**PROGRAMACION ORIENTADA A OBJETOS**

**PRACTICA 2**

**SISTEMA DE GESTION BIBLIOTECARIO**

**INTEGRANTES:**

**Andres Yair Carvajal Bolivar**

**Juan Jose Pineda Sanchez**

**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

**USO DEL SISTEMA:**

**-Gestor Bibliotecario:**

**Es la pantalla principal del proyecto, aquí se el usuario tendrá opciones para ver imágenes relacionadas al proyecto, información de los desarrolladores del proyecto, imágenes de los desarrolladores del proyecto, ver y abrir en una ventana emergente de navegador predeterminado la hoja de vida de los das arrolladores, ver la descripción de la aplicación, mensaje Bienvenida y ejecutar la ventana de ingreso al sistema.**

**IMAGENES DEL SISTEMA: estas imágenes cambian con cada ingreso del ratón nuevo al Label (imagen) del sistema. CAMBIO DE DESARROLLADOR: la información de imagen y texto de cada desarrollador cambia con clic en el texto. ABRIR HOJA DE VIDA: la ventana emergente de navegador es ejecutada cuando se da clic en la primera imagen de cada desarrollador INICIO/DESCRIPCION: el mensaje de Bienvenida cambia cuando se da clic en el menú inicio y luego en descripción. INICIO/SALIR: la aplicación se cierra cuando se le da clic en el menú inicio y luego salir. BOTON INGRESAR AL SISTEMA: Una nueva ventana emergente aparece y la actual se minimiza cuando se le da clic al botón ingresar al sistema.**

**-Sistema de gestión Bibliotecario:**

**esta ventana posee todos los procesos necesarios para el uso de las funcionalidades del sistema, aquí se podrán ingresar registros mostrar registros, Eliminar registros, realizar prestamos, realizar devoluciones, volver a la misma interfaz de bienvenida, guardar y salir del sistema, obtener ayuda acerca de los programadores.**

**INTERFAZ DE INICIO: esta interfaz posee el mensaje de explicación acerca de las funcionalidades.**

1. **Ingresar Registros:**

**Al hacer clic en la opción del menú "Procesos y Consultas" y luego en el submenú ingresar registros el usuario puede dar clic en una lista desplegable la cual posee 4 opciones:**

**\*estantería:**

**Al hacer clic en la opción del menú El usuario va a visualizar el número de estanterías que han sido creadas a lo largo de la existencia del programa. CONSIDERACION 1: el Numero de estantería no necesariamente corresponde al id que se le va a asignar a esta estantería, pues debido a que este atributo es un caunter de las creaciones no disminuye cuando se elimina una estantería, en cambio este valor queda presente en el sistema.” Procesos y Consultas" y luego en el submenú ingresar registros el usuario puede dar clic en una lista desplegable la cual posee 4 opciones:**

**El usuario también debe asignar un Piso que es donde ira la estantería que se está creando la cual debe ser un numero entero, en caso de asignar cual otro valor la aplicación lanzará un error interno el cual no está vinculado como una excepción por lo tanto si se intenta dar click en el botón Aceptar no se enviará la petición correctamente y el sistema tendrá la apariencia no haber realizado ninguna acción. El usuario también debe ingresar un Limite Sup y un Limeta inf los cuales deben ser strings y se busca que sean letras con el objetivo de darle sentido a la aplicación, en caso de ingresar un numero entero en lugar de un String el sistema lanzara una excepción con un mensaje de informativo de alerta, por otro lado, en caso de ingresar un numero en forma de decimal el sistema lo tomara como un Sting y se registrara correctamente.**

**\*Autor:**

**El usuario podrá ingresar el Nombre, el Id será automático por medio de un Sting de una letra y un número, el usuario**

