

# Software de resolución de problemas de Ingeniería

Chávez Atanacio Yael Antonio, *Instituto Tecnológico Superior del Occidente del Estado de Hidalgo, Mixquiahuala, Hgo., 42700, Mexico*

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

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

Maldonado Olguin Irving, *Instituto Tecnológico Superior del Occidente del Estado de Hidalgo, Mixquiahuala, Hgo., 42700, Mexico*

*Abstract—Este informe presenta una revisión de diversos conceptos matemáticos y sus aplicaciones prácticas. Los conceptos abordados incluyen la determinación de la ecuación de una recta a partir de dos puntos, la resolución de raíces en números reales, la distancia entre dos puntos, la verificación de la posición de un punto respecto a una circunferencia, y la conversión de números decimales a binarios. El informe proporciona una visión integral de estos conceptos, destacando su importancia y utilidad en el mundo real. Los ejemplos y ejercicios presentados permiten a los lectores comprender cómo aplicar estos conceptos en sus propias áreas de interés.*

La resolución precisa y eficiente de problemas matemáticos es fundamental en diversos campos. En este artículo, exploraremos la aplicación de la metodología de las 6 D's en la resolución de problemas matemáticos. A través de estos casos de estudio, demostraremos cómo la metodología de las 6 D's se aplica en la determinación de la ecuación de una recta y el cálculo de ángulos, el cálculo de distancias y la verificación de puntos dentro de una circunferencia, así como la conversión de números decimales a binarios. La metodología se presenta como una guía efectiva y sistemática para resolver problemas matemáticos, proporcionando una estructura sólida que asegura la comprensión del problema, el diseño adecuado de soluciones, la implementación de algoritmos eficientes, la depuración del código, la documentación clara y la evaluación exhaustiva de los resultados. Los casos de estudio presentados, se evidencia la importancia de seguir esta metodología para obtener resultados confiables y de calidad en la resolución de problemas matemáticos.

## Resolución Problema 1

### 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 propuesta emplea tres funciones matemáticas clave. Primero, se calcula la inclinación de la recta mediante la función `inclinación`, luego se determina la intersección en el eje  $Y$  con la función `intersección`, y finalmente se calcula el ángulo interno  $\alpha$  con la función `ángulo`. Estas funciones trabajan en conjunto para proporcionar la ecuación de la recta y el ángulo deseado.

### Diseño de la solución:

Para el diseño de la solución observamos en la ecuación de la recta, si dos puntos distintos  $P(x_1, y_1)$  y  $Q(x_2, y_2)$  se ubican en la curva  $y = f(x)$ , la pendiente de la recta secante que une los dos puntos es:

$$m_{sec} = \frac{y_2 - y_1}{x_2 - x_1} = \frac{f(x_2) - f(x_1)}{x_2 - x_1} \quad (1)$$

La forma punto-pendiente de la ecuación de la recta, con una coordenada específica en el plano cartesiano se define como:

$$b = y_1 - m * x_1$$

(2)

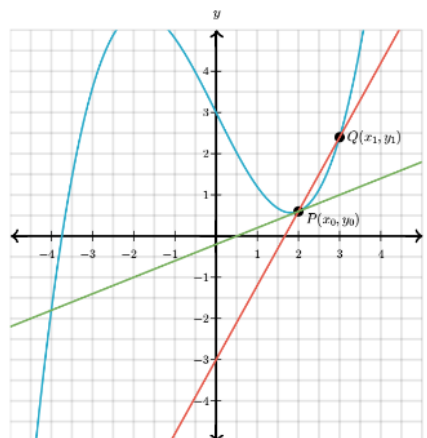


Fig. 1. Gráfica de la ecuación de la recta

Figura 1. Gráfica de la ecuación de la recta

Utilizando este método, es posible encontrar la ecuación de la recta a partir de dos puntos. Recordando que si los dos puntos son idénticos la recta será una línea vertical.

El algoritmo de solución para encontrar la ecuación de la recta pendiente (ec. ) comienza solicitando al usuario dos puntos  $P(x_1, y_1)$  y  $Q(x_2, y_2)$ .

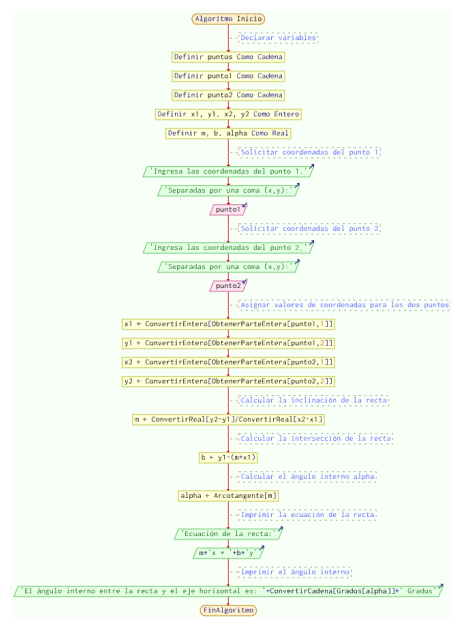


Figura 2. Diagrama de flujo usado.

