

# Software de resolución de problemas de Ingeniería

1er Calva Abraham

*Estudiante de iTIC'S*

*ITSOEH*

*Mixquiahuala, Hgo.*

*230110637@itsoeh.edu.mx*

2<sup>do</sup> Vidal Ceron Jean Paul

*Estudiante de iTCS'S*

*ITSOEH*

*Mixquiahuala, Hgo.*

*230110625@itsoeh.edu.mx* 3<sup>o</sup> Martinez Escamilla Erick

*Estudiante de iTCS'S*

*ITSOEH*

*Texcatepec, Hgo.*

*230110626@itsoeh.edu.mx*

4<sup>o</sup> Barrera Rojo Roberto

*Estudiante de iTCS'S*

*ITSOEH*

*Mixquiahuala, Hgo.*

*230110946@itsoeh.edu.mx*

5<sup>o</sup> Jacob Salas Andrea

*Estudiante de iTCS'S*

*ITSOEH*

*Santiago de Anaya, Hgo.*

*230110449@itsoeh.edu.mx* ,

## Índice

<b>Ecuación de la recta</b>	4
Descripción del problema . . . . .	4
Definición del problema . . . . .	4
Diseño de la solución . . . . .	4
Desarrollo de la solución . . . . .	4
Depuración de pruebas . . . . .	5
<b>Formula cuadrática</b>	5
Descripción del problema . . . . .	5
Definición del problema . . . . .	5
Diseño de la solución . . . . .	5
Desarrollo de la solución . . . . .	5
Depuración de pruebas . . . . .	6
<b>Circunferencia</b>	6
Descripción del problema . . . . .	6
Definición de la solución . . . . .	6
Diseño de la solución . . . . .	6
Desarrollo de la solución . . . . .	7
Depuración y pruebas . . . . .	7
<b>Conversión decimal a binario</b>	7
Descripción del problema . . . . .	7
Definición de la solución . . . . .	7
Diseño de la solución . . . . .	8
Desarrollo de la solución . . . . .	8
Depuración de pruebas . . . . .	8
<b>Conversión binario a decimal</b>	8
Descripción del problema . . . . .	8
Definición de la solución . . . . .	8
Diseño de la solución . . . . .	9
Desarrollo de la solución . . . . .	9
Depuración de pruebas . . . . .	9
<b>Tabla de verdad</b>	9
Descripción del problema . . . . .	9
Definición de la solución . . . . .	9
Diseño de la solución . . . . .	9
Desarrollo de la solución . . . . .	9
Depuración de las pruebas . . . . .	10
<b>CONCLUSION</b>	10
<b>AGRADECIMIENTOS</b>	11
<b>Referencias</b>	11
<b>Referencias</b>	11



<b>Biographies</b>	11
Jacob Salas Andrea	11
Calva Abraham	11
Martinez Escamilla Erick	11
Vidal Ceron Jean Paul	11
Barrera Rojo Roberto	11

En el campo de la ingeniería, la resolución de problemas es esencial y requiere precisión, conocimiento y eficiencia. Este proyecto tiene como objetivo diseñar y desarrollar software especializado en la resolución de problemas de ingeniería. Cada software proporciona funciones clave para abordar problemas específicos. Se utiliza la metodología de las 6D, que incluye la descripción del problema, la definición de la solución, el diseño de la solución, el desarrollo de la solución, la depuración y las pruebas, y finalmente la documentación.

El propósito de este proyecto integrador es ofrecer herramientas que abarquen diversas áreas matemáticas relevantes en la ingeniería, con el fin de resolver problemas de manera eficiente y precisa, optimizando así el tiempo de trabajo.

## Ecuación de la recta

### Descripción del problema

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.

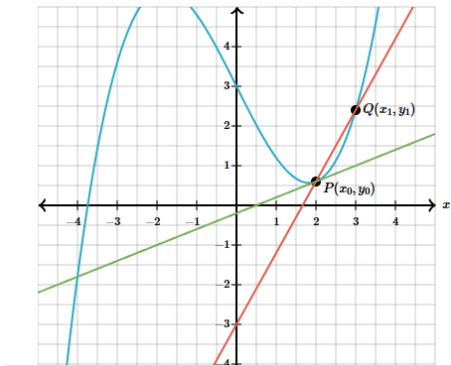


Figura 1. Gráfica de la ecuación de la recta.

### Definición del problema

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 uno los dos puntos es:

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

Para identificar la intersección en el eje vertical se utiliza cualquiera de los 2 puntos para este caso se

utilizo  $P(x_1, y_1)$  de la siguiente forma:

$$b = y_1 - m_{sec} * x_1 \quad (2)$$

utilizando este método, puedes encontrar la ecuación de la recta a partir de 2 puntos dados. y para calcular el ángulo de dicha pendiente se usa:

$$= \arctan(m) \quad (3)$$

el algoritmo de solución implementado en el lenguaje Java empieza por la entrada de los valores  $P(x_1, y_1)$  y  $Q(x_2, y_2)$  solicitando sean separados por una coma.

