

A thick dark blue vertical bar runs down the left side of the page. A blue arrow-shaped banner points to the right from this bar, containing the date '1-1-2020'. In the bottom left corner, there are several thin, curved, light blue lines that sweep upwards and to the right.

1-1-2020

Organized Books

Biblioteca en Casa

Diseño de una interfaz utilizando Pencil y Java Swing

Antonio Gómez Giménez

i72gogia@uco.es

Grado de ingeniería informática

Universidad de Córdoba



Índice:

1. Introducción e identificación del problema.....	3
2. Especificación de requisitos.	4
2.1 Requisitos de usuario	4
2.2 Requisitos de sistema	5
2.2.1. Requisitos de información	5
2.2.2. Requisitos funcionales.....	5
2.2.3. Requisitos no funcionales.....	6
2.2.4. Requisitos de interfaz	6
3. Casos de Uso.	6
4. Iteraciones:.....	8
4.1 Iteración 1:	8
4.1.1 Diseño o especificación del interfaz	8
4.1.2 Implementación	19
4.1.3 Evaluación	28
4.2 Iteración 2:	30



1. Introducción e identificación del problema.

Muchas veces cuando hablo con mis familiares surge el tema de conversación siguiente: “Ya no se ni los libros que tengo” o “Ojalá saber donde guarde aquel libro que me gustó tanto” incluso “No sabía ni que tenía este libro”, por ello, planteando el trabajo de la asignatura se me ocurrió solucionar este problema.

El principal problema que nos encontramos es el **olvido**, es decir, si nuestra casa tiene muchas estanterías o incluso una biblioteca en miniatura, se nos puede volver algo bastante complicado buscar un libro concreto. Un segundo problema que podemos encontrar es saber **si un libro lo hemos leído o no**, ya que hay ciertas personas que cuando leen muchos libros se olvidan de aquellos que han leído he incluso de si les gustaba o no.

Cabe decir, que en la actualidad los e-books están apareciendo para quedarse y a mucha gente les gusta. Aunque estos libros no ocupan espacio en la estantería y son fáciles de encontrar en el propio e-book, si aparece el segundo problema nombrado anteriormente.

Por tanto, lo que se va a buscar con este programa es una aplicación donde un usuario pueda ver donde tiene sus libros físicos y obtener información del libro (físico o digital) como, por ejemplo, un breve resumen, cuanto le gusta ese libro, si lo ha leído, etc.

En mi caso me voy a centrar en la medida de lo posible en la interfaz, creando una interfaz sencilla e intuitiva para un usuario inexperto. Por tanto, no se creará una base de datos con todos los libros y datos sobre los mismos, se usarán datos de ejemplo para ver el uso de la interfaz.



2. Especificación de requisitos.

La especificación de requisitos de software (ERS) describe el comportamiento del sistema que se está desarrollando. Esta descripción incluye también una definición las interacciones del usuario con el software.

Las interacciones usuario-sistema se describen a través de los requisitos funcionales que normalmente hacen referencia a función del sistema de software o sus componentes, también puede verse como una declaración abstracta de alto nivel de un servicio que el sistema debe proporcionar o una definición detallada y formal de una función del sistema.

De forma complementaria también existen los requisitos no funcionales, que son requisitos que imponen restricciones en el diseño o la implementación, como, por ejemplo, restricciones en el diseño o estándares de calidad.

En resumen, describe aquellas funcionalidades que se esperan sobre nuestro software y que deben satisfacer al usuario.

Se describe a continuación aquellos requisitos básicos de nuestra aplicación.

2.1 Requisitos de usuario

Los requisitos de usuario nos dicen como debe funcionar nuestra aplicación y como debe funcionar:

- **RU-1:** La aplicación debe proporcionar la capacidad de saber dónde se encuentra un libro físico de forma sencilla.
- **RU-2:** La aplicación también permitirá ver cierta información sobre libros físicos o e-books.
- **RU-3:** El usuario debe poder asignar o modificar una opinión a un libro, tanto físico como e-book para recordar la opinión sobre el libro.
- **RU-4:** La aplicación debe estar disponible en Español, quizás en futuras versiones se le aplique internalización permitiendo también estar disponible en Ingles.
- **RU-5:** EL usuario debe poder borrar y modificar tanto los libros físicos como los ebooks.
- **RU-6:** La interfaz debe ser usable y permitir buscar un libro ya sea para saber donde está o obtener más información sobre el mismo.



2.2 Requisitos de sistema

Los requisitos de sistema nos permiten explicar las restricciones y la funciones que nuestro sistema debe cumplir para que funcione de forma exitosa cumpliendo los requisitos de usuario. Por ello nos encontramos:

2.2.1. Requisitos de información

- **RI-1:** Cada libro físico a la hora de ser buscado, la aplicación nos debe decir el lugar de la casa donde se encuentra.
- **RI-2:** Cada libro, ya sea físico o e-book, si se desea obtener más información del mismo, se nos proporcionara junto al título, un breve resumen del mismo y nombre del autor, aparte del lugar, si es un libro físico.
- **RI-3:** Junto a cada libro y la información del mismo, tendremos un sistema de ranking del 1 a 5 donde podremos cambiar su valor para opinar sobre un libro, esto se aplica tanto al libro físico como al e-book.

2.2.2. Requisitos funcionales

- **RF-1:** La aplicación debe permitir opinar sobre los libros que se disponen.
- **RF-2:** La aplicación debe permitir trabajar tanto con libros físicos como e-books
- **RF-3:** La aplicación debe permitir añadir nuevos libros a la biblioteca.
- **RF-4:** La aplicación debe permitir obtener información adicional sobre los libros.
- **RF-5:** La aplicación debe dar la información de donde se encuentra un libro físico al buscarlo.
- **RF-5:** La aplicación permitir modificar un libro.
- **RF-6:** La aplicación debe permitir borrar los libros.



2.2.3. Requisitos no funcionales

- RNF-1: La solución debe realizarse para interfaz web y tableta.
- RNF-2: La aplicación debe ser sencilla e intuitiva, siendo esta fácil de utilizar y cómoda al usuario.
- RNF-3: La solución debe estar disponible para mi familia pero debe poder ser utilizada para terceros que tengan un problema similar.

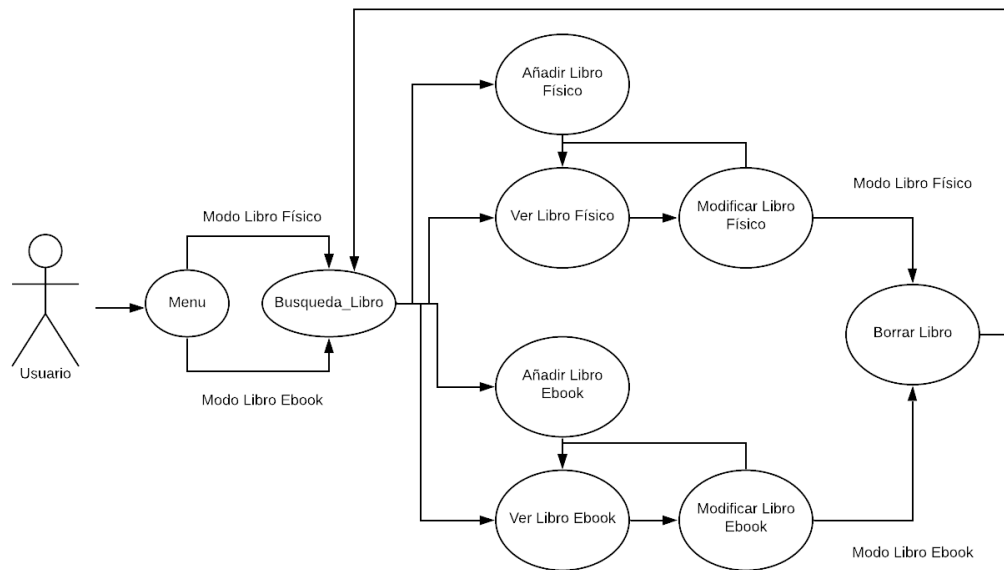
2.2.4. Requisitos de interfaz

- **RINT-1:** El usuario accede a un menú principal donde puede elegir entre libro físico y e-book.
- **RINT-2:** La aplicación debe tener una interfaz que permita modificar un libro a la biblioteca.
- **RINT-3:** La aplicación debe tener una interfaz que permita borrar un libro de la biblioteca.
- **RINT-4:** El usuario tiene una interfaz de búsqueda para libros físicos donde le dice donde se encuentra el libro físico y al hacer click en él, debe tener más información sobre ese libro.
- **RINT-5:** El usuario tiene una interfaz de búsqueda para e-books donde le permite, al hacer click en el libro, tener más información sobre ese libro.
- **RINT-6:** La aplicación debe tener una interfaz que permita añadir un libro a la biblioteca.

