

Raúl A. Torres Cobián  
Y00566589

20 de mayo de 2020  
Coen 2220  
Profesor Jaime Yeckler

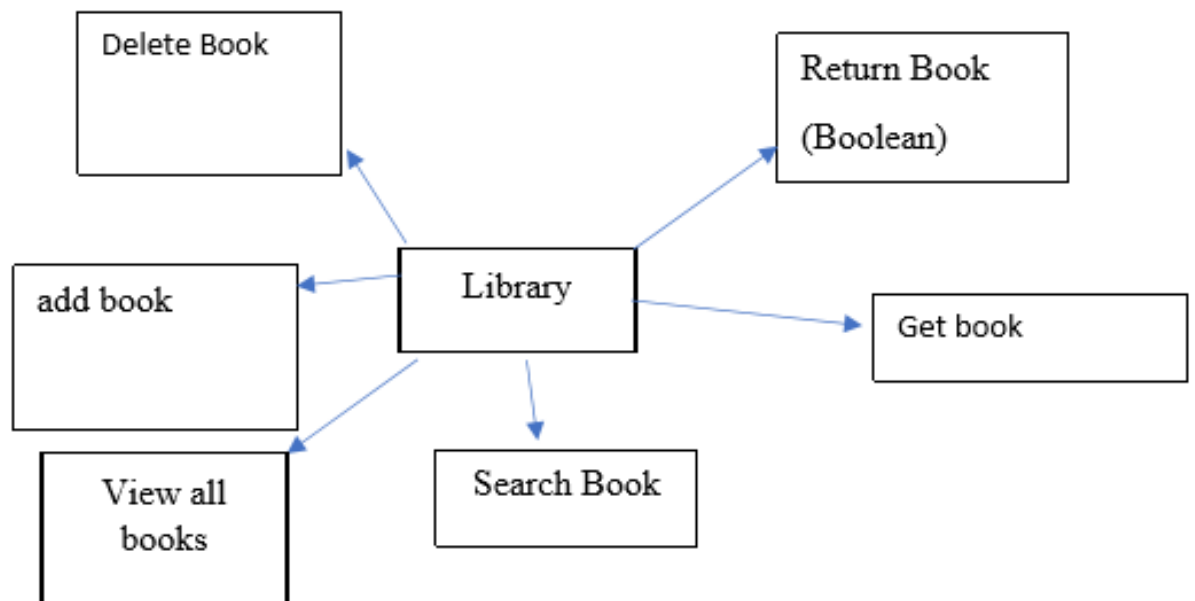
## **Informe final de manejo de libros alquiler**

### **Propuestas Simple sistema de gestión de alquiler de libros**

#### 1. Descripción de proyecto

Este proyecto final que elegí se llama Simple rent book management system (simple sistema de gestión de alquiler de libros). Lo que hace es una tienda de Liberia alquilan libros de cualquier género incluido libros para uso de clase por no tiene que comprar un libro más caro. En mi proyecto se va a basar de registrar libro con la cuenta de la persona o universidad que tiene fecha de entrega de 2 semana para personas general y 4 semanas para universidad. También incluye cálculos de precio y cantidad de libro que vas a prestar. Incluye almacén de libros capacidad, Y si el libro estas cogido.

## 2. Diagrama de bloques (entrada y salidas detalladas)



## 3. ¿Por qué eliges el tema?

Lo elegí un tema porque estoy pensando montar un negocio de libros porque algunos de libros de compras valen caro o pdf piratear causa que los dueños de libros pierden millones de dólares hasta que termina quebrado, decidí montar un negocio de libro de alquiler.

4. Si su proyecto no es original, describa el origen de la idea y su contribución.

Mi proyecto final no es original por que busque de internet de Youtube sobre proyectos de java ideas, hay muchas ideas que quieres de escoger vi uno que dice simple book managment y decidí coger ese tema de ventas de libro busque como era tiendas de libro como amazon, ebay etc. Me atrae la atención. Este diagrama hice yo con requisitos de la OOP.

