

Software de resolución de problemas de Ingeniería

Diego Antonio Badillo Morales, José Manuel Cortes Cerón, Oswaldo Gabriel Villaverde Mendoza, Carmen Anahi Cornejo Lopez, Sebastian Hernandez Angeles, Instituto Tecnológico Superior del Occidente del Estado de Hidalgo, Mixquiahuala, Hgo., 42700, Mexico

Abstract—The project focuses on the resolution of 6 mathematical problems associated with the subjects of discrete mathematics and differential calculus of the first semester of the Information and Communication Technologies Engineering course. The objective of this project is to give a solution to each of the problems in an optimal way, using the knowledge of the subjects learned during the course. This file is available at <https://github.com/Joseecodm/ProyectoIntegrador.git>.

Introducción

El manuscrito presenta la resolución de seis problemas matemáticos y lógicos propuestos en este proyecto de las asignaturas de Matemáticas discretas y cálculo diferencial.

El objetivo principal de este proyecto es demostrar la comprensión de los conceptos y herramientas aprendidas durante ambas asignaturas, aplicando de manera eficiente para dar solución a los problemas planteados, también se busca obtener la solución correcta así como justificar detalladamente cada paso del proceso seguido, para conseguir estos objetivos se usó la metodología de las 6D la cual consiste en descripción de problema, definición de la solución, diseño de la solución, desarrollo de la solución, depuración y pruebas, documentación, como resultado, se lograron resolver los problemas de la manera más óptima, demostrando con ello la comprensión de conceptos clave tan relevantes para el desempeño académico y profesional como ingenieros en tecnologías de la información y la comunicación.

Problema 1

1. Descripción del problema

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

2. Definición del problema:

Los datos del problema son los siguientes:

Los valores de las coordenadas x_1 y y_1 de uno de los puntos de la recta.

Los valores de las coordenadas x_2 y y_2 del otro punto de la recta.

El deseo del problema es obtener los siguientes resultados:

La ecuación de la recta, en la forma $y = m \cdot x + b$.

El ángulo con el eje horizontal, en grados.

3. Diseño de la solución:

De acuerdo a la Figura 1

La solución se puede dividir en los siguientes pasos:

- Calcular la pendiente de la recta, m .

- Calcular el ángulo de la recta, a , utilizando la fórmula $a = \text{atan}(m)$.

- Convertir el ángulo a grados, $a \text{ Grados} = a \cdot 180 / \pi$.

- Obtener la ecuación de la recta, $y = m \cdot x + b$, donde $b = y_1 - m \cdot x_1$.

4. Desarrollo de la solución:

El código comienza con las importaciones necesarias y algunos comentarios que proporcionan información sobre la clase (figura 2).

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

Después para la ecuación de la recta para (y) se utiliza la fórmula

Figura 1. Diagrama de flujo problema uno

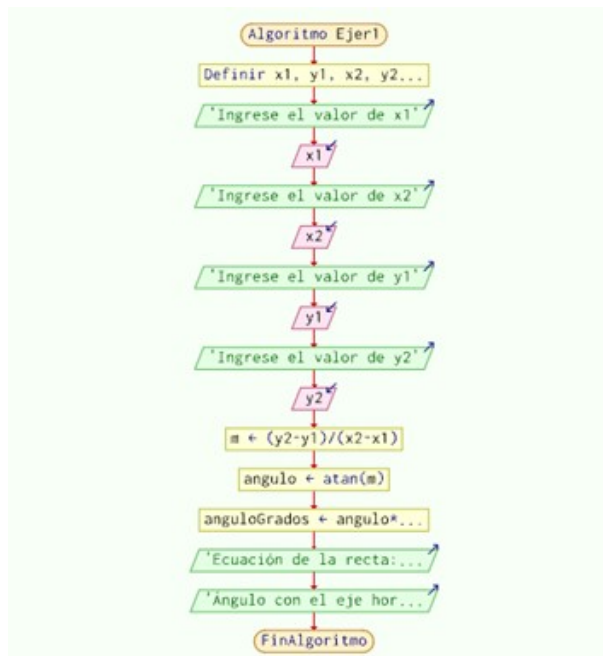


Figura 2. Código Problema Uno

```

public static void main(String[] args) {
    Scanner hola=new Scanner(System.in);
    double x1, y1,x2,y2;
    System.out.println("Ingrese el valor de x1");
    x1=hola.nextFloat();
    System.out.println("Ingrese el valor de x2");
    x2=hola.nextFloat();
    System.out.println("Ingrese el valor de y1");
    y1=hola.nextFloat();
    System.out.println("Ingrese el valor de y2");
    y2=hola.nextFloat();
    double m = (y2 - y1) / (x2 - x1);
    double angulo = Math.atan(m);
    double anguloGrados = Math.toDegrees(angulo);
    System.out.println("Ecuación de la recta: y = " + m + " * x + " + (y1 - m * x1));
    System.out.println("Angulo con el eje horizontal (en grados): " + anguloGrados);
}

