

---

# PRÁCTICA 1

---

Metodología de la Programación



---

**Universidad de Valladolid**

26 DE JUNIO DE 2020

JUAN BLANCO MARTÍN

MIGUEL MARAZUELA BELLA

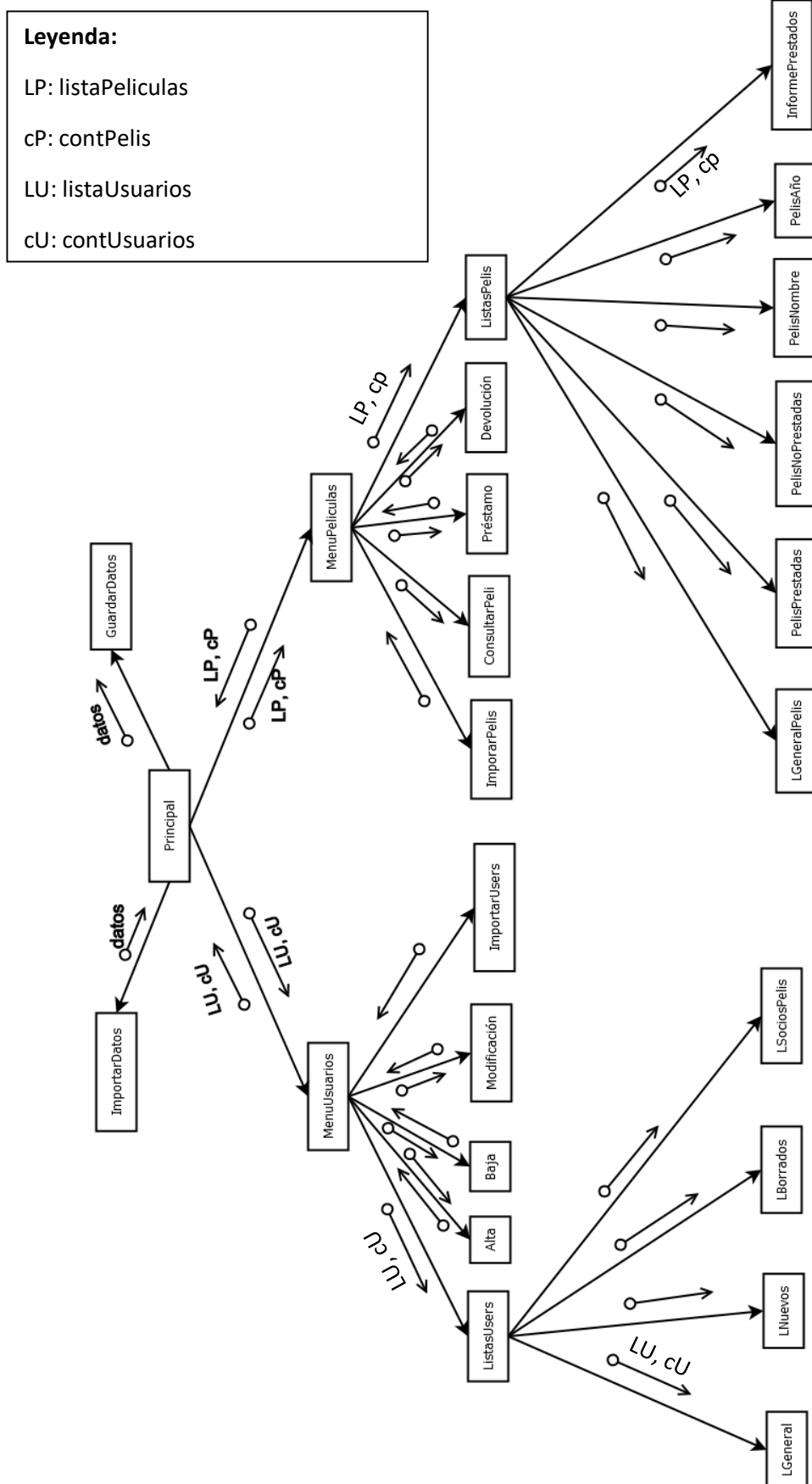
Laboratorio 3L - David Atauri - Grupo 49

# 1.- Objetivos

En esta primera práctica desarrollaremos un programa que gestione el préstamo de películas de un centro cultural durante un periodo de cinco años a los socios del mismo, para ello el programa estará dividido en: menú principal, y cuatro submenús (gestión de socios/películas y listado de socios/películas), los cuales se mostrarán de forma repetitiva al usuario.

Para ello utilizaremos programación modular y estructurada mediante la implementación de subprogramas, en la memoria principal utilizaremos arrays y registros, mientras que en la memoria secundaria trabajaremos con ficheros de distinto tipo (de texto o binario), para permitir la permanencia de los datos.

## 2.- Diagrama de descomposición modular



### 3.- Pruebas de ejecución

```
Menu Principal
-----
1-Gestion de Socios
2-Gestion de Peliculas
0-Salir y guardar

Introduzca una opcion:
```

```
Numero de socios: 0

Gestion de Socios
-----
1-Alta
2-Baja
3-Modificacion
4-Importar Socios
5-Listados de Socios

0-Volver

Introduzca una opcion:
```

```
ALTA DE SOCIOS
-----
DNI del usuario a dar de alta: 99988877x

Nombre: Jara Martinez

Fecha de nacimiento
    Dia: 7
    Mes: 5
    Anno: 1994

Direccion: caaa 56

Telefono: 111222333

Son correctos los datos? (S/N)S

Introduzca la fecha de hoy:
    Dia: 16
    Mes: 6
    Mes: 2020

El usuario ha sido dado de alta con el DNI: 99988877x

Pulse <Intro> para continuar...
```

# BAJA DE SOCIOS

-----

DNI del usuario a dar de baja: 99999999v

Seguro que desea borrar al usuario? (S/N): S

El usuario ha sido borrado.

Pulse <Intro> para continuar...

## MODIFICACION DE SOCIOS

-----

DNI del usuario que quiere modificar: 12422961v

- 1.- Nombre: Juan Blanco Martin
- 2.- Nacimiento: 30/4/1998
- 3.- Direccion: Caaa 56
- 4.- Telefono: 123456777

0.- Salir

Que desea modificar: 3

Nueva direccion: Calle C 56

Datos actualizados.

- 1.- Nombre: Juan Blanco Martin
- 2.- Nacimiento: 30/4/1998
- 3.- Direccion: Calle C 56
- 4.- Telefono: 123456777

0.- Salir

Que desea modificar:

Introduzca una opcion: 4

Socios importados: 4

Pulse <Intro> para continuar...

## LISTADOS DE USUARIOS

=====

- 1.- Listado general
- 2.- Listado de nuevos socios registrados
- 3.- Listado de socios que tienen películas prestadas
- 4.- Informe de de bajas de socios
- 0.- Volver

Que desea hacer?:

Numero de socios: 4

### LISTADO GENERAL

DNI	Nombre y apellidos	F. Nacimiento	Telefono
99988877x	Jara Martinez	7/5/1994	111222333
12422961v	Juan Blanco Martin	30/4/1998	123456777
00000001v	Jesus de Nazaret	25/12/1963	999888777
999888777	999888777	999888777/0/0	999888777

Pulse <Intro> para continuar...