## Desarrollo de la solución:

En el primer fragmento, se solicitan las coordenadas de dos puntos, A y B, que se ingresan en formato (x, y). Las coordenadas se leen como cadenas para separar las componentes x e y. Finalmente, se cierra el objeto Scanner para liberar los recursos.

```
Scanner puntos = new Scanner (System.in);

//Solicitar puntos para la ecuacion de la recta
System.out.println("""
    Ingrese las coordenadas
    del punto 1.
    separadas por una coma
    (x,y):
    """);

String[] punto1 =puntos.nextLine().split(",");

System.out.println("""
    Ingrese las coordenadas
    del punto 1.
    separadas por una coma
    (x,y):
    """);

String[] punto2 =puntos.nextLine().split(",");

//cerrar el escaneo
puntos.close();
```

Las coordenadas separadas se convierten de cadenas a números enteros utilizando Integer.parseInt(). El método trim() se utiliza para eliminar cualquier espacio en blanco que pueda haber alrededor de las coordenadas.

```
//Asignar valor de coordenadas a x,y para dos
puntos
int x1= Integer.parseInt(punto1[0].trim());
int y1= Integer.parseInt(punto1[1].trim());

int x2= Integer.parseInt(punto2[0].trim());
int y2= Integer.parseInt(punto2[1].trim());
```

Aquí se calcula la pendiente ( $m$ ) de la recta utilizando la fórmula

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

y luego se calcula la intersección en el eje Y ( $b$ ) utilizando la fórmula. (ec. ) La ecuación de la recta resultante es

$$y = mx + b.$$

```
//Calculo para la inclinacion de la recta
Double m = (double) (y2 - y1) / (x2 - x1);

//Calcular Interseccion de la recta
Double b= y1 -(m * x1);
```

Se calcula el ángulo interno ( $\alpha$ ) entre la recta y el eje horizontal utilizando la función Math.atan( $m$ ). El resultado se convierte de radianes a grados.

```
// Calcular el ngulo interno $\alpha$
Double alpha = Math.atan(m);
```

Finalmente, se imprime la ecuación de la recta y el ángulo interno en grados. La ecuación se imprime en formato  $mx + by$ , y el ángulo se imprime en grados.

```
//Imprimir ecuacion de la recta

System.out.println("Ecuacion de la recta
    igual= \n" +
    m + " x + " + b + " y ");
//Imprimir el resultado de el ngulo
interno
System.out.println("El ngulo interno
    entre la recta y el eje horizontal es:
    "
    + String.format("%.3f",
    Math.toDegrees(alpha)) + "
    Grados");
```

La función String.format() formatea un número decimal según el formato especificado en la cadena de formato. (En este caso 3f es el formato que indica que el número debe redondearse a tres) decimales

## Depuración y pruebas:

Prueba	Punto 1	Punto 2	Pendiente (m)	Intersección (b)	Ángulo
1	(1, 2)	(3, 4)	1.0	1.0	45.00
2	(-1, 0)	(2, -3)	-1.0	1.0	-45.00
3	(-4, 1)	(1, 4)	0.6	3.4	30.96

**Cuadro 1.** Pruebas para el programa de ecuación de la recta

## 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 está en el conjunto de los números complejos.

### Descripción del problema:

El problema planteado se centra en la resolución de una ecuación cuadrática y la determinación de sus posibles soluciones. La ecuación cuadrática tiene la forma general  $ax^2 + bx + c$ , donde  $a$ ,  $b$  y  $c$  son los coeficientes dados. El objetivo es aplicar la fórmula general:

$$x = -b \pm \sqrt{b^2 - 4ac} / 2a \quad (3)$$

Para obtener las soluciones de la ecuación. Para determinar la naturaleza de las soluciones, se utiliza el discriminante, que está dado por la expresión

$$b^2 - 4ac \quad (4)$$

Dependiendo del valor del discriminante, se obtienen tres casos posibles:

- Si el discriminante  $> 0$  se dan dos soluciones
- Si el discriminante  $= 0$  se da una solución
- Si el discriminante es  $< 0$  la solución se encuentra en el conjunto de los números complejos.

Estos casos permiten clasificar las soluciones de la ecuación cuadrática en función de su naturaleza y son determinantes para comprender las propiedades y características de la ecuación en cuestión.