```

$$(y2 - y1)/(x2 - x1) \quad (1)$$

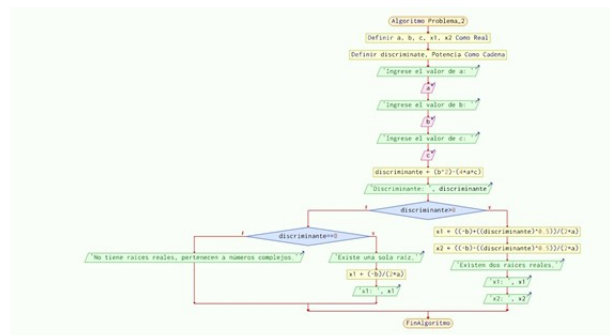
y se almacena en la variable m.
Después para (x) se utiliza la siguiente operación

$$(y1 - m * x1) \quad (2)$$

Y para sacar el ángulo en grados se utiliza la función Math.atan(m) para calcular el ángulo en radianes y después se usa Math.toDegrees() para pasar de radianes a grados.

Y finaliza dándote el resultado de la Ecuación

Figura 3. Diagrama Problema Dos



de la recta y el Ángulo con el eje horizontal.

5. Depuración y pruebas

Prueba basica:

(x1, y1) = (0, 0)

(x2, y2) = (1, 1)

Ingrese el valor de x1

Ingrese el valor de x2

Ingrese el valor de y1

Ingrese el valor de y2

Ecuación de la recta: y = 1.0 x + 0.0 Ángulo con el eje horizontal (en grados): 45.0

Problema 2

1. Descripción del problema: Dada la ecuación que ingrese se regresara los valores de las raíces en caso de que estén sobre el conjunto de los números reales, en el caso de que no indicara la solución esta en el conjunto de los números complejos.

2. Definición del problema:

Se solicita al usuario que ingrese a, b y c.

Se calcula el discriminante de la ecuación cuadrática, dado por la formula b^2-4ac .

Te arrojará como resultado: los valores de (x1, x2) y escribirá si las raíces son reales, idéntica o si no tiene.

3. Diseño de la solución:

La solución del problema se basa en los siguientes pasos (figura 3): 1. Ingresar los datos de a, b y c. 2. Modificar los datos de la ecuación. 3. Dado el resultado se darán dos

Figura 4. Código Problema Dos

```

package com.mycountry.integrador;
import java.util.Scanner;
/**
 *
 * @author Carmen Anahi Cornejo
 */
public class Integrador {
    public static void main(String[] args) {
        double a, b, c, x1, x2;
        Scanner obj1 = new Scanner(System.in);
        System.out.println("teclea el valor de a:");
        a = obj1.nextFloat();
        System.out.println("teclea el valor de b:");
        b = obj1.nextFloat();
        System.out.println("teclea el valor de c:");
        c = obj1.nextFloat();
        double discriminante = (Math.pow(b, 2)) - (4 * a * c);
        System.out.println("Discriminante: " + discriminante);
        if (discriminante > 0) {
            x1 = ((-b) + (Math.sqrt(discriminante))) / (2 * a);
            x2 = ((-b) - (Math.sqrt(discriminante))) / (2 * a);
            System.out.println("Existen dos raíces reales.");
            System.out.println("x1: " + x1);
            System.out.println("x2: " + x2);
        } else if (discriminante == 0) {
            System.out.println("Existe una sola raíz.");
            x1 = (-b) / (2 * a);
            System.out.println("x1: " + x1);
        } else {
            System.out.println("No tiene raíces reales, pertenecen a números complejos.");
        }
    }
}

```

respuestas que serán x1 y x2. 4. Y te arrojará un mensaje que diga si hay dos raíces, solo una o ninguna.

