

Proyecto 1 - Gestión de librería

Estudiantes

Geovanni González Aguilar

Daryll Martínez Rodríguez

Gerny Diaz Hall

Escuela de Ingeniería en Computación

Curso

IC4700 - Lenguajes de programación

Profesor

Allan Rodríguez Dávila

Grupo

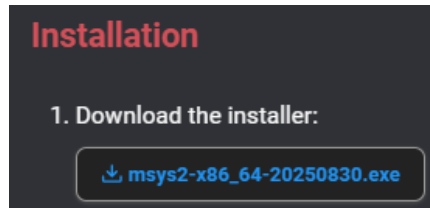
60

II Semestre 2025

1. Manual de usuario:

Primero que nada, debemos tener un entorno listo para ejecutar el proyecto, así que primeramente debemos instalar MSYS2 y después instalar gcc y make dentro del Msys2

Muy bien a primero debemos descargar el instalador de MSYS2: para eso la página oficial de MSYS2 y descarga el instalador:



Después en el proceso de instalación lo único que debemos de realizar es tocar el botón de **Next** o **Siguiente**.

Después de eso, dentro de la terminal de MSYS2 MSYS hay que ejecutar los siguientes comandos:

- **pacman -Syu**: Este comando sirve para actualizar el sistema de paquetes
- **pacman -Su**: Este comando sirve para terminar la actualización de paquetes

Después de eso se tiene que ejecutar el siguiente comando:

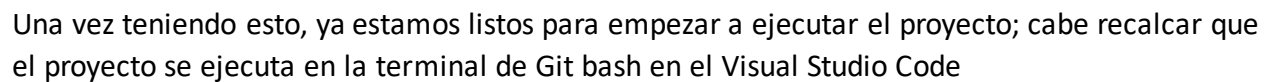
- **pacman -S --needed base-devel mingw-w64-x86_64-toolchain**: Lo que hace este comando es instalar el grupo de herramientas de desarrollo gcc (que incluye gcc, g++) y Make; cuando pregunte qué paquetes queremos instalar, simplemente hay que presionar **Enter** para seleccionar todos los paquetes por defecto del grupo de herramientas.

Después para verificar que la instalación se haya realizado de forma correcta nos vamos a la terminal de comandos MSYS2 MINGW64 y ejecutamos los siguientes comandos:

- **gcc --version**: Para verificar la versión e instalación de gcc
- **make --version**: Para verificar la versión e instalación de make

Antes de empezar a ejecutar el proyecto también es recomendable revisar las variables de entorno del sistema y del usuario específicamente en la opción de path, después de revisar que se muestren de la siguiente manera:

Variables de entorno del usuario:



Para ejecutar el proyecto primero debemos irnos a la carpeta raíz del proyecto:



En esta opción nos pedirá un usuario y contraseña, las cuales estan en el archivo admin.json:

```

{ } admin.json X
programa > { } admin.json > ...
1  [
2      "nombreLocalComercial": "Libreria Central",
3      "telefono": "555-1234",
4      "cedulaJuridica": "123456789",
5      "horarioAtencion": "Lunes a Viernes 9:00-18:00",
6      "numeroSiguietePedido": 22,
7      "usuario": "admin",
8      "password": "admin123"
9  ]
10

```

```

===== SISTEMA DE GESTIÓN DE LIBRERÍA =====
1. Opciones Administrativas (requiere usuario y contraseña)
2. Opciones Generales
3. Salir
Seleccione una opción: 1
Ingrese usuario: admin
Ingrese contraseña: admin123

```

Una vez estando dentro del menú principal, iremos a la opción de ingresar un nuevo libro, en el cual nos pedirán el nombre del libro, autor de obra, precio, y la cantidad de libros:

```
===== MENÜ ADMINISTRATIVO =====
1. Registrar libros
2. Manejo de inventario
3. Registrar clientes
4. Crear pedido
5. Estadísticas
6. Volver al menú principal
Seleccione una opción: 1
Ingrese el título del libro: Liga de los pelirrojos
Ingrese el autor del libro: Edgar allan poe
Ingrese el precio del libro: 30000
Ingrese la cantidad del libro: 20
Libro agregado exitosamente: Liga de los pelirrojos - Edgar allan poe
```

Y como se puede ver en el archivo .txt se puede ver donde se agregó el nuevo libro:

```
libros.txt IM, M X
programa > data > libros.txt
1 L001;Liga de los pelirrojos;Edgar allan poe;30000.00;20
2
```

Ahora iremos con la opción 2, cargar el inventario:

```
===== MENÜ ADMINISTRATIVO =====
1. Registrar libros
2. Cargar inventario
3. Registrar clientes
4. Crear pedido
5. Estadísticas
6. Volver al menú principal
Seleccione una opción: 2
```

En esta opción podemos ver en un archivo .txt cuál es el inventario de libros que se tiene

