

Prueba de Caja Negra

“Optimización de gestión de pagos de una institución privada”

Integrantes:
Julio Andrade
Julio Sánchez
Anthony Caisaguano

Fecha: 2025-01-09

Historia de revisión

Fecha	Versión	Descripción	Autores
20/01/2025	1	Se cambio los requisitos	Anthony Caisaguano
23/01/2025	2	Se cambio los verbos infinitivos de los requisitos	Julio Sánchez
11/02/2025	3	Se realizado las capturas de pantalla de los códigos	Julio Andrade
18/02/2025	4	Se implemento las tablas de descripción de cada requisito	Julio Sánchez Anthony Caisaguano Julio Andrade
20/02/2025	5	Se implementa actualización de código	Julio Sánchez Anthony Caisaguano Julio Andrade

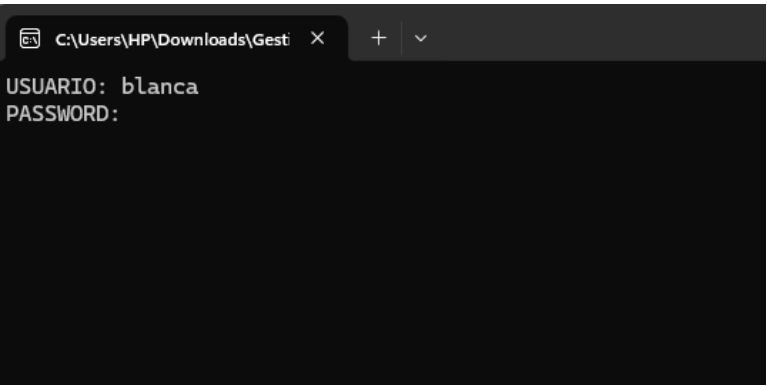
REQ 001 INICIAR SESION

Historia de usuario	
Numero: REQ 001	Usuario: Administrador
Nombre de la historia: Iniciar sesión	
Prioridad: Alta	
Programador Responsable: Julio Sánchez	
Descripción: <ul style="list-style-type: none">Se le dará una credencial al administrador proporcionado por nosotros solo para el administrador	
Validación: <ul style="list-style-type: none">Las credenciales proporcionadas por nosotros son los siguientes usuarios “blanca” y contraseña “contra” para que tenga acceso al sistema.	

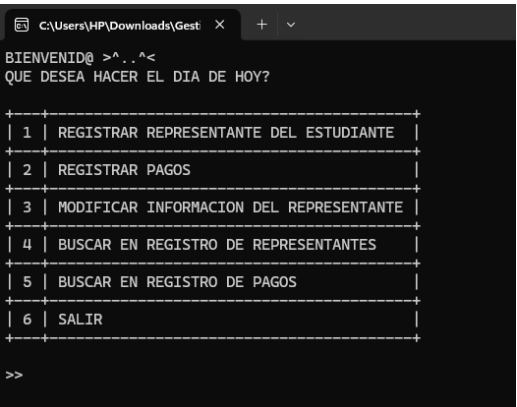
Partición de clases de equivalencia

VARIABLE	CLASE DE EQUIVALENCIA	ESTADO	REPRESENTANTE
usuario	C1: usuario== usuario	Valido	“blanca”
	C2: usuario! == usuario	No valido	“Blanca”
password	C1: password == password	Valido	“contra”
	C2: password! == password	No valido	“contraseña”

Validación de las credenciales



Interfaz del menú después de ingresar correctamente las credenciales



CÓDIGO FUENTE

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

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

        /* Entrada segura de contraseña */
        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 contraseña */