4. Desarrollo de la solución:

Antes que nada, tenemos que tener claro que la ecuación cuadrática siempre será (figura 4):

$$ax^2 + bx + c = 0 \quad (3)$$

siempre que la ecuación esta ordenada podemos ver cuál es el valor de a, b y c.

El código comienza import Scanner y algunos comentarios que nos proporcionan información sobre la clase.

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

Se crea un objeto como Scanner para poder leer a, b y c.

Al ya tener ingresado los valores de a, b y c los reemplazaremos en la ecuación

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

que es de tipo double.

Para la siguiente parte tuvimos que tener claro que:

-Si el discriminante > cero, hay dos raíces reales distintas.

-Si el discriminante = cero, existe una sola raíz real idéntica.

-Si el discriminante < cero, no tiene raíces reales, pertenecen a números complejos.

Después de eso utilice un if (discriminante > 0) para poder sacar los dos resultados los cuales son x1 que es positivo y x2 que es negativo.

Figura 5. Prueba de escritorio Problema Dos

```

--- exec:3.1.0:exec (default-cli) @ Integrador ---
teclea el valor de a:
3
teclea el valor de b:
-9
teclea el valor de c:
10
Discriminante: -39.0
No tiene raíces reales, pertenecen a números complejos.

```

Seguido con un else if (discriminante==0) que ejecutara la ecuación:

$$x1 = (-b)/(2 * a) \quad (5)$$

Finalizamos con un else que solo ejecutara un texto diciendo "No tiene raíces reales, pertenece a números completos"

5. Depurar:

Se comprobó si el código se ejecuta correctamente y para eso se realizó pruebas de escritorio (figura 5).

Problema 3

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

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

3. Diseñar de solución (figura 6):

4. Desarrollar (figura 7):

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

Se solicita que ingrese el radio del círculo.

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

Se solicita que ingrese las coordenadas del punto a buscar.

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

Se realiza una operación matemática en donde se restan las coordenadas y se elevan al cuadrado, después se suman ambos resultados y sacas raíz cuadrado.

Figura 6. Diagrama Problema Tres

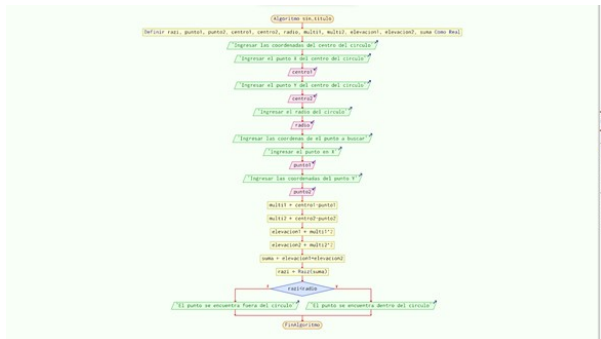


Figura 7. Código Problema Tres

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

    float puntal, puntal2, centro1, centro2, radio;
    float multil, multil2, elevacion1, elevacion2, suma;

    System.out.println("Ingresar las coordenadas del centro del círculo");
    System.out.println("Ingresar el punto en X del centro del círculo");
    centro1=obj.nextFloat();
    System.out.println("Ingresar el punto Y del centro del círculo");
    centro2=obj.nextFloat();
    System.out.println("Ingresar el radio del círculo");
    radio=obj.nextFloat();
    System.out.println("Ingresar las coordenadas de el punto a buscar");
    System.out.println("Ingresar el punto en X");
    puntal=obj.nextFloat();
    System.out.println("Ingresar el punto en Y");
    puntal2=obj.nextFloat();

    multil = centro1 - puntal;
    multil2 = centro2 - puntal2;

    elevacion1 = (float) Math.pow(multil, 2);
    elevacion2 = (float) Math.pow(multil2, 2);

    suma = elevacion1 + elevacion2;

    double radio = (double) Math.sqrt(suma);

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

Usamos un if para saber si el punto esta fuera o dentro del círculo, si la distancia es menor al radio el punto se encontrara dentro del círculo, de lo contrario el punto se encontraría fuera del círculo.

5. Depurar (figura 8):

Se comprobó si el código se ejecuta correctamente y para eso se realizó pruebas de escritorio.

