

Prueba de Caja Blanca

“Optimización de gestión de pagos de una institución privada”

Integrantes:
Julio Sánchez
Anthony Caisaguano
Julio Andrade

Fecha 2025/01/19

Gestión de pegos para una institución privada	Caja blanca	Versión:5
		Página: 2
		Fecha: 12/05/2025

índice

HISTORIA DE REVISIÓN	3
REQUISITO PLANTEADO REQ 001-INICIAR SESION	4
CÓDIGO FUENTE.....	4
<i>DIAGRAMA DE FLUJO (DF)</i>	4
REQUISITO PLANTEADO REQ 002-REGISTRAR PADRE DE FAMILIA	8
CÓDIGO FUENTE.....	8
<i>DIAGRAMA DE FLUJO</i>	9
REQUISITO PLANTEADO REQ 003-REGISTRAR PAGO	14
CÓDIGO FUENTE.....	14
<i>DIAGRAMA DE FLUJO</i>	15
REQUISITO PLANTEADO REQ 004-BUSCAR EL REGISTRO DEL REPRESENTANTE	20
CÓDIGO FUENTE.....	20
<i>DIAGRAMA DE FLUJO</i>	21
REQUISITO PLANTEADO REQ 005-BUSCAR EL REGISTRO DE PAGOS	24
CÓDIGO FUENTE.....	24
<i>DIAGRAMA DE FLUJO</i>	25
REQUISITO PLANTEADO REQ 006-MODIFICAR INFORMACIÓN DEL REPRESENTANTE	28
CÓDIGO FUENTE.....	28
<i>DIAGRAMA DE FLUJO</i>	29

Gestión de pagos para una institución privada	Caja Blanca	Versión: 4
		Página: 3
		Fecha: 11/02/2025

Historia de Revisión

Fecha	Versión	Descripción	Autores
20/01/2025	1	Versión inicial	Anthony Caisaguano Julio Sánchez Julio Andrade
23/01/2025	2	Diagramas de flujo de cada requisito	Anthony Caisaguano Julio Sánchez Julio Andrade
28/01/2025	3	Grafos de flujo de cada requisito	Anthony Caisaguano Julio Sánchez Julio Andrade
11/02/2025	4	Identificación de las rutas y complejidad cíclica	Anthony Caisaguano Julio Sánchez Julio Andrade
20/02/2025	5	Redacción	Anthony Caisaguano Julio Sánchez Julio Andrade

Gestión de pegos para una institución privada	Caja blanca	Versión:5
		Página: 2
		Fecha: 12/05/2025

REQUISITO PLANTEADO REQ 001-INICIAR SESION

CÓDIGO FUENTE

```

void IniciarSesion(){
    char usuario[MAX_USER_SIZE];
    char password[MAX_PASSWORD_SIZE];
    int contador=0;

1   while(contador<3){
        printf("USUARIO: ");
        scanf("%s", usuario);

        /* Entrada segura de contrasenia */
        printf("PASSWORD: ");
        HANDLE hStdin = GetStdHandle(STD_INPUT_HANDLE);
        DWORD mode = 0;

        GetConsoleMode(hStdin, &mode);
        SetConsoleMode(hStdin, mode & (~ENABLE_ECHO_INPUT));

        scanf("%s", password);
        SetConsoleMode(hStdin, mode);
        /* Fin entrada segura de contrasenia */

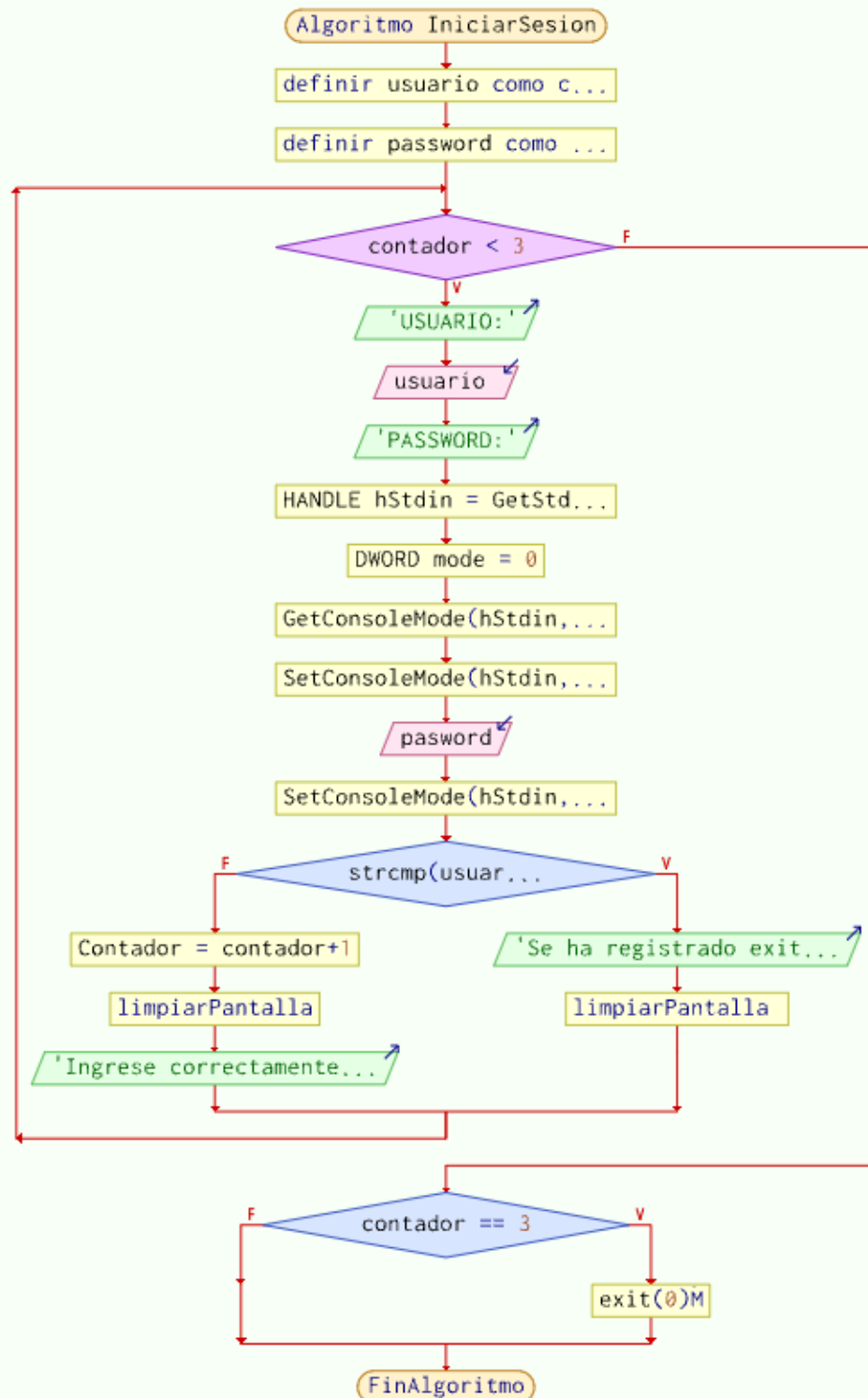
2   if(strcmp(usuarioSave, usuario) == 0 && strcmp(passwordSave, password) == 0){
        printf("Se ha registrado exitosamente\n");
        limpiarPantalla();
        return;
    } else{
        contador=contador+1;
        limpiarPantalla();
        puts("Ingrese correctamente sus credenciales");
        getch();
    }

3   if(contador==3){
        exit(0);
    }

    return;
}

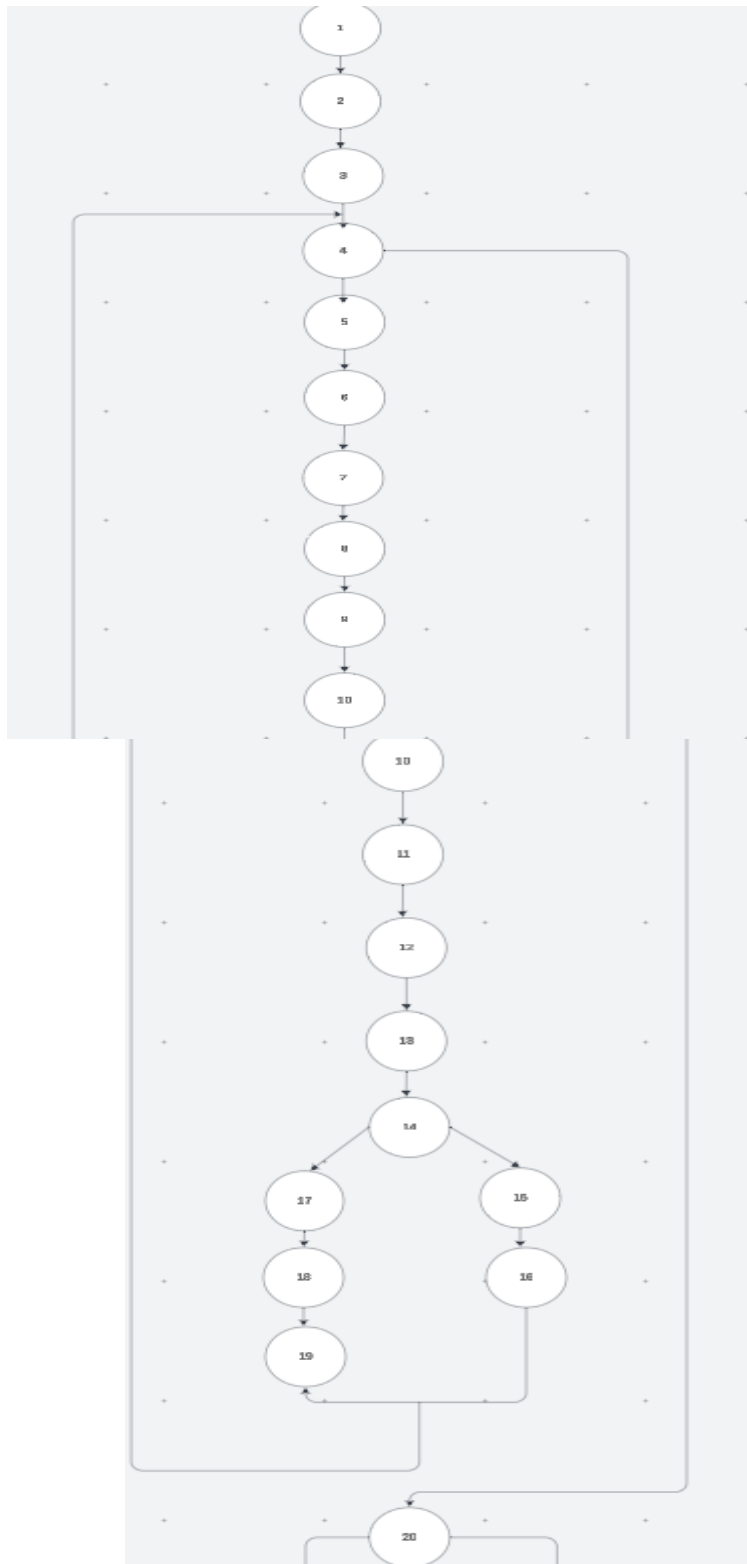
```