DNI	Nombre y apellidos	Telefono	Fecha alta
99988877x	Jara Martinez	111222333	16/6/2020

Pulse <Intro> para continuar...

### LISTADO DE SOCIOS CON PELICULAS PRESTADAS

DNI	Nombre y apellidos	Titulo/s
99988877x	Jara Martinez	Mullholand Drive
12422961v	Juan Blanco Martin	Alta Fidelidad

Pulse <Intro> para continuar...

Total de usuarios borrados: 6

Pulse <Intro> para continuar...

```
Pelis registradas: 0

Gestion de Peliculas
-----
1-Importar peliculas
2-Mostrar los datos de una pelicula
3-Prestamo
4-Devolucion
5-Listados de peliculas
0-Volver

Introduzca una opcion:
```

```
Introduzca una opcion: 1

7 Peliculas importadas.

Pulse <Intro> para continuar...
```

```
Clave:          X123
Titulo:         Nace una cancion
Director:       El Cigala
Anno de estreno: 1963

Pulse <Intro> para continuar...
```

```
Peliculas disponibles
-----

Nace una cancion --> X123
Enter the void --> A111
Mullholand Drive --> M222
Inland Empire --> I666
Alta Fidelidad --> W699
Las Torres Gemelas (doc) --> L333
Shrek --> S888

Clave de la pelicula que quiere tomar prestada: A111
DNI del usuario: 12422961v

Mes en el que se esta prestando: 6
Anno en el que se esta prestando: 2020

Pelicula prestada :-)
```

```
Pulse <Intro> para continuar...
```

Clave de la pelicula que quiere devolver: A111

La pelicula \* Enter the void \* ha sido devuelta

Pulse <Intro> para continuar...

#### LISTADOS DE PELICULAS

=====

- 1.- Listado general
- 2.- Listado de peliculas no prestadas
- 3.- Listado de peliculas prestadas
- 4.- Listado de peliculas por titulo
- 5.- Listado de peliculas por anno
- 6.- Informe de prestamos
- 0.- Volver

Que desea hacer?:

#### LISTADO GENERAL

Clave	Titulo	Director	Anno	Prestado	Socio
X123	Nace una cancion	El Cigala	1963	NO	----
A111	Enter the void	Gaspar Noe	2013	NO	----
M222	Mullholand Drive	David Lynch	2001	SI	6405488
I666	Inland Empire	David Lynch	2005	NO	----
W699	Alta Fidelidad	Anonimo	2006	SI	6405656
L333	Las Torres Gemelas (doc)	Bin Laden	2001	NO	----
S888	Shrek	Andrew Adamson	2001	NO	----

Numero de peliculas: 7

Numero de peliculas prestadas: 2

Pulse <Intro> para continuar...

#### PELICULAS NO PRESTADAS

Clave	Titulo	Director	Anno
X123	Nace una cancion	El Cigala	1963
A111	Enter the void	Gaspar Noe	2013
I666	Inland Empire	David Lynch	2005
L333	Las Torres Gemelas (doc)	Bin Laden	2001
S888	Shrek	Andrew Adamson	2001

Pulse <Intro> para continuar...



```

      PELICULAS PRESTADAS
-----
Clave      Titulo      Socio
M222      Mullholand Drive      99988877x      6399720
W699      Alta Fidelidad      12422961v      6399720

Pulse <Intro> para continuar...
```

Anno de estreno que quiere buscar: 1963

```

      PELICULAS ESTRENADAS EN 1963
-----
Clave      Titulo      Director
X123      Nace una cancion      El Cigala

Pulse <Intro> para continuar...
```

Para probar la aplicación por primera vez, primero importar los usuarios y las películas desde las opciones 1.4 y 2.1 respectivamente. Esto se debe a que la versión enviada no contiene los ficheros binarios que permitirán la persistencia una vez se haya iniciado la aplicación por primera vez y se haya salido del programa correctamente (opción 0).

# Implementación

---

```

/*****
        Alumno 1: Juan Blanco Martín
        Alumno 2: Miguel Marazuela Bella
        Grupo: 49 - Turno: 3L
        Fecha: 26/06/2020
*****/

/* Compilar y ejecutar desde terminal *****/
/* Windows *****/
gcc main.c -o rec && rec.exe

* Linux *****/
gcc main.c -o rec
./rec

*****/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

#define MAXSOCIO 50
#define MAXPELI 200

/*****Estructuras de datos*****/

typedef struct
{
    char clave[5]; //X123
    char titulo[30];
    char director[30];
    int anno; // 1950 - 2100
    char DNIPrestado[10];
}tRegPelicula;

typedef struct
{
    int dia;
    int mes;
    int anno;
}regFecha;

typedef struct
{
    char dni[10];
    char nombre[50];
    int telefono;
    regFecha fecha;