Problema 4

1. Descripción del problema El problema resuelto con este código es la conversión de un número

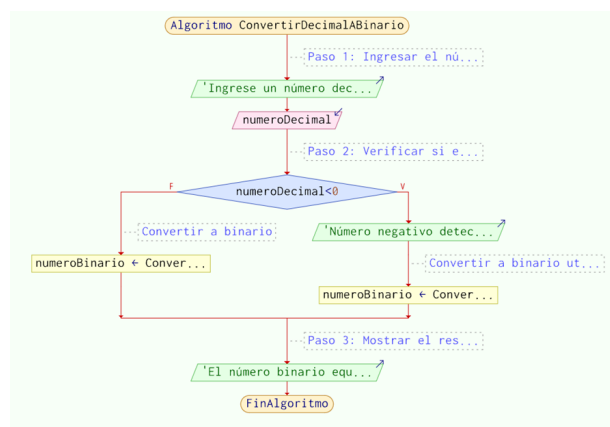
Figura 8. Prueba de escritorio Problema Tres

```

Ingresar las coordenadas del centro del círculo
Ingresar el punto en X del centro del círculo
2
Ingresar el punto Y del centro del círculo
3
Ingresar el radio del círculo
4
Ingresar las coordenadas de el punto a buscar
Ingresar el punto en X
3
Ingresar el punto en Y
2
El punto se encuentra dentro del círculo

```

Figura 9. Diagrama de flujo Problema Cuatro



decimal entero a binario. El número binario se representa como una cadena de bits, donde cada bit tiene un valor de 0 o 1.

2. Definición del problema

El tipo de datos del número decimal. En este caso, el número decimal es un número entero. El rango de valores del número decimal. En este caso, el número decimal puede tomar cualquier valor entero, desde -2 elevado a 32 hasta 2 elevado a 32 - 1.

El formato del número binario. En este caso, el número binario se representa como una cadena de bits, donde cada bit tiene un valor de 0 o 1.

3. Diseño de la solución (figura 9):

La solución al problema se basa en los siguientes pasos: 1. Comprobar si el número es positivo o negativo. 2. Convertir el número decimal a binario. 3. Si el número es negativo, aplicar el método de complemento a2.

Figura 10. Código Problema Cuatro

```

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

```

4. Desarrollo de la solución(figura 10):

1. Comprobar si el número es positivo o negativo. Este paso es sencillo. Se puede utilizar el operador de comparación \geq para comprobar si el número es mayor o igual que cero.

2. Convertir el número decimal a binario, si es positivo o negativo. Si el número es positivo, se puede utilizar el método `toBinaryString()` de la clase `Integer` para convertir el número decimal a binario.

Si el número es negativo, se aplica el método de complemento a2.

El método de complemento a2 consiste en los siguientes pasos:

Invertir el orden de los bits del binario del número negativo. Añadir un 1 al principio.

Ejemplo: El número decimal -1 se convierte a binario en 11111111111111111111111111111110 (en complemento a2).

5. Depuración y pruebas

Número decimal: 10

Número binario: 1010

Número decimal: -1

Número binario :

11111111111111111111111111111110

Número decimal: 0

Número binario: 0

Problema 5

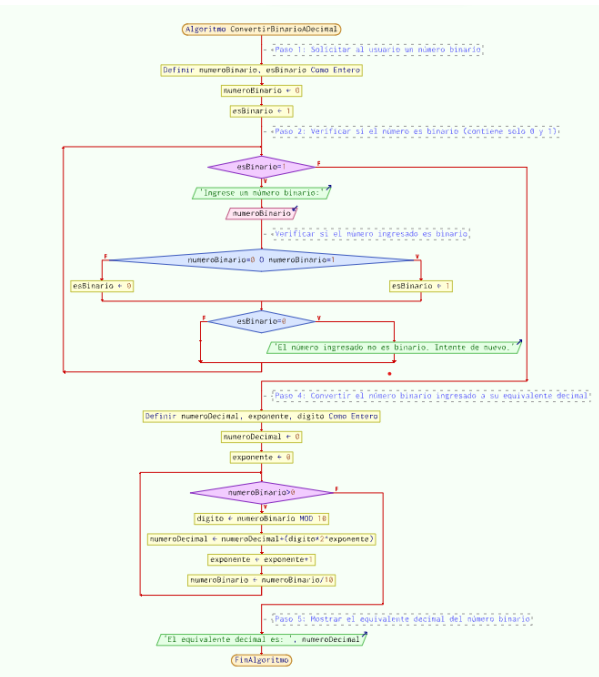
1. Descripción del problema:

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

2. Definición del problema:

- Problema: Convertir un número binario ingre-

Figura 11. Diagrama de flujo Problema Cinco



sado por el usuario a su equivalente decimal.