3. Casos de Uso.

Un caso de uso es una descripción de pasos o secuencias que se deben seguir para hacer una función de nuestro sistema, aparte de las entidades o actores que participan y son necesarios para hacer estas funciones.

En nuestro caso voy a realizar un esquema donde se puede observar el funcionamiento completo de toda la aplicación y desde donde puede ir el usuario en cada una de las opciones.



CU0_Menu: Muestra la interfaz principal, es el menú donde se podrá escoger entre trabajar con libros Físicos o libros Ebooks.

CU1_Busqueda_Libro: En esta interfaz encontraremos un buscador que nos permitirá ver los libros para poder luego acceder a ellos. También contará con un botón para añadir libro. El funcionamiento de este caso de uso cambiará dependiendo del modo en el que estemos, es decir, si el modo es libro Físico, entonces accederemos únicamente a los nodos AñadirLibroFísico y VerLibroFísico y nos impedirá acceder a AñadirLibroEbook y VerLibroEbook.

CU2_AñadirLibroFísico: Interfaz que nos muestra unas cajas donde meter el nuevo nombre, autor, lugar, opinión y breve resumen del libro.

CU3_VerLibroFísico: Interfaz que nos muestra el nombre del libro, autor, lugar, opinión actual y breve resumen del libro. También cuenta con botón para volver atrás, como todas las ventanas y un botón para realizar cambios sobre el libro que estamos viendo.

CU4_AñadirLibroEbook: Interfaz que nos muestra unas cajas donde meter el nuevo nombre, autor, opinión y breve resumen del libro.



CU5_VerLibroEbook: Interfaz que nos muestra el nombre del libro, autor, opinión actual y breve resumen del libro. También cuenta con botón para volver atrás, como todas las ventanas y un botón para realizar cambios sobre el libro que estamos viendo.

CU6_ModificarLibroFísico: Interfaz que nos muestra el nombre del libro, autor, lugar, opinión actual y breve resumen del libro para poder modificarlo. También cuenta con botón para volver atrás, como todas las ventanas y un botón para borrar el libro que estamos viendo.

CU7_ModificarLibroEbook: Interfaz que nos muestra el nombre del libro, autor, opinión actual y breve resumen del libro para poder modificarlo. También cuenta con botón para volver atrás, como todas las ventanas y un botón para borrar el libro que estamos viendo.

CU8_BorrarLibro: Interfaz que permite borrar el libro seleccionado, antes de borrarlo avisa por si acaso fuera un fallo del usuario.

4. Iteraciones:

Por cada iteración se tiene pensado realizar todo el diseño hablado sobre los casos de uso, la implementación de estos casos (tanto interfaz como funcionalidad) y por último se evaluará el trabajo realizado para encontrar posibles fallos o incongruencias en nuestra aplicación.

4.1 Iteración 1:

4.1.1 Diseño o especificación del interfaz

Para hablar sobre el diseño se mostrará con pencil como se tiene pensado que sea la interfaz, ya que esta es la manera más fácil de visualizarlo, por tanto, lo que se va a ver en pencil se intentará realizar prácticamente igual en java swing en la medida de lo posible.

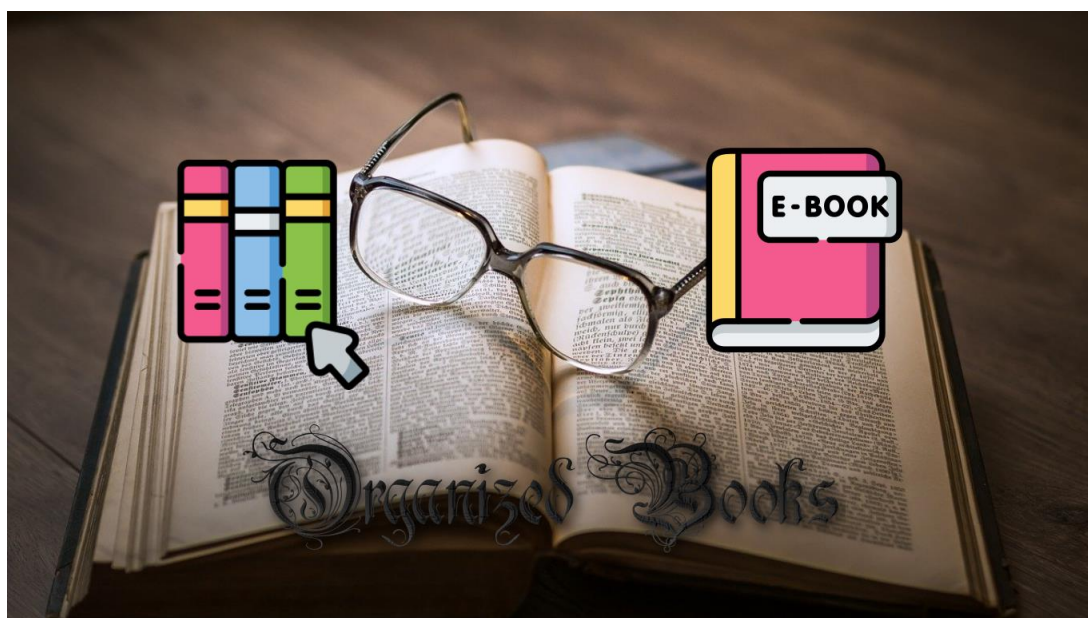
Por tanto, se va a nombrar el caso de uso y posteriormente se verá el diseño de ese caso de uso. También se mostrará una vez realizado para que se puedan apreciar las mejoras realizadas durante la fase de implementación.



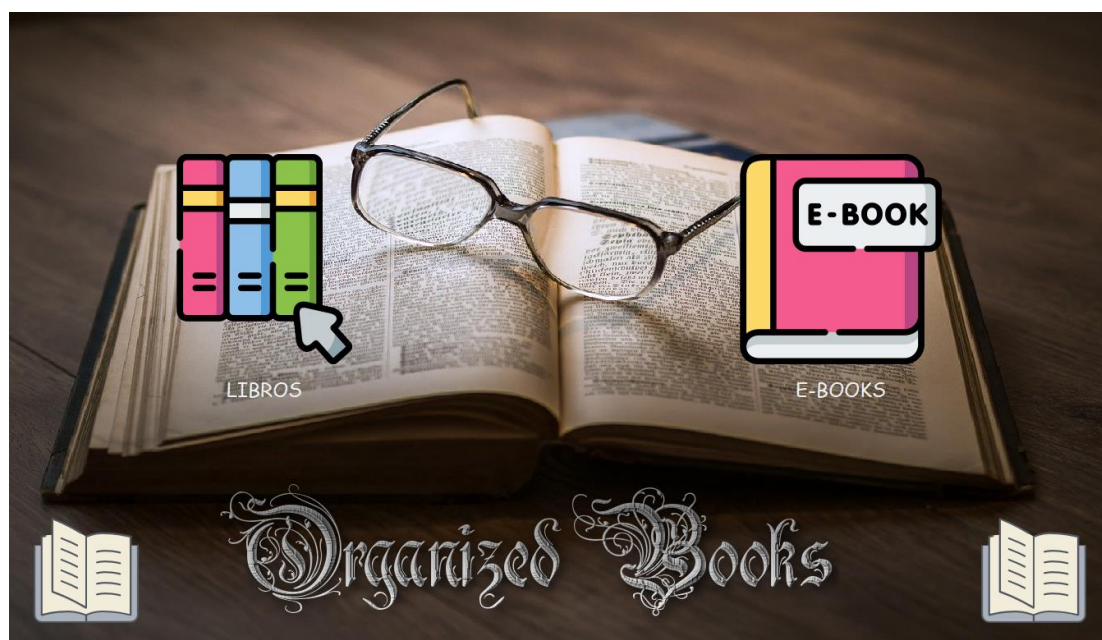
Junto al trabajo se adjuntará el pencil donde podemos encontrar todos los primeros diseños, en este caso algunos enlaces no funcionarán bien ya que por ejemplo buscar lo comparten ambos y no puedo redireccionar a Ebooks o a libros Físico desde el mismo botón, este problema no ocurrirá en el programa ya que se controlará.

