando metodologías y herramientas de desarrollo de software estandarizadas.

PLANTEAMIENTO DEL PROYECTO.

Los estudiantes deben implementar en programas computacionales un conjunto de funciones matematicas utilizadas en ciencia de datos, electronica, redes, almacenamiento de datos y seguridad informatica. La metodologia que deberan usar sera la metodologia de las 6D. Las herramientas que deberan usar sera LaTeX para la redaccion del informe de la implemetacion de los programas, Git y Github para el versionamiento del codigo base y JAVA y NetBeans para la parte de codificacion.

**Oswaldo Gabriel
Villaverde Mendoza**

**Dados 2 puntos A y B con coordenadas x_1, y_1 y x_2, y_2 respectivamente.
Regresar la ecuación de la recta y el ángulo interno α que se forma
entre el eje horizontal y la recta.**

**Carmen Anahí
Cornejo López**

**Dada una ecuación cuadrática regresar los valores de las raíces en caso de
que estén sobre el conjunto de los números reales, en caso contrario indicar
que la solución esta en el conjunto de los números complejos.**

**Sebastián Hernández
Ángeles**

**Dada una circunferencia con centro en el punto C con coordenadas (x_1, y_1) y radio r , evaluar si un punto I con coordenadas (x_2, y_2) esta dentro
del arco de la circunferencia.**

**Diego Antonio Badillo
Morales**

**Dado un numero decimal entero positivo o negativo regresar
su equivalente en binario.**

**José Manuel
Cortes Cerón**

**Dado un numero binario de n bits regresar su equivalente en
decimal.**

**José Manuel
Cortes Cerón**

**Dada una tabla de verdad de n bits generar la expresión
booleana que genere de manera fidedigna las salidas de esta
tabla.**

- **EJERCICIO 1**

Descripción del problema

El problema consiste en calcular la ecuación de una recta y su ángulo con el eje horizontal, dados dos puntos de la recta.

Definición del problema.

- El deseo del problema es obtener los siguientes resultados:
- La ecuación de la recta, en la forma $y = m \cdot x + b$.
- El ángulo con el eje horizontal, en grados.

Diseño de la solución.



Desarrollo de la solución.

1.

La función main comienza solicitando al usuario las coordenadas de A y B

2.

La ecuación de la recta para (y) se utiliza la formula $(y2 - y1) / (x2 - x1)$ y se almacena en la variable m.

```
] public static void main(String[] args) {  
  
    Scanner hola=new Scanner(System.in);  
  
    double x1, y1,x2,y2;  
  
    System.out.println("Ingrese el valor de x1");  
    x1=hola.nextFloat();  
    System.out.println("Ingrese el valor de x2");  
    x2=hola.nextFloat();  
    System.out.println("Ingrese el valor de y1");  
    y1=hola.nextFloat();  
    System.out.println("Ingrese el valor de y2");  
    y2=hola.nextFloat();  
  
    . . . . .
```

Desarrollo de la solución.

```
double m = (y2 - y1) / (x2 - x1);

double angulo = Math.atan(m);

double anguloGrados = Math.toDegrees(angulo);

System.out.println("Ecuación de la recta: y = " + m + " * x + " + (y1 - m * x1));

System.out.println("Ángulo con el eje horizontal (en grados): " + anguloGrados);

}
```

3.

Para (x) se utiliza la siguiente operación $y1 - m * x1$.

4.

Para sacar el ángulo en grados se utiliza `Math.atan(m)` y para calcular el ángulo en radianes, se usa `Math.toDegrees()` para pasar de radianes a grados.

➤ Prueba básica:

- $(x1, y1) = (0, 0)$
- $(x2, y2) = (1, 1)$

o Ingrese el valor de x1

0

o Ingrese el valor de x2

1

o Ingrese el valor de y1

0

o Ingrese el valor de y2

1

Ecuación de la recta: $y = 1.0 \ x = 0.0$

Angulo con el eje horizontal (en grados): 45.0

Depuración.

