



*Instituto Tecnológico de Costa Rica*

*Ingeniería en Computación*

*Lenguajes de Programación*

*Proyecto 1*

*Profesor:*

*Ing. Allan Rodriguez Davila*

*Estudiantes:*

*Aaron Piñar - 2019063903*

*Duan Espinoza - 2019079490*

*Segundo Semestre*

*Fecha: 06 de septiembre*

*Año:2023*

## **Gestión de biblioteca**

### **Descripción del problema.**

Una biblioteca es el lugar donde se guardan libros, sin embargo, en la actualidad esta concepción se ha visto superada para pasar a referirse tanto a las colecciones bibliográficas y audiovisuales, como a las instituciones que las crean y las ponen en servicio para satisfacer las necesidades de los usuarios. Las bibliotecas pretenden responder a la amplia gama de necesidades que pueden demandar sus usuarios. Además de obras literarias clásicas, sus fondos pueden estar integrados por textos que proporcionan información sobre servicios sociales, obras de referencia, discos, películas, libros recreativos, entre otros. El sistema para gestionar una biblioteca puede incluir funciones operativas para la administración del catálogo, así como, funciones para los usuarios como por ejemplo búsqueda y préstamo de libros.

El proyecto consiste en crear un “Sistema de gestión de biblioteca” en C que permita gestionar las principales funciones operativas y de usuario. El programa se utilizará a través de la consola, no se implementará interfaz gráfica. La información se almacenará y consultará desde archivos json.

### **Diseño del programa.**

El diseño del programa se basa en un sistema de gestión de usuarios y un catálogo de libros. El código se divide en tres archivos principales:

``manageUsers.h``, ``manageCatalog.h``, y ``main.c``. A continuación, se explican las decisiones de diseño y los algoritmos utilizados en función del código proporcionado en estos archivos:

### **En manageUsers.h:**

1. *Estructura de Datos struct Usuario:* Se utiliza una estructura de datos llamada Usuario para representar la información de un usuario. Esta estructura contiene campos para el nombre, la cédula y el correo del usuario.
2. *Validación de Cédula:* Se implementa una función `validarCedula` que verifica si una cadena de texto representa una cédula válida. La función comprueba si la cédula tiene exactamente 9 dígitos y si todos los caracteres son dígitos.
3. *Ingreso de Datos del Usuario:* La función `ingresarUsuario` permite al usuario ingresar sus datos, incluyendo nombre, cédula y correo. Los datos ingresados se almacenan en una estructura de tipo `Usuario`.
4. *Carga y Guardado de Usuarios en Formato JSON:* Se implementan funciones para cargar y guardar la lista de usuarios desde y hacia un archivo JSON. El código utiliza la biblioteca cJSON para el análisis y la generación de JSON. Si el archivo JSON no existe o está vacío, se crea un nuevo objeto JSON para almacenar los usuarios.

### **En manageCatalog.h:**

1. *Estructura de Datos struct Libro:* Se define una estructura `Libro` que almacena información sobre los libros, incluyendo un ID, título, autor, año, género, descripción, cantidad y estado.

2. *Actualización de la Base de Datos de Libros:* La función `actualizaBaseLibros` actualiza una base de datos de libros. Lee los datos desde un archivo de texto `registro.txt` y los convierte en estructuras `Libro`. Luego, crea un archivo JSON llamado `libros.json` y guarda los libros en formato JSON.

3. *Búsqueda de Libros por Texto:* La función `buscarLibroPorTexto` permite buscar libros en el archivo JSON `libros.json` por un texto específico en los campos de título, autor y descripción. Se utiliza la biblioteca `cJSON` para analizar el archivo JSON y buscar coincidencias.

4. *Búsqueda Avanzada de Libros:* La función `buscarLibroAvanzado` permite realizar una búsqueda más avanzada de libros en el archivo JSON `libros.json` utilizando múltiples criterios, como título, autor, resumen y género. Se pueden especificar valores "no" para los criterios que no se desean tener en cuenta en la búsqueda.

