

Prueba de Caja Blanca

***“LECTODE PSICOPATOLOGÍAS
(DEPRESIÓN)”***

REQUISITO 004 Versión 1.0

Integrantes:

**Joel Arguello
Cesar Loor
Josué Villavicencio**

Fecha 2022-01-18

1. REQUISITO PLANTEADO

Historia de Usuario	
Número: REQ 004	Usuario: C.I
Nombre de la Historia: Ejecución del programa	
Prioridad: Alta	
Programador Responsable: Joel Arguello	
Descripción: <ul style="list-style-type: none">Se realiza el examen y se crea la ficha del mismo	
Validación: <ul style="list-style-type: none">Si se ejecutan las tareas que se les pide se puede retornar al menú de inicio.	

2. CÓDIGO FUENTE

2.1. Código C

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include <string.h>
#include <windows.h>

void exm(void)
{
    FILE *Fp;
    FILE *Fd;
    char fitxizen[30],palabra[60],pala2[30],texto[80],
    delimitador[] = ",";
    char *token = strtok(palabra, delimitador);
    int i,tmp1,tmp2,konta=0;
    int opc;
    do{
        system("cls");
        printf(CI"\t\t\t=====
\n");
        printf("\t\t\t##
\t\t\t##"ANSI_COLOR_RESET"                ##\n");
        printf("\t\t\t##"ANSI_COLOR_RESET"                EXAMEN DDEDPS                "CI"##\n");
        printf("\t\t\t##"ANSI_COLOR_RESET"                [1]Para comenzar                "CI"##\n");
        printf("\t\t\t##"ANSI_COLOR_RESET"                [2]Volver                "CI"##\n");
        printf("\t\t\t##"ANSI_COLOR_RESET"                "CI"##\n");
        printf("\t\t\t=====
ANSI_COLOR_RESET);
        printf("\t\t\tDigite opcion a escoger: ");
        fflush( stdin );
        scanf("%d",&opc);
        switch(opc){
            case 1:
                system("cls");
                exmla();
                break;
            case 2:
                system("cls");
                menu();
                break;
            default:
                printf("\nOpción incorrecta");
                getch();
                break;}
    } while(opc!=2);
}

void exmla(void){
    fflush( stdin );
    system("cls");
    FILE* Fp;
    FILE* Fd;
    int i,tmp1,tmp2,prom=0, cal, n2;
    char cadena[500],
    delimitador[] = " ",texto[50000];
```

```

Fp=fopen("c:/DDEPS/asd.txt", "r");
fgets(cadena,500,Fp);
char *token = strtok(cadena, delimitador);
Fd=fopen("C:/DDEPS/DDEPSa.txt","r");
if (Fd==NULL){
    printf("Error abriendo el fichero");
    if(token != NULL)
        while(token != NULL){

            while (feof(Fd)==0)
            {
                fgets(texto,80,Fd);
                for(i=0;i<strlen(texto);i++)
                {

                    if (token[0]==texto[i])
                    {
                        tmp1=0;
                        tmp2=i;
                        while ((token[tmp1]==texto[tmp2])&&(tmp2<strlen(texto))&&(tmp1!=strlen(token)))
                        {
                            tmp1++;
                            tmp2++;
                            if (tmp1==strlen(token));
                        }
                    }
                }
                prom++;
                token = strtok(NULL, delimitador);
            }
            printf("%d",prom);
            cal = (5 * prom)/15;
            printf("\nEl examen realizado consta de un nivel de probabilidad de cuan depresiva puede estar
"RED"%d"ANSI_COLOR_RESET" ",cal);
            printf("\nSe recomienda hacer un analisis profundo si es necesario* ");
            printf("\nEnter para continuar con la creacion de la ficha del paciente");
            getch();
            regsp();
        }
}

```

```

void regsp(void){

```

```

    fflush( stdin );
    FILE* avr;
    char textoa[5000];
    char infa[MAX];
    const char* str1 = "c:/DDEPS/registros/";
    char ubifin[MAX];
    int opc, op2;
    system("cls");

```

```

    printf(CI"\t\t\t=====

```

```

\n");

```

```

    printf("\t\t\t##
    printf("\t\t\t##"ANSI_COLOR_RESET"

```

```

        ##\n");
    RECUERDE QUE PARA LA CREACIÓN

```

```

"CI"##\n");

```

```

    printf("\t\t\t##"ANSI_COLOR_RESET"

```

```

    DEL HISTORIAL DE LA CONSULTA

```

```

"CI"##\n");

```

```
printf("\t\t\t###ANSI_COLOR_RESET" SE CREAR CON EL SIGUIENTE FORMATO:
"CI"##\n");
printf("\t\t\t###ANSI_COLOR_RESET" DIA/MES/AÑO/NOMBRE/APELLIDO
"CI"##\n");
printf("\t\t\t###ANSI_COLOR_RESET" EJM: 10_04_21_daniel_espinosa "CI"##\n");
printf("\t\t\t###ANSI_COLOR_RESET" RESPETE EL ORDEN O INFORME QUE SE INTRODUJO
"CI"##\n");
printf("\t\t\t###ANSI_COLOR_RESET" SI INTRODUCIO MAL LA INFORMACION AVISAR AL
ADMINISTRADOR "CI"##\n");
printf("\t\t\t###ANSI_COLOR_RESET" "CI"##\n");

printf("\t\t\t=====\\n
ANSI_COLOR_RESET);
printf("Ingresa informacion: ");
scanf("%s",infa);
strcat(strcpy(ubifin, str1), infa);
avr = fopen(ubifin, "r");
if (avr == NULL){
do{
printf("\\nRevisa si se introdujo bien la información, para continuar digita '1' o si no es así digita
'2' para reintentar :\\n");
fflush( stdin );
scanf("%d",&opc);
switch(opc){
case 1:
system("cls");
fflush( stdin );
printf("Escribe aqui el resultado y la conclusión que se llevo con el paciente\\n");
printf("Recuerda NO agragar saltos de linea o enter mientras escribes!\\n");
gets(textoa);
system("cls");
printf("Esto es lo que se guardara en el registro del paciente: \\n");
puts(textoa);
printf("Se volvera al menu al dar enter");
avr = fopen(ubifin, "w+");
fputs(textoa, avr);
fclose(avr);
getch();
menu();
break;
case 2:
system("cls");
regsp();
break;
default:
printf("\\nOpción incorrecta");
getch();
break;}
} while(opc!=2);
}else{
system("cls");