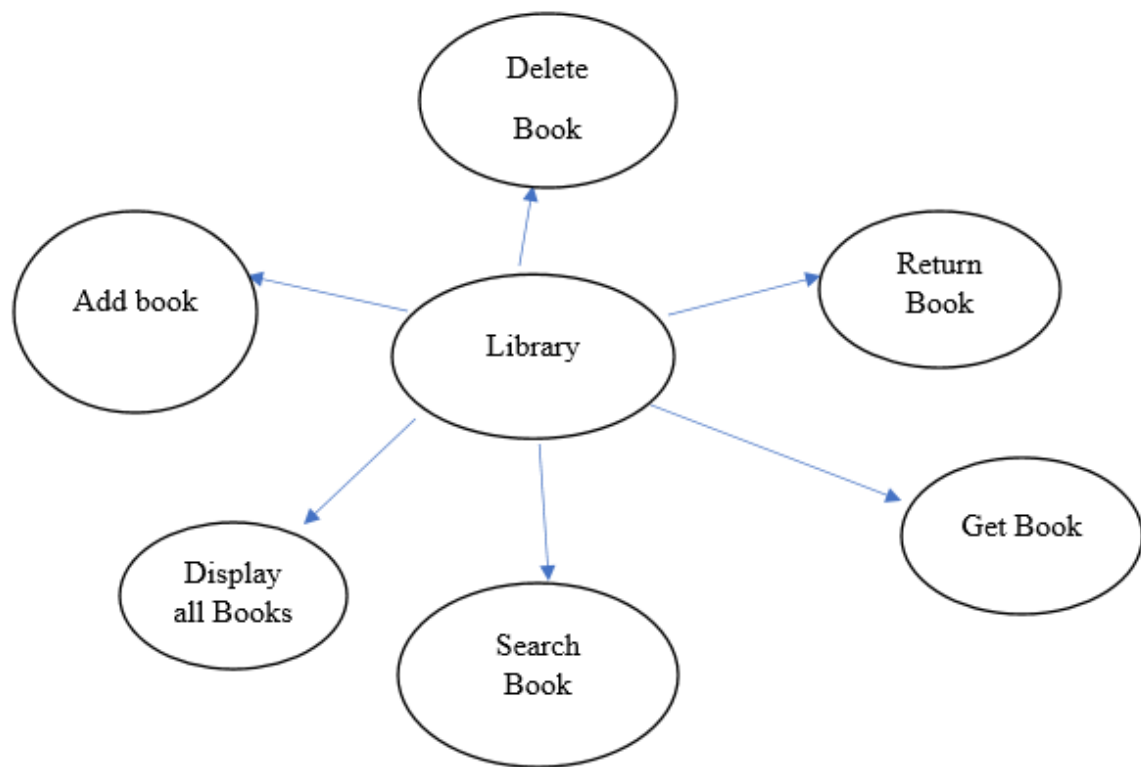
5. Referencias (nombre y dirección de Internet)

1. <https://www.youtube.com/watch?v=7XIduOKTaM0&t=32s> (java projects ideas for beginners and engineering students youtube).
2. <https://projectsgeek.com/2015/11/college-book-management-system-java-project.html> (Projects geek).

**Fase 1: Casos de uso y menú del programa**

1. Especificaciones del programa

A) Entrada y salida del programa (Diagrama)



## B) Menú de la aplicación

Login:
Username:
Password:

Menu para usuario
<ol style="list-style-type: none"><li>1. Ver todo en el libro</li><li>2. Buscar libro</li><li>3. Coger libros</li><li>4. Regresar libros</li><li>0. salir</li></ol>

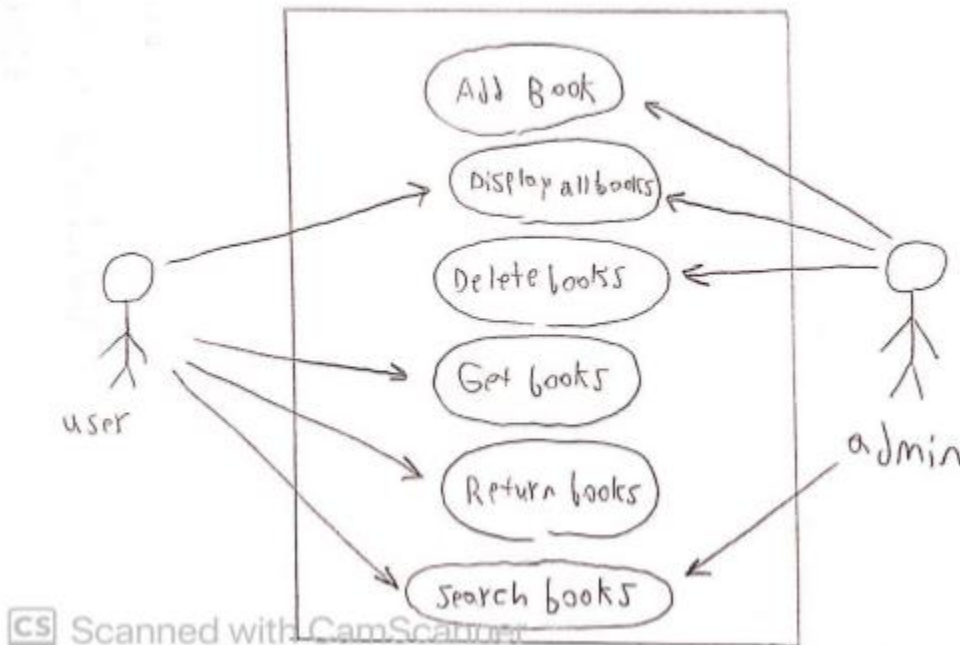
Menu para administrador
<ol style="list-style-type: none"> <li>1. Anadir libros</li> <li>2. Ver todo los libros</li> <li>3. Borrar libros</li> <li>4. Buscar libros</li> <li>0. salir</li> </ol>

### C) Tareas de aplicación va a desarrollar

En esta tarea de aplicación que van a desarrollar es buscar un libro que vas a rentar y registrar la cuenta que va a llevar incluido el registro tiempo límite de la entrega. Esto incluye el estatus de la persona que tiene fuera del libro con cuanto ha llevado.

