



<b>GUIA MANUAL TÉCNICO DE UN SISTEMA WEB DE LA LIBRERÍA ENTRE LINEAS</b>	Código: ING_MT_01
	Versión: 1.0
	Fecha: Mayo 13 de 2025
	Página 1 de 21

1	Introducción	
1.1	Objetivos	
1.2	Grupos involucrados	
1.3	Definiciones, acrónimos y abreviaturas	
2	Descripción de la Construcción	
2.1	Entorno de desarrollo	
2.1.1	.....Estructura del proyecto	
2.1.2	.....Diseño a Modelo Relacional	
2.1.3	..... Normalización	
2.1.4	.....Diccionario de Datos	
2.1.5	.....Scripts de creación de B.D.	
2.2	Aplicación / Sistema	
2.2.1	..... Backend	
2.2.2	..... Front-End	
2.2.4	..... Código Fuente	
3	Observaciones Adicionales	

## **1 Introducción**

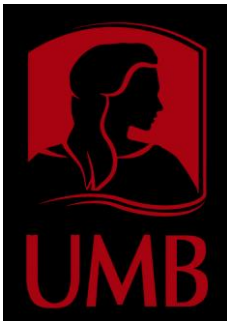
Este manual describe los pasos necesarios para cualquier persona que tenga ciertas bases de conocimiento en software, pueda realizar la implementación del sistema web creado para la librería “Entre líneas” desarrollado por el grupo de ingeniería web\_A3 de la UMB.

Es importante tener en cuenta que en el presente manual se hace mención a las especificaciones mínimas de hardware y software para la correcta implementación del sistema web

### **1.1 Objetivos**

#### **General**

Brindar la información necesaria para poder realizar la instalación y configuración



<b>GUIA MANUAL TÉCNICO DE UN SISTEMA WEB DE LA LIBRERÍA ENTRE LINEAS</b>	<b>Código:</b> ING_MT_01
	<b>Versión:</b> 1.0
	<b>Fecha:</b> Mayo 13 de 2025
	<b>Página</b> 2 de 21

del aplicativo.

### Específicos

- Representar la funcionalidad técnica de la estructura, diseño y definición

del aplicativo.

- Definir claramente el procedimiento de instalación del aplicativo.

- Detallar la especificación de los requerimientos de Hardware y Software

necesarios para la instalación de la aplicación.

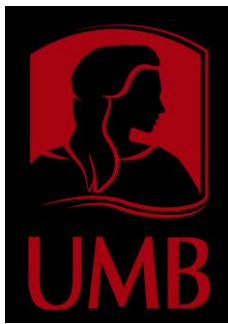
- Describir las herramientas utilizadas para el diseño y desarrollo del

prototipo.

## 1.2 Grupos involucrados

<b>Grupo</b>	<b>Front-End</b>
<b>Nombre</b>	Brayan Stiven Failach Durango Laura Valentina Gonzales Rojas Valentina Blanco Alvis Julián Mauricio Montoya Rodríguez Bryan Andrés Puentes Vargas
<b>Rol desempeñado</b>	Este grupo se encarga del diseño atractivo e intuitivo de las interfaces de usuarios.

<b>Grupo</b>	<b>Back-End</b>
<b>Nombre</b>	Hemer Santiago Perez Nieves

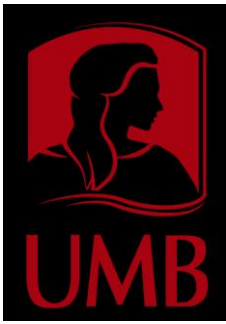


<b>GUIA MANUAL TÉCNICO DE UN SISTEMA WEB DE LA LIBRERÍA ENTRE LINEAS</b>	<b>Código:</b> ING_MT_01
	<b>Versión:</b> 1.0
	<b>Fecha:</b> Mayo 13 de 2025
	<b>Página</b> 3 de 21

	Miguel Angel Pinilla Baez Hugo Andres Forero Miguel Anguel Urrea Camacho Yesid Alejandro Varela Alfaro
<b>Rol desempeñado</b>	Este grupo se encarga del funcionamiento de los módulos tales como usuarios, carrito de compras, libros, inventario; A la par se encargan de crear la base de datos para el sistema web.

<b>Grupo</b>	<b>Integración</b>
<b>Nombre</b>	Miguel Ángel Romero Vaca La otra chica q no es asly El que ya perdió por fallas
<b>Rol desempeñado</b>	Este grupo se encarga de la conexión entre el Back y el Front buscando que estos interactúen de la forma correcta; a la par se encargan del despliegue del sistema web, para querer lograr una accesibilidad más sencilla por parte del usuario.

