

Documentación de la App Neo Alexandria

Índice

Contenido

Idea y Alcance de la App de Biblioteca Virtual	2
Idea Principal.....	2
Alcance	2
Funcionalidades Principales:	3
Problemas que Resuelve:	4
Modelado de datos:.....	4
Explicación:	5
1. Tabla Categoría.....	5
2. Tabla Libros	5
3. Tabla Usuarios.....	5
4. Tabla Autores	6
5. Tabla Carrito	6
6. Tabla DetalleCarrito.....	6
7. Tabla ValoraciondeUsuarios	7
8. Tabla Notas.....	7
Diagramas de caso de uso:	8
Diagrama de navegación de Figma.....	10
Vista Administrador:	10
Vista Usuario	14
Resumen de las pantallas de la app 23:.....	16
Información Técnica.....	17
Tecnologías utilizadas	17
Frontend (aplicación móvil y escritorio):	17
Seguridad.....	17
Lectura y almacenamiento del JWT	¡Error! Marcador no definido.
Estructura técnica	18
Arquitectura general	19
Paquetes NuGet utilizados	19
Hosting	20

Endpoints de la API.....	20
Guía de Uso - App Neo Alexandria	21
Ingreso	21
Pantalla Principal.....	21
Secciones de la App	21
Libros	21
Notas	21
Valoraciones.....	22
Usuarios	22
Carritos.....	22
Categorías	22
Autores	22

Idea y Alcance de la App de Biblioteca Virtual

Idea Principal

Neo Alexandria es una app de Biblioteca Virtual diseñada para facilitar la compra y consulta de libros, permitiendo a los usuarios explorar títulos, autores y categorías desde cualquier lugar. Su objetivo es simplificar el acceso a contenido literario y ofrecer una experiencia amigable, tanto para usuarios comunes como para administradores.

Alcance

Roles:

- Usuarios (Clientes):
Pueden registrarse, iniciar sesión, ver libros, autores, categorías, entre otros. Pueden ver detalles de libros, agregar productos al carrito, ver detalles de autores, ver las notas específicas de cada libro o las valoraciones, pueden agregar un usuario nuevo solo si no están registrados.
Además, pueden crear Libros, y los demás ítems a los cuales tiene acceso desde los botones.

- **Administradores:**

Acceden a funcionalidades adicionales además de las ya mencionadas para usuario como la gestión (Alta, baja, modificación) de libros, usuarios (se añaden solo desde el login), categorías, carritos, notas, valoraciones y autores.

El contenido visible en la app varía en función del rol del usuario. Solo los administradores pueden ver y usar los botones de modificar y eliminar contenido.

Funcionalidades Principales:

Gestión de Usuarios: Registro, autenticación y control de acceso mediante rol. El registro incluye selección de rol desde un combo.

Exploración de Libros: Visualización de libros por nombre, autor, categoría, descripción, precio, stock y fecha de publicación.

Carrito de Compras: Los clientes y administradores pueden añadir libros a un carrito de compras personal. Permitiendo añadir un libro al presionar el botón "Añadir al carrito". Se implementó un sistema automático compra con el botón arriba mencionado, además, el usuario puede visualizar y organizar el libro agregado en su carrito desde su apartado.

Autores: Visualización de autores, con biografía y años de nacimiento/fallecimiento. Se pueden gestionar desde el rol administrador.

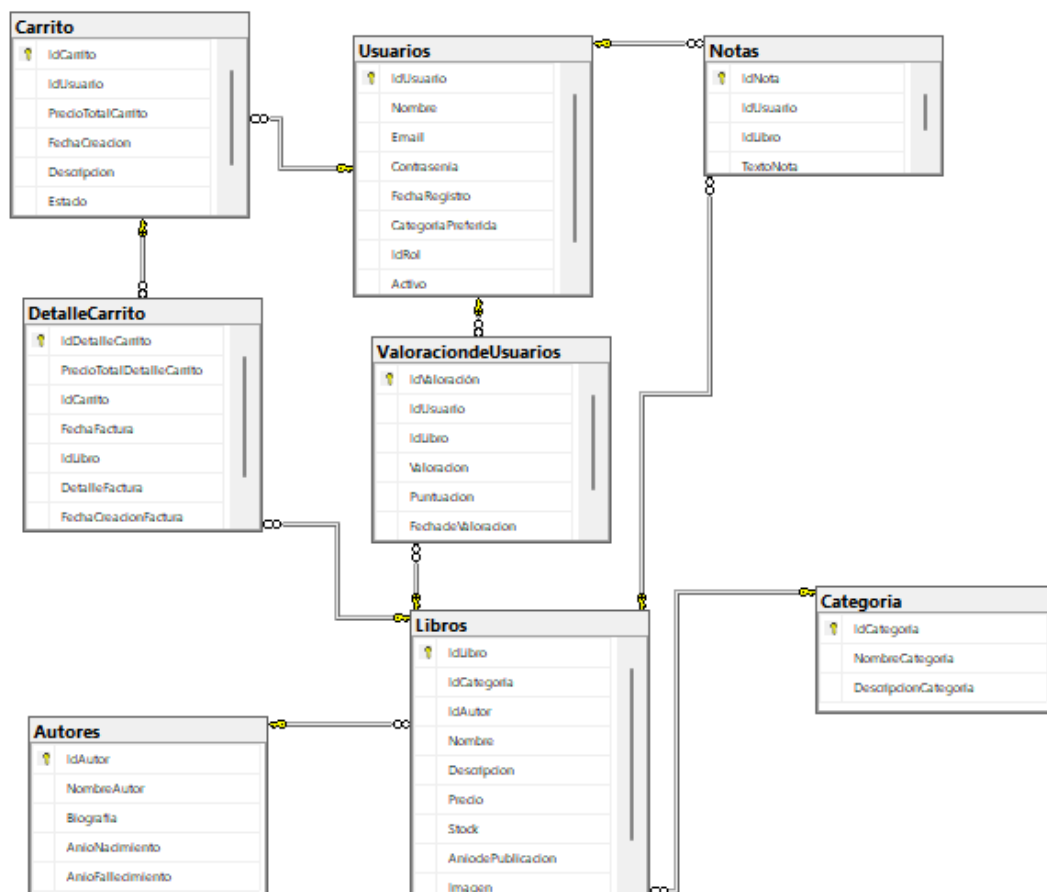
Notas y Valoraciones: Los usuarios y administradores pueden dejar notas y valoraciones en forma de comentarios y estrellas (1 a 5) para cada libro.

Gestión de Datos (Administradores): CRUD completo de usuarios, libros, autores, categorías, carritos y notas, desde interfaces específicas.

Problemas que Resuelve:

- Accesibilidad: Permite acceso a una biblioteca digital sin requerir presencia física.
- Gestión Centralizada: Reúne libros, autores, categorías, reseñas y carrito en una única app.
- Interacción Personalizada: El contenido mostrado y las acciones permitidas se ajustan al perfil del usuario.
- Apoyo a la Decisión: Las notas y valoraciones permiten a otros usuarios conocer opiniones antes de interactuar con los libros.

Modelado de datos:



Explicación:

1. Tabla Categoría

Atributos:

IdCategoría: Identificador único de la categoría (PRIMARY KEY).

NombreCategoría: Nombre de la categoría.

DescripcionCategoría: Descripción de la categoría.

Relaciones: Relacionada con la tabla Libros, indicando la categoría a la que pertenece un libro.

2. Tabla Libros

Atributos:

IdLibro: Identificador único del libro (PRIMARY KEY).

IdCategoría: Referencia a Categoría para indicar a qué categoría pertenece el libro.

IdAutor: Referencia a Autores para indicar el autor del libro.

