

Proyecto

“Biblioteca ETSIDI-CEV”

Principales características del programa

Etor González Perea - EtorGonzalezPerea,

Valeria Muñoz Salazar - Sarael444

Carlos Rodríguez Funes - CarlosRdgzFunes

| INFORMÁTICA E100 |

# Introducción

El programa simula una biblioteca, permitiendo la gestión de usuarios, de libros y los préstamos.

Para poder visualizar correctamente la información, se recomienda configurar la pantalla para que tenga 160 caracteres de ancho y al menos 30 de alto.

Para el tratamiento de los datos se han definido **tipos de datos struct** para los usuarios, los libros y los préstamos:

// Declaración de estructura definida para USUARIOS

typedef struct

{

int num\_usuario;

char DNI[100];

char nombre[100];

char apellido[100];

char email[100];

} Usuario;

// Declaración de estructura definida para LIBROS

typedef struct

{

int num\_libro;

char ISBN[100];

char titulo[100];

char genero[100];

char autor[100];

} Libro;

// Declaración de estructura definida para PRESTAMOS

typedef struct

{

int num\_prestamo;

char DNI[100];

char ISBN[100];

char fecha\_prestamo[100];

char fecha\_devolucion[100];

} Prestamo;

En el menú principal se han definido vectores de cada una de estas estructuras (de 100 posiciones cada uno), lo que permitirá trabajar en memoria con los usuarios, libros y préstamos.

Adicionalmente, cada vez que se da de alta un nuevo usuario (o libro) o se realiza un préstamo, se modifican los datos o se da de baja un registro, se almacenan los datos en disco, en ficheros de texto plano (.txt). El contenido de estos ficheros es leído y cargado en los vectores de las estructuras de usuarios, libros y préstamos al principio del programa, de tal forma que se mantiene la información almacenada entre sesiones.

No se han utilizado **variables globales**. Los vectores de las estructuras de usuarios, libros y préstamos se pasan como parámetros a las funciones, con paso por referencia (el nombre de cada vector contiene la dirección del primer elemento de cada vector), lo que permite modificar sus valores dentro de las funciones.

Todo el programa está realizado con **funciones** definidas por el usuario:

* void menu\_general(Usuario mis\_usuarios[], Libro mis\_libros[], Prestamo mis\_prestamos[]);
* void menu\_usuarios(Usuario mis\_usuarios[], Prestamo mis\_prestamos[]);
* void menu\_libros(Libro mis\_libros[], Prestamo mis\_prestamos[]);
* void menu\_prestamos(Usuario mis\_usuarios[], Libro mis\_libros[], Prestamo mis\_prestamos[]);
* void alta\_usuario(Usuario mis\_usuarios[]);
* void listado\_usuarios\_por\_apellido(Usuario mis\_usuarios[]);
* void grabar\_usuarios\_fichero(Usuario mis\_usuarios[]);
* void leer\_usuarios\_fichero (Usuario mis\_usuarios[]);
* void eliminar\_espacios (char sLiteral[]);
* void ordenar\_usuarios (Usuario mis\_usuarios[]);
* int busqueda\_usuarios(Usuario mis\_usuarios[]);
* void modificacion\_usuarios(Usuario mis\_usuarios[], Prestamo mis\_prestamos[]);
* void baja\_usuarios(Usuario mis\_usuarios[], Prestamo mis\_prestamos[]);
* int comprobar\_email(char email[]);
* void alta\_libro(Libro mis\_libros[]);
* void listado\_libros\_ordenado (Libro mis\_libros[], int iOpcion);
* void grabar\_libros\_fichero(Libro mis\_libros[]);
* void leer\_libros\_fichero (Libro mis\_libros[]);
* void ordenar\_libros\_titulo (Libro mis\_libros[]);
* void ordenar\_libros\_autor (Libro mis\_libros[]);
* void ordenar\_libros\_genero (Libro mis\_libros[]);
* int busqueda\_libros(Libro mis\_libros[]);
* void modificacion\_libros(Libro mis\_libros[], Prestamo mis\_prestamos[]);
* void baja\_libros(Libro mis\_libros[], Prestamo mis\_prestamos[]);
* void leer\_prestamos\_fichero (Prestamo mis\_prestamos[]);
* void grabar\_prestamos\_fichero(Prestamo mis\_prestamos[]);
* void prestamo\_libro(Usuario mis\_usuarios[], Libro mis\_libros[], Prestamo mis\_prestamos[]);
* void listado\_prestamos\_por\_libro(Usuario mis\_usuarios[], Libro mis\_libros[], Prestamo mis\_prestamos[]);
* void ordenar\_prestamos (Prestamo mis\_prestamos[]);
* void devolucion\_libro(Usuario mis\_usuarios[], Libro mis\_libros[], Prestamo mis\_prestamos[]);
* void consulta\_prestamos\_libro(Usuario mis\_usuarios[], Libro mis\_libros[], Prestamo mis\_prestamos[]);
* void consulta\_prestamos\_usuario(Usuario mis\_usuarios[], Libro mis\_libros[], Prestamo mis\_prestamos[]);