        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();
        }
    }

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

    return;
}
```

REQ 002 REGISTRAR PADRE DE FAMILIA

Historia de usuario	
Numero: REQ 002	Usuario: Administrador
Nombre de la Historia: Registrar Padre de Familia	
Prioridad: Alta	
Programador Responsable: Julio Sánchez	
Descripción:	
<ul style="list-style-type: none">Se registrará al padre de familia y al representado	
Validación:	
<ul style="list-style-type: none">Deberá ingresar el numero de identificación, nombre del padre , apellido, correo electrónico, numero de celular, dirección de calle principal y secundaria, nombre del hijo y sector donde vive.	

Partición de clases equivalentes

VALIDACION	CLASE DE EQUIVALENCIA	ESTADO	REPRESENTANTE
idRepresentante	C1: idRepresentante == idRepresentante	Valido	"1728073022"
	C2: idRepresentante! == idRepresentante	No valido	"-1728073022"
nombreRepresentante	C1: nombreRepresentante == nombreRepresentante	Valido	"Anthony"
	C2: nombreRepresentante! == nombreRepresentante	No valido	"El cuadro debe ser llenado"
apellidoRepresentante	C1: apellidoRepresentante == apellidoRepresentante	Valido	"Caisaguano"
	C2: apellidoRepresentante! == apellidoRepresentante	No valido	"El cuadro debe ser llenado"
correoElectronico	C1: correoElectronico == correoElectronico	Valido	"adeceu2006@gmail.com"
	C2: correoElectronico! == correoElectronico	No valido	"El cuadro debe ser llenado"
numeroCelular	C1: numeroCelular == numeroCelular	Valido	"0992547259"
	C2: numeroCelular! == numeroCelular	No valido	"El cuadro debe ser llenado"
direccionCP	C1: direccionCP == direccionCP	Valido	"Colorado"
	C2: direccionCP! == direccionCP	No valido	"El cuadro debe ser llenado"
direccionCS	C1: direccionCS == direccionCS	Valido	"Roca fuerte"
	C2: direccionCS! == direccionCS	No valido	"El cuadro debe ser llenado"
sector	C1: sector == sector	Valido	"Tandapi"
	C2: sector! == sector	No valido	"El cuadro debe ser llenado"
nombreHijo	C1: nombreHijo == nombreHijo	Valido	"Juanito"
	C2: nombreHijo! == nombreHijo	No valido	"El cuadro debe ser llenado"

Captura de la pantalla de la ejecución

```
>> 1
Realmente desea seleccionar esta opcion?
[1] SI || [2] NO
1
Ingrese el numero de cedula del representante:
1735729951
Ingrese el primer nombre del representante:
Julio
Ingrese el primer apellido del representante:
Andrade
Ingrese el correo electronico del representante:
jasudaudasda
Ingrese el numero de celular del representante:
0995855730
Ingrese la calle principal de domicilio:
mags
Ingrese la calle secundaria de domicilio:
bbb
Ingrese el sector del domicilio:
bbb
Ingrese el primer nombre del estudiante representado:
Juan
Se registro la informacion del representante exitosamente...|
```

En este apartado se vera como de be ingresar los datos para el registro del nuevo padre de familia y su hijo.

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

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

    infoRegistroRepresentante registroRepresentante;

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

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

    if(registroRepresentante.idRepresentante<0){
        puts("Ingrese un numero positivo");
        return;
    }
    while(getchar()!='\n');

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

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

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

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

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

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

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

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

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

    fclose(fptri);

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

    return;
}
```

REQ 003 REGISTRO DE LA PENCION

Historia de usuario	
Nombre: REQ003	Usuario: Administrador
Nombre de la Historia: Registrar la pensión de pago	
Prioridad: Alta	
Programador Responsable: Julio Sánchez	
Descripción: <ul style="list-style-type: none">Se registrará los pagos realizados de los padres de familia.	
Validación: <ul style="list-style-type: none">Debe ingresar el número de cedula, nombre y apellido del padre de familia, nombre del hijo, correo electrónico y numero de cedula.	

Partición de clases equivalentes

VARIABLES	CLASE DE EQUIVALENCIA	ESTADO	REPRESENTANTE
idReprsant	C1: idRprsant == idRprsant	Valido	"1728073022"
	C2: idRprsant! = idRprsant	No valido	"-1728073022"
nombreRprsant	C1: nombreRprsant == nombreRprsant	Valido	"Anthony"
	C2: nombreRprsant! = nombreRprsant	No valido	"El cuadro debe ser llenado"
apellidoRprsant	C1: apellidoRprsant == apellidoRprsant	Valido	"Caisaguano"
	C2: apellidoRprsant! = apellidoRprsant	No valido	"El cuadro debe ser llenado"
nombreHj	C1: nombreHj == nombreHj	Valido	"Juanito"
	C2: nombreHj! = nombre Hj	No valido	"El cuadro debe ser llenado"
pago	C1: pago == pago	Valido	"El monto que debe ingresar"
	C2: pago! = pago	No valido	"El cuadro debe ser llenado"