Nombre: Nombre del libro.

Descripción: Descripción del libro.

Precio: Precio del libro.

Stock: Cantidad disponible en inventario.

AniodePublicacion: Fecha de publicación del libro.

Imagen: Ruta o URL de la imagen del libro.

Relaciones:

Relacionada con Categoría a través de IdCategoría.

Relacionada con Autores mediante IdAutor.

Relacionada con DetalleCarrito y ValoraciondeUsuarios, indicando en qué carrito está el libro y las valoraciones de los usuarios.

3. Tabla Usuarios

Atributos:

IdUsuario: Identificador único del usuario (PRIMARY KEY).

Nombre: Nombre del usuario.

Email: Correo electrónico del usuario.

Contraseña: Contraseña del usuario.

FechaRegistro: Fecha de registro del usuario en la plataforma.

CategoriaPreferida: Categoría favorita del usuario.

IdRol: Rol del usuario (puede referirse a permisos o nivel de acceso).

Activo: Indica si la cuenta del usuario está activa o no.

Relaciones: Relacionada con Carrito, Notas y ValoraciondeUsuarios, indicando el propietario del carrito, las notas y las valoraciones.

4. Tabla Autores

Atributos:

IdAutor: Identificador único del autor (PRIMARY KEY).

NombreAutor: Nombre del autor.

Biografia: Breve biografía del autor.

AnioNacimiento: Año de nacimiento del autor.

AnioFallecimiento: Año de fallecimiento del autor (puede ser NULL si aún vive).

Relaciones: Relacionada con Libros para indicar el autor de cada libro.

5. Tabla Carrito

Atributos:

IdCarrito: Identificador único del carrito (PRIMARY KEY).

IdUsuario: Referencia a Usuarios para indicar el propietario del carrito.

PrecioTotalCarrito: Precio total del carrito.

FechaCreacion: Fecha de creación del carrito.

Descripcion: Descripción del carrito.

Estado: Estado del carrito (probablemente indique si está activo o finalizado).

Relaciones: Relacionada con Usuarios y DetalleCarrito para especificar el usuario propietario del carrito y los detalles de compra.

6. Tabla DetalleCarrito

Atributos:

IdDetalleCarrito: Identificador único del detalle del carrito (PRIMARY KEY).

PrecioTotalDetalleCarrito: Precio total de los productos en este detalle.

IdCarrito: Referencia a Carrito para indicar a qué carrito pertenece.

FechaFactura: Fecha de emisión de la factura.

IdLibro: Referencia a Libros para indicar qué libro se agregó al carrito.

DetalleFactura: Detalles adicionales de la factura.

FechaCreacionFactura: Fecha de creación de la factura.

Relaciones:

Relacionada con Carrito para indicar el carrito al que pertenece este detalle.

Relacionada con Libros para indicar qué libro contiene este detalle.

7. Tabla ValoraciondeUsuarios

Atributos:

IdValoracion: Identificador único de la valoración (PRIMARY KEY).

IdUsuario: Referencia a Usuarios para indicar quién realizó la valoración.

IdLibro: Referencia a Libros para indicar el libro valorado.

Valoracion: Comentario o valoración escrita del usuario.

Puntuacion: Puntuación asignada (ej. 1 a 5).

FechadeValoracion: Fecha de la valoración.

Relaciones:

Relacionada con Usuarios y Libros para indicar quién realizó la valoración y qué libro fue valorado.

8. Tabla Notas

Atributos:

IdNota: Identificador único de la nota (PRIMARY KEY).

IdUsuario: Referencia a Usuarios para indicar el usuario que crea la nota.

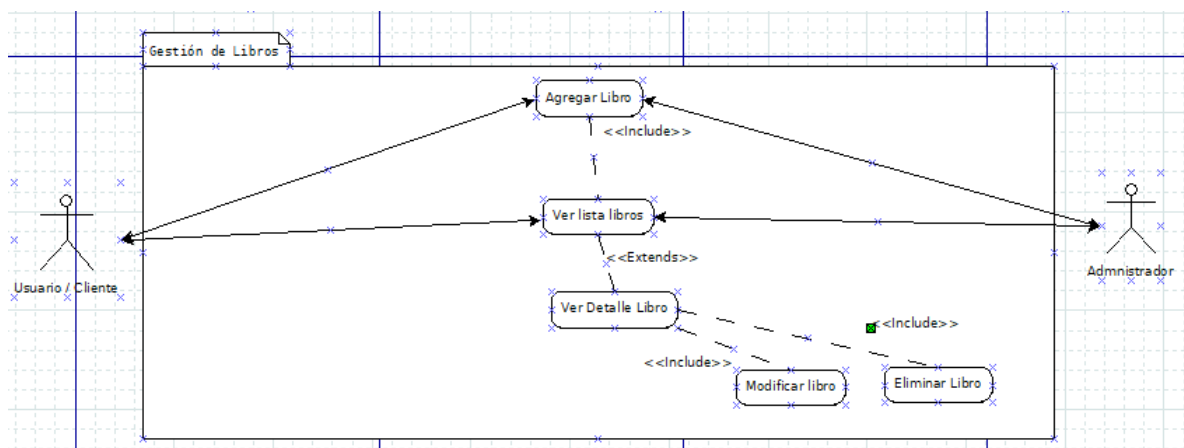
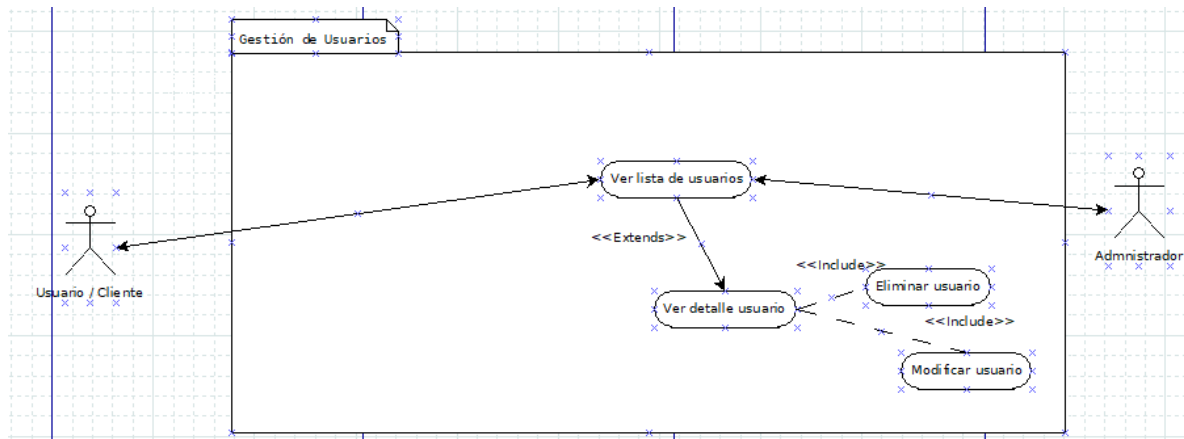
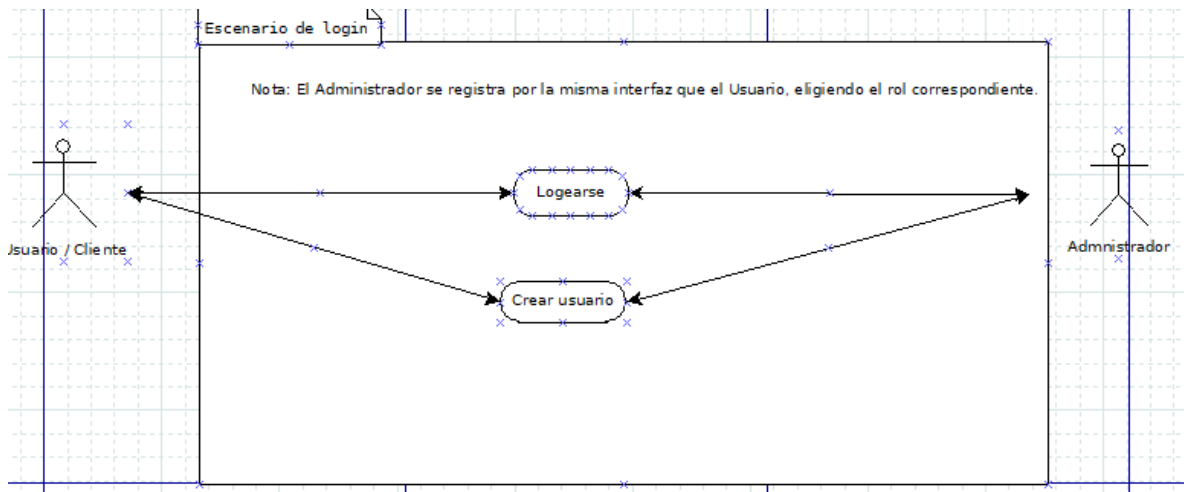
IdLibro: Referencia a Libros para indicar el libro relacionado con la nota.