```

```
    char direccion[30];
}tRegSocio;

/*****PROTOTIPOS*****/

//Menus
void MenuGestionSocios(tRegSocio listaSocios[MAXSOCIO], int *contSocios,
tRegPelicula peliculas[MAXPELI], int *contPelis);
void MenuGestionPelis(int *contPelis, tRegPelicula peliculas[MAXPELI], int
matrixPrestamos[5][12]);

//Utilidades
void Pausa();
int BuscarDNI(char dni[10], tRegSocio listaSocios[50]);
int BuscarClave(char clave[5], tRegPelicula peliculas[MAXPELI]);
int DiaHoy();
int MesHoy();
int AnnoHoy();

//Manejo de ficheros
void IniciarMatrizPrestamos(int matrixPrestamos[5][12]);
void IniciarUsuarios(tRegSocio listaSocios[MAXSOCIO], int *contSocios);
void IniciarPeliculas(tRegPelicula peliculas[MAXPELI], int *contPelis);
void GuardarUsuarios(tRegSocio listaSocios[MAXSOCIO], int contSocios);
void GuardarPelis(tRegPelicula peliculas[MAXPELI], int contPelis);
void GuardarMatrizPrestamos(int matrixPrestamos[5][12]);

//Usuarios
void AltaUsuario(tRegSocio listaSocios[MAXSOCIO], int *contSocios);
void BajaUsuario(tRegSocio listaSocios[MAXSOCIO], int *contSocios);
void ModificarUsuario(tRegSocio listaSocios[MAXSOCIO], int *contSocios);
void ImportarSocios(tRegSocio listaSocios[MAXSOCIO], int *contSocios);
void ListadoSocios(tRegSocio listaSocios[MAXSOCIO], int *contSocios, tRegPelicula
peliculas[MAXPELI], int contPelis);
void ListadoGeneralSocios(tRegSocio listaSocios[MAXSOCIO], int *contSocios);
void ListadoNuevosSocios();
void ListadoSociosPelisPrestadas(tRegSocio listaSocios[MAXSOCIO], int *contSocios,
tRegPelicula peliculas[MAXPELI], int *contPelis);
void InformeBajas();

//Peliculas
void ImportarPeliculas(int *contPelis, tRegPelicula peliculas[MAXPELI]);
void MostrarDatosPeli(tRegPelicula peliculas[MAXPELI]);
void Prestamo(int *contPelis, tRegPelicula peliculas[MAXPELI], int
matrixPrestamos[5][12]);
void Devolucion(tRegPelicula peliculas[MAXPELI]);
void ListadosPeliculas(int *contPelis, tRegPelicula peliculas[MAXPELI], int
matrixPrestamos[5][12]);
void ListadoGeneralPeliculas(int *contPelis, tRegPelicula peliculas[MAXPELI]);
void ListadoPelisNoPrestadas(int *contPelis, tRegPelicula peliculas[MAXPELI]);
void ListadoPelisPrestadas(int *contPelis, tRegPelicula peliculas[MAXPELI]);
void ListadoPelisNombre(int *contPelis, tRegPelicula peliculas[MAXPELI]);
void ListadoPelisAnno(int *contPelis, tRegPelicula peliculas[MAXPELI]);
void InformePrestamos(int matrixPrestamos[5][12]);
```

```
/* ***** */
/** PROGRAMA PRINCIPAL */
/* ***** */

int main()
{
    int opcion = 0;

    int contPelis = 0, contPrestamos = 0, contSocios = 0;
    int matrixPrestamos[5][12]; //matriz anno/mes

    tRegPelicula peliculas[MAXPELI];
    tRegSocio listaSocios[MAXSOCIO];

    IniciarMatrizPrestamos(matrixPrestamos);
    IniciarPeliculas(peliculas, &contPelis);
    IniciarUsuarios(listaSocios, &contSocios);

    do{
        system("cls");
        printf("\nMenu Principal\n");
        printf("-----\n\n");
        printf("1-Gestion de Socios\n");
        printf("2-Gestion de Peliculas\n");
        printf("0-Salir y guardar\n");
        printf("\n");
        printf("Introduzca una opcion: ");
        fflush(stdin);
        scanf("%d",&opcion);

        switch(opcion){

            case 1:
                MenuGestionSocios(listaSocios, &contSocios, peliculas,
&contPelis);
                break;

            case 2:
                MenuGestionPelis(&contPelis, peliculas, matrixPrestamos);
                break;

            case 0:
                GuardarMatrixPrestamos(matrixPrestamos);
                GuardarPelis(peliculas, contPelis);
                GuardarUsuarios(listaSocios, contSocios);
                printf("\nDatos guardados");
                Pausa();
                break;

        }

    } while(opcion != 0);

    return 0;
}
```

```

}

/*****
*****/
/* Subprograma: MenuGestionSocios */
/* Tipo: Procedimiento (función tipo void) */
/* Prerrequisitos: No tiene. */
/* Objetivo: Permite navegar por las opciones del menu */
/*****
*****/
void MenuGestionSocios(tRegSocio listaSocios[MAXSOCIO], int *contSocios,
tRegPelicula peliculas[MAXPELI], int *contPelis) {

    int opcion;

    do{
        system("cls");
        printf("Numero de socios: %d\n", *contSocios);
        printf("\nGestion de Socios\n");
        printf("-----\n");
        printf("1-Alta\n");
        printf("2-Baja\n");
        printf("3-Modificacion\n");
        printf("4-Importar Socios\n");
        printf("5-Listados de Socios\n\n");
        printf("0-Volver\n");
        printf("\nIntroduzca una opcion: ");
        fflush(stdin);
        scanf("%d",&opcion);

        switch (opcion) {
            case 1:
                AltaUsuario(listaSocios, &*contSocios); //DONE
                break;
            case 2:
                BajaUsuario(listaSocios, &*contSocios);
                break;
            case 3:
                ModificarUsuario(listaSocios, &*contSocios); //DONE
                break;
            case 4:
                ImportarSocios(listaSocios, &*contSocios); //done
                break;
            case 5:
                ListadoSocios(listaSocios, &*contSocios, peliculas, &*contPelis);
                break;
            case 0:
                break;
        }

    } while (opcion != 0);
}

```

```

/*****
*****/
/* Subprograma: MenuGestionPelis */
/* Tipo: Procedimiento (función tipo void) */
/* Prerrequisitos: No tiene. */
/* Objetivo: Permite navegar por las opciones del menu */
/*****
*****/
void MenuGestionPelis(int *contPelis, tRegPelicula peliculas[MAXPELI], int
matrixPrestamos[5][12]) {

    int opcion;
    do {
        system("cls");
        printf("Pelis registradas: %d\n", *contPelis);
        printf("\nGestion de Peliculas\n");
        printf("-----\n");
        printf("1-Importar peliculas\n");
        printf("2-Mostrar los datos de una pelicula\n");
        printf("3-Prestamo\n");
        printf("4-Devolucion\n");
        printf("5-Listados de peliculas\n");
        printf("0-Volver\n");
        printf("\nIntroduzca una opcion: ");
        fflush(stdin);
        scanf("%d",&opcion);

        switch (opcion) {
            case 1:
                ImportarPeliculas(&*contPelis, peliculas);
                break;
            case 2:
                MostrarDatosPeli(peliculas);
                break;
            case 3:
                Prestamo(&*contPelis, peliculas, matrixPrestamos);
                break;
            case 4:
                Devolucion(peliculas);
                break;
            case 5:
                ListadosPeliculas(&*contPelis, peliculas, matrixPrestamos);
                break;
            case 0:
                break;
        }

    } while (opcion != 0);

}

/*****
*****/

```

```

/* Subprograma: Pausa */
/* Tipo: Procedimiento (función tipo void) */
/* Prerrequisitos: No tiene. */
/* Objetivo: Espera a que el usuario pulse <Intro> para continuar el
procedimiento. */
/*****
*****/
void Pausa() {

    printf("\n\nPulse <Intro> para continuar...");
    fflush(stdin);
    getchar();
}

/*****
*****/
/* Subprograma: BuscarDni */
/* Tipo: int */
/* Prerrequisitos: No tiene. */
/* Objetivo: Devuelve la posicion de un usuario con el DNI pasado o
-1 si no encuentra al suario */
/*****
*****/
int BuscarDNI(char dni[10], tRegSocio listaSocios[MAXSOCIO]){

    int pos;
    for (pos = 0; pos<MAXSOCIO; pos++){
        if (strcasecmp(dni, listaSocios[pos].dni) == 0){
            return pos;
        }
    }
    return -1;
}

/*****
*****/
/* Subprograma: BuscarClave */
/* Tipo: int */
/* Prerrequisitos: No tiene. */
/* Objetivo: Devuelve la posicion de una pelicula con la clave pasada o
-1 si no encuentra al suario */
/*****
*****/
int BuscarClave(char clave[5], tRegPelicula peliculas[MAXPELI]) {

    int pos;
    for(pos = 0; pos < MAXPELI; pos++){
        if(strcmp(peliculas[pos].clave, clave)==0)
            return pos;
    }
    return -1;
}

/*****
*****/

