UNIVERSIDAD MARIANO GALVEZ

Facultad De Ingeniería En Sistemas De Información

Programación 1

Catedrático: Miguel Catalán

Proyecto Semestre/Automatización Biblioteca-Manual técnico

Edgar Romario Chajón Chávez (7590 18 3531)

Sección A

Guatemala, San Juan Sacatepéquez mayo de 2024

Introducción

El presente proyecto tiene como objetivo el desarrollo de un Sistema de Gestión de Biblioteca, aprovechando los conceptos y habilidades adquiridas en el curso de programación 1. Este sistema permitirá a los usuarios gestionar de manera eficiente los libros, clientes y transacciones de préstamo, facilitando así la administración de la biblioteca.

El sistema desarrollado abarcará las siguientes funcionalidades principales

- Registro y gestión de libros por parte de administradores.
- Registro y gestión de usuarios.
- Préstamo y devolución de libros, con gestión automática de disponibilidad y multas por retraso.
- Búsqueda avanzada de libros.
- Operaciones CRUD para la gestión de información de libros y usuarios.

Mantenimiento de un historial detallado de todas las transacciones.
Interfaz de usuario diseñada para ser intuitiva y fácil de usar.
Manejo adecuado de situaciones excepcionales para asegurar la robustez del sistema.

Este proyecto no solo servirá como una herramienta útil para la gestión de bibliotecas, sino que también demostrará la capacidad de aplicar conocimientos de programación 1 en un contexto práctico, abarcando desde la creación de interfaces de usuario, la programación a objetos, hasta la implementación de lógicas de negocio complejas y el manejo seguro y eficiente de datos.

