

# Software de resolución de problemas de Ingeniería

Martinez Martinez Freyra Wendy, *Instituto Tecnológico Superior del Occidente del Estado de Hidalgo, Mixquiahuala, Hgo., 42700, Mexico*

Mera Ibarra Carol, *Instituto Tecnológico Superior del Occidente del Estado de Hidalgo, Mixquiahuala, Hgo., 42700, Mexico*

Gonzalez Yañez Dulce Karen, *Instituto Tecnológico Superior del Occidente del Estado de Hidalgo, Mixquiahuala, Hgo., 42700, Mexico*

Mendoza Calva Marco Antonio, *Instituto Tecnológico Superior del Occidente del Estado de Hidalgo, Mixquiahuala, Hgo., 42700, Mexico*

Gomez Mejia Eduardo, *Instituto Tecnológico Superior del Occidente del Estado de Hidalgo, Mixquiahuala, Hgo., 42700, Mexico*

*Abstract—En este informe, se presenta una metodología efectiva para resolver problemas de ingeniería. Se abordan ejercicios como calcular el ángulo entre una recta y el eje horizontal, encontrar la ecuación de una recta que pasa por dos puntos, convertir números decimales enteros a su equivalente binario y convertir números binarios a decimales. Se propone un enfoque estructurado y paso a paso para abordar estos problemas, brindando claridad y precisión en los resultados.*

**D**urante este proyecto de software de resolución de problemas de ingeniería se colaboró en equipo para llegar a soluciones que ponen a prueba los conocimientos adquiridos a lo largo del primer semestre como estudiantes de Ingeniería en Tecnologías de la Información y Comunicaciones. Durante el desarrollo se va a utilizar la metodología de las 6 D's para resolver de manera efectiva los ejercicios propuestos relacionados a la geometría analítica, álgebra y matemáticas discretas mediante el lenguaje de programación de Java. Considerando los datos obtenidos, se ha obtenido experiencia en ámbitos como la programación, en matemáticas o incluso en el desarrollo de soluciones mediante ejercicios como encontrar la ecuación de una recta que pasa por dos puntos y calcular el ángulo entre esta recta y el eje horizontal, resolver una ecuación cuadrática con la fórmula general, la conversión de números decimales enteros a su equivalente binario y la conversión de números binarios a decimales.

### Resolución Problema 1

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 ( $\alpha$ ) que se forma entre el eje horizontal y la recta.

#### Descripción del problema:

En el contexto de geometría analítica, la tarea es obtener la ecuación de la recta que pasa por los puntos A y B, así como determinar el ángulo interno  $\alpha$  que se forma entre la recta y el eje horizontal. Por lo tanto la solución implica el uso de conceptos trigonométricos y algebraicos para abordar este problema geométrico de manera eficiente.

#### Definición de solución:

La solución implica calcular la pendiente de la recta que pasa por los puntos A y B y utilizarla para obtener la ecuación de la recta en la forma  $y = mx + b$ , donde  $m$  es la pendiente y  $b$  es la intersección en el eje  $y$ . Una vez obtenida la ecuación de la recta, se puede calcular el ángulo interno utilizando la función trigonométrica arcotangente.

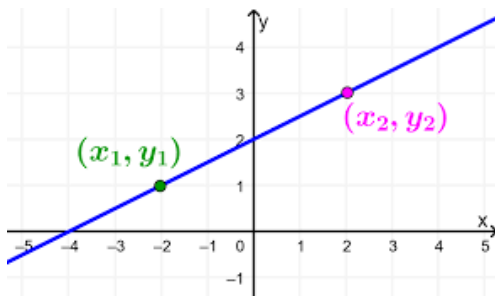


Figura 1. Gráfica de dos puntos en la recta.

#### Diseño de la solución:

- Calcular la pendiente ( $m$ ) de la recta utilizando la fórmula:

$$m = (y_2 - y_1) / (x_2 - x_1) \quad (1)$$

- Calcular el término independiente ( $b$ ) utilizando la fórmula

$$b = y_1 - mx_1. \quad (2)$$

- Obtener la ecuación de la recta

$$y = mx + b. \quad (3)$$

- Calcular el ángulo interno ( $\alpha$ ) utilizando la fórmula

$$\alpha = \tan^{-1}(m) \quad (4)$$

Utilizando este método, puedes encontrar la ecuación de la recta a partir de dos puntos. Recuerda que si los dos puntos son idénticos la recta será una línea vertical [?]