### Diseño de la solución

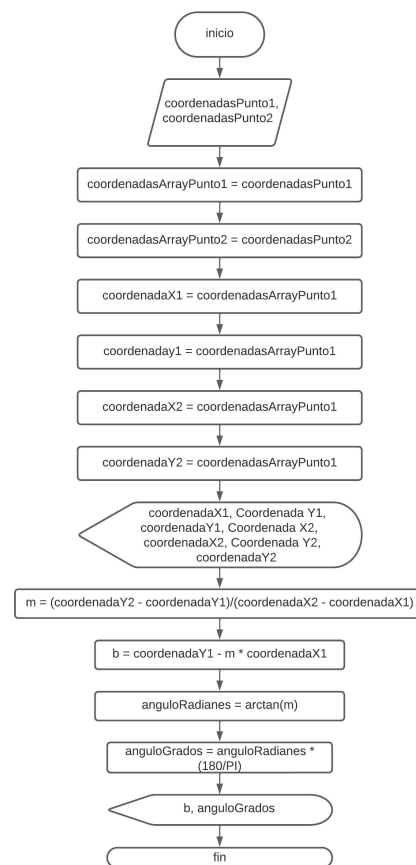


Figura 2. Representación gráfica del algoritmo implementando la solución

### Desarrollo de la solución

Para resolver el primer problema implementamos el siguiente código, en el cual pide al usuario que ingrese

las coordenadas el punto uno y dos separadas por una coma (,)

```
String coordenadasPunto1 = JOptionPane.showInputDialog("Ingrese las coordenadas del punto1(X y Y) separadas por una coma(,): ");
```

```
String coordenadasPunto2 = JOptionPane.showInputDialog("Ingrese las coordenadas del punto2(X y Y) separadas por una coma(,): ");
```

después separamos los valores que el usuario ingreso, para separarlos usamos el metodo split()

```
String[] coordenadasArrayPunto1 = coordenadasPunto1.split(","); String[] coordenadasArrayPunto2 = coordenadasPunto2.split(",");
```

después con los datos ya separados los imprimimos uno por uno según los valores que ingreso el usuario, para después empezar a calcular el ángulo interno y la intersección

```
JOptionPane.showMessageDialog(null, "Coordenada X1: - coordenadaX1 + "-Coordenada Y1: - coordenadaY1 + "- Coordenada X2: - coordenadaX2 + "- Coordenada Y2: - coordenadaY2);
```

```
double m = (coordenadaY2 - coordenadaY1)/(coordenadaX2 - coordenadaX1);
```

```
double b = coordenadaY1 - m * coordenadaX1;
```

```
double anguloRadianes = Math.atan(m); double anguloGrados = anguloRadianes * (180/Math.PI);
```

y por ultimo imprimimos los resultados del ángulo interno de la recta y de la intersección

```
JOptionPane.showMessageDialog(null, "La intersección de la recta: - b + "- .El ángulo interno: -anguloGrados);
```

## Depuración de pruebas

**Cuadro 1.** Tabla de corridas para el Ejercicio 1.

X1	Y1	X2	Y2	intersección	ángulo
1	2	2	3	1	45
2	4	6	8	2	0
2	4	6	13	1	63.4349
10	20	30	40	10	45
79	23	77	12	-372	78.6900

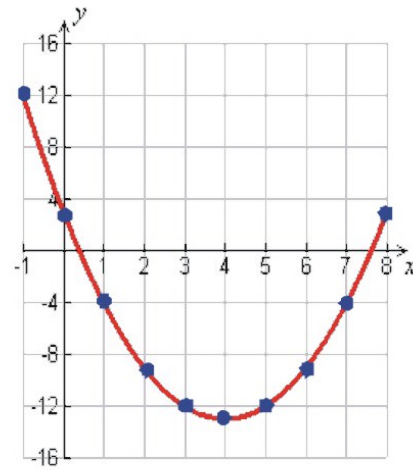
## Formula cuadrática

### 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 esta en el conjunto de los números complejos.

### Definición del problema

Primero, tenemos que verificar que la ecuación esté en la forma estándar:



**Figura 3.** Gráfica de la ecuación cuadrática

$$ax^2 + bx + c = 0 \quad (4)$$

donde  $a$ ,  $b$  y  $c$  son coeficientes reales y  $x$  es la variable desconocida. Posteriormente, se deben identificar los valores de  $a$ ,  $b$  y  $c$  en la ecuación. A continuación, se debe utilizar la fórmula cuadrática para encontrar las soluciones:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (5)$$

Posteriormente, se sustituyen los valores de  $a$ ,  $b$  y  $c$  en la fórmula cuadrática y se realizan los cálculos necesarios.