### 2. Análisis de caso de uso

#### A) Lista de caso de uso



## B) descripción de cada caso de uso

Display all books	Search Book
Input: ninguno	Input: escribe el título de libro quien estas buscando
Output: Mencionar todo el libro incluido disponibilidad	Output: Mencionar el libro que buscas
Tareas: <ol style="list-style-type: none"><li>1. Mencionar un libro que vas a seleccionar.</li><li>2. nombres de libros</li><li>3. Salir la Pantalla de almacén de libro</li></ol>	Tareas: <ol style="list-style-type: none"><li>1. Escribir un libro que quieres buscar tiene que ser el nombre de libro en donde está en la lista de libro.</li><li>2. Hacer una condición si el libro está disponible</li><li>3. Si no encuentra del libro lo que buscar, seleccionar las listas del libro.</li></ol>
Return book	
Input: escribe un libro que vas a entregar	
Output: El libro han entregado	
Tareas: <ol style="list-style-type: none"><li>1. La persona va a entregar el libro de unos cuantos.</li><li>2. Registrar el libro que vas a entregar.</li><li>3. Hacer condición de libro que estas entregado.</li></ol>	

Add libros
<p>Input:</p> <p>Añadir titulo</p> <p>Añadir autor</p> <p>Añadir isbn</p> <p>Añadir cantidad</p> <p>Añadir Precio</p> <p>Añadir Fecha de publicidad</p>
Output: Salir pantalla del título, autor, precio, cantidad, fecha publicada y disponibles.
<p>Tareas:</p> <ol style="list-style-type: none"> <li>1. Adicionar el libro que vas a llevar con el nombre con Autor.</li> <li>2. ingresar el precio de libro cuanto llevas y calcular el precio total.</li> <li>3. Registrar la persona que vas a llevar.</li> <li>4. Guardar un nombre de elemento de libro.</li> </ol>

Delete Book
Input: escribir el nombre de libro que vas a borrar
Output: borrar aceptada
<p>Tareas:</p> <ol style="list-style-type: none"> <li>1. Escoger el título de libro que vas a borrar en la lista.</li> <li>2. Hacer una condición de libro que quieres borrar.</li> <li>3. Borrar echo.</li> </ol>

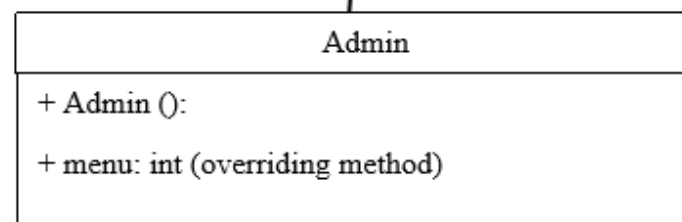
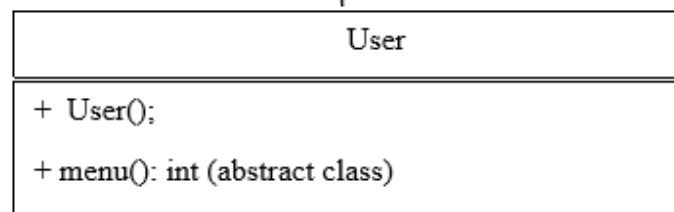
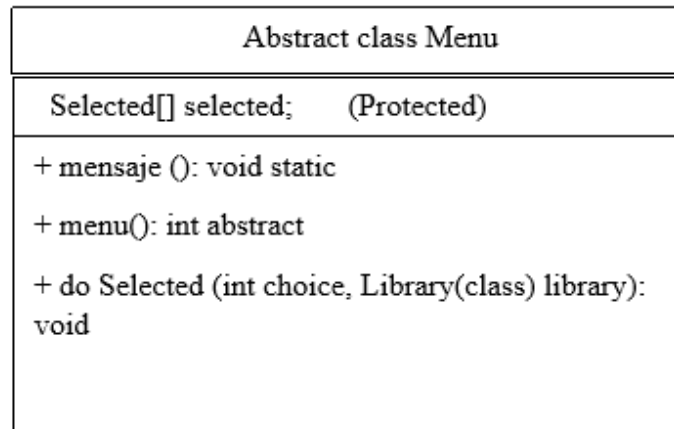
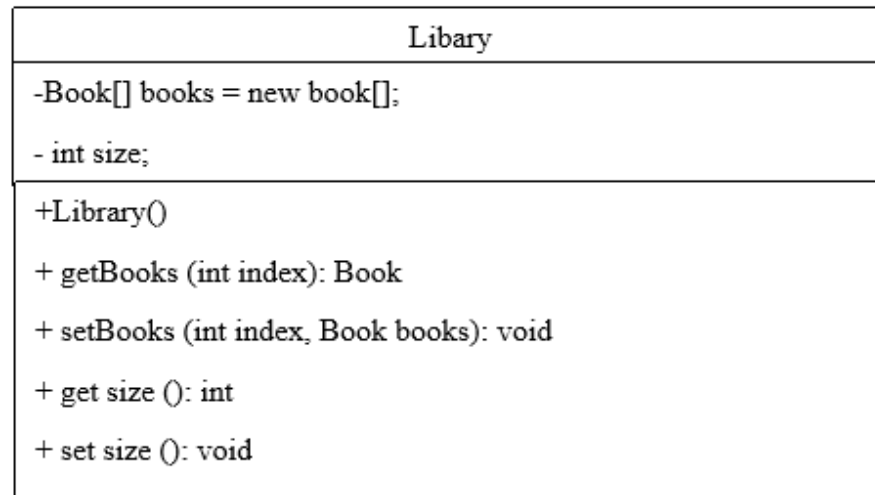
Get Books
Input: Escribir el nombre que vas a escoger
Output: Libro escogido
<p>Tareas:</p> <ol style="list-style-type: none"> <li>1. Escribir un título de libro que vas a decidir</li> <li>2. Hacer un loop de índice para especificar el libro que escoges</li> <li>3. Crear una condición de if y else si el libro esta disponible o no disponible o libro que ya está escogido.</li> </ol>