#### Diagrama de flujo

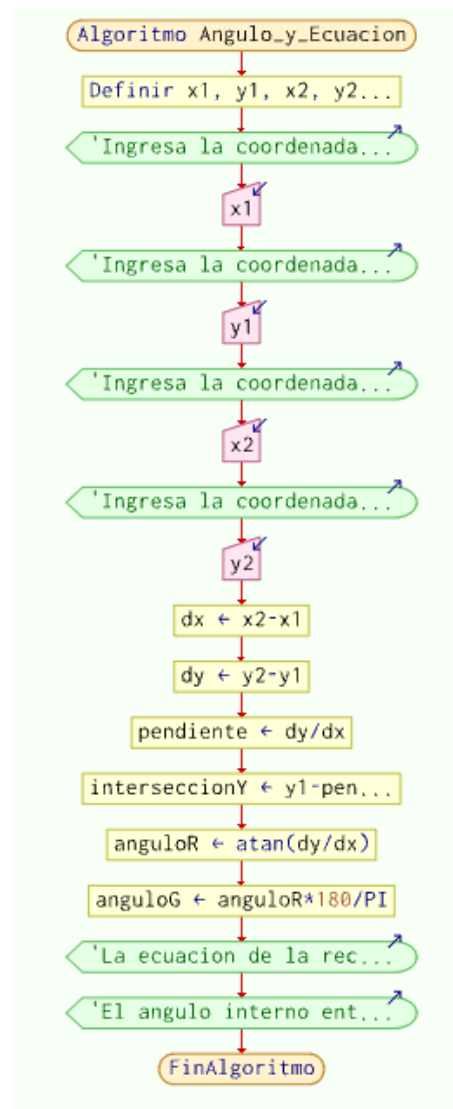


Figura 2. Diagrama de flujo usado para la solución del problema

## Desarrollo de la solución:

// Comienza solicitando al usuario dos puntos  $A(x_1, y_1)$  y  $B(x_2, y_2)$ .

```
Scanner sc = new Scanner(System.in);
// Solicita datos del primer punto
System.out.println("Ingresa la coordenada x1:");
double x1 = sc.nextDouble();
System.out.println("Ingresa la coordenada y1:");
double y1 = sc.nextDouble();

// Solicita datos del segundo punto
System.out.println("Ingresa la coordenada x2:");
double x2 = sc.nextDouble();
System.out.println("Ingresa la coordenada y2:");
double y2 = sc.nextDouble();
```

//Se calcula la diferencia entre coordenadas, con el fin de determinar la distancia o el desplazamiento entre dos puntos en un sistema de coordenadas.

```
// Calcula la diferencia entre coordenadas
double dx = x2 - x1;
double dy = y2 - y1;
```

//Calculamos la pendiente usando la formula  $m = (y_2 - y_1)/(x_2 - x_1)$  Una vez obtenida la pendiente, obtenemos la intersección con el eje "Y"

```
// Calcula la pendiente y la intersección con el eje Y
double pendiente = dy / dx;
double interseccionY = y1 - pendiente * x1;
```

// Calcula el ángulo en radianes

```
double anguloR = Math.atan2(dy, dx);
```

// Convierte el ánguloR a grados

```
double anguloG = Math.toDegrees(anguloR);
```

//Finalmente obtenemos como resultado la ecuación de la recta que pasa a través de dos puntos dados, A y B. Además del el ángulo interno formado entre el eje horizontal y la recta.

```
// Imprime la ecuación y el ángulo interno
System.out.println("La ecuación de la recta es: y = " + pendiente + "x + " + interseccionY);
System.out.println("El ángulo interno entre la recta y el eje horizontal es: " + anguloG + " grados");
```

## Código general

```
Scanner sc = new Scanner(System.in);
// Solicita datos del primer punto
System.out.println("Ingresa la coordenada x1:");
```

```
double x1 = sc.nextDouble();
System.out.println("Ingresa la coordenada y1:");
double y1 = sc.nextDouble();

// Solicita datos del segundo punto
System.out.println("Ingresa la coordenada x2:");
double x2 = sc.nextDouble();
System.out.println("Ingresa la coordenada y2:");
double y2 = sc.nextDouble();

// Calcula la diferencia entre coordenadas
double dx = x2 - x1;
double dy = y2 - y1;

// Calcula la pendiente y la intersección con el eje y
double pendiente = dy / dx;
double interseccionY = y1 - pendiente * x1;

// Calcula el ángulo en radianes
double anguloR = Math.atan2(dy, dx);

// Convierte el ánguloR a grados
double anguloG = Math.toDegrees(anguloR);

// Imprime la ecuación y el ángulo interno
System.out.println("La ecuación de la recta es: y = " + pendiente + "x + " + interseccionY);
System.out.println("El ángulo interno entre la recta y el eje horizontal es: " + anguloG + " grados");
```

## Depuración y pruebas:

```
Ingresa la coordenada x1:
3
Ingresa la coordenada y1:
2
Ingresa la coordenada x2:
5
Ingresa la coordenada y2:
1
La ecuación de la recta es: y = -0.5x + 3.5
El ángulo interno entre la recta y el eje horizontal es: -26.56505117707799 grados
```

Figura 3. Prueba de escritorio.

Número de pruebas	x1	y1	x2	y2	Resultado Ecuación / Ángulo
1	3	2	5	1	Ecuación: -0.5x + 3.5 Ángulo: -26.56505117707799
2	-4	8	12	17	Ecuación: 0.5625x + 10.25 Ángulo: 29.357753542791276
3	56	-78	-112	156	Ecuación: -1.3928571428571428x + 0.0 Ángulo: 125.676408221862
4	588	900	1230	789	Ecuación: -0.17289719626168223x + 1001.6635514018692 Ángulo: -9.809302247879007

Figura 4. Tabla de corridas.

## Resolución Problema 2

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

### Descripción del problema:

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 está en el conjunto de los números complejos. El principal objetivo será encontrar el valor de “x” para lograr que la ecuación sea verdadera.