- **EJERCICIO 2**

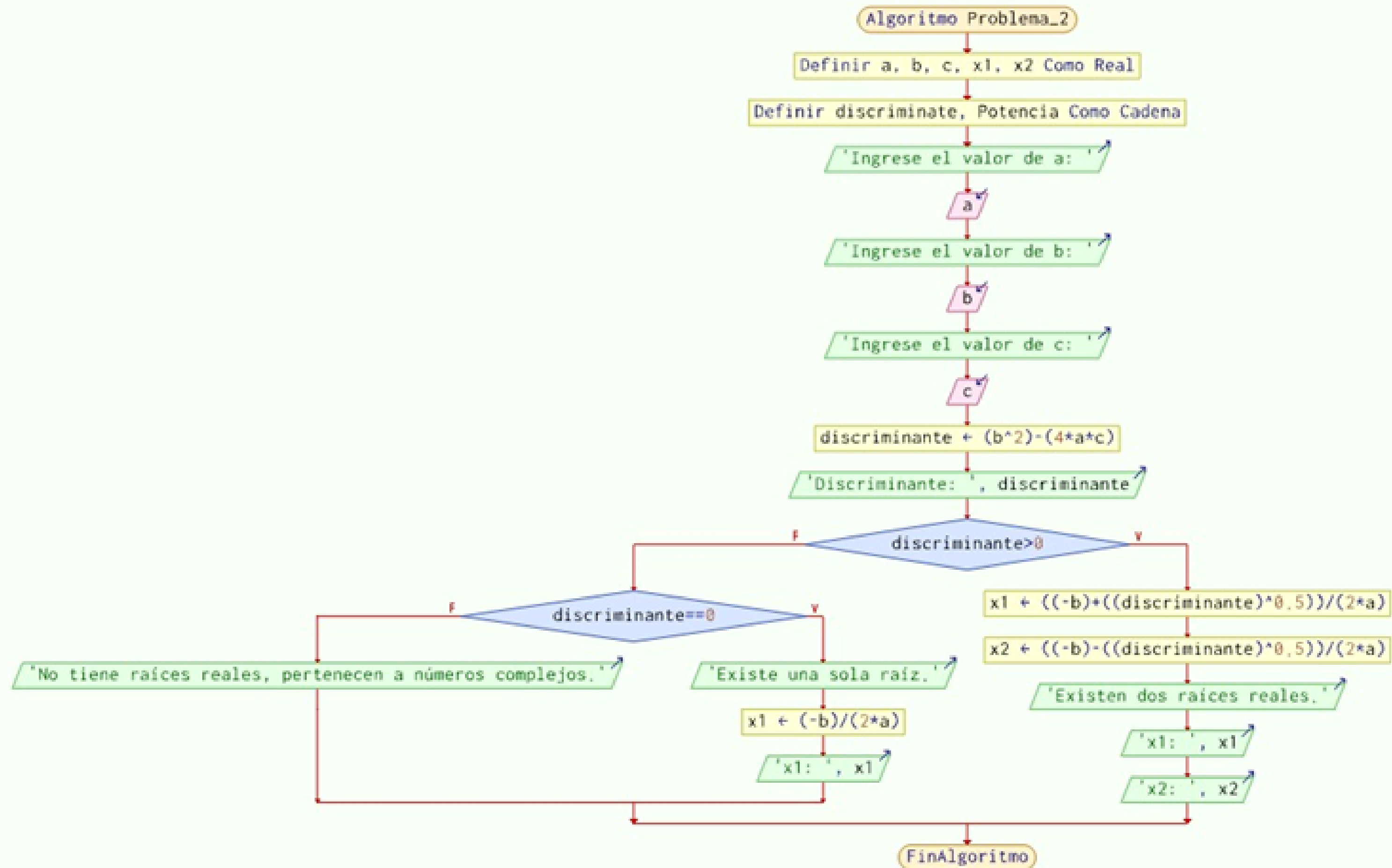
Descripción del problema

Dada la ecuación que ingrese se regresara los valores sobre el conjunto de los números reales, en el caso de que no indicara la solución esta en el conjunto de los números.

Definición del problema.

- **Entrada:** Se solicita al usuario que ingrese a , b y c .
- **Salida:** Te arrojará como resultado: los valores de (x_1, x_2) y escribirá si las raíces son reales, idénticas o si pertenecen al conjunto de números complejos

Diseño de la solución.



Desarrollo de la solución.

1.

- Antes que nada, tenemos que tener claro que la ecuación cuadrática siempre será $ax^2+bx+c=0$

2.

- Definimos a, b, c, x1, x2 como double para que pueda ver resultados decimales.

3.

