**PROGRAMACION ORIENTADA A OBJETOS**

**PRACTICA 2**

**SISTEMA DE GESTION BIBLIOTECARIO**

**INTEGRANTES:**

**Andres Yair Carvajal Bolivar**

**UNIVERSIDAD NACIONAL DE COLOMBIA SEDE MEDELLIN**

**2023**

**MANUAL DE USUARIO**

**Descripción General de la aplicación**

Nuestro objetivo es implementar un sistema de gestión bibliotecario basado en el paradigma de POO. El sistema permitirá administrar y gestionar los diferentes aspectos relacionados con una biblioteca, como el registro de libros, préstamos, devoluciones, búsqueda de libros, entre otros. El propósito es brindar una herramienta eficiente y organizada para facilitar la gestión de la biblioteca.

Cabe destacar que el desarrollo del sistema sigue los principios y conceptos de la Programación Orientada a Objetos (POO), con el fin de lograr una estructura modular, reutilizable y fácil de mantener. Aunque se han realizado ajustes y adaptaciones para cumplir con los requisitos específicos del proyecto, se ha buscado mantener la coherencia en el diseño y la implementación del sistema.

Este manual de usuario proporcionará instrucciones detalladas sobre cómo utilizar las diferentes funciones y características del sistema de gestión bibliotecario. Se explicarán los pasos necesarios para realizar diversas acciones, así como también se brindarán recomendaciones y consejos para un uso óptimo del sistema.

En primer lugar, se ha realizado un análisis detallado para determinar las clases más adecuadas que representen de manera óptima el problema de gestión y organización de datos en un sistema bibliotecario. Con base en este análisis, se ha llegado a la conclusión de que se necesitan 11 clases en la maqueta lógica del sistema.

Estas clases se dividen en tres paquetes principales: el paquete de Obras, el paquete de Personas y el paquete de Préstamo. En el paquete de Obras se encuentran los módulos Estantería, Folleto, Libro, Revista y Publicación, que representan los diferentes tipos de obras que pueden encontrarse en una biblioteca. En el paquete de Personas se encuentran los módulos Autor, Persona, Estudiante, Profesor, Externo y Usuario, que representan los diferentes roles y tipos de personas involucradas en la biblioteca. Por último, en el paquete de Préstamo se encuentra el módulo del mismo nombre, que se encarga de gestionar los préstamos de obras.

Cada una de estas clases ha sido diseñada y estructurada de manera coherente para describir y simular de la mejor manera posible un sistema bibliotecario. Se han implementado operaciones CRUD (Crear, Leer, Actualizar y Eliminar) y otras funcionalidades que agilizan la gestión de datos en el sistema.

El objetivo principal de esta estructura de clases es proporcionar una base sólida y eficiente para el desarrollo de un sistema de gestión bibliotecario que cumpla con los requisitos y necesidades del usuario.

El Sistema se desarrolló en Python dentro de 3 paquetes:

* Uno llamado *baseDatos*, donde se almacenan los datos (objetos) creados en bytes y se diseñan el archivo necesario para serializar y deserializar objetos
* Otro llamado *gestorAplicacion*, donde se programó toda la capa lógica; se definen todas las clases a almacenar con su respectivos atributos y métodos
* Un último paquete llamado ventanas, donde se desarrollaron todos los archivos necesarios para implementar la interfaz de una forma adecuada y organizada

Todos estos paquetes se encuentran en la carpeta *src*, donde también se ubica el respectivo archivo main (*Main.py*). Cabe agregar que la aplicación está hecha para lanzarse abriendo la carpeta *practica2* y ejecutando el archivo *Main.py*

Adicionalmente para garantizar la persistencia de la información, al iniciar la *Ventana de Inicio*, se crean todos los archivos donde se guardarán los objetos en el dispositivo en forma de bytes (si estos no están creados). Luego, cuando se ingresa al sistema y se abre la *Ventana (Grafica) Principal*, se cargan todos los datos de los archivos almacenados anteriormente y; al salir de la *Ventana Principa*l mediante la opción *Guardar y Salir*, se guardan todos los objetos que se crearon o modificaron durante la navegación en el sistema.