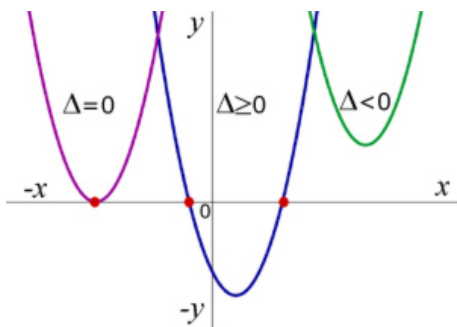


Figura 3. Discriminante

### Definición de solución:

Se plantea la resolución de una ecuación cuadrática en la forma

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

donde se procede al cálculo del discriminante y a la evaluación de tres condiciones. En caso de que el discriminante sea mayor que cero, se llevan a cabo dos operaciones y se presenta el resultado correspondiente. Si el discriminante es igual a cero, se realiza una operación y se muestra el resultado obtenido. Por otro lado, si el discriminante resulta ser negativo, se informa al usuario que la ecuación en cuestión no cuenta con soluciones reales.

Acto seguido se muestra el diagrama de flujo, el cual es la base del programa.

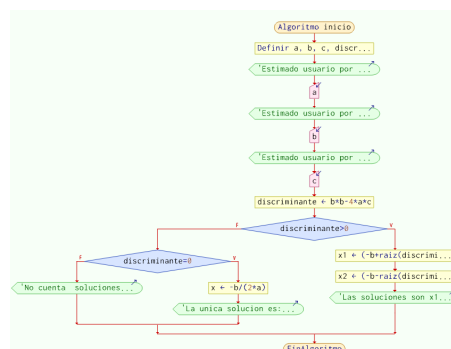


Figura 4. Diagrama de flujo

### Diseño de la solución:

Se plantea la resolución de una ecuación cuadrática en la forma

$$ax^2 + bx + c \quad (6)$$

donde se procede al cálculo del discriminante y a la evaluación de tres condiciones. En caso de que el discriminante sea mayor que cero, se llevan a cabo dos operaciones y se presenta el resultado correspondiente. Si el discriminante es igual a cero, se realiza una operación y se muestra el resultado obtenido. Por otro lado, si el discriminante resulta ser negativo, se informa al usuario que la ecuación en cuestión no cuenta con soluciones reales.

Acto seguido se muestra el diagrama de flujo, el cual es la base del programa.

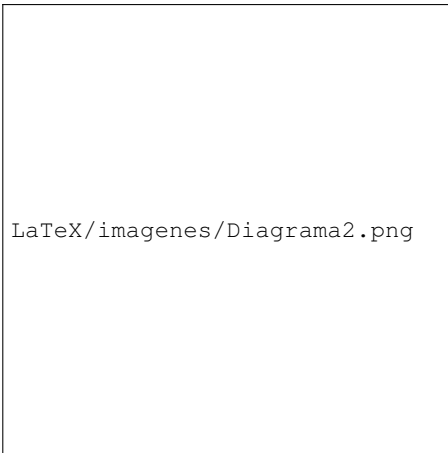


Figura 5. Diagrama de flujo

### Desarrollo de la solución:

eguidamente se mostrara el programa de java, donde se trabajo con la formula general. Para ello, se ocupan las librería Math y el objeto Scanner.

```
import java.util.Scanner;
```

```
Crear un objeto Scanner para leer la
entrada del usuario
\begin{javaCode}
Scanner coeficiente = new
Scanner(System.in);
```

Solicitar al usuario que ingrese el valor de A

```
System.out.println("Estimado usuario, por
favor ingrese el valor de A: ");
double a = coeficiente.nextDouble();
```

Solicitar al usuario que ingrese el valor de B

```
System.out.println("Estimado usuario, por
favor ingrese el valor de B: ");
double b = coeficiente.nextDouble();
```

Solicitar al usuario que ingrese el valor de C

```
System.out.println("Estimado usuario, por
favor ingrese el valor de C: ");
double c = coeficiente.nextDouble();
```

Cierra el objeto Scanner

```
coeficiente.close();
```

Calcula el discriminante de la ecuación cuadrática

```
double discriminante = b * b - 4 * a * c;
```

El objetivo es verificar si el discriminante de una ecuación cuadrática es mayor que cero. Si se cumple esta condición, se determina que la ecuación tiene dos soluciones. Luego, se resuelven las soluciones y se muestra el resultado al usuario.

```
if (discriminante > 0) {
    double x1 = (-b +
        Math.sqrt(discriminante)) / (2 *
        a);
    double x2 = (-b -
        Math.sqrt(discriminante)) / (2 *
        a);

    System.out.println("Las soluciones son
        x1 = " + x1 + " y x2 = " + x2);
}
```