### Diseño de la solución

Se realizó una representación gráfica de un algoritmo para resolver el problema

### Desarrollo de la solución

Para solucionar este problema tendrá que resolver los tres coeficientes de los tres parámetros de una ecuación cuadrática que se tenga, donde se desarrollara primero el discriminante y después se evaluara el mismo para determinar si hay dos soluciones, una solución o si todo es falso, no existe, para ello se implemento Este código que solicita al usuario que ingrese los coeficientes de la ecuación cuadrática  $a$   $b$   $c$ .

```
float a = Float.parseFloat(JOptionPane.showInputDialog("Ingresa coeficiente respecto al parámetro
```

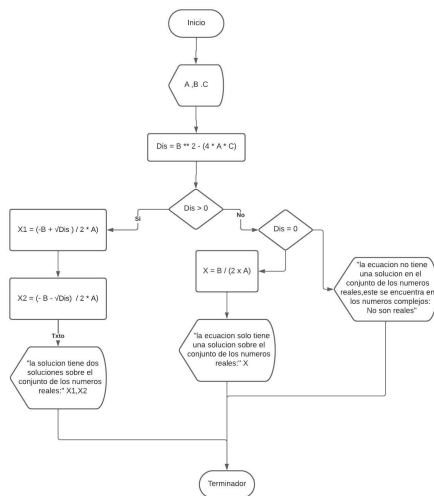


Figura 4. Diagrama de flujo problema 2.

float b = Float.parseFloat(JOptionPane.showInputDialog(Ingresa coeficiente respecto al parámetro b)); float c = Float.parseFloat(JOptionPane.showInputDialog(Ingresa coeficiente respecto al parámetro c));

posterior a lo que hicimos se va a calcular el discriminante con la ecuación cuadrática.

double discriminante = Math.pow(b, 2) - (4 \* a \* c);

Para finalizar la discriminante pasa por la primera condición donde se obtendrán dos soluciones, la segunda solución solo nos dará una solución, en caso diferente no hay solución posible.

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

JOptionPane.showMessageDialog(null, "La solución tiene dos soluciones sobre el conjunto de los números reales:- "x1 = - x1 + "- "x2 = - x2); else if (discriminante == 0) double x = -b / (2 \* a); JOptionPane.showMessageDialog(null, "La ecuación solo tiene una solución sobre el conjunto de los números reales:- "x = - x);

## Depuración de pruebas

### Circunferencia

#### Descripción del problema

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.

a	b	c	$x_1, x_2$ o x
2	10	2	-0.20, -4.79
7	8	9	"La ecuación no tiene una solución en el conjunto de los números reales, este se encuentra en los números complejos."
-5	6	-1	0.2, 1.0
3	464	46	"La ecuación no tiene una solución en el conjunto de los números reales, este se encuentra en los números complejos."
1	-4	4	2

Cuadro 2. Tabla de corridas.

#### Definición de la solución

La circunferencia es un punto que está a una misma distancia de un punto fijo llamado centro. La distancia de cada punto de la circunferencia al centro se llama radio.

$$(x - h)^2 + (y - k)^2 = r^2 \quad (6)$$

#### Diseño de la solución

Para resolver estas ecuaciones y determinar si un punto T con coordenadas  $(x_2, y_2)$  está dentro del área de la circunferencia con centro en el punto C el cual utiliza coordenadas  $(x_1, y_1)$  y radio, se calcula la distancia del punto T y el centro de la circunferencia utilizando la fórmula de la distancia entre dos puntos  $(x_1, y_1)$  y  $(x_2, y_2)$ .

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

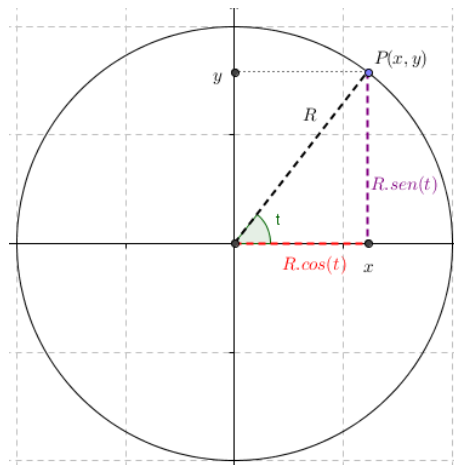


Figura 5. Gráfica de la ecuación.

1. Si  $d \leq r$ , el punto T está dentro del área de la circunferencia.

2. Si  $d > r$ , el punto T esta fuera del área de la circunferencia.

Se realizo un diagrama de flujo para saber como solucionar el problema.