printf(CI"\t\t\t=====\\n");
printf("\t\t\t###
###\n");
printf("\t\t\t###ANSI_COLOR_RESET" REVISAR QUE TODO ESTE CORRECTO
"CI"##\n");
printf("\t\t\t###ANSI_COLOR_RESET" ES POSIBLE QUE EL REGISTRO EXISTA
"CI"##\n");
```

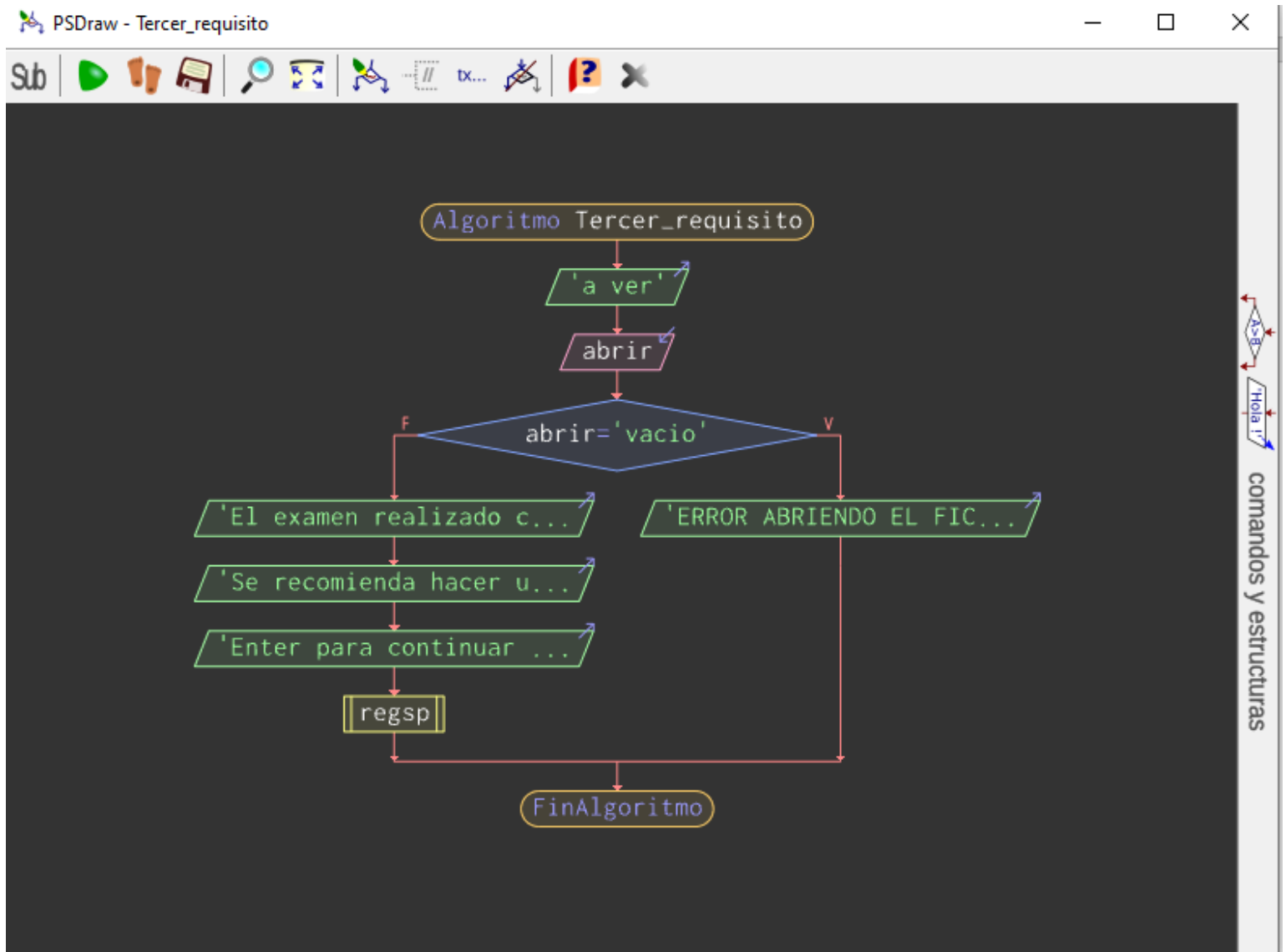
```

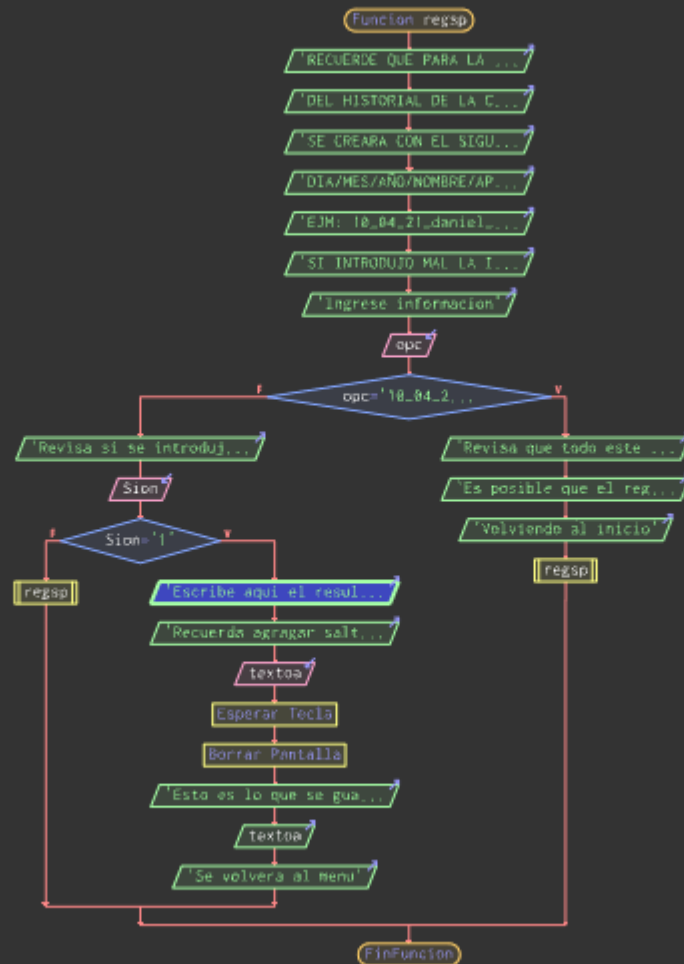
printf("\t\t\t###ANSI_COLOR_RESET"
"CI"##\n");
printf("\t\t\t###ANSI_COLOR_RESET"
"CI"##\n");