Verificar si el discriminante es igual a 0, si cumple con la condición, realiza la operación donde solo existe una única solución y la muestra al usuario

```
else if (discriminante == 0)

    double x = -b / (2 * a);

    System.out.println("La solucion es: "
        + x);
```

En caso de que el discriminante sea negativo, se determina que la ecuación cuadrática no tiene soluciones reales. Esta información se comunica al usuario.

```
else {
    System.out.println("no cuenta con
        soluciones reales.");
}
```

### Depuración y pruebas:

del mismo. Donde se muestra los diferentes resultados posibles.

No.	A	B	C	Resultado
1	19	17	18	no cuenta con soluciones reales.
2	10	15	5	La solución es: $x_1 = -0,5$ y $x_2 = -1,0$
3	1	-2	1	La solución es: $x = 1$
4	2	4	-6	La solución es: $x_1 = 1,0$ y $x_2 = -3,0$
5	1	1	1	No cuenta con soluciones reales.

### 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  $(x_2, y_2)$  esta dentro del área de la circunferencia.

#### Descripción del problema:

El problema implica calcular la distancia de dos puntos. Donde el punto C y el punto T pueden ser definidos en el plano cartesiano  $(X, Y)$  y  $R$  es definida así los cuatro ejes, además debemos tomar en cuenta que el punto T se puede encontrar dentro o fuera de la circunferencia.

#### Definición de solución:

En el primer segmento considerando las variables  $X_1, X_2$  y  $Y_1, Y_2$  se plantea la siguiente solución se emplea la formula: distancia:

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

Con esta formula se calcula la distancia entre los dos puntos (C y T). Posteriormente se verifica que el radio ingresado sea positivo, en dado caso de que el radio sea negativo se utilizara la formula:  $\text{Radio} * -1$ ; con esta formula se convierte el radio negativo a positivo, finalmente verificamos que el punto se encuentre dentro de la circunferencia utilizando el radio.

-Observamos como se plasma en el plano cartesiano utilizando la formula de la distancia

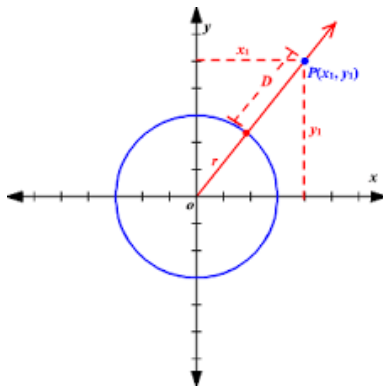


Figura 6. Gráfica del programa

#### Diseño de la solución:

label=0.

1. Considerando las variables  $X, Y$  de cada punto calcula la distancia entre los dos puntos utilizando la siguiente formula: distancia =  $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ .

2. Considerando que la variable  $r$  sea positiva se procede al siguiente paso, de lo contrario se aplicara la siguiente ecuacion  $r * -1$ .
  3. considerando que la distancia sea menor al radio se puede decir que el punto T se encuentra dentro de la circunferencia, de lo contrario se considera que el punto se encuentra fuera de la circunferencia.
- A continuación veremos el diagrama de flujo que fue la base para el desarrollo del programa.

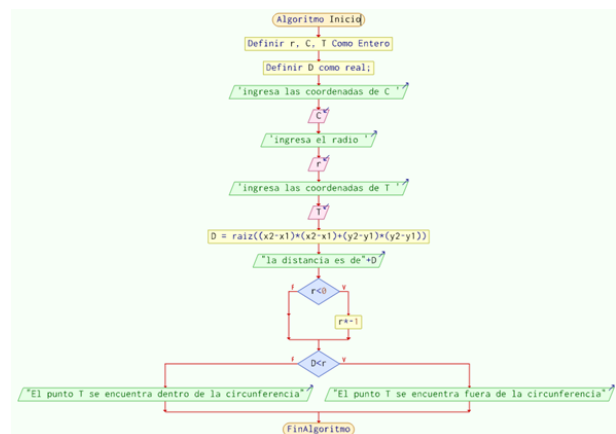


Figura 7. Diseño del programa

#### Desarrollo de la solución:

A continuación, se muestra el código en Java para calcular la distancia entre dos puntos y verificar si se encuentran dentro de una circunferencia:

Primero se solicita al usuario las coordenadas del punto C  $(x_1, y_1)$

```
Scanner dato = new Scanner(System.in);
System.out.print("Ingrese las coordenada
del punto C primero x1: ");
int x1 = dato.nextInt();
System.out.print("Ingrese las coordenada
del punto C despues y1: ");
int y1 = dato.nextInt();
```

se solicita el radio de la circunferencia

```
System.out.println("Ingrese el radio de la
circunferencia: ");
int radio = dato.nextInt();
```

solicitamos al usuario las coordenadas del punto T  $(x_2, y_2)$

```
System.out.print("Ingrese las coordenada
del punto T X2: ");
int x2 = dato.nextInt();
System.out.print("Ingrese las coordenada
del punto T Y2: ");
int y2 = dato.nextInt();
```

Se realiza el cálculo la distancia entre el punto C y el punto T con ayuda de la formula antes mencionada.

```
double distancia = Math.sqrt((x2 - x1) *
    (x2 - x1) + (y2 - y1) * (y2 - y1));
```

Se imprime el resultado del calculo de la distancia.

```
System.out.println("la distancia es de
    "+distancia);
```

Se verifica si el radio es positivo o negativo, si es negativo lo pasamos a positivo

```
int radio2=radio*-1;
```

Se calcula, si la distancia resulta menor o igual al radio entonces el punto T se encontrara dentro de la circunferencia, de lo contrario el punto T se encontrara fuera de la circunferencia .

```
if(radio<0){

    int radio2=radio*-1;
    if (distancia <= radio2) {
        System.out.println("El punto T est
            dentro de la circunferencia");
    } else {
        System.out.println("El punto T no
            est dentro de la
            circunferencia");
    }
}

if(radio>0){

    if (distancia <= radio) {
        System.out.println("El punto T est
            dentro de la circunferencia");
    } else {
        System.out.println("El punto T no
            est dentro de la
            circunferencia");
    }
}

}
```

No. prueba	X1	Y1	radio	X2	Y2	resultado
1	1	2	-5	2	1	La distancia es de 1.4142135623730951 El punto T está dentro de la circunferencia
2	-3	2	8	4	3	La distancia es de 7.0710678118654755 El punto T está dentro de la circunferencia
3	6	-4	3	-3	2	La distancia es de 10.816653826391969 El punto T no está dentro de la circunferencia
4	2	5	10	7	4	La distancia es de 5.0990195135927845 El punto T está dentro de la circunferencia
5	-7	-3	-6	-1	-5	La distancia es de 6.324555320336759 El punto T no está dentro de la circunferencia

Figura 8. Tabla de pruebas

## Depuración y pruebas:

Se realizaron 5 pruebas al programa para validar su funcionamiento, a continuación veremos los siguientes resultados

### Resolución Problema 4

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

#### Descripción del problema:

Consiste en diseñar un enfoque matemático que dado un número decimal, facilite su conversión a su equivalente en binario. Esta problemática requiere el uso de principios fundamentales para abordar eficazmente el desafío en cuestión. Desarrollando un código en Java que emplee una solución formal y eficiente que realice la conversión precisa de números decimales a binarios, incorporando conceptos algebraicos y numéricos relevantes para lograr una comprensión clara y estructurada del proceso.

#### Definición de solución:

La solución propuesta para el problema se basa en la aplicación de un proceso matemático eficiente y estructurado donde se emplean tres funciones fundamentales para lograr la conversión precisa.

En primer lugar, la función `divisiónSucesiva` se encarga de realizar la división sucesiva por 2 del número decimal, registrando los residuos en el proceso. Después, la función `registroResiduos` utiliza estos residuos para construir la representación binaria del número. Finalmente, la función `obtenerBinario` organiza los resultados para la representación binaria. Para convertir un número decimal a binario mediante la división sucesiva por 2, puedes utilizar la siguiente ecuación.  $\text{residuo } m - \text{cociente } n - 1 \text{ mód } 2$ .

Donde  $\text{cociente}_n$  es el cociente en la  $n$ -ésima división, y  $\text{cociente}_{n-1}$  es el cociente de la división anterior.

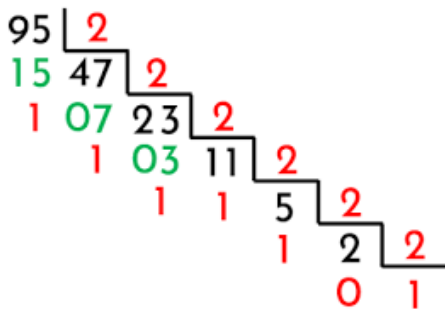


Figura 9. Calcula número Binario

#### Diseño de la solución:

Para poder entender como debemos empezar el código tenemos que empezar por diseñar el diagrama de flujo