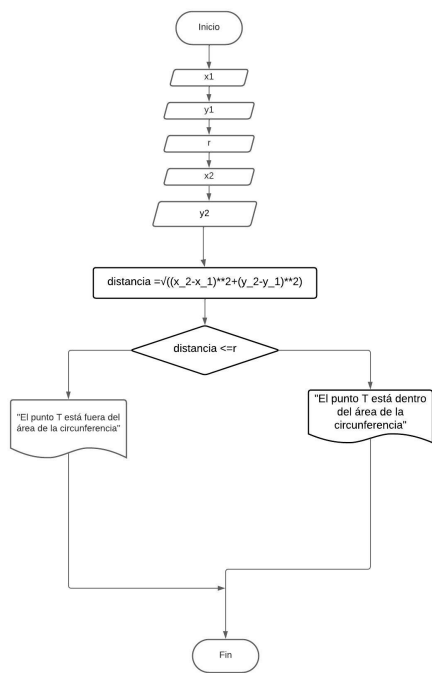


Figura 6. Diagrama de flujo de la solución

## Desarrollo de la solución

Para este problema se implemento este código en Apache NetBeans 14 con lenguaje en Java, donde al principio del bloque pide al usuario que ingrese las variables de la ecuación de la distancia entre dos puntos, las cuales son:  $x_1, y_1, r, x_2, y_2$

```
double x1 = Double.parseDouble(JOptionPane.showInputDialog("Ingrese la coordenada x1:"));
double y1 = Double.parseDouble(JOptionPane.showInputDialog("Ingrese la coordenada y1:"));
double r = Double.parseDouble(JOptionPane.showInputDialog("Ingrese el radio r de la circunferencia:"));
double x2 = Double.parseDouble(JOptionPane.showInputDialog("Ingrese la coordenada x2 del punto T:"));
double y2 = Double.parseDouble(JOptionPane.showInputDialog("Ingrese la coordenada y2 del punto T:"));

Después se programo el proceso del código el cual es la ecuación de la distancia.
double distancia = Math.sqrt(Math.pow(x2 - x1, 2) + Math.pow(y2 -
```

$y_1, 2)$ ); En el bloque final del código se presenta una condición, si la distancia es menor o igual al radio entonces el punto T está dentro del área de la circunferencia y si, no el punto T está fuera del área de la circunferencia.

```
if (distancia <= r) // Salida
JOptionPane.showMessageDialog(null, "El punto T está dentro del área de la circunferencia.");
else
JOptionPane.showMessageDialog(null, "El punto T está fuera del área de la circunferencia.");
```

## Depuración y pruebas

$x_1$	$y_1$	$r_x$	$x_2$	$y_2$	Fuera o Dentro
0	0	2	4	4	Fuera
0	1	2	3	4	Fuera
7	7	8	6	6	Dentro
20	10	9	5	5	Fuera
0	0	3	1	1	Dentro

Cuadro 3. Tabla de corridas.

## Conversión decimal a binario

### Descripción del problema

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

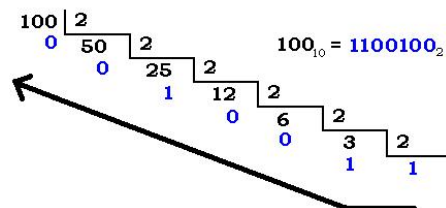


Figura 7. Conversión a Binario.

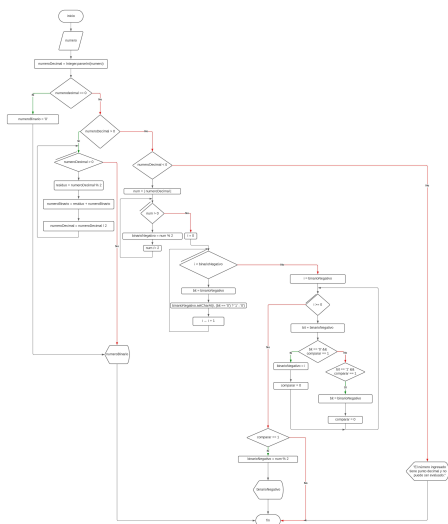
## Definición de la solución

Para obtener un número binario, se requiere realizar una serie de operaciones. Consiste en la división sucesiva del número decimal por dos, obteniendo el residuo en cada paso. Luego, los residuos se combinan en orden inverso para formar el número binario correspondiente. Este proceso se repite hasta que el cociente de la división sea igual a cero. Sin embargo, esta metodología presenta una problemática debido a su ineficiencia y el riesgo de cometer errores en los cálculos.