DIAGRAMA DE FLUJO (DF)

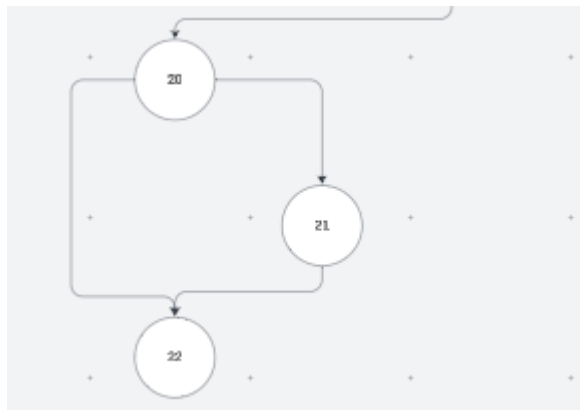


Gestión de pegos para una institución privada	Caja blanca	Versión:5
		Página: 2
		Fecha: 12/05/2025

GRAFO DE FLUJO (GF)



Gestión de pagos para una institución privada	Caja Blanca	Versión: 4
		Página: 3
		Fecha: 11/02/2025



IDENTIFICACIÓN DE LAS RUTAS (Camino básico)

Ruta 1: 1,2,3,4,20,21,22

Ruta 2: 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,4

Ruta 3: 1,2,3,4,5,6,7,8,9,10,11,12,13,14,17,18,19,4

Ruta 4: 1,2,3,4,20,22

COMPLEJIDAD CICLOMÁTICA

Fórmula basada en los nodos predicados (decisiones):

$V(G)$: Numero de nodos predicados(decisiones)+1	3+1
$V(G)=$	4
$V(G)=A(\text{Aristas})-N(\text{Nodos})+2$	24-22+2
$V(G)=$	4

DONDE:	
P: Numero de nodos predicado	4
A: Numero de aristas	24
N: Numero de nodos	22

Gestión de pegos para una institución privada	Caja blanca	Versión:5
		Página: 2
		Fecha: 12/05/2025