**podría ingresar la fecha de nacimiento como un string, al igual que el País y si está vivo o no.**

**CONSIDERACION 2: tener en cuenta que en el atributo Nacimiento debe ir un string, en caso de asignar un entero el**

**sistema lanzara una excepción con una ventana de alerta informativa.**

**\*publicación:**

**el usuario tendrá otra lista desplegable en la que debe elegir el tipo de publicación entre:**

**+ Libro:**

**el usuario debe ingresar el Nombre el cual debe ser un String, en caso de ser un numero el sistema lo**

**tomará como un string, al igual que AÑO y ejemplar. El usuario debe ingresar un Autor del libro que puede ser**

**un Autor creado por el sistema de gestión bibliotecario el cual podría ser asignado por medio del ID que posee.**

**CONSIDERACION 3: estas opciones no manejan excepciones por lo que independientemente de la información suministrada**

**por el usuario el Libro será creado, para que pueda funcionar correctamente posterior a la creación y tenga sentido**

**se recomienda ingresar información acorde con la lógica del programa, por el contrario, simplemente se continuará con el**

**flujo normal de la aplicación.**

**+ Folleto:**

**el usuario podrá agregar un string con un Nombre, Año, Ejemplar, Referencia, estantería.**

**CONSIDERACION 4: al igual que para la opción Libro, Folleto no usa excepciones acerca de las entradas suministradas**

**por el usuario pero se busca que sea en el orden: String, integer,integer, integer,integer, integer, siendo el ultimo**

**integer una estantería creada previamente.**

**+ Revista:**

**el usuario podrá ingresar : Nombre (string), Año ( integer), Ejemplar ( Integer), Numero (integer), Mes(String),**

**Temporada (Integer), estantería (integer).**

**CONSIDERACION 5: tener en cuenta que este apartado no tiene manejo de excepciones.**

**\*Usuario:**

**el usuario tendrá otra Lista Desplegable en la cual tendrá dos opciones:**

**+interno:**

**el usuario podrá agregar Nombre, id, Correo, Telefono,Direccion, Nacimiento, país Rol como atributos del Usuario**

**que se está creando.**

**CONSIDERACION 6: EL APARTADO NO MANEJA EXCEPCIONES DE ENTRADA.**

**+externo:**

**el usuario podrá agregar Nombre, id, Correo, Telefono,Direccion, Nacimiento, país Rol y Universidad**

**como atributos del Usuario que se está creando.**

**CONSIDERACION 7: EL APARTADO NO MANEJA EXCEPCIONES DE ENTRADA.**

**2. Mostrar Registros:**

**el usuario podrá seleccionar una opción de una Lista Desplegable: estantería, Autor, publicación, Usuario, préstamo**

**todas las opciones una vez seleccionadas tendrá un botón para ver un texto con la lista total de registros y la información de sus atributos**

**o un campo para ingresar el id especifico de la opción que busca.**

**CONSIDERACION 8: solo Ver Registros no posee el campo de búsqueda por ID.**

**3. Eliminar Registros:**

**el usuario tendrá una lista desplegable con las opciones estantería, Autor, publicación, Usuario. Una vez elegida la opción**

**tendrá un campo en donde debe introducir el Numero del objeto que desea Eliminar.**

**4. Registrar Préstamos**

**El usuario tendrá un campo en donde debe introducir un integer con la cantidad de préstamos que un usuario en especifico**

**desea realizar, en caso de introducir algun otro tipo de dato el sistema lanzara un error interno por lo que el sistema**

**dará la impresión de no hacer nada. (no hay un manejo de excepción). Una vez introducido el tipo de dato correcto el usuario**

**podrá seleccionar un ID para el préstamo el cual debe er un integer, debera introducir una fecha, un ID de publicacion, y un ID**

**del usuario en cuestión que desea hacer el préstamo.**

**CONSIDERACION 9: EN CASO DE INTRODUCIR UN OBJETO NO EXISTENTE LANZARA UNA EXPCION CON UNA VENTANA Y TEXTO INFORMATIVO.**

**5. Registrar Devoluciones**

**el usuario debe introducir el código del préstamo en cuestión que desea eliminar (se considera devolución al momento que**