### Definición de solución:

La forma general de la ecuación de segundo grado es la siguiente:

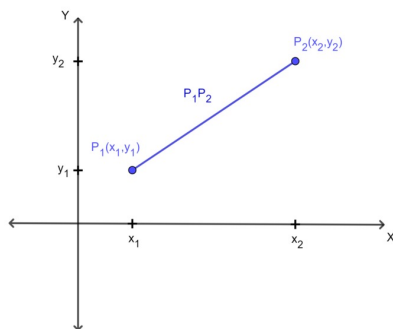
$$ax^2 + bx + c = 0 \quad (5)$$

A continuación se muestra la fórmula general, con la cual se realiza la sustitución de los valores correspondientes a “a”, “b” y “c” de acuerdo a la ecuación de segundo grado (1) que se desee resolver y por ende, conocer su resultado.

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (6)$$

Es importante saber que el resultado puede variar en ninguna, una o dos soluciones reales.

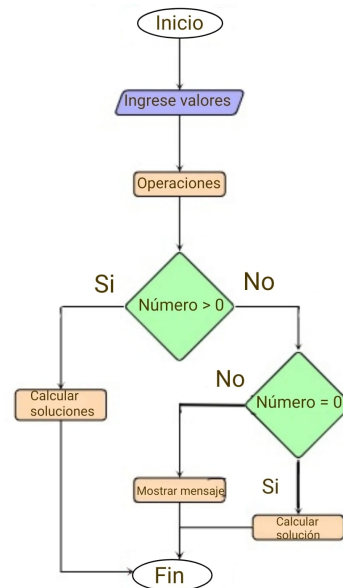
La representación gráfica es la siguiente:



**Figura 5.** Gráfica de una ecuación cuadrática con dos soluciones.

## Diseño de la solución:

Para llevar a cabo la elaboración del código de java, se requiere diseñar el proceso para asegurar que el progreso sea correcto.



**Figura 6.** Diagrama de flujo que muestra el proceso para resolver la fórmula general.

Prosiguiendo con el diseño, ahora se va a mostrar el código como resultado de la elaboración del diagrama donde:

- El usuario debe ingresar tres valores, es decir, los coeficientes pertenecientes a la forma estándar de la ecuación cuadrática.
- El programa realiza las operaciones correspondientes a la fórmula general.
- Al final el o los resultado serán mostrados.

## Desarrollo de la solución:

### Programa en Java

En seguida, se muestra el código del programa. Se utiliza la clase Scanner y la clase Random que se encuentran en el paquete de java.util, y además se agrega el tipo de dato.

```

public static void main(String[] args) {
    Scanner constante = new Scanner(System.in);

    double a, b, c, numero, x1, x2, x;
  
```

\*\* El usuario ingresa los valores solicitados, siendo las variables “a”, “b” y “c”. Posteriormente se cierra el escaneo.

```
//Solicita el valor de los coeficientes para
//realizar la ecuaci n
System.out.print("Ingrese el valor de a:
");
a = constante.nextDouble();

System.out.print("Ingrese el valor de b:
");
b = constante.nextDouble();

System.out.print("Ingrese el valor de c:
");
c = constante.nextDouble();
//Cierra el escaneo
constante.close();
```

Continúa cerrando y a realizar las primeras operaciones de la fórmula general, que serán determinantes para la siguiente condición de la estructura de control if.

```
//Cierra el escaneo
constante.close();

numero = b * b - 4 * a * c;
```

Prosiguiendo con la operación de la fórmula general, ahora se debe determinar cuál será la respuesta dependiendo del resultado de la operación anterior, entonces la estructura de control if va a permitir tomar decisiones en función de la condición y así llegar a un resultado.

```
//Se va a determinar cu l de las tres
//posibles soluciones es
//la indicada dependiendo de los valores
//ingresados
if (numero > 0) {
    x1 = (-b + Math.sqrt(numero)) / (2 *
a);
    x2 = (-b - Math.sqrt(numero)) / (2 *
a);
    System.out.println("Tiene dos
soluciones, las cuales son x1 = "
+ x1 + " y x2 = " + x2);
} else if (numero == 0) {
    x = -b / (2 * a);
    System.out.println("Solo tiene una
soluci n, la cual es x = " + x);
} else {
    System.out.println("La ecuaci n no
tiene soluciones reales.");
}
```

Después de que la condición evaluada sea verdadera, procede a imprimir en la consola el resultado de las operaciones de la fórmula general realizadas junto con el mensaje correspondiente.

### Código completo

```
import java.util.Scanner;

public class ecuacionCuadratica {

    public static void main(String[] args) {
        Scanner constante = new Scanner
(System.in);

        double a, b, c, numero, x1, x2, x;
```

```
//Solicita el valor de los coeficientes
//para realizar la ecuaci n
System.out.print("Ingrese el valor de a:
");
a = constante.nextDouble();

System.out.print("Ingrese el valor de b:
");
b = constante.nextDouble();

System.out.print("Ingrese el valor de c:
");
c = constante.nextDouble();
//Cierra el escaneo
constante.close();

numero = b * b - 4 * a * c;
//Se va a determinar cu l de las tres
//posibles soluciones es la indicada
//dependiendo de los valores ingresados
if (numero > 0){
    x1 = (-b + Math.sqrt(numero)) / (2* a);
    x2 = (-b - Math.sqrt(numero)) / (2* a);
    System.out.println("Tiene dos
soluciones, las cuales son x1 = "
+ x1 + " y x2 = " + x2);
} else if (numero == 0){
    x = -b / (2*a);
    System.out.println("Solo tiene una
soluci n, la cual es es x = " +
x);
} else {
    System.out.println("La ecuaci n no
tiene soluciones reales.");
}
}
```