Captura de la pantalla de la ejecución

```
>> 2
Realmente desea seleccionar esta opcion?
[1] SI || [2] NO
1
Ingrese el numero de cedula del representante:
1753729951
Ingrese el primer nombre del representante: (Escriba todo en mayusculas o minusculas)
Julio
Ingrese el primer apellido del representante: (Escriba todo en mayusculas o minusculas)
Andrade
Ingrese el primer nombre del estudiante representado: (Escriba todo en mayusculas o minusculas)
bbbb
Ingrese el pago realizado:
582
Modo de pago:
[1] Efectivo || [2] Bancario
1
Se registro el pago exitosamente...
```

CÓDIGO FUENTE

```
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 funcion time y almacenamos en la variable 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 escribiremos en el archivo
    fprintf(fptr2, "\nID: %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;
}
```


REQ 004- BUSCAR EN EL REGISTRO DE REPRESENTANTE

Historia de usuario	
Nombre: REQ 004	Usuario: Administrador
Nombre de la Historia: Buscar en el Registro de Representante	
Prioridad: Alta	
Programador Responsable: Anthony Caisaguano	
Descripción: <ul style="list-style-type: none">El usuario podrá visualizar el historial de la información de los padres de familia y del representado.	
Validación: <ul style="list-style-type: none">Podrá visualizar la información ingresando el numero de cedula "ID" o el apellido	

Partición de clases equivalentes

VARIABLE	CLSE DE EQUIVALENCIA	ESTADO	REPRESNTANTE
criterio	C1:criterio=criterio	Valido	"1728073022"
	C2:criterio=criterio	Valido	"Caisaguano"
	C3:criterio!=criterio	No valido	"/*-"

Captura de pantalla de la ejecución con numero de cedula " ID"

```
C:\Users\HP\OneDrive\Escrito... x + v - □ x
[2] REGISTRAR PAGOS
[3] MODIFICAR INFORMACION DEL REPRESENTANTE
[4] BUSCAR EN REGISTRO DE REPRESENTANTES
[5] BUSCAR EN REGISTRO DE PAGOS
[6] LIMPIAR PANTALLA
[7] SALIR

>> 4
Realmente desea seleccionar esta opcion?
[1] SI || [2] NO
1
Ingresa el ID o Apellido del representante a buscar:
1728073022

Representante encontrado:
ID: 1728073022
Apellido: Caisaguano
Nombre: Anthony
Correo electronico: adeceu2006@gmail.com
Numero celular: 992547259
Nombre del representado: carlitos
Direccion del domicilio: aloag santodomingo_roac/*-/*-/+
```

Captura de pantalla de la ejecución con el apellido

```
C:\Users\HP\OneDrive\Escrito x + v
|-----|
|[2]| REGISTRAR PAGOS |
|-----|
|[3]| MODIFICAR INFORMACION DEL REPRESENTANTE |
|-----|
|[4]| BUSCAR EN REGISTRO DE REPRESENTANTES |
|-----|
|[5]| BUSCAR EN REGISTRO DE PAGOS |
|-----|
|[6]| LIMPIAR PANTALLA |
|-----|
|[7]| SALIR |
|-----|

>> 4
Realmente desea seleccionar esta opcion?
[1] SI || [2] NO
1
Ingresa el ID o Apellido del representante a buscar:
Caisaguano

Representante encontrado:
Apellido: Caisaguano
Nombre: Anthony
Correo electronico: adeceu2006@gmail.com
Numero celular: 992547259
Nombre del representado: carlitos
Direccion del domicilio: aloag santodomingo_roac/*_/*-/-+
|-----|
```

Captura de pantalla no valida

```
C:\Users\HP\OneDrive\Escrito x + v
BIENVENID@ >^..^<
QUE DESEA HACER EL DIA DE HOY?

|[1]| REGISTRAR REPRESENTANTE DEL ESTUDIANTE |
|-----|
|[2]| REGISTRAR PAGOS |
|-----|
|[3]| MODIFICAR INFORMACION DEL REPRESENTANTE |
|-----|
|[4]| BUSCAR EN REGISTRO DE REPRESENTANTES |
|-----|
|[5]| BUSCAR EN REGISTRO DE PAGOS |
|-----|
|[6]| LIMPIAR PANTALLA |
|-----|
|[7]| SALIR |
|-----|

>> 4
Realmente desea seleccionar esta opcion?
[1] SI || [2] NO
1
Ingresa el ID o Apellido del representante a buscar:
/**-+
No se encontro ningun representante con el criterio '/**-+'.
```