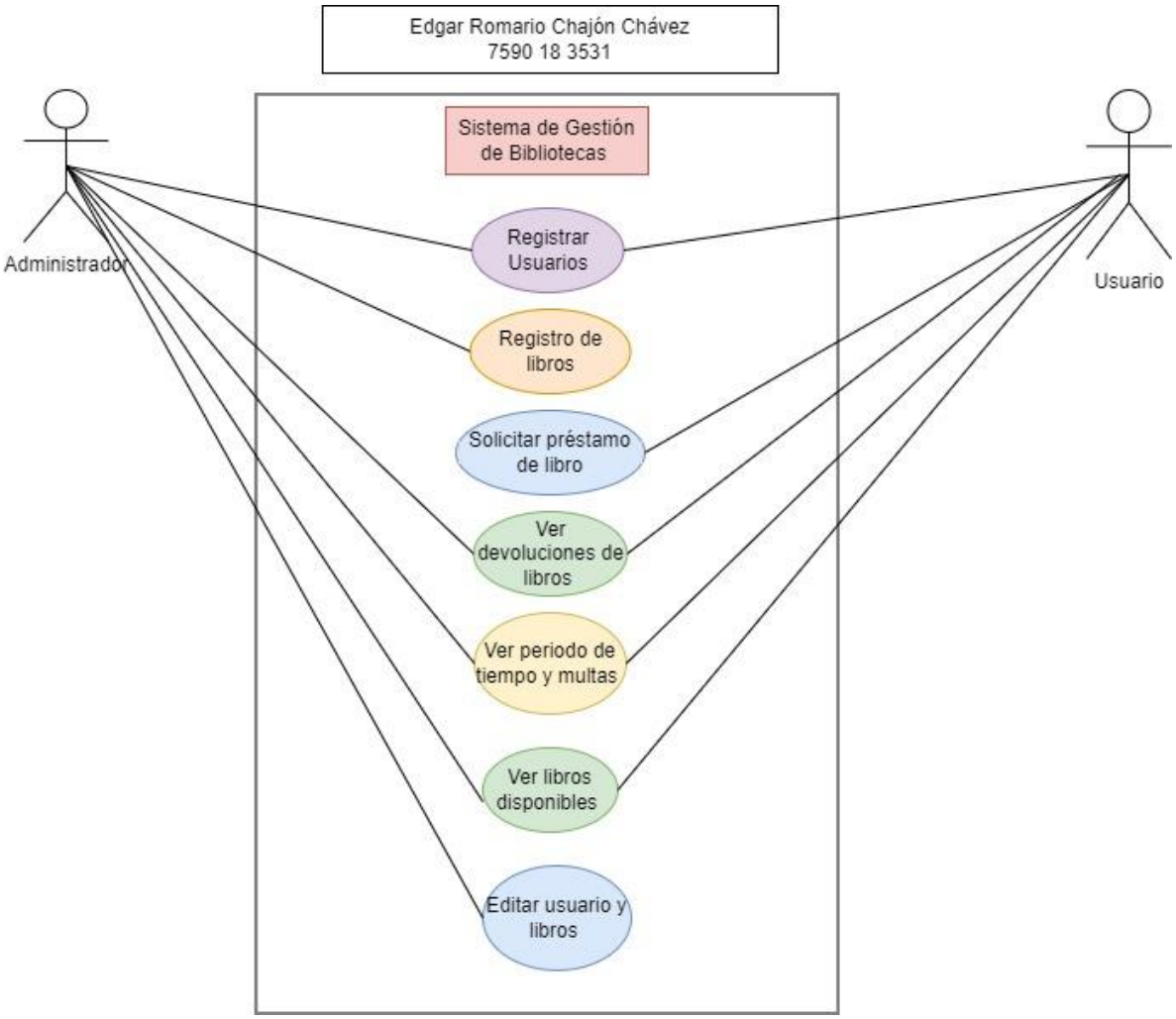
índice

Contenido

Manual técnico Biblioteca

El proyecto consiste en el desarrollo de un Sistema de Gestión de Biblioteca que aplica los conceptos fundamentales de la programación orientada a objetos. Este sistema tiene como propósito facilitar la administración de libros, clientes y transacciones de préstamos en una biblioteca. Está diseñado para resolver problemas comunes en la gestión de bibliotecas, como el seguimiento de préstamos y devoluciones, la administración de inventarios de libros y la gestión de información de clientes.

Diagrama clases de usos



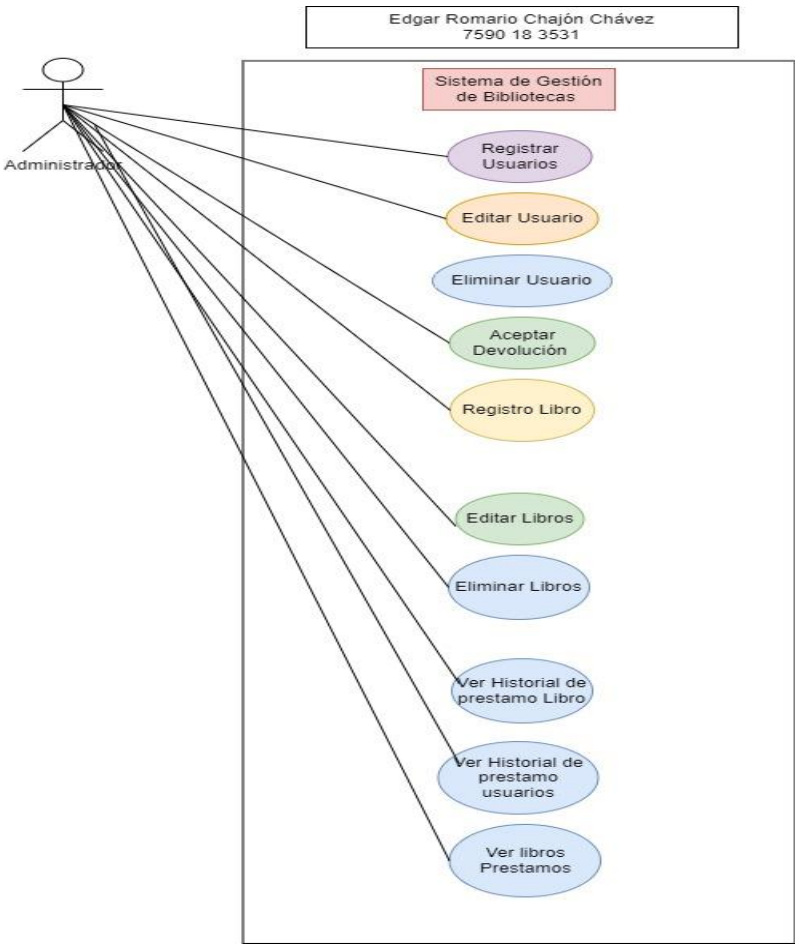
Como podemos analizar el administrador puede hacer lo siguiente:

- Registrar Usuario
- Registrar Libros
- Ver devoluciones de libros
- Ver Periodos y multas
- Ver libros disponibles
- Editar usuario y libros

Ahora bien el usuario puede

- Crear su cuenta o registrarse
- Solicitar Prestamos
- Ver devoluciones de libros
- Ver Periodos y multas
- Ver libros disponibles .

Aquí va un poco más detallado de lo que puede hacer el administrador .



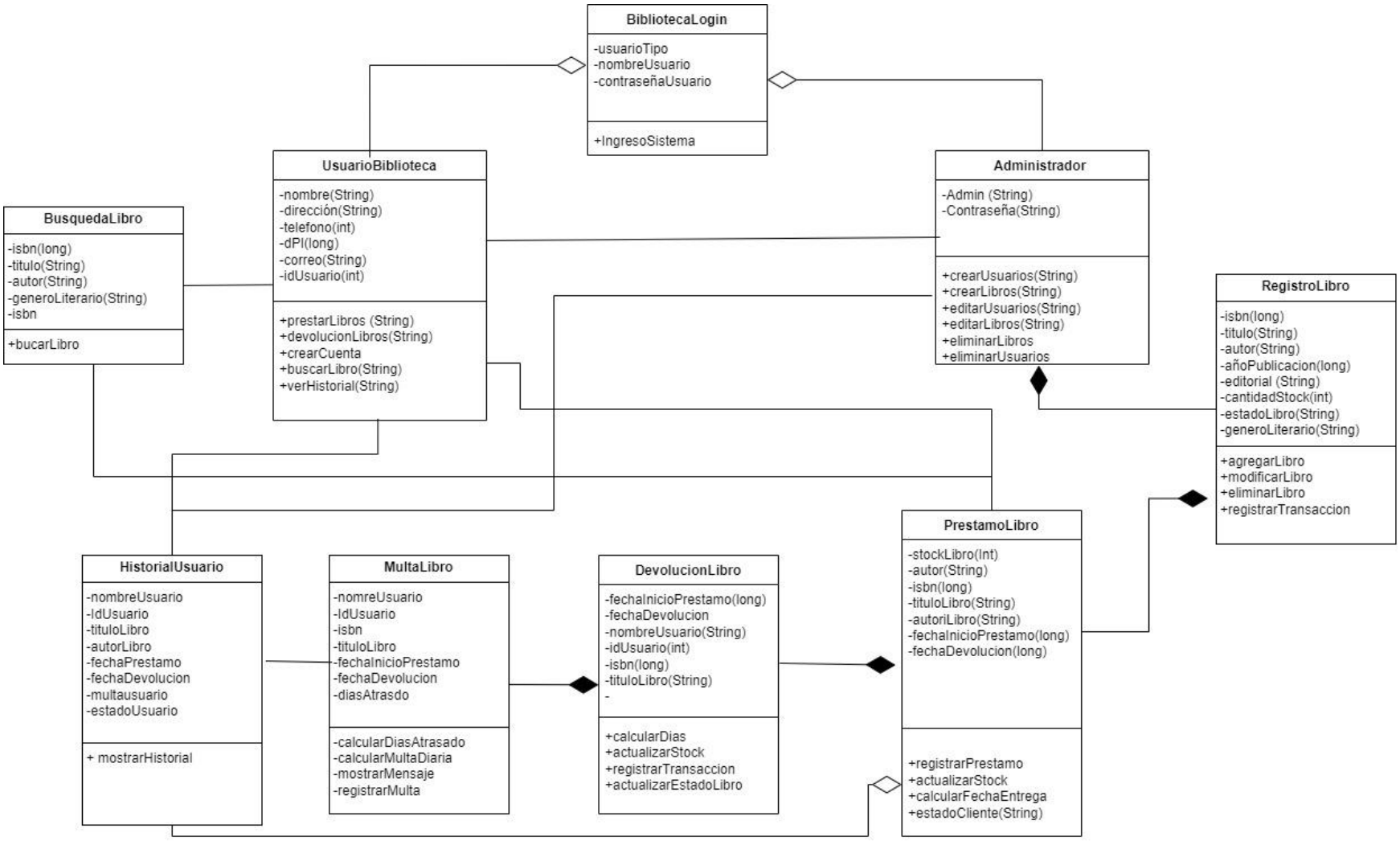
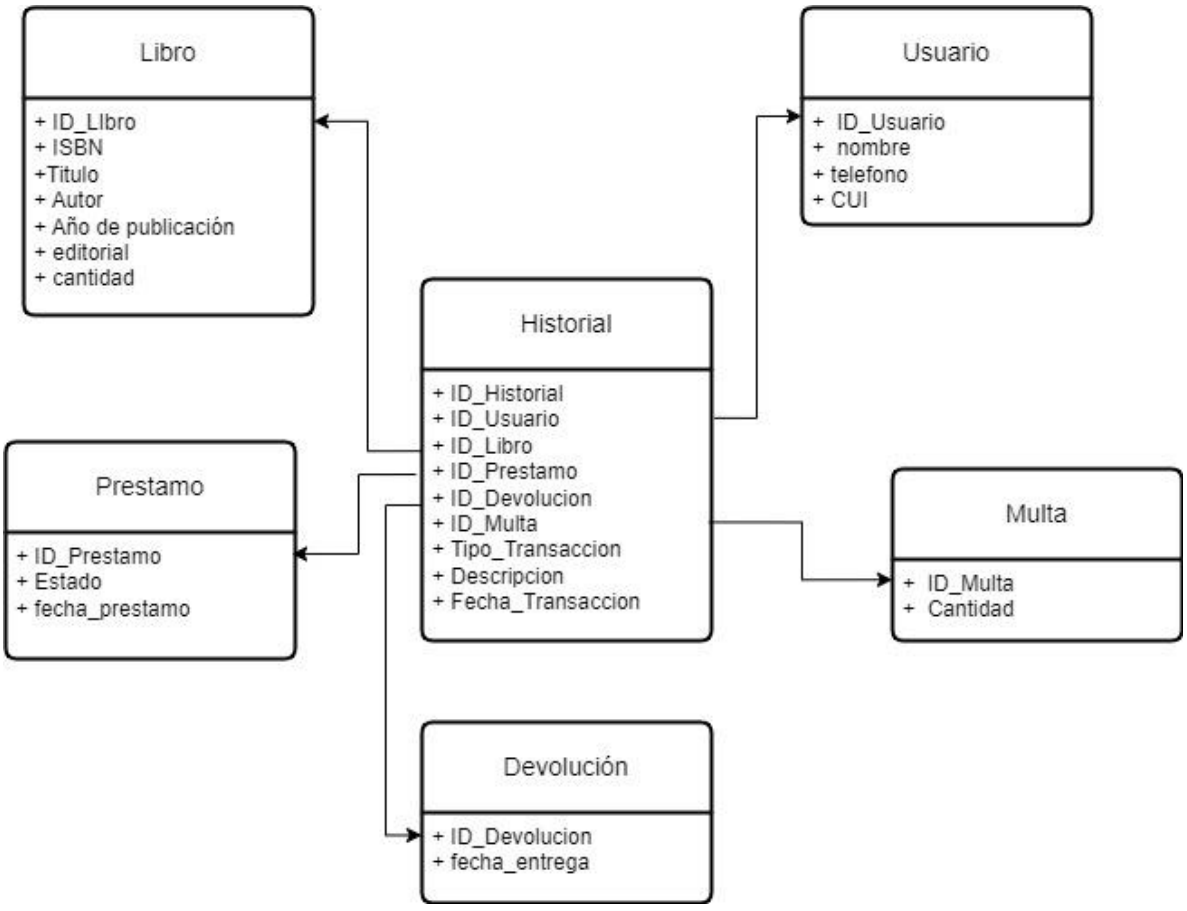