- Se crea un objeto como Scanner para poder leer a, b, c, x1, x2

```
package com.mycompany.integrador;
import java.util.Scanner;

/**
 *
 * @author Carmen Anahi Cornejo
 */
public class Integrador {
    public static void main(String[] args) {
        double a, b, c, x1, x2;
        Scanner obj1 = new Scanner(System.in);
        System.out.println("teclea el valor de a: ");
        a = obj1.nextFloat();
        System.out.println("teclea el valor de b: ");
        b = obj1.nextFloat();
        System.out.println("teclea el valor de c: ");
        c = obj1.nextFloat();
    }
}
```



```

double discriminante = (Math.pow(a: b, b: 2)) - (4 * a * c);
System.out.println("Discriminante: " + discriminante);
if (discriminante > 0) {
    x1 = ((-b) + (Math.sqrt(a: discriminante))) / (2 * a);
    x2 = ((-b) - (Math.sqrt(a: discriminante))) / (2 * a);
    System.out.println(x: "Existen dos raíces reales. ");
    System.out.println("x1: " + x1);
    System.out.println("x2: " + x2);
} else if (discriminante == 0) {
    System.out.println(x: "Existe una sola raíz. ");
    x1 = (-b) / (2 * a);
    System.out.println("x1: " + x1);
} else {
    System.out.println(x: "No tiene raíces reales, pertenecen a números complejos.");
}
}

```

- Si el discriminante > cero, hay dos raíces reales distintas.
- Si el discriminante = cero, existe una sola raíz real idéntica.
- Si el discriminante < cero, no tiene raíces reales, pertenecen a números complejos.

```
tecla el valor de a:
4
tecla el valor de b:
8
tecla el valor de c:
-32
Discriminante: 576.0
Existen dos raíces reales.
x1: 2.0
x2: -4.0
```

```
tecla el valor de a:
3
tecla el valor de b:
-9
tecla el valor de c:
10
Discriminante: -39.0
No tiene raíces reales, pertenecen a números complejos.
```

```
teclea el valor de a:
0
teclea el valor de b:
0
teclea el valor de c:
0
Discriminante: 0.0
No tiene raíces reales, pertenecen a números complejos.
```

Depuración.