La solución planteada para abordar esta problemática ha sido la creación de una técnica computacional

en el lenguaje de programación Java. Este diseño automatiza y asegura la conversión de números enteros positivos y negativos a su equivalente en el sistema binario de manera precisa y confiable.

### Diseño de la solución



**Figura 8.** Representación gráfica del algoritmo implementando la solución

### Desarrollo de la solución

En este bloque de código, se solicita al usuario que ingrese un número entero, ya sea positivo o negativo, para luego utilizarlo. `import javax.swing.JOptionPane;`

```
public class decimalBinario
{
    public static void main(String[] args)
    {
        String numero = JOptionPane.showInputDialog("Ingrese un número decimal:");
        try
        {
            int numeroDecimal = Integer.parseInt(numero);
            String numeroBinario = "";
            // Conversión de números positivos
            if (numeroDecimal > 0)
            {
                while (numeroDecimal > 0)
                {
                    int residuo = numeroDecimal % 2;
                    numeroBinario = residuo + numeroBinario;
                    numeroDecimal = numeroDecimal / 2;
                }
            }
            // Conversión de números negativos (complemento a 2)
            else if (numeroDecimal < 0)
            {
                int absNumero = Math.abs(numeroDecimal);
                // Conversión de la parte positiva
                while (absNumero > 0)
                {
                    int residuo = absNumero % 2;
                    numeroBinario = residuo + numeroBinario;
                    absNumero = absNumero / 2;
                }
                // Obtención del complemento a 2
                int longitud = numeroBinario.length();
                String binarioNegativo = "";
                for (int i = 0; i < longitud; i++)
                {
                    char bit = numeroBinario.charAt(i);
                    binarioNegativo = (bit == '0' ? '1' : '0') + binarioNegativo;
                }
                // Suma de 1 al complemento a 2
                int comparador = 1;
                for (int i = longitud - 1; i >= 0; i--)
                {
                    char bit = binarioNegativo.charAt(i);
                    if (bit == '0' && comparador == 1)
                    {
                        binarioNegativo = '1' + binarioNegativo;
                        comparador = 0;
                    }
                    else if (bit == '1' && comparador == 1)
                    {
                        binarioNegativo = '0' + binarioNegativo;
                    }
                    else
                    {
                        comparador = 0;
                    }
                }
                numeroBinario = binarioNegativo;
            }
            // Mensaje de resultado
            JOptionPane.showMessageDialog(null, "El número binario equivalente es: " + numeroBinario);
        }
        catch (NumberFormatException e)
        {
            JOptionPane.showMessageDialog(null, "El número ingresado tiene punto decimal y no puede ser evaluado.");
        }
    }
}
```

En el siguiente bloque de código se realiza una comparación con el número ingresado anteriormente. Dentro de las estructuras condicionales, se encuentra un ciclo que se encarga de descomponer el número entero en su respectiva representación binaria. `if (numeroDecimal == 0) numeroBinario = "0"; else if(numeroDecimal >0) while (numeroDecimal >0) int residuo = numeroDecimal % 2; numeroBinario = residuo + numeroBinario; numeroDecimal = numeroDecimal / 2; else` En este bloque de código se presenta una condición en la que, si el número entero ingresado es negativo, se realizará la conversión a su representación binaria utilizando el método del complemento a 2. El complemento a 2 es una operación utilizada para representar números

negativos en sistemas binarios. `if(numeroDecimal <0) StringBuilder binarioNegativo = new StringBuilder(); int num = Math.abs(numeroDecimal); while (num >0) binarioNegativo.insert(0, num % 2; for (int i = 0; i <binarioNegativo.length(); i++) char bit = binarioNegativo.charAt(i); binarioNegativo.setCharAt(i, (bit == '0' ? '1' : '0'));`

`int comparador = 1; for (int i = binarioNegativo.length() - 1; i >= 0; i--) char bit = binarioNegativo.charAt(i); if (bit == '0' && comparador == 1) binarioNegativo.setCharAt(i, '1'); comparador = 0; else if (bit == '1' && comparador == 1) binarioNegativo.setCharAt(i, '0'); comparador = 1; if (comparador == 1) binarioNegativo.insert(0, '1');`

Posteriormente, en otro bloque de código, se muestra en una ventana emergente la equivalencia del número entero ingresado en su respectiva representación binaria. `JOptionPane.showMessageDialog(null, "El número binario equivalente es: " + binarioNegativo.toString()); return;` De lo contrario, si se ingresó un número entero negativo, en la siguiente línea de código se mostrará una ventana emergente que contendrá el número binario negativo generado mediante su representación en complemento a 2. `JOptionPane.showMessageDialog(null, "El número binario equivalente es: " + numeroBinario);` Por último, en caso de que el usuario haya ingresado un número decimal, se mostrará una ventana emergente con un mensaje. `JOptionPane.showMessageDialog(null, "El número ingresado tiene punto decimal y no puede ser evaluado.");`

### Depuración de pruebas

numero	numeroBinario / binarioNegativo
-42	010110
-28	00100
5	101
3	11
10.5	El número ingresado tiene punto decimal y no puede ser evaluado.