```

```
*****/
/* Subprograma: DiaHoy */
/* Tipo: int */
/* Prerrequisitos: No tiene. */
/* Objetivo: Pide al usuario el valor del dia */
/*****
*****/
int DiaHoy() {

    int dia;

    printf("\n\nIntroduzca la fecha de hoy: \n\tDia: ");
    fflush(stdin);
    scanf("%d", &dia);

    return dia;
}

/*****
*****/
/* Subprograma: DiaMes */
/* Tipo: int */
/* Prerrequisitos: No tiene. */
/* Objetivo: Pide al usuario el valor del mes */
/*****
*****/
int MesHoy() {

    int mes;

    printf("\tMes: ");
    fflush(stdin);
    scanf("%d", &mes);

    return mes;
}

/*****
*****/
/* Subprograma: DiaAnno */
/* Tipo: int */
/* Prerrequisitos: No tiene. */
/* Objetivo: Pide al usuario el valor del anno */
/*****
*****/
int AnnoHoy() {

    int anno;

    printf("\tMes: ");
    fflush(stdin);
    scanf("%d", &anno);

    return anno;
}
```



```

}

/*****
*****/
/* Subprograma: IniciarMatrizPrestamos */
/* Tipo: Procedimiento (función tipo void) */
/* Prerrequisitos: No tiene. */
/* Objetivo: Inicializa la matriz de prestamos */
/*****
*****/
void IniciarMatrizPrestamos(int matrixPrestamos[5][12]) {

    int i = 0;
    int j = 0;

    FILE* HcoPrestamos;
    HcoPrestamos = fopen("HcoPrestamos.dat", "rb");

    if (HcoPrestamos != NULL) {
        fread(&matrixPrestamos, (sizeof(int)*5*12), 1, HcoPrestamos);
    } else {
        HcoPrestamos = fopen("HcoPrestamos.dat", "wb");
        for(i = 0; i<5; i++) {
            for(j = 0; j<12; j++) {
                matrixPrestamos[i][j] = 0;
            }
        }

        fwrite(&matrixPrestamos, (sizeof(char)*12*31*2*12), 1, HcoPrestamos);
    }
    fclose(HcoPrestamos);
}

/*****
*****/
/* Subprograma: IniciarUsuarios */
/* Tipo: Procedimiento (función tipo void) */
/* Prerrequisitos: No tiene. */
/* Objetivo: Inicializa la lista de usuarios guardados en el fichero binario */
/*****
*****/
void IniciarUsuarios(tRegSocio listaSocios[MAXSOCIO], int *contSocios) {

    int i = 0;

    FILE* Socios;
    Socios = fopen("Socios.dat", "rb");

    if (Socios == NULL) {
        Socios = fopen("Socios.dat", "wb");
    } else {
        while(!feof(Socios)) {
            fread(&listaSocios[i], sizeof(tRegSocio), 1, Socios);

```

```

        i++;
        *contSocios = *contSocios +1;
    }
}

fclose(Socios);
}

/*****
*****/
/* Subprograma: IniciarPelículas */
/* Tipo: Procedimiento (función tipo void) */
/* Prerrequisitos: No tiene. */
/* Objetivo: Inicializa la lista de películas guardadas en el fichero binario */
/*****
*****/
void IniciarPelículas(tRegPelícula películas[MAXPELI], int *contPelis) {

    int i = 0;

    FILE* Pelis;
    Pelis = fopen("Pelis.dat", "rb");

    if (Pelis == NULL) {
        Pelis = fopen("Pelis.dat", "wb");
    } else {
        while(!feof(Pelis)) {
            fread(&películas[i], sizeof(tRegSocio), 1, Pelis);
            i++;
            *contPelis = *contPelis +1;
        }
    }
    fclose(Pelis);
}

/*****
*****/
/* Subprograma: GuardarUsuarios */
/* Tipo: Procedimiento (función tipo void) */
/* Prerrequisitos: No tiene. */
/* Objetivo: Guarda la lista de usuarios en el fichero binario */
/*****
*****/
void GuardarUsuarios(tRegSocio listaSocios[MAXSOCIO], int contSocios){

    int i;
    FILE* Socios;
    Socios = fopen("Socios.dat", "wb");

    if(Socios == NULL) {
        printf("No se ha podido crear el fichero de guardado...");
        Pausa();
        exit(1);
    }
}

```

```

    for(i = 0; i < contSocios; i++){
        fwrite(&listaSocios[i], sizeof(tRegSocio), 1, Socios);
    }

    fclose(Socios);
}

/*****
*****/
/* Subprograma: GuardarPelis */
/* Tipo: Procedimiento (función tipo void) */
/* Prerrequisitos: No tiene. */
/* Objetivo: Guarda la lista de peliculas en el fichero binario */
/*****
*****/
void GuardarPelis(tRegPelicula peliculas[MAXPELI], int contPelis) {

    int i;
    FILE* Pelis;
    Pelis = fopen("Pelis.dat", "wb");

    if(Pelis == NULL) {
        printf("No se ha podido crear el fichero de guardado...");
        Pausa();
        exit(1);
    }
    for(i = 0; i < contPelis; i++){
        fwrite(&peliculas[i], sizeof(tRegPelicula), 1, Pelis);
    }
    fclose(Pelis);
}

/*****
*****/
/* Subprograma: GuardarMatrixPrestamos */
/* Tipo: Procedimiento (función tipo void) */
/* Prerrequisitos: No tiene. */
/* Objetivo: Guarda los prestamos almacenados en la matriz de prestamos */
/*****
*****/
void GuardarMatrixPrestamos(int matrixPrestamos[5][12]) {

    int i, j;
    FILE* HcoPrest;
    HcoPrest = fopen("HcoPrest.dat", "wb");

    if(HcoPrest == NULL) {
        printf("No se ha podido crear el fichero de guardado...");
        Pausa();
        exit(1);
    }

    for(i = 0; i < 5; i++) {
        for(j = 0; j < 12; j++) {

```

```

        fwrite(&matrixPrestamos[i][j], sizeof(int)*5*12, 1, HcoPrest);
    }
}
fclose(HcoPrest);
}

/*****
*****/
/* Subprograma: AltaUsuario */
/* Tipo: Procedimiento (función tipo void) */
/* Prerrequisitos: No tiene. */
/* Objetivo: Permite al usuario registrar nuevos socios */
/*****
*****/
void AltaUsuario(tRegSocio listaSocios[MAXSOCIO], int *contSocios) {

    tRegSocio aux;
    char DNI[10];
    char confirmacion;
    FILE* NuevoSoc;

    if(*contSocios < 50){
        system("cls");
        printf("ALTA DE SOCIOS\n");
        printf("-----\n");

        printf("\nDNI del usuario a dar de alta: ");
        fflush(stdin);
        scanf("%s", &DNI);
        if(BuscarDNI(DNI, listaSocios) == -1){
            strcpy(aux.dni, DNI);
            printf("\nNombre: ");
            fflush(stdin);
            scanf("%[^\n]s", &aux.nombre);

            do{
                printf("\nFecha de nacimiento \n\tDia: ");
                fflush(stdin);
                scanf("%d", &aux.fecha.dia);
                printf("\tMes: ");
                fflush(stdin);
                scanf("%d", &aux.fecha.mes);
                printf("\tAnno: ");
                fflush(stdin);
                scanf("%d", &aux.fecha.anno);
            } while((aux.fecha.dia < 1) || (aux.fecha.dia > 31) || (aux.fecha.mes
< 1) || (aux.fecha.mes > 12) || (aux.fecha.anno < 1900) || (aux.fecha.anno >
2019));

            printf("\nDireccion: ");
            fflush(stdin);
            scanf("%[^\n]s", &aux.direccion);