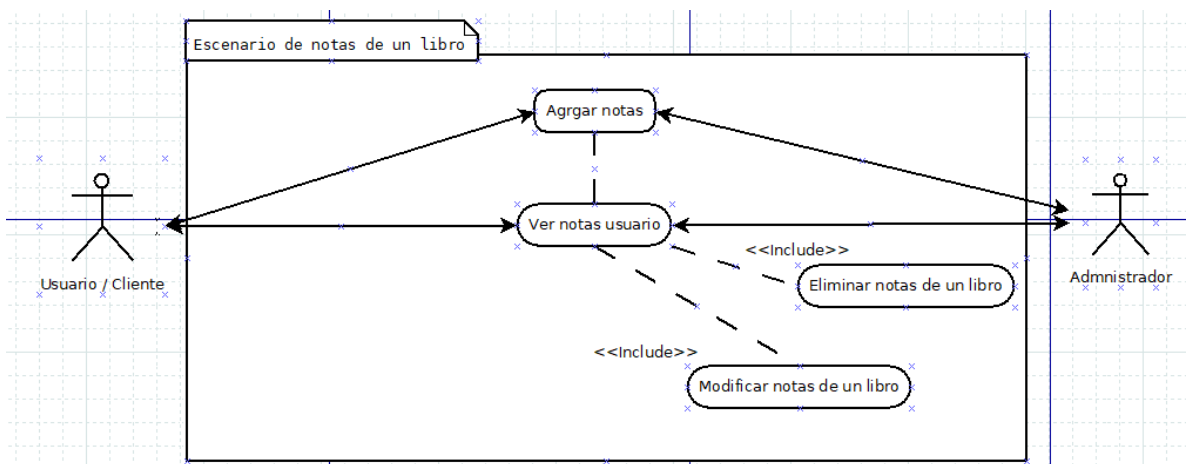
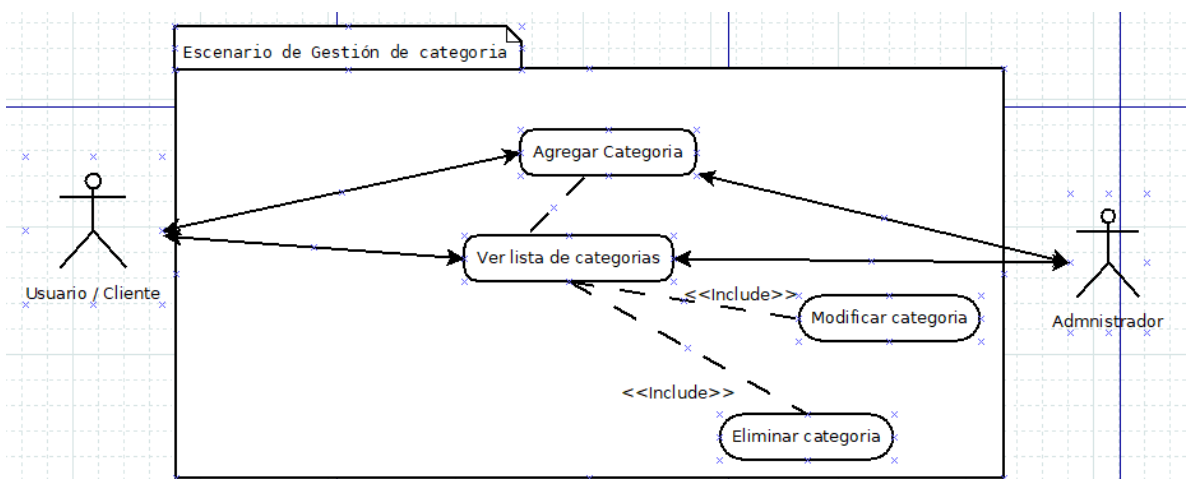
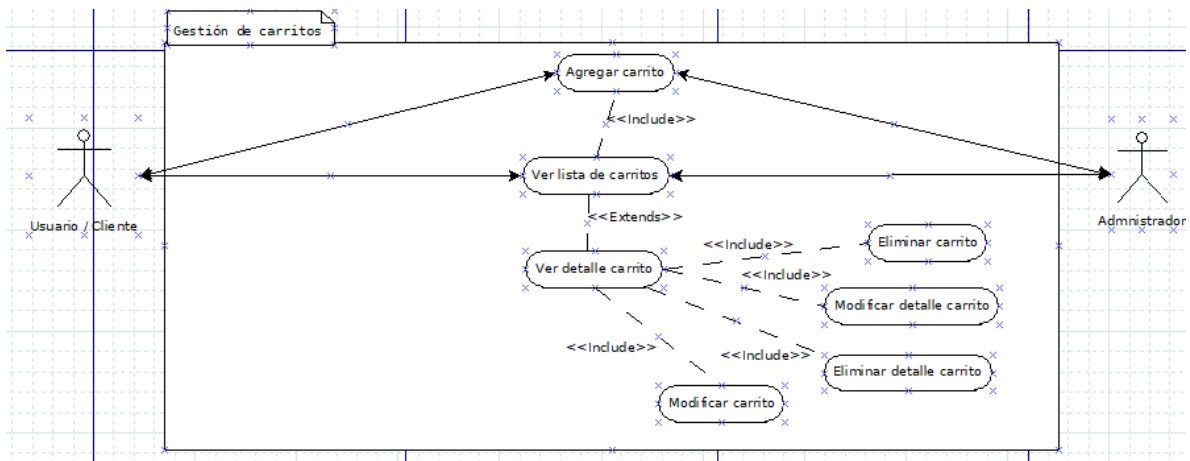
TextoNota: Contenido de la nota.

Relaciones:

Relacionada con Usuarios y Libros para identificar el creador de la nota y el libro al que se refiere.