**Cuadro 4.** Tabla de corridas para el Ejercicio 4.

## Conversión binario a decimal

### Descripción del problema

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

### Definición de la solución

El proceso de conversión de un número binario a decimal se realiza de la siguiente manera: se enumeran los dígitos de derecha a izquierda, comenzando



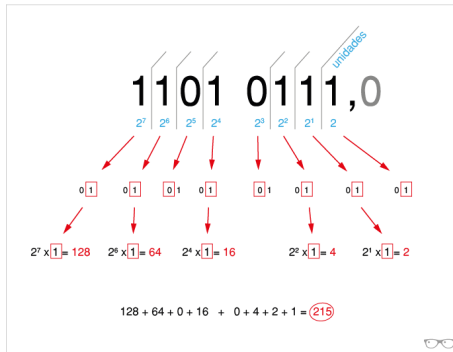


Figura 9. Conversión binario-decimal

desde la posición cero. Se utiliza como base el número 2 y se suma el valor correspondiente a cada posición. Es decir, se eleva la base (2) al exponente de la posición en la que se encuentra el dígito, pero solo se realiza la suma si en esa posición hay un 1. Si se encuentra un 0, no se realiza ninguna suma.

### Diseño de la solución

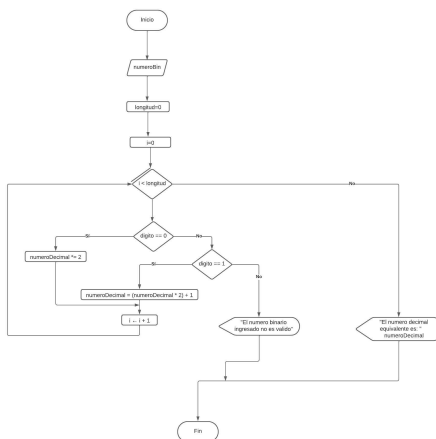


Figura 10. Diagrama de flujo del problema 4

### Desarrollo de la solución

En el siguiente bloque de código, se muestra una ventana flotante que pide al usuario que ingrese un número binario

```
String numeroBin = JOptionPane.showInputDialog("Ingresa un numero binario");
int longitud = numeroBin.length();
int numeroDecimal = 0;
```

Después se toma ese número y se compara si solo contiene ceros y unos, si es así el código hace las

respectivas operaciones para poder obtener el número decimal y en dado caso que el número ingresado contenga algo más que ceros y unos se mostrará el mensaje: "El número binario ingresado no es válido"

```
for (int i = 0; i < longitud; i++)
    char digito = numeroBin.charAt(i);
    if (digito == '0') numeroDecimal *= 2; else if (digito == '1') numeroDecimal = (numeroDecimal * 2) + 1; else
        JOptionPane.showMessageDialog(null, "El numero binario ingresado no es valido"); return;
```

Por último si el número ingresado solo contiene ceros y unos se muestra una ventana flotante con el resultado de la conversión binario-decimal

```
JOptionPane.showMessageDialog(null, "El numero decimal equivalente es - " + numeroDecimal);
```

### Depuración de pruebas

La siguiente tabla muestra el resultado de la compilación del código

Cuadro 5. Tabla de corridas para el Ejercicio 5.

Binario	Decimal
111	7
10101010	170
675	.El numero binario ingresado no es valido"
110101	53
2	.El numero binario ingresado no es valido"

### Tabla de verdad

### Descripción del problema

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

### Definición de la solución

Para llevar a cabo los teoremas booleanos, es necesario contar con datos específicos previamente obtenidos. Estos datos se obtienen a partir de una tabla de verdad que ha sido previamente creada y resuelta. Una vez que se disponen de los datos, es posible utilizar los teoremas booleanos con el fin de simplificar la expresión obtenida de la tabla de verdad, sin que esto afecte el resultado.