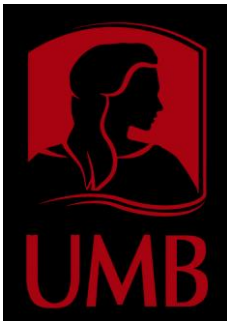
<b>Grupo</b>	<b>Documentación</b>
<b>Nombre</b>	Juan José Guerrero Ramírez Juan Sebastian Mendoza Moncada Mickey Collazos Serpa Sebastian Lopez Alvarez Asly Yulitza Camelo Arias
<b>Rol desempeñado</b>	Este grupo se encarga de documentar el proceso de planificación, elaboración y desarrollo del demás grupo referente a sistema web; A la par generara manuales de usuario y técnico, ayudando la comprensión de este, evidenciando errores que se generaron en el proceso.



<b>GUIA MANUAL TÉCNICO DE UN SISTEMA WEB DE LA LIBRERÍA ENTRE LINEAS</b>	Código: ING_MT_01
	Versión: 1.0
	Fecha: Mayo 13 de 2025
	Página 4 de 21

### 1.3 Definiciones, acrónimos y abreviaturas

Termino	Descripción
<b>Modelo Vista controlador</b>	<p>Es un patrón de arquitectura de software en el cual se dividen los procesos en tres “campos” los cuales son:</p> <p><b>Modelo:</b> Se encarga de operar la información del negocio, consultas, actualizaciones, accesos, en general toda la lógica del negocio.</p> <p><b>Vista:</b> Se encarga de presentar al “Modelo” de una manera de interacción al usuario</p> <p><b>Controlador:</b> Responde a las interacciones del usuario, realizando las peticiones al modelo y actualizando y/o modificando la vista.</p>
<b>Base de datos</b>	<p>Herramienta de almacenamiento de datos pertenecientes a un mismo contexto, sistematizada lógicamente para una futura recuperación, análisis o socialización</p>
<b>Base de datos relacional</b>	<p>Es un tipo de base de datos que alacena datos relacionados entre si de manera directa y con una identificación única para cada tabla</p>
<b>API</b>	<p>Es una pieza de código la cual permite la comunicación entre 2 aplicaciones permitiendo compartir datos y funciones entre sí.</p>
<b>API Fetch</b>	<p>Es un tipo de API la cual permite buscar y traer datos a</p>



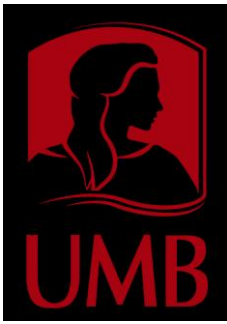
<b>GUIA MANUAL TÉCNICO DE UN SISTEMA WEB DE LA LIBRERÍA ENTRE LINEAS</b>	<b>Código: ING_MT_01</b>
	<b>Versión: 1.0</b>
	<b>Fecha: Mayo 13 de 2025</b>
	<b>Página 5 de 21</b>

través de “request” y “response”

**CSS**

Es un lenguaje enfocado a definir y formal la  
presentación de un documento HTML

---



# GUIA MANUAL TÉCNICO DE UN SISTEMA WEB DE LA LIBRERÍA ENTRE LINEAS

Código: ING\_MT\_01

Versión: 1.0

Fecha: Mayo 13 de 2025

Página 6 de 21

## 2 Descripción de la Construcción

En este capítulo se detalla el proceso de construcción del sistema, informando sobre tecnologías implementadas, arquitectura usada, fases de desarrollo y decisiones tomadas durante el proceso de construcción.

Buscando como objetivo facilitar el trabajo de mantenimiento y escalabilidad futuro a los próximos desarrolladores.

### 2.1 Entorno de desarrollo

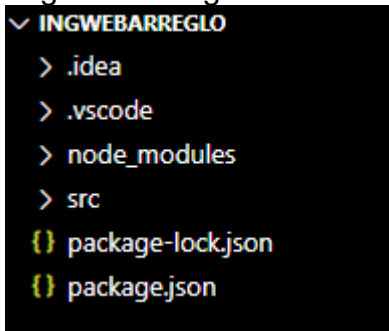
Entorno de desarrollo	Tecnología usada
Lenguajes utilizados	Front-End: HTML5, CSS y JavaScript Back-End: JavaScript (node.js)
Librerías adicionales	Express.js Morgan.js
Base de datos	Inicial: PostgreSQL Final: MYSQL
Sistema de control de versiones	Git, en el repositorio “proyectoingweb” alojado en GitHub
Entorno de desarrollo	Visual Studio Code, IntelliJ IDEA, Postman/insomnia



<b>GUIA MANUAL TÉCNICO DE UN SISTEMA WEB DE LA LIBRERÍA ENTRE LINEAS</b>	Código: ING_MT_01
	Versión: 1.0
	Fecha: Mayo 13 de 2025
	Página 7 de 21

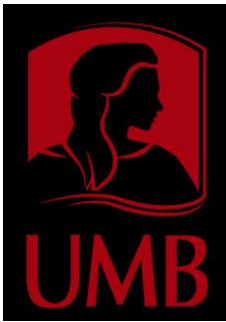
### 2.1.1

Figura 1  
Organización general de carpetas



Nota: Esta figura ilustra la distribución general de carpetas, en el proyecto “Entre líneas”