- EJERCICIO 3

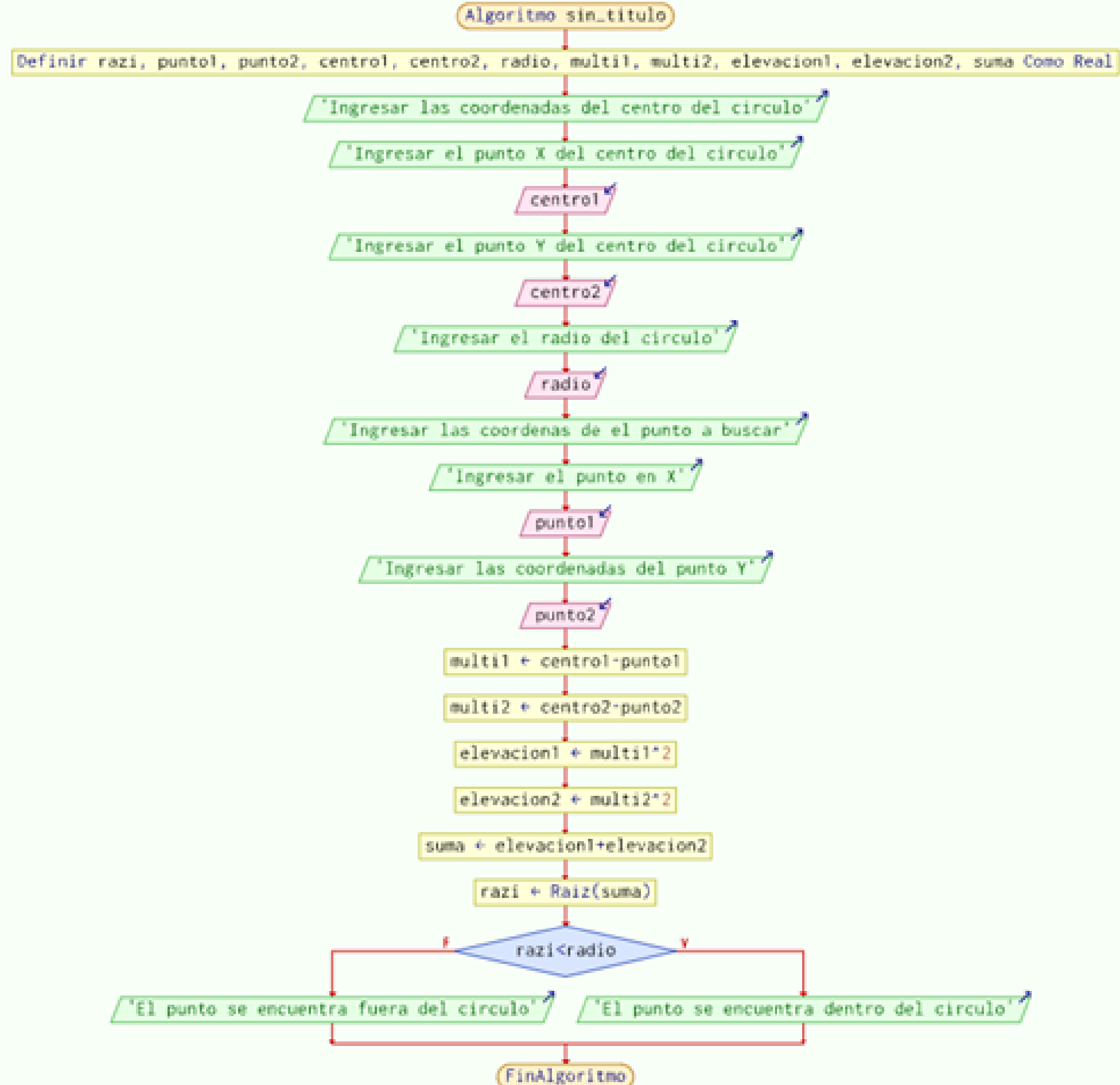
Descripción del problema

• **Problema:** Dada una circunferencia con centro en el punto C con coordenadas $(X1, X2)$ y el radio R, evaluar si un punto T con coordenadas $(X2, Y2)$ está dentro del área de la circunferencia.

Definición del problema.

- Problema: Saber si un punto se encuentra dentro o fuera de un círculo.
- Entradas: El punto que se desea encontrar.
- Salidas: Saber si el punto se encuentra dentro o fuera de círculo.

Diseño de la solución.



Desarrollo de la solución.

1.

- main comienza solicitando al usuario

2.

- Se crean dos objetos Scanner (x1, x2) para leer la entrada del usuario.

3.

- Se solicita que ingrese el radio del círculo.

4.

- Se crea un objeto Scanner (r) para leer la entrada del usuario.

```
public static void main(String[] args) {  
    Scanner obj = new Scanner(System.in);  
  
    float puntol, punto2, centrol, centro2, radio;  
    float multil, multi2, elevacion1, elevacion2, suma;  
  
    System.out.println(x: "Ingresa las cordenadas del centro del circulo");  
    System.out.println(x: "Ingresa el punto en X del centro del circulo");  
    centrol=obj.nextFloat();  
    System.out.println(x: "Ingresa el punto Y del centro del circulo");  
    centro2=obj.nextFloat();  
    System.out.println(x: "Ingresa el radio del circulo");  
    radio=obj.nextFloat();  
    System.out.println(x: "Ingresa las cordenadas de el punto a buscar");  
    System.out.println(x: "Ingresa el punto en X");  
    puntol=obj.nextFloat();  
    System.out.println(x: "Ingresa el punto en Y");  
    punto2=obj.nextFloat();  
}
```

1.

- Se solicita que ingrese las coordenadas del punto a buscar.

2.

- Se crean dos objetos Scanner (x1, x2) para leer la entrada del usuario.

3.

- Se crea dos objetos Scanner (x2, y2) para leer la entrada del usuario.

4.

- Se realiza la operación matemática en donde se restan las coordenadas y se elevan al cuadrado, después se suman los resultados y saca la raíz cuadrado.

```
multi1 = centro1 - punto1;  
multi2 = centro2 - punto2;
```

```
elevacion1 = (float) Math.pow(a: multi1, b: 2);  
elevacion2 = (float) Math.pow(a: multi2, b: 2);
```

```
suma = elevacion1 + elevacion2;
```

```
double rais = (double) Math.sqrt(a: suma);
```

```
if (rais < radio) {  
    System.out.println(x: "El punto se encuentra dentro del circulo");  
}else System.out.println(x: "El punto se encuentra fuera del circulo");  
}
```

```
1  ESCRIBIR EL PROGRAMA QUE EJECUTE
```

```
2  Ingresar las coordenadas del centro del círculo
```

```
3  Ingresar el punto en X del centro del círculo
```

```
4
```

```
5  Ingresar el punto Y del centro del círculo
```

```
6
```

```
7  Ingresar el radio del círculo
```

```
8
```

```
9  Ingresar las coordenadas de el punto a buscar
```

```
10 Ingresar el punto en X
```

```
11
```

```
12 Ingresar el punto en Y
```

```
13
```

```
14 El punto se encuentra dentro del círculo
```

Depuración.

