UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA

FUNDAMENTOS DE BASES DE DATOS

Proyecto Final

Manual Técnico y de Usuario

Alumnos / Matriculas:

- Danna Maribel Corral Salcedo / 358147
- Emiliano Rivera Armendariz / 358193
- Héctor Daniel Medrano Meza / 361345

Docente: De Lira Montes Jose Saul

Fecha: 2024-05-29

Tema: IV - Diseño de Bases de Datos

Proyecto Final

Manual Técnico

1. Sistema Aplicación

a. Nombre del sistema

Libroteca

b. Descripción y delimitación del sistema

Libroteca es una aplicación de gestión de bibliotecas diseñada para automatizar y simplificar todas las tareas relacionadas con la administración de libros, usuarios, préstamos y devoluciones en una biblioteca.

c. Objetivo general

Facilitar la gestión eficiente de una biblioteca, permitiendo a los bibliotecarios llevar un registro preciso de los libros disponibles, los usuarios y sus préstamos, así como agilizar el proceso de préstamo y devolución de libros.

d. Objetivos específicos

- Llevar a cabo un sistema donde puedan registrarse diversos usuarios relevantes a la biblioteca, dentro de los cuales figuran clientes y bibliotecarios.
- Implementar un sistema de registro detallado para cada libro individual dentro de la biblioteca, donde pueda documentarse y consultarse el estado particular de cada libro.

- Facilitar el sistema de préstamos y devoluciones de los libros dentro de la biblioteca.
- Implementar un sistema de seguimiento y registro de multas para los casos en los que haya retrasos en la devolución de libros.

e. Descripción de tipos de usuarios

El usuario principal al que está dirigida esta aplicación es a administradores de bibliotecas, quienes harán uso de la aplicación para llevar un registro de sus trabajos diarios.

f. Entorno operativo del sistema

- Sistemas Operativos Compatibles:
  - Windows
  - Linux
- Software Necesario:
  - NodeJS
  - Oracle Database 21c
  - SQL Developer

2. Especificación de requerimientos

a. Requerimientos funcionales

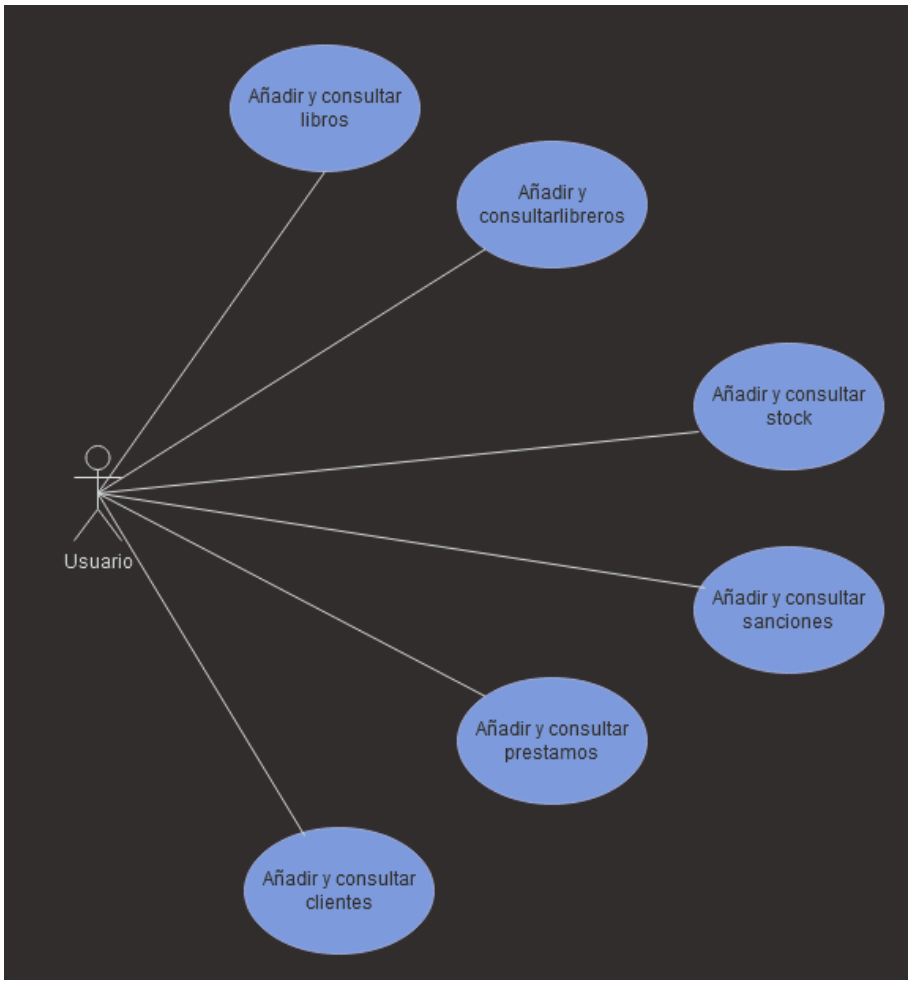
- Se debe permitir añadir y consultar el catálogo.
- Se debe proporcionar la funcionalidad de añadir y consultar los libros del stock de la biblioteca.
- Se debe tener la posibilidad de añadir y consultar los clientes y bibliotecarios.

