**MEMORIA ESCRITA**

**Descripción General**

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 mas 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 modulo con el mismo nombre. Todo el conjunto de clases que conforman este sistema buscan describir y simular de la mejor manera el sistema bibliotecario a través de operaciones CRUD y demás implementaciones que agilizan la gestión del sistema de datos

El Sistema se desarrollo en Java dentro de 3 paquetes:

* Uno llamado baseDatos, donde se almacenan los datos (objetos) creados y se diseñan todas las clases necesarias 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 uiMain, donde se implementa toda la interfaz para que el usuario haga las operaciones que necesite mediante el despliegue de menús en pantalla, elección de opciones y recolección de datos (o números o cadenas de caracteres)

Adicionalmente para garantizar la persistencia de la información al iniciar la interfaz del usuario se cargan todos los objetos anteriormente creados y al finalizar el programa y digitar la opción de guardar cambios, serializamos todos los datos creados y los almacenamos para un uso posterior

**Descripción del diseño estático**

Los diagramas se encuentran en el documento adjunto con el nombre “diagramas.pdf”

**Descripción de la implementación de características de POO**

Aquí se copian donde se pudieron todos los requisitos a implementar

**Descripción de las funcionalidades implementadas**

Las funcionalidades que implementamos son:

Manual de Usuario:

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.

Algunos puntos clave que se deben tomar en cuenta es que 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:

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

Además, tenga en cuenta que 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 y en el formato correspondiente, puesto que también se lanzará un error que implicará un reinicio del sistema si no se ingresa adecuadamente

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.

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) y un código para enlazarlo con un autor correspondiente. Si estos dos números no corresponden a un objeto de su correspondiente tipo ya creado, el programa fallará y tendrá que ejecutarse nuevamente

En caso de que no se sepa el autor del libro (o directamente sea Anónimo) y se le solicite, puede asignarle el autor Anónimo, el cual ya está creado y su id correspondiente es el 0

Análogamente, si desea ingresar la publicación con una estantería desconocida, ingrese el número 0 cuando se le pida el número de estantería

NOTA: los autores se crean a la hora de crear un libro (corregir en inbterfaz y acça en memoria escrita)