- **EJERCICIO 4**

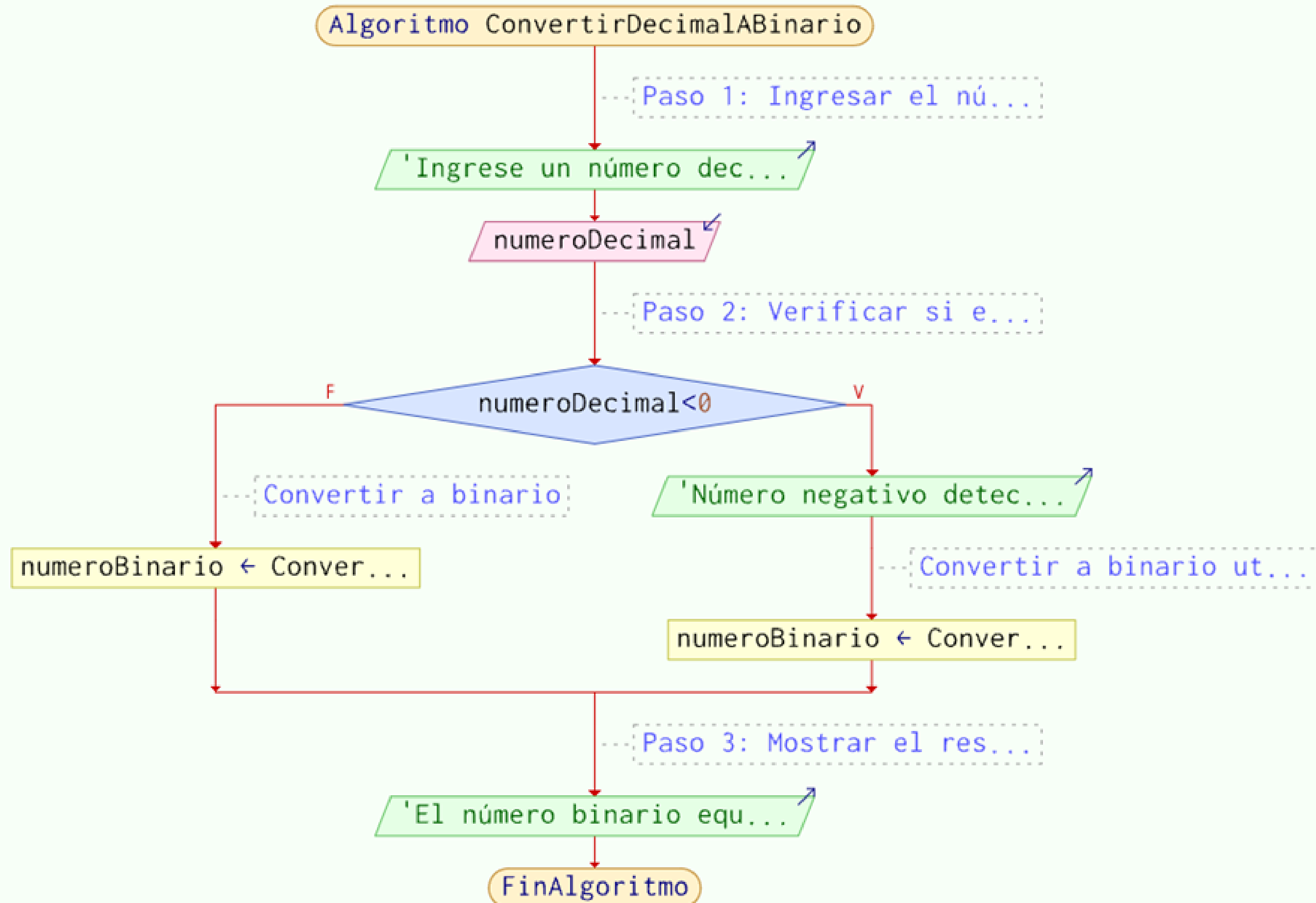
Descripción del problema

El problema resuelto con este código es la conversión de un número decimal entero a binario. El número binario se representa como una cadena de bits, donde cada bit tiene un valor de 0 o 1.

Definición del problema.

- **El tipo de datos de número decimal:** En este caso, el número decimal es un número entero.
- **Rango de valores decimal:** desde -2^{32} hasta $2^{32} - 1$.
- **El número binario se representa como una cadena de bits y cada bit tiene un valor de 0 o 1.**

Diseño de la solución.