Diagrama de clase

Pare registrar libro depende del admin si no hay un admin no hay registro de libro y la consecuencia seria no habría libros por prestar que englobaría la existencia multa libro o historial ya que no hay quien los registre,

Ahora con el usuario no depende del admin ya que por si solo puede crear su cuenta.

Diagrama de base de Datos



Tablas creadas

```
CREATE TABLE registrolibro(
  isbn NUMERIC PRIMARY KEY,
  titulolibro VARCHAR(255),
  autorlibro VARCHAR(255),
  anopublicacion NUMERIC,
  editorial VARCHAR(255),
  generoliterario VARCHAR(255),
  cantidadstock NUMERIC,
  estadolibro BOOLEAN,
  fecha_creacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE UsuarioBiblioteca(
  id SERIAL PRIMARY KEY,
  nombre VARCHAR(150),
  direccion VARCHAR(200),
  telefono INTEGER,
  cui NUMERIC,
  estado BOOLEAN,
  rolusuario VARCHAR(100),
  contra VARCHAR(300),
  correo VARCHAR(300),
  fecha_creacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE prestamo (
    id SERIAL PRIMARY KEY,
    id_prestamo varchar(50),  -- Identificador común para un préstamo
    Usuario_id INT REFERENCES UsuarioBiblioteca(id),
    isbn NUMERIC REFERENCES registrolibro(isbn),
    cantidad_prestada NUMERIC,
    fecha_prestamo TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    fecha_devolucion DATE,
    fecha_vencimiento DATE,
    estado BOOLEAN DEFAULT FALSE
```

Tecnologías Utilizadas

Java FX: es una biblioteca de software de código abierto que se utiliza para crear aplicaciones de escritorio y aplicaciones web enriquecidas. Desarrollada por Oracle Corporation, JavaFX está diseñada para reemplazar Swing como la principal biblioteca GUI (interfaz gráfica de usuario) de Java.

Scene Builder : es una herramienta de diseño visual para crear interfaces de usuario en JavaFX. Permite a los desarrolladores diseñar sus aplicaciones de forma intuitiva arrastrando y soltando componentes GUI sin necesidad de escribir código manualmente. Las interfaces creadas se guardan en archivos FXML, que pueden ser integrados fácilmente con el código Java para gestionar la lógica de la aplicación.

Apache NetBeans: es un entorno de desarrollo integrado (IDE) gratuito y de código abierto que soporta múltiples lenguajes de programación, como Java, JavaScript, PHP, y HTML5. Ofrece herramientas robustas para la edición de código, depuración, pruebas, y gestión de proyectos, además de tener una interfaz gráfica intuitiva.

PgAdmin: es una herramienta de administración y desarrollo de código abierto para PostgreSQL, uno de los sistemas de gestión de bases de datos relacionales más avanzados y utilizados.