### **En prestamo.h:**

1. *Estructura del préstamo (struct Loan):* Se utiliza una estructura `struct Loan` para representar un préstamo de libro. Esta estructura incluye los siguientes campos:

- `loanId`: Identificador único del préstamo.
- `userId`: Cédula del usuario que realiza el préstamo.
- `bookId`: Identificador del ejemplar de libro prestado.
- `startDate`: Fecha de inicio del préstamo en formato "dd/mm/yyyy".
- `endDate`: Fecha de entrega del libro en formato "dd/mm/yyyy".

2. *Variables globales*: Se utiliza una variable global `lastLoanId` para llevar un seguimiento del último identificador de préstamo utilizado. Esta variable se carga desde un archivo de configuración al inicio del programa y se actualiza cada vez que se genera un nuevo préstamo.

3. *Funciones de gestión de archivos*: El programa utiliza funciones para cargar y guardar datos en archivos JSON. Esto permite mantener registros de libros, usuarios y préstamos en archivos separados.

Las funciones `cargarLibrosDesdeArchivo`, `cargarUsuariosDesdeArchivo`, `cargarPrestamosDesdeArchivo`, `guardarPrestamosEnArchivo` y otras se utilizan para interactuar con archivos JSON.

4. *Operaciones de préstamo*: La función `prestarEjemplar` permite a un usuario realizar un préstamo de un ejemplar de libro. Verifica la disponibilidad del ejemplar, actualiza la cantidad disponible y genera un comprobante de préstamo.

La función `actualizarCantidadYVerificarEstado` se encarga de reducir la cantidad de ejemplares disponibles y marcar un ejemplar como no disponible si es necesario.

*5. Operaciones de devolución:* La función `devolverEjemplar` permite a un usuario realizar la devolución de un ejemplar de libro. Se calcula el monto a cancelar en función de la duración del préstamo y se marca el ejemplar como disponible nuevamente.

La función `calcularMontoACancelar` calcula el monto a pagar por el préstamo en función de la duración y la fecha de devolución.

*6. Gestión de préstamos:* Las funciones `agregarPrestamoAPrestamosJson`, `realizarDevolucionEjemplar`, `buscarPrestamoPorId` y `eliminarPrestamoPorId` se utilizan para realizar operaciones de gestión de préstamos, como agregar nuevos préstamos, buscar préstamos por identificador, realizar devoluciones y eliminar préstamos.

*7. Verificación de vencimiento de préstamos:* La función `verificarVencimientoPrestamos` comprueba si hay préstamos que están vencidos o próximos a vencerse. Si encuentra préstamos en esta condición, los muestra al usuario.

### **Consideraciones Generales:**

- Se utilizan archivos JSON para almacenar datos de usuarios y libros, lo que permite una estructura de datos flexible y fácil de leer/escribir.

- Se han implementado algoritmos de procesamiento de cadenas para la validación de cédulas y la búsqueda de libros basada en texto.
- La biblioteca cJSON se utiliza para analizar y generar archivos JSON, lo que facilita el manejo de datos estructurados.
- El diseño del programa separa claramente la gestión de usuarios y la gestión del catálogo de libros en archivos de encabezado (manageUsers.h y manageCatalog.h), lo que facilita su mantenimiento y extensión.
- *Tanto la fecha de inicio como la de entrega no están validadas bien al igual que ciertas validaciones para casos o excepciones en los inputs del usuario*

## **Diseño del programa.**

En el código, se han incluido varias bibliotecas estándar de C y una biblioteca externa llamada cJSON. A continuación, se dará una explicación de las bibliotecas utilizadas:

### **1. stdio.h (Standard Input/Output Library):**

Esta biblioteca proporciona funciones para entrada y salida estándar, como printf y scanf. Se utiliza para mostrar mensajes en la consola y obtener entrada del usuario.

## **2. `stdlib.h` (Standard Library):**

Contiene funciones para la gestión de memoria dinámica, como `malloc` y `free`. En el programa, se utiliza `malloc` para asignar memoria dinámica a cadenas de caracteres.

## **3. `string.h` (String Library):**

Ofrece funciones para manipular cadenas de caracteres, como ``strcpy``, ``strcmp`` y ``strlen``. Se usa para trabajar con datos de tipo cadena, como fechas y cédulas de usuarios.