- En este proyecto se manejaron 2 carpetas principales “node\_modules”, “src “, las carpetas “.idea” y “.vscode” son configuraciones predeterminadas de los entornos en los cuales fue desarrollada la aplicación (Visual Studio Code, IntelliJ IDEA).
- La carpeta “node\_modules” contiene todas las dependencias (librerías y paquetes) que el proyecto necesita para funcionar, según lo definido en el archivo package.json. Esta carpeta se genera automáticamente al ejecutar el comando “npm install”, y contiene el código fuente de Node.js necesario para que el proyecto utilice esas librerías. No se debe modificar manual y normalmente no se incluye en el repositorio (ya que se puede regenerar con npm install).



<b>GUIA MANUAL TÉCNICO DE UN SISTEMA WEB DE LA LIBRERÍA ENTRE LINEAS</b>	Código: ING_MT_01
	Versión: 1.0
	Fecha: Mayo 13 de 2025
	Página 8 de 21

**Figura 2**  
**Distribución carpeta “src”**



**Nota: Esta figura ilustra la distribución de la carpeta “src”, en el proyecto “Entre líneas”**

- Dentro de la carpeta “src” se encuentra el núcleo del proyecto, dividido en componentes específicos. La subcarpeta “backend” corresponde a la implementación del back-end, encargada de gestionar la lógica del servidor, la comunicación con la base de datos y la respuesta a las peticiones del cliente.
- Por otro lado, la carpeta frontend reúne todos los archivos relacionados con la interfaz del usuario. Esta fue desarrollada utilizando HTML, CSS y JavaScript, proporcionando una experiencia visual clara e interactiva para los usuarios.

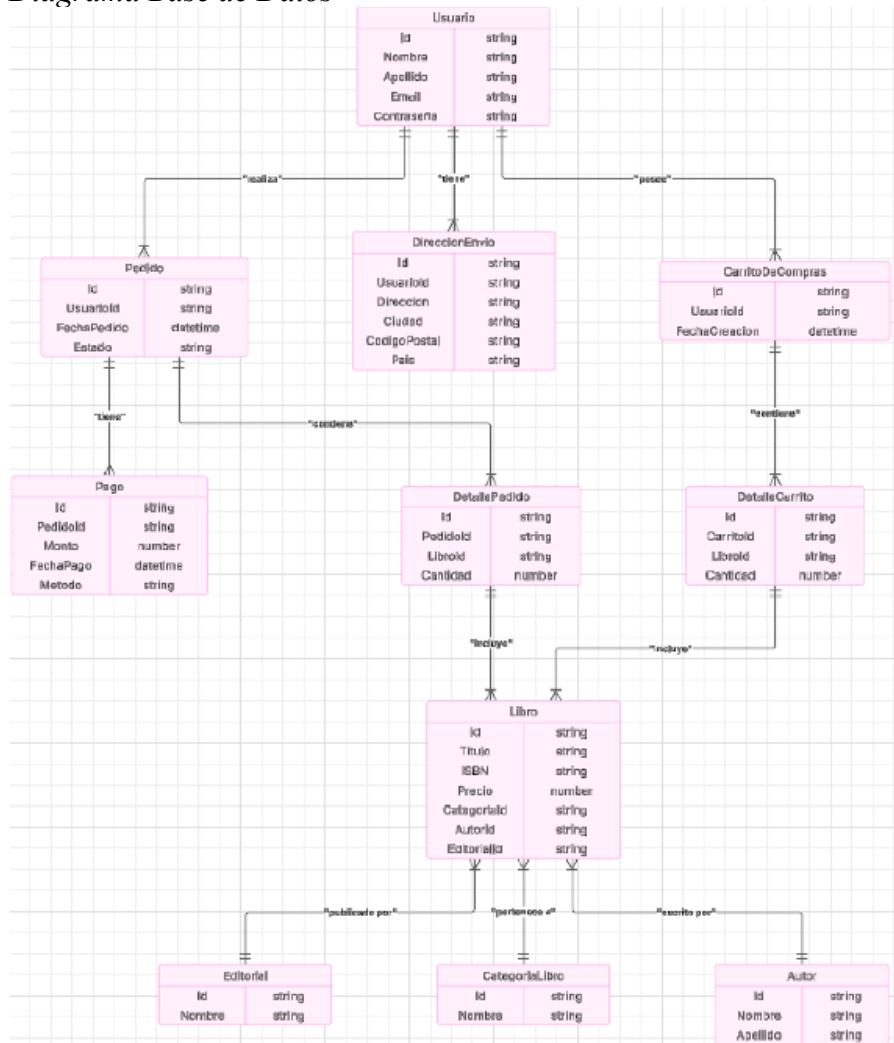




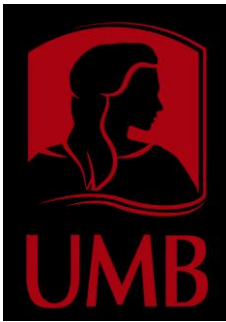
<h1 style="text-align: center;">GUIA MANUAL TÉCNICO DE UN SISTEMA WEB DE LA LIBRERÍA ENTRE LINEAS</h1>	Código: ING_MT_01
	Versión: 1.0
	Fecha: Mayo 13 de 2025
	Página 9 de 21