```

```

printf("\nTelefono: ");
fflush(stdin);
scanf("%d", &aux.telefono);

printf("\n\nSon correctos los datos? (S/N)");
fflush(stdin);
scanf("%c", &confirmacion);
do{
    if(confirmacion == 'S'){
        listaSocios[*contSocios] = aux;

        NuevoSoc = fopen("NuevoSoc.txt","a");
        if(NuevoSoc != NULL){
            fprintf(NuevoSoc, "%s*",aux.dni);
            fprintf(NuevoSoc, "%s*",aux.nombre);
            fprintf(NuevoSoc, "%d*",aux.telefono);
            fprintf(NuevoSoc, "%d/",DiaHoy());
            fprintf(NuevoSoc, "%d/",MesHoy());
            fprintf(NuevoSoc, "%d*\n",AnnoHoy());

        } else {
            NuevoSoc = fopen("NuevoSoc.txt","w"); //Si no existe el
fichero lo crea

            fprintf(NuevoSoc, "%s*",aux.dni);
            fprintf(NuevoSoc, "%s*",aux.nombre);
            fprintf(NuevoSoc, "%d*",aux.telefono);
            fprintf(NuevoSoc, "%d/",DiaHoy());
            fprintf(NuevoSoc, "%d/",MesHoy());
            fprintf(NuevoSoc, "%d*\n",AnnoHoy());
        }

        fclose(NuevoSoc);

        printf("\nEl usuario ha sido dado de alta con el DNI:
%s",aux.dni);

        *contSocios = *contSocios + 1;

    } else printf("\nEl usuario no ha sido dado de alta.");

} while ((confirmacion != 'S') && (confirmacion != 'N'));

} else printf("\nYa existe un usuario con ese DNI...");

} else ("ERROR: No se pueden dar de alta a mas de 50 usuarios.");

Pausa();
}

/*****
/* Subprograma: BajaUsuario */
/* Tipo: Procedimiento (función tipo void) */

```

```

/* Prerrequisitos: No tiene. */
/* Objetivo: Permite al usuario borrar socios */
/*****
*****/
void BajaUsuario(tRegSocio listaSocios[MAXSOCIO], int *contSocios) {

    char DNI[10], confirmacion;
    int pos, i;

    FILE* HcoSoc;
    HcoSoc = fopen("HcoSoc.dat", "ab");

    system("cls");
    if(*contSocios != 0) {
        system("cls");
        printf("BAJA DE SOCIOS\n");
        printf("-----\n");

        printf("\nDNI del usuario a dar de baja: ");
        fflush(stdin);
        scanf("%s", &DNI);
        pos = BuscarDNI(DNI, listaSocios);
        if( pos != -1) {
            printf("\n\nSeguro que desea borrar al usuario? (S/N): ");
            fflush(stdin);
            scanf("%c", &confirmacion);

            do {
                if(confirmacion == 'S') {

                    fwrite(&listaSocios[pos], sizeof(tRegSocio), 1, HcoSoc);

                    if(HcoSoc == NULL) {
                        HcoSoc = ("HcoSoc.dat", "wb"); //Por si no existe crearlo
                        fwrite(&listaSocios[pos], sizeof(tRegSocio), 1, HcoSoc);
                    }

                    fclose(HcoSoc);

                    for(i = pos; i < MAXSOCIO; i++){
                        listaSocios[i] = listaSocios[i+1]; //COMPROBAR
                    }

                    printf("\nEl usuario ha sido borrado.");

                    *contSocios = *contSocios - 1;

                } else printf("\nEl usuario NO ha sido borrado.");

            } while ((confirmacion != 'S') && (confirmacion != 'N'));

            } else printf("\nNo existe un usuario con ese DNI.");

        } else printf("\nNo hay ningun usuario registrado.");
    }

```

```

    Pausa();
}

/*****
*****/
/* Subprograma: ModificarUsuario */
/* Tipo: Procedimiento (función tipo void) */
/* Prerrequisitos: No tiene. */
/* Objetivo: Permite al usuario modificar un socio */
/*****
*****/
void ModificarUsuario(tRegSocio listaSocios[MAXSOCIO], int *contSocios) {

    char DNI[10];
    int pos, opcion;

    system("cls");
    if(*contSocios != 0){
        system("cls");
        printf("MODIFICACION DE SOCIOS\n");
        printf("-----\n");
        printf("\nDNI del usuario que quiere modificar: ");
        fflush(stdin);
        scanf("%s", &DNI);
        pos = BuscarDNI(DNI, listaSocios);
        if( pos != -1) {
            do{
                printf("\n1.- Nombre:      %s",listaSocios[pos].nombre);
                printf("\n2.- Nacimiento:   %d/%d/%d",listaSocios[pos].fecha.dia,
listaSocios[pos].fecha.mes, listaSocios[pos].fecha.anno);
                printf("\n3.- Direccion:    %s",listaSocios[pos].direccion);
                printf("\n4.- Telefono:     %d",listaSocios[pos].telefono);
                printf("\n\n0.- Salir");
                printf("\n\nQue desea modificar: ");

                fflush(stdin);
                scanf("%d",&opcion);

                switch (opcion) {
                    case 1:
                        printf("\nNuevo nombre: ");
                        fflush(stdin);
                        scanf("%[^\\n]s", &listaSocios[pos].nombre);
                        printf("\nDatos actualizados.");
                        break;

                    case 2:
                        do{
                            printf("\nNueva fecha de nacimiento: \\n\\tDia:  ");
                            fflush(stdin);
                            scanf("%d", &listaSocios[pos].fecha.dia);
                            printf("\\tMes:  ");
                            fflush(stdin);