## **4. `stdbool.h` (Standard Boolean Library):**

Proporciona un tipo de datos booleano (`bool`) y valores booleanos `true` y `false``. Se utiliza para representar valores booleanos, como el estado de disponibilidad de los ejemplares de libros.

## **5. `time.h` (Time Library):**

Contiene funciones relacionadas con la manipulación de tiempo y fechas. En el programa, se utiliza para obtener la fecha actual en el formato deseado y calcular la duración de los préstamos.

## **6. `cJSON`:**

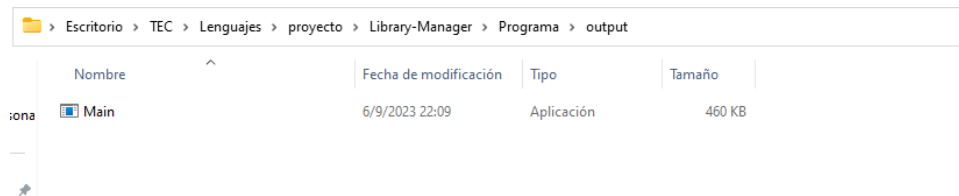
Esta no es una biblioteca estándar de C, sino una biblioteca externa llamada `cJSON`. Se utiliza para trabajar con datos JSON, como cargar, crear, modificar y guardar archivos JSON. Es fundamental para gestionar los datos de libros,



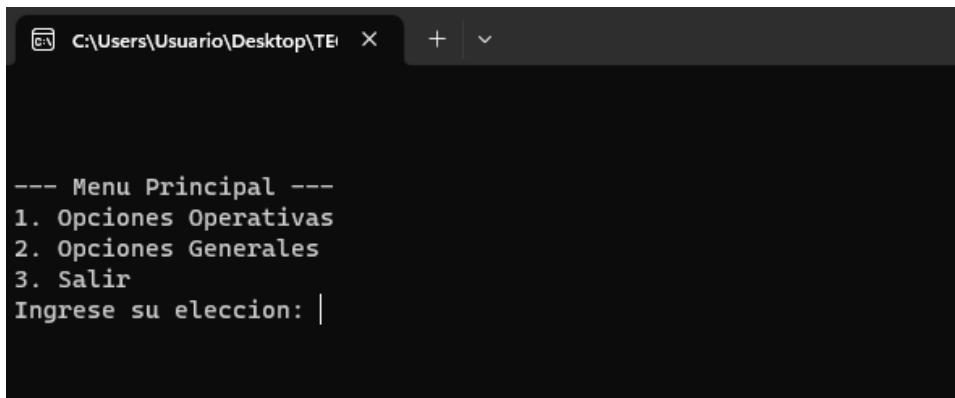
usuarios y préstamos en EL programa. Esta biblioteca proporciona funciones para analizar y generar datos JSON de manera eficiente.

## Manual de usuario

1. Debe abrir la carpeta Programa y entrar carpeta output para dar doble click al archivo Main.exe



2. Una vez abierto el Main se mostrara un menu en el cual se puede interactuar en base a los numeros de opciones. En el caso de que el usuario indique algo no puesto se le avisara con un mensaje y volvera al menu



3. En el caso de seleccionar opciones operativas. Se desplegara otro menu el cual de misma forma

```
Ingrese su eleccion: 1
```

```
--- Opciones Operativas ---
```

1. Gestion de catalogo
2. Gestion de usuarios
3. Historial de prestamos
4. Vencimiento de prestamos
5. Estadisticas
6. Volver

```
Ingrese su eleccion: |
```

4. Caso gestion de catalogo. Para cargar datos sera atravez de un txt colocado en la carpeta data con el nombre de registro.txt. El registro va ir separado por | entre cada etiqueta. Al seleccionar la opcion dara un mensaje y creara un archivo json con la informacion del txt.