printf("\t\t\t=====\\n
ANSI_COLOR_RESET);
Sleep(2200);
regsp();
}
}

```

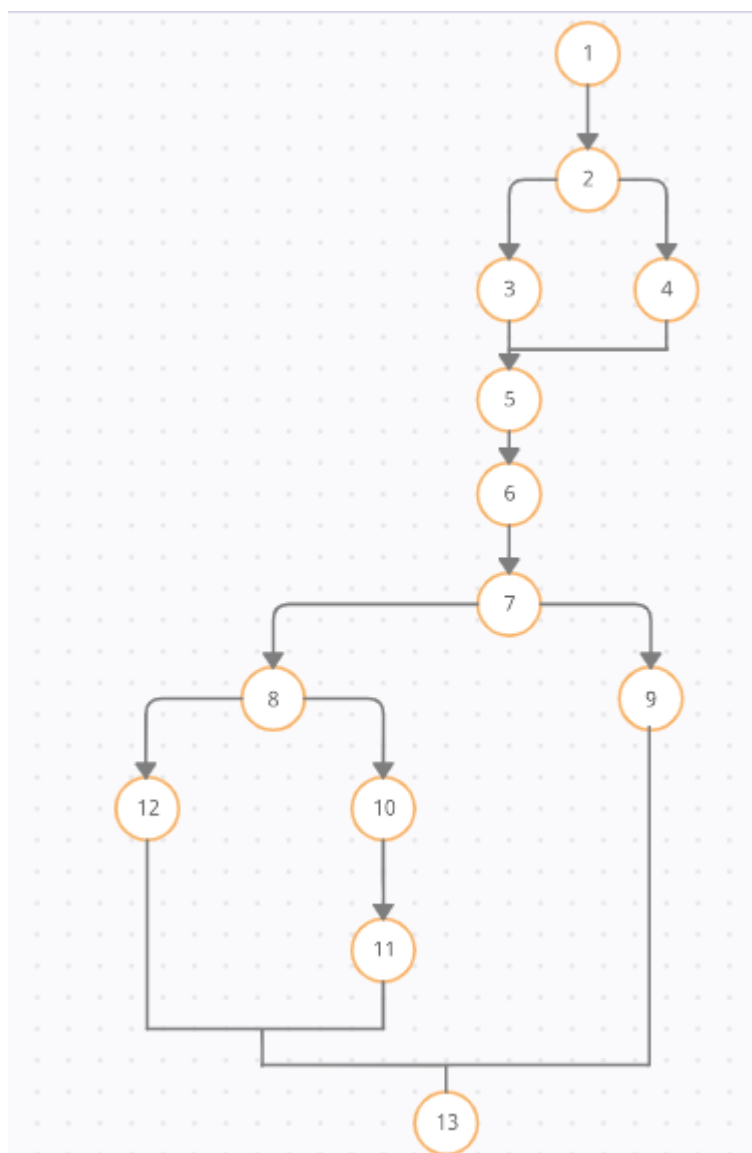
3. DIAGRAMA DE FLUJO





'Escribe aqui el resultado y la conclusión que se llego con el paciente'

4 GRAFO



RUTAS

R1: 1,2,3,5,6,7,8,10,11,13

R2: 1,2,4,5,6,7,9,13

R3: 1,2,3,5,6,7,8,12,13

COMPLEJIDAD CICLOMÁTICA

EJEMPLO:

Se puede calcular de las siguientes formas:

- $V(G) = A - N + 2 = 8 - 7 + 2 = 3$

DONDE:

A: Número de aristas

N: Número de nodos

- $V(G) = \text{número de nodos predados (IF)} + 1 = 3$

- $V(G) = A - N + 2 = 9 - 13 + 2 = 6$

- $V(G) = 5 + 1 = 6$

Prueba de Caja Blanca

***“LECTO DE PSICOPATOLOGÍAS
(DEPRESIÓN)”***

REQUISITO 001
Versión 4.0

Integrantes:

Joel Arguello
Cesar Loor
Josué Villavicencio

Fecha 2022-01-11

1. REQUISITO PLANTEADO

Historia de Usuario	
Número: REQ 001	Usuario: Administrador
Nombre de la Historia: Login	
Prioridad: Alta	
Programador Responsable: Cesar Loor	
Descripción: <ul style="list-style-type: none">• Ingresar al sistema con el usuario y contraseña existentes	
Validación: <ul style="list-style-type: none">• El usuario inicio correctamente sesión	

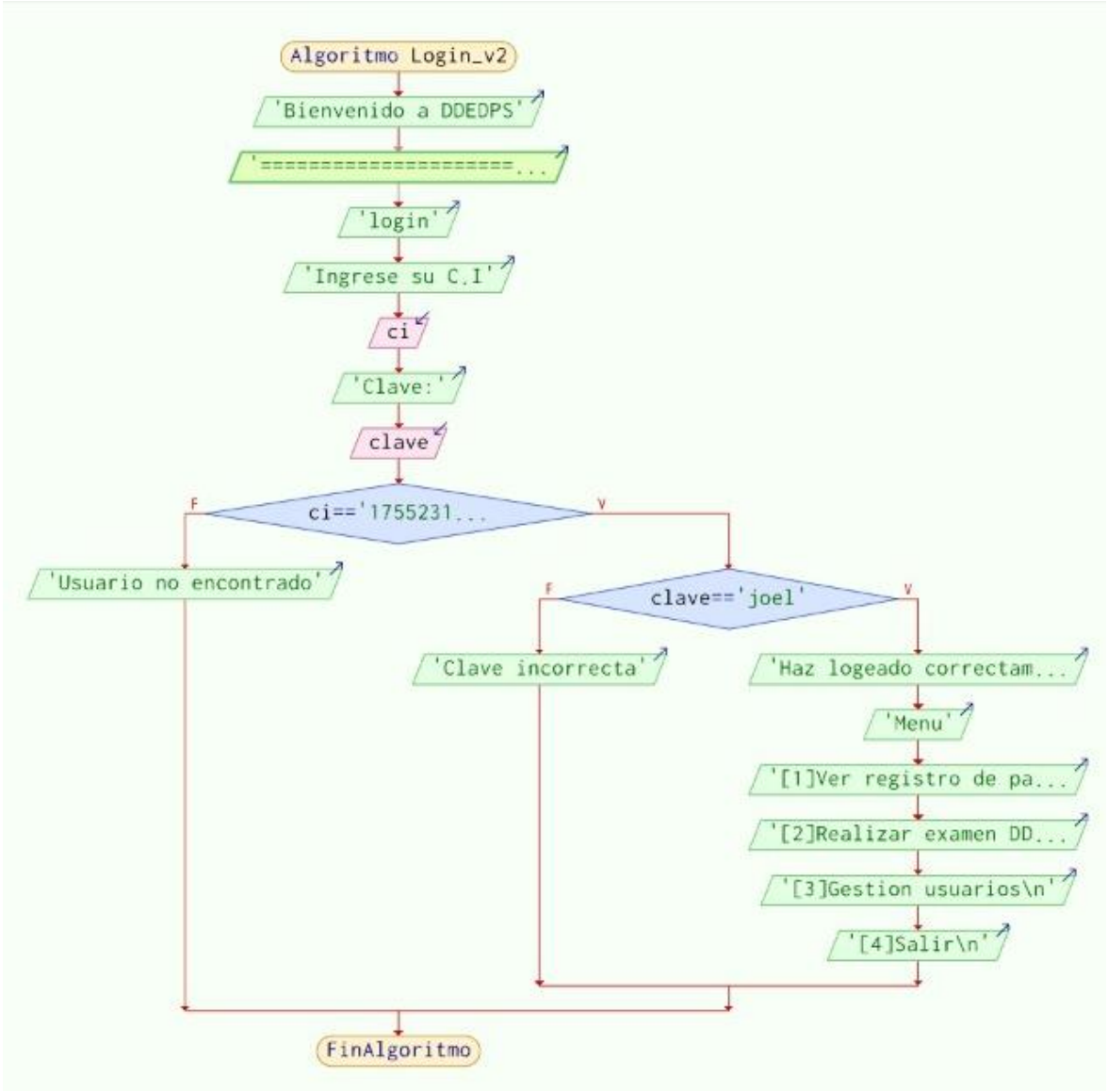
2.1. Código C

[illegible]

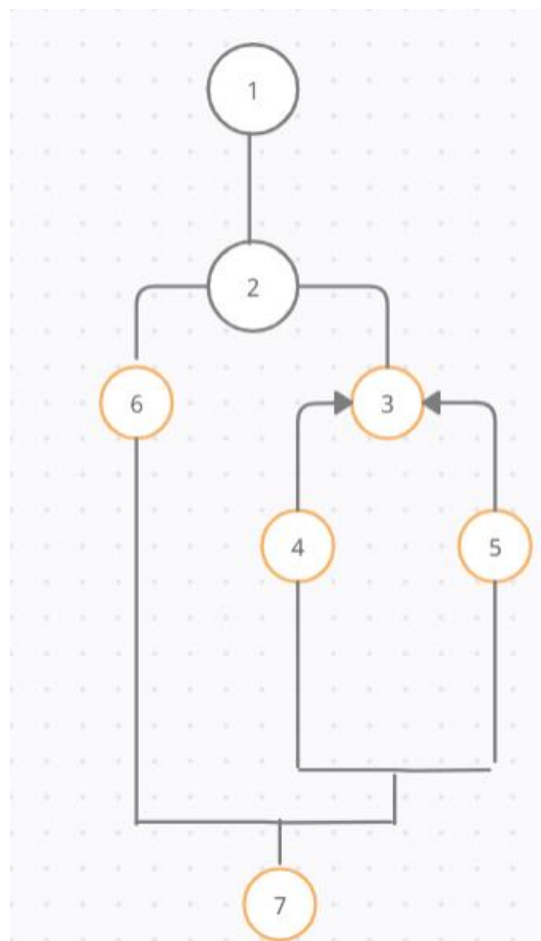
[illegible]