## 2.1.2 Diseño a Modelo Relacional

**Figura 2**  
*Diagrama Base de Datos*



*Nota: Autoría propia hecho con DataGrip IDE, diagrama de base de datos*



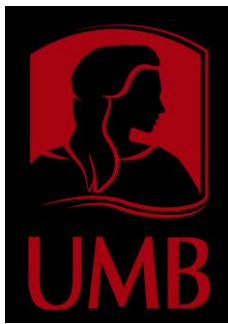
<b>GUIA MANUAL TÉCNICO DE UN SISTEMA WEB DE LA LIBRERÍA ENTRE LINEAS</b>	Código: ING_MT_01
	Versión: 1.0
	Fecha: Mayo 13 de 2025
	Página 10 de 21

### 2.1.3 Normalización

- El modelo entidad-relación presentado cumple con los principios de normalización hasta la Tercera Forma Normal (3FN). Todas las tablas contienen atributos atómicos y no repetitivos, asegurando la integridad estructural del modelo (1FN). Además, los atributos no clave dependen completamente de sus respectivas claves primarias, lo que elimina dependencias parciales (2FN).
- Asimismo, se evita la presencia de dependencias transitivas al separar entidades como Autor, Editorial y CategoríaLibro, y referenciarlas mediante claves externas en la tabla Libro. Esta organización permite un diseño más limpio, reduce la redundancia de datos y facilita el mantenimiento y escalabilidad del sistema.

### 2.1.4 Diccionario de Datos

TABLA	ATRIBUTOS
USUARIO	<b>Id</b> <b>Nombre</b> <b>Apellidos</b> <b>Email</b> <b>Password</b> <b>Telefono</b>
PEDIDO	<b>ID_pedido</b> <b>Fecha_pedido</b> <b>Estado</b> <b>FK_User</b>
PAGO	<b>ID_Pago</b> <b>Fecha_pago</b> <b>Metodo</b> <b>Monto</b> <b>FK_Pedido_ID</b>
DETALLE_PEDIDO	<b>Detalle_pedido</b> <b>ISBN_Libro</b>



<b>GUIA MANUAL TÉCNICO DE UN SISTEMA WEB DE LA LIBRERÍA ENTRE LINEAS</b>	Código: ING_MT_01
	Versión: 1.0
	Fecha: Mayo 13 de 2025
	Página 11 de 21

**DIRECCION\_ENVIO**

**ID\_libro  
Cantidad  
FK\_pedido  
  
ID\_Direccion  
Pais  
Ciudad  
Direccion  
Codigo\_postal  
FK\_User\_DIR**

**CARRITO\_COMPRAS**

**ID\_Carrito  
Fecha\_creacion  
Precio\_total**

**DETALLE\_CARRITO**

**ID\_Detalle\_carro  
ID\_Libro  
Cantidad  
FK\_Carro\_ID**

**EDITORIAL**

**ID\_Editorial  
Nombre\_Editorial**

**CATEGORIA\_LIBRO**

**ID\_Categoria  
Nombre\_Categoria**

**AUTOR**

**ID\_Autor  
Nombre\_Autor  
Apellido\_Autor**

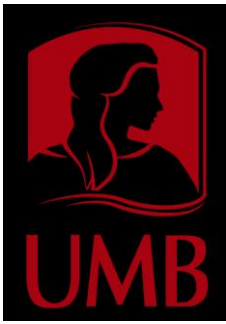
### **2.1.5 Scripts de creación de B.D.**

El script se encuentra alojado en el repositorio de GitHub “**proyectoingweb**”, específicamente en la ruta /INGWEBARREGLO/src/backend/Dump20250513.sql

## **2.2 Aplicación / Sistema**

En este apartado se expondrá la implementación de las tecnologías nombradas previamente

### **2.2.1 Backend**



<b>GUIA MANUAL TÉCNICO DE UN SISTEMA WEB DE LA LIBRERÍA ENTRE LINEAS</b>	Código: ING_MT_01
	Versión: 1.0
	Fecha: Mayo 13 de 2025
	Página 12 de 21

**Figura 3**  
**Importaciones**

```
const express : e | () => core.Express = require('express');  
const mysql : {...} = require('mysql2');  
const cors : function(any): function(any, a... | {...} = require('cors');  
const path : PlatformPath | path = require('path');
```

**Nota:** *Figura de autoría propia*

- Importaciones usadas por server.js para conectarse a la base de datos

#### **Express**

- Crea el servidor y define rutas (/registro, /login, etc.).