- Se debe tener la posibilidad de añadir y consultar los préstamos y las multas.

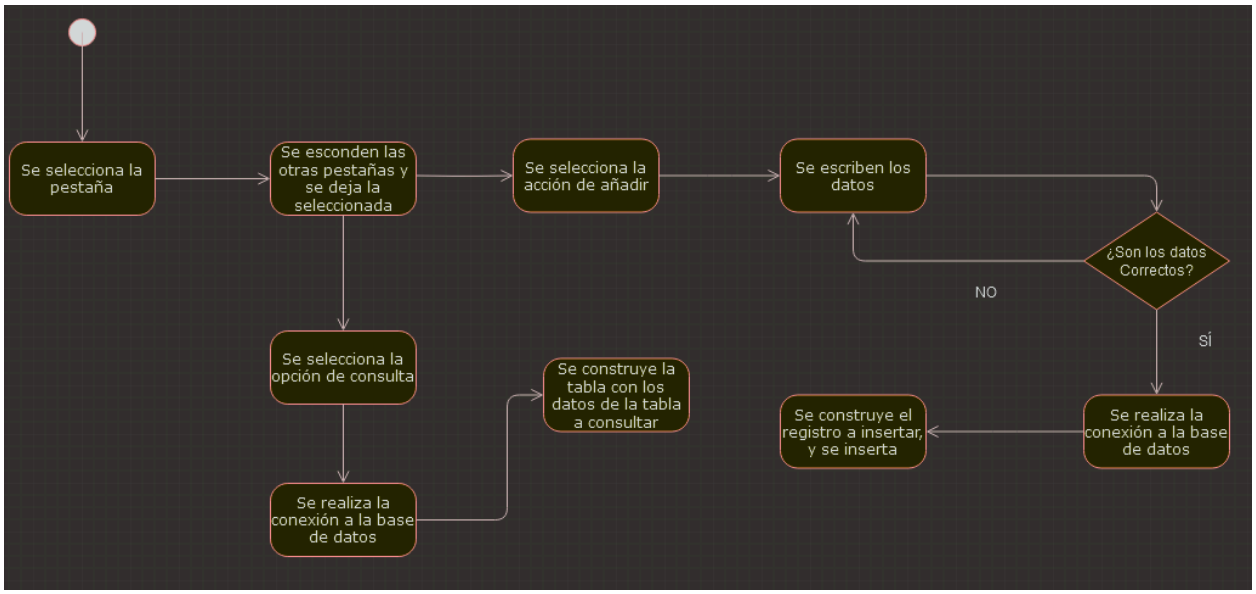
b. Modelado del sistema

i. Diagrama de clases

ii. Diagrama de casos de uso



iii. Diagrama de actividades



c. Requerimientos no funcionales

- El sistema debe de ser compatible con múltiples navegadores web.
- Se debe de tener un buen rendimiento, capaz de procesar consultas en cortos periodos de tiempo.
- Debe de tenerse una interfaz de usuario básica, limpia, entendible y fácil de utilizar