Diagramas de caso de uso:





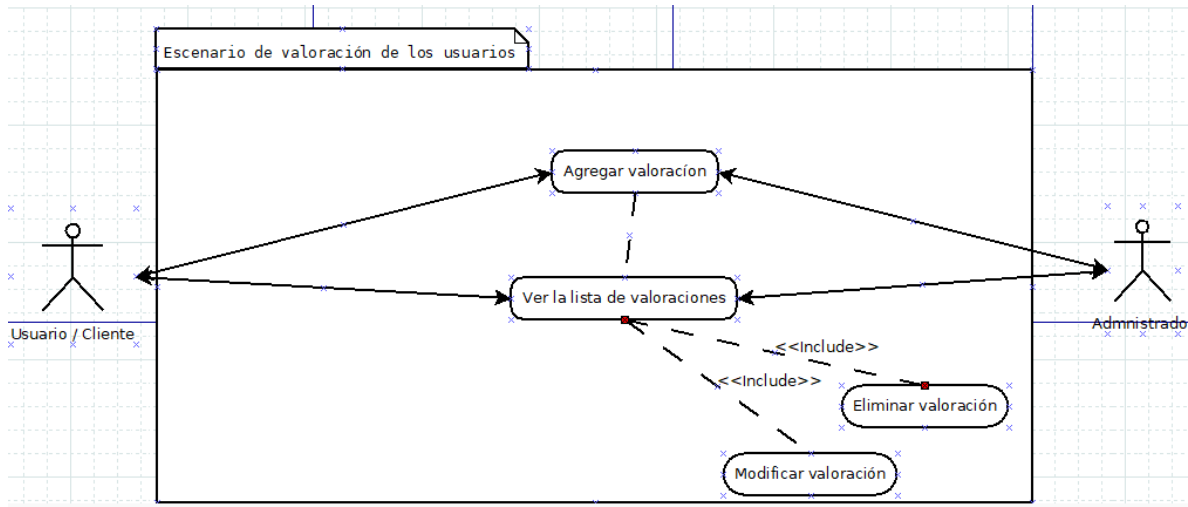
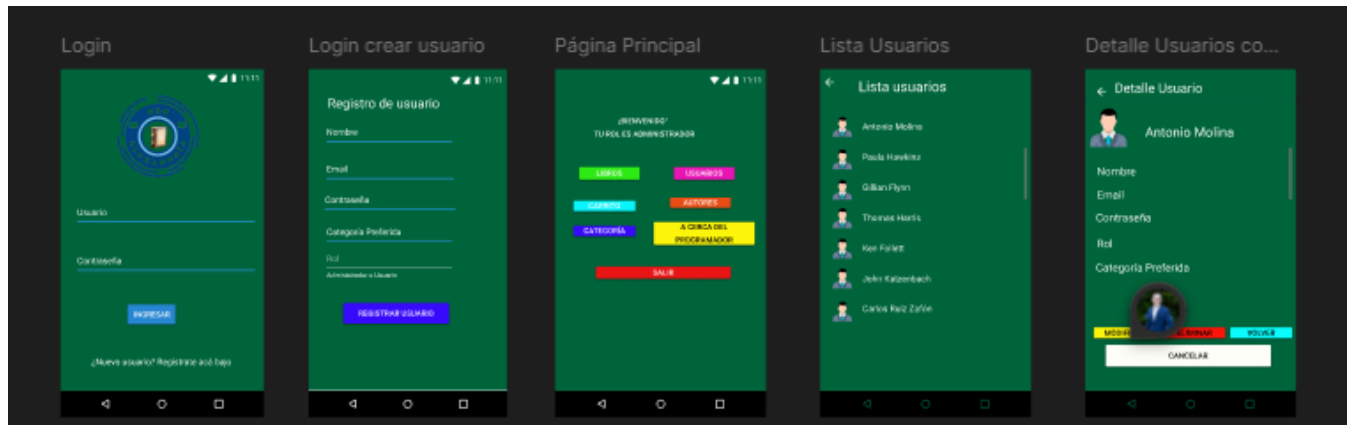


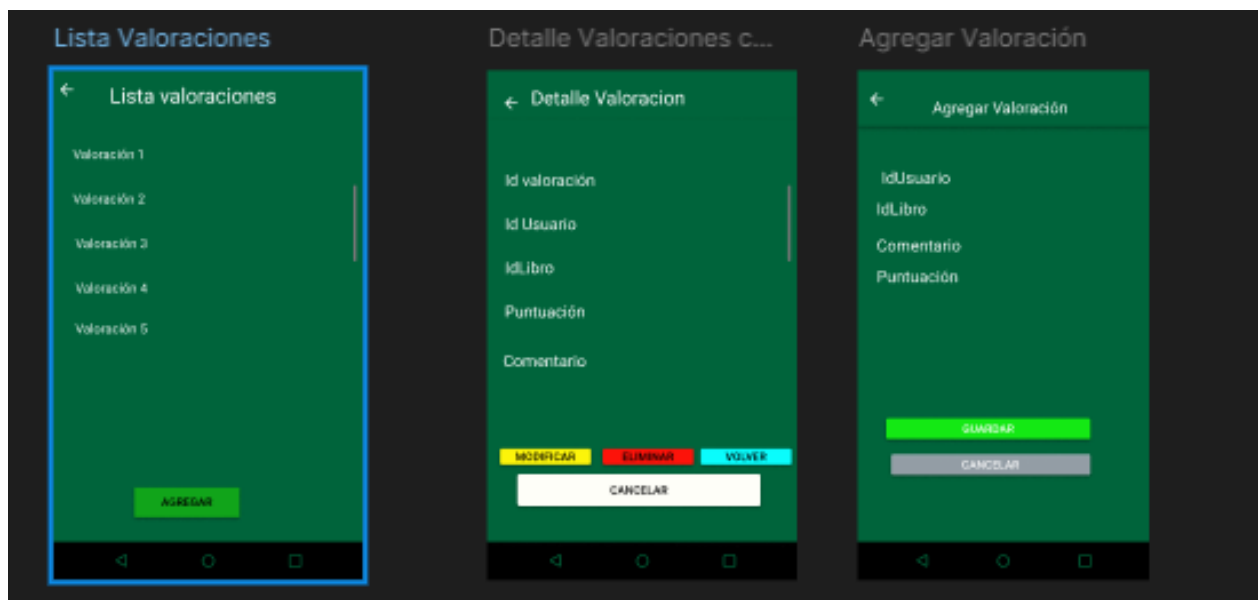
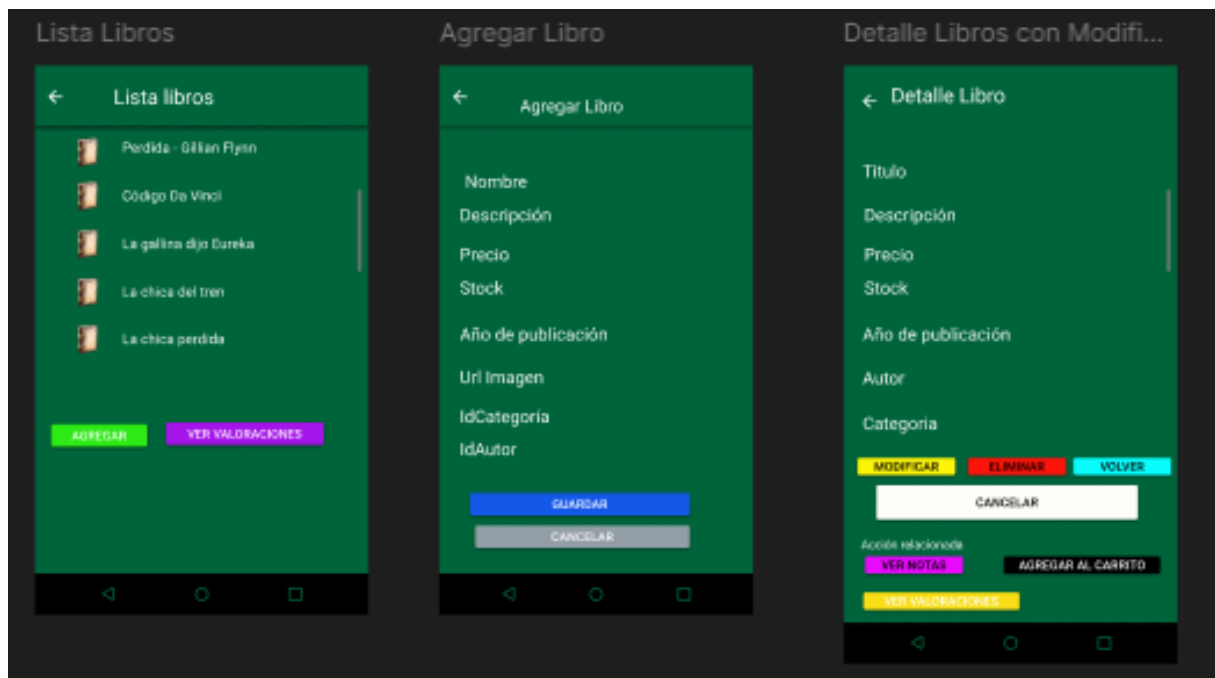
Diagrama de navegación de Figma