Para acceder a las **diferentes opciones del programa** se han utilizado DO-WHILE para solicitar un número de opción válido y sentencias SWITCH-CASE para la selección de cada una de las opciones del menú. Debido al gran número de funciones, se han definido **tres niveles de menú** con las siguientes posibilidades:

* 1. Gestion de Usuarios
  + 1. Alta de Usuario
  + 2. Modificar datos de un Usuario
  + 3. Borrar Usuario
  + 4. Busqueda de Usuarios
  + 5. Listado de Usuarios ordenado por Apellido
  + 6. Volver a menu general
* 2. Gestion de Libros
  + 1. Alta de Libro
  + 2. Modificar datos de un Libro
  + 3. Borrar Libro
  + 4. Consulta de datos de un Libro
  + 5. Listado de Libros ordenado por Titulo
  + 6. Listado de Libros ordenado por Autor;
  + 7. Volver a menu general
* 3. Gestion de Prestamos
  + 1. Prestamo de Libro.");
  + 2. Devolucion de Libro
  + 3. Consulta de estado de prestamo de un Libro
  + 4. Consulta de prestamos de un Usuario
  + 5. Listado de prestamos ordenado por libro
  + 6. Volver a menu general
* 4. Salir

## funcionamiento del programa

El programa tiene las siguientes características principales:

**Gestión de ficheros:**

* Se utilizan los ficheros “usuarios.txt”, “libros.txt” y “prestamos.txt” para almacenar en disco la información que se va introduciendo en los vectores de estructura correspondientes. Esto permite mantener los datos entre sesiones.
* Las funciones de carga del contenido de los ficheros se invocan en el main, para tener los datos desde el principio, y las funciones de grabación de datos se invocan cada vez que se modifica algún dato (alta, modificación o baja de usuarios / libros; préstamo / devolución de libros).
* Se valida que los ficheros se puedan abrir correctamente (tanto para lectura como para escritura), dando un mensaje de error en caso contrario.
* Para permitir cadenas de caracteres que incluyan espacios, se limitan los campos por medio del carácter ‘;’. En la lectura de las cadenas de caracteres se eliminan los espacios en blanco al final de las cadenas para evitar errores en las comparaciones.

**Entrada de datos y presentación en pantalla:**

* Se usa la función ‘**system("cls");**” para limpiar la pantalla y comenzar su escritura desde el principio. Lo anterior se utiliza por ejemplo al navegar por los diferentes menús (al entrar o salir de un menú se limpia la pantalla) y cada vez que la opción seleccionada no es válida. De esta forma mejora la apariencia del programa
* El programa permite la creación y modificación de usuarios y libros, introduciendo los datos por medio de la función **fgets**, en lugar de scanf. Esto permite tener cadenas de caracteres con espacios en el medio (por ejemplo para permitir nombres compuestos) y además controlar el número de caracteres máximo que se admiten del teclado (impidiendo que se supere el tamaño de la cadena de caracteres tenga definida).
* Se ha utilizado la función ‘**fflush(stdin);**’ antes de cada scanf o gets para vaciar el buffer de teclado y evitar problemas en la introducción de datos.
* Los usuarios quedan identificados por su DNI, por lo que el programa no permite que se pueda duplicar este campo (se tiene en cuenta al dar de alta un nuevo usuario y al modificar sus campos).
* Los libros quedan identificados por su ISBN, por lo que el programa no permite que se pueda duplicar este campo (se tiene en cuenta al dar de alta un nuevo libro y al modificar sus campos). Por simplicidad, se asume que no se tiene más que una copia de cada libro.
* Se valida el formato del correo electrónico para que contenga al menos un carácter ‘@’ y un carácter ‘.’ posterior, tanto en el alta de un usuario como en su modificación.
* En todos los listados de usuarios, libros o préstamos, se muestran cabeceras para indicar los campos que se muestran.
* La mayor parte de los campos son cadenas de caracteres y se muestran alineadas a la izquierda y con un número concreto de caracteres definido en el prinft (%-20d). De esta forma se consigue que toda la información aparezca en columnas alineadas.
* En los listados de búsqueda de usuarios o libros, se muestra siempre una fila por cada registro (usuario o libro), junto con toda la información relevante.

**Uso de vectores a estructuras para usuarios, libros y préstamos:**

* Los recorridos por los vectores de estructuras de usuarios, libros y préstamos se realizan con bucles FOR. La condición de finalización del recorrido es que el DNI o el ISBN sea nulo. Es frecuente tener que realizar el recorrido completo para contabilizar el número de registros de ese tipo.
* En los listados de usuarios, libros y préstamos, los vectores se muestran ordenados según diferentes criterios (apellido para usuarios, título o autor para libros, ISBN del libro para préstamos). Para realizar lo anterior se ha utilizado el **algoritmo de la burbuja** (encontrado en Internet), que permite la ordenación de un vector por un campo. Aunque es más lento que otros algoritmos, es el más sencillo de implementar.
* Las funciones de búsqueda de usuarios o libros son invocadas desde varios puntos del programa: Modificación y baja de usuario (o libro), préstamo y devolución de libros, consulta de datos de usuario (o libro). En ellas se introduce un literal que es buscado en ciertas cadenas (nombre y apellido para usuarios; título, género y autor para libro). Si la cadena contiene ese literal, el registro se muestra. Para ello se utiliza la **función ‘strstr’** de la librería <string.h>.
* Se utiliza a menudo la **función ‘strcmp’** de la librería <string.h>, que permite comparar dos cadenas de caracteres. Esto es necesario en las búsquedas de usuarios (o libros), en recorridos de los vectores de las estructuras, etc.