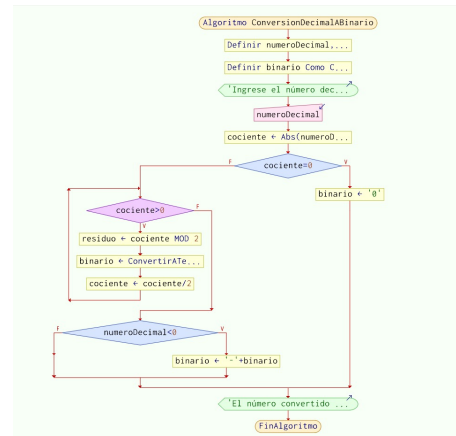


Figura 10. Diagrama de flujo para la solución

#### Desarrollo de la solución:

En este fragmento, se utiliza un Scanner para obtener la entrada del usuario, que es un número decimal entero, ya sea positivo o negativo

```

Scanner in = new Scanner(System.in);
System.out.println("Ingrese el número decimal entero positivo o negativo:");
int numeroDecimal = in.nextInt();
in.close();

```

Se inicializa una cadena binario para almacenar la representación binaria y se crea una variable numero que se utilizará para trabajar con el valor absoluto del número decimal.

```

String binario = "";
int numero = numeroDecimal;

```

Aquí si el número es negativo, se multiplica por -1 para obtener su valor absoluto

```

//Numero Negativo
if (numero < 0) {
    numero *= -1;
}

```

//Conversión a Binario (Parte Positiva) Se realiza la conversión a binario utilizando un bucle while.

```

if (numero > 0) {
    while (numero != 0) {
        int residuo = numero % 2;
        binario = residuo + binario;
        numero = numero / 2;
    }
}

```



Si el número original es cero, se establece la representación binaria como "0".

```
//Manejo de caso "0"
(
    if (numero == 0) {
        binario = "0";
    }
}
```

Si el número original es negativo, se busca el índice del último '1' en la cadena binaria invertida. Se invierte y modifica la cadena binaria para cambiar los '0' por '1' y viceversa. Se agrega el signo negativo al resultado final.

```
// Inversión y Modificación de la Cadena Binaria
(Parte Negativa)
if (numeroDecimal < 0) {
    // Obtener el índice del último '1' en la
    // cadena binaria invertida
    int indiceUltimoUno = 0;
    for (int i = binario.length() - 1; i >= 0; i--) {
        if (binario.charAt(i) == '1') {
            indiceUltimoUno = i;
            break;
        }
    }

    // Invertir y modificar la cadena binaria para
    // números negativos
    String binarioInvertido = binario.substring(0,
        indiceUltimoUno);
    binarioInvertido =
        binarioInvertido.replaceAll("1", "#");
    binarioInvertido =
        binarioInvertido.replaceAll("0", "1");
    binarioInvertido =
        binarioInvertido.replaceAll("#", "0");

    // Agregar el signo negativo al resultado
    binarioInvertido = "-" + binarioInvertido +
        binario.substring(indiceUltimoUno);
    binario = binarioInvertido;
}
```

Finalmente se imprime el resultado final en formato binario. (positivo o negativo) y maneja adecuadamente los casos especiales, como cero y números negativos.

```
//Mostrar el resultado
System.out.println("\nEl número convertido a
    binario es: " + binario);
```

### Depuración y pruebas:

Se realizaron las pruebas de decimal a binario para encontrar errores

Número Decimal	Número Binario
1	01
5	01
3	0

## Resolución Problema 5

Dado un numero binario de  $n$  bits regresa su equivalente a decimal.

### Descripción del problema:

El problema consiste en convertir un número binario de  $n$  bits a su equivalente en decimal. Un número binario está compuesto únicamente por los dígitos 0 y 1, mientras que un número decimal utiliza la base 10 y está compuesto por los dígitos del 0 al 9.

La tarea es tomar el número binario de entrada, que tiene una longitud específica de  $n$  bits, y convertirlo a su representación decimal correspondiente. Esto implica realizar un cálculo que involucra la suma ponderada de los dígitos binarios, donde cada dígito se multiplica por la potencia de 2 correspondiente a su posición en el número binario.

### Definición de solución:

La solución para convertir un número binario de  $n$  bits a su equivalente en decimal consiste en recorrer cada bit del número binario, multiplicar su valor por 2 elevado a la potencia correspondiente según su posición, y sumar los resultados. Al finalizar el recorrido, el valor obtenido será el equivalente decimal del número binario. Es importante verificar la validez del número binario y asegurarse de que esté compuesto únicamente por 0 y 1. El resultado se muestra como el equivalente decimal del número binario.

### Diseño de la solución:

El programa solicita al usuario que ingrese un número binario de  $n$  bits. Se asegura de que el número ingresado consista únicamente en dígitos binarios 0 y 1. Si se detecta algún dígito decimal, el programa no funcionará.