3. Arquitectura del Sistema/Aplicación.

a. Layer/Tier's

- Capa de Presentación:

Esta capa se ve representada por el front-end del programa, el cual es una interfaz clara y sencilla de utilizar donde residen todas las funcionalidades de la aplicación. Creada haciendo uso de HTML y CSS.

- Capa de Lógica de Negocios:

El programa se encarga de revisar si los datos introducidos son o no correctos mediante funcionalidades básicas de HTML, las cuales se encargan de asegurarse que el usuario introdujo tipos de datos correctos. Más al respecto dentro de la sección de *Lógica de Negocios*.

- Capa de Acceso a Datos:

Después de que la lógica interna del programa ha decidido que el input es correcto, se accede a la base de datos mediante un método POST, el cual realiza la tarea de mandar un archivo JSON como forma de comando SQL, el cual la base de datos es capaz de recibir y convertir en una entrada más.

b. Frontback/Backend

Frontend

El front-end está realizado con *HTML* y *CSS*, sin hacer uso de ningún framework. Para la realización de este, se utilizaron múltiples formularios correspondientes a sus respectivas pestañas (las cuales se muestran del lado izquierdo dentro de la página) en todo momento, las cuales proporcionan una manera intuitiva y sencilla de utilizar la aplicación. Para salir hay un botón en la parte superior que permite des-logearse.

Backend

En el caso del Backend, este fue hecho con JavaScript y NodeJS. La base de datos utilizada es Oracle Express Edition 21c, a través de SQL Developer.

Para realizar inserciones dentro de la base de datos, se hace uso de un fetch, el cual sirve como conexión a la base de datos; llama a un método POST, mediante el cual se manda un

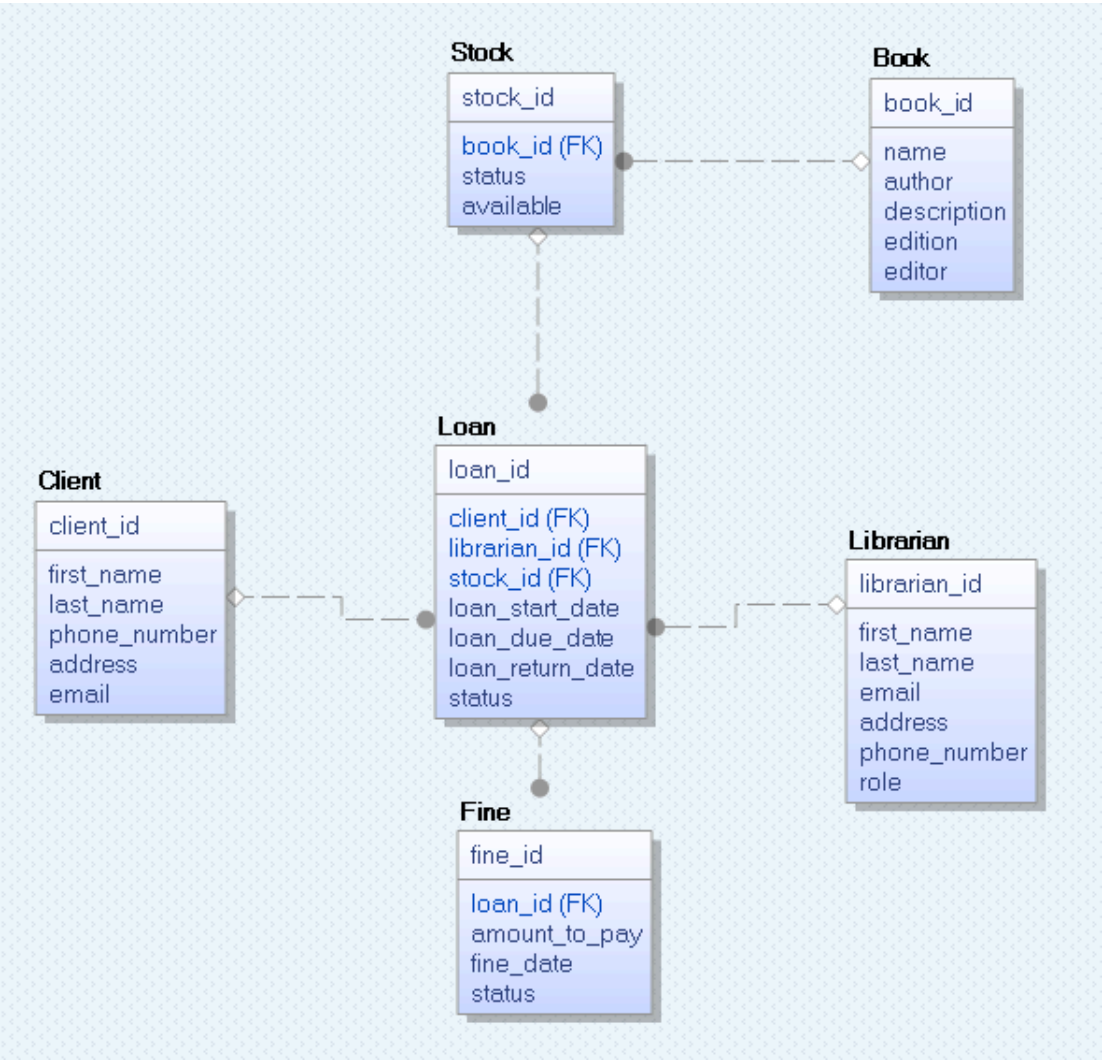
archivo JSON con todos los datos respectivos a la tabla a la que se está agregando, y de dicha manera se agrega una entrada a la tabla.