**Descripción de las funcionalidades implementadas**

Las funcionalidades que implementamos son:

1. Ingresar Registros: Solicita el tipo de dato (Objeto) que se desea ingresar y se recogen valores que servirán como argumentos para los constructores. Adicionalmente, los números que funcionan como identificadores únicos se generan automáticamente
2. Mostrar registros: Una vez el usuario seleccione el tipo de datos (Objeto) del que desea información; puede ver una lista resumida con información de todos los objetos de dicho tipo almacenados o puede ver información más detallada de dicho objeto ingresando su identificador único correspondiente
3. Eliminar registros: Usando los identificadores únicos, se puede borrar cualquier objeto almacenado, indicando antes que tipo de dato se desea borrar
4. Registrar préstamos: Inicialmente se preguntará cuántos préstamos desea registrar, puesto que cuando se haga clic en *Registrar Préstamos* se genera una interfaz que se debe llenar para registrar el préstamo que se repite em base al número de préstamos que el usuario indicó que haría
5. Registrar devoluciones: Así como se puede registrar cuando se presta una publicación, con el *código* de dicho objeto se puede registrar cuando se esta se devuelve (la fecha no es una variable de entrada para garantizar que no se altere dicha fecha y sólo se utilice la actual)

**Consideraciones adicionales para el adecuado funcionamiento:**

La interfaz es bastante clara e intuitiva. La idea es dejarse llevar por las opciones que se ofrecen en pantalla e interactuar con los menús, botones y entradas. Al describir las funcionalidades también se dio un buen preámbulo al modo de funcionamiento por si hay alguna duda y por medio de mensajes emergentes se le indican muchos errores que puede cometer el usuario al ingresar datos

Las siguientes unas normas que hay que seguir para garantizar el funcionamiento adecuado del sistema:

1. Inicialmente se abre una Ventana de Inicio con información de desarrolladores y de la aplicación, para ingresar al sistema, se debe hacer clic en el botón que indica (ubicado en la parte inferior izquierda)
2. Para hacer uso de las funcionalidades se debe hacer uso de la opción *Procesos y Consultas* ubicada en la parte de arriba de la ventana (en el menú). Si se equivoca de objeto, puede dar clic en la funcionalidad para que nuevamente salga la opción de elegir uno nuevo
3. Debe estar atento a hacer uso de los botones de aceptar para que almacene correctamente un dato (si no se genera ningún mensaje emergente de confirmación significa que no se ha almacenado correctamente)
4. Para poder guardar los datos (Objetos) creados en su dispositivo debe hacer uso de la opción *Guardar y Salir* ubicada en la sección *Archivo* del menú de la ventana. Si no lo hace y, por ejemplo, cierra la ventana Principal con la *equis(x),* los archivos se eliminarán una vez cierre a la ventana de Inicio. Teniendo en cuenta lo anterior, si quiere deshacer todos los cambios hechos en una sesión de trabajo en el sistema; cierre la ventana de Inicio
5. Otro aspecto importante es que para poder registrar una publicación (libro, revista o folleto) debe tener un número de la estantería en la que se debe ingresar (la cual debe estar ya registrada) .Si este número no corresponde a un objeto de su correspondiente tipo ya creado, no se podrá completar el proceso
6. Análogamente, debe tener el número de un id almacenado para poder enlazarlo cuando se está ingresando un nuevo libro
7. Para eliminar registros se necesita tener el correspondiente código diferenciador del objeto. Para hallarlo simplemente haga uso de la funcionalidad *Mostrar Registros*
8. Para realizar un préstamo asegúrese de que la publicación y el usuario están registrados (esto lo puede hacer utilizando la implementación de *Mostrar Registros*). Si no se asegura de esto e ingresa un ID de un usuario no almacenado, no se podrá completar el proceso. Hay un id del préstamo que se asigna internamente de forma automática