```
    }  
}  
while (ingresa == 0);  
}
```

3. DIAGRAMA DE FLUJO



4. GRAFO



5. Rutas

R1: 1,2,3,4,7

R2: 1,2,3,5,7

R3: 1,2,6,7

6. COMPLEJIDAD CICLOMÁTICA

EJEMPLO:

Se puede calcular de las siguientes formas:

- $V(G) = A - N + 2 = 8 - 7 + 2 = 3$

DONDE:

A: Número de aristas

N: Número de nodos

- $V(G) = \text{número de nodos predcados (IF)} + 1 = 3$

$$V(G) = A - N + 2 = 5 - 7 + 2 = 4$$

$$V(G) = \text{número de nodos predcados} + 1 = 3 + 1 = 4$$

Prueba de Caja Blanca

***“LECTODE PSICOPATOLOGÍAS
(DEPRESIÓN)”***

REQUISITO 005 Versión 1.0

Integrantes:

**Joel Arguello
Cesar Loor
Josué Villavicencio**

Fecha 2022-02-20

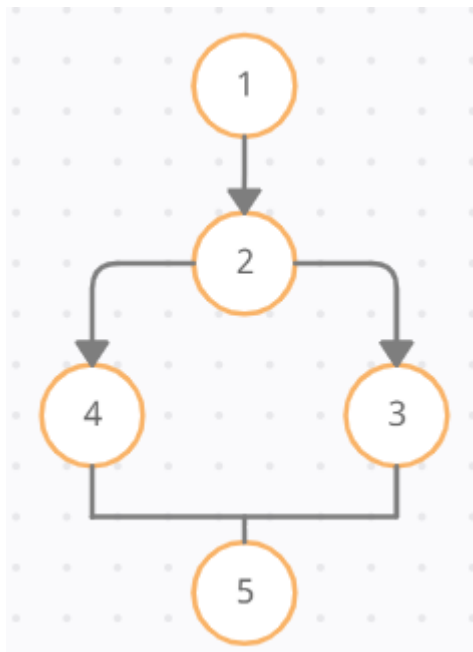
1. REQUISITO PLANTEADO

Historia de Usuario	
Número: REQ 005	Usuario: C.I
Nombre de la Historia: Ejecución del programa	
Prioridad: Alta	
Programador Responsable: Joel Arguello	
Descripción: <ul style="list-style-type: none">Se visualiza la ficha	
Validación: <ul style="list-style-type: none">Se ingresa el documento si no existe le regresa al menú.	

2.1. Código C

[illegible]

4 GRAFO



RUTAS

R1: 1,2,3,5

R2: 1,2,4,5

COMPLEJIDAD CICLOMÁTICA

EJEMPLO:

Se puede calcular de las siguientes formas:

- $V(G) = A - N + 2 = 8 - 7 + 2 = 3$

DONDE:

A: Número de aristas

N: Número de nodos

- $V(G) = \text{número de nodos predicados (IF)} + 1 = 3$

- $V(G) = A - N + 2 = 5 - 5 + 2 = 2$

- $V(G) = 1 + 1 = 2$

Prueba de Caja Blanca

“LECTO DE PSICOPATOLOGÍAS (DEPRESIÓN)”

REQUISITO 002
Versión 4.0

Integrantes:

**Joel Arguello
Cesar Loor
Josué Villavicencio**

Fecha 2022-02-13

1. REQUISITO PLANTEADO

Historia de Usuario	
Número: REQ 002	Usuario: C.I
Nombre de la Historia: Menú	
Prioridad: Alta	
Programador Responsable: Josue Villavicencio	
Descripción: <ul style="list-style-type: none">• Completado el proceso de ligar, automáticamente se desplegará el menú con diferentes opciones con las cuales el usuario podrá interactuar	
