

Software de resolución de problemas de Ingeniería

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

Isai Montaña Chávez , *Instituto Tecnológico Superior del Occidente del Estado de Hidalgo, Mixquiahuala, Hgo., 42700, Mexico*

Gerardo Daniel Olivares Álvarez , *Instituto Tecnológico Superior del Occidente del Estado de Hidalgo, Mixquiahuala, Hgo., 42700, Mexico*

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

Gustavo Barrera Martínez , *Instituto Tecnológico Superior del Occidente del Estado de Hidalgo, Mixquiahuala, Hgo., 42700, Mexico*

Pedro Erik Escudero Martinez, *Instituto Tecnológico Superior del Occidente del Estado de Hidalgo, Mixquiahuala, Hgo., 42700, Mexico*

Abstract—

The present experimental study focuses on solving problems through the intentional manipulation of one or more independent variables, to analyze the consequences of such manipulation on one or more dependent variables, they are addressed through the implementation of methodologies defining, discovering, diagnosing, designing, developing and deploying for said resolution, based on differential calculus and discrete mathematics, as well as standardized tools in the field of software development, with the set of mathematical functions to be able to address different areas of IT engineering, information and communications.

The proposed problems are focused on being used in data science, electronics, networks, data storage and computer security; With this study it was possible to understand the relationship that the different branches of mathematics have with their application in the workplace.

INTRODUCCIÓN

Este documento presenta un proyecto integrador, donde los estudiantes con el compromiso y la dedicación aplican los principios de las ramas de las matemáticas y de programación en situaciones de

ámbito profesional.

A través de este proyecto se busca poder resolver los problemas planteados, identificando qué es lo que pide el problema, analizar el problema donde se divide en partes más pequeñas. Una vez se entendió qué es lo que quiere el problema, se resuelve.

La esencia de este proyecto no reside solo en las respuestas encontradas, sino también en las preguntas formuladas. Por lo tanto, este proyecto no solo busca proporcionar las soluciones, sino que también busca nuevas vías de indagación y reflexión.

Se invita al lector a sumergirse en este proyecto, donde podrá darle otro enfoque a las matemáticas, que no solo son ejercicios tras ejercicios, sino en cómo aplicar todos estos conocimientos en un ámbito más real.

COPYRIGHT Y ACCESO ABIERTO

Una vez que los autores entreguen este documento para su evaluación también seden los derechos del contenido de este manuscrito a la carrera de Ingeniería en Tecnológicas de la Información y Comunicaciones (ITICs) del Instituto Tecnológico Superior del Occidente del Estado de Hidalgo (ITSOEH). Esto conlleva que la carrera puede usar el contenido de este artículo para efectos de difusión del quehacer de los estudiantes de la carrera o en cualquier otra actividad que la carrera considere pertinente. Cabe mencionar que en ningún momento el orden o los nombres de los autores será

modificado de ninguna manera y siempre se les dará el crédito correspondiente.

PROBLEMAS

A continuación se describen los problemas que el equipo deberá resolver.

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 α que se forma entre el eje horizontal y la recta.
2. 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.
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.
4. Dado un número decimal entero positivo o negativo regresar su equivalente en binario.
5. Dado un número binario de n bits regresar su equivalente en decimal.
6. Dada una tabla de verdad de n bits generar la expresión booleana que genere de manera fidedigna las salidas de esta tabla.

SECCIONES

EJERCICIO 1:

Dados dos puntos A y B con coordenadas x_1, y_1 y x_2, y_2 respectivamente, se busca obtener la ecuación de la recta y el ángulo interno formado entre el eje horizontal y la recta.

Este problema es de suma importancia, ya que conocer el ángulo y la ecuación de una recta puede resultar útil en ciencias de datos. A continuación, se mencionan algunas aplicaciones:

1. **Análisis de tendencias:** En ciencias de datos, la ecuación de una recta se utiliza para modelar y predecir el comportamiento de los datos lineales a lo largo del tiempo. El ángulo de la recta indica la pendiente de la tendencia, revelando información sobre la dirección y la intensidad del cambio en los datos.
2. **Regresión lineal:** La regresión lineal es una técnica estadística para modelar la relación entre una variable dependiente y una o más variables independientes. Se utiliza la ecuación de una recta para ajustar los datos y encontrar la

mejor línea de ajuste. El ángulo de la recta ayuda a interpretar la relación entre las variables y proporciona información sobre la dirección y magnitud del efecto.

3. **Análisis de datos espaciales:** En el análisis de datos espaciales, la ecuación de una recta se utiliza para representar y analizar relaciones espaciales. En el análisis de datos geoespaciales, las rectas se usan para modelar los cambios en los datos según la ubicación geográfica, proporcionando información sobre la dirección y la intensidad de estos cambios.
4. **Visualización de datos:** La representación gráfica de datos es esencial en el análisis de datos. Conocer la ecuación de una recta y su ángulo permite trazar líneas de tendencia o regresión en gráficos, lo que facilita la visualización y comunicación de patrones y relaciones en los datos.