**Alta se usuarios (o libros):**

* Los nuevos registros se añaden al final del vector de la estructura de usuarios (o libros). Para ello se hace un recorrido del vector hasta el final.
* Se utiliza una variable como número de usuario o número de libro, que es un identificador único. Este número de registro sirve para seleccionar el usuario o libro que se quiere modificar, dar de baja, etc.
* Al dar de alta un usuario o un libro, no se permite duplicar el campo DNI o ISBN respectivamente, por lo que da un mensaje de error en cada caso (buche DO-WHILE hasta que se introduce un valor no repetido).

**Modificación y baja de usuarios (o libros):**

* Tanto en la modificación como en la baja de un usuario (o libro), se invoca a la función de búsqueda de usuario (o libro), que permite visualizar previamente todos los registros que cumplen una condición (contiene una cadena de caracteres introducida por teclado).
* Las funciones de modificación de usuarios (o libros), permiten modificar cualquier campo introducido durante el alta. La única restricción es que no se puede modificar el campo DNI en usuarios o el campo ISBN en libros, cuando el usuario o el libro ya están en préstamo (actual o pasado).
* En las funciones de modificación, si no se introduce un nuevo literal (se pulsa ENTER), se mantiene el antiguo. En cualquier caso, se muestra el valor actual del campo antes de ser modificado.
* En el caso de usuarios, se verifica que el EMAIL tenga un formato válido si se modifica.
* En las funciones de baja de usuarios (o libros), no se permite dar de baja un registro si el usuario tiene / ha tenido prestado algún libro o si el libro ya ha sido prestado anteriormente. En esta situación avisa con un mensaje de error.
* En la baja de usuarios (o libros), siempre se pide confirmación de si se quiere eliminar el registro.
* La baja de un usuario (o un libro), consiste en el recorrido del vector de la estructura correspondiente, desde el registro a eliminar y hasta el final, machacando cada registro con la información del siguiente.

**Préstamo y devolución de libros:**

* En las funciones de préstamo y devolución de libros se invoca a las funciones de búsqueda de usuarios y búsqueda de libros para que muestren un usuario (o libro) que cumpla una condición y se pueda seleccionar (eligiendo el número de usuario o número de libro).
* Antes de proceder al préstamo de un libro se verifican las siguientes situaciones:
  + Si el libro está actualmente en préstamo, muestra mensaje y no deja prestar el libro.
  + Si el usuario tiene más de 3 libros ya prestados, muestra mensaje y no deja prestar el libro.
* En el préstamo, el campo 'fecha\_prestamo' contendrá la fecha actual, obtenida del sistema con las funciones 'time' y 'localtime' y copiando resultado a una cadena de caracteres. En el campo 'fecha\_devolucion' se indica el literal 'NO DEVUELTO'.
* Además, se indica la fecha máxima de devolución, obtenida del sistema con las funciones 'time' y 'localtime' y a la que se han añadido 14 días.
* En el proceso de devolución de un libro, la condición para encontrar el préstamo buscado es que coincida el DNI del usuario, el ISBN del libro y el libro todavía esté prestado (fecha\_devolucion = 'NO DEVUELTO').
* Si encuentra el préstamo, pone la fecha actual en la ‘fecha\_devolucion’ y muestra mensaje
* En caso contrario (no se ha encontrado el préstamo tras recorrer la estructura de préstamos), se indica con un mensaje.
* Se verifica si el libro fue prestado hace más de 14 días, en cuyo caso muestra un mensaje indicando que el usuario tiene una multa de 1 € por retraso en la devolución.

**Consulta de préstamos de un usuario y un libro:**

* Existen dos funcionalidades que consisten en mostrar el historial de todos los libros prestados a un usuario determinado, o bien el historial de todos los préstamos que se han realizado a un libro concreto.
* En el primer caso se hace una búsqueda del usuario a seleccionar y en el segudno, una búsqueda del libro.
* Una vez encontrado el usuario (o el libro), se recorre la estructura de préstamos y se muestra la información que concuerde con el usuario (o libro) seleccionado.
* En este caso concreto, dado que la estructura de préstamos tan solo contiene el DNI del usuario y el ISBN del libro, se hace un recorrido por los vectores de usuarios y libros para buscar el resto de información a mostrar (nombre y apellido del usuario; título del libro).
* La información anterior muestra una fila por cada registro.