Código Fuente

```
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 línea[128];
    int encontrado = 0;

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

            do {
                printf("%s", línea);
                while (fgets(línea, 128, fptr) && strcmp(línea, "-----\n") != 0);
                printf("-----\n");
            } while (1);

        }
    }

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

    fclose(fptr);
}
```

REQ 005-BUSCAR EL REGISTRO DE PAGOS

Historia de usuario	
Nombre: REQ 005	Usuario: Administrador
Nombre de la Historia: Buscar el registro de pagos	
Prioridad: Alta	
Programador Responsable: Julio Andrade	
Descripción: <ul style="list-style-type: none">Historial que busca los pagos realizados por los padres de familia	
Validación: <ul style="list-style-type: none">Podrá visualizar el registro del pago realizado por el padre de familia mediante el numero de cedula “ID” o el apellido.	

Partición de clases equivalentes

VARIABLE	CLASES DE EQUIVALENCIA	ESTADO	REPRESENTANTE
criterio	C1: criterio=criterio	Valido	“1728073022”
	C2: criterio=criterio	Valido	“Caisaguano”
	C3: criterio!= criterio	No valido	“/*-+”
	C4: criterio!= criterio	No valido	“caisaguano”
	C5: criterio!= criterio	No valido	“-1728073022”

Captura de pantalla de ejecución (por el ID)

```
C:\Users\HP\OneDrive\Escrito  X  +  v

[[1]] REGISTRAR REPRESENTANTE DEL ESTUDIANTE
[[2]] REGISTRAR PAGOS
[[3]] MODIFICAR INFORMACION DEL REPRESENTANTE
[[4]] BUSCAR EN REGISTRO DE REPRESENTANTES
[[5]] BUSCAR EN REGISTRO DE PAGOS
[[6]] LIMPIAR PANTALLA
[[7]] SALIR

>> 5
Realmente desea seleccionar esta opcion?
[1] SI || [2] NO
1
Ingrese el ID o el apellido del Representante a buscar: 1728073022

Pago encontrado:
ID: 1728073022
Apellido: Caisaguano
Nombre: Anthony
Nombre del Representado: Juanito
Monto del pago efectuado: 140.00
Fecha y hora del registro de pago: 2025/02/19 a las 04:56 h
```

Captura de pantalla de ejecución (por el apellido)

```
C:\Users\HP\OneDrive\Escrito  X  +  v

[[1]] REGISTRAR REPRESENTANTE DEL ESTUDIANTE
[[2]] REGISTRAR PAGOS
[[3]] MODIFICAR INFORMACION DEL REPRESENTANTE
[[4]] BUSCAR EN REGISTRO DE REPRESENTANTES
[[5]] BUSCAR EN REGISTRO DE PAGOS
[[6]] LIMPIAR PANTALLA
[[7]] SALIR

>> 5
Realmente desea seleccionar esta opcion?
[1] SI || [2] NO
1
Ingrese el ID o el apellido del Representante a buscar: Caisaguano

Pago encontrado:
Apellido: Caisaguano
Nombre: Anthony
Nombre del Representado: Juanito
Monto del pago efectuado: 140.00
Fecha y hora del registro de pago: 2025/02/19 a las 04:56 h
```

Captura de pantalla de la ejecución (por /*-*)

```
C:\Users\HP\OneDrive\Escrito X + v
BIENVENID@ >^..^<
QUE DESEA HACER EL DIA DE HOY?

| [1] | REGISTRAR REPRESENTANTE DEL ESTUDIANTE |
|-----|
| [2] | REGISTRAR PAGOS |
|-----|
| [3] | MODIFICAR INFORMACION DEL REPRESENTANTE |
|-----|
| [4] | BUSCAR EN REGISTRO DE REPRESENTANTES |
|-----|
| [5] | BUSCAR EN REGISTRO DE PAGOS |
|-----|
| [6] | LIMPIAR PANTALLA |
|-----|
| [7] | SALIR |
|-----|

>> 5
Realmente desea seleccionar esta opcion?
[1] SI || [2] NO
1
Ingrese el ID o el apellido del Representante a buscar: /*-
No se encontro ningun pago con el criterio '/*-'.
```