```

```

        scanf("%d", &listaSocios[pos].fecha.mes);
        printf("\tAnno: ");
        fflush(stdin);
        scanf("%d", &listaSocios[pos].fecha.anno);
    } while((listaSocios[pos].fecha.dia < 1) ||
(listaSocios[pos].fecha.dia > 31) || (listaSocios[pos].fecha.mes < 1) ||
(listaSocios[pos].fecha.mes > 12) || (listaSocios[pos].fecha.anno < 1900) ||
(listaSocios[pos].fecha.anno > 2019));

    printf("\nDatos actualizados.");
    break;

    case 3:
        printf("\nNueva direccion: ");
        fflush(stdin);
        scanf("%[^\n]s", &listaSocios[pos].direccion);
        printf("\nDatos actualizados.");
        break;

    case 4:
        printf("\nNuevo telefono: ");
        fflush(stdin);
        scanf("%d", &listaSocios[pos].telefono);
        printf("\nDatos actualizados.");
        break;

    case 0:
        break;
}

} while (opcion != 0);

} else printf("\nNo existe un usuario con ese DNI.");

} else printf("No hay usuarios en la plataforma.");

Pausa();
}

/*****
*****/
/* Subprograma: ImportarSocios */
/* Tipo: Procedimiento (función tipo void) */
/* Prerrequisitos: No tiene. */
/* Objetivo: Importa usuarios desde el fichero Socios.txt */
/*****
*****/
void ImportarSocios(tRegSocio listaSocios[MAXSOCIO], int *contSocios) {

    char linea[50];
    int contImportados = 0;

    char *token;

```



```
FILE* Socios;
Socios = fopen("Socios.txt", "r");

if(*contSocios != MAXSOCIO) {
    if(Socios != NULL) {

        while(!feof(Socios)){
            if(*contSocios < 50) {
                fgets(linea, 50, Socios);
                token = strtok(linea, "\n");
                if(BuscarDNI(token, listaSocios) == -1){
                    strcpy(listaSocios[*contSocios].dni, token);
                } else {
                    printf("ERROR: Ya hay un usuario con DNI: %s. Por favor
revisa el fichero Socios.txt", token);
                    Pausa();
                    return 0;
                }

                fgets(linea, 50, Socios);
                token = strtok(linea, "\n");
                strcpy(listaSocios[*contSocios].nombre, token);

                fgets(linea, 50, Socios);
                token = strtok(linea, "/");
                listaSocios[*contSocios].fecha.dia = atoi(token);
                token = strtok(NULL, "/");
                listaSocios[*contSocios].fecha.mes = atoi(token);
                token = strtok(NULL, "\n");
                listaSocios[*contSocios].fecha.anno = atoi(token);

                fgets(linea, 50, Socios);
                token = strtok(linea, "\n");
                strcpy(listaSocios[*contSocios].direccion, token);

                fgets(linea, 50, Socios);
                listaSocios[*contSocios].telefono = atoi(linea);

                *contSocios = *contSocios + 1;
                contImportados++;

            } else printf("\nYa se ha alcanzado el maximo de usuarios
registrados.");
        }

        printf("\nSocios importados: %d", contImportados);
        fclose(Socios);

    } else printf("No se ha podido abrir el fichero Socios.txt");

} else printf("\nERROR: No se pueden registrar mas de 50 usuarios");

Pausa();
}
```

```

/*****
*****/
/* Subprograma: ListadoSocios */
/* Tipo: Procedimiento (función tipo void) */
/* Prerrequisitos: No tiene. */
/* Objetivo: Menu para navegar por los diferentes listados */
/*****
*****/
void ListadoSocios(tRegSocio listaSocios[MAXSOCIO], int *contSocios, tRegPelicula
peliculas[MAXPELI], int contPelis) {

    int opcion;

    do{
        system("cls");
        printf("LISTADOS DE USUARIOS \n");
        printf("=====\n");
        printf("\n1.- Listado general");
        printf("\n2.- Listado de nuevos socios registrados");
        printf("\n3.- Listado de socios que tienen películas prestadas");
        printf("\n4.- Informe de de bajas de socios");
        printf("\n0.- Volver");
        printf("\n\nQue desea hacer?: ");
        fflush(stdin);
        scanf("%d", &opcion);

        switch(opcion){
            case 1:
                ListadoGeneralSocios(listaSocios, &contSocios); //done
                break;
            case 2:
                ListadoNuevosSocios(); //done
                break;
            case 3:
                ListadoSociosPelisPrestadas(listaSocios, &contSocios, peliculas,
contPelis);
                break;
            case 4:
                InformeBajas();
                break;
            case 0:
                break;
        }

    }while(opcion != 0);
}

/*****
*****/
/* Subprograma: ListadoGeneralSocios */
/* Tipo: Procedimiento (función tipo void) */
/* Prerrequisitos: No tiene. */

```

```

/* Objetivo: Muestra todos los socios */
/*****/
void ListadoGeneralSocios(tRegSocio listaSocios[MAXSOCIO], int *contSocios) {

    int i;

    system("cls");
    printf("Numero de socios: %d\n", *contSocios);
    printf("                                LISTADO GENERAL");
    printf("\n DNI          Nombre y apellidos          F. Nacimiento
Telefono");
    printf("\n-----");
    printf("\n-----");

    for(i = 0; i < *contSocios; i++){
        printf("\n%s    %s    \t\t%d/%d/%d    \t%d", listaSocios[i].dni,
listaSocios[i].nombre, listaSocios[i].fecha.dia, listaSocios[i].fecha.mes,
listaSocios[i].fecha.anno, listaSocios[i].telefono);
    }
    Pausa();
}

/*****/
/* Subprograma: ListadoGeneralSocios */
/* Tipo: Procedimiento (función tipo void) */
/* Prerrequisitos: No tiene. */
/* Objetivo: Muestra todos los socios dados de alta desde la aplicacion */
/*****/
void ListadoNuevosSocios() {

    char linea[1024], DNI, nombre, fecha;
    int telefono;
    char* token;

    FILE* NuevoSoc;
    NuevoSoc = fopen("NuevoSoc.txt", "r");

    system("cls");
    printf("                                LISTADO NUEVOS SOCIOS\n");
    printf("\n DNI          Nombre y apellidos          Telefono          Fecha
alta");
    printf("\n-----\n");

    if(NuevoSoc != NULL){
        while(fgets(linea, 1024, NuevoSoc)){
            //Leer clave
            token = strtok(linea, "*");
            printf("\n%s", token);

            //Leer nombre

```

```

        token = strtok(NULL, "*");
        printf("\t%s", token);

        //Leer telefono
        token = strtok(NULL, "*");
        printf("\t\t\t%d", atoi(token));

        //Leer fecha
        token = strtok(NULL, "/");
        printf("\t%d/", atoi(token));
        token = strtok(NULL, "/");
        printf("%d/", atoi(token));
        token = strtok(NULL, "*");
        printf("%d", atoi(token));

    }

} else printf("ERROR: No se ha podido abrir el fichero NuevoSoc.txt");

Pausa();
}

/*****
*****/
/* Subprograma: ListadoSociosPelisPrestadas */
/* Tipo: Procedimiento (función tipo void) */
/* Prerrequisitos: No tiene. */
/* Objetivo: Muestra las peliculas que tiene presatas cada socio, si tiene alguna
*/
/*****
*****/
void ListadoSociosPelisPrestadas(tRegSocio listaSocios[MAXSOCIO], int *contSocios,
tRegPelicula peliculas[MAXPELI], int *contPelis) {

    int i, j;

    system("cls");
    printf("          LISTADO DE SOCIOS CON PELICULAS PRESTADAS");
    printf("\n-----");
    printf("\n DNI          Nombre y apellidos          Titulo/s");

    for(i = 0; i < *contSocios; i++){
        for(j = 0; j < *contPelis; j++){
            if(strcmp(listaSocios[i].dni, peliculas[j].DNIPrestado) == 0) {
                printf("\n%s   %s   %s", listaSocios[i].dni,
listaSocios[i].nombre, peliculas[j].titulo);
            }
        }
    }
    Pausa();
}

/*****
*****/