Menú:

Pencil



Aplicación Final

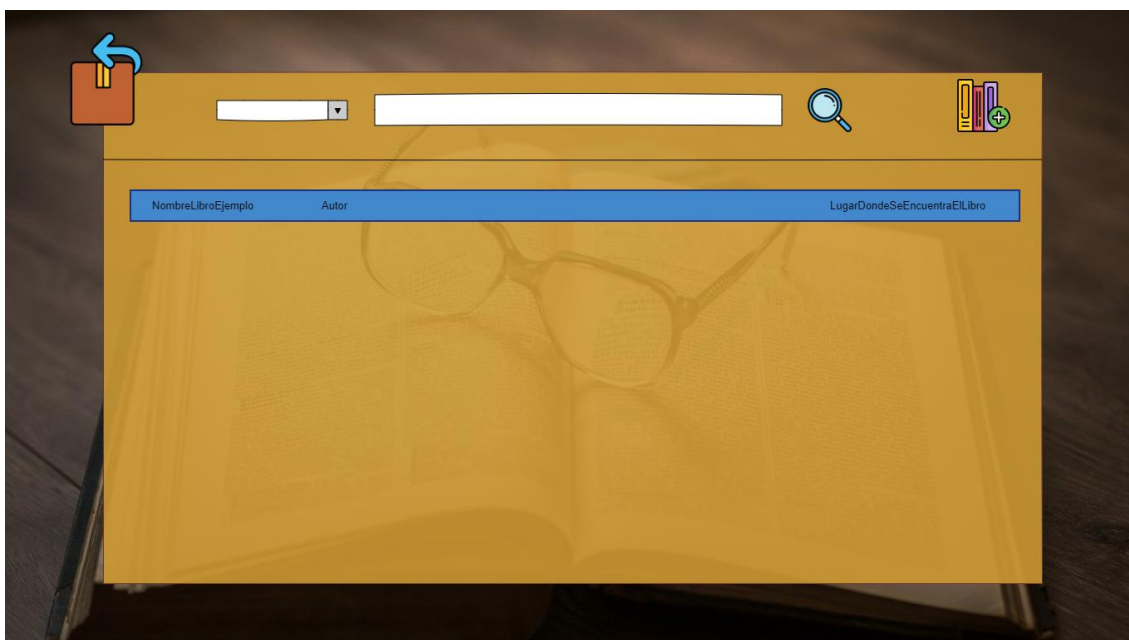
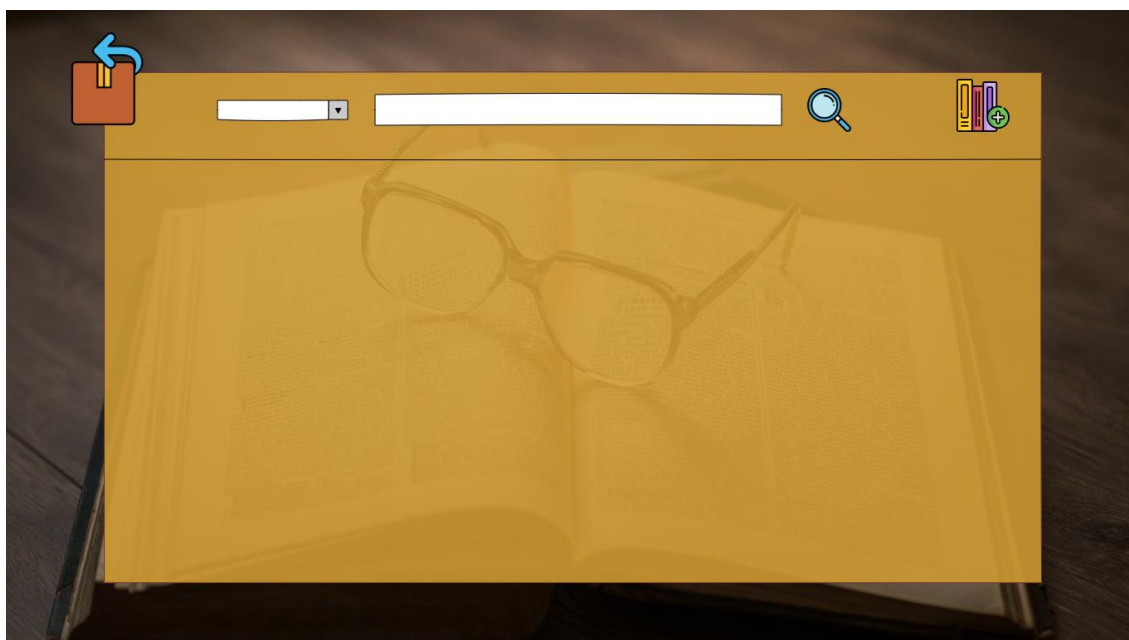




Cabe destacar que el título se ha puesto de un color más claro permitiendo ser más fácil leerlo. También se han añadido dos componentes nuevos (son decorativos), estos se han añadido ya que se iba a crear una simulación de pantalla de carga, pero por motivos de implementación, no he sido capaz y para evitar desaprovechar el componente se ha añadido en el menú de forma decorativa.

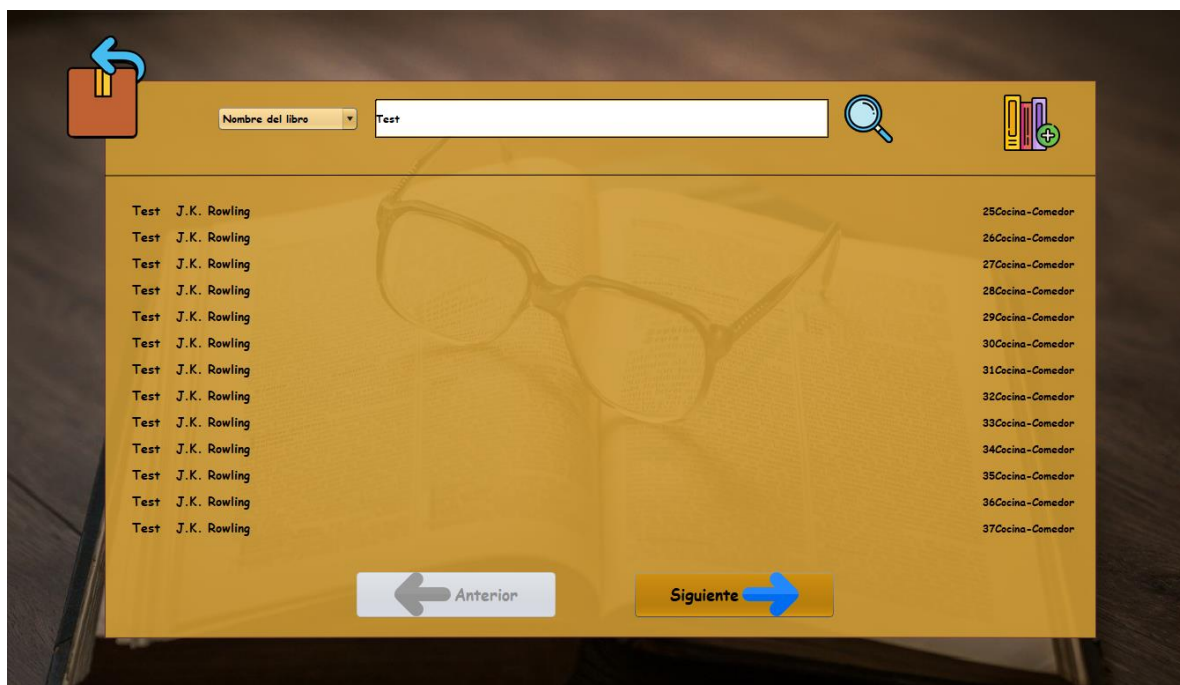
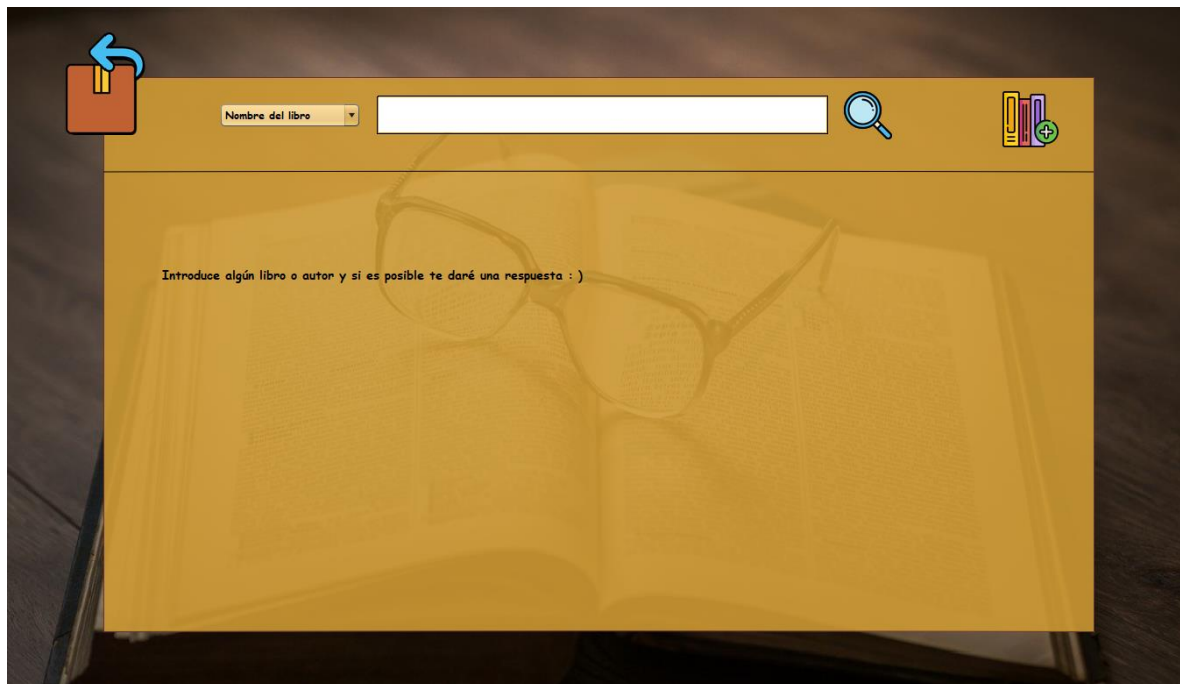
Búsqueda Libro:

Pencil





Aplicación Final



Como no se saben los libros que hay, se genera de forma dinámica esta interfaz, también se han incluido botones por si no caben todos los libros en la misma página.



AñadirLibroFísico:

Pencil

Titulo

Autor

Lugar

Nota

Componente_Opinión

Resumen

Aceptar

Aplicación Final

Titulo:

Autor/a

Lugar

Nota:
☆☆☆☆
- 0 +

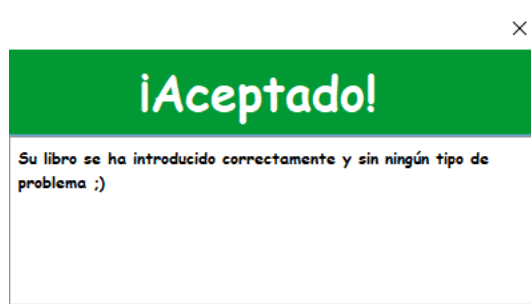
Resumen:

Aceptar



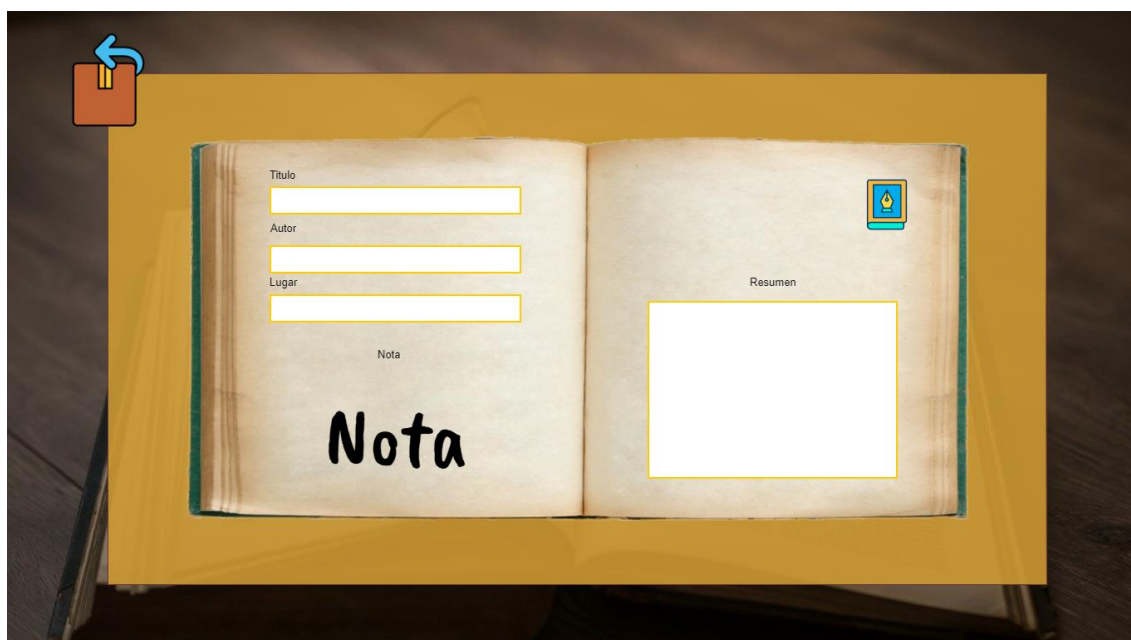
Se han realizado pequeñas mejoras y se ha añadido el componente de modificar la opinión del libro. Cabe destacar que se controlan errores por todo el programa como buscar sin añadir nada o que no se encuentre por autor o por libro, por ello se modifican las interfaces para arreglar estos problemas.

En este caso y similar en el Ebook, pueden dejarse campos vacíos, por ello salta una alerta para este problema. Ocurre lo mismo si el libro se introduce de forma correcta.



VerLibroFísico:

Pencil





Aplicación Final

The screenshot shows a digital representation of an open book. On the left page, there are four input fields: 'Título:' with the text 'Test', 'Autor/a' with 'J.K. Rowling', 'Lugar' with '28Cocina-Comedor', and a 'Nota:' field containing a large '0'. On the right page, there is a 'Resumen:' section with a sub-label 'breve resumen' and a large empty rectangular box for text entry. A small icon of a book with a flame is visible in the top right corner of the right page.

Se han realizado cambios menores.

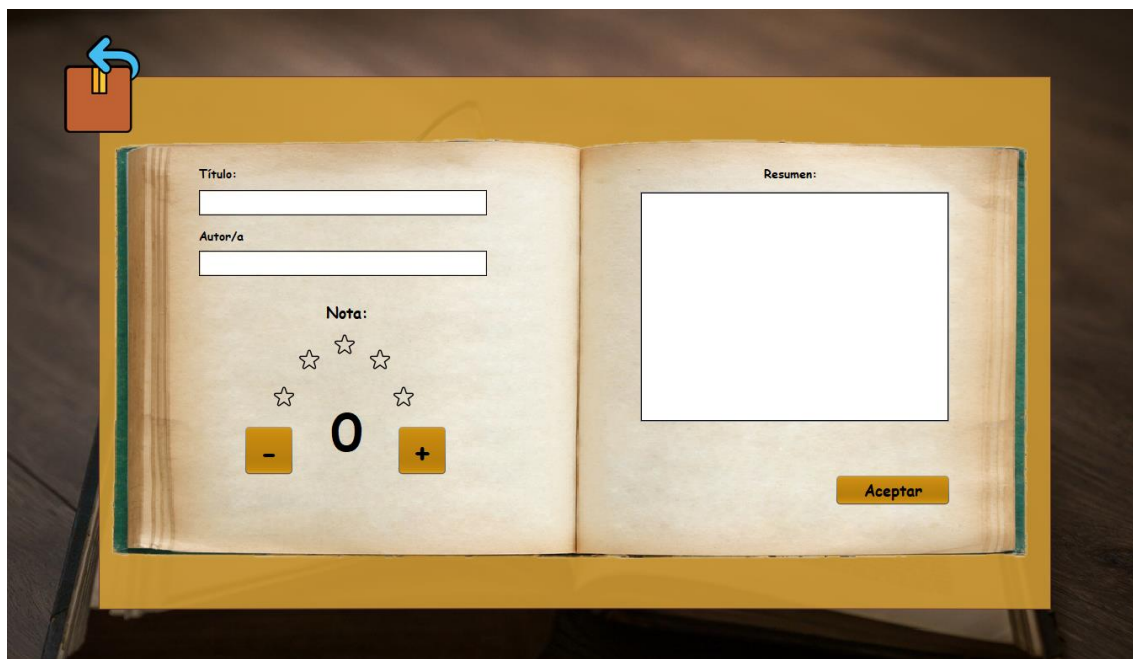
AñadirLibroEbook:

Pencil

The screenshot shows a digital representation of an open book. On the left page, there are three input fields: 'Titulo' (empty), 'Autor' (empty), and 'Nota' (empty). Below the 'Nota' field is a box labeled 'Componente_Opinión'. On the right page, there is a 'Resumen' section with a large empty rectangular box for text entry. A small icon of a book with a flame is visible in the top right corner of the right page. At the bottom right of the right page, there is a yellow button labeled 'Aceptar'.