Validación: <ul style="list-style-type: none">• Si se ejecutan las tareas que se les pide se puede retornar al menú de inicio o cerrar el programa	

2. CÓDIGO FUENTE

2.1. Código C

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include <iostream>
#include <string.h>
#include<locale.h>
#include <windows.h>

#define PS    "\x1b[30m"
#define CI   "\x1b[36m"
#define ANSI_COLOR_RESET  "\x1b[0m"
#define RED  "\x1b[31m"


void gesuser(void);
void ruser(void);
void registr(void);
void exm(void);
void regist(void);
void elus(void);
void exam(void);
void menu(void);
void login(void);
void vru(void);

void menu(void)
{
    int opcc;
    int n;
    do {
        system("cls");

printf(CI"\t\t\t===== \nANSI_COLOR_RESE
T);
        printf(CI"\t\t\t##\n");
        printf("\t\t\t###ANSI_COLOR_RESET          MENU          "CI"##\n");
        printf("\t\t\t###ANSI_COLOR_RESET          [1]Ver registro de paciente      "CI"##\n");
        printf("\t\t\t###ANSI_COLOR_RESET          [2]Realizar examen DDEDPS       "CI"##\n");
        printf("\t\t\t###ANSI_COLOR_RESET          [3]Gestion usuarios         "CI"##\n");
        printf("\t\t\t###ANSI_COLOR_RESET          [4]Salir                      "CI"##\n");
        printf("\t\t\t###ANSI_COLOR_RESET          "CI"##\n");

printf(CI"\t\t\t===== \nANSI_COLOR_RESE
T);

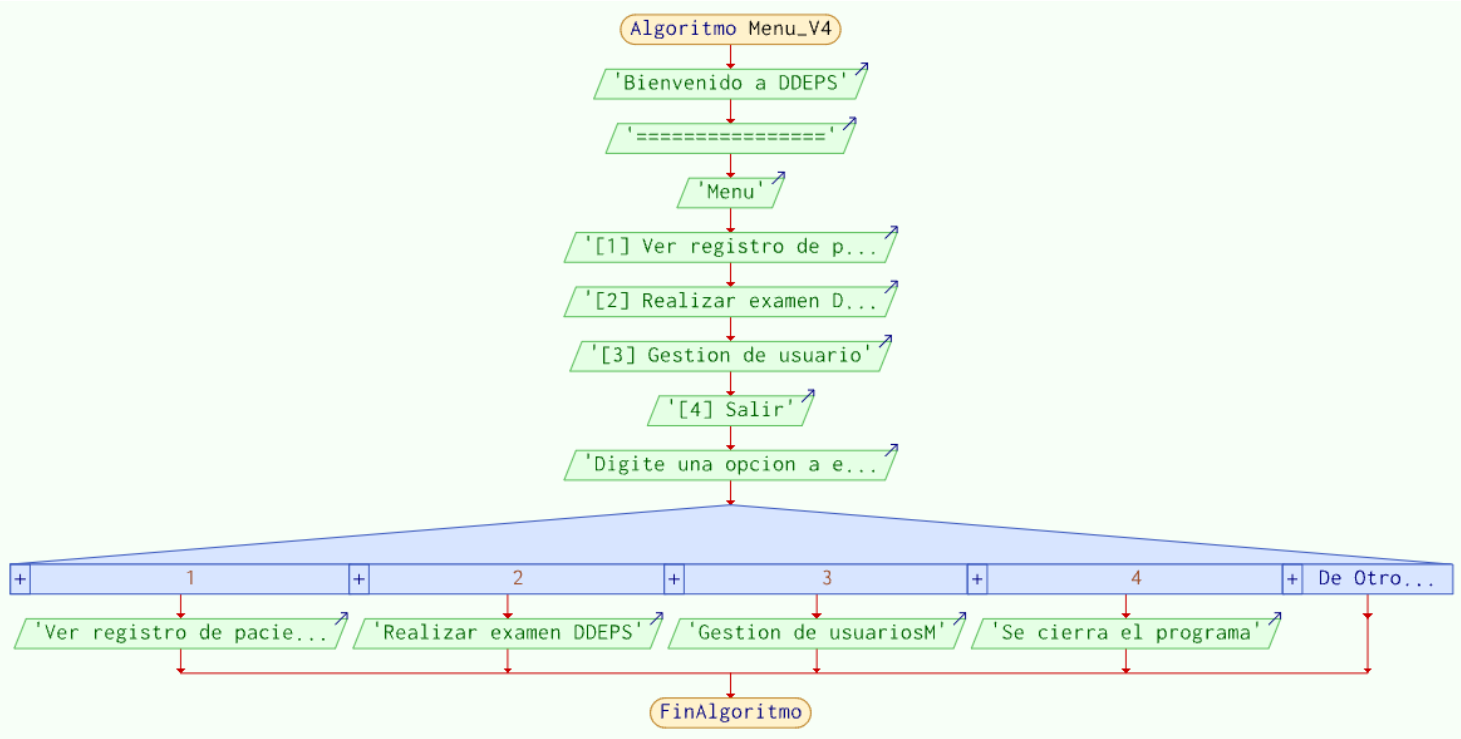
                printf("\nDigite opción a escoger: ");
                fflush( stdin );
                scanf("%d",&opcc);
                switch(opcc){
```

```

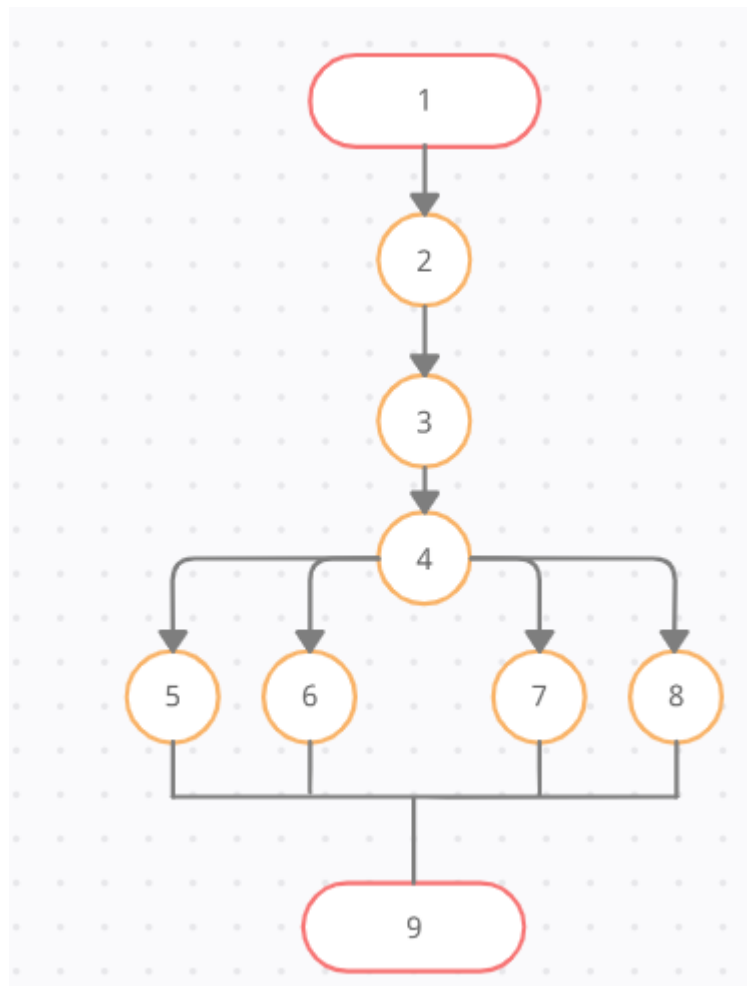
case 1 :
    system("cls");
    registr();
    break;
case 2:
    system("cls");
    exm();
    break;
case 3:
    gesuser();
    break;
case 4:
    exit(1);
    break;
default:
    printf("\nOpci%cn incorrecta",162);
    }
} while(opcc!=4);
}