#### **Mysql2**

- Conecta y hace consultas a la base de datos MySQL, explícitamente con MariaDB.

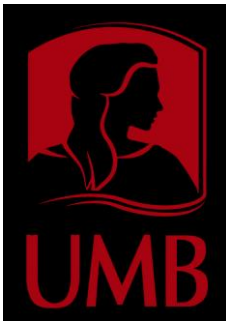
#### **Cors**

- Permite que el Front-End (otro puerto o dominio) se comuniquen con el Back-End sin bloqueos del navegador.

#### **Path**

- Maneja rutas de archivos de forma segura para servir HTML u otros archivos estáticos.

**Figura 4**  
**Conexión BD**



<b>GUIA MANUAL TÉCNICO DE UN SISTEMA WEB DE LA LIBRERÍA ENTRE LINEAS</b>	Código: ING_MT_01
	Versión: 1.0
	Fecha: Mayo 13 de 2025
	Página 13 de 21

```
// Conexión MySQL
const db : Connection = mysql.createConnection( config: {
  host: 'localhost',
  user: 'maple',
  password: 'maple',
  database: 'libreriadb'
});

db.connect( callback: err : QueryError | null => {
  if (err) {
    console.error('Error de conexión a la base de datos:', err);
    return;
  }
  console.log('Conectado a la base de datos MySQL');
});
```

***Nota: Autoría propia***

- Creamos una conexión nueva con el createConnection usando las credenciales apropiadas, en este caso se utilizamos las credenciales del servidor que esta hosteando la página, las cuales se pueden evidenciar en la imagen.
- Luego conecta el Back-End a la base de datos MySQL a través del método “connect”. Si la conexión falla, muestra un error; si tiene éxito en la conexión, confirma que está conectado.

***Figura 4***  
Registro



# GUIA MANUAL TÉCNICO DE UN SISTEMA WEB DE LA LIBRERÍA ENTRE LINEAS

Código: ING\_MT\_01

Versión: 1.0

Fecha: Mayo 13 de 2025

Página 14 de 21

```
app.post( path: '/registro', handlers: (req : Request<P, ResBody, ReqBody, ReqQuery, LocalsObj> , res : Response<ResBody, LocalsObj> ) : void => {
  const { nombre, apellidos, email, password, telefono } = req.body;

  const query :string = 'INSERT INTO usuarios (nombre, apellidos, email, password, telefono) VALUES (?, ?, ?, ?, ?)';
  db.query(query, [nombre, apellidos, email, password, telefono], (err : QueryError|null , result : T ) :any|undefined => {
    if (err) {
      console.error('Error al registrar usuario:', err);
      console.error('Error', result)
      return res.status( code: 500).json( body: { mensaje: 'Error al registrar' });
    }
    res.json( body: { mensaje: 'Usuario registrado exitosamente' });
  });
});
```

*Nota: Autoría propia*

Cuando el Front envía los datos al Back a través de registro.js, esos datos se insertan en una query, la cual se envía a través de una petición al MYSQL para añadir el nuevo usuario a la base de datos.



<b>GUIA MANUAL TÉCNICO DE UN SISTEMA WEB DE LA LIBRERÍA ENTRE LINEAS</b>	Código: ING_MT_01
	Versión: 1.0
	Fecha: Mayo 13 de 2025
	Página 15 de 21

**Figura 5**  
**Login**

```
// Ruta para login
app.post( path: '/login', handlers: (req :Request<P, ResBody, ReqBody, ReqQuery, LocalsObj> , res :Response<ResBody, LocalsObj> ) :any | undefined => {
  const { email, password } = req.body;

  if (!email || !password) {
    return res.status( code: 400 ).json( body: { error: 'Todos los campos son obligatorios' } );
  }

  const query :string = 'SELECT * FROM usuarios WHERE email = ? AND password = ?';

  db.query(query, [email, password], (err :QueryError | null , results :T ) :any | undefined => {
    if (err) {
      console.error('Error al hacer login:', err);
      return res.status( code: 500 ).json( body: { error: 'Error del servidor' } );
    }

    if (results.length > 0) {
      res.json( body: { message: 'Inicio de sesión exitoso' } );
    } else {
      res.status( code: 401 ).json( body: { error: 'Correo o contraseña incorrectos' } );
    }
  } );
});
```

*Nota: Autoría propia*

Luego de recibir los datos de Login.js, verifica si el usuario existe en la base de datos con el correo y contraseña. Si falta algún dato o hay error, lo indica; si las credenciales son correctas, permite el inicio de sesión.