Para mostrar la tabla se hace uso de un GET para mostrar la información requerida mediante tablas.

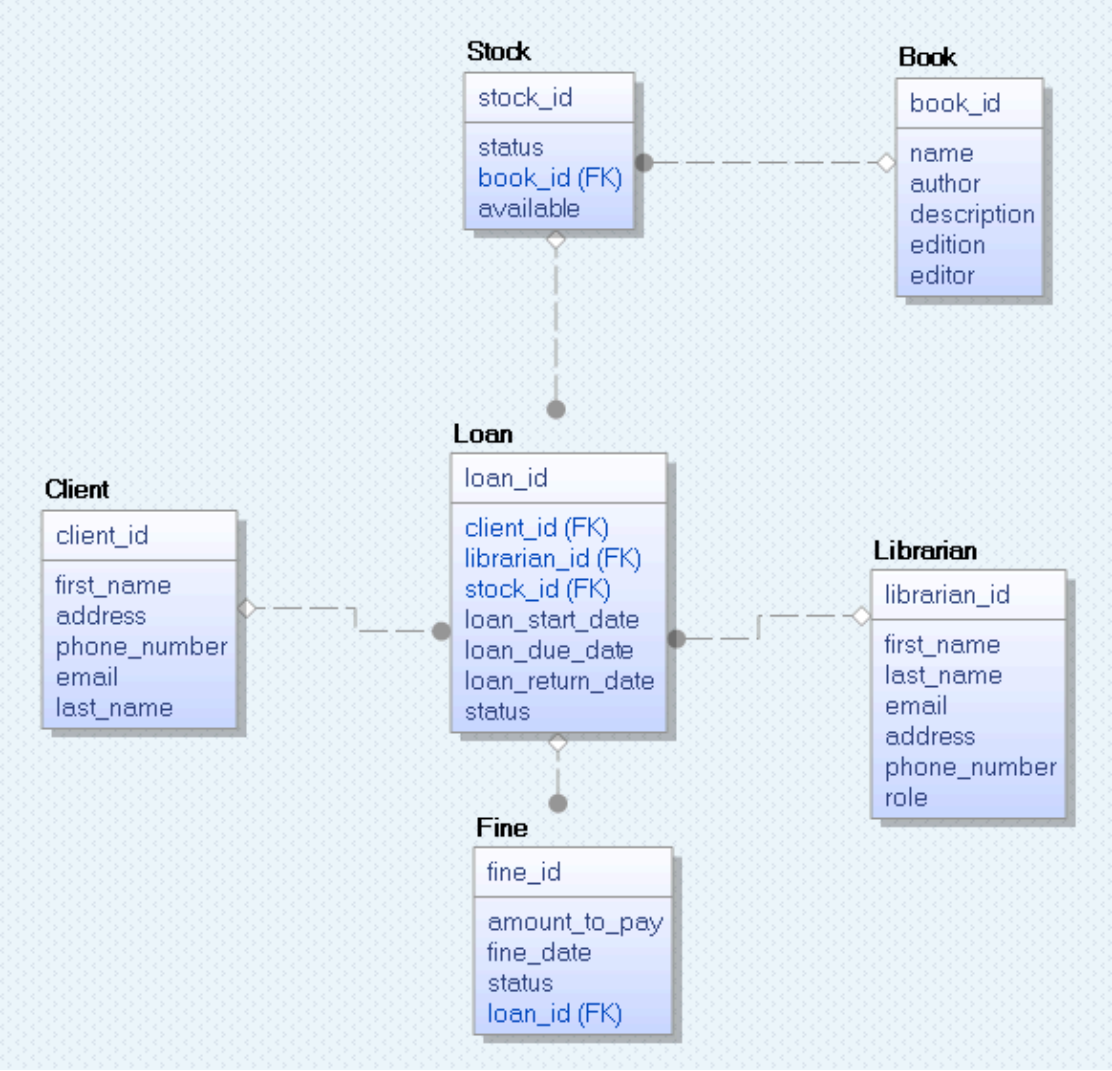
4. Base de datos

a. Diagrama Conceptual (Entidad/Relación)

Esquema Lógico



Esquema Físico



b. Esquema Lógico de la Base de Datos

i. Especificación Tablas (Normalizadas hasta BCNF)

Hay 6 tablas dentro del proyecto (todas normalizadas hasta BCNF), entre ellas están:

- Book
    - book_id: No. de identificación del libro.
    - name: Nombre del libro.
    - author: Nombre del autor.
    - description: Descripción del libro.
    - edition: Número de edición.
    - editor: Nombre del editor.
- Stock
    - stock_id: No. de identificación del libro particular en el stock.
    - book_id (FK): No. de identificación del libro al que pertenece.
    - status: Descripción de la condición del libro.
    - available: Si se encuentra o no disponible.
- Client
    - client_id: No. de identificación del cliente.
    - first_name: Primer nombre del cliente.
    - last_name: Apellido del cliente.
    - phone_number: Número de teléfono del cliente.
    - address: Dirección del cliente.
    - email: Dirección de correo electrónico del cliente.
- Librarian
    - librarian_id: No. de identificación del bibliotecario.
    - first_name: Primer nombre del bibliotecario.