### Código general

```
package com.mycompany.programacorrecto;

import java.util.Scanner;

public class Programacorrecto {

    public static void main(String[] args) {
        Scanner h=new Scanner(System.in);
        System.out.println("Teclea el valor en
decimal : ");
        int decimal=h.nextInt();
        int bits = (int) (Math.log(Math.abs(decimal))
/ Math.log(2)) + 1;//calcular los bits
del numero entero.
        int valorinicial=decimal;

        // Verificar si el n mero es negativo (-)
        if (decimal < 0) {
            // Obtener la representacin en
            //complemento a 2
            decimal = (int) (Math.pow(2, bits) +
decimal);
        }

        // Convertir Numero decimal a binario
        StringBuilder binary = new StringBuilder();
        do {
            // Obtener el bit menos significativo
            int bit = decimal % 2;
            // Agregar el bit al inicio de la
            //cadena
            binary.insert(0, bit);
            // Dividir el n mero por 2
            decimal /= 2;
```

```

    } while (decimal > 0);

    // Ajustar la longitud al n mero de bits
    especificado
    while (binary.length() < bits) {
        binary.insert(0, '0');
    }

    String binario =binary.toString();
    //converitr a cadena
    System.out. println("      RESULTADOS
    ");
    System.out.println("El N mero decimal es
    : \t" + valorinicial);
    System.out. println("La cantidad de bits
    son: \t " + bits);
    System.out.println("El numero binario es
    :\t " + binario);
}
}

```

### Depuración y pruebas:

Comprobación del funcionamiento a través de pruebas, ingresando los valores solicitados al usuario y ejecutando en Java, un total de cinco veces.

No. De corridas	Valor a	Valor b	Valor c	Resultado
1	1	-5	6	X1 =2, x2=3
2	1	4	4	X=-2
3	1	2	5	Sin solución
4	4	-12	9	X=1.5
5	2	5	-3	X1=0.5, x2=-3

**Figura 7.** Prueba de escritorio.

### Resolución Problema 3

Dada una circunferencia con centro en el punto  $C$  con coordenadas  $(x_1, y_1)$  y radio  $r$ , evaluar si un punto  $T$  con coordenadas  $(x_2, y_2)$  está dentro del área de la circunferencia.

#### Descripción del problema:

1. El problema tiene como objetivo determinar si un punto  $T$  con coordenadas  $(x_2, y_2)$  se encuentra dentro del área de una circunferencia. El usuario proporcionará las coordenadas  $(x_1, y_1)$  del centro de la circunferencia, las coordenadas  $(x_2, y_2)$  del punto  $T$ , y el radio  $r$ . El proceso implica utilizar la ecuación de distancia entre dos puntos y presentar la respuesta resultante.

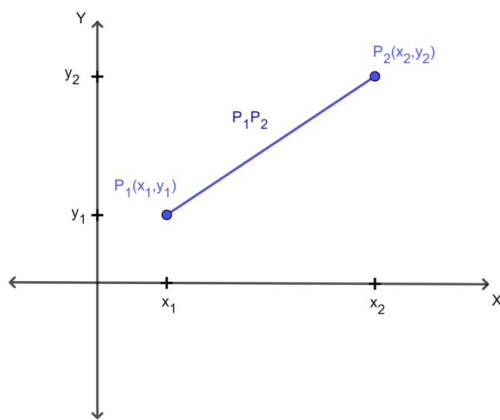


Figura 8. La distancia entre dos puntos.

#### Definición de solución:

La solución consiste en evaluar si un punto  $T$  con coordenadas  $(x_2, y_2)$  está dentro del área de una circunferencia con centro en  $(x_1, y_1)$  y radio  $r$ . Para ello se utiliza la fórmula de la distancia entre dos puntos y se compara la distancia calculada con el radio  $r$  de la circunferencia:

$$\text{distancia} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (7)$$

1. Si la distancia es mayor que el radio  $r$ , el punto  $T$  está fuera de la circunferencia.
2. Si la distancia es igual al radio  $r$ , el punto  $T$  está en el borde de la circunferencia.
3. Si la distancia es menor que el radio  $r$ , el punto  $T$  está dentro del área de la circunferencia.

#### Diseño de la solución:

1. Solicitar al usuario las coordenadas del centro de la circunferencia  $(x_1, y_1)$ , las coordenadas del punto  $T$   $(x_2, y_2)$ , y el radio  $r$ .
2. Calcular la distancia entre el centro y el punto  $T$  utilizando la fórmula de distancia entre dos puntos:

$$\text{distancia} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (8)$$

3. Verificar si la distancia calculada es mayor que el radio  $r$ :
  - a) Si es mayor, el punto  $T$  está fuera de la circunferencia.
4. Verificar si la distancia calculada es igual al radio  $r$ :
  - a) Si es igual, el punto  $T$  está en el borde de la circunferencia.
5. Verificar si la distancia calculada es menor que el radio  $r$ :
  - a) Si es menor, el punto  $T$  está dentro del área de la circunferencia.
6. Presentar el resultado al usuario.