**Figura 6**  
**Inicio servidor**

```
// Iniciar servidor
app.listen(PORT, hostname: () :void => {
  console.log(`Servidor corriendo en http://localhost:${PORT}`);
});
```

*Nota: Autoría propia*

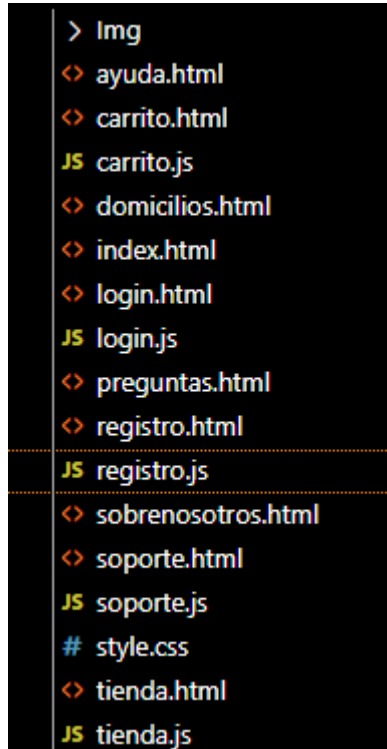
Inicia el servidor en el puerto indicado y muestra un mensaje confirmando que está en funcionamiento.

## 2.2.2 Front-End



<b>GUIA MANUAL TÉCNICO DE UN SISTEMA WEB DE LA LIBRERÍA ENTRE LINEAS</b>	Código: ING_MT_01
	Versión: 1.0
	Fecha: Mayo 13 de 2025
	Página 16 de 21

**Figura 7**  
**Diseño Front**



*Nota: Autoría propia*

- En la parte de Front se desarrollaron en total 10 pantallas de usuario, de las cuales solo login y registro tienen conexión con el back, las demás se manejaron en local-storage.

**Figura 8**  
**Front-login.js**





# GUIA MANUAL TÉCNICO DE UN SISTEMA WEB DE LA LIBRERÍA ENTRE LINEAS

Código: ING\_MT\_01

Versión: 1.0

Fecha: Mayo 13 de 2025

Página 17 de 21

```
1 // frontend/login.js
2 document.getElementById('login-form').addEventListener('submit', async function (e) {
3     e.preventDefault();
4
5     const email = document.getElementById('email').value.trim();
6     const password = document.getElementById('password').value.trim();
7
8     try {
9         const response = await fetch('/login', {
10             method: 'POST',
11             headers: {
12                 'Content-Type': 'application/json'
13             },
14             body: JSON.stringify({ email, password })
15         });
16
17         const data = await response.json();
18
19         if (response.ok) {
20             localStorage.setItem('logueado', 'true'); // <-- Agrega esta línea
21             alert(data.message);
22             window.location.href = 'tienda.html';
23             return; // Detiene la ejecución después de redirigir
24         } else {
25             alert(data.error || 'Error al iniciar sesión');
26         }
27     } catch (error) {
28         console.error('Error al hacer login:', error);
29         alert('Error al hacer login');
30     }
31 });
```

*Nota: Autoría propia*

- Se usa el método `getElementById` para obtener los datos del front, luego a través de una petición `fetch` se envían al back, esperando una respuesta para redireccionar o rechazar el ingreso. Este mismo procedimiento se implementa para el apartado de registro

## 2.2.3 Despliegue o Instalación del Sistema

### Descripción General

En la parte de Despliegue se hizo uso de un mini servidor del compañero Juan Jose Guerrero, en este



<b>GUIA MANUAL TÉCNICO DE UN SISTEMA WEB DE LA LIBRERÍA ENTRE LINEAS</b>	Código: ING_MT_01
	Versión: 1.0
	Fecha: Mayo 13 de 2025
	Página 18 de 21

caso se cuenta con la gestión del DNS y del Hosting por medio de [cloudflare.com](https://cloudflare.com), en este caso se asignó el host .tech teniendo de nombre maplehugs.tech, este servicio es gratuito para la página la cual se está trabajando.

En la figura 39 se puede observar la creación de diferentes dominios, en este caso se centra en maplehugs.tech, el cual está de primeras para uso de la página web “Entre Líneas”.