    - last_name: Apellido del bibliotecario.

    - email: Dirección de correo electrónico del bibliotecario.
    - address: Dirección del bibliotecario.
    - phone_number: Número de teléfono del bibliotecario.
    - role: Rol dentro de la biblioteca del bibliotecario.
- Loan
    - loan_id: No. de identificación del prestamo.
    - client_id (FK): No. de identificación del cliente que solicita el prestamo.
    - librarian_id (FK): No. de identificación del bibliotecario encargado del prestamo.
    - stock_id (FK): No. de identificación de los libros del stock que fueron prestados.
    - loan_start_date: Fecha en la que se hizo el prestamo.
    - loan_due_date: Fecha en la que se debió de haber devuelto el prestamo.
    - loan_return_date: Fecha en la que se devolvió el libro.
    - status: Estado del libro (si fue o no devuelto aun).
- Fine
    - fine_id: No. de identificación de la sanción.
    - loan_id: No. del prestamo al que está asociado la sanción.
    - amount_to_pay: Cantidad a pagar por la sanción.
    - fine_date: Fecha en la que la sanción se pagó.
    - status: Estado de la sanción (si se pagó o no).

ii. Integridad de Datos (Constraints)

Dentro de las tablas únicamente existen constraints de tipo NOT NULL para las llaves primarias.