Procedemos a resolverlo sabiendo que la fórmula de la recta es:

$$y = mx + b(1)$$

donde “ y ” representa la variable dependiente, va a cambiar dependiendo de la variable x , después tenemos lo que es la pendiente que es “ m ” esta dirá en la forma que esté la pendiente si es ascendente o descendente, la pendiente estará multiplicando con la variable independiente que es x , x puede representar cualquier valor y no va a cambiar de valor como la variable “ y ”, y por último tenemos lo que el intercepto, el intercepto nos dice en qué punto del eje de las y atraviesa o toca el eje de las y , esta se representa con una “ b ”.

Sabiendo esto para poder encontrar la pendiente tenemos que usar la siguiente fórmula Figura ??.

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad (2)$$

Con esta fórmula podemos sacar la pendiente y también no ayudará a saber si la pendiente es creciente o decreciente, una vez que ya tenemos la pendiente utilizamos la siguiente fórmula para sacar b .

$$b = y_1 - (m \cdot x_1) \quad (3)$$

Con esto ya tenemos la mayor parte de la ecuación (1), en la variable x podemos poner cualquier valor.

Por último necesitamos sacar el ángulo interno, esto se saca con una fórmula

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

Esto nos ayudará para poder a poder sacar el ángulo de la recta, pero el resultado nos lo da en radianes y nosotros lo queremos en grados, para eso el resultado de la fórmula (4) tendremos que convertirlos a grados.

$$G = \frac{x \cdot 180}{\pi} \quad (5)$$

Diseño de la solución:

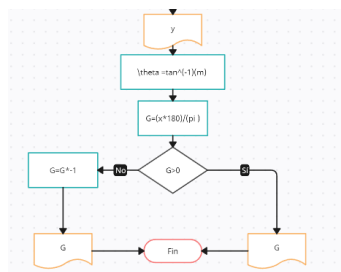
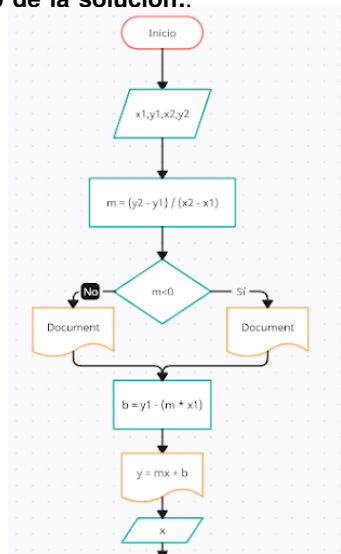


Figura 1. Nota que “Figura” Diagrama de flujo del ejercicio 1

Depuración:.

```

Working to compile - all classes are up to date

--- exec:1, frame (stack):0 of 8 frames ---
Impresora valor de n1
n1
Impresora valor de y1
y1
Impresora valor de x1
x1
Impresora valor de y1
y1
La pendiente es constante
El valor de la pendiente es: 0.43000000000000003
El valor del intercepto es: 1.3000000000000001
El siguiente error fue en la ecuación de la recta (y = m * la orden de la siguiente fila
+ la constante)
Impresora valor para a
a
Impresora la fórmula que da la siguiente muestra: 0.43000000000000003*(0.0+1)+1.3000000000000001
El resultado es: 1.7300000000000001
El segundo comando es 23.3525651249571

```

Figura 2. Nota que “Figura” podemos observar como el código funciona como debería y una forma de comprobarlo es resolver el problema y poner los puntos en un plano cartesiano.

EJERCICIO 2:

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.

Al resolver estos tipos de problemas que implican tener que resolver ecuaciones cuadráticas para determinar dónde se encuentran las raíces, si en el conjunto de números reales o en los complejos lo que resulta beneficioso en muchos campos laborales. Puesto que estas habilidades matemáticas pueden ayudarnos a abordar problemas desafiantes y tomar decisiones informadas basadas en datos y modelos matemáticos.

Descripción del problema:

Lo que nos pide es tomar los coeficientes de una ecuación cuadrática, para poder calcular las raíces de dicha ecuación y determinar si las raíces están en el conjunto de los números reales o si se encuentran dentro de los números complejos.

Diseño del problema:.

Entradas: a, b y c.

Salidas: X1, X2 Nos muestra si las raíces son números complejos, números reales distintos o si son números reales iguales.