1. Se utilizo Apache NetBeans como IDE de desarrollo
2. Se programo en el lengua de JAVA, programacion orientada a objetos
3. La apariencia física se utiliza scene builder
4. La mayoría de librerías de java fx

Instrucciones de configuración del entorno de trabajo

Para desarrollar una aplicación utilizando JavaFX, PostgreSQL, y Apache NetBeans, es necesario asegurarse de que todas las librerías y herramientas necesarias estén instaladas y actualizadas para garantizar una ejecución fluida del proyecto.

Instalación y Configuración de Dependencias en Apache NetBeans

Verificación de Dependencias: Antes de comenzar, es fundamental verificar si las dependencias necesarias para trabajar con PostgreSQL ya están instaladas en Apache NetBeans. Esto incluye librerías como el controlador JDBC de PostgreSQL. Para comprobar esto, navegue a la sección de dependencias del proyecto en NetBeans y verifique si el controlador PostgreSQL está presente.

Instalación de Dependencias: Si las dependencias necesarias no están presentes, se pueden añadir manualmente. Visite la página oficial de PostgreSQL para descargar el controlador JDBC correspondiente. Una vez descargado, puede agregarlo a su proyecto en NetBeans. Para ello, haga clic derecho en el proyecto, seleccione "Propiedades", y luego vaya a la sección de "Bibliotecas" para añadir el archivo JAR descargado.

Instalación de Scene Builder

Descarga e Instalación: Scene Builder es una herramienta esencial para facilitar el diseño de interfaces gráficas en JavaFX. Descargue la última versión de Scene Builder desde el sitio web oficial de Gluon. Asegúrese de que la versión de Scene Builder sea compatible con la versión de JavaFX que está utilizando.

Integración con NetBeans: Después de instalar Scene Builder, es necesario integrarlo con NetBeans. Vaya a "Herramientas" -> "Opciones" -> "Java" -> "JavaFX" en NetBeans y configure la ubicación de Scene Builder. Esto permitirá que pueda abrir archivos FXML directamente en Scene Builder desde NetBeans, facilitando así el diseño de interfaces gráficas.

Configuración y Uso de PostgreSQL

Instalación de PostgreSQL: Asegúrese de tener PostgreSQL instalado en su sistema. Puede descargarlo desde el sitio web oficial de PostgreSQL. Siga las instrucciones de instalación para configurar el servidor de bases de datos.

Configuración de la Base de Datos: Cree las tablas necesarias para su aplicación en PostgreSQL. Puede utilizar pgAdmin, una herramienta de

administración para PostgreSQL, para crear y gestionar sus bases de datos y tablas de forma gráfica.

Desarrollo del Proyecto

Diseño de Interfaces: Con Scene Builder y NetBeans integrados, puede comenzar a diseñar las interfaces de usuario de su aplicación. Utilice Scene Builder para crear las diferentes ventanas y componentes visuales necesarios, como formularios de registro, pantallas de inicio de sesión, y vistas de gestión de libros y usuarios.

Implementación de la Lógica de Negocio: Después de diseñar las interfaces, proceda a implementar la lógica de negocio en su aplicación JavaFX. Esto incluye la conexión a la base de datos PostgreSQL, la implementación de las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) para libros y usuarios, y la gestión de transacciones de préstamo y devolución de libros.

Pruebas y Depuración: Una vez implementada la lógica de negocio, realice pruebas exhaustivas para asegurar que todas las funcionalidades de la aplicación funcionan correctamente. Depure cualquier error que encuentre y optimice el código según sea necesario.

Conclusión

Asegurarse de tener todas las herramientas y librerías correctamente instaladas y configuradas es crucial para el desarrollo exitoso de una aplicación en JavaFX con PostgreSQL utilizando Apache NetBeans. Con Scene Builder facilitando el diseño de interfaces y PostgreSQL gestionando la base de datos, puede enfocarse en implementar la lógica de negocio y entregar una aplicación robusta y funcional.