Desarrollo de la solución.

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print(s: "Ingrese un número decimal entero: ");
    int numero = sc.nextInt();
    // Convertimos el número decimal a binario
    String binario = Integer.toBinaryString(numero);
    // Si el número es negativo, aplicamos el método de complemento a2
    if (numero <= 0) {
        // Añadir un 1 al principio
        binario = "1" + binario;
        // Completamos el binario con ceros hasta que tenga la misma longitud que el complemento a2
        for (int i = binario.length(); i < 32; i++) {
            binario = "0" + binario;
        }
    }
    System.out.println("El número binario es: " + binario);
}
```

- Comprobar si el número es positivo o negativo.
- Convertir el número decimal a binario, si es positivo o negativo.
- Si el número es negativo, se aplica el método de complemento a2.
- El método de complemento a2 consiste en: Invertir el orden de bits de binario del número negativo y por ultimo añadir el 1.

Pruebas de escritorio

Introduzca un número decimal: 10

El número binario es: 1010

Ingrese un número decimal entero: -10

El número binario es: 1111111111111111111111111111111110110

Introduzca un número decimal: 0

El número binario es: 0

Depuración.

- **EJERCICIO 5**

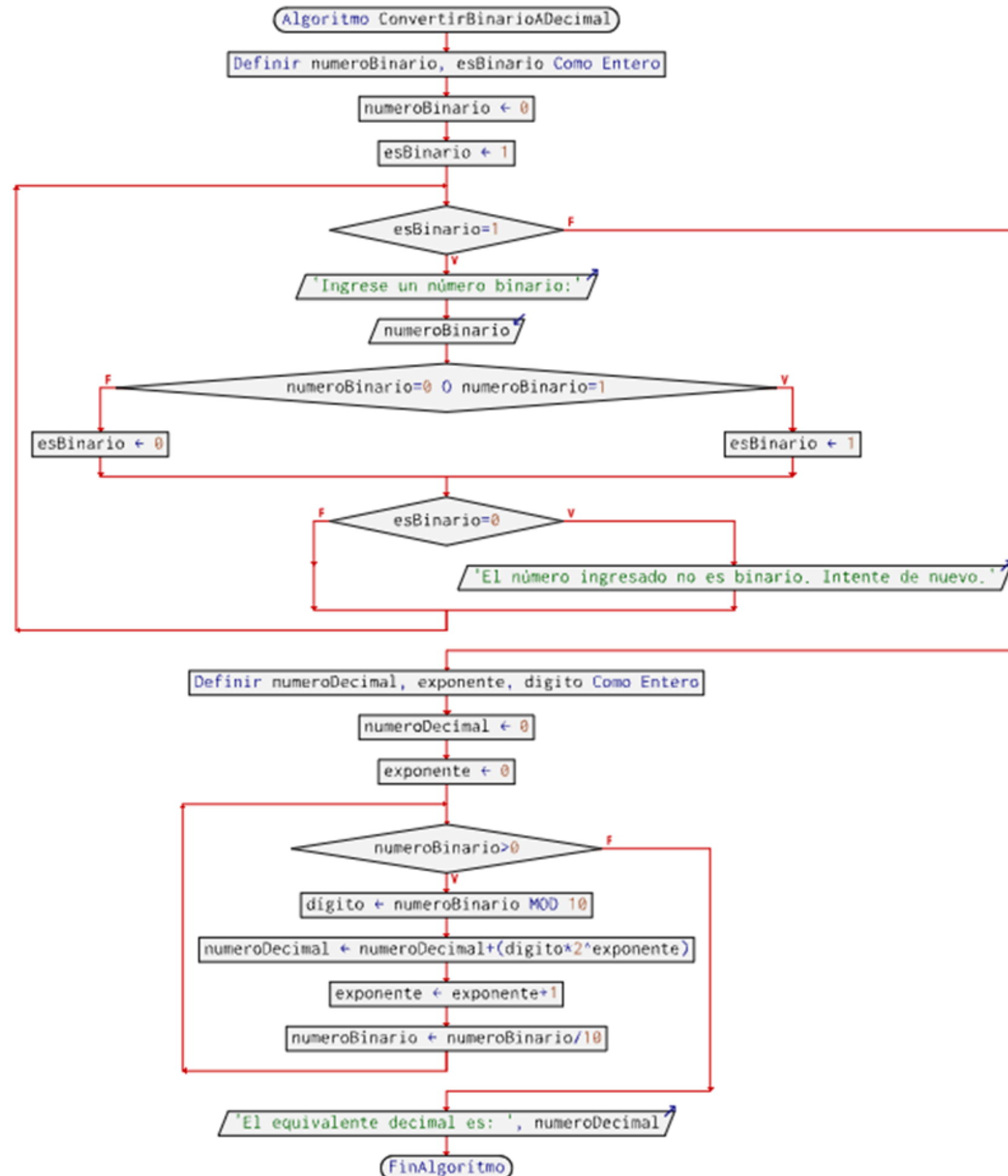
Descripción del problema

Dado un número binario de n bits regresar su equivalente en decimal.

Definición del problema.

- **Problema:** Convertir un número binario ingresado por el usuario a su equivalente decimal.
- **Entradas:** El número binario ingresado por el usuario.
- **Salidas:** El equivalente decimal del número binario.

Diseño de la solución.



Desarrollo de la solución.

- main declara una variable n para almacenar el número binario.
- Scanner (binario)
- Se usa un bucle do-while .
- Se ingrese un número binario válido (solo 0 y 1).