Diseño de solución:

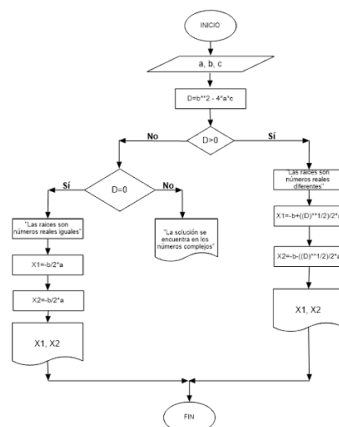


Figura 3. Nota que “Figura” Diseño de un diagrama de flujo para la resolución de un problema matemático donde se emplea la fórmula general.

Desarrollo de solución:.

En la resolución de este ejercicio, se tiene claro que una ecuación cuadrática es de la forma $ax^2+bx+c=0$. Asimismo, conociendo esto se le solicita al usuario ingresar tres valores para los coeficientes de la ecuación (a, b y c), los cuales se definen como variables de tipo “double” con el fin de obtener una

mayor precisión en los resultados. Posteriormente se guardan los valores en las variables y se calcula el discriminante, utilizando la fórmula $(b^2 - 4*a*c)$, los resultados se almacenan en una variable denominada como "D".

Luego, se utiliza una estructura de control selectiva (if) para determinar si se cumple o no la condición propuesta y así poder determinar donde se encuentran las raíces:

Si el valor de "D" es mayor que 0, esto indica que las raíces son números reales diferentes. En este caso, el programa calcula las raíces utilizando la fórmula mencionada anteriormente y las muestra en pantalla junto con el mensaje de que se encuentra dentro de los números reales distintos..

En caso de que el valor de "D" sea igual a 0, significa que las raíces son números reales iguales. Para esta situación, el programa calcula una única raíz y la muestra dos veces en pantalla junto con la descripción de que son números reales iguales.

En el último caso si "D" es menor o igual a 0, esto indica que las raíces son números complejos. Aquí el programa muestra un mensaje indicando que la solución se encuentra en el conjunto de los números complejos.

Al cumplir alguna de estas condiciones, el programa se ejecuta correctamente, validando así el código realizado.

Depuración:.

```
run:
De acuerdo a la ecuacion (b**2-4*a*c)
Dame el valor para a:
1
Dame el valor para b:
10
Dame el valor para c:
25
El resultado de X1 es: -5.0
El resultado de X2 es: -5.0
Las raices son numeros reales iguales
BUILD SUCCESSFUL (total time: 9 seconds)
```

Figura 4. Nota que "Figura" Pruebas de escritorio que son de ayuda para validar el código.

EJERCICIO 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.

Este programa en Java solicita al usuario las coordenadas del centro de una circunferencia (C) y su radio, así como las coordenadas de un punto T. Luego, determina si el punto T está dentro o fuera de la circunferencia.

Diseño de solución:.

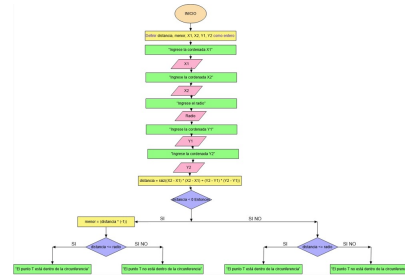


Figura 5. Nota que "Figura" El punto (T) se encuentra adentro del radio del punto (X)

Entrada de datos:.

- Se utiliza la clase Scanner para solicitar datos al usuario.
- Se pide al usuario que ingrese las coordenadas del centro de la circunferencia (x_1, y_1) , el radio de la circunferencia y las coordenadas del punto T (x_2, y_2) .

Cálculo de la distancia:.

- Se utiliza la fórmula de la distancia entre dos puntos en un plano cartesiano: $\text{distancia} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$
- La distancia se calcula utilizando las coordenadas del centro de la circunferencia (x_1, y_1) y las coordenadas del punto T (x_2, y_2) .

Condicionales para determinar si el punto T está dentro de la circunferencia:.

- Si la distancia es menor que 0, se toma el valor absoluto de la distancia (se multiplica por -1) y se almacena en la variable menor.
- Luego, se compara el menor con el radio para determinar si el punto T está dentro de la circunferencia.
- Si la distancia es mayor o igual a 0, se compara directamente con el radio.

Impresión del resultado:.

- Se imprime en consola si el punto T está dentro o fuera de la circunferencia, según las condiciones evaluadas.