```

3. DIAGRAMA DE FLUJO



4 GRAFO



4. Rutas:

R1: 1,2,3,4,5,9

R2: 1,2,3,4,6,9

R3: 1,2,3,4,7,9

R4: 1,2,3,4,8,9

EJEMPLO:

Se puede calcular de las siguientes formas:

- $V(G) = A - N + 2 = 8 - 7 + 2 = 3$

DONDE:

A: Número de aristas

N: Número de nodos

- $V(G) = \text{número de nodos predcados (IF)} + 1 = 3$

COMPLEJIDAD CICLOMÁTICA

$$V(G) = A - N + 2 = 10 - 9 + 2 = 3$$

$$V(G) = \text{número de nodos predcados} + 1 = 2 + 1 = 3$$

Prueba de Caja Blanca

***“LECTURA DE PSICOPATOLOGÍAS
(DEPRESIÓN)”***

REQUISITO 003 Versión 4.0

Integrantes:

**Joel Arguello
Cesar Looz
Josué Villavicencio**

Fecha 2022-01-18

1. REQUISITO PLANTEADO

Historia de Usuario	
Número: REQ 002	Usuario: C.I
Nombre de la Historia: Ejecución del programa	
Prioridad: Alta	
Programador Responsable: Joel Arguello	
Descripción: <ul style="list-style-type: none">El usuario llegando al menú lo único que haría sería escoger cualquiera de las opciones que dispone y si estas se ejecutan de forma correcta el usuario podrá retornar al menú para escoger otra opción o cerrar el programa	
Validación: <ul style="list-style-type: none">Si se ejecutan las tareas que se les pide se puede retornar al menú de inicio o cerrar el programa	

2. CÓDIGO FUENTE

2.1. Código C

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include <string.h>
#include <windows.h>

void gesuser(void)
{
    int opc;
    FILE *fp;
    char elimu[MAX];
    char cie[MAX];
    const char* str1 = "c:/DDEPS/usuarios/";
    char ubielu[MAX];
    char ubieu[MAX];
    char cir[10];
    char newpas[90];
    char clave[50];
    do{
        system("cls");
        printf(CI"\t\t\t=====
\nANSI_COLOR_RESET);
        printf(CI"\t\t\t##
        printf("\t\t\t###ANSI_COLOR_RESET"
        printf("\t\t\t###ANSI_COLOR_RESET"
        printf("\t\t\t###ANSI_COLOR_RESET"
        printf("\t\t\t###ANSI_COLOR_RESET"
        printf("\t\t\t###ANSI_COLOR_RESET"
        printf("\t\t\t###ANSI_COLOR_RESET"
        printf(CI"\t\t\t=====
\nANSI_COLOR_RESET);
        printf("\nDigite la opción a escoger: ");
        strcat(strcpy(ubielu, str1), cie);
        fflush( stdin );
        scanf("%d", &opc);
        switch(opc){
            case 1:
                system("cls");
                ruser();
                break;
            case 2:
                system("cls");
                fflush(stdin);
                system("cls");

        printf(CI"\t\t\t=====
\nANSI_COLOR_RESET);
        printf(CI"\t\t\t##
        printf("\t\t\t###ANSI_COLOR_RESET"
        printf(CI"\t\t\t##
        printf("\t\t\t###ANSI_COLOR_RESET"

        printf(CI"\t\t\t=====
```

```

\n"ANSI_COLOR_RESET);
    printf("Ingrese el C.I del doctor: ");
    scanf("%s",cie);
    strcat(strcpy(ubielu, str1), cie);
    fp = fopen (ubielu , "r");
    if(fp!=NULL)
    {
        fgets (clave, 50, fp);
        printf("\nUsuario: %s\n",cie);
        printf("La clave es la siguiente: ");
        puts(clave);
        getch();
        menu();
        fclose(fp);}
    else
    {

        printf(CI"\t\t\t=====
\n"ANSI_COLOR_RESET);
        printf(CI"\t\t\t##
        printf("\t\t\t##"ANSI_COLOR_RESET"
        No existe el usuario!
"CI"##\n");
        printf("\t\t\t##"ANSI_COLOR_RESET"
"CI"##\n");

        printf(CI"\t\t\t=====
\n"ANSI_COLOR_RESET);
        printf("No existe el usuario");
        getch();
        fclose(fp);
        menu();}
        break;
    case 3:
        system("cls");
        char cir[10];

        printf(CI"\t\t\t=====
\n");
        printf("\t\t\t##
        printf("\t\t\t##"ANSI_COLOR_RESET"
        ##\n");
        ELIMINAR USUARIO
"CI"##\n");
        printf("\t\t\t##"ANSI_COLOR_RESET"
"CI"##\n");

        printf("\t\t\t=====
\n"ANSI_COLOR_RESET);
        fflush(stdin);
        printf("Digite el numero de cédula del usuario a eliminar: ");
        scanf("%s",elimu);
        strcat(strcpy(ubieu, str1), elimu);
        if(remove(ubieu)==0){
            system("cls");

            printf(CI"\t\t\t=====
\n");
            printf("\t\t\t##
            printf("\t\t\t##"ANSI_COLOR_RESET"
            ##\n");
            C.I DEL USUARIO ELIMINADO
"CI"##\n");
            printf("\t\t\t##"ANSI COLOR RESET"
            %s

```

