Documentación Sobre el Código:

*Por Santos de la Torre Montalvo*

# Definición:

Se trata de un programa planteado para la gestión de unos registros de Libros y sus correspondientes préstamos. Basado en la utilización de ficheros de extensión .txt, de tal forma que en una localización concreta se generan unos ficheros donde se almacenaran dichos datos.

En este caso al ser una tarea se sobre entiende que no es necesario elegir la dirección de guardado de estos archivos, y se guardaran directamente en la carpeta del proyecto.

La aplicación es de sencilla utilización y se plantea para el uso de usuario de bajo conocimiento informático. Ejemplos de uso:

+ Bibliotecas pequeñas

+ Registro personal

# Especificaciones:

Se trata de un proyecto en java, con la utilización del JavaSE-19, que no utiliza ninguna librería ajena. Y que utiliza en este listado de clases de java:

+ java.awt

+ java.util

+ javax.swing

+ java.text

+ java.io

+ extends de Serializable

Por lo tanto, para editarlo quizás sea necesario una versión igual o superior del JDK.

Cabe destacar que al ser un proyecto no se han realizado test exhaustivos sino los propios de un desarrollo básico. Por tanto, solo se garantiza su funcionamiento en Windows 11, con la versión de java correspondiente.

Se trata de un proyecto creado originalmente para esta tarea, y en el que no se han implementado códigos antiguos, por mi persona, Santos de la Torre Montalvo

# Procedimiento de uso:

Al iniciar la aplicación se mostrará la página de inicio, con una tabla y diversos botones de navegación, junto con un botón a la derecha del todo en la parte superior que sirve para cambiar el orden, y facilitar la búsqueda de libros.

Si se accede al apartado de agregar libro, se podrán meter datos para generar el registro de un libro. Se obliga al usuario a que todos los campos estén rellenos, para evitar problemas de formato y tener un registro mucho más exhaustivo de los datos.

Si se accede a la pantalla de historial de registros, se podrán observar en la tabla los registros existentes de préstamos. Siendo esta ventana un reflejo de la pantalla principal en cuanto a funciones. Se destaca también que debido a la complejidad de comparar Dates, no sé a implementado su orden. Por lo tanto, en esta ventana no se podrá ordenar nada más que por libro o autor

Si se accede a la pantalla de registro de préstamos se podrá registra un prestamos o una devolución de préstamo.

(A de tenerse en cuenta que no se plantea bajo ningún concepto de la aplicación que existan varios libros con el mismo nombre)

# Diseño del modelo de datos

Se ha utilizado una jerarquía sencilla de interfaz y dos clases, para poder hacer que los métodos sean comunes, pese a que a conllevado muchos casteos, la interfaz Dato que implementa serializable y obliga a las clases que la implementan a llevar un método toString(). Y las clases que representan esos datos y implementan la interfaz dato son Libros y Prestamos.

## Libros:

Clase que sirve para crear instancias que guarden los datos de libros, para su posterior guardado en el fichero. Atributos: (String) Titulo, (String) Autor, (long) Fecha de salida y (long) ISBN

## Prestamos:

Clase que sirve para crear instancias que guarden los datos de los préstamos, para su posterior guardado en el fichero. Atributos: (Libro) Libro, (String) user, (Date) Fecha de préstamo y (Date) fecha de devolución.

## Métodos de guardado en fichero:

El programa tiene su propia clase para representar la conexión con los ficheros, y que así se sencillo de diferencia uno de otro, llamada ConexionFichero, con un constructor que te pide el nombre del fichero. Esta tiene un método de guardado en archivo, y otro para cargar desde el archivo.

Estos métodos convierten las listas de Dato a bytes automáticamente al escribir en el fichero con el método writeObject de la clase ObjectOutputstream. Que a su veces esta conectado a un fileoutputstream.