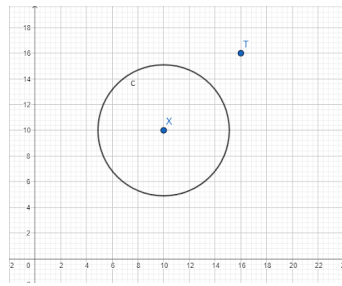


Figura 6. Nota que “Figura” El punto (T) se encuentra fuera del radio del punto (X)

- › X1=10
- › Y1=10
- › Radio=15
- › X2=16
- › Y2=16

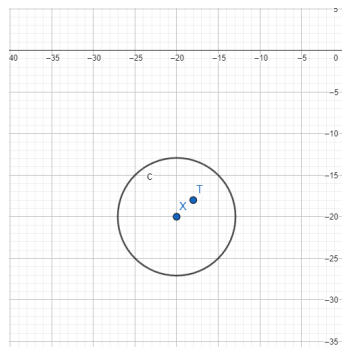


Figura 7. Nota que “Figura” El punto (T) se encuentra adentro del radio del punto (X)

- › X1=20
- › X1=20
- › Radio= 5
- › X2=-18
- › Y2=-18

EJERCICIO 4:

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

Descripción del problema:

Lo que nos pide es ingresar un numero decimal y el código deberá realizar la conversión para que nos dé como resultado en binario

Diseño del problema:

Entradas: decimal Salidas: valorinicial, bits, binario. Nos muestra el decimal ingresado como numero binario.

Diseño de solución:

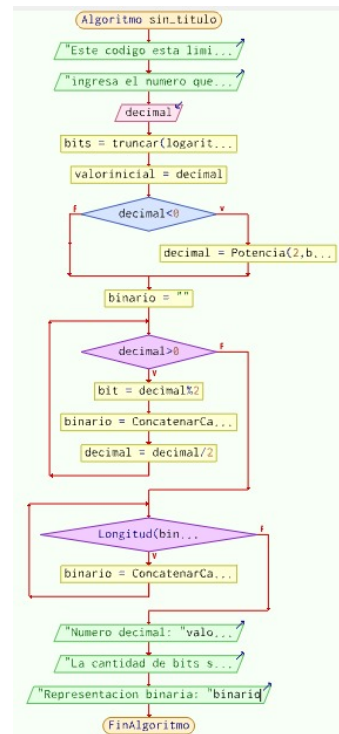


Figura 8. Nota que “Figura” Diseño de un diagrama de flujo para crear tablas de verdad.

Desarrollo de solución:

- › Solicita al usuario que ingrese un número decimal.
- › Calcula la cantidad de bits necesarios para representar ese número decimal.
- › Verifica si el número es negativo y, si es así, lo convierte a su representación en complemento a 2.
- › Convierte el número decimal en su representación binaria mediante la división repetida por 2.
- › Ajusta la longitud de la representación binaria agregando ceros al principio si es necesario.
- › Muestra en la salida el número decimal ingresado, la cantidad de bits necesarios y su representación binaria.

Depuración:

```

Ingresa el numero que deseas transformar
25
*****
RESULTADOS
*****
Número decimal: 25
la cantidad de bits son: 5
Representación binaria: 11001
  
```

Figura 9. Nota que “Figura” Prueba de que el código funciona

EJERCICIO 5:

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

Diseño del problema.:

Entradas: Ingresar el valor Binario, longitud, mod.
Salidas: i, digito, exponente y long).

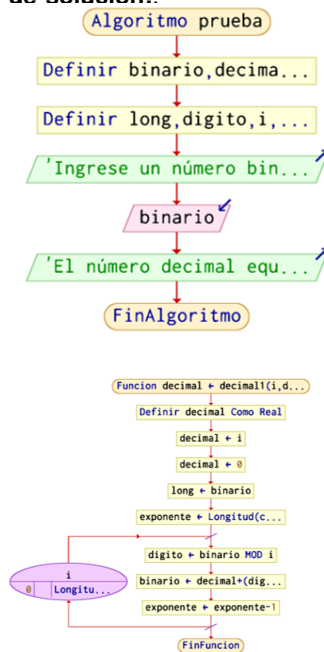
Diseño de solución.:

Figura 10. Nota que "Figura" Diseño de un diagrama de flujo para convertir numeros binarios a decimal.

Desarrollo de solución.:

El programa Integrador.^{es} un programa en Java que convierte un número binario ingresado por el usuario a su equivalente decimal.

El programa utiliza la clase "Scanner" para leer la entrada del usuario. Primero, se solicita al usuario que ingrese un número binario. El número binario se almacena en una variable de tipo "String" llamada "nb". Luego, se llama a la función convertir b a d pasando como argumento el número binario ingresado por el usuario. La función convertir b a d convierte el número binario a decimal utilizando la fórmula: $nd += digito * Math.pow(2, exponente)$;

Donde "nd" representa el número decimal resultante, "digito" es el dígito binario actual, y "exponente" es el exponente utilizado en la fórmula, que comienza con el valor de la longitud del número binario menos 1 y se disminuye en cada iteración del bucle.

Finalmente, el programa imprime el número decimal equivalente en la consola.