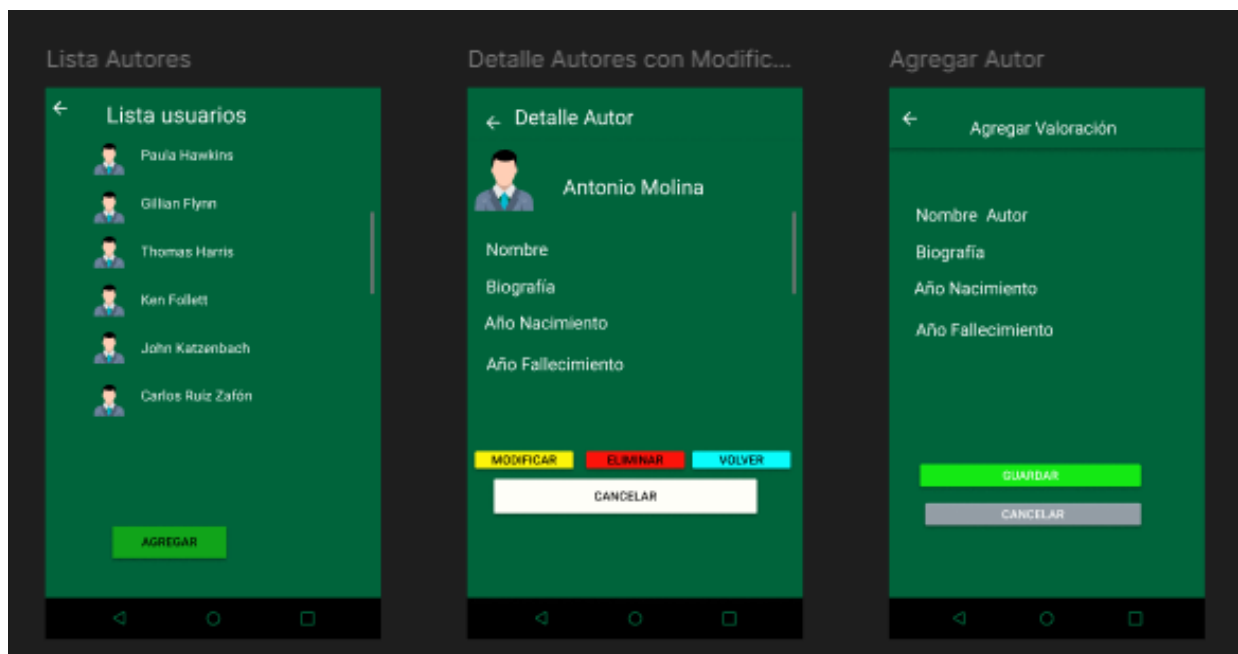
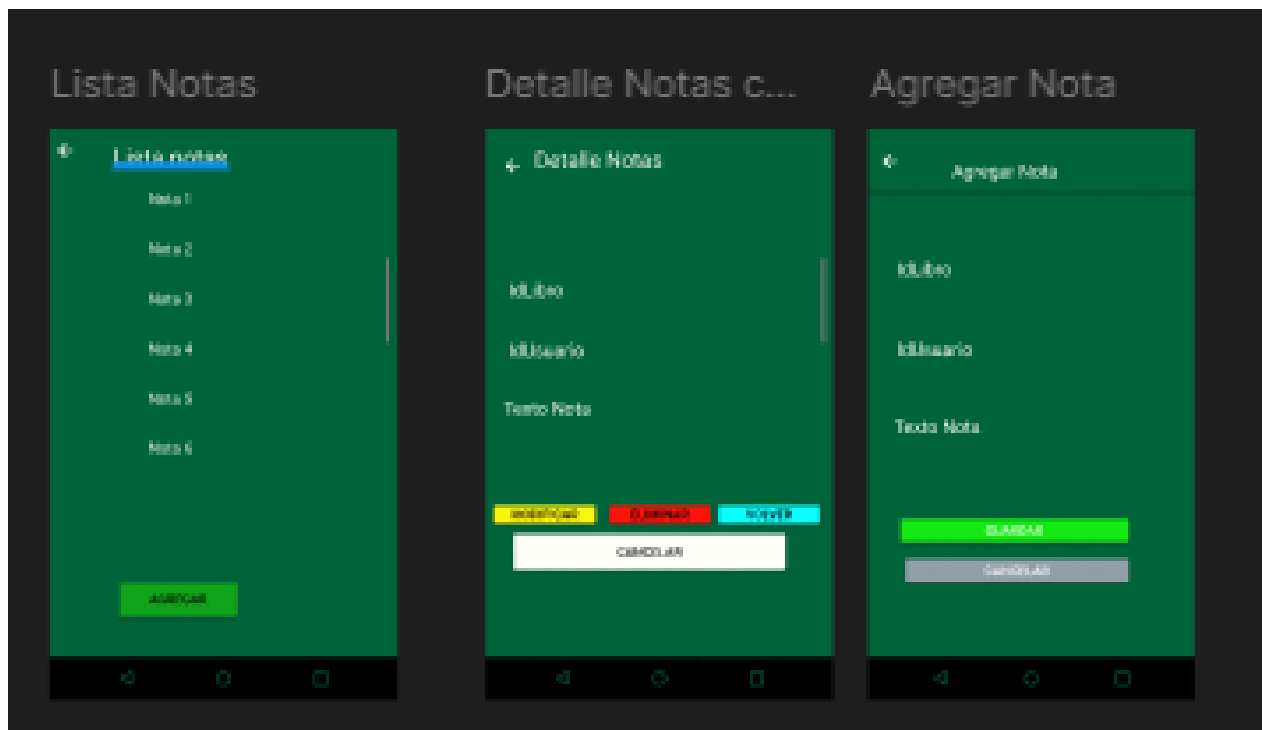
(24 pantallas, pero login y login crear usuarios se manejan en una sola en la app)