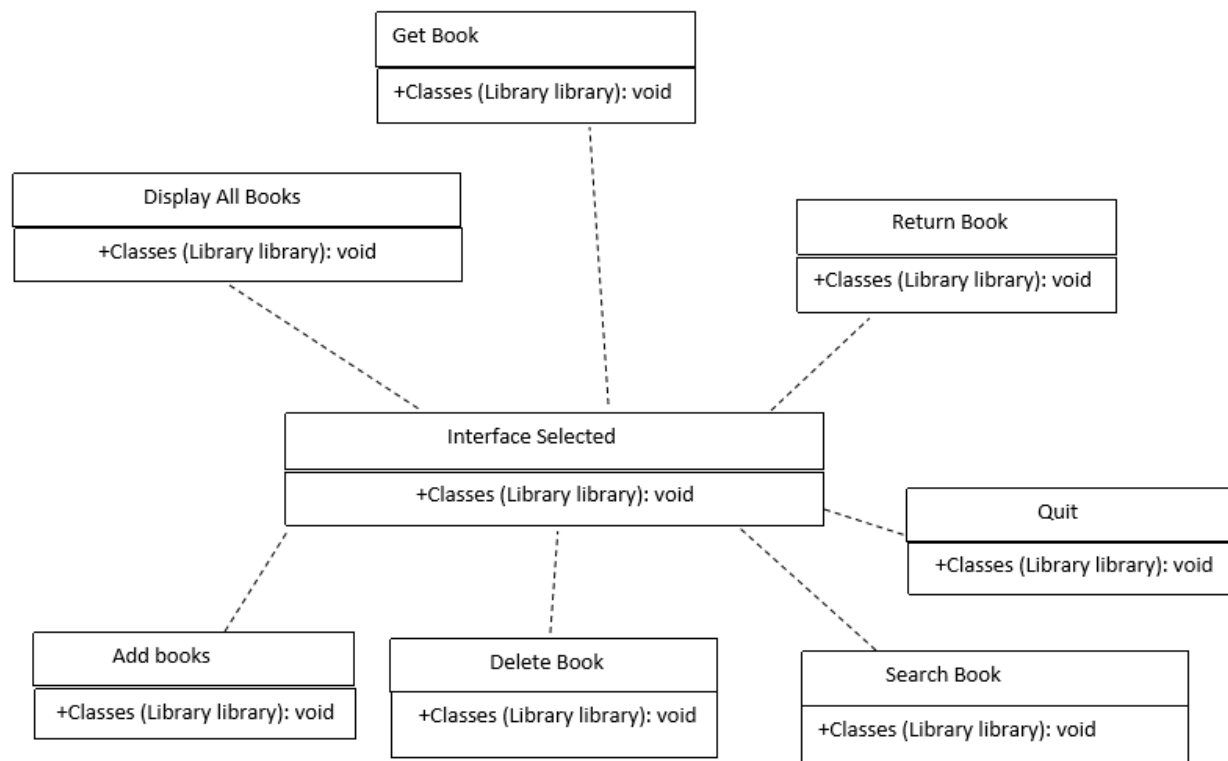
## Fase 2: Diseño clase y objetos

Book
-String title; -String author; -int isbn; -double price; -double quantity; -String release date; -Boolean borrows;
+ Book () +Book (string name, string author, double price, int isbn, double quantity, String release date) +get Title (): String +set Title (): void +get Author (): String +set Author (): void +get Price (): double + set Price (): void +get ISBN (): Int +set ISBN (): void +get borrow (): Boolean +set borrow (): void + subtotal (): double + Total price (): double + to String (): String

---







a)

### **Book class:**

- Declarar variables globales privados del string, números enteros y booleanos.
- Los métodos usan públicos de constructor de título en default con otro método de libro con parámetros.
- Los métodos incluyen el captado y colocarlo declara las variables. Lleva en enteros, void y booleana.
- Hay dos métodos incluye el total de precios calculados.
- To String va todo declarado las variables globales para que sobre escriba el almacén de libros.

### **Library class:**

- Las librerías primero declarar variables globales en privado de enteros de medidas y objetos de la capacidad del libro.
- El constructor de métodos de la Liberia incluye objetos de libros de cada uno con variable de medidas.
- Los 3 métodos con parámetros de captado y colocarlo de captar libro incluye el arreglo para usar los elementos de índices de cada línea. Y colocarlo las medidas usa medidas sumando de contar para que añada el libro con la interface que se llama añadir libros.

### **Class abstract Menu**

- Las clases abstracta usa el import scanner para seleccionar el menú driven usando global protegida de interface llamado Seleccionar con arreglo.
- Los métodos estáticos públicos creo mensaje y escribir lo que quieras.
- Otro método de menú minúscula usa números entero en abstracto para seleccionar el menú.
- El ultimo método incluyen parámetros de escoger entero con clase Liberia y dentro de método va a seleccionar con escoger en arreglo y clases método dentro de paréntesis Liberia que va en interface.