```

"CI"##\n",cir );
printf("\t\t\t###ANSI_COLOR_RESET"
Se eliminó de manera correcta

"CI"##\n");

printf("\t\t\t===== \n
ANSI_COLOR_RESET);
    getch();
    }else{
printf("\t\t\tHubo un error vuelva a intentar.\n");
printf("\t\t\tRecuerde que no puede borrar su propio usuario!\n");
    getch();}
    break;
    case 4:
        system("cls");
        menu();
        break;
    case 5:
        printf("Gracias por confiar en nosotros!\n");
        printf("Digite enter para salir\n");
        getch();
        exit(1);
        break;
    default:
        printf("\nOpción incorrecta");
        getch();
        break;
    }
}while(opc!=6);
}

void validar(char num[]){
    for (int i = 0; i < strlen(num); i++)
    {
        if(!isdigit(num[i]) ) {
            system("cls");

            printf(CI"\t\t\t=====
\nANSI_COLOR_RESET);
            printf(CI"\t\t\t##
            ##\n");
            printf("\t\t\t###ANSI_COLOR_RESET"
            ¡SOLO SE ACEPTAN CÉDULAS!

"CI"##\n");
            printf("\t\t\t###ANSI_COLOR_RESET"
            REGRESANDO AL MENU

"CI"##\n");
            printf("\t\t\t###ANSI_COLOR_RESET"
            "CI"##\n");

            printf(CI"\t\t\t=====
\nANSI_COLOR_RESET);
            getch();
            gesuser();
            break;
        }
    }
}

void ruser(void)
{
    int cedula;
    int ced,pares,impares,total,dec=0;
    int a,b,c,d,e,f,g,h,i,j,k;

```

```

FILE *fp;
char cir[MAX];
const char* str1 = "c:/DDEPS/usuarios/";
char ubieu[MAX];
char nombre[30];
char newpas[90];
char character;
system("cls");
printf(CI"\t\t\t=====
\n"ANSI_COLOR_RESET);
printf(CI"\t\t\t##
printf("\t\t\t###ANSI_COLOR_RESET"          ##\n");
printf("\t\t\t###ANSI_COLOR_RESET"          REGISTRO DE USUARIOS          "CI"##\n");
printf(CI"\t\t\t=====
\n"ANSI_COLOR_RESET);
fflush(stdin);
printf("Digite numero de cedula: ");
scanf("%s",cir);
strcat(strcpy(ubieu, str1), cir);
validar(cir);
cedula = atoi(cir);
ced=cedula;
a=cedula/1000000000;
cedula=cedula-(a*1000000000);
b=cedula/100000000;
cedula=cedula-(b*100000000);
c=cedula/10000000;
cedula=cedula-(c*10000000);
d=cedula/1000000;
cedula=cedula-(d*1000000);
e=cedula/100000;
cedula=cedula-(e*100000);
f=cedula/10000;
cedula=cedula-(f*10000);
g=cedula/1000;
cedula=cedula-(g*1000);
h=cedula/100;
cedula=cedula-(h*100);
i=cedula/10;
cedula=cedula-(i*10);
j=cedula/1;
cedula=cedula-(j*1);
if (cedula>2400000000){
printf("wtf");
}else{
    pares=b+d+f+h;
    a=a*2;
    if (a>9){
        a=a%10+a/10;
    }
    c=c*2;
    if (c>9){
        c=c%10+c/10;
    }
    e=e*2;
    if (e>9){
        e=e%10+e/10;
    }
    g=g*2;

```

```

if (g>9){
    g=g%10+g/10;
}
i=i*2;
if (i>9){
    i=i%10+i/10;
}
impares=a+c+e+g+i;
total=pares+impares;
while (dec-total!=j && dec<total+10){
    dec=dec+10;
}
if (dec-total==j)
{
    fflush(stdin);
    fp = fopen (ubieu      , "r");
    if(fp!=NULL){
        printf("\nError");
        printf("\nYa existe este usuario");
        getch();
        fclose(fp);}
    else {
        i = 0;
        fp = fopen (ubieu      , "w+");
        printf("Digite su clave: ");
        while (caracter = getch()) {
            if (caracter == TECLA_ENTER) {
                newpas[i] = '\0';
                break;
            } else if (caracter == TECLA_BACKSPACE) {
                if (i > 0) {
                    i--;
                    printf("\b \b");
                }

            } else {
                if (i < LONGITUD) {
                    printf("*");
                    newpas[i] = caracter;
                    i++;
                }
            }
        }
    }
}

```

```

printf("a\n"ANSI_COLOR_RESET);
fputs(newpas, fp);
system("cls");

```

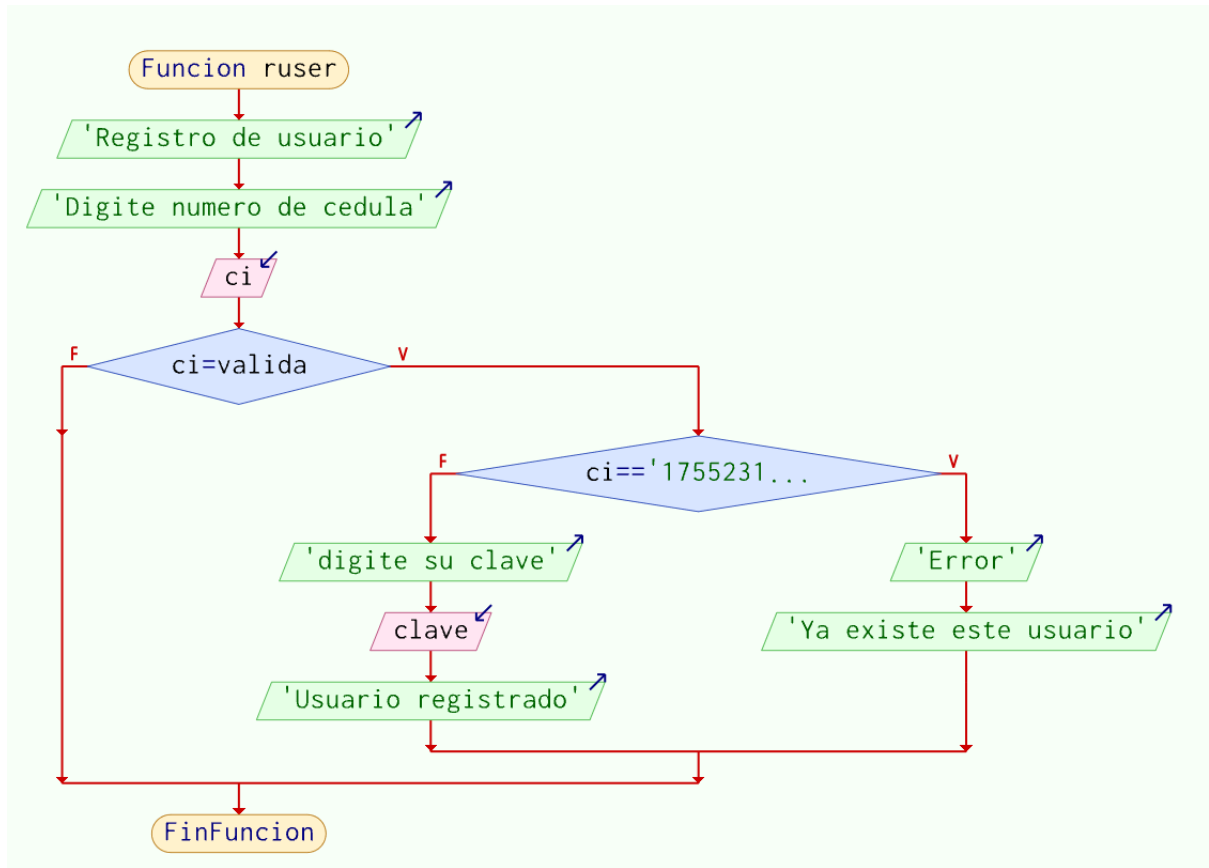
```