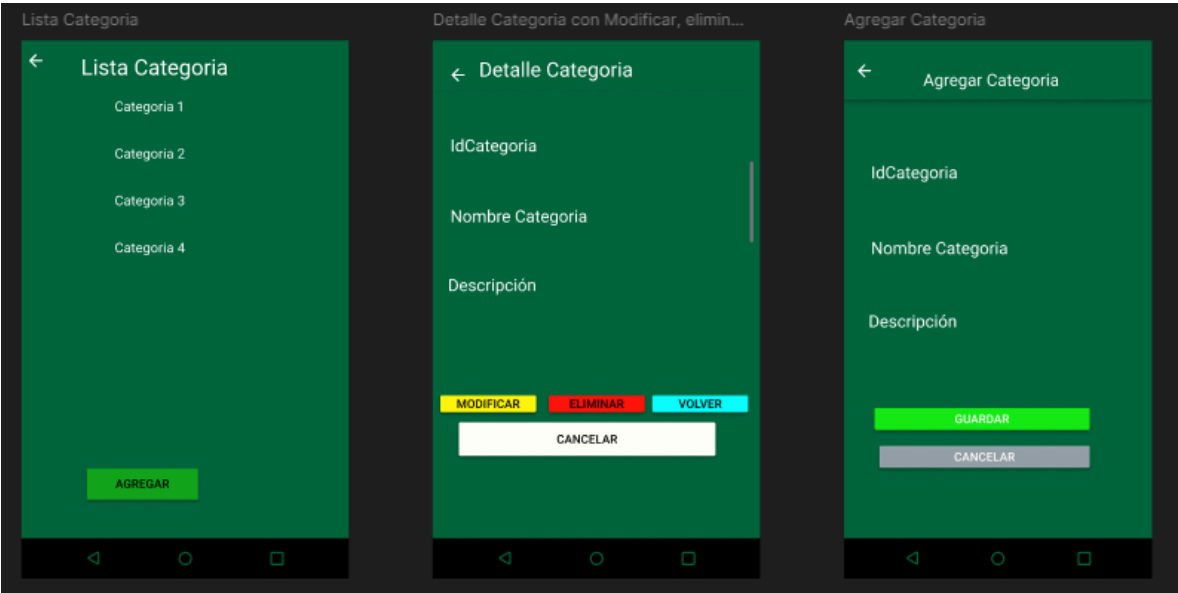
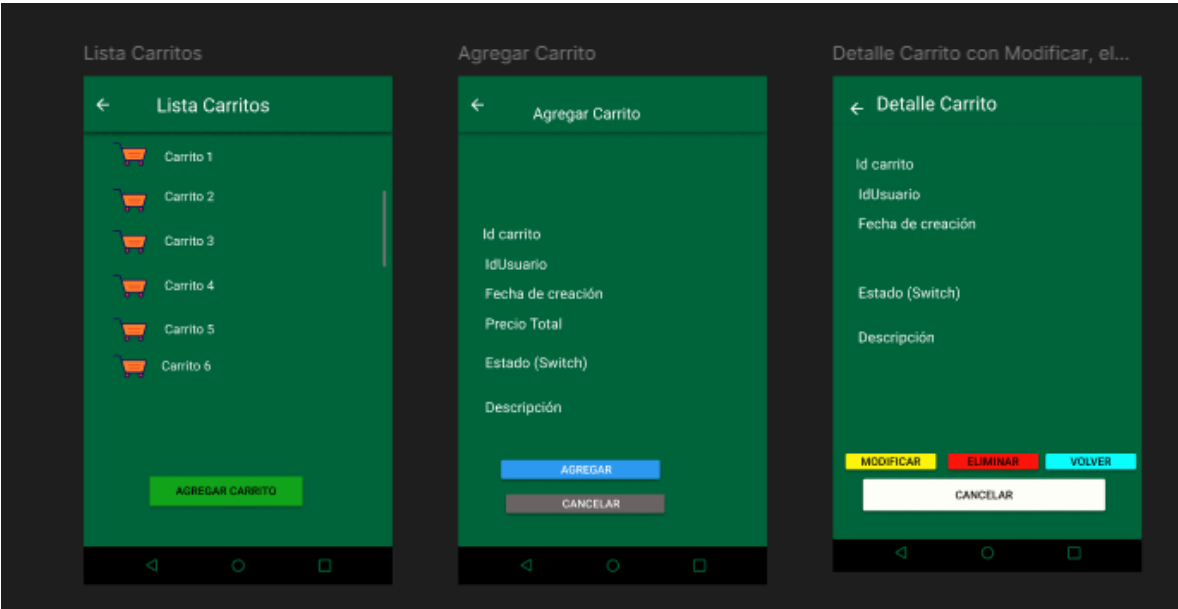
Vista Administrador:



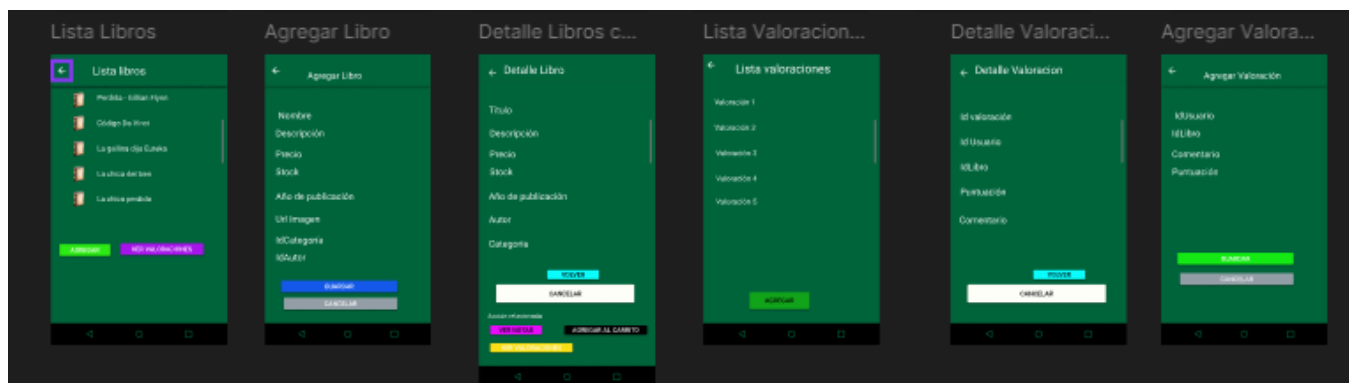
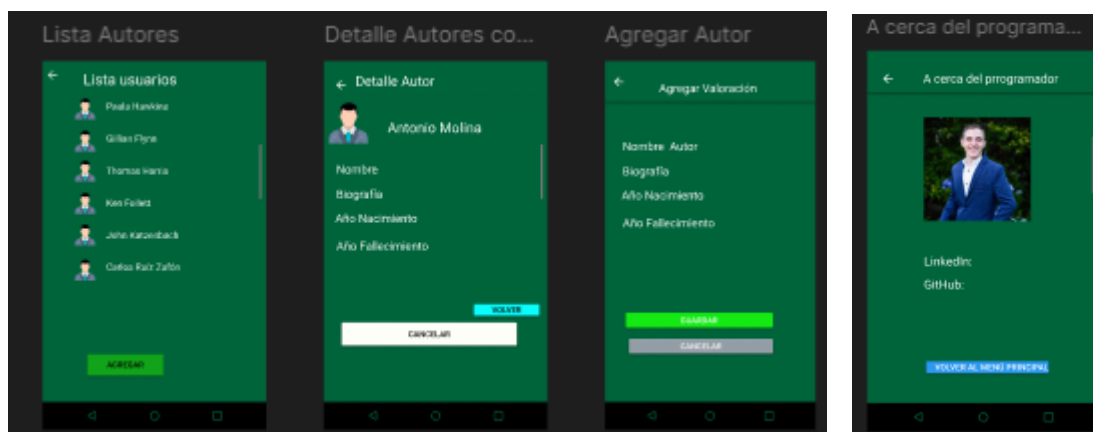
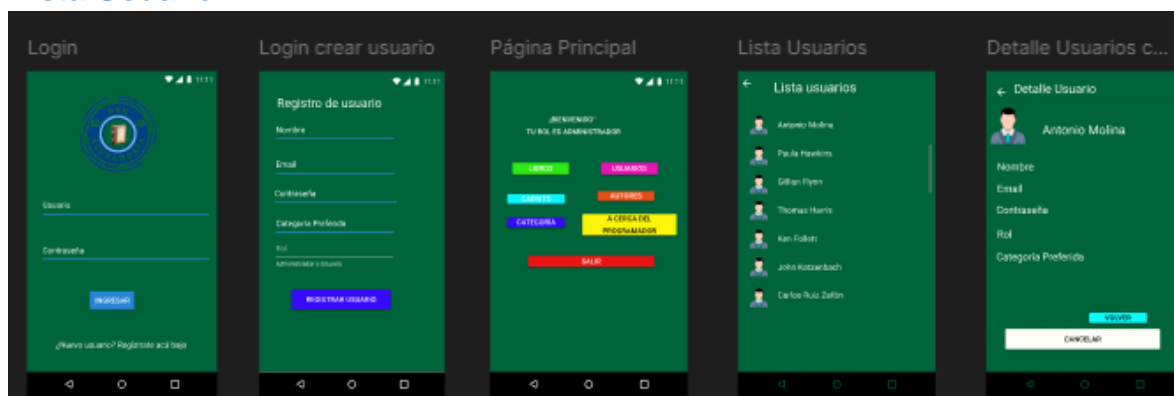
-El comentario sobre el botón modificar dice "Al darle click a modificar el botón volver se cambia a guardar y hay que presionar ahí para que se guarden los cambios"