Proceso de compilación del proyecto

El proceso de desarrollo de este proyecto siguió un orden lógico y sistemático para asegurar la funcionalidad y la usabilidad del sistema. A continuación se detalla cada paso realizado:

1. Diseño de Interfaz de Usuario con Scene Builder:

   Se inició el proyecto diseñando las interfaces de usuario utilizando Scene Builder. Este enfoque permitió crear una experiencia amigable e intuitiva para los usuarios finales.

2. Creación de Tablas en la Base de Datos:

Se crearon las tablas necesarias en la base de datos PostgreSQL, incluyendo tablas para los usuarios del sistema y el registro de libros.

3. Ingreso de Información en la Tabla de Libros:

   Con las tablas creadas, se procedió a desarrollar la funcionalidad para ingresar información en la tabla de libros. Esto incluyó campos como ISBN, título, autor, año de publicación, editorial y cantidad disponible.

4. Implementación de Funcionalidades CRUD para Libros:

   Se implementaron las funcionalidades para crear, leer, actualizar y eliminar registros de libros. Esto permitió gestionar de manera completa el inventario de libros.

5. Registro de Usuarios:

   Siguiendo el proceso desarrollado para los libros, se implementaron las funcionalidades CRUD para el registro de usuarios, facilitando la gestión de la información personal de los usuarios del sistema.

6. Gestión de Préstamos de Libros:

   Con los usuarios y libros registrados, se relacionaron las tablas para gestionar los préstamos de libros. Se implementaron las funcionalidades para registrar nuevos préstamos, verificando la disponibilidad de los libros y las condiciones de préstamo de los usuarios
   .
7. Visualización de Préstamos:

   Se desarrollaron vistas para que tanto los usuarios como los administradores pudieran ver los libros prestados. Esto incluyó detalles sobre los plazos de préstamo y los estados de cada transacción.

8. Devolución de Libros:

Finalmente, se implementó la funcionalidad para la devolución de libros. Esto permitió actualizar el inventario y calcular automáticamente las multas por devoluciones tardías si aplicaba.

Mejoras Propuestas o Sección de soluciones

Se le brindaron unas recomendaciones para evitar errores comunes por una mala implementación

Estructura del Código:

Mantener una estructura clara y organizada del código fuente, dividiendo el proyecto en paquetes lógicos para separar las capas de la aplicación (controladores, servicios, repositorios, entre otros). Esto ayudara evitar errores

Validaciones y Manejo de Errores:

Implementar validaciones robustas y manejo de errores para asegurar que las entradas de datos sean correctas y para manejar adecuadamente situaciones excepcionales.

Pruebas Automatizadas:

Incorporar pruebas automatizadas, tanto unitarias como de integración, para asegurar que todas las funcionalidades del sistema funcionen correctamente y se mantenga la calidad del código a lo largo del tiempo.

Documentación:

Mantener una documentación detallada y actualizada del proyecto, incluyendo manuales de usuario y guías de instalación para facilitar el uso y mantenimiento del sistema.

Optimización de la Interfaz de Usuario:

Continuar refinando la interfaz de usuario con Scene Builder para mejorar la experiencia del usuario, basándose en la retroalimentación de pruebas de usabilidad.
Estas mejoras contribuirán a hacer el proyecto más robusto, fácil de mantener y amigable para los usuarios finales.

Datos de Soporte técnico

Estudiante de ingeniería en la sede San Juan Sacatepéquez UMG

Edgar Romario Chajon Chavez

Correo: romariochajon10@gmail.com

Tel.:+50230242963

INNOVACION TECNOLOGICA GT

Telecomunicaciones y
Tecnología al alcance

Romario Chajon

Cuenta de WhatsApp Business

Link de Github
https://github.com/EdgarChajon/PrimerSemestre2024UMG.git