Luego, se realiza la conversión del número binario ingresado a su equivalente decimal utilizando la fórmula de posición y peso explicada anteriormente.

Finalmente, el programa imprime el resultado de la conversión, es decir, el número decimal equivalente al número binario ingresado por el usuario.

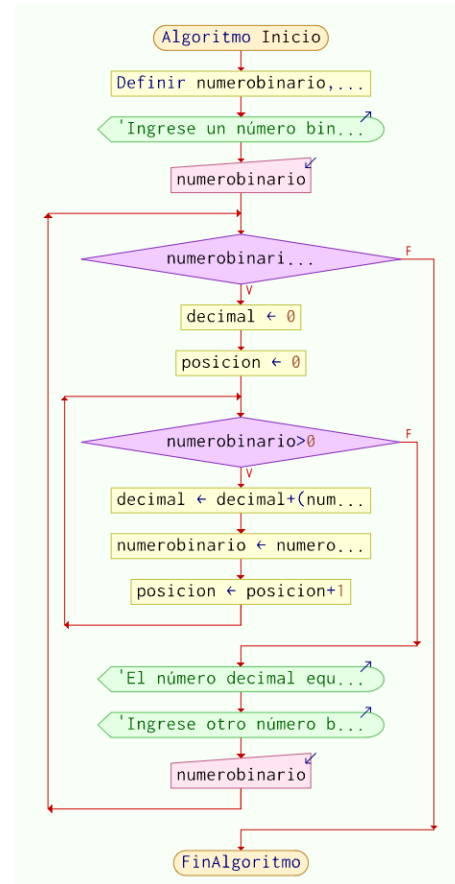


Figura 11. Diagrama de flujo usado como base .

### Desarrollo de la solución:

El desarrollo del código del programa en Java para la conversión de números binarios a decimales:

```

import java.util.Scanner;
import java.math.BigInteger;

public class NewClass {
    public static void main(String[] args) {
        Scanner bin = new Scanner(System.in);
        boolean continuar = true;
  
```

En esta paso en el programa son colocadas las variables *Scanner BigInteger* las cuales ayudaran al correr el programa

Se solicita al usuario ingresar un numero binario

```

// Bucle que permite al usuario ingresar n meros
// binarios hasta que ingrese 'x'

while (continuar) {

    System.out.print("Ingrese un n mero
                      en binario (o 'x' para terminar ):
                      ")

    String entrada = bin.nextLine();
  
```



## Resolución Problema 6

## CONCLUSION

En conclusión, este artículo ofrece una guía integral para abordar diversos procesos matemáticos, proporcionando comprensión detallada y herramientas prácticas. La eficaz conversión de números binarios a decimales y el cálculo de distancias entre puntos son destacados, respaldados por un código en Java y un enfoque cuidadoso en la presentación de resultados. Además, la implementación de bucles y operaciones matemáticas en la conversión de números decimales a binarios y la propuesta modular para determinar la ecuación de una recta y su ángulo interno demuestran precisión y eficiencia. Se enfatiza la importancia de mantener una estructura clara en la presentación de resultados y se sugiere que las referencias a figuras y tablas se realicen en el cuerpo del artículo, no en la conclusión. Asimismo, se alienta a los autores a considerar las posibles direcciones futuras de investigación para ampliar las soluciones propuestas a otros problemas matemáticos.

## AGRADECIMIENTOS

Agradecemos sinceramente a aquellos que contribuyeron al desarrollo de este artículo sobre la aplicación de la metodología de las 6 D's en la resolución de problemas matemáticos.

Queremos expresar nuestro agradecimiento al profesor Neri Pérez Giovany Humberto por sus valiosas asesorías en el uso de LaTeX, fundamental para la presentación clara y profesional de los conceptos matemáticos. A Eunice Santiago Manzano le reconocemos y agradecemos por proporcionarnos las herramientas necesarias para la programación en Java, resultando en soluciones eficientes en los casos de estudio. También extendemos nuestro agradecimiento a Agustín Soto Arista por su contribución en la redacción adecuada y corrección de estilo del artículo. Finalmente, agradecemos al doctor Cuadros Romero Francisco Javier por su orientación y apoyo fundamentales para la realización de este trabajo. Su guía y conocimientos fueron esenciales para el éxito de este proyecto.