Escritorio > TEC > Lenguajes > proyecto > Library-Manager > Programa > data

Nombre	Fecha de modificación	Tipo	Tamaño
config	6/9/2023 21:45	Documento de te...	1 KB
libros	6/9/2023 22:27	JSON File	3 KB
prestamos	6/9/2023 21:58	JSON File	1 KB
registro	6/9/2023 15:15	Documento de te...	2 KB
usuarios	6/9/2023 21:41	JSON File	1 KB

La sombra del viento Carlos Ruiz Zafon 2001 Ficcion Una historia sobre un libro maldito 4
Cien anos de soledad Gabriel Garcia Marquez 1967 Realismo magico La saga de la familia Buendia en Macondo 4
La sombra del viento Carlos Ruiz Zafon 2001 Ficcion Una historia sobre un libro maldito 4
Cien anos de soledad Gabriel Garcia Marquez 1967 Realismo magico La saga de la familia Buendia en Macondo 4
El nombre del viento Patrick Rothfuss 2007 Fantasia Las aventuras de Kvothe en busca de conocimiento 4
1984 George Orwell 1949 Ciencia ficcion Un mundo totalitario y su lucha por la verdad 4
Harry Potter y la piedra filosofal J.K. Rowling 1997 Fantasia El inicio de las aventuras de Harry en Hogwarts 4
Los juegos del hambre Suzanne Collins 2008 Ciencia ficcion Un futuro distopico y lucha por la supervivencia 4
Los juegos del hambre Suzanne Collins 2008 Ciencia ficcion Un futuro distopico y lucha por la supervivencia 4
Los juegos del susto Suzanne Collins 2008 Ciencia ficcion Un futuro distopico y lucha por la supervivencia 4

```
--- Opciones Operativas ---
```

1. Gestion de catalogo
2. Gestion de usuarios
3. Historial de prestamos
4. Vencimiento de prestamos
5. Estadisticas
6. Volver

```
Ingrese su eleccion: 1
```

```
Se Actualizo la base de datos de libros registrados
```

```
--- Opciones Operativas ---
```

5. Gestion de usuarios. El sistema permitira registrar usuarios siempre y cuando no existan cedulas registradas con anterioridad y sean iguales a las entrantes. Por ultimo va a agregar los datos al archivo json con los datos nuevos.

```
--- Opciones Operativas ---
1. Gestion de catalogo
2. Gestion de usuarios
3. Historial de prestamos
4. Vencimiento de prestamos
5. Estadisticas
6. Volver
Ingrese su eleccion: 2
Por favor, ingrese los datos del nuevo usuario:
Ingrese el nombre: Ester
Ingrese la cédula (9 dígitos): 123456789
Ingrese el correo: e@gmail.com

Datos del usuario registrados y guardados en ../data/usuarios.json
```

6. Opciones generales (Busqueda simple) se le solicitara al usuario un autor, titulo o descripcion y se buscara en base a lo colocado y mostrara los matchs

```
C:\Users\Usuario\Desktop\TE x + -
-----
Bienvenido a busqueda simple
La busqueda se realiza en base a: Titulo, Autor, Descripción
~Indique el contenido relacionado a buscar:Gabriel Garcia Marquez|
```

```
-----
Libro encontrado
ID: 2
Titulo: Cien anos de soledad
Descripción: La saga de la familia Buendia en Macondo
Estado: Disponible

-----
Libro encontrado
ID: 4
Titulo: Cien anos de soledad
Descripción: La saga de la familia Buendia en Macondo
Estado: Disponible
-----
```

7. Opciones generales (Busqueda avanzada) se le solicita al usuario espacio por espacio de lo que desea buscar y si desea realizarlo por el metodo exacto o contiene. Ademas el usuario puede indicar no en una etiqueta para no contarla en la busqueda.

```
-----
Bienvenido a la búsqueda avanzada

Debe rellenar las siguientes etiquetas con la información de búsqueda. Si no desea buscar por esa etiqueta, coloque 'no'
. La búsqueda debe ser sin caracteres especiales

Indique el Nombre del libro:no

Indique el autor:no

Indique el resumen:no

Indique el género:fantasia|
```

```
-----

Libro encontrado
ID: 5
Título: El nombre del viento
Descripción: Las aventuras de Kvothe en busca de conocimiento
Género: Fantasia

-----

Libro encontrado
ID: 7
Título: Harry Potter y la piedra filosofal
Descripción: El inicio de las aventuras de Harry en Hogwarts
Género: Fantasia
```