### **Class User (extend Menu)**

- Esta clase va a heredar el Menu clases para usar el menú driven.

- El constructor de seleccionar menú usa el este seleccionar con arreglo para usar el índice de menú. El primero va a salir el programa que empieza cero por parte de programa de clase usando nuevo.
- El ultimo método usa abstracto implementar el menú, dentro de menú escribe las listas de menú para el usuario. Con entero de decidir y regresar método de decidir.

### **Class Admin (extend User)**

- Esta clase de administrador va a heredar el usuario para contar el menú con el método de sobre escritura.
- Lo mismo de constructor como user excepto los menus esta diferente de usuario por que el administrador son los que encarga la Liberia.

### **Interface Selected**

- La interface de seleccionar usar otro input para escoger el abstracto y seleccionar lista de menú como añadir, borrar, buscar, coger, entregar el libro.
- Hay un método solamente que entre las clases void incluyendo los parámetros Liberia por el captador y colocarlo.

### **Class Add Book (implements)**

- En esta clase iba a implementar de decidir para escoger el menú de un arreglo.

- El método implementa los mismo de las clases con parámetros de Liberia.
- Dentro de un método iba la global de números enteros de medidas y crear un libro usando de input cada uno (titulo, autor, isbn, precio, cantidad y fecha de publicidad).
- Después de input iba un objeto de libro con paréntesis de objetos para añadir el almacén de libro. Colocar libro en paréntesis pone la medida y el libro y colocarlo el size poner numero 1 para que añaden el libro e imprimir los resultados.

### **Class Display all books (implements)**

- Esta clase implementa de decidir para escoger el próximo menú de arreglo.
- Este código solamente implementa el for loop usando, captando la medida e imprimir todos los libros que tiene en el almacén usando, captando libros y dentro de paréntesis coloca el índice.

### **Class Delete Books (implements)**

- Esta clase implementa de decidir para escoger el próximo menú de arreglo.
- Los pasos es escribir un título de libro que vas a borrar.
- El global de variable solamente aclaran el índice fuera para que for loop funcione bien.
- Los loop usa captar la medida para escoger libros almacén específico con break.

- Crear una condición de si escoges el libro bien título Salir la pantalla el libro estas borrado sino el libro no existe o no escribir bien especifico como lleva mayúscula ejemplo: Eragon -> eragon (eso no va a funcionar).

### **Class Get Book (implements)**

- Esta clase implementa de decidir para escoger el próximo menú de arreglo.
- Este código hace escribir el título de libro y captar la medida como borrar el libro.
- El global usa el índice afuera para for loop y usar condición que, si el libro esta cogido usa índice, si el libro está disponible sale el libro has cogido si no el título de libro no está o no escribes con mayúsculas con especifico siempre usa el ver todos los libros para que no pasen el error.

### **Class Return Book (implements)**

- Esta clase implementa de decidir para escoger el próximo menú de arreglo.
- Este código es entregar libro usando el título de libro que esta fuera de la Liberia.
- Usando el for loop de captar la medida del almacen y crear nested loop.
- Uno loop de si el nombre de libro esta correcto imprime de else y cambiar la bandera booleana falsa. El segundo if si el libro no está bien escrito imprime que no existe.

- El else usa captar libro índice con captar escoger es igual a falso el libro no ha regresado significa que la Liberia todavía esta disponibles.

### **Class Search Book (implements)**

- Esta clase implementa de decidir para escoger el próximo menú de arreglo.
- El paso de esta clase hacer un input de título para buscar el libro.
- La variable global usa entero de índice fuera de loop.
- Usar for loop de captar la medida de libro en el almacén.
- Usar nested if como if de captar libro con el índice con captar título y es igual con paréntesis título usar break para escoger un nombre de libro específico.
- Otro if de input validation de captar size sin índice imprime el libro no existe y si el captar la medida incluye el índice imprime el libro que escogiste.

### **Class Test**

- En la prueba incluye el import scanner para usar los usuarios de cuenta
- Primero usar el método principal para usar los dos objetos 1 de user a login y la Liberia
- El loop usar while con la condición cierta para usar loop infinito hasta que seleccione el sistema si quieres salir.
- Dentro de while declara variable de decidir con los objetos de menú a user.

- El ultimo método publico estático es login.
- Crear el scanner para escribir los registros que vas a entrar de nombre usuarios y la contraseña.
- Y el ultimo usar la condición para contraseña Si la contraseña es 0 accede solamente a los usuarios si no lo registra como administrador.

#### **4. Explicación de cómo se está cumpliendo los requisitos del proyecto: Dar respuesta a los ítems (a) al (h) establecido en la sección de Requisitos.**

**a) encapsulación de variables (“Public”, “Private”, “Protected”):** En este caso de variables está cumpliendo parte de privado en el área las clases hay de Book.java y Library.java de global. En el protegido declara solamente en la clase abstracta llamado Selected con arreglo para cada uno que implementa para el menú driven sin usar switch case menú y el público solamente declara decidir en el input para seleccionar un menú de programa y métodos esta todo en el publico y varios métodos incluye con métodos estáticos y no estáticos.