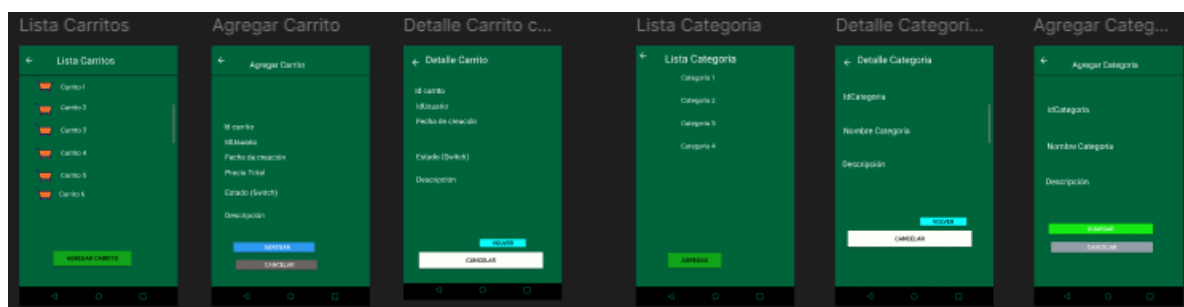
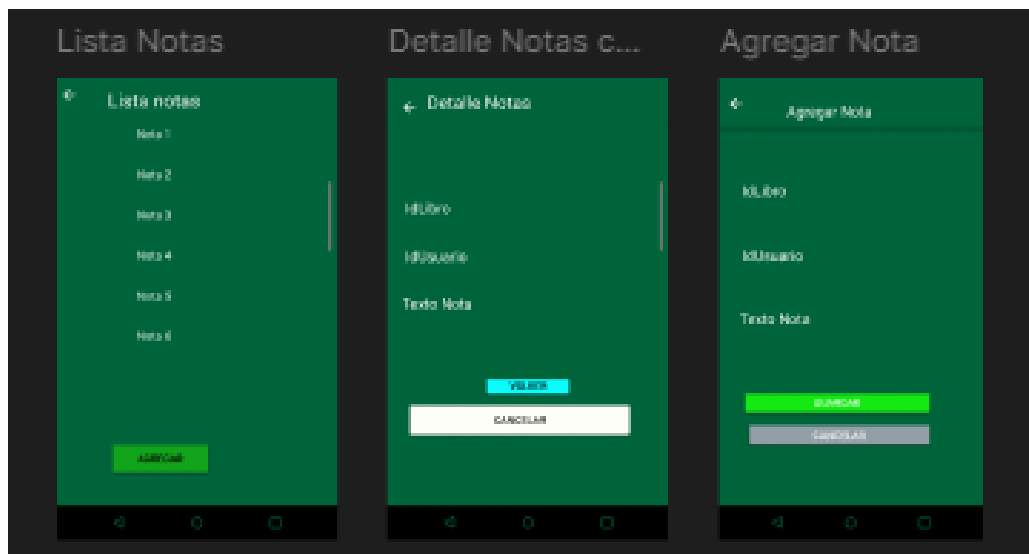






Vista Usuario





Resumen de las pantallas de la app 23:

Módulo / Sección	Nombre del archivo .xaml	Función principal
Autor	AutorPage.xaml	Ver lista de autores
	AutorAgregarPage.xaml	Agregar nuevo autor
	AutorDetallePage.xaml	Ver / Modificar / Eliminar autor
Carrito	CarritoPage.xaml	Ver lista de carritos
	CarritoAgregarPage.xaml	Agregar nuevo carrito (manual)
	CarritoDetallePage.xaml	Ver / Modificar / Eliminar carrito
Categoría	CategoriaPage.xaml	Ver lista de categorías
	CategoriaAgregarPage.xaml	Agregar nueva categoría
	CategoriaDetallePage.xaml	Ver / Modificar / Eliminar categoría
Libro	LibroPage.xaml	Ver lista de libros
	LibroAgregarPage.xaml	Agregar nuevo libro
	LibroDetallePage.xaml	Ver / Modificar / Eliminar libro
Notas	NotasPage.xaml	Ver lista de notas
	NotasAgregarPage.xaml	Agregar nota
	NotasDetallePage.xaml	Ver / Modificar / Eliminar nota
Usuario	UsuarioPage.xaml	Ver lista de usuarios
	UsuarioDetallePage.xaml	Ver / Modificar / Eliminar usuario
Valoraciones	ValoraciondeUsuariosPage.xaml	Ver lista de valoraciones
	ValoraciondeUsuariosAgregarPage.xaml	Agregar nueva valoración
	ValoraciondeUsuariosDetallePage.xaml	Ver / Modificar / Eliminar valoración
General / Navegación	LoginyRegistroPage.xaml	Login y registro de usuario
	MainPage.xaml	Página principal post-login
	AcercaPage.xaml	Información institucional de la app

Información Técnica

Tecnologías utilizadas

Frontend (aplicación móvil y escritorio):

- .NET MAUI
- Lenguaje: C#
- XAML para el diseño de interfaces
- CommunityToolkit.Mvvm para MVVM y binding

Backend (API REST):

- ASP.NET Core Web API
- Entity Framework Core
- Autenticación con JWT (Json Web Token)
- Base de datos: SQL Server

Comunicación entre App y API:

- HTTP mediante HttpClient
- La clase ApiService centraliza las llamadas REST (GET, POST, PUT, DELETE).

Seguridad

Inicio de sesión y registro:

- Todo se maneja en una sola pantalla (LoginyRegistroPage.xaml).
- Al iniciar sesión, la API devuelve un token JWT.
- El token contiene claims como IdUsuario, Nombre, Email y Rol.

Control de acceso por rol:

- Se reconocen dos roles: 'ADM' (Administrador) y 'USR' (Usuario).
- El token JWT se lee al iniciar sesión y sus datos se almacenan en memoria (Transport) y en Preferences.
- Según el rol, se habilitan o deshabilitan botones de edición y eliminación en toda la app.