Pruebas de funcionalidad

Gestión de catálogo

📁 Escritorio > TEC > Lenguajes > proyecto > Library-Manager > Programa > data

	Nombre	Fecha de modificación	Tipo	Tamaño
na	config	6/9/2023 21:45	Documento de te...	1 KB
.	libros	6/9/2023 22:27	JSON File	3 KB
📁	prestamos	6/9/2023 21:58	JSON File	1 KB
📄	registro	6/9/2023 15:15	Documento de te...	2 KB
👤	usuarios	6/9/2023 21:41	JSON File	1 KB

registro

Archivo Editar Ver

La sombra del viento|Carlos Ruiz Zafon|2001|Ficcion|Una historia sobre un libro maldito|4  
Cien anos de soledad|Gabriel Garcia Marquez|1967|Realismo magico|La saga de la familia Buendia en Macondo|4  
La sombra del viento|Carlos Ruiz Zafon|2001|Ficcion|Una historia sobre un libro maldito|4  
Cien anos de soledad|Gabriel Garcia Marquez|1967|Realismo magico|La saga de la familia Buendia en Macondo|4  
El nombre del viento|Patrick Rothfuss|2007|Fantasia|Las aventuras de Kvothe en busca de conocimiento|4  
1984|George Orwell|1949|Ciencia ficcion|Un mundo totalitario y su lucha por la verdad|4  
Harry Potter y la piedra filosofal|J.K. Rowling|1997|Fantasia|El inicio de las aventuras de Harry en Hogwarts|4  
Los juegos del hambre|Suzanne Collins|2008|Ciencia ficcion|Un futuro distopico y lucha por la supervivencia|4  
Los juegos del hambre|Suzanne Collins|2008|Ciencia ficcion|Un futuro distopico y lucha por la supervivencia|4  
Los juegos del susto|Suzanne Collins|2008|Ciencia ficcion|Un futuro distopico y lucha por la supervivencia|4

Ln 1, Col 1100%Windows (CRLF)UTF-8

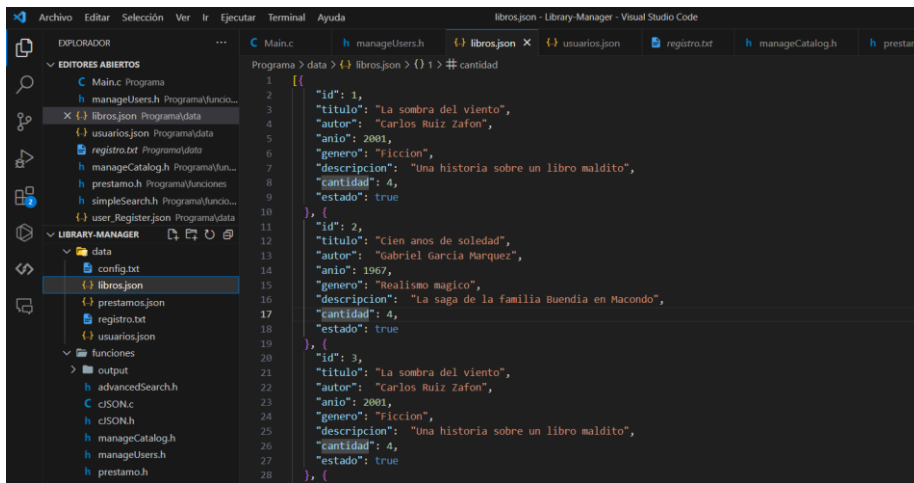
```

--- Opciones Operativas ---
1. Gestion de catalogo
2. Gestion de usuarios
3. Historial de prestamos
4. Vencimiento de prestamos
5. Estadisticas
6. Volver
Ingrese su eleccion: 1

Se Actualizo la base de datos de libros registrados

--- Opciones Operativas ---