**b) Variable estática y no estática:** variable estática cumplen en globales de información de libro, autor, isbn, precio y cantidad, por otro lado, de la medida (size) y objeto de book con arreglo para almacenar los libros hasta el 20. No estático era de índice fuera de for loop que yo aprendí en los laboratorios anteriores era de decidir de menú y string de título, autor, números en el área de añadir libros.

**c) Método estáticos y no estáticos:** El método estático hay pocos en mi proyecto final y era del test.java que cree para cumplir el requisito de sobre escritura de método (polimorfismo dinámico) y el mensaje en donde está en la clase abstracta de output que dice: “Este es mi proyecto



final.” Y los métodos no estáticos están casi todo en la 14 clase de método de público de método return, void, toString incluido de la construcción de método default sin parámetros en donde dice Book.java, Library.java, User.java y Admin.java.

**d) Herencia:** Las herencias basado mi proyecto final cumplen tres clases de herencia porque mi proyecto es casi toda clase independiente, una clase abstracta de Menu.java hereda al User.java para cumplir de this selected de arreglo para dar objetos de new usar menú driven sin tener que usar nested ni switch cases. Y el user hereda de Admin.java para cumplir de sobre escritura de método para completar todo de object oriented.

**e) Sobre Carga (“Overloading”) de métodos:** En sobre de carga de método (polyformismo estático) hay una clase de primera clase que cree llamando Book.java hay un método constructor default sin parámetros llamado Book que asigna de título string usando entre comillas y los números de entero y double 0. Segundo método de Book con parámetros lo mismo de título, autor etc. Y en corchete va el this method para cumplir método sobre carga que significa dos métodos iguales pero diferentes parámetros, uno que es constructor sin parámetros y otros con parámetros.

**f) Sobre Escritura (“Overriding”) de método:** Hay una sobre escritura de método en mi proyecto que es el administrador extiende al usuario para cumplir los requisitos de

sobre escritura esta localizados en public int menú sin parámetros lo cuadra los dos gracias la herencia de user que cumple los requisitos de método sobre escritura.

**g) Clase Abstracta:** En la clase abstracta hay una clase abstracta llamado Menu.java. en el menú java de global puse el import de scanner y dentro método clase abstracta incluir mensaje estático, menú tiene publico abstracto entero para seleccionar el menú de usuario y administrador y método void doSelected con parámetro de int choice y Library library declarando los objetos y dentro de método tiene selected con arreglo adentro choice. Class de interface porque interface entra clase abstracta porque son los mismo. Dentro de paréntesis classes puse library para escoger el menú. La clase abstracta cumple de User y Admin.

**h) Interface:** En interface hay una y la única interface de mi proyecto final se llama Selected (Seleccionar) hay un método void de interface llamado class con parámetros Library library de mayúscula con minúscula. Incluido de input de Scanner para cumplir clases de selecciones que usa implementar cada uno con añadir libros, borrar, mencionar el almacén de libro y lo incluye salir el sistema que es con system.exit(0).

## **5) Resultados**

- a) Explique las pruebas realizadas y resultados contenidos. Mostrar “screen pictures” de la salida de su programa.

## Menú para el usuario

```
Username: kirito3310
Enter you password: 0
Este es mi proyecto final

Welcome kirito3310 to the Torres Library

For the user Selected Menu
1. View all books
2. Search books
3. Get books
4. Return books
0. Quit
```

Este menú lo que hace es hacer un registro para el cliente solamente entra el nombre de usuario y la contraseña que solamente tiene que entrar el numero cero (0) para acceder el usuario, si no entra 0 registra el administrador. Y ahí lo va a mencionar el menú que vas a seleccionar es usar el this selected de arreglo extender la clase interface de arreglo sin usar el switch case ni nested if que también puede aplicar el menú driven.

## Opción 1: Mencionar todos los libros en el almacén.

```
For the user Selected Menu
1. View all books
2. Search books
3. Get books
4. Return books
0. Quit
4
Enter the title of the book you want to return:
Eragon
Return succeeded!

For the user Selected Menu
1. View all books
2. Search books
3. Get books
4. Return books
0. Quit
1
Book Title: Eragon, author: Crishopher Paolini, price: 50.0, ISBN: 254352, release date: 26/08/2002 Aavailable
Book Title: Sword art online, author: Reki Kawahara, price: 68.0, ISBN: 4023452, release date: 10/04/2009 Aavailable
Book Title: Narnia, author: C.S.LEWIS, price: 16.0, ISBN: 4023452, release date: 16/10/1950 Aavailable
```

La primera opción es el almacén de libro lo que hace es mostrar todos los libros en el almacén en donde se localizan de Library.Java. En objetos de libros registrado con size para mencionar la pantalla. Si el size pone cero no va a aparecer el output del libro

## Opción 2 Buscar libros

```
For the user Selected Menu
1. View all books
2. Search books
3. Get books
4. Return books
0. Quit
2
Enter the title of the book that you want to search
Eragon
Book Title: Eragon,  author: Crishopher Paolini,  price: 50.0,  ISBN: 254352, release date: 26/08/2002 Available

For the user Selected Menu
1. View all books
2. Search books
3. Get books
4. Return books
0. Quit
```

Esta opción 2 lo que hace es buscar primero, hay que asegurar el libro que está escribiendo con la letra mayúscula como esta en el almacén que sea específico. Imprimen toda la información de libro de título, autor, precio, isbn y la fecha de publicación.