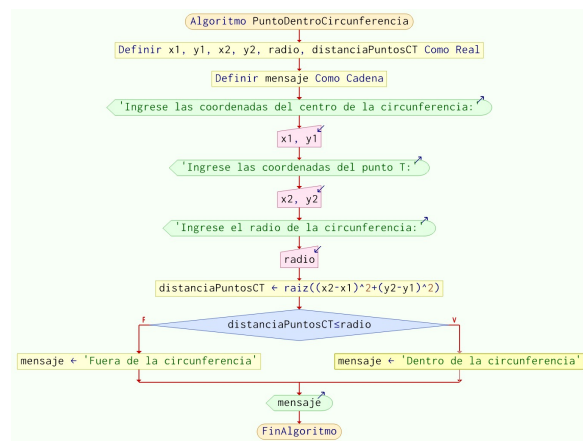


Figura 9. Diagrama de flujo.

#### Desarrollo de la solución:

Programa en Java.

Se utiliza un objeto Scanner para obtener datos de entrada desde la consola.

Se solicitan al usuario las coordenadas del centro de una circunferencia (punto C), el radio de la circunferencia y las coordenadas de un punto T. Los valores ingresados se almacenan en arreglos de cadenas de caracteres y se cierra el objeto Scanner para liberar recursos.

```
Scanner datos = new Scanner(System.in);
System.out.println("Ingresa las
    coordenadas del centro de una
    circunferencia (punto C) separado por
    una coma (x,y):");
String[] puntoC =
    (datos.nextLine()).split(",");
System.out.println("Ingresa el radio de la
    circunferencia:");
float radio = datos.nextFloat();
datos.nextLine();
System.out.println("Ingresa los valores de
    \"x\" y \"y\" del punto a evaluar
    (punto T) separado por una coma
    (x,y):");
String[] puntoT =
    (datos.nextLine()).split(",");
datos.close();
```

Se convierten las coordenadas del punto C y del punto T de cadenas de caracteres a números enteros (int). Luego, se utiliza la fórmula de la distancia para calcular la distancia entre los puntos C y T, almacenando el resultado como un número de punto flotante (float).

```
int xc =
    Integer.parseInt(puntoC[0].trim());
int yc =
    Integer.parseInt(puntoC[1].trim());
int xt =
    Integer.parseInt(puntoT[0].trim());
int yt =
    Integer.parseInt(puntoT[1].trim());
float distanciaPuntosCT =
    (float) Math.sqrt(Math.pow(xt-xc, 2) +
    Math.pow(yt-yc, 2));
```

Se declara una cadena de caracteres vacía llamada "mensaje".

Se utiliza una estructura condicional para determinar la posición relativa del punto T con respecto a la circunferencia y asignar un mensaje correspondiente. Finalmente, se imprime el mensaje en la consola.

```
String mensaje = "";
if (distanciaPuntosCT > radio) {
    mensaje = "El punto T("+xt+", "+yt+")
        esta fuera de la circunferencia";
} else if (distanciaPuntosCT == radio) {
    mensaje = "El punto T("+xt+", "+yt+")
        esta en la circunferencia";
} else {
    mensaje = "El punto T("+xt+", "+yt+")
        esta dentro de la circunferencia";
}
System.out.println(mensaje);
```

$(x_1, y_1)$ $(x_2, y_2)$	Radio	Resultado
(3, 4) (9, 2)	10	El punto T está adentro
(5, 2) (8, 1)	12	El punto T está adentro
(34, 23) (-90, 35)	67	El punto T está afuera
(-27, 3) (34, -5)	30	El punto T está afuera
(2, 8) (4, 9)	17	El punto T está adentro
(0, 0) (0, 0)	5	El punto T está en el borde

Figura 10. Tabla de pruebas

## Depuración y pruebas:



## Resolución Problema 4

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

### DESCRIPCIÓN DEL PROBLEMA

Implica comprender y definir claramente el problema que se quiere resolver. En este caso, se trata de convertir un número decimal a binario.

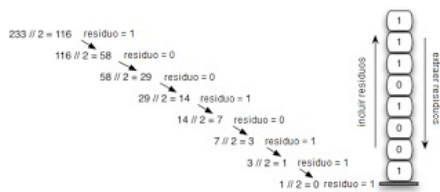


Figura 11. Decimal a binario

### DEFINICIÓN DE SOLUCIÓN

En esta etapa, se determina el enfoque o algoritmo general que se utilizará para convertir el número decimal a binario.

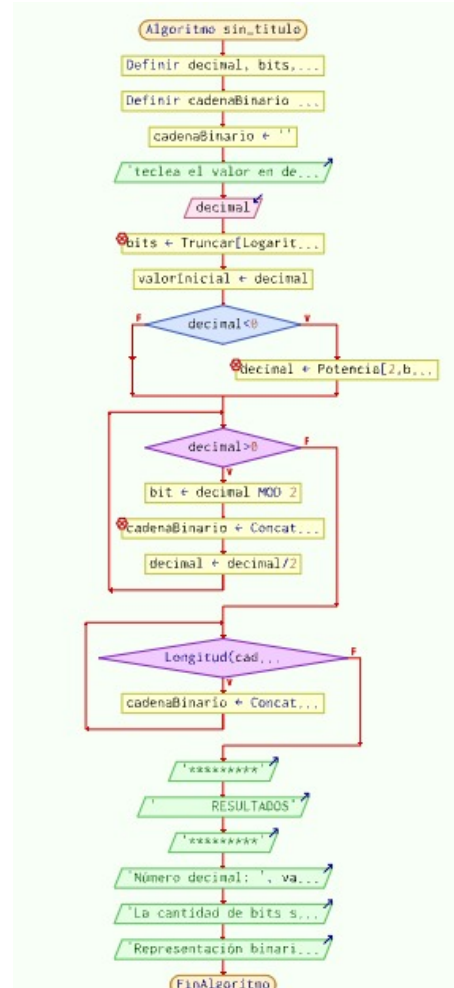


Figura 12. Diagrama de flujo que muestra el proceso para resolver

### DISEÑO DE SOLUCIÓN

#### Programa en Java

Se detallan los pasos específicos y la lógica necesaria para implementar el algoritmo definido anteriormente. Se pueden utilizar diagramas, pseudocódigo u otras herramientas para representar visualmente el flujo de la solución.