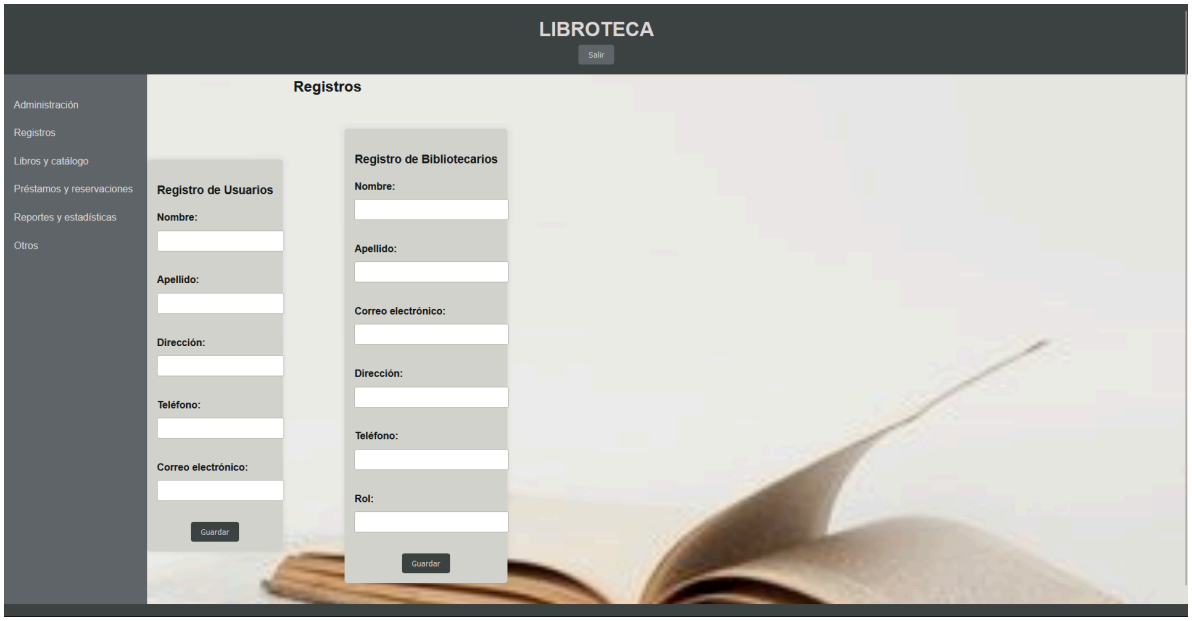
5. Lógica/Reglas del Negocio

- **Campos Obligatorios**: Todos los formularios deben validar que los campos obligatorios no estén vacíos antes de permitir la presentación del formulario.
    - **Regla Específica**: Si un usuario intenta enviar un formulario con campos obligatorios vacíos, el sistema debe mostrar un mensaje de error indicando que dichos campos deben ser completados.
- **Notificación de Errores**: En caso de que algún campo obligatorio esté vacío, se debe proporcionar retroalimentación inmediata y clara al usuario.
    - **Regla Específica**: Los mensajes de error deben ser visibles y estar ubicados cerca del campo correspondiente o en un área designada para mensajes de error.

6. Descripción Interfaz de la aplicación

Al entrar por primera vez dentro de la aplicación se nos son presentadas las pestañas del lado izquierdo, las cuales son aquellas de las que haremos uso para la manipulación de datos que nosotros queramos. A su vez, existe un botón en la parte superior que permite des-logearse.