## Opción 2: Error de buscar libros

```
For the user Selected Menu
1. View all books
2. Search books
3. Get books
4. Return books
0. Quit
2
Enter the title of the book that you want to search
eragon
|This book does not exist or type specific program
```

Este input validación es que si no escriben específico como están en el libro sin poner mayúscula o poner espacio accidente sale el error que no existen o no escribiste bien específico al programa.

### Opción 3: Llevar libros

```
For the user Selected Menu
1. View all books
2. Search books
3. Get books
4. Return books
0. Quit
3
Selected the book you want to pick?
Eragon
the book has sucess you have to weeks to return

For the user Selected Menu
1. View all books
2. Search books
3. Get books
4. Return books
0. Quit
```

La opción 3 es llevar los libros escoger el libro que vas a llevar para leer el libro. Este es lo mismo en la opción 2 de búsqueda que tiene que escribir el título solamente. Cuando escoges el libro ve a la primera opción de almacén del libro mira al final de la derecha que dice disponible cuando ves el libro que esta escogido la pantalla de la booleana dice No disponible que significa que ese libro se lo han llevado.

### Input validación opción 3

```
For the user Selected Menu
1. View all books
2. Search books
3. Get books
4. Return books
0. Quit
3
Selected the book you want to pick?
narnia
This book does not exist or you type without Capital letter
```

El input validarse es si el libro no escribe especifico tira este error mi recomendación de este es primero seleccionar opción 1 y entonces selecciona en 3 para tener bien escrito.

### Input validación parte 2 de opción 3

```
For the user Selected Menu
1. View all books
2. Search books
3. Get books
4. Return books
0. Quit
3
Selected the book you want to pick?
Eragon
This book has been taken
```

En esta pantalla si el libro vas a llevar el libro que ya está escogido sale esta pantalla que este libro que quieres llevas ya este cogido. Porque si vas a la primera opción de view all books dice en la derecha que esta no disponible después el libro que selecciona el libro que vas a llevar.

### Opción 4 Regresar libros

```
For the user Selected Menu
1. View all books
2. Search books
3. Get books
4. Return books
0. Quit
4
Enter the title of the book you want to return:
Eragon
Return succeeded!

For the user Selected Menu
1. View all books
2. Search books
3. Get books
4. Return books
0. Quit
1
Book Title: Eragon, author: Crishopher Paolini, price: 50.0, ISBN: 254352, release date: 26/08/2002 Aavailable
Book Title: Sword art online, author: Reki Kawahara, price: 68.0, ISBN: 4023452, release date: 10/04/2009 Aavailable
Book Title: Narnia, author: C.S.LEWIS, price: 16.0, ISBN: 4023452, release date: 16/10/1950 Aavailable
```

La final opción numero 4 para el cliente es si quieres regresar los libros solamente entra el título de libro como los menús anteriores para que salga el booleana de libro para almacenar el libro con la disponibilidad.

### Input validación

```
For the user Selected Menu
1. View all books
2. Search books
3. Get books
4. Return books
0. Quit
4
Enter the title of the book you want to return:
Narnia
|This book has not return or is still available
```

El único de input validación es que si lo vas a regresar el libro si lo pone el título de libro que este almacén o no está en el almacén aparece este error que si el libro no está regresado o sigue disponibilidad de los libros.

### Opción 0: salir de programa

```
For the user Selected Menu
1. View all books
2. Search books
3. Get books
4. Return books
0. Quit
0
|You quit the program
```

El ultimo opción es si lo decides el numero cero (0) sales de programa por que la clase implementa con la interface de Selected puse `System.exit(0)` para terminar el output.

## 2. Administrador

### Menu de administrador

```
Welcome to the Torres Library for the administrator selected the menu
1. Add books
2. View all books
3. Delete books
4. Search books
0. Exit the system
|
```

Este menú se registra el administrador lo que hace es registrar de nuevo excepto la contraseña que solamente entra cualquier número que no sea 0. Si entras el número 0 entra el usuario. El menú es casi lo mismo el user por que tiene almacén y buscar excepto hay 2 opciones distintos para el administrador que es añadir y borrar libros.

### Opción 1: añadir libros

```
Welcome to the Torres Library for the administrator selected the menu
1. Add books
2. View all books
3. Delete books
4. Search books
0. Exit the system
1
Please enter the title
The best seller code
Please enter the author
Jadie Archer
Please Enter number isbn
45676587
Please enter price
3
Please enter quantity
2
Please enter the release date
07/31/2014
Added successfully
```

Este menú es para el administrador solamente hay que seleccionar la opción 1 añadir libros. Lo que hacen es añadir el título de libro con autor, barcode (isbn), el precio, la cantidad de lo que hace es multiplicar los precios y la



fecha de publicación de libro que han salido. No hay condiciones statement solamente es escribir para guardar el almacén de libros.

### **Opción 2: almacén de libros (con añadido)**

```
Welcome to the Torres Library for the administrator selected the menu
1. Add books
2. View all books
3. Delete books
4. Search books
0. Exit the system
2
Book Title: Eragon, author: Crishopher Paolini, price: 50.0, ISBN: 254352, release date: 26/08/2002 Available
Book Title: Sword art online, author: Reki Kawahara, price: 68.0, ISBN: 4023452, release date: 10/04/2009 Available
Book Title: Narnia, author: C.S.LEWIS, price: 16.0, ISBN: 4023452, release date: 16/10/1950 Available
Book Title: The best seller code, author: Jadie Archer, price: 6.0, ISBN: 45676587, release date: 07/31/2014 Available
```

Esta pantalla después de añadir los libros sale esta pantalla que estas registradoras con la bandera booleana que el libro esta disponibles. En donde está en la fila 4 está el libro que esta añadido.