### Diseño de la solución

### Desarrollo de la solución

En el siguiente bloque de código, se muestra una ventana donde pide al usuario que inserte el número de bits. `public static void main(String[] args) Set<Integer>filasCambiar = new HashSet<>();`

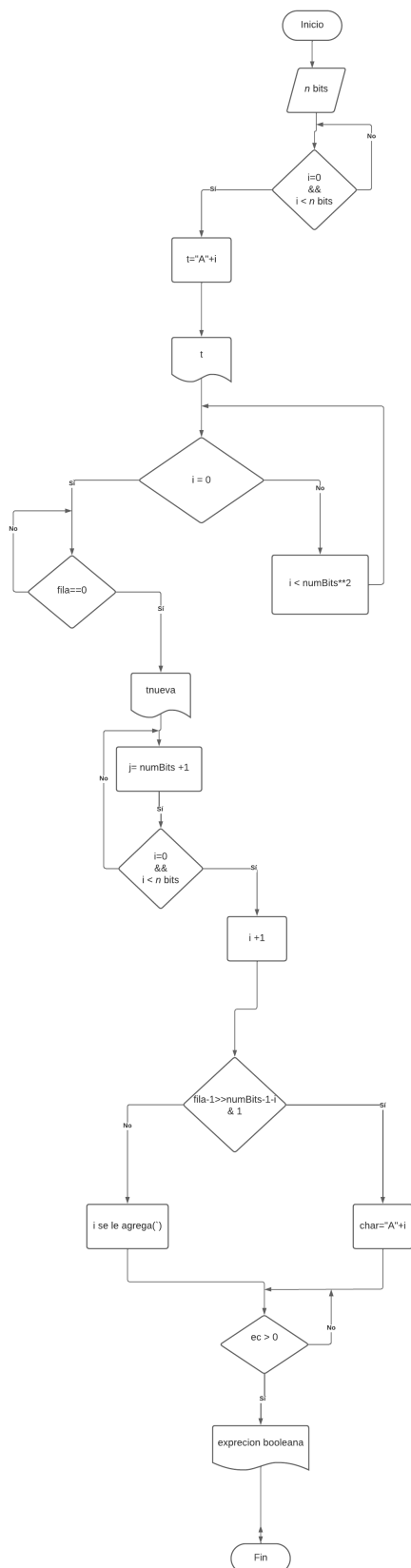


Figura 11. Diagrama de Tabla de Verdad del problema 6

```
String numBitsStr = JOptionPane.showInputDialog("Ingrese el Número de Bits que deberá ser la Tabla de Verdad:");
int numBits = Integer.parseInt(numBitsStr);
```

### Depuración de las pruebas

Ingrese el número de la fila que desea cambiar el resultado a 1 (1-8) o ingrese 0 para finalizar: 0  
 Expresiones booleanas al finalizar:  $A' + B' + C'$   $A' + B' + C$   $A' + B + C$   $A + B' + C$   $A + B + C'$   $A + B + C$ .

Expresión booleana final:  $(A' + B' + C') + (A' + B' + C) + (A' + B + C) + (A + B' + C) + (A + B + C') + (A + B + C)$

En la siguiente tabla se muestra lo antes explicado:

Cuadro 6. Tabla de corridas para el Ejercicio 6.

A	B	C	RESULTADO
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

### CONCLUSION

En este documento se desarrollaron seis problemáticas de ingeniería en las cuales se realizó un exhaustivo análisis e implementación de diferentes soluciones. Estas soluciones buscan facilitar la vida de la comunidad escolar, especialmente en conceptos relacionados con la tecnología, las redes y la electrónica. El objetivo principal es brindar un gran apoyo mediante soluciones prácticas y acciones concretas que tengan un impacto positivo en aspectos clave. Además, se enfoca en abordar desafíos importantes y promover un enfoque innovador centrado en el desarrollo integral de los estudiantes.

Posteriormente, se propone brindar oportunidades de desarrollo profesional que les permitan adquirir conocimientos en temas relevantes y adaptarse a las necesidades del usuario para lograr resultados más eficientes. Esto significa que a medida que el usuario utiliza el sistema, este se vuelve cada vez más preciso. Esta orientación nos ha enseñado la importancia de la creatividad, la cooperación y la implementación práctica de los conocimientos para poder entregar un trabajo bien elaborado.

## AGRADECIMIENTOS

Queremos expresar nuestro agradecimiento a la Ing. Citlali Azucena Martínez Calva tutora de nuestro grupo por el conocimiento brindado en programación y por la paciencia dada a los alumnos que desconocían sobre el tema, también al Mtro. José Martín Oropeza Méndez el cual brindo de su tiempo y transmitió sus conocimientos sobre matemáticas discretas, igual agradecemos al Doc. Francisco Javier Cuadros Romero por compartir sus experiencias y conocimientos sobre las Tics lo cual fue fundamental para obtener datos significativos y valiosos para este proyecto, agradecemos al Ing. Giovany Humberto Neri Pérez por el gran apoyo que dio al darnos accesorios para este proyecto, por la paciencia, tiempo y comprensión así como enseñarnos más sobre cálculo diferencial, también agradecemos al Lic. Agustín Soto Arista por contribuir a este proyecto, su orientación en fundamentos de investigación lo cual ayudó a redactar este proyecto y se agradece al Lic. Juan Cornejo Hernández Jefe de División de Ingeniería en TIC's por el apoyo, sus valiosos comentarios a lo largo del semestre, la ayuda y disposición que tuvo para nosotros. A todos ustedes, nuestros más sinceros agradecimientos, sin su apoyo este proyecto no habría sido posible.