**solicita al usuario que teclee el valor decimal**

#### Desarrollo de la solución:

// Comienza solicitando al usuario dos puntos  $A(x_1, y_1)$  y  $B(x_2, y_2)$ .

```

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

```
public class Programacorrecto {
    public static void main(String[] args) {
        Scanner h=new Scanner(System.in);
        System.out.println("Teclea el valor en
            decimal : ");
    }
}
```

**calcula los bits del numero entero que dio el usuario**

```
int decimal=h.nextInt();
int bits = (int) (Math.log(Math.abs(decimal))
    /
    Math.log(2)) + 1;//calcular los bits del
    numero entero.
```

**Verificar si el número es negativo (-)**

```
int valorinicial=decimal;

// Verificar si el n mero es negativo (-)
if (decimal < 0)
```

**Obtener la representación en complemento a 2**

```
// Obtener la representacin en complemento
a 2
decimal = (int) (Math.pow(2, bits) +
    decimal);
}
```

**Convertir Numero decimal a binario**

```
// Convertir Numero decimal a binario
StringBuilder binary = new StringBuilder();
do {
```

**Obtener el bit menos significativo**

```
// Obtener el bit menos significativo
int bit = decimal % 2;
```

**Agregar el bit al inicio de la cadena**

```
// Agregar el bit al inicio de la cadena
binary.insert(0, bit);
```

**Dividir el número por 2**

```
// Dividir el n mero por 2
decimal /= 2;
} while (decimal > 0);
```

**Ajustar la longitud al número de bits especificado**

```
// Ajustar la longitud al n mero de bits
especificado
while (binary.length() < bits) {
    binary.insert(0, '0');
}
```

**convertir a cadena**

```
String binario =binary.toString(); //
```

**final mente se obtienen los resultados**

```
System.out.println("      RESULTADOS      ");
System.out.println("El N mero decimal es
    : \t" + valorinicial);
System.out.println("La cantidad de bits
    son: \t" + bits);
System.out.println("El numero binario es
    : \t" + binario);
}
```

## DEPURACIÓN Y PRUEBAS

verifica y corrige cualquier error que pueda existir en el código. Se realizan pruebas exhaustivas utilizando diferentes casos de prueba, incluyendo números decimales enteros positivos y negativos, para garantizar que la conversión a binario funcione correctamente en todos los escenarios posibles.



Figura 13. CORRIDA

Número de prueba	Número entero (positivo o negativo)	Número binario
1	234	11101010
2	-567	0111001001
3	890	1101111010
4	-123	0000101
5	-987	0000100101

Figura 14. TABLA DE CORRIDAS

## Resolución Problema 5

La conversión de números binarios a decimales es un proceso fundamental en el campo de la informática y las matemáticas. Para obtener la conversión de binario a decimal, se empleó la siguiente fórmula inicialmente:

$$D = \sum_{i=0}^{n-1} (B[i] \cdot 2^i) \quad (9)$$

En esta fórmula, cada dígito binario  $B[i]$  se multiplica por la potencia correspondiente de la base binaria, que es 2. El exponente  $i$  varía desde 0 hasta  $n - 1$ , donde  $n$  representa el número total de dígitos en la representación binaria. Al sumar estos productos, se obtiene el número decimal  $D$ , que es el resultado de la conversión.

### Descripción del problema:

1. Tomar el número binario de  $n$  bits como entrada.
2. Recorrer cada bit del número binario de izquierda a derecha.
3. Para cada bit, multiplicar su valor (0 o 1) por la potencia de 2 correspondiente a su posición.
4. Sumar todos los resultados de las multiplicaciones realizadas en el paso anterior.
5. El resultado obtenido es el equivalente en decimal del número binario de entrada.

### Definición de solución:

La solución para convertir un número binario a decimal se basa en la siguiente fórmula:

$$D = \sum_{i=0}^{n-1} (B[i] \cdot 2^i) \quad (10)$$

En esta fórmula, se multiplican los bits del número binario  $B$  por las potencias correspondientes de 2, según su posición en la secuencia. Luego, se suman los productos obtenidos. El resultado de esta suma es el número decimal equivalente al número binario.

### Desarrollo de la solución:

En esta sección se comienza solicitando al usuario que ingrese un número binario válido, compuesto únicamente por los dígitos 0 y 1.