printf(CI"\n\n\n\t\t\t=====
=====\\n"ANSI_COLOR_RESET);
printf(CI"\t\t\t##
printf("\t\t\t###"ANSI_COLOR_RESET"
printf("\t\t\t###"ANSI_COLOR_RESET"
printf("\t\t\t###"ANSI_COLOR_RESET"
printf(CI"\t\t\t##

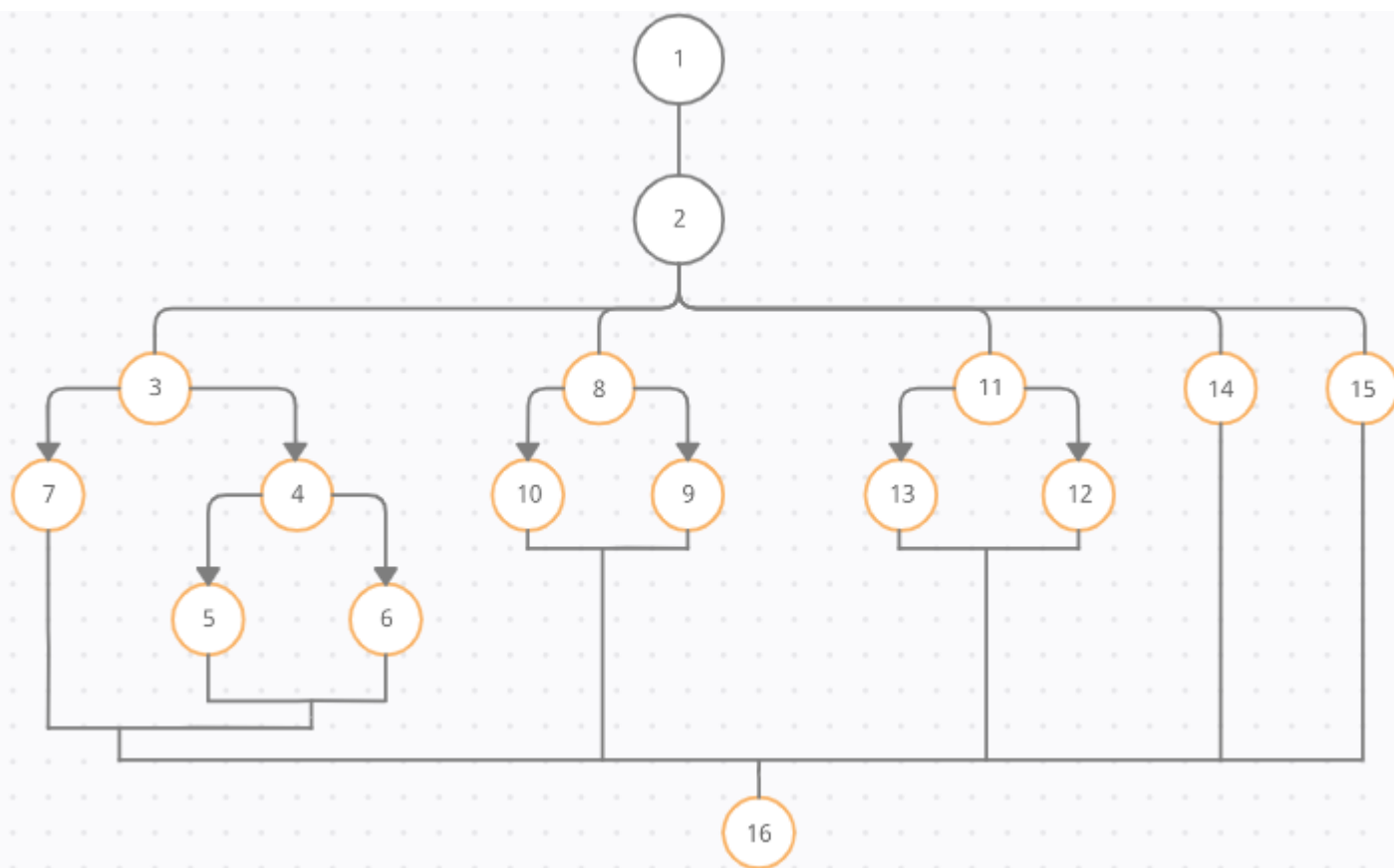
```

ÉXITO! "CI"##\n");
 "CI"##\n");
 resguardo "CI"##\n");

#####\n");
 USUARIO REGISTRADO CON
 :)
 Recuerde tener su clave en
 #####\n");



4. GRAFO



RUTAS

R1: 1,2,3,4,5,6,16

R2: 1,2,3,7,16

R3: 1,2,8,9,16

R4: 1,2,8,10,16

R5: 1,2,11,12,16

R6: 1,2,11,13,16

R7: 1,2,14,16

R8: 1,2,15,16

COMPLEJIDAD CICLOMÁTICA

EJEMPLO:

Se puede calcular de las siguientes formas:

- $V(G) = A - N + 2 = 8 - 7 + 2 = 3$

DONDE:

A: Número de aristas

N: Número de nodos

- $V(G) = \text{número de nodos predados (IF)} + 1 = 3$

- $V(G) = A - N + 2 = 29 - 16 + 2 = 15$

- $V(G) = 14 + 1 = 15$