```

```

/* Subprograma: InformeBajas */
/* Tipo: Procedimiento (función tipo void) */
/* Prerrequisitos: No tiene. */
/* Objetivo: Muestra el numero de bajas que ha habido desde que existe la
aplicacion */
/*****
*****/
void InformeBajas() {

    tRegSocio sociosBorrados[MAXSOCIO];
    int i, cont = 0, anno;

    FILE* HcoSoc;
    HcoSoc = fopen("HcoSoc.dat", "rb");

    system("cls");

    if (HcoSoc != NULL) {
        while(!feof(HcoSoc)){
            fread(&sociosBorrados[cont], sizeof(tRegSocio), 1, HcoSoc);
            cont++;
        }
        printf("Total de usuarios borrados: %d", cont);

    } else printf("\nNo se pudo encontrar el fichero HcoSoc.dat");
    fclose(HcoSoc);
    Pausa();
}

/*****
*****/
/* Subprograma: ImportarPeliculas */
/* Tipo: Procedimiento (función tipo void) */
/* Prerrequisitos: No tiene. */
/* Objetivo: Importa als peliculas que se encuentran el fichero Pelis.txt */
/*****
*****/
void ImportarPeliculas(int *contPelis, tRegPelicula peliculas[MAXPELI]) {

    char linea[30];
    char *token;
    int cont = 0;

    FILE* Pelis;
    Pelis = fopen("Pelis.txt", "r");

    if(Pelis != NULL) {

        while(!feof(Pelis)) {
            if(*contPelis < 200) {

                fgets(linea, 50, Pelis);
                token = strtok(linea, "\n");
            }
        }
    }
}

```

```

        if(BuscarClave(token, peliculas) == -1){
            strcpy(peliculas[*contPelis].clave, token);
        } else {
            printf ("ERROR: Pelicula con clave %s repetida. \nElimine la
pelicula del fichero y vuelva a importarlas.");
            Pausa();
            return 0;
        }

        fgets(linea, 50, Pelis);
        token = strtok(linea, "\n");
        strcpy(peliculas[*contPelis].titulo, token);

        fgets(linea, 50, Pelis);
        token = strtok(linea, "\n");
        strcpy(peliculas[*contPelis].director, token);

        fgets(linea, 50, Pelis);
        peliculas[*contPelis].anno = atoi(linea);

        peliculas[*contPelis].DNIPrestado[0] = '-'; //Al no estar prestada
el DNIPrestado se inicializa como <->
        *contPelis = *contPelis + 1;
        cont++;

    } else printf("\nImposible registrar mas de 200 peliculas");
}

fclose(Pelis);

} else {
    printf("ERROR: No se ha podido acceder al fichero de peliculas...");
    exit(0);
}
printf("\n%d Peliculas importadas.\n", cont);
Pausa();
}

/*****
*****/
/* Subprograma: MostrarDatosPeli */
/* Tipo: Procedimiento (función tipo void) */
/* Prerrequisitos: No tiene. */
/* Objetivo: Muestra los datos de la pelicula con la clave pasada */
/*****
*****/
void MostrarDatosPeli(tRegPelicula peliculas[MAXPELI]) {

    int pos = 0;
    char clave[5];

    system("cls");
    printf("MOSTRAR DATOS DE PELICULA\n");

```

```

printf("-----\n");
printf("\nIntroduzca una clave de pelicula: ");
fflush(stdin);
scanf("%s",&clave);

pos = BuscarClave(clave, peliculas);
if(pos != -1) {
    system("cls");
    printf("\nClave:          %s",clave);
    printf("\nTitulo:           %s",peliculas[pos].titulo);
    printf("\nDirector:          %s",peliculas[pos].director);
    printf("\nAnno de estreno:  %d",peliculas[pos].anno);
} else printf("\nERROR: Pelicula no encontrada");

Pausa();
}

/*****
*****/
/* Subprograma: Prestamo */
/* Tipo: Procedimiento (función tipo void) */
/* Prerrequisitos: No tiene. */
/* Objetivo: Permite realizar un prestamo */
/*****
*****/
void Prestamo(int *contPelis, tRegPelicula peliculas[MAXPELI], int
matrixPrestamos[5][12]) {

    int i;
    char clave[5], DNI[10];
    int mes, anno;
    int diaHoy = 0, mesHoy = 0, annoHoy = 0;

    system("cls");
    printf("\nPeliculas disponibles ");
    printf("\n-----\n");
    for(i = 0; i < *contPelis; i++){
        if(peliculas[i].DNIPrestado[0] == '-'){
            printf("\n%s --> %s",peliculas[i].titulo, peliculas[i].clave);
        }
    }
}

printf("\n\nClave de la pelicula que quiere tomar prestada: ");
fflush(stdin);
scanf("%s",&clave);
int pos = BuscarClave(clave, peliculas);
if(pos != -1){
    printf("\nDNI del usuario: ");
    fflush(stdin);
    scanf("%s",&DNI);
    strcpy(peliculas[pos].DNIPrestado, DNI);

    do{

```

```

        printf("\nMes en el que se esta prestando: ");
        fflush(stdin);
        scanf("%d",&mes);
        mes = mes - 1;
        printf("\nAnno en el que se esta prestando: ");
        fflush(stdin);
        scanf("%d",&anno);
        anno = anno - 2018;

    } while((mes < 0) || (mes > 12) || (anno < 0) || (anno > 4));

    matrixPrestamos[anno][mes]++;

    printf("\nPelícula prestada :-)");

} else printf("ERROR: Clave no encontrada...");

Pausa();

}

/*****
*****/
/* Subprograma: Devolucion */
/* Tipo: Procedimiento (función tipo void) */
/* Prerrequisitos: No tiene. */
/* Objetivo: Permite realizar una devolucion de una película que haya sido
prestada */
/*****
*****/
void Devolucion(tRegPelícula películas[MAXPELI]){

    char clave[5];

    system("cls");
    printf("Clave de la película que quiere devolver: ");
    fflush(stdin);
    scanf("%s",&clave);
    int pos = BuscarClave(clave, películas);
    if(pos != -1){
        if(películas[pos].DNIPrestado[0] != '-'){
            películas[pos].DNIPrestado[0] = '-';
            printf("\nLa película * %s * ha sido devuelta",
películas[pos].titulo);

        } else printf("La película * %s * no estaba prestada",
películas[pos].titulo);

    } else printf("ERROR: Clave no encontrada...");

    Pausa();

}

/*****
*****/