**se elimina el préstamo de la base de datos.).**

**-DETALLES DEL CODIGO:**

**En la interfaz principal de la aplicación, se utiliza un frame principal llamado WindowBegin, el cual actúa como contenedor principal de todos los elementos. Este frame representa la ventana principal de la aplicación y tiene un tamaño de 1000x1000 píxeles.**

**Dentro de este frame principal, se encuentran dos frames secundarios: p1 y p2.**

**El frame p1 está ubicado en el lado izquierdo de la ventana principal. Tiene un ancho de 500 píxeles y una altura de 1000 píxeles. Este frame contiene elementos como una descripción de bienvenida y algunas imágenes relacionadas con el sistema.**

**El frame p2 se encuentra en el lado derecho de la ventana principal. Tiene un ancho de 490 píxeles y una altura de 1000 píxeles. Este frame contiene elementos relacionados con la hoja de vida, como texto y varias imágenes.**

**Dentro del frame p1, se encuentran dos sub-frames: p3 y p4.**

**El frame p3 está ubicado en la parte superior del frame p1. Tiene un ancho de 500 píxeles y una altura de 155 píxeles. En este frame se muestra una descripción de bienvenida al sistema.**

**El frame p4 se sitúa en la parte inferior del frame p1. Tiene un ancho de 500 píxeles y una altura de 600 píxeles. En este frame se muestran algunas imágenes relacionadas con el sistema y un botón para ingresar al sistema.**

**En el frame p2, también se encuentran dos sub-frames: p5 y p6.**

**El frame p5 está ubicado en la parte superior del frame p2. Tiene un ancho de 490 píxeles y una altura de 150 píxeles. En este frame se muestra un texto que corresponde a la hoja de vida.**

**El frame p6 se sitúa en la parte inferior del frame p2. Tiene un ancho de 500 píxeles y una altura de 700 píxeles. En este frame se muestran varias imágenes relacionadas con la hoja de vida.**

**En resumen, la organización de los frames es jerárquica, donde el frame principal WindowBegin contiene los frames p1 y p2. A su vez, p1 contiene los sub-frames p3 y p4, mientras que p2 contiene los sub-frames p5 y p6. Cada sub-frame tiene su propio propósito y contiene los elementos correspondientes.**

**description(): Esta función se activa cuando el usuario selecciona la opción "Descripción" en el menú de inicio. Su propósito es mostrar una descripción detallada del sistema en el marco p3. La mejora que hemos implementado en esta función es ocultar cualquier otro contenido que pueda estar presente en p3 antes de mostrar la descripción, lo que garantiza que solo se muestre la información relevante al usuario.**

**change\_image(): Esta función se utiliza en conjunto con eventos de movimiento del mouse en el marco p4. Su objetivo es cambiar la imagen mostrada en el marco p4 cada vez que el mouse pasa por encima de ella. Se ha optimizado esta función para asegurar que las imágenes se alternen correctamente y de forma cíclica, permitiendo una visualización fluida y atractiva para el usuario.**

**p4\_enter(e) y p4\_leave(e): Estas funciones se activan cuando el ratón entra y sale del marco p4, respectivamente. La mejora que hemos implementado en estas funciones es controlar de manera más eficiente la habilitación y deshabilitación del cambio de imagen, lo que garantiza que solo se produzcan cambios cuando el mouse está dentro del marco p4, evitando así cambios innecesarios y mejorando la eficiencia del sistema.**

**load\_images(): Esta función se encarga de cargar las imágenes y mostrarlas en el marco p6, junto con la funcionalidad adicional de abrir un archivo PDF relacionado con cada imagen al hacer clic en ella. La mejora que hemos implementado aquí es la carga eficiente de las imágenes en segundo plano, lo que permite una visualización instantánea sin retrasos perceptibles para el usuario.**

**change\_text\_and\_images(event=None): Esta función se activa cuando se hace clic en el texto en el marco p5. Su propósito es cambiar tanto el texto como las imágenes mostradas en los marcos p5 y p6, respectivamente. La mejora que hemos implementado aquí es asegurar que tanto el texto como las imágenes se actualicen correctamente y de manera sincronizada, brindando una experiencia coherente y agradable para el usuario.**