REQUISITO PLANTEADO REQ 002-REGISTRAR PADRE DE FAMILIA

CÓDIGO FUENTE

```

1 void RegistroPadreFamilia() {
    FILE *fptrl=fopen("DB\\listadoRepresentante\\listaRepresentantes1.txt", "a");

    if (fptrl==NULL) {
        printf("Archivo no encontrado\n");
    }

    infoRegistroRepresentante registroRepresentante;

    //Entrada de la informacion registrada por el usuario (En esta casa el product owner) acerca de los datos personales de los representantes

    puts("Ingresar el numero de cedula del representante: ");
    scanf("%d", &registroRepresentante.idRepresentante);

    if (registroRepresentante.idRepresentante<0) {
        puts("Ingresar un numero positivo");
        return;
    }

    while (getchar() != '\n');

    puts("Ingresar el primer nombre del representante: ");
    fgets(registroRepresentante.nombreRepresentante, 32, stdin);
    registroRepresentante.nombreRepresentante[strlen(registroRepresentante.nombreRepresentante, "\n")] = 0;

    puts("Ingresar el primer apellido del representante: ");
    fgets(registroRepresentante.apellidoRepresentante, 32, stdin);
    registroRepresentante.apellidoRepresentante[strlen(registroRepresentante.apellidoRepresentante, "\n")] = 0;

    puts("Ingresar el correo electronico del representante: ");
    fgets(registroRepresentante.correoElectronico, 32, stdin);
    registroRepresentante.correoElectronico[strlen(registroRepresentante.correoElectronico, "\n")] = 0;

    puts("Ingresar el numero de celular del representante: ");
    scanf("%d", &registroRepresentante.numeroCelular);
    while (getchar() != '\n');

    puts("Ingresar la calle principal de domicilio: ");
    fgets(registroRepresentante.direccionCP, 32, stdin);
    registroRepresentante.direccionCP[strlen(registroRepresentante.direccionCP, "\n")] = 0;

    puts("Ingresar la calle secundaria de domicilio: ");
    fgets(registroRepresentante.direccionCS, 32, stdin);
    registroRepresentante.direccionCS[strlen(registroRepresentante.direccionCS, "\n")] = 0;

    puts("Ingresar el sector del domicilio: ");
    fgets(registroRepresentante.sector, 32, stdin);
    registroRepresentante.sector[strlen(registroRepresentante.sector, "\n")] = 0;

    puts("Ingresar el primer nombre del estudiante representado: ");
    fgets(registroRepresentante.nombreNiño, 32, stdin);
    registroRepresentante.nombreNiño[strlen(registroRepresentante.nombreNiño, "\n")] = 0;

    //Salida y lo que se escribirá en el archivo sobre los datos personales del representante
    fprintf(fptrl, "%d\n", registroRepresentante.idRepresentante);
    fprintf(fptrl, "%s\n", registroRepresentante.apellidoRepresentante);
    fprintf(fptrl, "%s\n", registroRepresentante.nombreRepresentante);
    fprintf(fptrl, "%s\n", registroRepresentante.correoElectronico);
    fprintf(fptrl, "%d\n", registroRepresentante.numeroCelular);
    fprintf(fptrl, "%s\n", registroRepresentante.nombreNiño);
    fprintf(fptrl, "%s\n", registroRepresentante.direccionCP);
    fprintf(fptrl, "%s\n", registroRepresentante.direccionCS);
    fprintf(fptrl, "%s\n", registroRepresentante.sector);
    fprintf(fptrl, "-----\n");

    fclose(fptrl);

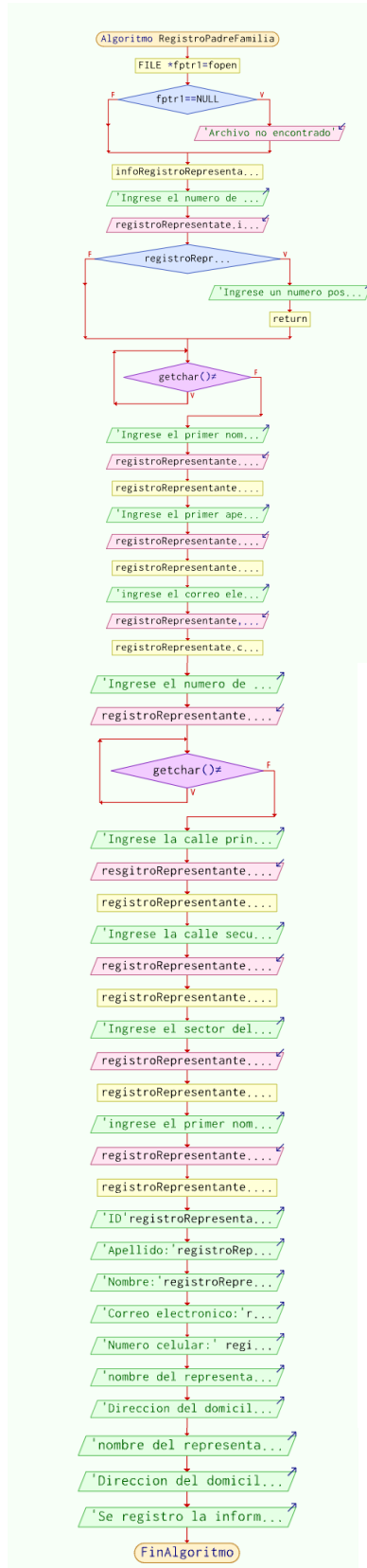
    printf("Se registro la informacion del representante exitosamente...\n");
    getch();

    return;
}

```

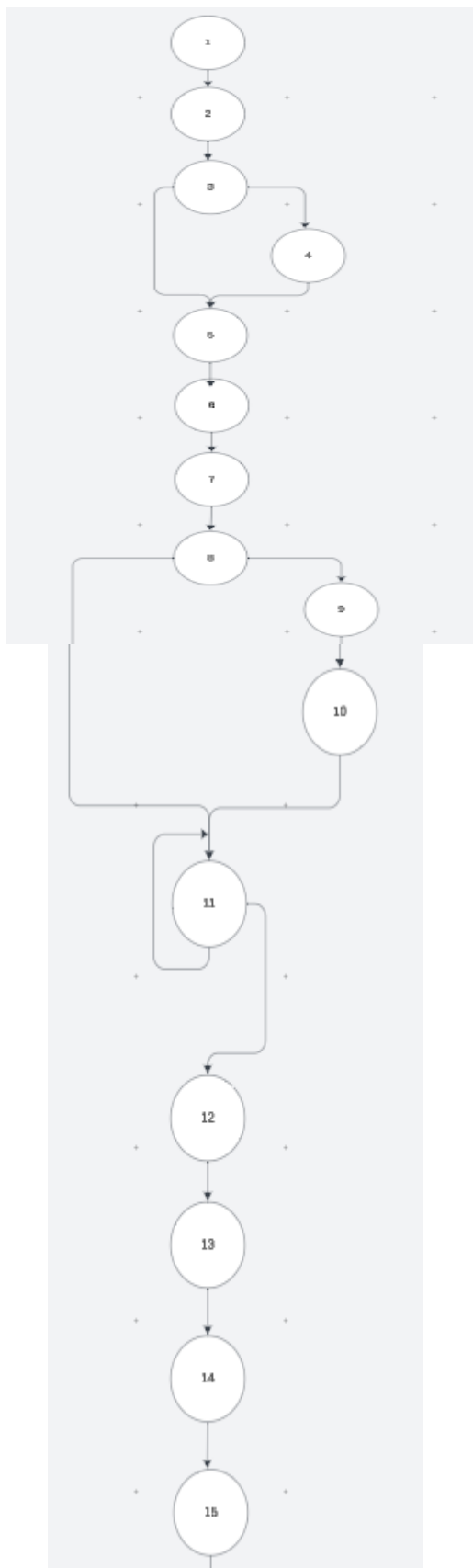

Gestión de pagos para una institución privada	Caja Blanca	Versión: 4
		Página: 3
		Fecha: 11/02/2025

DIAGRAMA DE FLUJO

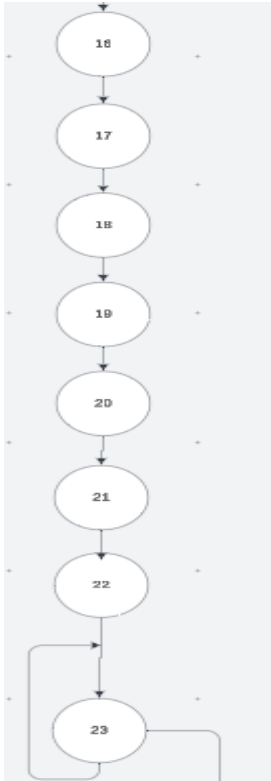


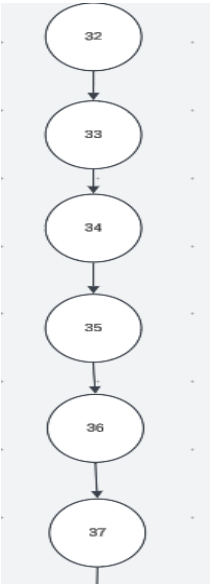
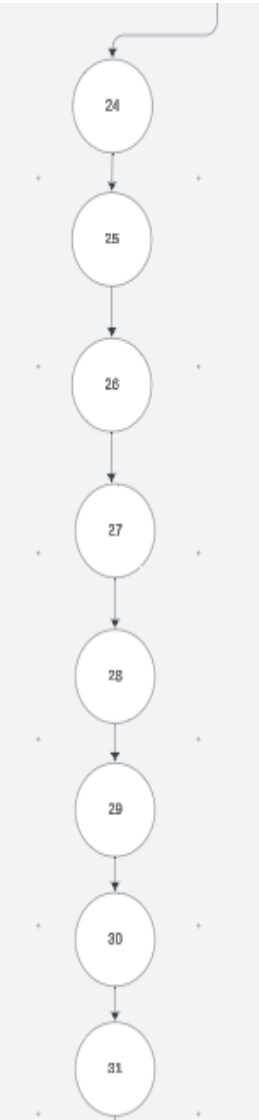
Gestión de pegos para una institución privada	Caja blanca	Versión:5
		Página: 2
		Fecha: 12/05/2025

GRAFO DE FLUJO



Gestión de pagos para una institución privada	Caja Blanca	Versión: 4
		Página: 3
		Fecha: 11/02/2025





Gestión de pagos para una institución privada	Caja Blanca	Versión: 4
		Página: 3
		Fecha: 11/02/2025



IDENTIFICACION DE RUTAS

Ruta1:

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46

Ruta2: 1,2,3,5,6,7,8,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,45,46

Ruta3: 1,2,3,4,5,6,7,8,9,10,11

Ruta4: 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23

Ruta5:

1,2,3,5,6,7,8,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,45,46

COMPLEJIDAD CICLOMÁTICA

$V(G)$: Numero de nodos predicados(decisiones)+1	4+1
$V(G)=$	5
$V(G)=A(\text{Aristas})-N(\text{Nodos})+2$	49-46+2
$V(G)=$	5

DONDE:	
P: Numero de nodos predicado	4
A: Numero de aristas	15

Gestión de pegos para una institución privada	Caja blanca	Versión:5
		Página: 2
		Fecha: 12/05/2025

N: Numero de nodos	15
--------------------	----

REQUISITO PLANTEADO REQ 003-REGISTRAR PAGO

CÓDIGO FUENTE

```

1 void RegistrarPagos(){
    FILE *fptr2=fopen("DB\\listadoPagos\\listaPagos1.txt", "a");
    if(fptr2==NULL){
        printf("Archivo no encontrado\n");
    }

    time_t t; //Para almacenar Los segundos
    struct tm *tmInfo; //Estructura donde se va a guardar

    time(&t); //Obtenemos La hora actual en segundos mediante La función time y almacenamos en La varibla t tipo time_t
    tmInfo=localtime(&t); //Localtime toma La informacion de La cantidad de segundos almacenados en La variable t y Lo convierte en La estructura tm

    char fecha[32];
    strftime(fecha, sizeof(fecha), "%Y/%m/%d a las %H:%M h", tmInfo);

    infoRegistroPg registroPg;

    //Entrada por teclado del usuario (En este caso el product owner)
    puts("Ingrese el numero de cedula del representante: ");
    scanf("%d", &registroPg.idRepsant);

    while(getchar() != '\n');

    puts("Ingrese el primer nombre del representante: (Escriba todo en mayusculas o minusculas)");
    fgets(registroPg.nombreRpsant, 32, stdin);
    registroPg.nombreRpsant[strcspn(registroPg.nombreRpsant, "\n")] = 0;

    /*busca La primera aparición de un carácter en string1 que pertenece al conjunto de caracteres
    especificado por string2 y devolverá La posición en name en La cual fue encontrado el match (Los caracteres nulos no se tienen en cuenta en La búsqueda: /0)*/

    puts("Ingrese el primer apellido del representante: (Escriba todo en mayusculas o minusculas)");
    fgets(registroPg.apellidoRpsant, 32, stdin);
    registroPg.apellidoRpsant[strcspn(registroPg.apellidoRpsant, "\n")] = 0;

    puts("Ingrese el primer nombre del estudiante representado: (Escriba todo en mayusculas o minusculas)");
    fgets(registroPg.nombreHj, 32, stdin);
    registroPg.nombreHj[strcspn(registroPg.nombreHj, "\n")] = 0;

    puts("Ingrese el pago realizado: ");
    scanf("%f", &registroPg.pago);

    //Salida y lo que escribira en el archivo
    fprintf(fptr2, "Id: %d\n", registroPg.idRepsant);
    fprintf(fptr2, "Apellido: %s\n", registroPg.apellidoRpsant);
    fprintf(fptr2, "Nombre: %s\n", registroPg.nombreRpsant);
    fprintf(fptr2, "Nombre del Representado: %s\n", registroPg.nombreHj);
    fprintf(fptr2, "Monto del pago efectuado: %.2f\n", registroPg.pago);

    fprintf(fptr2, "Fecha y hora del registro de pago: %s\n", fecha);
    fprintf(fptr2, "-----\n");

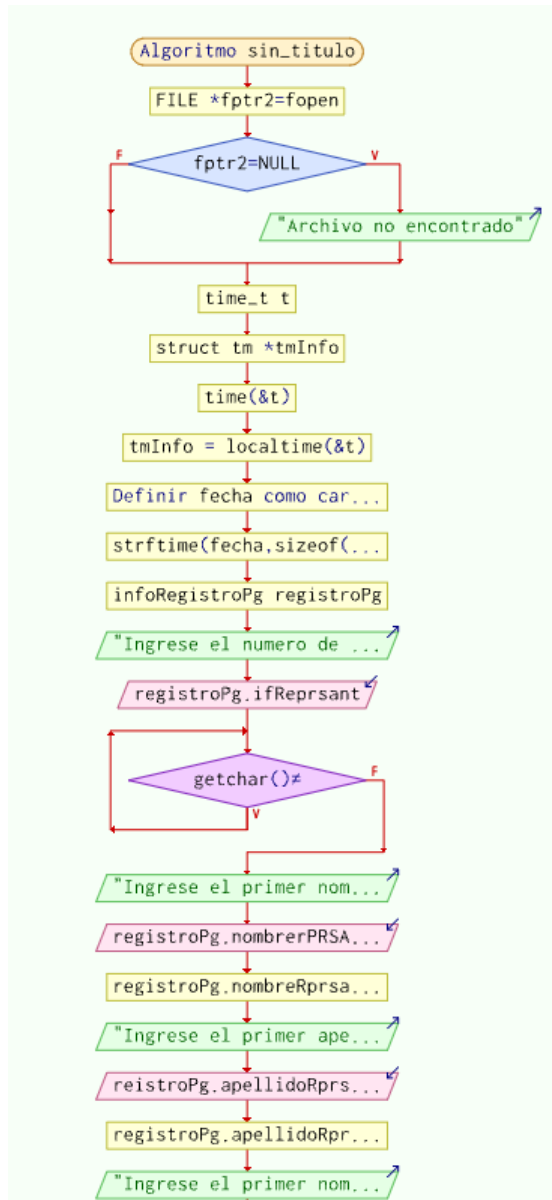
    fclose(fptr2);

    puts("Se registro el pago exitosamente...");
    getch();
    return;
}

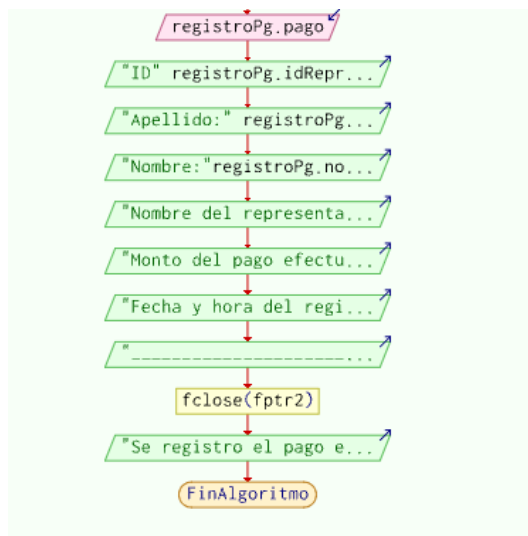
```

Gestión de pagos para una institución privada	Caja Blanca	Versión: 4
		Página: 3
		Fecha: 11/02/2025

