**MANUAL DE USUARIO**

**Descripción General de la aplicación**

Nuestro objetivo es implementar un sistema de gestión para una biblioteca basado en el paradigma de POO y en la forma de almacenar datos de la Biblioteca Efe Gómez de la Universidad Nacional. Cabe aclarar que no se sigue estrictamente toda la metodología tratada allí, dado que para poder implementar lo pedido con el trabajo se adecuaron ciertos puntos que no concuerdan como tal.

En primera instancia se analizan las clases que podrían representar de la manera más optima el problema de gestión y organización de los datos en una biblioteca, así que basándonos en la biblioteca Efe Gómez se llegó a la conclusión de representar mediante 11 clases ( en cuanto a la maqueta lógica) el problema planteado. Estas clases se dividen en tres paquetes principales que son: paquete de Obras en donde se encuentran los módulos Estantería, Folleto, Libro, Revista y publicación; el paquete de Personas donde se encuentran los módulos Autor, Persona, Estudiante Profesor, Externo y Usuario; además un paquete de Préstamo donde se encuentra un módulo con el mismo nombre. Todo el conjunto de clases que conforman este sistema busca describir y simular de la mejor manera un sistema bibliotecario a través de operaciones CRUD y demás implementaciones que agilizan la gestión del sistema de datos

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 (*Inicio.py*). Cabe agregar que la aplicación está hecha para lanzarse abriendo la carpeta *practica2-grupo2-Equipo3* y ejecutando el archivo *Inicio.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 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
2. Mostrar registros
3. Eliminar registros
4. Registrar préstamos
5. Registrar devoluciones
6. Realizar Préstamos: Consiste en enlazar un préstamo a un usuario. Intervienen:
   1. EstudianteProfesor y Externo: implementa la interfaz usuario y heredan de la clase Persona
   2. Prestamo: Es lo que se va a registrar;
   3. Publicación: Revista o Libro
   4. Autor: teniendo en cuenta que la publicación debe estar asociada a uno
   5. Múltiples métodos para llegar al método final
7. Realizar una Renovación: Consiste en corroborar si existe el préstamo ingresado y ampliar el plazo de la fecha de vencimiento de este. Intervienen:
   1. EstudianteProfesor y Externo: implementa la interfaz usuario y heredan de la clase Persona. Hay una instancia asociada al préstamo que debe estar registrado
   2. Prestamo: A este se le cambia el atributo fin (fecha de vencimiento)
   3. Publicación: Revista o Libro
   4. Autor: teniendo en cuenta que la publicación debe estar asociada a uno
   5. Múltiples métodos para llegar al método final

**Consideraciones para el adecuado funcionamiento:**

La interfaz por medio de menús es bastante clara e intuitiva. La idea es dejarse llevar por las opciones que se ofrecen en pantalla e ir usando sus funcionalidades.

Las siguientes son las normas que hay que cumplir para garantizar su funcionamiento adecuado:

1. Tenga en cuenta que cuando en los menús de opciones el usuario digite un número incorrecto , se volverá a desplegar el menú de opciones, esto se repetirá hasta que se ingrese la opción válida. También hay unos pocos casos en los que directamente se devolverá a un menú anterior, pero en estos no hay mayor complejidad, simplemente no se completa la operación.
2. Hay que ser precisos con el tipo de dato que se ingrese , puesto que si se ingresa uno incorrecto arrojará error y tocará volver a correr el programa. Afortunadamente solo se ingresan 2 tipos de datos (o numéricos u listas de caracteres) y , para general mayor claridad, antes de solicitar ingresar algún dato la interfaz mostrará que dato se está ingresando y entre paréntesis que tipo de dato se debe ingresar.

Las convenciones para estos tipos de datos son:

* 1. (N) : Simboliza que se debe ingresar un dato únicamente numérico
  2. (S) : Simboliza que se debe ingresar un dato alfanumérico

1. Para que se lea la información correspondiente, el usuario debe presionar la tecla “enter” justo después de terminar de escribir el dato que se quiere almacenar. Sea muy cauteloso a la hora de ingresar una fecha válida en el formato correspondiente, puesto que también se lanzará un error que implicará un reinicio del sistema si no se ingresa adecuadamente
2. 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, el programa fallará y tendrá que ejecutarse nuevamente
   1. Si desea ingresar la publicación con una estantería desconocida, ingrese el número 0 cuando se le pida el número de estantería
3. El autor se creará al momento de ingresar un libro si no tiene el código de un autor ya creado
4. Para eliminar registros se necesita tener el correspondiente código diferenciador del objeto. Para hallarlo simplemente ir a mostrar registros de los menús de borrado o entrar directamente a la funcionalidad de mostrar registros. Dicho código diferenciador estará precedido de “(CP)”
5. 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, se genera un error y se deberá reiniciar el sistema. Además, el usuario le debe asignar manualmente el ID al préstamo (este no debe estar registrado en ninguna otra parte antes) y el día en el que se quiere iniciar el préstamo
6. Para realizar una renovación asegúrese de que el usuario y el préstamo 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 o un código de préstamo no almacenado, se genera un error y se deberá reiniciar el sistema.
7. Para almacenar en archivos y hacer que perduren los objetos creados debe elegir la opción 6 del menú principal (“Guardar datos y salir del sistema”)