Para añadir información se utilizan cualquiera de las pestañas anteriormente mencionadas. El funcionamiento es bastante simple e intuitivo, y la propia página web va guiando al usuario durante el proceso de inserción. Tras ello, proporcionará un mensaje sobre si hubo éxito o no.



Existe otra pestaña relevante para nuestra página, refiriéndonos específicamente a la de *Administración*, donde podemos consultar la información de nuestra base de datos.

7. Descripción de reglas de seguridad(acceso/operación)

Actualmente, la página web no implementa medidas específicas de seguridad.

8. Acceso a la Base de Datos mediante la Aplicación

La base de datos es accesada dentro del programa en dos ocasiones distintas:

- **Al insertar:** Se accede mediante un método fetch, el cual envía una solicitud HTTP al servidor. El método de solicitud es un POST, el cual envía datos en formato JSON, los cuales constituyen el nuevo registro a insertar. Finalmente la respuesta del servidor se maneja mediante promesas. En caso de errores, se manejan dependiendo de su caso específico.
- **Al consultar:** Se accede mediante un método fetch, el cual envía una solicitud HTTP al servidor. A partir de aquí, mediante promesas, se maneja la solicitud, y la

respuesta se convierte en un formato JSON, el cual luego es leído por el programa y convertido en una tabla HTML. En caso de errores, se manejan dependiendo de su caso específico.

9. Conclusión

El programa desarrollado proporciona una funcionalidad básica pero esencial para la gestión de usuarios a través de una interfaz web sencilla. Aunque, si bien su funcionalidad simple cumple con varios de los requerimientos básicos mínimos necesarios para el funcionamiento de una biblioteca, aun requiere de mucho trabajo para poder ser utilizada de manera realista; esto debido a que múltiples funcionalidades fundamentales se encuentran ausentes y deberían de ser implementadas aun.

Manual Técnico

Descripción técnica de la operación del sistema o aplicación

Nuestro sistema permite gestionar una biblioteca, permitiendo registrar bibliotecarios, libros, clientes, prestamos, stocks y multas. Utiliza Node.js con Express para el backend y OracleDB para la base de datos

Componentes

1. Servidor (server,js)

Express: Framework para manejar solicitud HTTP

OracleDB: Conexion a la base de datos Oracle

BodyParser: Middleware para parsear datos JSON

2. Cliente (script.js)

Funciones JavaScript: Manejan eventos en la interfaz de usuario y envian solicitudes al servidor

3. Base de Datos

Tablas: client, librarian, book, loan, stock y fine

Funcionalidades

- Autenticacion

Verifica las credenciales del administrador para otorgar acceso

- Registro

Clientes: Nombre, apellido, dirección, telefono, email

Bibliotecarios: Nombre, apellido, email, dirección, teléfono, rol

Libros: Nombre, autor, descripción, edición, editor

- Prestamos

Registra un préstamo con el cliente, bibliotecario, libro, fechas y estado

- Stock

Registra la disponibilidad de un libro con un estado (disponible o no disponible)

- Multas

Registra una multa asociada a un préstamo, con monto y estado

Como funciona

El administrador inicia sesión con credenciales predefinidas

Las funciones del cliente (JavaScript) envían solicitudes POST al servidor (Express). El servidor maneja estas solicitudes, ejecuta las consultas SQL necesarias y devuelve la respuesta al cliente

Rutas del servidor

GET /users: Obtiene todos los usuarios

GET /books: Obtiene todos los libros

POST /registerUser: Registra un nuevo usuario

POST /registerLibrarian: Registra un nuevo bibliotecario

POST /addBook: Registra un nuevo libro

POST /createLoan: Registra un nuevo préstamo

POST /createStock: Registra un nuevo stock

POST /createFine: Registra una nueva multa

- Uso basico

Iniciamos el servidor con node server.js

Vamos a

http://localhost:5555

Iniciamos sesion con nuestras credenciales

Usuario: admin

Contrasena: admin10

Y ya podemos navegar por las funcionalidades

Registrar usuarios, bibliotecarios, libros

Crear préstamos, stocks y multas