```
public static void main(String[] args) {  
    String numeroBin;  
  
    Scanner binario = new Scanner(System.in);  
  
    do {  
        System.out.print("Ingresa un número binario: ");  
        numeroBin = binario.nextLine();  
  
        if (numeroBin.isEmpty()) {  
            System.out.println("Por favor ingresa un numero binario");  
        } else if (!esBinario(numeroBin)) {  
            System.out.println("El número ingresado no es binario (solo 0 y 1).");  
        }  
  
    } while (numeroBin.isEmpty() || !esBinario(numeroBin));  
  
    int resultadoDecimal = binarioADecimal(numeroBin);  
    System.out.println("El número binario " + numeroBin + " en decimal es: " + resultadoDecimal);  
}
```

- Para obtener un número llama a la función binarioADecimal para convertirlo a su equivalente decimal .

Desarrollo de la solución.

```
public static boolean esBinario(String numero) {  
    // Verifica si el número ingresado es binario (contiene solo 0 y 1)  
    return numero.matches(regex: "[01]+");  
}
```

- Esta función toma un String como argumento y verifica si el número ingresado consiste solo en caracteres '0' y '1'. Retorna true si es un número binario válido; de lo contrario, retorna false.

Desarrollo de la solución.

Función binarioADecimal:

```
public static int binarioADecimal(String binario) {  
    // Convierte un número binario a decimal  
    int decimal = 0;  
    int longitud = binario.length();  
  
    for (int i = 0; i < longitud; i++) {  
        int digito = Character.getNumericValue(binario.charAt(longitud - i - 1));  
        decimal += digito * Math.pow(2, i);  
    }  
  
    return decimal;  
}
```

- Esta función convierte un número binario dado en un número decimal.
- Itera sobre cada dígito del número binario.
- convirtiendo cada dígito a su equivalente decimal y sumándolo al resultado final.
- Usa la fórmula de conversión de binario a decimal: suma de (dígito * $2^{\text{posición}}$).

```
run:  
Ingresa un número binario: 101001010101101001  
El número binario 101001010101101001 en decimal es: 169321  
¿Deseas convertir otro? 1=SI 0=NO
```

Depuración.

- **EJERCICIO 6**

Descripción del problema

Se busca obtener una expresión booleana que represente de manera fidedigna las salidas de una tabla de verdad de n bits.

Definición del problema.

- **Entrada de datos:** Se requiere la tabla de verdad de n bits que contiene las combinaciones de entrada y sus salidas correspondientes.
- **Análisis de datos:** Identifique las combinaciones de entrada que producen una salida de 1 y se utiliza esta información para construir la expresión booleana.

Diseño de la solución.

Algoritmo:

1. Inicio
2. Ingresar la cantidad de bits (numBits):
3. Inicializar el arreglo de letras (letras):
4. Calcular el total de combinaciones posibles (totalCombinaciones):.
5. Mostrar la tabla de verdad:
6. Generar expresiones booleanas para las filas con valor de salida 1
7. Fin

Desarrollo de la solución.

1.

Importa las clases necesarias (Scanner, ArrayList, List).

```
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;
```

2.

Define un conjunto de letras para representar los bits.

```
/**
 *
 * @author Josee Dada una tabla de verdad de hasta 4 bits generar la expresion
 * booleana que genere de manera fidedigna la salidas de esa tabla
 */
public class Problema_6 {
```

3.

Crea un objeto Scanner para la entrada de datos.