## REFERENCES

ónReferencias del problema 11-Fórmula de la distancia. (2011, February 20). <https://es.khanacademy.org/math/geometry/hs-geo-analytic-geometry/hs-geo-distance-and-midpoints/v/distance-formula>  
[https://es.khanacademy.org/math/geometry/hs-](https://es.khanacademy.org/math/geometry/hs-geo-analytic-geometry/hs-geo-distance-and-midpoints/v/distance-formula)

geo-analytic-geometry/hs-geo-distance-and-midpoints/v/distance-formula

Referencias del problema 22-Repaso del discriminante (artículo). (s/f). Khan Academy. Recuperado el 22 de noviembre de 2023, de <https://es.khanacademy.org/math/algebra/x2f8bb11595b61c86:quadratic-functions-equations/x2f8bb11595b61c86:quadratic-formula-a1/a/discriminant-review3-Introducción>. (2016, abril 20). Portal Académico del CCH. <https://e1.portalacademico.cch.unam.mx/alumno/matematicas2/unidad1/form>  
Comprender la fórmula de la cuadrática (artículo). (s/f). Khan Academy. Recuperado el 21 de noviembre de 2023, de <https://es.khanacademy.org/math/algebra/x2f8bb11595b61c86:quadratic-functions-equations/x2f8bb11595b61c86:quadratic-formula-a1/a/quadratic-formula-explained-article>  
Referencias del problema 35- Fórmula de la distancia (artículo). (s/f). Khan Academy. Recuperado el 21 de noviembre de 2023, de <https://es.khanacademy.org/math/geometry/hs-geo-analytic-geometry/hs-geo-distance-and-midpoints/a/distance-formula.6-Fórmula-de-la-distancia>. (2011, February 20). <https://es.khanacademy.org/math/geometry/hs-geo-analytic-geometry/hs-geo-distance-and-midpoints/v/distance-formula7>  
<https://es.khanacademy.org/math/geometry/hs-geo-analytic-geometry/hs-geo-distance-and-midpoints/v/distance-formula7>  
Referencias del problema 48-Menezes, P. M. S. (2018). Binary to Decimal Conversion Algorithm. International Journal of Computer Science and Information Security (IJCSIS), 16(3), 104-108. DOI: 10.5281/zenodo.12170829-Li, K. H. (2012). Binary to Decimal Conversion Algorithm and Its Application. En Proceedings of the 2012 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC) (pp. 202-205). DOI: 10.1109/CyberC.2012.52  
Referencias del problema 510-El, P. (2016, diciembre 28). Sistemas de <https://matemelga.wordpress.com/2016/12/28/sistemas-de-numeracion-posicionales/11-El>, P. (2016, diciembre 28). Sistemas de [https://matemelga.wordpress.com/2016/12/28/sistemas-de-numeracion-posicionales/12-\(S/f\)](https://matemelga.wordpress.com/2016/12/28/sistemas-de-numeracion-posicionales/12-(S/f)). Java.com. Recuperado el 22 de noviembre de 2023. [https://www.java.com/es/download/help/whatis\\_java.html](https://www.java.com/es/download/help/whatis_java.html)

**Chávez Atancio Yael Antonio** Es un estudiante de la ingeniería en Tecnologías de la Información y Comunicaciones con una pasión por los E-Sports, escuchar y crear musica de todos los generos. Nacido

y criado en Mixquiahuala de Juárez Hidalgo, desde muy pequeño quiso tener una computadora para poder diseñar presentaciones y hacer trabajos simples en Word y Paint. Aunque parezca que no tiene intereses, tiene intereses distintos, Yael encuentra la paz y tranquilidad escuchando musica es Melomano. Esto le ha enseñado la importancia de aprender de sus errores y no carese. El objetivo de Yael es terminar la carrera en ITICs es para que le digan Inge aunque aun no sabe que rama estudiar, pronto la encontrara.

**Hernandez Martinez Brayan** Es un estudiante del instituto ITSOEH de la carrea TICS, le gusta mucho la musica y tambien le gusta mucho hacer musica, siempre le gustaron las tecnologias desde pequeño se dedico a su investigacion. Nacido en Mixquiahuala de Juarez, criado en Tlaxcoapan Hidalgo, su objetivo es terminar sus estudios.

**Cruz Martinez Alejandro** Es un estudiante del instituto ITSOEH de la carrea TICS, le gusta mucho salir de paseo y la musica de banda, siempre le gusto diseñar paginas web desde la secundaria. Nacio en Tlaxcoapan Hidalgo, fue criado en chilcuautla Hidalgo, su objetivo es tener una empresa.

**Maldonado Olguin Irving** Estudiante del Instituto Tecnológico Superior del Estado de Hidalgo, en la carrera de Tics, le apasiona el ejercicio y la música, además de interesarle las tecnologías y desea aprender más. Nacido en el municipio de Mixquiahuala de Juárez, Hgo. Siendo una persona bastante tranquila, es bastante amable.