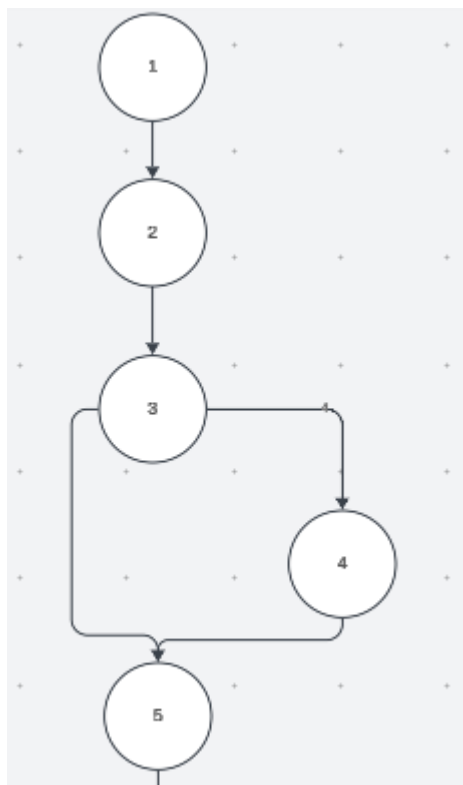
DIAGRAMA DE FLUJO



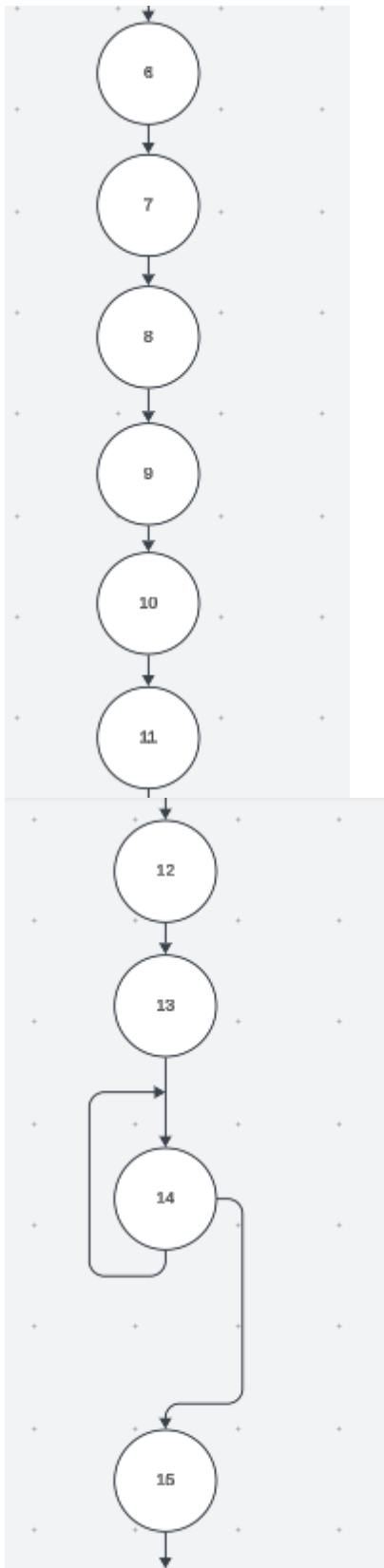
Gestión de pegos para una institución privada	Caja blanca	Versión:5
		Página: 2
		Fecha: 12/05/2025



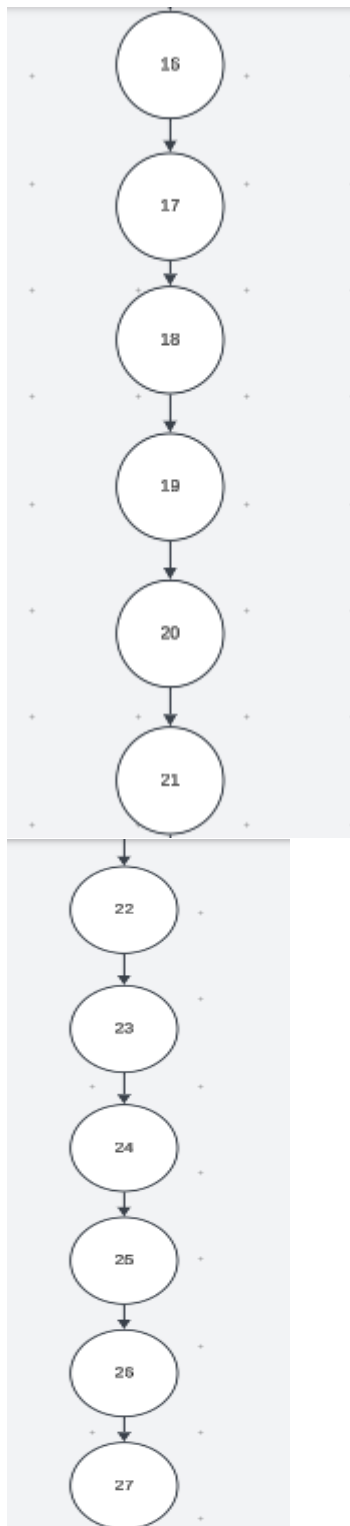
GRAFO DE FLUJO



Gestión de pagos para una institución privada	Caja Blanca	Versión: 4
		Página: 3
		Fecha: 11/02/2025



Gestión de pegos para una institución privada	Caja blanca	Versión:5
		Página: 2
		Fecha: 12/05/2025



Gestión de pagos para una institución privada	Caja Blanca	Versión: 4
		Página: 3
		Fecha: 11/02/2025



IDENTIFICACIÓN DE RUTAS

Ruta1: 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33

Ruta2: 1,2,3,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32

Ruta3: 1,2,3,4,5,6,7,8,9,10,11,12,13,14

COMPLEJIDAD CICLOMATICA

V(G): Numero de nodos predicados(decisiones)+1	2+1
V(G)=	3
V(G)=A(Aristas)-N(Nodos)+2	33-32+2
V(G)=	3

DONDE:	
P: Numero de nodos predicado	3
A: Numero de aristas	33
N: Numero de nodos	32

Gestión de pegos para una institución privada	Caja blanca	Versión:5
		Página: 2
		Fecha: 12/05/2025

REQUISITO PLANTEADO REQ 004-BUSCAR EL REGISTRO DEL REPRESENTANTE

CÓDIGO FUENTE

```

1 void buscarRepresentante() {
    FILE *fptr = fopen("DB/listadoRepresentante/listaRepresentantes1.txt", "r");
    if (fptr == NULL) {
        puts("No se ha encontrado el archivo de representantes!");
        return;
    }

    char criterio[32];
    puts("Ingrese el ID o Apellido del representante a buscar: ");
    scanf("%s", criterio);

    char linea[128];
    int encontrado = 0;

2 while (fgets(linea, 128, fptr)) {
    if (strstr(linea, criterio)) {
3         encontrado = 1;
        puts("\nRepresentante encontrado: ");

4         do {
            printf("%s", linea);
        } while (fgets(linea, 128, fptr) && strcmp(linea,
5             "-----\n") != 0);
            printf("-----\n");
        }

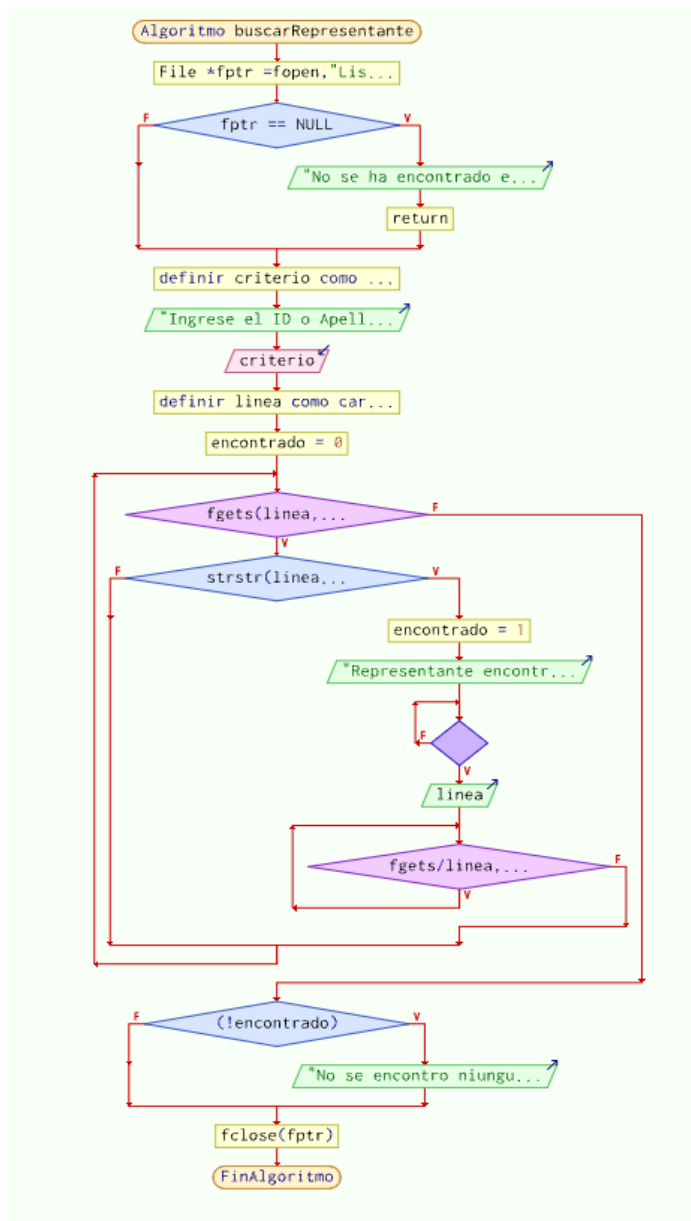
6 if (!encontrado) {
    printf("No se encontro ningun representante con el criterio '%s'.\n", criterio);
}

    fclose(fptr);
}