**Figura 9**  
**Cloudflare para despliegue**

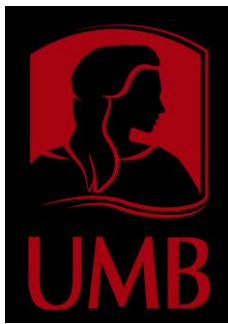
DNS management for <b>maplehugs.tech</b>						
Review, add, and edit DNS records. Edits will go into effect once saved.						
DNS Setup: Full ⓘ Import and Export ▾ ⚙ Dashboard Display Settings						
Search DNS Records						
<div>Add filter</div> <div>Q</div> <div>Search</div> <div>+ Add record</div>						
<input type="checkbox"/>	Type ⓘ	Name ⓘ	Content ⓘ	Proxy status ⓘ	TTL ⓘ	Actions
<input type="checkbox"/>	CNAME	maplehugs.tech	maplehugs.ddns.net	DNS only	Auto	Edit ▶
<input checked="" type="checkbox"/>	MX	maplehugs.tech	route3.mx.cloudflare.net	17 DNS only	Auto	Edit ▶
<input checked="" type="checkbox"/>	MX	maplehugs.tech	route2.mx.cloudflare.net	99 DNS only	Auto	Edit ▶
<input checked="" type="checkbox"/>	MX	maplehugs.tech	route1.mx.cloudflare.net	51 DNS only	Auto	Edit ▶
<input type="checkbox"/>	SRV	_minecraft_tcp	0 10 25565 maplehugs.ddn... 0	DNS only	Auto	Edit ▶
<input checked="" type="checkbox"/>	TXT	cf2024-1_domainkey	"v=DKIM1; h=sha256; k=rsa;...	DNS only	Auto	Edit ▶
<input type="checkbox"/>	TXT	_discord	"dh=548a5b1b1a5b839f4eb...	DNS only	Auto	Edit ▶
<input type="checkbox"/>	TXT	maplehugs.tech	"v=spf1 include:_spf.mx.clou...	DNS only	Auto	Edit ▶

**Nota:** *Página Cloudflare con el dominio de la página, además de la gestión de otros dominios como ejemplo de su funcionamiento, página con dominios gratuitos.*

### Entorno de Despliegue

En la parte del entorno del despliegue en caso del servidor, se utiliza un servidor con Arch Linux, Host optiPlex 7050, de CPU utiliza un Intel Core I5-6500, GPU Intel HD Graphics 530, Memoria de 16 GB de Ram, además de un disco de 237.36 GB, en la Figura 10 se muestran las especificaciones del servidor por consola.

**Figura 10** *Especificaciones del servidor de alojamiento*



<b>GUIA MANUAL TÉCNICO DE UN SISTEMA WEB DE LA LIBRERÍA ENTRE LINEAS</b>	Código: ING_MT_01
	Versión: 1.0
	Fecha: Mayo 13 de 2025
	Página 19 de 21

```
[maple@maple ~]$ fastfetch

      _
     .o+
    'ooo/
   +oooo:
  +oooooo:
 -+ooooooo+:
  /:-++oooo+:
 /++++/+++++:
 /+++++++:
 /+++++ooooooo/
 ./ooooSSSSO++oSSSSSSO+
 .oSSSSSSO-''''/oSSSSSS+
 -oSSSSSSO.      :SSSSSSSO.
 :oSSSSSSS/      oSSSSO+++.
 /oSSSSSSS/      +SSSSOoo/-
 /oSSSSSO+/-     -:/+oSSSSO+-
 +SSO+:-         .-/+oSO:
 ++:             -/+//
               -/+//
               /

[maple@maple ~]$ |
```

```
maple@maple
-----
OS: Arch Linux x86_64
Host: OptiPlex 7050
Kernel: Linux 6.12.29-1-lts
Uptime: 5 days, 8 hours, 55 mins
Packages: 603 (pacman)
Shell: bash 5.2.37
Terminal: /dev/pts/0
CPU: Intel(R) Core(TM) i5-6500 (4) @ 3.60 GHz
GPU: Intel HD Graphics 530 @ 1.05 GHz [Integrated]
Memory: 6.24 GiB / 15.48 GiB (40%)
Swap: 0 B / 4.00 GiB (0%)
Disk (/): 17.83 GiB / 237.36 GiB (8%) - xfs
Local IP (enp0s31f6): 192.168.1.87/24
Locale: en_US.UTF-8
```