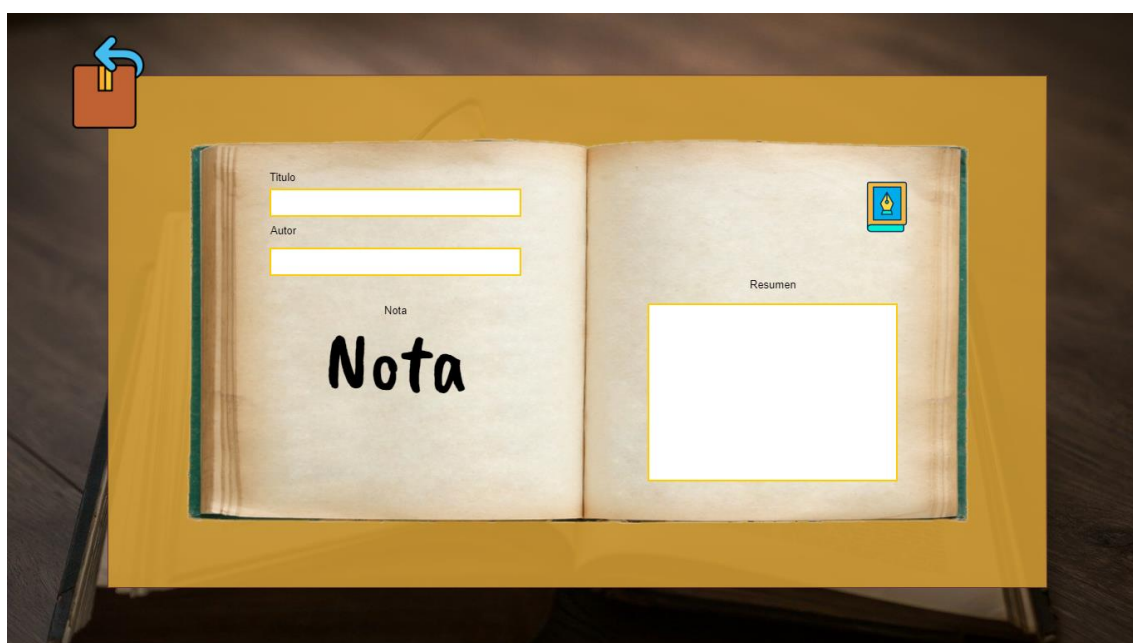


Aplicación Final



Se han realizado pequeñas mejoras y se ha añadido el componente de modificar la opinión del libro. También se encuentran las alertas mencionadas en AñadirLibroFísico.

VerLibroEbook:





Aplicación Final

The screenshot shows a digital book interface. On the left page, there are three input fields: 'Título:' with the text 'Harry Potter y la piedra Filosofal', 'Autor/a' with 'J.K. Rowling', and a large 'Nota:' field containing the number '0'. On the right page, there is a 'Resumen:' section with a large empty box for 'breve resumen'. A small icon of a book with a flame is in the top right corner of the right page. A blue arrow icon is in the top left corner of the left page.

Se han realizado cambios menores.

ModificarLibroFísico:

Pencil

The screenshot shows a digital book interface for modification. On the left page, there are four input fields: 'Titulo', 'Autor', 'Lugar', and 'Nota'. Below the 'Nota' field is a box labeled 'Componente_Opinión'. On the right page, there is a 'Resumen' section with a large empty box. At the bottom of the right page, there are two buttons: 'Borrar' and 'Aceptar'. A small icon of a book with a flame is in the top right corner of the right page. A blue arrow icon is in the top left corner of the left page.



Aplicación Final

The screenshot shows a digital book interface. On the left page, there are input fields for 'Título:' (containing 'Test'), 'Autor/a:' (containing 'J.K. Rowling'), and 'Lugar:' (containing '25Cocina-Comedor'). Below these is a rating section labeled 'Nota:' with five stars and a large '0' in the center, flanked by '-' and '+' buttons. On the right page, there is a large text area labeled 'Resumen:' with the placeholder 'breve resumen'. At the bottom right, there are two buttons: 'Borrar Libro' and 'Aceptar'. A blue arrow icon is in the top left corner.

Se han realizado cambios menores, es muy similar al de AñadirLibroFísico.

ModificarLibroEbook:

Pencil

The screenshot shows a digital book interface. On the left page, there are input fields for 'Titulo' and 'Autor'. Below them is a text area labeled 'Nota' with the placeholder 'Componente_Opinión'. On the right page, there is a large text area labeled 'Resumen'. At the bottom right, there are two buttons: 'Borrar' and 'Aceptar'. A blue arrow icon is in the top left corner.



Aplicación Final

The screenshot shows a digital interface designed to look like an open book. On the left page, there are input fields for 'Título:' (containing 'Harry Potter y la piedra Filosofal') and 'Autor/a:' (containing 'J.K. Rowling'). Below these is a rating section labeled 'Nota:' with five stars and a large '0' in the center, flanked by '-' and '+' buttons. On the right page, there is a large text area labeled 'Resumen:' with the sub-label 'Breve resumen'. At the bottom of the right page are two buttons: 'Borrar Libro' and 'Aceptar'. In the top-left corner of the interface, there is a small orange book icon with a blue arrow pointing left.

Se han realizado cambios menores, es muy similar al de AñadirLibroEbook.

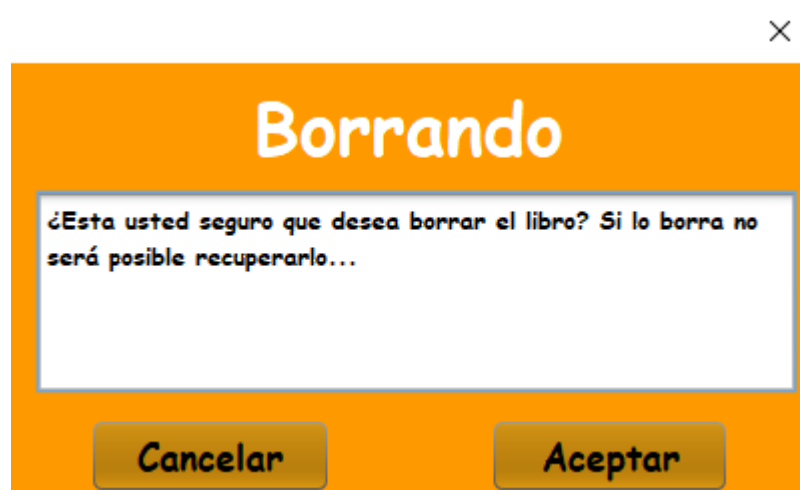
BorrarLibro:

Pencil

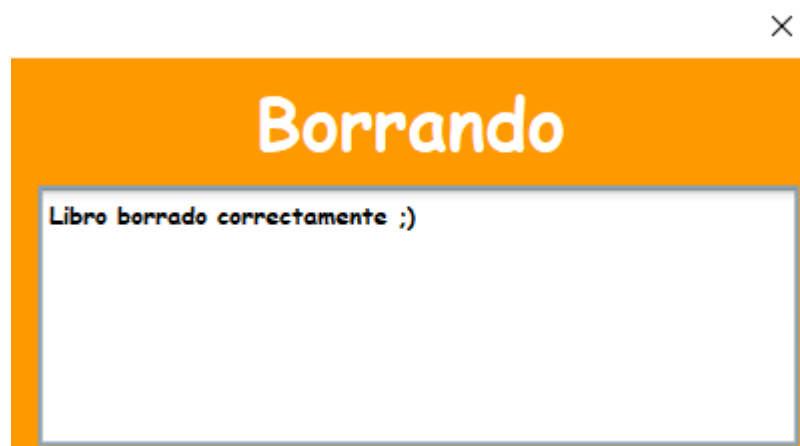
The screenshot shows a yellow dialog box titled 'Borrando' in a large, bold, white font. Inside the dialog, there is a white rectangular area containing the text 'Texto preguntando si quiere borrar el libro'. At the bottom of the dialog, there are two buttons: 'cancelar' on the left and 'Aceptar' on the right.