- Entradas: El número binario ingresado por el usuario.

- Salidas: El equivalente decimal del número binario.

3. Diseño de solución:

(figura 11)

Solicitar al usuario un número binario. Verificar si el número es binario (contiene solo 0 y 1). Si no es binario, solicitar nuevamente el número binario. Convertir el número binario ingresado a su equivalente decimal. Mostrar el equivalente decimal del número binario.

4. Desarrollar (figura 12):

El código comienza con las importaciones necesarias y algunos comentarios que proporcionan información sobre la clase.

Función 'main':

- La función `main` comienza declarando una variable `numeroBin` para almacenar el número binario ingresado por el usuario.
- Se crea un objeto `Scanner` (`binario`) para leer la entrada del usuario.

- Se usa un bucle do-while para pedir al usuario que ingrese un número binario. Continúa solicitando la entrada hasta que se ingrese un número binario válido (solo 0 y 1).

- Después de obtener un número binario válido, llama a la función binarioADecimal para convertirlo a su equivalente decimal y muestra el resultado.

Función 'esBinario':

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

Función 'binarioADecimal':

- Esta función convierte un número binario dado en un número decimal.

- Itera sobre cada dígito del número binario, convirtiendo cada dígito a su equivalente decimal y sumándolo al resultado final.

- Usa la fórmula de conversión de binario a decimal:

$$\text{sumade}(d_{\text{dígito}} * 2$$

$\text{posicion}) (6)$

, donde la posición comienza desde 0 y se incrementa hacia la izquierda en el número binario.

5. Depurar:

- Verifiqué si hay errores, comprobé si el código se ejecuta correctamente y realicé pruebas de escritorio para asegurarme de que la conversión binario-decimal funcionara correctamente.

Problema 6

1. Definición del problema:

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

2. Diseño:

- Entrada de datos: Se requiere la tabla de verdad de n bits que contiene las combinaciones de entrada y sus salidas correspondientes.
- Análisis de datos: Identifique las combinaciones

Figura 12. Código Problema Cinco

./latex-imagenes/codigoprob5.png

de entrada que producen una salida de 1 y se utiliza esta información para construir la expresión booleana.

Algoritmo:

1. Inicio

2. Ingresar la cantidad de bits (numBits):

- Se lee la cantidad de bits que se utilizarán en la tabla de verdad.

3. Inicializar el arreglo de letras (letras):

- Se define un conjunto de letras (A, B, C, D) para representar los bits.

4. Calcular el total de combinaciones posibles (totalCombinaciones):

- Se calcula el número total de combinaciones posibles de bits usando la fórmula

$$2^{\text{numBits}} \quad (7)$$

5. Mostrar la tabla de verdad:

- Se muestran las letras que representan los bits y se crea la estructura de la tabla.

6. Generar expresiones booleanas para las filas con valor de salida 1:

- Para cada combinación de bits posible:

- Se recorren los bits y se genera la expresión booleana correspondiente.

- Si el bit es 0, se muestra la negación de la letra; si es 1, se muestra la letra sola.

3. Desarrollo:

- Identificar combinaciones con salida 1: Se revisan todas las combinaciones de la tabla de

Figura 13. Código Uno Problema Seis

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

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

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

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

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

Figura 14. Código Dos Problema Seis

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

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

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

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

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

verdad y se registran aquellas que tengan una salida de 1.

- Crear la expresión booleana: Se utilizan las combinaciones identificadas con salida 1 para generar la expresión booleana.

Configuración inicial y solicitud de datos(figura 13):

- Importa las clases necesarias (Scanner, ArrayList, List).
- Define un conjunto de letras para representar los bits.
- Crea un objeto Scanner para la entrada de datos.
- Pide al usuario que ingrese la cantidad de bits.