**Nota:** Características del servidor de alojamiento utilizado para la página.

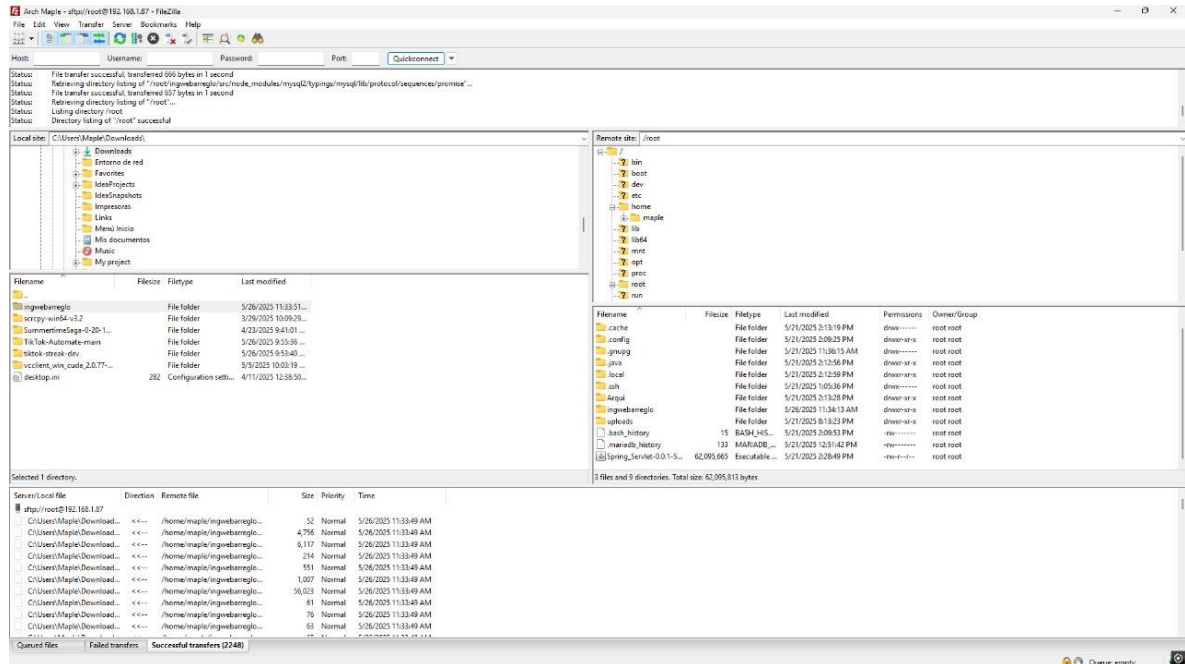
Para subir los archivos, carpetas de Front, Back y los paquetes Json de integración se movieron estos paquetes al servidor por medio de FileZilla el cual está conectado al servidor, se mueven los archivos dentro de su interfaz para subirlos al servidor, este servidor quedará encendido y por tanto dejará funcionando la página las 24 horas durante los días requeridos, en la figura 41 se muestra la interfaz de FileZilla, la cual tiene los archivos de la página web para el despliegue.

Igualmente se inicializa el archivo y sistema del servidor por medio de comandos del Command Center, donde se conecta por medio de ssh con el nombre del administrador más la dirección IP asignada al servicio, una vez conectado, se ingresa a la carpeta utilizada para el despliegue denominada "ingwebarreglo/src/" cogiendo todo lo correspondiente en la carpeta, finalmente, ingresando el comando "npm start" para inicializar el servicio, esto se visualiza en la Figura 42.

**Figura 11** Interfaz de FileZilla

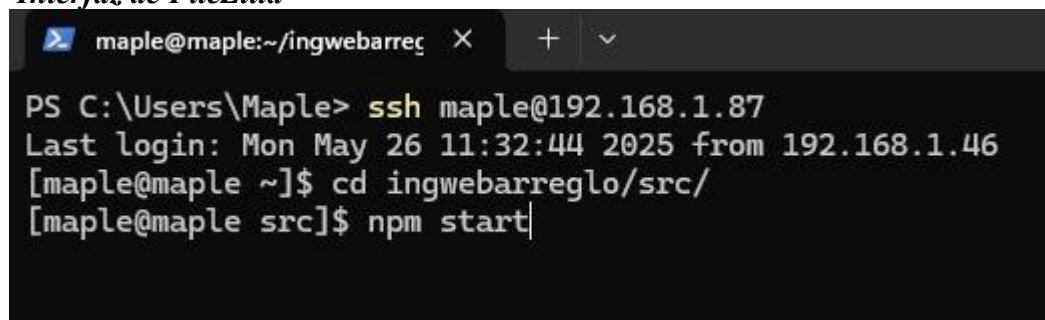


Código: ING_MT_01
Versión: 1.0
Fecha: Mayo 13 de 2025
Página 20 de 21



**Nota:** Archivos subidos al servidor por medio de FileZilla donde se ejecutan de manera constante, junto con los formatos Json para la conexión de bases de datos.

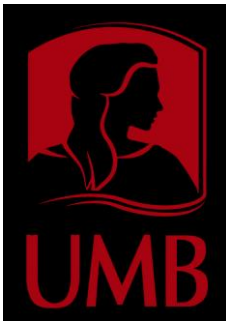
**Figura 12**  
**Interfaz de FileZilla**



**Nota:** Powershell utilizado para realizar la conexión del servidor, ingresar a la carpeta desplegada y al final ejecutar la carpeta.

## 2.2.4 Código Fuente

- El código fuente se encuentra alojado en el repositorio de GitHub “proyectoingweb”



<b>GUIA MANUAL TÉCNICO DE UN SISTEMA WEB DE LA LIBRERÍA ENTRE LINEAS</b>	Código: ING_MT_01
	Versión: 1.0
	Fecha: Mayo 13 de 2025
	Página 21 de 21

### 3 Observaciones Adicionales

- Falta implementar un back funcional para sistemas como compras y CRUD de libros
- Las sesiones del usuario no se mantienen activas, se recomienda usar JWT para solucionar este inconveniente
- La sobrecarga de código en el archivo CSS, genera dificultades para modificar el aspecto de las pantallas
- Falta la implementación de herramientas que permita modificar “headers” o “divs” de manera general y no manual como en este caso.
- Se considera que la base de datos contiene tablas extra, las cuales podrían ser reducidas en una sola.