```



```
*****/
/* Subprograma: ListadoPelículas */
/* Tipo: Procedimiento (función tipo void) */
/* Prerrequisitos: No tiene. */
/* Objetivo: Menu para navegar por los listados de películas */
/*****
*****/
void ListadosPelículas(int *contPelis, tRegPelícula películas[MAXPELI], int
matrixPrestamos[5][12]) {

    int opcion;

    do{
        system("cls");
        printf("LISTADOS DE PELÍCULAS\n");
        printf("=====\n");
        printf("\n1.- Listado general");
        printf("\n2.- Listado de películas no prestadas");
        printf("\n3.- Listado de películas prestadas");
        printf("\n4.- Listado de películas por título");
        printf("\n5.- Listado de películas por año");
        printf("\n6.- Informe de préstamos");
        printf("\n0.- Volver");
        printf("\n\nQue desea hacer?: ");
        scanf("%d", &opcion);

        switch(opcion){
            case 1:
                ListadoGeneralPelículas(&contPelis, películas);
                break;
            case 2:
                ListadoPelisNoPrestadas(&contPelis, películas);
                break;
            case 3:
                ListadoPelisPrestadas(&contPelis, películas);
                break;
            case 4:
                ListadoPelisNombre(&contPelis, películas);
                break;
            case 5:
                ListadoPelisAño(&contPelis, películas);
                break;
            case 6:
                InformePrestamos(matrixPrestamos);
                break;
            case 0:
                break;
        }

    }while(opcion != 0);

}

/*****
```

```

*****/
/* Subprograma: ListadoGeneralPeliculas */
/* Tipo: Procedimiento (función tipo void) */
/* Prerrequisitos: No tiene. */
/* Objetivo: Muestra todas las peliculas que hay en la listaPeliculas */
/*****
*****/
void ListadoGeneralPeliculas(int *contPelis, tRegPelicula peliculas[MAXPELI]) {

    int i;
    int contPrestadas = 0;

    system("cls");
    printf("                                LISTADO GENERAL");
    printf("\n-----");
    printf("\n");
    printf("\n Clave      Titulo                Director          Anno      Prestado
Socio");

    for(i = 0; i < *contPelis; i++){
        printf("\n%s      %s      %s      %d",peliculas[i].clave, peliculas[i].titulo,
peliculas[i].director, peliculas[i].anno);
        if(peliculas[i].DNIPrestado[0] != '-') {
            printf("          SI          %d",peliculas[i].DNIPrestado);
        } else printf("          NO          ----");
    }

    printf("\n-----");
    printf("\n Numero de peliculas: %d",*contPelis);
    for(i = 0; i < *contPelis; i++){
        if(peliculas[i].DNIPrestado[0] != '-')
            contPrestadas++;
    }
    printf("\n Numero de peliculas prestadas: %d",contPrestadas);
    Pausa();
}

/*****
*****/
/* Subprograma: ListadoPelisNoPrestadas */
/* Tipo: Procedimiento (función tipo void) */
/* Prerrequisitos: No tiene. */
/* Objetivo: Muestra todas las peliculas que no han sido prestadas
/* o sea que DNIPrestado == - */
/*****
*****/
void ListadoPelisNoPrestadas(int *contPelis, tRegPelicula peliculas[MAXPELI]) {

    int i;

    system("cls");
    printf("                                PELICULAS NO PRESTADAS");
    printf("\n-----");
    printf("\n Clave      Titulo                Director          Anno");

```

```

        for(i = 0; i < *contPelis; i++){
            if(peliculas[i].DNIPrestado[0] == '-')
                printf("\n%s    %s    \t %s    %d",peliculas[i].clave,
peliculas[i].titulo, peliculas[i].director, peliculas[i].anno);
        }

        Pausa();
    }

/*****
/* Subprograma: ListadoPelisPrestadas */
/* Tipo: Procedimiento (función tipo void) */
/* Prerrequisitos: No tiene. */
/* Objetivo: Muestra todas las peliculas que no han sido prestadas */
/* o sea que DNIPrestado != - */
*****/
void ListadoPelisPrestadas(int *contPelis, tRegPelicula peliculas[MAXPELI]) {

    int i;

    system("cls");
    printf("          PELICULAS PRESTADAS");
    printf("\n-----");
    printf("\n Clave      Titulo          Socio");

    for(i = 0; i < *contPelis; i++){
        if(peliculas[i].DNIPrestado[0] != '-')
            printf("\n%s    %s    %s    %d",peliculas[i].clave,
peliculas[i].titulo, peliculas[i].DNIPrestado);
    }

    Pausa();
}

/*****
*****/
/* Subprograma: ListadoPelisNombre */
/* Tipo: Procedimiento (función tipo void) */
/* Prerrequisitos: No tiene. */
/* Objetivo: Muestra todas las peliculas que tengan en su nombre la cadena */
/* de caracteres pasada */
*****/
void ListadoPelisNombre(int *contPelis, tRegPelicula peliculas[MAXPELI]) {

    int i;
    char titulo[30];

    system("cls");
    printf("Titulo que quiere buscar: ");
    fflush(stdin);

```

```

scanf("%s", &titulo);

printf("\n\n          PELICULAS CON TITULO * %s *", titulo);
printf("\n-----");
printf("\n Clave      Titulo          Director          Anno");

for(i = 0; i < *contPelis; i++){
    if(strstr(peliculas[i].titulo, titulo) != NULL)
        printf("\n%s      %s      %s      %d", peliculas[i].clave,
peliculas[i].titulo, peliculas[i].director, peliculas[i].anno);
}

Pausa();
}

/*****
/* Subprograma: ListadoPelisAnno*/
/* Tipo: Procedimiento (función tipo void) */
/* Prerrequisitos: No tiene. */
/* Objetivo: Muestra todas las peliculas cuyo anno de estreno sea igual al pasado
*/
*****/

void ListadoPelisAnno(int *contPelis, tRegPelicula peliculas[MAXPELI]) {

    int i, anno;

    system("cls");
    printf("Anno de estreno que quiere buscar: ");
    fflush(stdin);
    scanf("%d", &anno);

    printf("\n\n          PELICULAS ESTRENADAS EN %d", anno);
    printf("\n-----");
    printf("\n Clave      Titulo          Director          ");

    for(i = 0; i < *contPelis; i++){
        if(peliculas[i].anno == anno)
            printf("\n%s      %s      %s", peliculas[i].clave, peliculas[i].titulo,
peliculas[i].director);
    }

    Pausa();
}

/*****
*****/
/* Subprograma: InformePrestamos*/
/* Tipo: Procedimiento (función tipo void) */
/* Prerrequisitos: No tiene. */
/* Objetivo: Muestra el nuemero de prestamos que se han hecho por mes y por anno
*/
*****/

```

```
*****/  
void InformePrestamos(int matrixPrestamos[5][12]) {  
  
    int i, j;  
  
    printf("INFORME DE PRESTAMOS\n");  
    printf("\nAnno    Mes    Prestamos");  
    printf("\n-----");  
    for(i = 0; i < 5; i++){  
        printf("%d\n",i+2017);  
        for(j = 0; j < 12; j++) {  
            printf("%d\t%d\n",j, matrixPrestamos[i][j]);  
        }  
    }  
    Pausa();  
}
```