*UNIVERSIDAD CENTRAL MARTHA ABREU DE LAS VILLAS*

*INFORME DEL PROYECTO EN C++*

*“BIBLIOTECA”*

*AUTORA: BEATRIZ FADRAGA SANCHEZ*

*2do CIENCIAS DE LA COMPUTACIÓN*



*ABRIL DE 2024*

Este informe detalla el desarrollo de un sistema de biblioteca virtual, diseñado para gestionar libros, solicitudes de préstamo y devoluciones. El sistema permite a los usuarios agregar libros, buscar libros por título, mostrar todos los libros disponibles, solicitar préstamos, devolver libros y eliminar libros de la base de datos. Además, el sistema maneja solicitudes de préstamo, permitiendo a los usuarios solicitar libros y gestionar el estado de las solicitudes.

La biblioteca tradicional enfrenta varios desafíos en la era digital, como la gestión ineficiente de los libros y las solicitudes de préstamo, la falta de accesibilidad y la dificultad para mantener un registro actualizado de los libros disponibles y los préstamos en curso. Estos problemas pueden llevar a una mala experiencia de usuario y a una gestión ineficiente de los recursos de la biblioteca.

Para abordar estos problemas, se desarrolló un sistema de biblioteca virtual que utiliza programación orientada a objetos (POO) para modelar los conceptos de libros, solicitudes de préstamo y la biblioteca en sí. El sistema se basa en una arquitectura de clases que permite una gestión eficiente de los recursos y una interacción fácil y clara para los usuarios.

El siguiente diagrama de clases proporciona una visión general de la estructura del proyecto, incluyendo las clases principales, sus atributos y métodos, y las relaciones entre ellas.

Libro LibroDigital

-titulo: string -formato: string

-autor: string

-estaPrestado: bool +mostrarInfo(): void

+mostrarInfo(): void

+getTitulo(): string |

+isPrestado(): bool

+setPrestado(bool): void

(1,\*)

Solicitud Biblioteca

-tituloLibro: string (1,\*) -libros: vector<Libro\*>

-nombreUsuario: string -solicitudes: vector<Solicitud\*>

-estado: string +agregarLibro(Libro\*): void

+getLibros(): vector<Libro\*>

+aceptarSolicitud(): void +getSolicitudes(): vector<Solicitud\*>

+rechazarSolicitud(): void +buscarLibro(string): void

+mostrarInfo(): void +mostrarTodosLosLibros(): void

+getTituloLibro(): string +agregarSolicitud(Solicitud\*): void

+devolverLibro(string): void

+eliminarLibro(string): void

El programa se estructura alrededor de varias clases principales. La Clase Libro que representa un libro en la biblioteca. Puede ser un libro físico o digital, con atributos como título, autor y estado de préstamo. La clase Libro utiliza herencia para permitir la creación de libros digitales, que son una subclase de Libro con un atributo adicional para el formato del libro digital. Esta clase hija sobrescribe el método mostrarInfo() para incluir información adicional específica de los libros digitales.

La clase Solicitud representa una solicitud de préstamo de un libro. Contiene información sobre el título del libro, el nombre del usuario y el estado de la solicitud. Esta clase permite gestionar el estado de la solicitud, permitiendo aceptar o rechazar la solicitud.

La clase Biblioteca gestiona la colección de libros y solicitudes de préstamo. Proporciona métodos para agregar libros, buscar libros, mostrar todos los libros, gestionar solicitudes de préstamo, devolver libros y eliminar libros. Utiliza contenedores dinámicos (std::vector) para almacenar libros y solicitudes, permitiendo una gestión eficiente de los recursos.

El programa principal (main) proporciona un menú interactivo para que los usuarios puedan realizar varias operaciones en la biblioteca, como agregar libros, buscar libros, mostrar todos los libros, solicitar préstamos, devolver libros y eliminar libros. Este menú interactivo permite a los usuarios interactuar fácilmente con el sistema, realizando operaciones de gestión de libros y solicitudes de préstamo.

Se utiliza la sobrecarga del operador para mejorar la interacción entre los objetos de la clase Libro y el flujo de salida estándar (std::cout). La sobrecarga del operador << permite imprimir directamente la información de un objeto Libro en la consola, lo que facilita la visualización de los detalles de los libros en la biblioteca. Esta sobrecarga del operador se utiliza en varias partes del programa para imprimir información sobre los libros. Por ejemplo, en la función buscarLibro de la clase Biblioteca, se utiliza para imprimir los detalles de un libro que coincide con el título buscado y en la función mostrarTodosLosLibros, se utiliza para imprimir la información de todos los libros en la biblioteca.

El sistema de biblioteca virtual desarrollado aborda eficazmente los desafíos de la gestión de libros y solicitudes de préstamo en la era digital. Utilizando programación orientada a objetos y un diseño de clases bien estructurado, el sistema permite una gestión eficiente de los recursos y una interacción fácil y clara para los usuarios. Este proyecto demuestra la aplicación práctica de los conceptos de programación orientada a objetos y la importancia de un diseño de clases bien pensado para el desarrollo de sistemas complejos.