Aplicación Final



Se han realizado fuertes mejoras, ya que el diseño principal dejaba mucho que desear. Cabe destacar que se ha añadido la siguiente alerta para informar al usuario que el libro se ha eliminado correctamente.



4.1.2 Implementación

Para evitar que este punto se vuelva demasiado tedioso y largo, se va a explicar algunas clases y sobre todo aquello que se considera más importante. Por ejemplo todo el apartado de Ebook no se va a explicar ya que es muy similar a los libros Físico, de hecho comparten muchos métodos los cuales diferencian su forma de funcionar dependiendo en el modo en el que estemos (libros Físicos o Ebooks).



Clase Index:

```
public final class Index extends javax.swing.JFrame {  
  
    FondoPanel fondoMenu = new FondoPanel(0);  
    public static Bd baseDatos = new Bd();  
  
}    public Index(){  
  
        this.setContentPane(fondoMenu);  
        initComponents();  
  
        Menu menu_principal = new Menu(this);  
        menu_principal.setSize(1366,789);  
        contenedor.add(menu_principal);  
  
    }  
}
```

Esta clase es un JFrame donde se va a arrancar todo el programa, contiene un JPanel donde se añadirá la clase Menú siendo esta el corazón de la aplicación. También se inicializa la clase Bd que simula una base de datos donde todas las clases pueden acceder a ella.

Clases FondoBoton y FondoPanel:

He creado estas clases que me permiten crear paneles y botones personalizados, de esta forma podía añadirle a un panel una imagen de fondo o a un botón. Se pueden añadir todas las imágenes nueva que queramos, simplemente creamos un nuevo case y cambiamos la imagen que queremos añadir.

```
//clase que me permite crear botones personalizados  
class FondoBoton extends javax.swing.JButton{  
    private Image imagen;  
    private int _valor=0;  
  
    public FondoBoton(int valor) {  
        _valor=valor;  
    }  
  
    @Override  
    public void paint(Graphics g){  
  
        switch (_valor) {  
            case 0:  
                imagen = new ImageIcon(getClass().getResource("/imagenes/libro.png")).getImage();  
                g.drawImage(imagen, 0, 0, getWidth(), getHeight(), this);  
                setContentAreaFilled(false);  
                setOpaque(false);  
                setBorder(null);  
                super.paint(g);  
                break;  
            case 1:  
                imagen = new ImageIcon(getClass().getResource("/imagenes/ebook.png")).getImage();  
                g.drawImage(imagen, 0, 0, getWidth(), getHeight(), this);  
                setContentAreaFilled(false);  
                setOpaque(false);  
                setBorder(null);  
                super.paint(g);  
                break;  
        }  
    }  
}
```



```
//clase que me permite crear paneles personalizados
class FondoPanel extends javax.swing.JPanel{
    private Image imagen;
    private int _valor=0;

    public FondoPanel(int valor) {
        _valor=valor;
    }

    @Override
    public void paint(Graphics g){
        switch (_valor) {
            case 0:
                imagen = new ImageIcon(getClass().getResource("/imagenes/fondo_menu2.jpg")).getImage();
                g.drawImage(imagen, 0, 0, getWidth(), getHeight(), this);
                setOpaque(false);
                super.paint(g);
                break;
            case 1:
                imagen = new ImageIcon(getClass().getResource("/imagenes/busquedaalibrofisico.png")).getImage();
                g.drawImage(imagen, 0, 0, getWidth(), getHeight(), this);
                setOpaque(false);
                super.paint(g);
                break;
        }
    }
}
```

Clase Libro:

Es una clase sencilla donde hay getter y setter que simulan la información que tiene un libro de verdad.

Clase Bd:

Esta clase contiene toda la información de los libros almacenados y contiene los métodos que permiten realizar los cambios sobre los mismos (se tiene en cuenta que el funcionamiento de estos métodos varían según si se trabajan con libros físicos o Ebooks). Cabe destacar que hay dos vectores diferentes, uno para libros Físicos y otro para Ebooks.

```
//clase que simula una base de datos
public class Bd {

    public static ArrayList<Libro> v_librosFisicos;
    public static ArrayList<Libro> v_librosDigitales;

    public Bd() {...56 lines }

    //métodos de la clase
    public static ArrayList<Libro> getV_librosFisicos() {...3 lines }
    public static ArrayList<Libro> getV_librosDigitales() {...3 lines }

    //se devuelve un libro segun la id que tenga y la bd que pertenezca
    public static Libro libro_For_ID(int id ,int bd){...9 lines }

    //nos devuelve el ultimo id de la bd que pidamos
    public static int getUltimoId(int mode){...8 lines }

    //inserta un libro en la bd que queramos
    public static void insertarLibro(Libro libro, int mode){...7 lines }

    //modifica un libro con la id que hayamos pasado en la bd que queramos
    public static void ModificarLibro(int id, Libro libro, int mode){...7 lines }

    //borra el libro de la id que le pasemos en la bd que digamos
    public static void borrarLibro(int id, int mode){...17 lines }

    //busca un libro en la bd que digamos y en el modo que seleccionemos (nombre, autor, id)
    public static ArrayList<Libro> buscarlibro(int mode, String comparacion, int bd){...56 lines }
}
```



Clase Menú:

Es el núcleo de todo el programa, en esta clase se encuentran todos los paneles que vamos a utilizar, de tal forma que, haciendo visibles los que queremos y ocultando los restantes, damos la sensación de que estamos cambiando de página. Como añadir todo el código sería un problema (y aun así hay demasiado código y es difícil de controlar), se han creado clases de apoyo que son invocadas a la hora de crear una página. Un ejemplo de esto es la función ver libro, cuando se quiere ver un libro se vuelve visible el panel correspondiente y se le añade a ese panel la clase verLibro(hay dos clases, dependiendo de si es LibroFísico o Ebook), de esta forma se hace algo más viable controlar todo el programa.

Caben destacar algunas funciones interesantes que se han creado como pueden ser las siguientes:

RealizarBusqueda:

Esta función es bastante complicada, en resumen, permite realizar una búsqueda tanto de libros físicos como ebooks, en el caso de que no se encuentren libros nos avisa y en el caso contrario nos los muestra. Para que nos los muestre, como no sabemos en principio cuantos serán, se realiza de manera dinámica. Para poder realizar todo esto se ha creado la clase Componente_lbr que es un componente de un botón que tiene el nombre, autor y lugar de un libro que se le pase, entonces estos componentes se van añadiendo de forma dinámica en un Jpanel que lo permite (la clase Clase_contenedor) de tal forma que creamos un vector de Clase_contenedores dinámicos donde vamos añadiendo esto contenedores una vez estos se llenan. Una vez realizado todo lo anterior se muestra por pantalla y se permite navegar por los paneles de nuestro vector de Clases Clase_contenedor, gracias a la función deshabilitar_botones evitamos problemas como salirse del vector.

```
for(int i=0;i<v_librosEncontrados.size();i++){
    //si nos hemos pasado del tamaño del tamaño permitido se guarda, se crea uno nuevo
    if(posicion>360){
        contenedor_libros_aux.setSize(1090, 403);
        contenedor_libros_aux.setLocation(0,0);
        contenedor_libros_aux.setVisible(false);
        v_contenedores.add(contenedor_libros_aux);
        contenedor_libros_aux = new Clase_contenedor();
        posicion = 0;
        i--;
    }else{
        libro_encontrado = new organized.books.Componente_lbr(v_librosEncontrados.get(i));
        libro_encontrado.setSize(1090, 40);
        libro_encontrado.setLocation(0,posicion);
        AccionVerLibro eventoVerLibro = new AccionVerLibro();
        libro_encontrado.btn_NombreAutor_libro().addActionListener(eventoVerLibro);
        contenedor_libros_aux.nuevo_componente(libro_encontrado);

        posicion = posicion + 30;
    }
}
```



```
//funcion que habilita y deshabilita los botones
private void deshabilitar_botones() {
    try{
        if(v_contenedores.size()<=1){
            btn_siguiente.setEnabled(false);
            btn_atras.setEnabled(false);
        }else if(v_contenedores.get(0).isVisible()){
            btn_siguiente.setEnabled(true);
            btn_atras.setEnabled(false);
        }else if(v_contenedores.get(v_contenedores.size()-1).isVisible()){
            btn_siguiente.setEnabled(false);
            btn_atras.setEnabled(true);
        }else{
            btn_siguiente.setEnabled(true);
            btn_atras.setEnabled(true);
        }
    }catch(Exception e){}
}
```