## Referencias

1. Khan Academy, October 20, 2023, Tangent lines and rates of change, <https://www.khanacademy.org/math/calculus-all-old/taking-derivatives-calc/using-the-formal-definition-of-derivative-calc/a/tangent-lines-and-rates-of-change>
2. Fórmula de la distancia (video). (n.d.). Khan Academy. <https://es.khanacademy.org/math/geometry/hs-geo-analytic-geometry/hs-geo-distance-and-midpoints/v/distance-formula>
3. Comprender la fórmula de la cuadrática (artículo) | Khan Academy. (s. f.). Khan Academy. <https://es.khanacademy.org/math/algebra-home/alg-quadratics/alg-solving-quadratics-using-the-quadratic-formula/a/quadratic-formula-explained-article>
4. Villalpando, Francisco, and Andrés García Sandoval. Matemáticas Discretas. Grupo Editorial Patria, 21 Oct. 2014.
5. "Números Binarios (Artículo)." Khan Academy, [es.khanacademy.org/computing/ap-computer-science-principles/x2d2f703b37b450a3:digital-information/x2d2f703b37b450a3:binary-numbers/a/bits-and-binary](https://es.khanacademy.org/computing/ap-computer-science-principles/x2d2f703b37b450a3:digital-information/x2d2f703b37b450a3:binary-numbers/a/bits-and-binary).
6. Villalpando, Francisco, and Andrés García Sandoval.

Matemáticas Discretas. Grupo Editorial Patria, 21 Oct. 2014.

**Jacob Salas Andrea** es una estudiante de la ingeniería en Tecnologías de la Información y las Comunicaciones en el Instituto Tecnológico Superior del Occidente del Estado de Hidalgo, criada en el Mezquital, municipio de Santiago de Anaya, Hidalgo, a lo largo de su vida ha tenido diversos intereses, como la danza, el baloncesto, la música, pero el más grande es por la tecnología, ya que le sorprende lo mucho que avanza en corto tiempo. El objetivo de Andrea es completar sus estudios universitarios en ITICs, con especialización en redes, sueña con trabajar en el extranjero ejerciendo su carrera.

[Github de Andrea](#)

**Calva Abraham** es un estudiante de la ingeniería en Tecnologías de la información y Comunicaciones del Instituto Tecnológico Superior del Occidente del Estado de Hidalgo nacido en Rodhe Islan EEUU. Y criado en Mixquiahuala Hidalgo, que a lo largo del tiempo a tenido una gran pasión por el fútbol, los videojuegos y la tecnología.

[Github de Abraham](#)

**Martínez Escamilla Erick** es un estudiante de la ingeniería en Tecnologías de la Información y las Comunicaciones del Instituto Tecnológico Superior del Occidente del Estado de Hidalgo, que a lo largo del tiempo a tenido una gran pasión por el fútbol, los videojuegos, el espacio y la tecnología.

[Github de Erick](#)

**Vidal Ceron Jean Paul** Jean Paul Vidal Cerón es un estudiante de Ingeniería en Tecnologías de la Información y Comunicaciones del Instituto Tecnológico Superior del Occidente del Estado de Hidalgo. Tiene una gran pasión por los videojuegos, tanto nuevos como retro, el anime y, sobre todo, por la programación. Nació en Actopan, Hidalgo, y creció en Mixquiahuala de Juárez, Hidalgo. A lo largo del tiempo, ha desarrollado una curiosidad e interés por la informática y los avances innovadores en tecnología. El objetivo principal de Jean Paul es completar sus estudios universitarios en ingeniería de TIC con especialización en IoT. Su mayor satisfacción al estar en esta especialidad es poder crear nueva tecnología eficiente para la comunidad y contribuir en la ayuda a la sociedad.

[Github de Jean Paul](#)

**Barrera Rojo Roberto** es un estudiante de la ingeniería en Tecnologías de la Información y Comunicaciones

con una pasión por el fútbol, MMA y por los videojuegos y criado en Mixquiahuala de Juárez Hidalgo, desde temprana edad demostró un gran interés por las artes marciales mixtas y las consolas de videojuegos. Roberto se encanta por el taekwondo y videojuegos. Estos medios le han enseñado la importancia de la perseverancia y el trabajo en equipo. El objetivo principal de Roberto es completar sus estudios universitarios en ITICs y jugar en la selección de fútbol de esa misma universidad. Sueña con trabajar en el campo de full stack, encargándose de coordinar todas las acciones entre el front-end y del back-end ya sea de una web o una app

[Github de Roberto](#)