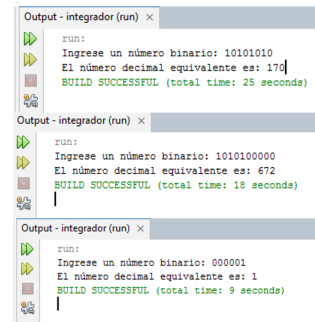
Depuración.:

Figura 11. Nota que "Figura" Prueba de que el código funciona.

EJERCICIO 6:

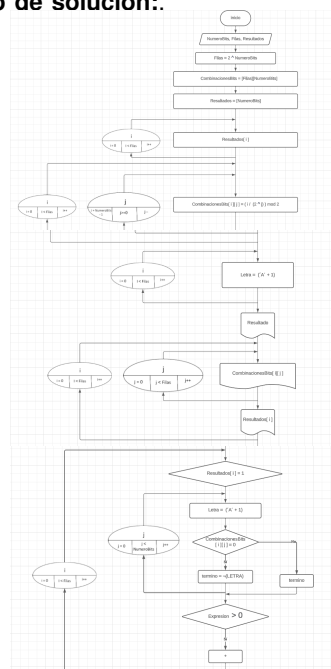
Dada una tabla de verdad de n bits generar la expresión booleana que genere de manera fidedigna las salidas de esta tabla.

Descripción del problema.:

Lo que nos pide es ingresar una cantidad de 1 a 4 bits y poder generar la tabla de verdad de dicha cantidad de bits después se deberán ingresar los resultados de la tabla de verdad y con los resultados poder generar su expresión booleana.

Diseño del problema.:

Entradas: NumeroBits, Filas, resultados. Salidas: CombinacionesBits, Expresion Nos muestra la tabla de verdad con sus resultados y su expresión booleana.

Diseño de solución.:

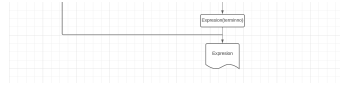


Figura 12. Nota que “Figura” Diseño de un diagrama de flujo para crear tablas de verdad.

Desarrollo de solución:

Primero, se pide al usuario que ingrese la cantidad de bits que tendrá la tabla de verdad. Esta cantidad debe estar entre 1 y 5, ya que el programa tiene limitaciones para manejar tablas más grandes debido a la gran cantidad de combinaciones de las filas.

Una vez que se ingresa la cantidad de bits, el programa calcula el número de filas de la tabla de verdad, que es 2 elevado a la cantidad de bits ingresada. Luego, crea matrices para almacenar las combinaciones de bits y los resultados correspondientes a esas combinaciones.

El usuario ingresa los resultados para cada fila de la tabla de verdad, asegurándose de que sean 0 o 1, y estas entradas se almacenan en un arreglo. Luego, se generan todas las combinaciones posibles de bits usando un enfoque de conversión de números binarios.

Una vez que se tienen todas las combinaciones de bits y sus resultados, se imprime la tabla de verdad para visualizar los datos ingresados por el usuario.

El paso crucial es la generación de la expresión booleana. Para cada fila de la tabla de verdad con un resultado verdadero (1), se construye un término booleano. Se asignan letras a los bits ('A' para el primer bit, 'B' para el segundo, etc.) y, según el valor del bit en esa fila, se construye el término booleano correspondiente: si es 0, se agrega "¬"(negación) seguido de la letra; si es 1, solo se agrega la letra.

Después de construir cada término booleano para las filas con resultado 1, se unen todos estos términos con el operador lógico OR ('+'). Se construye una expresión booleana que representa todas las condiciones requeridas para obtener los resultados de la tabla de verdad ingresada.

Finalmente, la expresión booleana se imprime para mostrarla al usuario.

Depuración:

```

Ingrese la cantidad de bits: 2
Ingrese el resultado de la fila 1: 1
Ingrese el resultado de la fila 2: 0
Ingrese el resultado de la fila 3: 1
Ingrese el resultado de la fila 4: 0

```

A	B	Resultado
0	0	1
1	0	0
0	1	1
1	1	0

Expresión booleana: $\neg A \neg B + \neg A B$

Figura 13. Nota que “Figura” Pruebas de escritorio que son de ayuda para validar el código.