```
/**
 * @param args the command line arguments
 */
public static void main(String[] args) {
    // Crear un objeto Scanner para la entrada de datos desde la consola
    Scanner dato = new Scanner(System.in);
```

4.

Pide al usuario que ingrese la cantidad de bits.

```
    // Definir un conjunto de letras para representar los bits
    char[] letras = {'A', 'B', 'C', 'D'};

    // Solicitar al usuario que ingrese la cantidad de bits
    System.out.println("Ingrese la cantidad de bits:");
    int numBits = dato.nextInt();
```


Generación de la tabla de verdad y solicitud de valores de salida:

1.

Calcula el número total de combinaciones posibles de bits.

2.

Muestra las letras que representan los bits y una línea separadora.

3.

Muestra todas las combinaciones posibles de bits.

4.

Solicita valores de salida para cada combinación de bits.

```
// Calcular el número total de combinaciones posibles de bits
int totalCombinations = (int) Math.pow(2, numBits);
```

```
// Mostrar las letras que representan los bits
for (int i = 0; i < numBits; i++) {
    System.out.print(letras[i] + "\t");
}
System.out.println("x");
```

```
// Mostrar una línea separadora para los encabezados
for (int i = 0; i < numBits + 1; i++) {
    System.out.print("-----");
}
System.out.println();
```

```
// Mostrar todas las combinaciones posibles de bits y solicitar valores de salida para cada combinación
for (int i = 0; i < totalCombinations; i++) {
    for (int j = 0; j < numBits; j++) {
        int bit = (i >> j) & 1;
        System.out.print(bit + "\t");
    }
    System.out.println();
}
```

```
// Solicitar valores de salida para cada combinación de bits
int cantSalidas = (int) Math.pow(2, numBits);
int[] salidas = new int[cantSalidas];
```

Procesamiento de valores de salida y generación de expresiones booleanas:

1.

Solicita valores de salida para cada combinación de bits

2.

Almacena en una lista las filas que tienen un valor de salida igual a 1.

```
int cantSalidas = (int) Math.pow(a: 2, b: numBits);
int[] salidas = new int[cantSalidas];

for (int i = 0; i < cantSalidas; i++) {
    System.out.println("Teclea el valor de la salida " + (i + 1) + " : ");
    salidas[i] = dato.nextInt();
}

// Almacenar las filas que tienen el valor de salida como 1
List<Integer> filasConUno = new ArrayList<>();
for (int i = 0; i < cantSalidas; i++) {
    if (salidas[i] == 1) {
        filasConUno.add(i);
    }
}

// Mostrar las expresiones booleanas correspondientes a las filas con valor de salida 1
if (!filasConUno.isEmpty()) {
    // Mostrar encabezados
    System.out.println();
    for (int i = 0; i < numBits; i++) { ...3 lines }
    System.out.println(x: "x");

    for (int i = 0; i < numBits + 1; i++) { ...3 lines }
    System.out.println();
}
```

3.

Si hay filas con valor de salida 1, se procede a mostrar las expresiones booleanas correspondientes a estas filas. En caso contrario, muestra un mensaje.

Procesamiento de valores de salida y generación de expresiones booleanas:

1.

- Si hay filas con valor de salida igual a 1, muestra los encabezados y las combinaciones.

2.

- Genera las expresiones booleanas correspondientes.

3.

- Las expresiones se construyen usando las letras que representan los bits.

```
// Mostrar las combinaciones de bits actualizadas con los valores de salida
for (int i = 0; i < totalCombinations; i++) {
    for (int j = 0; j < numBits; j++) {
        int bit = (i >> j) & 1;
        System.out.print(bit + "\t");
    }
    System.out.println(salidas[i]); // Mostrar el valor de salida actualizado
}

System.out.println("\nLas expresiones booleanas de las salidas 'x' en 1 son:");

// Generar las expresiones booleanas para cada fila que tiene un "1" como valor de salida
for (Integer fila : filasConUno) {
    for (int j = 0; j < numBits; j++) {
        int bit = (fila >> j) & 1;
        if (bit == 0) {
            System.out.print("~" + letras[j] + " ");
        } else {
            System.out.print(letras[j] + " ");
        }
    }
    System.out.println();
} else {
    System.out.println("\nNo hay filas con valor 1 para generar expresiones booleanas.");
}
```

- **Verificación de la lógica:** Se revisa que la expresión generada por la tabla de verdad produzca los resultados esperados.

Depuración.

Conclusión



GRACIAS