Lectura y almacenamiento del JWT

Al iniciar sesión correctamente en la app Neo Alexandria, la API devuelve un JWT que contiene los datos del usuario autenticado. Utilizamos el paquete NuGet

`Microsoft.IdentityModel.JsonWebTokens` para leer el contenido del token de forma segura y moderna. Desde el token, extraemos manualmente los claims relevantes (`IdUsuario`, `Nombre`, `Email`, `IdRol`), que utilizamos para identificar y diferenciar a los usuarios dentro de la app. Estos valores se almacenan en la clase estática `Transport` para mantenerlos disponibles en memoria durante la sesión en ejecución y en `Preferences` para persistirlos localmente, permitiendo mantener la sesión activa entre reinicios de la app. Esto nos permite aplicar control de roles en tiempo de ejecución (por ejemplo, ocultar botones de modificar y eliminar si el usuario no es administrador) y proteger funcionalidades sensibles según el nivel de permisos del usuario, asegurando que solo usuarios autorizados puedan ejecutar ciertas acciones dentro de la aplicación.

Control de roles en la aplicación

Una vez almacenados los datos del token, el campo `IdRol` extraído se utiliza para definir el comportamiento visual y funcional de la app. Si el usuario tiene rol de 'Administrador', se habilitan opciones de gestión como agregar, modificar y eliminar registros en entidades (Libros, Usuarios, Categorías, etc.). Si el rol es 'Usuario', estas acciones administrativas se ocultan dinámicamente en todas las vistas relevantes, permitiendo únicamente operaciones de lectura y funciones limitadas como añadir libros al carrito o realizar valoraciones. Este control de roles se implementa tanto en los `ViewModels` mediante la propiedad `EsAdmin`, como en las vistas XAML con bindings de visibilidad, garantizando que la seguridad y los permisos se apliquen correctamente en cada pantalla de la app.

Estructura técnica

- Cada entidad implementada tiene:
 - Una página para ver lista de elementos
 - Una página para agregar un nuevo registro
 - Una página de detalle (que permite ver, modificar y eliminar)
- Las vistas están organizadas por carpetas:

`/Views/`

`/Autor/`

`/Libro/`

`/Carrito/`

`/Categoria/`

`/Notas/`

`/Usuario/`

`/ValoraciondeUsuarios/`

- La app cuenta con 23 pantallas `.xaml` en total.

Arquitectura general

- Modelo de diseño: MVVM (Model-View-ViewModel)
- Navegación: Navigation.PushAsync y uso de Shell para páginas principales
- Persistencia de sesión: Xamarin.Essentials Preferences

Paquetes NuGet utilizados

En la API .NET:

- **Microsoft.Extensions.Caching.Memory**: permite **caché en memoria** para mejorar el rendimiento de consultas recurrentes.
- **Microsoft.VisualStudio.Web.CodeGeneration.Design**: facilita el **scaffolding de controladores y vistas** rápidamente.
- **Swashbuckle.AspNetCore**: se utiliza para generar **Swagger**, documentando de forma automática los endpoints de la API.
- **System.Text.Json / System.Formats.Asn1**: utilizados para **serialización y deserialización JSON**, fundamentales en las peticiones/respuestas de la API.
- **Microsoft.AspNetCore.Authentication.JwtBearer / System.IdentityModel.Tokens.Jwt**: permiten **autenticación y validación de JWT**, lo que usamos para controlar el login y los permisos por rol en la API.
- **Microsoft.Data.SqlClient y dependencias**: necesarios para la **conexión y manejo de SQL Server** en la API.
- **Microsoft.EntityFrameworkCore.SqlServer y Tools**: permiten utilizar **Entity Framework Core con SQL Server** para el mapeo de tablas, migraciones y control de datos en la base de datos.

En la App MAUI:

- **CommunityToolkit.Mvvm**: permite utilizar **MVVM simplificado** con [ObservableProperty] y [RelayCommand], optimizando el binding entre ViewModel y vistas.
- **Microsoft.Maui.Controls y UI. Xaml**: son el núcleo de **.NET MAUI** para construir interfaces gráficas multiplataforma con XAML.
- **Microsoft.Extensions. Logging.Debug**: utilizado para **log de depuración** durante el desarrollo.
- **System.IdentityModel.Tokens.Jwt** y **System.Text.Json**: igual que en la API, se usan para **leer y manejar tokens JWT** en el cliente, autenticando al usuario y controlando permisos.
- **Microsoft.NET.ILLink.Tasks**: gestiona **optimización y reducción del tamaño de publicación** en MAUI.
- **System.Text.Json** nuevamente se usa para **serialización/deserialización** al consumir la API.

En resumen, estos paquetes se usaron para implementar la arquitectura cliente-servidor con autenticación JWT, consumir la API desde MAUI, gestionar datos con Entity Framework y estructurar el proyecto con MVVM en MAUI, garantizando una app limpia, mantenible y con una interfaz funcional.

Preguntar a ChatGPT

Hosting

- La base de datos SQL Server y la API REST (ASP.NET Core) están hosteadas en la plataforma Somee (<https://somee.com>).

Endpoints de la API

- La API REST está organizada en controladores por entidad, cada uno expone endpoints típicos RESTful:

- GET para obtener listas o detalles
- POST para crear nuevos registros
- PUT para actualizar registros
- DELETE para eliminarlos

Algunos endpoints personalizados fueron utilizados para:

- Login y generación de JWT
- Registro de usuarios con validación de rol
- Acceso filtrado por IdLibro (en notas)

- La URL base de consumo desde la app se define en la clase ApiService y en /Utils/Constants y apunta al dominio de Somee.

Guía de Uso - App Neo Alexandria

Ingreso

- Al iniciar la app, se muestra una única pantalla que permite tanto iniciar sesión como registrarse.
- Si ya tenés una cuenta, ingresá tu email y contraseña y presioná “Iniciar sesión”.
- Si no tenés cuenta hace click al texto que dice “No tenes cuenta? Registrate”, completá tus datos, seleccioná un rol (“Administrador” o “Usuario”) y presioná “Registrarse”.

Pantalla Principal

- Una vez iniciada la sesión correctamente, accedés a la pantalla principal.
- Desde ahí podés ingresar a las secciones: Libros, Autores, Categorías, Carritos, Usuarios, A cerca de (Página con datos del programador).

Secciones de la App

Libros

- Ver la lista de libros disponibles.
- Ver el detalle de cada libro (con imagen, autor, categoría, descripción, precio y stock).
- Agregar un nuevo libro.
- Agregar un libro al carrito
- Ver las valoraciones en de cada libro
- Modificar o eliminar un libro (solo si sos administrador).
 - Desde el detalle del libro podés:
- Acceder a las notas de ese libro.

Notas

- Las notas se acceden desde el detalle de un libro.
- Ver la lista de notas vinculadas al libro seleccionado.
- Agregar una nota nueva de ese libro o de otro particular.
- Modificar o eliminar una nota desde el detalle (solo si sos administrador).

Valoraciones

- Se accede desde el detalle del libro.
- Ver la lista de valoraciones vinculada al libro seleccionado.
- Agregar una nueva valoración (comentario + puntuación de número que después se ven con estrellas).
- Modificar o eliminar una valoración desde el detalle de valoración (solo si sos administrador).

Usuarios

- Ver lista de usuarios registrados.
- Ver el detalle de cada usuario.
- Modificar o eliminar usuarios desde el detalle usuario (solo si sos administrador).

Carritos

- Ver la lista de carritos.
- Agregar un nuevo carrito manualmente o automáticamente desde el detalle de un libro al dar click al botón "Añadir al carrito".
- Ver detalle del carrito.
- Modificar o eliminar un carrito desde el detalle carrito (solo si sos administrador).

Categorías

- Ver la lista de categorías existentes.
- Agregar una nueva categoría.
- Modificar o eliminar una categoría desde el detalle categoría (solo si sos administrador).

Autores

- Ver la lista de autores disponibles.
- Ver detalle de autor.
- Agregar un nuevo autor.
- Modificar o eliminar un autor desde el detalle Autor (solo si sos administrador).