```

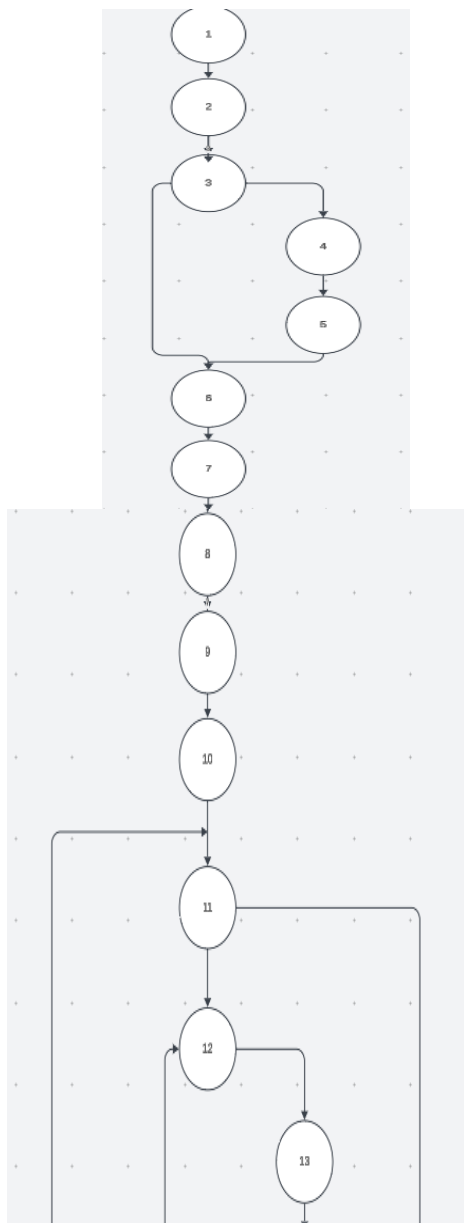
Gestión de pagos para una institución privada	Caja Blanca	Versión: 4
		Página: 3
		Fecha: 11/02/2025

DIAGRAMA DE FLUJO

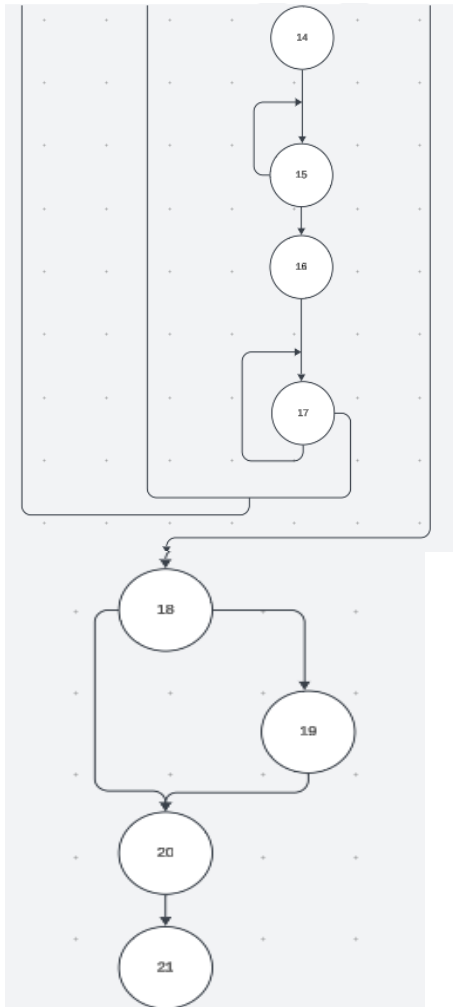


Gestión de pegos para una institución privada	Caja blanca	Versión:5
		Página: 2
		Fecha: 12/05/2025

GRAFO DE FLUJO



Gestión de pagos para una institución privada	Caja Blanca	Versión: 4
		Página: 3
		Fecha: 11/02/2025



IDENTIFICACIÓN DE RUTAS

- Ruta1:** 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17
Ruta2: 1,2,3,6,7,8,9,10,11,12,13,14,15,16,17
Ruta3: 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
Ruta4: 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,12
Ruta5: 1,2,3,4,5,6,7,8,9,10,11,18,19,20,21
Ruta6: 1,2,3,4,5,6,7,8,9,10,11,18,20,21
Ruta7: 1,2,3,6,7,8,9,10,11,12,13,14,15,16,17

COMPLEJIDAD CILCOMATICA

$V(G)$: Numero de nodos predicados(decisiones)+1	6+1
$V(G)=$	7
$V(G)=A(Aristas)-N(Nodos)+2$	26-21+2
$V(G)=$	2

DONDE:	
P: Numero de nodos predicado	7
A: Numero de aristas	26
N: Numero de nodos	21

Gestión de pagos para una institución privada	Caja blanca	Versión:5
		Página: 2
		Fecha: 12/05/2025

REQUISITO PLANTEADO REQ 005-BUSCAR EL REGISTRO DE PAGOS

CÓDIGO FUENTE

```

1 void buscarPago() {
    FILE *fptr = fopen("DB/listadoPagos/listaPagos1.txt", "r");
    if (fptr == NULL) {
        printf("No se ha encontrado el archivo de pagos!\n");
        return;
    }

    char criterio[32];
    printf("Ingrese el ID o el apellido del Representante a buscar: ");
    scanf("%s", criterio);

    char linea[128];
    int encontrado = 0;

    2 while (fgets(linea, 128, fptr)) {
        if (strstr(linea, criterio)) {
            encontrado = 1;
            printf("\nPago encontrado:\n",

            3 do {
                printf("%s", linea);
            } while (fgets(linea, 128, fptr) && strcmp(linea,

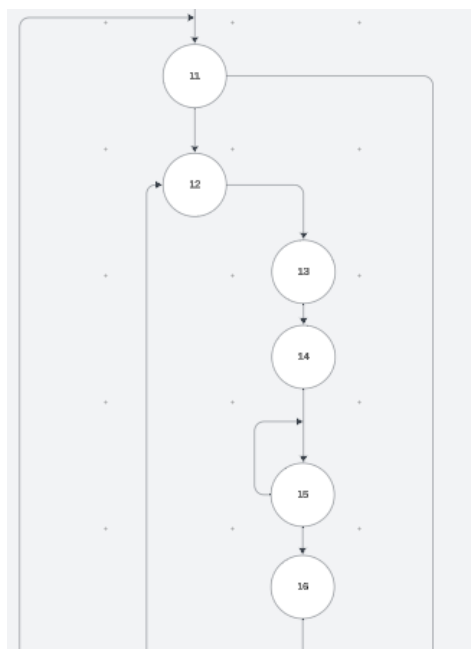
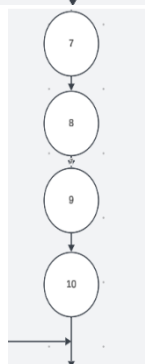
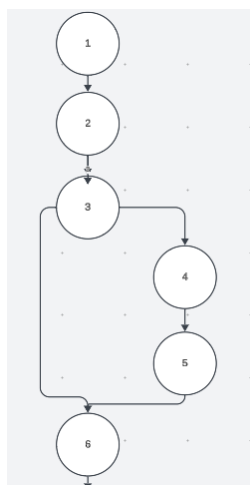
            4 ..... \n") != 0);
            printf(".....\n");
        }
    }

    5 if (!encontrado) {
        printf("No se encontro ningun pago con el criterio '%s'.\n", criterio);
    }
    /*puts("Ingrese nuevamente el ID para confirmar");
    scanf("%d", &idRepresentante);*/
    fclose(fptr);
}

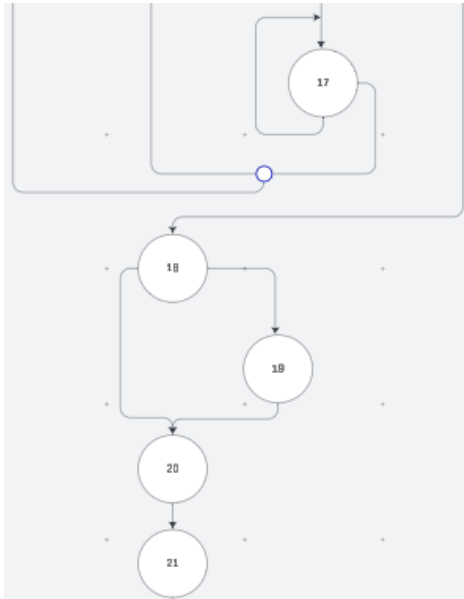
```