CÓDIGOS

CÓDIGO 1:

```

public class Experimenta {

    public static void main(String[] args)
    {
        double x1, y1, x2, y2, m, b, x, y0
            , x3;
        Scanner va = new Scanner(System.in
            );
        String di;
        System.out.println("Ingrese valor
            de x1");
        di=va.next();
        x1= Double.parseDouble(di);

        System.out.println("Ingrese valor
            de y1");
        di=va.next();
        y1= Double.parseDouble(di);

        System.out.println("Ingrese valor
            de x2");
        di=va.next();
        x2= Double.parseDouble(di);

        System.out.println("Ingrese valor
            de y2");
        di=va.next();
        y2= Double.parseDouble(di);

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

        if (m < 0) {
            System.out.println("La
                pendiente es decreciente");
        }
        else {
            System.out.println("La
                pendiente es creciente");
        }

        System.out.println("El valor de la
            pendiente es: " + m);

        b = y1 - (m * x1);
        System.out.println("El valor del

```

```

        intercepto es: " + b);

System.out.println("Al ingresar
    estos datos en la ecuaci n de
    la recta (y = mx + b) queda
    de la siguiente forma");
System.out.println("y = " + m + "x
    + " + b);

System.out.println("Ingrese un
    valor para x");
di=va.next();
x= Double.parseDouble(di);

y0 = (m * x) + b;
System.out.println("El resultado
    es: " + y0);

x3 = Math.atan(m);
double grados = Math.toDegrees(x3)
;
if (grados > 0) {
    System.out.println("El ngulo
        interno es " + grados + "
        ");
} else {
    grados = grados*(-1);
    System.out.println("El ngulo
        interno es " + grados + "
        ");
}
}
}

```

CÓDIGO 2:

```

public class Integrador {

    public static void main(String[] args) {

        Scanner d= new Scanner(System.in);
        String valor;

        System.out.println(" De acuerdo a
            la ecuacion (b**2-4*a*c) ");
        System.out.println(" Dame el valor
            para a: ");
        valor=d.next();
        Double a= Double.parseDouble(valor
            );

        System.out.println(" Dame el valor
            para b: ");
        valor=d.next();
        Double b= Double.parseDouble(valor
            );
        System.out.println(" Dame el valor
            para c: ");
        valor=d.next();
        Double c= Double.parseDouble(valor
            );
    }
}

```

```

Double D = (float) (Math.pow(b, 2))
    -4*a*c;

if (D>0)
{
    float X1= (float) ((-b +
        Math.sqrt(D))/(2*a));
    float X2= (float) ((-b -
        Math.sqrt(D))/(2*a));
    System.out.println(" El
        resultado de X1 es: "
        + X1);
    System.out.println(" El
        resultado de X2 es: "
        + X2);
    System.out.println(" Las
        raices son numeros
        reales diferentes ");
}
else
{
    if (D==0)
    {
        float X1= (float) (-b / (2 *
            a));
        float X2=X1;
        System.out.println(" El
            resultado de X1 es: "
            + X1);
        System.out.println(" El
            resultado de X2 es: "
            + X2);
        System.out.println(" Las
            raices son numeros
            reales iguales ");
    }
    else
    {
        System.out.println(" La
            solucion se encuentra en
            el conjunto de numeros
            complejos ");
    }
}
}
}

```

CÓDIGO 3:

```

public class PRUEBA {
    public static void main(String[] args)
    {
        Scanner dato = new Scanner(System.in);
        // solicitar al usuario las
        // cordenadas el punto C (x1,y1)
        System.out.println("Este problema

```



```

        solo acepta numeros enteros y
        asta 9 cifras");
System.out.print("Ingrese la
coordenada del centro de la
circunferencia primero x1: ");
int x1 = dato.nextInt();
System.out.print("Ingrese la
coordenada del centro de la
circunferencia despues y1: ");
int y1 = dato.nextInt();
// solicitar el radio de la
circunferencia
System.out.println("Ingrese el
radio de la circunferencia: ")
;
int radio = dato.nextInt();
// solicitar al usuario las
coordenadas del punto T (x2,y2
)
System.out.print("Ingrese la
coordenada del punto T X2: ");
int x2 = dato.nextInt();
System.out.print("Ingrese la
coordenada del punto T Y2: ");
int y2 = dato.nextInt();
// C lculo de la distancia entre
el punto C y el punto T
double distancia = Math.sqrt((x2 -
x1) * (x2 - x1) + (y2 - y1) *
(y2 - y1));
// Si la distancia es menor a 0
if (distancia<0){
double menor = (distancia * (-1));
// Si la distancia es menor o
igual al radio, el punto T
est dentro de la
circunferencia
if (menor <= 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");
}
} else {
// Si la distancia es menor o
igual al radio, el punto T
est dentro de la
circunferencia
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");
}
}
}
}
}

```

CÓDIGO 4:

```

public static void main(String args[]) {
    Scanner kevin=new Scanner(
        System.in);
    System.out.println("Ingresa el numero
que deseas transformar");
    int decimal=kevin.nextInt();
    int bits = (int) (Math.log(Math.abs(
decimal)) / Math.log(2)) + 1;//
    calcular los bits del numero.
    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 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("
***** ");
    System.out. println("
RESULTADOS ");
    System.out. println("
***** ");
    System.out.println("N mero
decimal: \t" + valorinicial);
    System.out. println("la cantidad
de bits son: \t "+bits);
    System.out.println("
Representacin binaria:\t " +
    binario);
}