Captura de pantalla de la ejecución (por el apellido mal escrito)

```
C:\Users\HP\OneDrive\Escrito X + v
BIENVENID@ >^..^<
QUE DESEA HACER EL DIA DE HOY?

[1] REGISTRAR REPRESENTANTE DEL ESTUDIANTE
[2] REGISTRAR PAGOS
[3] MODIFICAR INFORMACION DEL REPRESENTANTE
[4] BUSCAR EN REGISTRO DE REPRESENTANTES
[5] BUSCAR EN REGISTRO DE PAGOS
[6] LIMPIAR PANTALLA
[7] SALIR

>> 5
Realmente desea seleccionar esta opcion?
[1] SI || [2] NO
1
Ingrese el ID o el apellido del Representante a buscar: caisaguano
No se encontro ningun pago con el criterio 'caisaguano'.
```

Captura de pantalla de la ejecución (por el id con numero negativo)

```
C:\Users\HP\OneDrive\Escrito X + v
BIENVENID@ >^..^<
QUE DESEA HACER EL DIA DE HOY?

[1] REGISTRAR REPRESENTANTE DEL ESTUDIANTE
[2] REGISTRAR PAGOS
[3] MODIFICAR INFORMACION DEL REPRESENTANTE
[4] BUSCAR EN REGISTRO DE REPRESENTANTES
[5] BUSCAR EN REGISTRO DE PAGOS
[6] LIMPIAR PANTALLA
[7] SALIR

>> 5
Realmente desea seleccionar esta opcion?
[1] SI || [2] NO
1
Ingrese el ID o el apellido del Representante a buscar: -1728073022
No se encontro ningun pago con el criterio '-1728073022'.
```

CÓDIGO FUENTE

```
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;

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

            do {
                printf("%s", linea);
                while (fgets(linea, 128, fptr) && strcmp(linea, "-----\n") != 0);
                printf("-----\n");
            } while (1);
        }
    }

    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);
}
```

REQ 006- MODIFICAR LOS DATOS REGISTRADOS

Historia de usuario	
Nombre: REQ 006	Usuario: Administrador
Nombre de la Historia: Modificar los datos registrados	
Prioridad: Alta	
Programador Responsable: Julio Sánchez	
Descripción:	
<ul style="list-style-type: none"> Modificará la información de los padres de familia que haya ingresado. 	
Validación:	
<ul style="list-style-type: none"> Se podrá modificar la información personal del padre de familia 	

Partición de clases equivalentes

VARIABLE	CLASE DE EQUIVALENCIA	ESTADO	REPRESENTANTE
idRepresentante	C1: idRepresentante = idRepresentante	Valido	"1728032510"
nuevoNombre	C1: nuevoNombre = nuevoNombre	Valido	"Julio"
nuevoApellido	C1: nuevoApellido = nuevoApellido	Valido	"Andrade"
nuevoCE	C1: nuevoCE = nuevo CE	Valido	"adeceu500@gmail.com"
nuevoNC	C1: nuevoNC =nuevoNC	Valido	"0985361816"
nuevoCP	C1: nuevoCP = nuevoCP	Valido	"Villaflora"
nuevoCS	C1: nuevoCS = nuevoCS	Valido	"almeida"
nuevoSector	C1: nuevoSector = nuevoSector	valido	"Sangolqui"
nuevoNombreHijo	C1: nuevoNombreHijo =nuevoNombreHijo	Valido	"Carlitos"

Captura de pantalla de la ejecución

```
C:\Users\HP\OneDrive\Escrito x + - □ x
QUE DESEA HACER EL DIA DE HOY?

[1] REGISTRAR REPRESENTANTE DEL ESTUDIANTE
[2] REGISTRAR PAGOS
[3] MODIFICAR INFORMACION DEL REPRESENTANTE
[4] BUSCAR EN REGISTRO DE REPRESENTANTES
[5] BUSCAR EN REGISTRO DE PAGOS
[6] LIMPIAR PANTALLA
[7] SALIR

>> 3
Realmente desea seleccionar esta opcion?
[1] SI || [2] NO
1
Ingrese el ID del Representante que desea actualizar:
1728073022

Representante encontrado. Ingrese los nuevos datos:
Ingrese nuevamente el ID para confirmar
17280732510
Ingrese el nuevo nombre del representante:
Julio
Ingrese el nuevo apellido del representante:
Andrade
Ingrese el correo electronico del representante:
adeceu500@gmail.com
Ingrese el numero de celular del representante:
0985361816
Ingrese el nueva calle principal del domicilio:
Villaflores
Ingrese el nueva calle secundaria del domicilio:
almeida
Ingrese el nuevo sector del representante:
Sangolqui
Ingrese el nuevo nombre del representado:
Carlitos
La informacion del representante con ID '1728073022' ha sido actualizada.
BIENVENID@ >^..^<
QUE DESEA HACER EL DIA DE HOY?
```