Gestión de pegos para una institución privada	Caja blanca	Versión:5
		Página: 2
		Fecha: 12/05/2025

GRAFO DE FLUJO



Gestión de pagos para una institución privada	Caja Blanca	Versión: 4
		Página: 3
		Fecha: 11/02/2025



IDENTIFICACIÓN DE RUTAS

Ruta1:1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17
Ruta2:1,2,3,6,7,8,9,10,11,12,13,14,15,16,17
Ruta3:1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
Ruta4:1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,12
Ruta5:1,2,3,4,5,6,7,8,9,10,11,18,19,20,21
Ruta6:1,2,3,4,5,6,7,8,9,10,11,18,20,21
Ruta7:1,2,3,6,7,8,9,10,11,12,13,14,15,16,17

COMPLEJIDAD CILCOMATICA

V(G): Numero de nodos predicados(decisiones)+1	6+1
V(G)=	7
V(G)=A(Aristas)-N(Nodos)+2	26-21+2
V(G)=	7

DONDE:	
P: Numero de nodos predicado	7
A: Numero de aristas	26
N: Numero de nodos	21

Gestión de pegos para una institución privada	Caja blanca	Versión:5
		Página: 2
		Fecha: 12/05/2025

REQUISITO PLANTEADO REQ 006-MODIFICAR INFORMACIÓN DEL REPRESENTANTE

CÓDIGO FUENTE

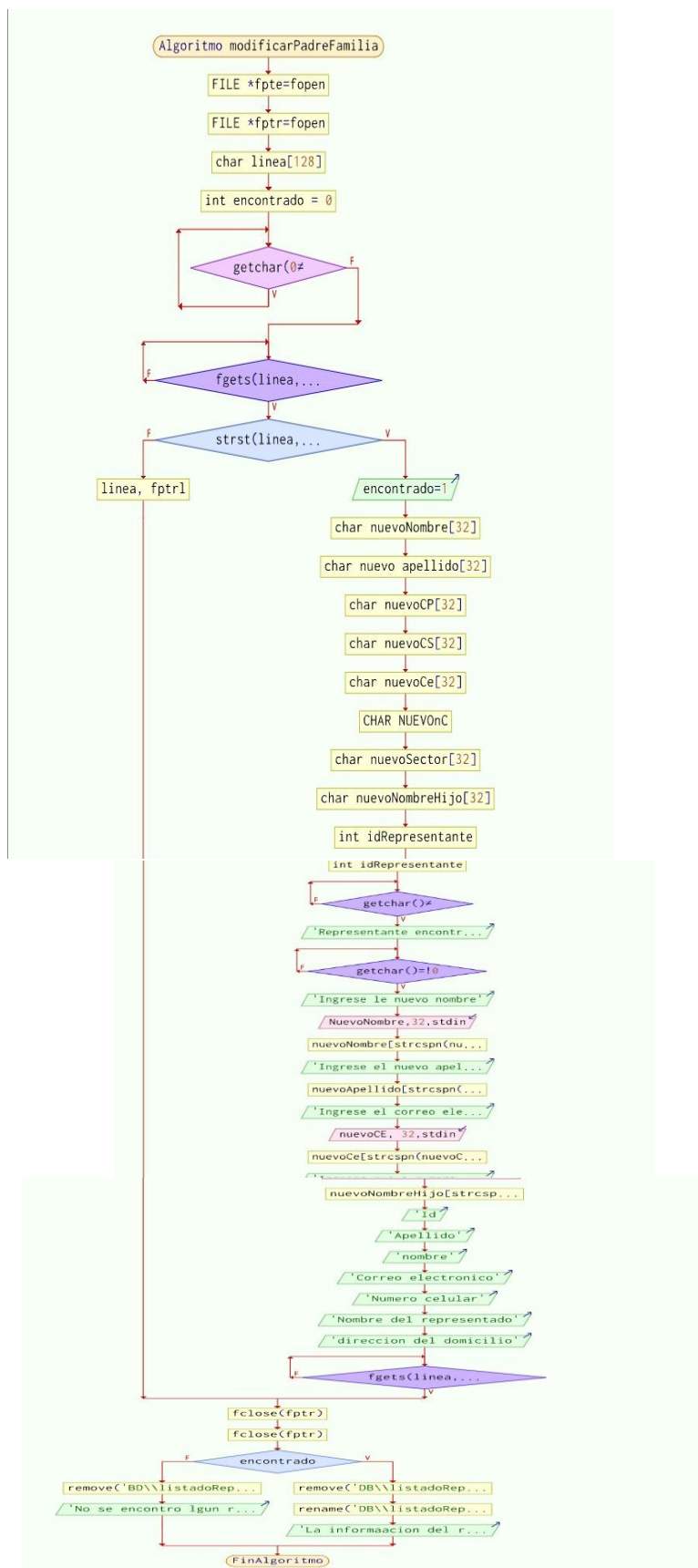
```

336 void modificarPadreFamilia(char *idRepresentante){
337     FILE *fptr=fopen("DB\\listadoRepresentante\\listaRepresentantes1.txt", "r");
338     FILE *fptr1 = fopen("DB\\listadoRepresentante\\temp.txt", "w");
339     char linea[128];
340     int encontrado = 0;
341
342     while (getchar() != '\n');
343     while (fgets(linea, 128, fptr)) {
344         // Buscar la línea que contiene el ID del producto
345         if (strstr(linea, idRepresentante) && strstr(linea, "ID: ")) {
346             encontrado = 1;
347
348             // Pedir nueva información al usuario
349             char nuevoNombre[32];
350             char nuevoApellido[32];
351             char nuevoCP[32];
352             char nuevoCS[32];
353             char nuevoCE[32];
354             int nuevoNC;
355             char nuevoSector[32];
356             char nuevoNombreHijo[32];
357             int idRepresentante;
358             while (getchar() != '\n');
359
360             puts("Representante encontrado. Ingrese los nuevos datos: ");
361             puts("Ingrese nuevamente el ID para confirmar");
362             scanf("%d", &idRepresentante);
363             while (getchar() != '\n');
364             puts("Ingrese el nuevo nombre del representante: ");
365             fgets(nuevoNombre, 32, stdin);
366             nuevoNombre[strcspn(nuevoNombre, "\n")] = 0;
367
368             printf("Ingrese el nuevo apellido del representante: \n");
369             fgets(nuevoApellido, 32, stdin);
370             nuevoApellido[strcspn(nuevoApellido, "\n")] = 0;
371
372             puts("Ingrese el correo electrónico del representante: ");
373             fgets(nuevoCE, 32, stdin);
374             nuevoCE[strcspn(nuevoCE, "\n")] = 0;
375
376             puts("Ingrese el número de celular del representante: ");
377             scanf("%d", &nuevoNC);
378             while (getchar() != '\n');
379
380             printf("Ingrese el nueva calle principal del domicilio: \n");
381             fgets(nuevoCP, 32, stdin);
382             nuevoCP[strcspn(nuevoCP, "\n")] = 0;
383
384             printf("Ingrese el nueva calle secundaria del domicilio: \n");
385             fgets(nuevoSector, 32, stdin);
386             nuevoSector[strcspn(nuevoSector, "\n")] = 0;
387
388             printf("Ingrese el nuevo sector del representante: \n");
389             fgets(nuevoSector, 32, stdin);
390             nuevoSector[strcspn(nuevoSector, "\n")] = 0;
391
392             printf("Ingrese el nuevo nombre del representante: \n");
393             fgets(nuevoNombreHijo, 32, stdin);
394             nuevoNombreHijo[strcspn(nuevoNombreHijo, "\n")] = 0;
395
396             // Escribir la nueva información en fptr1
397             fprintf(fptr1, "ID: %d\n", idRepresentante);
398             fprintf(fptr1, "Apellido: %s\n", nuevoApellido);
399             fprintf(fptr1, "Nombre: %s\n", nuevoNombre);
400             fprintf(fptr1, "Correo electrónico: %s\n", nuevoCE);
401             fprintf(fptr1, "Número celular: %d\n", nuevoNC);
402             fprintf(fptr1, "Nombre del representante: %s\n", nuevoNombreHijo);
403             fprintf(fptr1, "Dirección del domicilio: %s %s %s\n", nuevoCP, nuevoSector, nuevoCS);
404             fprintf(fptr1, "-----\n");
405
406             while (fgets(linea, 128, fptr) &&
407                    strcmp(linea, "-----\n") != 0);
408             fputs(linea, fptr1);
409         }
410     }
411     fclose(fptr);
412     fclose(fptr1);
413
414     if (encontrado) {
415         remove("DB\\listadoRepresentante\\listaRepresentantes1.txt");
416         rename("DB\\listadoRepresentante\\temp.txt", "DB\\listadoRepresentante\\listaRepresentantes1.txt");
417         printf("La información del representante con ID '%s' ha sido actualizada.\n", idRepresentante);
418     } else {
419         remove("DB\\listadoRepresentante\\temp.txt");
420         printf("No se encontro algun representante con el ID '%s'.\n", idRepresentante);
421     }
422     getch();
423     return;
424 }
425