Generación de la tabla de verdad y solicitud de valores de salida(figura 14):

- Calcula el número total de combinaciones posibles de bits.
- Muestra las letras que representan los bits y una línea separadora.
- Muestra todas las combinaciones posibles de

Figura 15. Código Tres Problema Seis

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

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

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

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

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

Figura 16. Código Cuatro Problema Seis

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

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

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

} else {
    System.out.println("\nNo hay filas con valor 1 para generar expresiones booleanas.");
}
```

bits.

- Solicita valores de salida para cada combinación de bits.

Procesamiento de valores de salida y generación de expresiones booleanas (figura 15):

- Solicita valores de salida para cada combinación de bits.

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

- Si hay filas con valor de salida 1, se procede a mostrar las expresiones booleanas correspondientes a estas filas. En caso contrario, muestra un mensaje indicando que no hay filas con valor 1.

Generación de expresiones booleanas para filas con valor de salida 1(figura ??):

Explicación:

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

de bits con sus valores de salida.

- Genera las expresiones booleanas correspondientes a las filas que tienen un valor de salida igual a 1. Las expresiones se construyen usando las letras que representan los bits y, si es necesario, la negación de estas letras.

4. Depuración:

- Verificación de la lógica: Se revisa que la expresión generada por la tabla de verdad produzca los resultados esperados.
- Manejo de errores: Se valida que la expresión generada sea coherente y represente correctamente la tabla de verdad.

5. Pruebas: • Validación de la expresión: Se prueba la expresión generada utilizando la tabla de verdad original para confirmar que reproduce los resultados correctamente.

Agradecimientos

Cada idea, cada contribución y cada hora de trabajo dedicada ha desempeñado un papel fundamental en el éxito de este proyecto. Vuestra dedicación y compromiso han sido pilares esenciales para alcanzar nuestros objetivos. Reconozco y valoro profundamente el arduo trabajo y la colaboración ejemplar que cada uno de nosotros demostró.

Referencias

Carmen Anahi Cornejo López es una estudiante del ITSOEH, en el cual cursa su primer año en la carrera de TIC's, nacida en Mixquiahuala de Juárez Hidalgo, actualmente tiene 18 años, tiene un gusto hacia el deporte y hacer ejercicio, al igual tiene un gusto hacia las redes, su principal objetivo es poder concluir su carrera y poder ejercerla en la área que más le gusta.

Diego Antonio Badillo Morales es un estudiante de TICs en la universidad ITSOEH, nacido en Actopan, Hidalgo y con 18 años de edad, tiene una fascinación por aprender cosas nuevas y generar experiencias memorables a donde sea que vaya, su objetivo es comprender y desarrollar tecnologías nuevas mientras

analiza las actuales, y mientras trata de completar su objetivo actual está abierto a nuevas posibilidades, mientras más atractivas sean mejor.

José Manuel Cortes Cerón, soy estudiante de Ing. En Tecnologías de la Información y Comunicaciones en el Instituto Tecnológico Superior del Occidente del Estado de Hidalgo, nacido en Tula de Allende, Hidalgo, actualmente tengo 19 años de edad, con una fascinación muy grande por aprender y dar el conocimiento que obtiene. Mi objetivo en la vida es desarrollar nuevas tecnologías, así mismo, concretar un nuevo modelo para enseñar a aprender a siguientes generaciones.

Sebastian Hernández Ángeles es un estudiante de Ingeniería en Tecnologías de la Información y Comunicaciones con gustos variados como los videojuegos y el voleibol. Nacido y criado en Tlaminulpa, Hidalgo. Aunque inicialmente no se interesaba por la tecnología, fue durante la preparatoria cuando descubrió su facilidad para las ciencias exactas y decidió estudiar algo relacionado con ellas. Si bien sus intereses personales no están directamente vinculados a la tecnología, éstos le han enseñado valiosas lecciones como el trabajo en equipo, la perseverancia ante los retos y el esfuerzo constante para alcanzar metas. Actualmente cursa la carrera de Ingeniería en Tecnologías de la Información y Comunicaciones en el ITSOEH, donde busca desarrollar su potencial aplicando sus habilidades matemáticas y científicas a esta disciplina. Su principal objetivo es terminar con éxito sus estudios universitarios para contribuir, desde su área, al avance tecnológico de México.

Oswaldo Gabriel Villaverde Mendoza estudiante de la institución ITSOEH cursando el primer semestre de la ingeniería de TIC's (Tecnologías de la Información y Comunicaciones), nacido en Pachuca y criado en Actopan Hidalgo, actualmente contando con 18 años, con gustos como el anime, videojuegos, la música y también las carreras de carros como por ejemplo el drifting y como objetivo tiene encontrar algo que al fin le guste para estudiar y trabajar en ella.