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
343     while (getchar() != '\n');
344     while (fgets(linea, 128, fptr)) {
345         // Buscar la Linea que contiene el ID del producto
346         if (strstr(linea, idRepresentante) && strstr(linea, "ID: ")) {
347             encontrado = 1;
348
349
350             // Pedir nueva información al usuario
351             char nuevoNombre[32];
352             char nuevoApellido[32];
353             char nuevoCP[32];
354             char nuevoCE[32];
355             char nuevoCC[32];
356             int nuevoNC;
357             char nuevoSector[32];
358             char nuevoNombreHijo[32];
359             int idRepresentante;
360             while (getchar() != '\n');
361
362             puts("Representante encontrado. Ingrese los nuevos datos: ");
363             puts("Ingrese nuevamente el ID para confirmar");
364             scanf("%d", &idRepresentante);
365             while (getchar() != '\n');
366             puts("Ingrese el nuevo nombre del representante: ");
367             fgets(nuevoNombre, 32, stdin);
368             nuevoNombre[strcspn(nuevoNombre, "\n")] = 0;
369
370             printf("Ingrese el nuevo apellido del representante: \n");
371             fgets(nuevoApellido, 32, stdin);
372             nuevoApellido[strcspn(nuevoApellido, "\n")] = 0;
373
374             puts("Ingrese el correo electronico del representante: ");
375             fgets(nuevoCE, 32, stdin);
376             nuevoCE[strcspn(nuevoCE, "\n")] = 0;
377
378             puts("Ingrese el numero de celular del representante: ");
379             scanf("%d", &nuevoNC);
380
381             while(getchar() != '\n');
382
383             printf("Ingrese el nueva calle principal del domicilio: \n");
384             fgets(nuevoCP, 32, stdin);
385             nuevoCP[strcspn(nuevoCP, "\n")] = 0;
386
387             printf("Ingrese el nueva calle secundaria del domicilio: \n");
```



```

388     fgets(nuevoSector, 32, stdin);
389     nuevoCS[strlen(nuevoCS, "\n")] = 0;
390
391     printf("Ingrese el nuevo sector del representante: \n");
392     fgets(nuevoSector, 32, stdin);
393     nuevoSector[strlen(nuevoSector, "\n")] = 0;
394
395     printf("Ingrese el nuevo nombre del representante: \n");
396     fgets(nuevoNombreHijo, 32, stdin);
397     nuevoNombreHijo[strlen(nuevoNombreHijo, "\n")] = 0;
398
399     // Escribir la nueva información en fptr1
400     fprintf(fptr1, "ID: %d\n", idRepresentante);
401     fprintf(fptr1, "Apellido: %s\n", nuevoNombre);
402     fprintf(fptr1, "Nombre: %s\n", nuevoApellido);
403     fprintf(fptr1, "Correo electrónico: %s\n", nuevoCE);
404     fprintf(fptr1, "Número celular: %d\n", nuevoNC);
405     fprintf(fptr1, "Nombre del representante: %s\n", nuevoNombreHijo);
406     fprintf(fptr1, "Dirección del domicilio: %s %s %s\n", nuevoCP, nuevoCS, nuevoSector);
407     fprintf(fptr1, "-----\n");
408
409
410     while (fgets(linea, 128, fptr) &&
411            strcmp(linea, "-----\n") != 0);
412 } else {
413
414     fputs(linea, fptr1);
415 }
416 }
417
418 fclose(fptr);
419 fclose(fptr1);
420
421 if (encontrado) {
422     remove("DB\\listadoRepresentante\\listaRepresentantes1.txt");
423     rename("DB\\listadoRepresentante\\temp.txt", "DB\\listadoRepresentante\\listaRepresentantes1.txt");
424     printf("La información del representante con ID '%s' ha sido actualizada.\n", idRepresentante);
425 } else {
426     remove("DB\\listadoRepresentante\\temp.txt");
427     printf("No se encontró algún representante con el ID %s.\n", idRepresentante);
428 }
429 getch();
430 return;
431 }

```