```

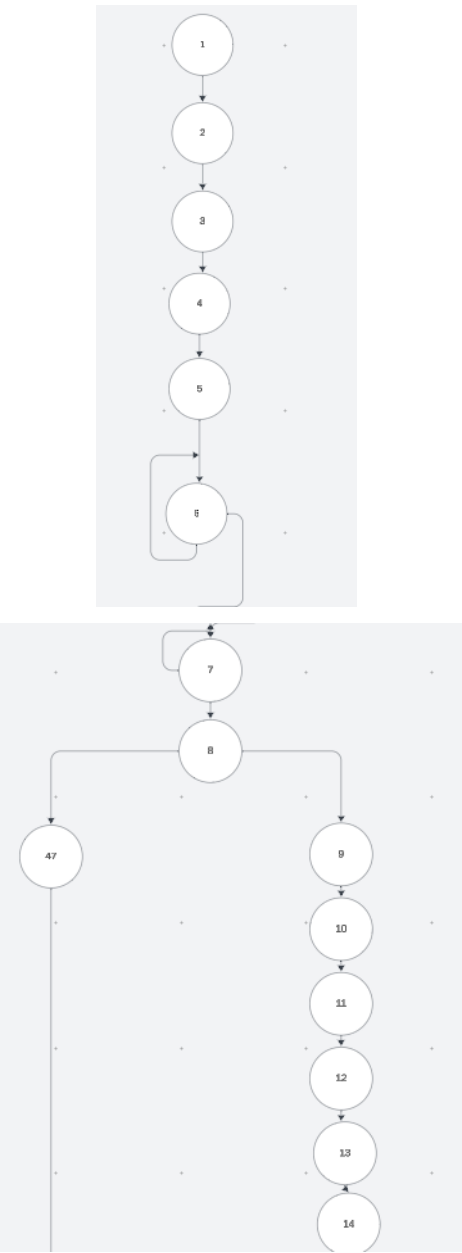
Gestión de pagos para una institución privada	Caja Blanca	Versión: 4
		Página: 3
		Fecha: 11/02/2025

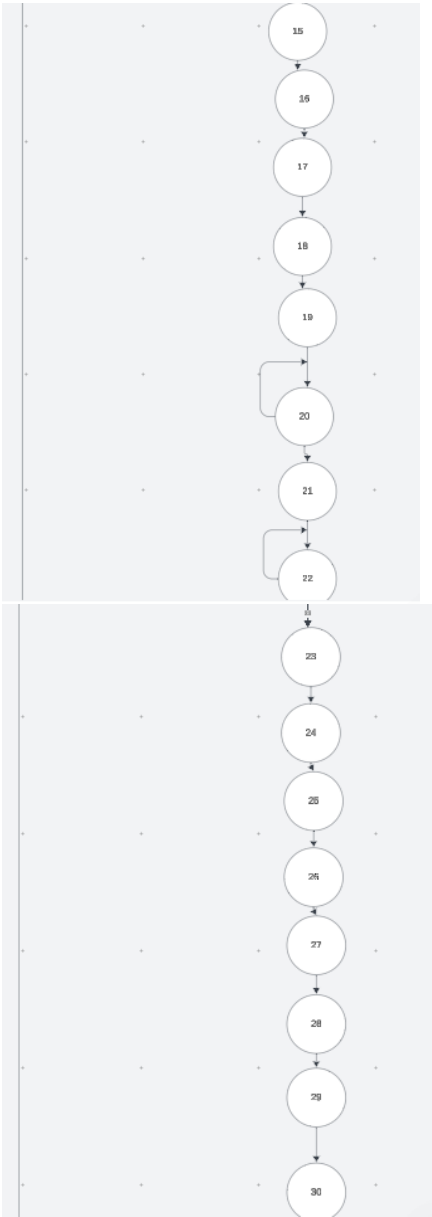
DIAGRAMA DE FLUJO



Gestión de pegos para una institución privada	Caja blanca	Versión:5
		Página: 2
		Fecha: 12/05/2025

GRAFO DE FLUJO

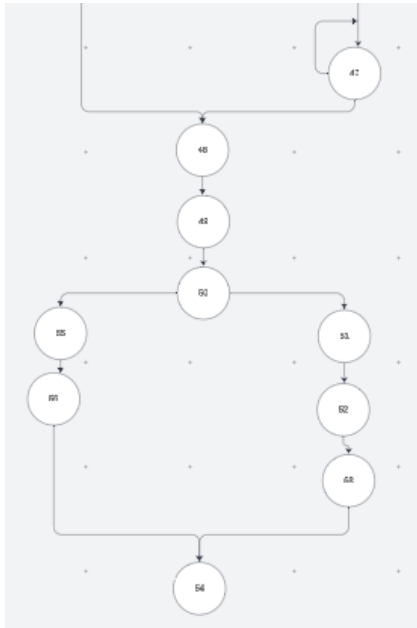




Gestión de pegos para una institución privada	Caja blanca	Versión:5
		Página: 2
		Fecha: 12/05/2025



Gestión de pagos para una institución privada	Caja Blanca	Versión: 4
		Página: 3
		Fecha: 11/02/2025



IDENTIFICACIÓN DE RUTAS

Ruta1: 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,48,49,50,51,52,53,54

Ruta2: 1,2,3,4,5,6,7,8,47,48,49,50,51,52,53,54

Ruta3: 1,2,3,4,5,6,7,8,47,48,49,50,55,56,54

Ruta4: 1,2,3,4,5,6

Ruta5: 1,2,3,4,5,6,7

Ruta6: 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20

Ruta7: 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22

Ruta8: 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34

Ruta9: 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46

COMPLEJIDAD CILCOMATICA

V(G): Numero de nodos predicados(decisiones)+1	8+1
V(G)=	9
V(G)=A(Aristas)-N(Nodos)+2	63-56+2
V(G)=	9

DONDE:	
P: Numero de nodos predicado	2
A: Numero de aristas	60
N: Numero de nodos	54