```
Scanner sc = new Scanner(System.in);
// Se le pide al usuario que ingrese un número binario.
System.out.println("Ingrese por favor el número binario (solo están permitidos 0 y 1): ");
String binario = sc.nextLine();
```

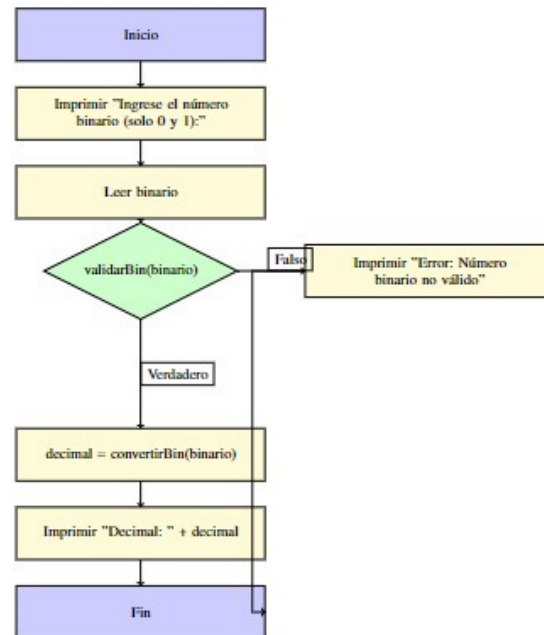


Figura 15. Diagrama de flujo.

Se llama a `validarBin` para verificar si el número binario ingresado es válido. Recorre cada dígito y verifica si es 0 o 1. Devuelve `false` si encuentra otro dígito, indicando que el número es inválido, y `true` si todos los dígitos son 0 o 1, continuando la ejecución.

```
// Condición para que solo se puedan agregar los caracteres 0 y 1.
if (!validarBin(binario))
    System.out.println("Error: El número binario ingresado contiene dígitos no válidos.");
return;
```

Después de la validación, se llama a `convertirBin` para convertir el número binario a decimal. Esta función toma el número binario como entrada y devuelve su equivalente en decimal.

```
int decimal = convertirBin(binario);
System.out.println("La conversión a número decimal es: " + decimal);
```

Se llama al método `convertirBin()` pasando el número binario como argumento. Este método realiza la conversión del número binario a decimal utilizando un bucle.

```
for (int i = 0; i < longitud; i++) {
    char digito = binario.charAt(i);
    int valor = Character.getNumericValue(digito);
    decimal += valor * Math.pow(2, exponente);
    exponente--;
}
return decimal;
```

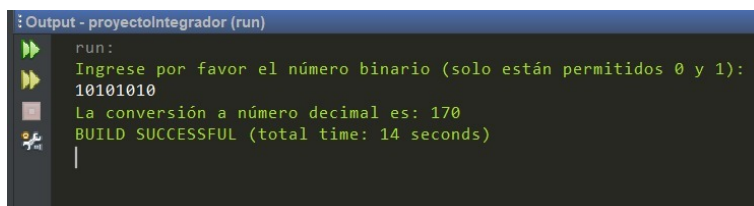
Se define un método llamado `validarBin` que se utiliza para validar si un número binario solo contiene los caracteres '0' y '1'. El método se utiliza para validar si un número binario es válido antes de realizar la conversión a decimal.

```

    }
    // Metodo para validar si un n mero binario solo
    // contiene los caracteres 0 y 1.
    public static boolean validarBin(String binario) {
        for (int i = 0; i < binario.length(); i++) {
            char digito = binario.charAt(i);
            if (digito != '0' && digito != '1') {
                return false;
            }
        }
        return true;
    }
}

```

### Depuración y pruebas:



```

Output - proyectoIntegrador (run)
run:
Ingrese por favor el número binario (solo están permitidos 0 y 1):
10101010
La conversión a número decimal es: 170
BUILD SUCCESSFUL (total time: 14 seconds)

```

**Figura 16.** Depuración del código en java del programa binario a decimal.

Número de prueba	Número binario	Resultado en decimal
1	10101	21
2	110010	50
3	1110001	113
4	10011010	154

**Figura 17.** Tabla de conversión de numero binaria a decimal.

## CONCLUSION

En conclusion, los programas presentados demuestran un rendimiento satisfactorio en sus funciones fundamentales. Sin embargo, existe margen para mejorarlos al incorporar información adicional pertinente a cada problema específico. Esta mejora sería beneficiosa para lograr una mayor precisión y comprensión de los resultados obtenidos, permitiendo así adaptar los programas de acuerdo a las necesidades individuales de los usuarios.

## Referencias

Graficando Ecuaciones con Forma Pendiente-Intersección. (s/f). Nroc.org. Recuperado el 25 de noviembre de 2023, de <https://content.nroc.org/Algebra.HTML5/U04L1T3/TopicText/es/text.html>

Ecuación de la recta en su forma pendiente - intersección. (2016, abril 1). Mi Profe. <https://miprofe.com/ecuacion-de-la-recta-en-su-forma-pendiente-interseccion/>

Gráfica de una recta. (s/f). Unam.Mx. Recuperado el 25 de noviembre de 2023, de [http://prepa8.unam.mx/academia/colegios/matematicas/paginacolmate/applets/matematicas\\_V/Applets\\_Geogebra/graficarectas.html](http://prepa8.unam.mx/academia/colegios/matematicas/paginacolmate/applets/matematicas_V/Applets_Geogebra/graficarectas.html)

Repaso de las gráficas de rectas en la forma pendiente-ordenada al origen (artículo). (s/f). Khan Academy. Recuperado el 25 de noviembre de 2023, de <https://es.khanacademy.org/math/algebra/x2f8bb11595b61c86:forms-of-linear-equations/x2f8bb11595b61c86:graphing-slope-intercept-equations/a/graphing-lines-review>

## Referencias

/u/anaserradobayes. (2016, diciembre 20). Representación gráfica de ecuaciones de segundo grado. GeoGebra. <https://www.geogebra.org/m/kYUBXrkw>

Castro, E. N. F. (2020). Ecuaciones de segundo grado. Con-Ciencia Boletín Científico de la Escuela Preparatoria No. 3, 7(13), 24-26. <https://repository.uaeh.edu.mx/revistas/index.php/prepa3/article/view/5179/6629>

## Referencias

Pardiño, F. F. (2023b, febrero 2). Distancia entre dos puntos ¿cómo calcularla? [Ejercicios resueltos]. Recuperado de <https://matetres14.com/distancia-entre-dos-puntos/>

Recta tangente a una circunferencia. secante de una circunferencia. recta exterior a una circunferencia. (s.

f.-c). Recuperado de [https://calculo.cc/temas/temas\\_geometria/fig\\_geometrica/teoria/circunferencia.html](https://calculo.cc/temas/temas_geometria/fig_geometrica/teoria/circunferencia.html)

## Referencias

Mano, M. M., Gonzalez, J. F. (2003). Diseño digital. Pearson Educación. [http://aniei.org.mx/paginas/uam/Descargas/Recursos/Diseno\\_Digital\\_de\\_Morris\\_Mano.pdf](http://aniei.org.mx/paginas/uam/Descargas/Recursos/Diseno_Digital_de_Morris_Mano.pdf)

de la Computación, H. A. V. M. P. de C. (s/f). Representación de números en binario. Uson.mx. Recuperado el 23 de noviembre de 2023, de <http://euler.mat.uson.mx/~havillam/ca/Common/Capitulo2.pdf>

Ingeniería, P. P. [@Pasosporingenieria]. (2022, agosto 6). NÚMEROS BINARIOS NEGATIVOS (Representación COMPLEMENTO A 2). Youtube.[https://www.youtube.com/watch?v=JQyN0D\\_2X\\_M](https://www.youtube.com/watch?v=JQyN0D_2X_M)

## Referencias

Sistemas Binarios y Decimales. (s. n.). Recuperado de <https://ed.team/blog/sistemas-binarios-y-decimales>

Maol, M. (2022, abril). CONVERTIR de BINARIO a DECIMAL - Ejercicio 1 [Video]. YouTube. 3:36 minutos. <https://www.youtube.com/watch?v=-4rUKINeCEs>

Khan Academy. (2023, 25 de noviembre). Números binarios (artículo). Recuperado de <https://es.khanacademy.org/computing/ap-computer-science-principles/x2d2f703b37b450a3:digital->

## BIOGRAFIAS

**Freyra Wendy Martinez Martinez**, tiene 18 años y vive en Tlaxcoapan, Hidalgo. Desde los 14 años ha demostrado interés por resolver problemas relacionados con la tecnología. Su meta a corto plazo es adquirir habilidades de programación sólidas y destacarse como estudiante por sus conocimientos, para así completar su carrera universitaria como ingeniera. En su tiempo libre, le gusta jugar videojuegos y pasar tiempo con su mascota.

<https://github.com/Freyramartinez>

**Mera Ibarra Carol**, es una estudiante de Ingeniería en Tecnologías de la Información y Comunicaciones con interés en explorar diferentes áreas de la carrera y encontrar una especialidad. También tiene interés en el trabajo remoto. Su objetivo principal es completar cada semestre de la carrera de manera satisfactoria.

<https://github.com/CarolMeralbarra>

**Dulce Karen Gonzalez Yañez** , es estudiante de Ingeniería en Tecnologías de la Información y Comunicaciones. Nació en la Ciudad de México pero creció y actualmente vive en Actopan, Hidalgo. Tiene una actitud positiva, es perseverante y está dispuesta a enfrentar desafíos técnicos. La meta principal de Karen es convertirse en una desarrolladora web altamente competente y tener la capacidad de crear aplicaciones web escalables y de alta calidad. Aspira a combinar sus habilidades técnicas con un enfoque centrado en el usuario para crear experiencias web excepcionales.

<https://github.com/GonzalezKarencin>

**Mendoza Calva Marco Antonio** , es estudiante de Ingeniería en Tecnologías de la Información y Comunicaciones. Nació y creció en Mixquiahuala de Juárez, Hidalgo. Desde temprana edad mostró un gran interés por poder acceder a la universidad y cumplir sus metas. Aunque sus intereses pueden parecer diversos, busca la manera de alcanzar sus objetivos. Estos medios le han enseñado la importancia de la perseverancia, la resolución de problemas y el trabajo en equipo. Su objetivo principal es completar sus estudios universitarios en ITICs, con especialización en ciberseguridad y hacking. Sueña con trabajar en el campo de la ciberseguridad, protegiendo sistemas e información vital.

<https://github.com/MARCOANTONIOMMMMM>

**Eduardo Gómez Mejía** , Desde su infancia, este estudiante ha sentido un profundo interés por los avances tecnológicos que ocurren en el mundo. Nacido en Actopan, una ciudad en el estado de Hidalgo, actualmente estudia la Ingeniería en Tecnologías de la Información y Comunicaciones. Su pasión por aprender constantemente lo impulsa a explorar nuevas áreas de conocimiento. Específicamente, siente una gran curiosidad por temas como la ciberseguridad, las redes y el Internet de las cosas.

<https://github.com/GomezEdu>