Después de eso tenemos la opción número 3, la cual es la función para agregar un cliente al sistema, en el cual se pueden ver los clientes agregados por medio del .txt:

```
===== MENÚ ADMINISTRATIVO =====
1. Registrar libros
2. Cargar inventario
3. Registrar clientes
4. Crear pedido
5. Estadísticas
6. Volver al menú principal
Seleccione una opción: 3
Ingrese la cédula del cliente: 703140731
Ingrese el nombre del cliente: Gerny Diaz Hall
Ingrese el teléfono del cliente: 83530919
Cliente registrado correctamente.
```

```
programa > data > clientes.txt
1 703110547;Geovanni;62189930
2 703140731;Gerny Diaz Hall;83530919
3
```

En el siguiente punto tenemos la opción 4, que es la de crear pedidos:

```
===== MENÚ ADMINISTRATIVO =====
1. Registrar libros
2. Cargar inventario
3. Registrar clientes
4. Crear pedido
5. Estadísticas
6. Volver al menú principal
Seleccione una opción: 4
```

Seguidamente tenemos el punto 5 que es el apartado de las estadísticas, como de pedidos/ventas y ordenadas por mes y año:

```
===== MENÜ ADMINISTRATIVO =====
1. Registrar libros
2. Cargar inventario
3. Registrar clientes
4. Crear pedido
5. Estadísticas
6. Volver al menú principal
Seleccione una opción: 5
```

Y la opción 6 no hay mucho que explicarle ya que esta solo nos devuelve al menú principal

Después tenemos las opciones generales:

```
===== SISTEMA DE GESTIÓN DE LIBRERÍA =====
1. Opciones Administrativas (requiere usuario y contraseña)
2. Opciones Generales
3. Salir
Seleccione una opción: 2
```

Aquí no voy a enfatizar mucho, pero lo que se muestra es lo siguiente:

```
===== MENÜ GENERAL =====
1. Consulta de catálogo
2. Consulta de cliente
3. Volver al menú principal
Seleccione una opción: 
```

En la opción número 1, se pueden consultar cual es el catálogo que tiene la librería, o sea, el stock que tiene y en la 2 se pueden consultar el usuario ingresa la cédula del cliente y el sistema deberá mostrar los datos relacionados con el cliente.

Después de eso nos salimos del menú general y nos devolvemos al menú principal con la opción 3 y estando en el menú principal nos salimos del programa con la opción 3.


```
===== MENÚ GENERAL =====
1. Consulta de catálogo
2. Consulta de cliente
3. Volver al menú principal
Seleccione una opción: 3
```

```
===== SISTEMA DE GESTIÓN DE LIBRERÍA =====
1. Opciones Administrativas (requiere usuario y contraseña)
2. Opciones Generales
3. Salir
Seleccione una opción: 3
Saliendo del sistema. ¡Hasta luego!
```

2. Descripción del problema:

El sistema permite administrar los recursos de una librería, incluyendo libros, clientes y pedidos. Los objetivos son:

- Mantener un **inventario actualizado** de libros.
- Gestionar clientes con validación de datos (cédula única y teléfono válido).
- Registrar y procesar pedidos, calculando precios, impuestos y descontando stock.
- Generar **estadísticas** de ventas por libro, cliente y año.
- Garantizar **persistencia** de datos mediante archivos de texto y JSON.
- Se busca un **programa modular y escalable** que permita agregar nuevas funcionalidades sin afectar la estabilidad del sistema.

El sistema está diseñado para ser **interactivo**, mediante menús que guían al usuario, con opciones administrativas protegidas por usuario y contraseña, y opciones generales accesibles sin autenticación.

El problema principal es organizar y manipular información dinámica (libros, clientes y pedidos) de manera segura y eficiente, garantizando consistencia en los datos y persistencia entre ejecuciones.

3. Diseño del programa

Módulo	Propósito principal	Dependencias
Libro.h/c	Manejo del inventario de libros: agregar, eliminar, mostrar, filtrar y almacenar libros.	stdlib.h, string.h, stdio.h, Utilidades.h
Cliente.h/c	Gestión de clientes: registro, validación, consulta, eliminación y almacenamiento.	stdlib.h, stdbool.h, stdio.h, string.h, ctype.h, Pedido.h, Utilidades.h,
Pedido.h/c	Gestión de pedidos: creación, modificación, cálculo de precios, relación con libros y clientes.	Libro.h, Config.h, Utilidades.h, Cliente.h, stdlib.h, string.h, stdio.h
Utilidades.h/c	Funciones de soporte: validaciones, lectura/escritura de archivos, manejo de strings y memoria dinámica.	Cliente.h, Pedido.h, Libro.h, Config.h, stdio.h, stdlib.h, string.h, ctype.h
Config.h/c	Configuración general del sistema: datos de la librería, credenciales de administrador y control de pedidos.	stdio.h, stdlib.h, string.h, cJSON.h, Utilidades.h
Estadistica.h/c	Generación de estadísticas: ventas por cliente, libro y año, autor más vendido, clientes frecuentes.	Pedido.h, Cliente.h, Pedido.h, Utilidades.h, stdio.h, stdlib.h, string.h

4.1 Separación por módulos:

- **Cliente:** Manejo de clientes, validación, registro y eliminación.
- **Libro:** Inventario de libros, agregación, eliminación y filtrado.
- **Pedido:** Gestión de pedidos, cálculo de precios, asociación con clientes y stock.