```



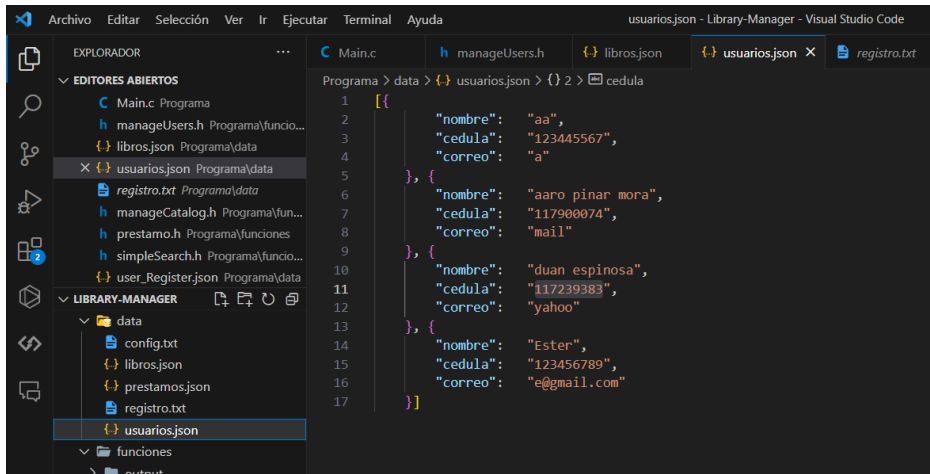
## Gestión de usuario

```

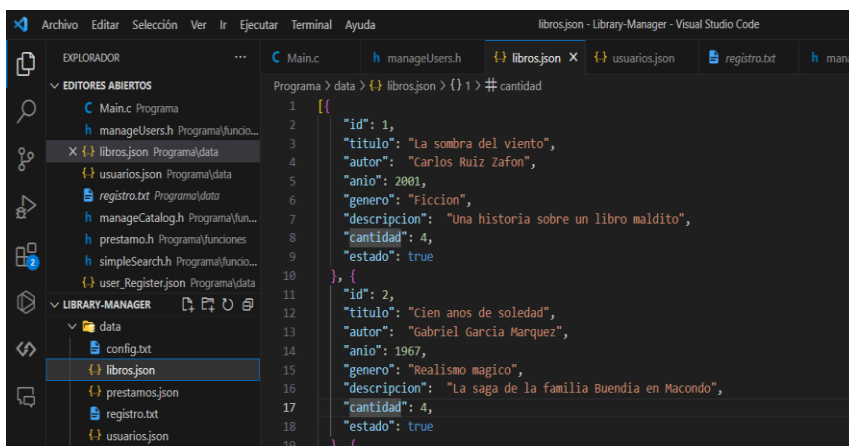
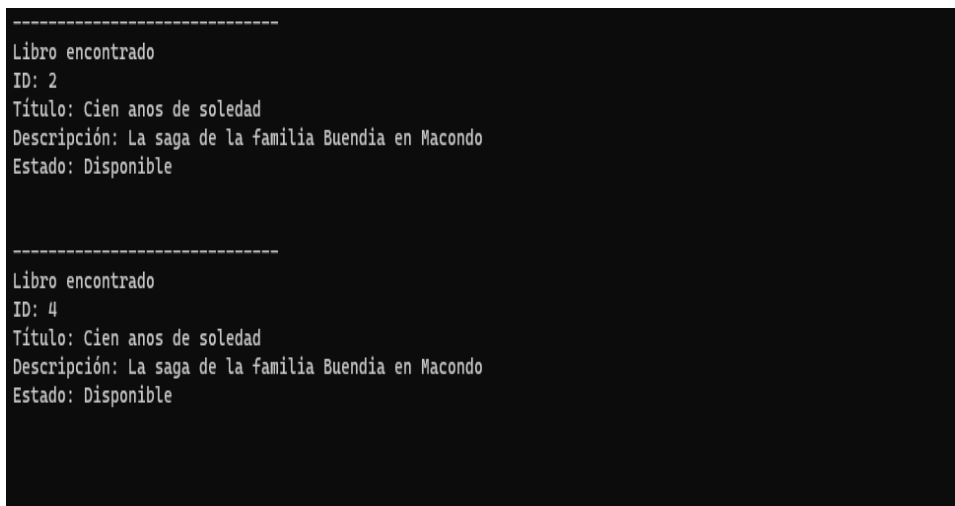
--- Opciones Operativas ---
1. Gestion de catalogo
2. Gestion de usuarios
3. Historial de prestamos
4. Vencimiento de prestamos
5. Estadisticas
6. Volver
Ingrese su eleccion: 2
Por favor, ingrese los datos del nuevo usuario:
Ingrese el nombre: Ester
Ingrese la cédula (9 dígitos): 123456789
Ingrese el correo: e@gmail.com

Datos del usuario registrados y guardados en ../data/usuarios.json