```

```

}

```

CÓDIGO 5:

```

public class Integrador {

    public static void main(String[] args)
    {
        Scanner erik = new Scanner(System.
            in);
        System.out.print("Ingrese un
            n mero binario: ");
        String nb = erik.nextLine();

        int nd = convertirbad(nb);
        System.out.println("El n mero
            decimal equivalente es: " + nd
        );
    }

    public static int convertirbad(String
        nb) {
        int nd = 0;
        int longitud = nb.length();
        int exponente = longitud - 1;

        for (int i = 0; i < longitud; i++)
        {
            int digito = Character.
                getNumericValue(nb.charAt(
                    i));
            nd += digito * Math.pow(2,
                exponente);
            exponente--;
        }

        return nd;
    }
}

```

CÓDIGO 6:

```

public class Ejercicio_6 {
    public static void main(String[] args)
    {
        Scanner obj = new Scanner(System.in);
        //Llamamos al objeto Scanner.
        StringBuilder Expresion = new
            StringBuilder(); //Se utiliza para
            construir y manipular cadenas de
            caracteres de manera eficiente.

        System.out.print("Ingrese la
            cantidad de bits: ");
        int NumeroBits = obj.nextInt(); //
            Se ingresa la cantidad de Bits
            que tendra la tabla.
    }
}

```

```

while (NumeroBits < 1 ||
    NumeroBits > 5) { //Si el
    numero ingresado es menor a 0
    o mayor a 6 bits deberea
    ingresar nevemente la cantidad
    .
        System.out.println("El
            n mero ingresado no
            es 0 ni 1. Ingrese
            nuevamente:");
        NumeroBits = obj.nextInt()
        ;
    }

    int Filas = (int) Math.pow(2,
        NumeroBits); // Calcular el
        n mero de filas de la tabla
        de verdad.

    int[][] combinacionesBits = new
        int[Filas][NumeroBits]; //
        Crear arreglos para almacenar
        las combinaciones de bits.
    int[] resultados = new int[Filas];
        // Crear arreglos para
        almacenar los resultados.

    for (int i = 0; i < Filas; i++) {
        // Solicitar al usuario
        ingresar los valores de
        resultado para cada
        combinaci n de bits.
        System.out.print("Ingrese el
            resultado de la fila " + (
                i + 1) + ": ");
        resultados[i] = obj.nextInt();

        while (resultados[i] != 0 &&
            resultados[i] != 1) { //Si
            el numero ingresado no es
            0 o 1 se debera volver a
            ingresar el valor.
            System.out.println("El
                n mero ingresado no
                es 0 ni 1. Ingrese
                nuevamente:");
            resultados[i] = obj.
                nextInt();
        }
    }

    for (int i = 0; i < Filas; i++) {
        // Generar todas las
        combinaciones de bits.
        for (int j = NumeroBits - 1; j
            >= 0; j--) {
            combinacionesBits[i][j] =
                (i / (int) Math.pow(2,
                    j)) % 2; //Formula
                para generar las
                combinaciones
        }
    }
}

```

```

for (int i = 0; i < NumeroBits; i
++) { // Imprimir encabezados
de las columnas usando letras
de la A a la Z.
char letra = (char) ('A' + i);
System.out.print(letra + "\t")
;
}
System.out.println("Resultado");

for (int i = 0; i < Filas; i++) {
// Imprime la tabla de verdad.
for (int j = 0; j < NumeroBits
; j++) {
System.out.print(
combinacionesBits[i][j
] + "\t");
}
System.out.println(resultados[
i]); //Imprime los
resultados.
}

for (int i = 0; i < Filas; i++) {
// Recorre las filas de la
tabla de verdad
if (resultados[i] == 1) { //
Verificar si el resultado
de la fila actual es igual
a 1
StringBuilder termino = new
StringBuilder();// Crear
un StringBuilder para
construir el t rmino
correspondiente a la fila
for (int j = 0; j < NumeroBits; j
++) { // Recorre los bits de
la fila
char letra = (char) ('A' + j);
// Obtener la letra
correspondiente al bit
actual
if (combinacionesBits[i][j] ==
0) { // Verificar si el
bit actual es igual a 0
termino
.append(" ").append(letra
); // Si el bit es 0
agregar el smbolo de
negaci n " " seguido de
la letra al t rmino
} else {
termino.append(letra);
}
}
if (Expresion.length() > 0) {
// Verificar si ya hay
t rminos en la expresi n
booleana
Expresion.append(" + "); //
Agregar el operador de
suma "+" antes del nuevo