- **Utilidades:** Funciones auxiliares para manejo de strings, validaciones y lectura de archivos.
- **Configuración:** Parámetros del sistema, información del local y credenciales de administrador.
- **Estadísticas:** Cálculo y presentación de reportes de ventas y libros más vendidos.

4.2 Uso de memoria dinámica:

Los arreglos de clientes, libros y pedidos se gestionan dinámicamente, permitiendo redimensionarlos automáticamente según crece la información, evitando desperdicio de memoria.

4.3 Validaciones centralizadas:

Todas las entradas de usuario (cédula, teléfono, fechas, cantidades) son validadas mediante funciones específicas, asegurando consistencia de datos y evitando errores de lógica.

4.4 Persistencia de datos en archivos:

- Clientes, libros y pedidos se almacenan en archivos de texto.
- La configuración del sistema se almacena en JSON para facilitar edición y lectura por otros sistemas.

5. Librerías utilizadas

El programa hace uso de **librerías estándar de C** y **librerías propias**, cada una con un propósito específico dentro del sistema:

5.1 Librerías estándar de C

- **<stdio.h>:**
Proporciona funciones para entrada y salida estándar, como printf, scanf, fgets y manejo de archivos con FILE.
- **<stdlib.h>:**
Permite la gestión de memoria dinámica (malloc, realloc, free) y funciones de conversión de datos (atoi, atof).
- **<string.h>:**
Utilizada para manipulación de cadenas de caracteres, incluyendo strlen, strcpy, strcmp y strcat.
- **<stdbool.h>:**
Permite el uso del tipo booleano bool para mejorar la legibilidad del código y la validación de condiciones.
- **<stddef.h>:**
Define tipos y macros generales como size_t y NULL, necesarios en funciones de manejo de memoria y buffers.

5.2 Librerías propias del proyecto

- **Ciente.h:**
Manejo de clientes, incluyendo registro, validación, consulta y eliminación.
- **Libro.h:**
Gestión del inventario de libros, funciones para agregar, eliminar, mostrar y filtrar libros.
- **Pedido.h:**
Manejo de pedidos, cálculo de montos, asociación de libros y clientes, y generación de estadísticas.
- **Utilidades.h:**
Funciones auxiliares para validaciones, manipulación de cadenas, lectura/escritura de archivos y utilidades generales.
- **Config.h:**
Manejo de configuración general del sistema, incluyendo credenciales del administrador, información del local y control de pedidos.
- **Estadistica.h:**
Funciones para generar reportes de ventas, libros más vendidos y clientes más activos.

5.3 Justificación del uso de librerías

- Las **librerías estándar** permiten un manejo eficiente de memoria, cadenas y archivos, asegurando portabilidad del programa.
- Las **librerías propias** modularizan el sistema, promoviendo la reutilización de código y facilitando el mantenimiento y ampliación del software.

Analisis de resultados

Todos los objetivos del programa fueron realizados y completados en su totalidad:
Se desarrollo Registrar Libros, Carga de inventario, Registrar cliente, Crear pedido, Consulta de pedidos, Estadística, Opciones Generales que incluye Consulta de catálogo, Consulta de cliente y Salir.

Justificación de toma de decisiones: Se uso la metodologia KANBAN para el desarrollo de este software, eso nos llevó a tomar unas decisiones: Se utilizaron punteros con estructuras, ya que en ocasiones era mejor pasar la copia que la dirección misma de la variable o entidad struct, ya que se pueden almacenar de manera dinámica y obtener copias, lo cual hace que se trabaje de manera segura siempre y cuando se liberase la memoria.

Punto a)

Ejemplos:

liberarLibros(libros, totalLibros): Es una función a la cual se le envían los parámetros temporales activos y no los reales ya que se necesita liberar la memoria de esas copias en particular por la memoria dura usada, mas no toda la memoria del puntero, o sea sus variables reales.

cargarClientesDesdeArchivo(&arregloClientes, &cantidadClientesActual, &capacidadDeClientesArreglo): En esta funcion pasamos una copia de la direccion, la cual cargara en las variables reales del struct, funciona ya que la memoria persiste al salir de la función, sus direcciones son accesibles y persistentes.

Punto b)

La modularidad y la integridad de los archivos se mantiene gracias a las cabeceras de cada archivo, plantillas y definiciones en los .h mientras que en los .c van sus implementaciones, esto permite que el programa no esté lleno de funciones al azar y se mantenga el orden, el uso de clases como Utilidades.h

es el punto más fuerte, ya que se hacen ahí las implementaciones y en él .c se ejecutan, estas se pueden usar en todo el programa con solo incluirlas.

La ocultación de la información se logró gracias a implementaciones de variable únicas de clase, variables como el arreglo de clientes o la cantidad que hay, la cual debería de ser información oculta, por otro lado, también se implementaron setters y getters en partes del código que se necesitaban, para no conocer valores delicados y mantener la privacidad del cliente.