```



## Búsqueda simple



## Búsqueda avanzada

*La funcionalidad no muestra bien los valores deseados*

## Vencimiento de préstamo

```
--- Opciones Operativas ---
1. Gestion de catalogo
2. Gestion de usuarios
3. Historial de prestamos
4. Vencimiento de prestamos
5. Estadisticas
6. Volver
```

*Nota: La funcionalidad produce un cierre inesperado*

## Préstamo

```
C:\Users\Usuario\Desktop\sei x + v
2. Opciones Generales
3. Salir
Ingrese su eleccion: 2
--- Opciones Generales ---
1. Busqueda simple
2. Busqueda avanzada
3. Prestamo de ejemplar
4. Devolucion de ejemplar
5. Volver
Ingrese su eleccion: 3
Ingrese su cedula: 123456789
Ingrese el identificador del ejemplar: 2
Ingrese la fecha de inicio (dd/mm/yyyy): 06/09/2023
Ingrese la fecha de entrega (dd/mm/yyyy): 10/09/2023
Comprobante de Préstamo:
ID de Préstamo: 5
Cedula del Usuario: 123456789
ID del Ejemplar: 2
Fecha de Inicio: 06/09/2023
Fecha de Entrega: 10/09/2023
--- Opciones Generales ---
1. Busqueda simple
2. Busqueda avanzada
3. Prestamo de ejemplar
4. Devolucion de ejemplar
5. Volver
Ingrese su eleccion: |
programa.exe
```

```
1  [{"loanId": 1,
2    "userId": "117900074",
3    "bookId": 1,
4    "startDate": "12/02/2023",
5    "endDate": "15/02/2024"},
6
7  ],
8  [{"loanId": 5,
9    "userId": "123456789",
10   "bookId": 2,
11   "startDate": "06/09/2023",
12   "endDate": "10/09/2023"}]
13 }
```

*Nota: Presenta un error donde no modifica la “cantidad” en libros.json*

Devolución

```
Ingrese su eleccion: 2
--- Opciones Generales ---
1. Busqueda simple
2. Busqueda avanzada
3. Prestamo de ejemplar
4. Devolucion de ejemplar
5. Volver
Ingrese su eleccion: 3
Ingrese su c|@dula: 123456789
Ingrese el identificador del ejemplar: 2
Ingrese la fecha de inicio (dd/mm/yyyy): 06/09/2023
Ingrese la fecha de entrega (dd/mm/yyyy): 10/09/2023

Comprobante de Pr|@stamo:
ID de Pr|@stamo: 5
C|@dula del Usuario: 123456789
ID del Ejemplar: 2
Fecha de Inicio: 06/09/2023
Fecha de Entrega: 10/09/2023

--- Opciones Generales ---
1. Busqueda simple
2. Busqueda avanzada
3. Prestamo de ejemplar
4. Devolucion de ejemplar
5. Volver
Ingrese su eleccion: 4
Ingrese el identificador de pr|@stamo: 5
Ingrese la fecha de devoluci|@n (dd/mm/yyyy): 07/09/2023
```

*Nota: La función produce un cierre inesperado después de mostrar el monto*

Análisis de resultados.

Persistencia de Datos.	
Guardar datos en archivos JSON.	100%
Cargar datos desde archivos JSON.	100%



Opciones Generales	
Búsqueda simple	100%
Busqueda avanzada	50%
Préstamo de ejemplar	85% (Errores en restar cantidades)
Devolución de ejemplar	50% (Errores en sumar cantidades y no borra el objeto en prestamos)

Opciones Generales	
Gestión de catálogo	100%
Gestión de usuarios	100%
Vencimiento de préstamos	40% (se cierra el programa)

Estadísticas	0%
--------------	----

### **Bitácora y referencias:**

[16-Days/Library-Manager: Proyecto de Lenguajes de Programación \(github.com\)](#)

Repositorio de Github

[DaveGamble/cJSON: Ultralightweight JSON parser in ANSI C \(github.com\)](#)

Librería cJSON