```

```

t rmino
}
Expresion.append(termino);
// Agregar el t rmino
actual a la
expresi n booleana
}
}
System.out.println("Expresi n
booleana: " + Expresion.
toString()); // Imprimir la
expresi n booleana resultante
}
}
}

```

CONCLUSION

En este proyecto integrador ha sido una experiencia enriquecedora y gratificante. Nuestros objetivos principales fueron como usamos las ramas de las matemáticas para poder resolver los ejercicios propuestos, y saber donde podemos usarlos en un ámbito más real. Durante el proyecto, nos enfrentamos a varios desafíos, desde el uso de GitHub, la programación de los ejercicios y problemas con la PC. Sin embargo, gracias a un buen trabajo en equipo, pudimos superar los obstáculos y poder solucionarlos. La aplicación de estos ejercicios lo podríamos utilizar en un ámbito profesional, como en ciencia de datos, redes y seguridad, etc.

REFERENCES

1. RODRÍGUEZ-CASTRO, R. A., RESÉNDIZ-BALDERAS, E. V. E. L. I. A. (2021). CONSTRUCCIÓN DE LA GEOMETRÍA ANALÍTICA EN SITUACIONES REALES: DISTANCIA ENTRE DOS PUNTOS. Revista Internacional de Ciencias Sociales y Humanidades SOCIOTAM, 31(1), 91-112.
2. Zuñiga, L. (s. f.). El cálculo en carreteras de ingeniería: un estudio cognitivo. <https://www.scielo.org.mx/scielo.php?pid=S1665-24362007000100007script=sciarttext>
3. Formula general Analisis del discriminante. (s. f.). PPT. <https://es.slideshare.net/UnozxcvDoszxc/formula-general-analisis-del-discriminante>
4. 7.9: Convertir tablas de verdad en expresiones booleanas. (2022, octubre 31). LibreTexts Español; Libretexts.

Jacqueline Hernandez Santiago es una estudiante de primer semestre de la ingeniería en

Tecnologías de la Información y comunicaciones. Nacida en la comunidad de Tezontepec, con gusto por la música en inglés y las series de ciencia ficción. Actualmente le llama la atención el hacking y en un futuro se ve como una gran programadora.

Isai Montaña Chávez es un estudiante universitario, estudia en el Instituto Tecnológico Superior del Occidente del Estado de Hidalgo nacido en Tlaxcoapan y criado en Tlamanca de Atitalaquia. Mi vida ha estado dividida entre mi pasión por el fútbol y la tecnología. Comencé a interesarme por la programación a los 15 años, atraído por los videojuegos y las tecnologías. Estudio Ingeniería en Tecnologías de la Información y Comunicaciones.

Gerardo Daniel Olivares Álvarez es un estudiante universitario, estudia en el Instituto Tecnológico Superior del Occidente del Estado de Hidalgo en la carrera de Tics, le gusta mucho jugar videojuegos y la música. Nació en Cuautla Morelos, desde los 12 mostró interés por la tecnología y la ciencia. Estudió en la UPP en la carrera de fisioterapia pero al no cumplir con las expectativas decidió salir de la escuela. A futuro le gustaría trabajar en una empresa y poder comprarse una consola de última generación.

Kevin Cruz Cruz Nació el cinco de Noviembre del dos mil cinco, en Mixquiahuala Hidalgo. En una familia de acceso medio, el joven se a de interesar principalmente por la música desde pequeño con canciones que escuchaba su abuela estas como: boleros, música ranchera y flamencos. Esto desencadenó que buscara más acerca de música y encontrar mas variedades, descubriendo la música en los videojuegos y así llamándole la atención la tecnología y el como ingresaban la música en los videojuegos, tubo acceso al internet en casa. Y pasar del gusto a la música a interesarle la computación y limpieza de los aparatos, dándose cuenta que puede hacer dos cosas que le gustan; escuchar música y limpiar aparatos electrónicos, eso al mismo tiempo. Hoy en día es un estudiante del Instituto Tecnológico Superior del Occidente del Estado de Hidalgo, en la carrera de Tecnologías de la Información y Comunicaciones, y aun que le cuesta ser responsable lo intenta y espera que

en el futuro sea un buen ingeniero en el área de desarrollo de videojuegos o internet de las cosas, todavía no lo tiene bien definido.

Gustavo Barrera Martínez es un estudiante universitario, estudia en el Instituto Tecnológico Superior del Occidente del Estado de Hidalgo en la carrera de Tics, le gusta mucho la música y salir a conocer diferentes partes de México. Desde niño su sueño a sido viajar a diferentes partes del mundo, se dice que es un chico muy alegre y carismático y que siempre tiene un gran cariño por las personas que lo rodean ademas de los animales.