Cabe destacar antes de hablar de las funciones de evento que hay un montón de funciones que mantienen el control de lo que se ve y lo que no se ve, esto se controla con las activaciones de los botones. En la mayoría de casos ha sido posible pero en botones que se encuentran dentro de las clases auxiliares no, o incluso en el caso del componente Componente_lbr era necesario acceder al botón y al ser dinámico era imposible, por ello se han creado varias funciones que se activan con eventos, posteriormente explicare una de ellas, el resto son similares, la diferencia se encuentra con la función que se aplica al activar el botón. Algunos de estos botones tienen doble funcionamiento y esto se puede permitir gracias a que tenemos una variable modoFísico_o_Ebook que nos permite saber si estamos en modo libro físico o modo ebook. Un ejemplo es el siguiente:

```
//boton que permite volver de la busqueda de libros al menu
private void boton_volver_menu(MouseEvent evt) {
    busquedaLibro.setVisible(false);
    menu1.setVisible(true);
}
```

Respecto a las funciones de evento creadas voy a explicar por ejemplo la función AccionVerLibro ya que el resto se basan en el mismo concepto.

En esta clase se comprueba si se activa el botón ver libro que se encuentra en el Componente_lbr, para ello, se recorren todos los contenedores con todos los libros para ver si la acción recibida coincide con algún ver libro, en el caso de que coincida vemos la id del libro para así poder enviarle este id a la clase VerLibro (funciona para los dos métodos), de esta forma cuando se cree la clase, se añade al panel del menú el contenedorVerLibro en el caso de libro físico, o contenedorVerLibroe en el caso de libro Ebook. Por último, se vuelve visible este panel y se vuelve invisible busquedaLibro.



```
//aquí se han creado varias clases, estas clases se han utilizado cuando queríamos controlar algún boton que se encontraba
//interno en las clases de apoyo y era necesario ver si se activaban para realizar cambios a los paneles en el menu
//por ello se han creado estas clases que comprueban si se activa un evento y si este coincide con el que busco

//cabe destacar que cada método creado aquí es necesario para que funcione añadirlo al botón donde queremos que se aplique

//comprueba si se activa el boton ver libro que se encuentra en el Componente_lbr
public class AccionVerLibro implements ActionListener
{
    @Override
    public void actionPerformed(ActionEvent ae){
        try{
            //se recorren todos los contenedores con todos los libros para ver si la acción recibida coincide con algún ver libro
            for(int i=0; i<v_contenedores.size();i++){
                for(int j=0; j<v_contenedores.get(i).getV_botoneslibros().size();j++){
                    if(ae.getSource().equals(v_contenedores.get(i).getV_botoneslibros().get(j))){
                        //en el caso de que coincida vemos la id del libro para así poder enviarle este id a la clase VerLibro (funciona para los dos metodos)
                        //de esta forma cuando se cree la clase se añade al panel del menu contenedorVerLibro en el caso de libro físico
                        //o contenedorVerLibro en el caso de libro ebook. Por último se vuelve visible y se vuelve invisible busquedaLibro
                        int id=v_contenedores.get(i).getV_librosId().get(j);

                        busquedaLibro.setVisible(false);
                        contenedorVerLibro.removeAll();
                        contenedorVerLibro.validate();

                        if(modosFisico_o_Ebook==0){
                            auxVerLibroFisico = new VerLibroFisico(id);
                            auxVerLibroFisico.setSize(1090, 500);
                            auxVerLibroFisico.setLocation(0,0);

                            Ir_a_ModificarLibro eventoIr_a_ModificarLibro = new Ir_a_ModificarLibro();
                            auxVerLibroFisico.getBoton_modificar_librof().addActionListener(eventoIr_a_ModificarLibro);

                            contenedorVerLibro.removeAll();
                            contenedorVerLibro.validate();
                            contenedorVerLibro.add(auxVerLibroFisico);
                            contenedorVerLibro.setVisible(true);
                            verLibroFisico.setVisible(true);
                        }else{
                            auxVerLibroEbook = new VerLibroEbook(id);
                            auxVerLibroEbook.setSize(1090, 500);
                            auxVerLibroEbook.setLocation(0,0);

                            Ir_a_ModificarLibro eventoIr_a_ModificarLibro = new Ir_a_ModificarLibro();
                            auxVerLibroEbook.getBoton_modificar_libroe().addActionListener(eventoIr_a_ModificarLibro);

                            contenedorVerLibro.removeAll();
                            contenedorVerLibro.validate();
                            contenedorVerLibro.add(auxVerLibroEbook);
                            contenedorVerLibro.setVisible(true);
                            verLibroEbook.setVisible(true);
                        }

                        //una vez encontrado no tiene sentido seguir con el bucle
                        i=v_contenedores.size();
                    }
                }
            }
        }catch(Exception e){}
    }
}
```

Alertas:

Se han creado varias alertas para informar al usuario e incluso una de ellas se ha usado para advertir al usuario de borrar un libro. Todas ellas son muy similares por ello voy a explicar una solo, aunque en total son Acceptar, AlertaBorrarLibro, AlertaLibroBorrado y Error.

En la clase Acceptar nos encontramos con lo siguiente:



```
//alerta que avisa que un libro se acepto correctamente(modificar o crear)
public class Aceptar extends javax.swing.JDialog {

    /**
     * Creates new form Aceptar2
     * @param parent
     * @param modal
     */
    public Aceptar(java.awt.Frame parent, boolean modal) {
        super(parent, modal);
        initComponents();

        this.setLocationRelativeTo(this);
        this.setResizable(false);
    }
}
```

El diseño es muy sencillito y no tiene nada que destacar pero en este caso cabe destacar que es necesario pasar el JFrame del padre para evitar que el usuario no cierre la ventana, al realizar esto evitamos posibles problemas.

También se ha hecho que la alerta sea de tamaño fijo y que esta aparezca en medio de la pantalla.

VerLibroFisico:

Es una clase sencilla donde se aloja la interfaz para ver un libro, esta clase se creará a partir de un id y se añadirá al menú en su panel correspondiente.

```
//clase para ver un Libro Fisico
public class VerLibroFisico extends javax.swing.JPanel {

    /**
     * Creates new form VerLibroFisico
     */

    Libro libro_elegido = new Libro();
    int _id;

    public VerLibroFisico() {
        initComponents();
    }

    public VerLibroFisico(int id) {
        initComponents();

        Libro libro_para_ver = Bd.libro_Por_ID(id, 0);
        campo_autor.setText(libro_para_ver.getAutor());
        campo_lugar.setText(libro_para_ver.getLugar());
        campo_nombre.setText(libro_para_ver.getNombre_libro());
        campo_resumen.setText(libro_para_ver.getResumen());
        campo_nota.setText(String.valueOf(libro_para_ver.getOpinion()));
        _id=id;
    }

    public JButton getBoton_modificar_librof() {
        return boton_modificar_librof;
    }

    public int getId() {
        return _id;
    }
}
```



ModificarLibroFisico:

Es una clase sencilla donde se aloja la interfaz para modificar un libro, esta clase se creará a partir de un id y se añadirá al menú en su panel correspondiente. Cabe destacar que se usa el componente ComponenteNota.

```
ComponenteNota comp_nota = new ComponenteNota();
int _id;

public ModificarLibroFisico() {
    initComponents();
}

public ModificarLibroFisico(int id) {
    initComponents();

    Libro libro_para_ver = Bd.libro_Por_ID(id, 0);
    campo_autor.setText(libro_para_ver.getAutor());
    campo_lugar.setText(libro_para_ver.getLugar());
    campo_nombre.setText(libro_para_ver.getNombre_libro());
    campo_resumen.setText(libro_para_ver.getResumen());

    comp_nota = new ComponenteNota(libro_para_ver.getOpinion());
    comp_nota.setSize(242, 171);
    comp_nota.setLocation(159, 285);
    contenedor.add(comp_nota);
    contenedor.setVisible(true);
    _id=id;
}

public JButton getBtn_aceptaModificacionLibro() {
    return btn_aceptaModificacionLibro;
}

public JButton getBtn_BorrarLibro() {
    return btn_BorrarLibro;
}

public boolean hayError() {
    return campo_autor.getText().isEmpty() || campo_lugar.getText().isEmpty() || campo_nombre.getText().isEmpty() || campo_resumen.getText().isEmpty();
}

public void ModificarLibro() {
    Libro libro_elegido = new Libro(_id, campo_nombre.getText(), campo_autor.getText(), campo_resumen.getText(), campo_lugar.getText(), comp_nota.getNota());
    Bd.ModificarLibro(_id, libro_elegido, 0);
}

public int getId() {
    return _id;
}
```

NuevoLibroFisico:

Es una clase sencilla donde se aloja la interfaz para añadir un nuevo libro, esta clase se creará y se añadirá al menú en su panel correspondiente. Cabe destacar que se usa el componente ComponenteNota.



```
//clase que contiene la interfaz para añadir un libro Fisico
public class NuevoLibroFisico extends javax.swing.JPanel {

    /**
     * Creates new form VerLibroFisico
     */

    int id;
    ComponenteNota comp_nota = new ComponenteNota();

    public NuevoLibroFisico() {
        initComponents();

        id=Bd.getUltimoId(0);
        id++;
        campo_autor.setText("");
        campo_lugar.setText("");
        campo_nombre.setText("");
        campo_resumen.setText("");
        comp_nota.setSize(242, 171);
        comp_nota.setLocation(159,285);
        contenedor.add(comp_nota);
        contenedor.setVisible(true);
    }

    public JButton getBtn_aceptaNuevoLibro() {
        return btn_aceptaNuevoLibro;
    }

    public boolean hayError(){
        return campo_autor.getText().isEmpty()||campo_lugar.getText().isEmpty()||campo_nombre.getText().isEmpty()||campo_resumen.getText().isEmpty();
    }

    public void insertarLibro(){
        Libro libro_elegido = new Libro(id, campo_nombre.getText(), campo_autor.getText(), campo_resumen.getText(), campo_lugar.getText(), comp_nota.getNota());
        Bd.insertarLibro(libro_elegido, 0);
    }

    public int getId() {
        return id;
    }
}
```

ComponenteNota:

Es un componente sencillo donde según la nota que se ponga se ven estrellas, de tal forma que si le damos un uno de nota se verá una sola estrella encendida. En necesario poder extraer la nota final para poder añadirla al vector de libros.

```
//componente para modificar la nota de un libro con un valor del 0 al
public class ComponenteNota extends javax.swing.JPanel {

    /**
     * Creates new form ComponenteNota
     */

    int nota;

    public ComponenteNota() {
        initComponents();
        nota=0;
        this.dibujarNota();
    }

    public ComponenteNota(int valor) {
        initComponents();
        nota=valor;
        this.dibujarNota();
    }

    private void dibujarNota() { ...55 lines }

    public int getNota() {
        return nota;
    }
}
```



4.1.3 Evaluación

Tras realizar la implementación, se ha buscado obtener **feedback** que nos permita determinar los puntos débiles del sistema y así poder corregirlos, para ello se les va a realizar un test a varios usuarios para ver su punto de vista sobre la aplicación.

A los usuarios se les pedirá que busquen el libro **Harry Potter y la piedra Filosofal**, añadan un libro, que busquen de nuevo ese libro, lo modifiquen, lo vean y por último lo borren. Se les dará un tiempo extra para que exploren lo que deseen e incluso que sean capaz de bloquear la aplicación si es que pueden.

Posteriormente, se les dará el test y una vez tengamos todos, se extraerá una conclusión final.

El test será el siguiente:

		1	2	3	4	5	NS/NC
1	Por lo general, estoy satisfecho con la facilidad de uso.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	Pude llevar a cabo la tarea propuestas de forma efectiva.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	Pude completar la tarea propuesta de forma rápida.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	La información proporcionada por la aplicación es fácil de interpretar.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	La organización de la información en las pantallas fue clara.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	La interfaz de la aplicación es elegante.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	La interfaz de la aplicación es amigable.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8	Me gusta usar la interfaz de la aplicación.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9	La aplicación cubre las funciones que esperaba de ella.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10	Si me equivoco navegando puedo regresar de forma sencilla y rápida.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
11	Me siento cómodo utilizando la aplicación.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
12	Prefiero utilizar la aplicación a las páginas web convencionales.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
13	Utilizaría esta aplicación en caso de estar disponible.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
14	Por lo general, estoy satisfecho con la aplicación.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
15	Comentarios adicionales sobre el uso de la aplicación						



Usuario: Alejandro García García (un amigo)

1	2	3	4	5	6	7
5	5	4	4	5	5	4
8	9	10	11	12	13	14
4	5	5	5	NS/NC	5	5

15->

- Al pulsar la tecla "TAB", pasa al siguiente apartado de forma lateral, es decir, no lo hace de una forma normal, como sería siguiendo el sentido de rellenado de campo (Título, nombre del autor, lugar y resumen).
- Estaría bien que se pudiera hacer "click" directamente en las estrellas para valorar un libro, por consiguiente, a la hora de valorarlo estas fuesen en dirección de las agujas del reloj, en lugar de ir en ascenso.
- El apartado de borrar cuesta un poco encontrarlo, pero con un manejo más o menos adecuado, es fácil de localizar.
- Sería necesario implementar una búsqueda avanzada, que permita buscar los libros sin la necesidad de introducir el nombre de este o del autor al completo.

Usuario: M^a del Pilar Gómez Giménez (mi hermana)

1	2	3	4	5	6	7
4	5	5	5	4	5	5
8	9	10	11	12	13	14
5	5	5	5	5	5	0

15->

- Sería muy positivo que se incluyera la búsqueda por lugares, dentro de los libros físicos. De ese modo se podrían consultar todos los libros que se guardan en una aplicación.
- Se podría incluir la búsqueda de libros según palabras clave del resumen.
- Asimismo, la búsqueda podría ser más intuitiva. Que no sea necesario poner en el buscador el nombre completo del libro o del autor.

*Apunte mío, ha encontrado un bug en el funcionamiento de la aplicación, se solucionará cuanto antes en las versiones futuras.



Usuario: M^a del Pilar Giménez Gómez (mi madre)

1	2	3	4	5	6	7
5	5	5	5	5	5	5
8	9	10	11	12	13	14
5	3	5	5	NS/NC	5	5

15->

- Me gustaría no tener que poner el titulo completo con mayúsculas incluidas
- La barra para empezar a escribir no debería estar pegada al extremo

En resumen, tras ver las evaluaciones, la verdad, no me esperaba que gustara tanto ya que los errores y donde más se han quejado es en como funciona la búsqueda, cosa que tiene sentido, me alegra que la interfaz haya sido un éxito.

Cabe destacar que me ha sorprendido el hecho de encontrar un fallo en la aplicación y sobre todo mi madre ha encontrado una posible mejora en la interfaz separando la barra para escribir del recuadro ya que no se daba cuenta que sí se podía escribir.

Los tres me han sugerido ideas y posibles mejoras, principalmente sobre la funcionalidad de la aplicación ya sea, búsqueda por lugar, búsqueda por nota o incluso poder dividir los libros por secciones.

En conclusión, la interfaz ha sido un éxito (con posibles mejoras) y la funcionalidad de la aplicación tiene mucho que mejorar.

4.2 Iteración 2:

Por falta de tiempo no se ha realizado una segunda iteración, pero tras lo visto en la evaluación anterior, en el caso de que se hiciera en el futuro, se aplicarían las ideas de mejoras realizadas en la evaluación anterior y aparte se intentaría implementar Internacionalización en el problema. Por último, se debería cambiar la simulación de la base de datos (el vector de datos), por una base de datos de verdad para que así se apliquen los cambios realizados sobre el vector (que se guarden las modificaciones en los libros, que se guarden los nuevos libros o incluso que se borre un libro del todo).