### **Opicion 3: Borrar libros**

```
Welcome to the Torres Library for the administrator selected the menu
1. Add books
2. View all books
3. Delete books
4. Search books
0. Exit the system
3
Please enter the title of the book to be deleted
Eragon
Book was remove

Welcome to the Torres Library for the administrator selected the menu
1. Add books
2. View all books
3. Delete books
4. Search books
0. Exit the system
2
Book Title: Sword art online, author: Reki Kawahara, price: 68.0, ISBN: 4023452, release date: 10/04/2009 Available
Book Title: Narnia, author: C.S.LEWIS, price: 16.0, ISBN: 4023452, release date: 16/10/1950 Available
```

La opción 3 es borrar libros para el administrador lo que hace es borrar libros en el almacén de arreglo. El input solamente es escribir el titulo como buscar, llevar libros y regresar el libro en el user. Hay condición que tienes que escribir bien exacto como están en el almacén que por eso puse almacén de libros y buscar los

libros para no cometer error. Si borra un libro que quieres borrar, borra el almacén, si selecciona el 2 ver todos los libros aparece los libros excepto el libro que esta borrado está desaparecido con autor, precio, isbn, etc. porque lo seleccionaste el libro que vas a borrar. El código de objeto de la clase Libary.java aparece por que use sin archivos solamente el array initializer con objetos.

### **Borrar libros con input validación**

```
Welcome to the Torres Library for the administrator selected the menu
1. Add books
2. View all books
3. Delete books
4. Search books
0. Exit the system
3
Please enter the title of the book to be deleted
eragon
This book does not exist or the name is not specific
```

La opción de borrar también está incluida de input validación de que si el título de libro no escribe bien especifico o no están en el almacén imprime el libro no existen o no escribiste bien exacto recomendando ver todos los libros primero en la opción 2 antes de seleccionar 3 de borrar libros.

### **opción 4: Buscar libros (parte de administrador)**

```
Welcome to the Torres Library for the administrator selected the menu
1. Add books
2. View all books
3. Delete books
4. Search books
0. Exit the system
4
Enter the title of the book that you want to search
The best seller code
Book Title: The best seller code, author: Jodie archer, price: 6.0, ISBN: 2354654, release date: 07/31/2016 Aavailable
```

El penúltimo de seleccionar el menú es buscar libros es lo mismo el cliente de usuarios, pero para el administrador para que el busque el libro que quiere buscar. También puede buscar el libro después que añadir libros en el almacén puede buscar libros que no están en el objeto de la clase independiente de Libary.Java

pero primero hay que añadir el libro para asegurar que si el libro está disponible. También incluye el input validación en el usuario la única diferencia que este menú está registrado como el administrador.

### **opción 0: Salir programa para administrador**

```
Welcome to the Torres Library for the administrator selected the menu
1. Add books
2. View all books
3. Delete books
4. Search books
0. Exit the system
0
You quit the program
```

El último seleccionar el menú es salir de programa es poner 0 porque es el índice de arreglo que siempre empieza el cero sales de programa los mismo el usuario de la cuenta también administrador puede salir de programa.

### **b) Tabla con la lista de cada archivo y su descripción**

```
//instance field with the object array the book capacity
private Book[] books = new Book[20];

//size for getter and setter and print the storage book array
private int size = 0;

//constructor method
public Library() {

    //objects array
    //Title, author, price, isbn, quantity, release date
    //this book you can add whatever you want
    books[0] = new Book("Eragon", "Crishopher Paolini", 25, 254352, 2, "26/08/2002");
    books[1] = new Book("Sword art online", "Reki Kawahara", 34, 4023452, 2, "10/04/2009");
    books[2] = new Book("Narnia", "C.S.LEWIS", 8, 4023452, 2, "16/10/1950");

    size = 3;
}

//get method with arguments index with Book objects
public Book getBooks(int index) {
    return books[index];
}

//set method with arguments index including book class
public void setBooks(int index, Book books) {

    //set books index array using books
    this.books[index] = books;
}

//the get side will not include parameter but return the instance variable to print de index of the book.
public int getsizes() {
    return size;
}

//this size will use for adding or deleting the book, search book or show all books
public void setsize(int count) {
    this.size = this.size+count;
}
}
```

No lo use los archivos, pero use en objetos de arreglos iniciales para salir la pantalla del libro la única diferencia es que no tiene el guardar ni cargar por que el programa es temporero porque arreglos tiene límites de los libros en el almacén, también incluí con el getter y el setter para añadir, borrar, buscar, coger y regresar los libros (con la bandera booleana). El book.java esta de To Srting de sobre escritura de método estándar para incluir los objetos de arreglo en donde esta los paréntesis.

## **6. Conclusiones**

a) Describir todos los inconvenientes que se tuvieron en el transcurso del desarrollo del programa y las soluciones empleadas

El problema que me causo era como almacenar los libros porque yo estoy usando de arreglo sin usar el archivo de CSV. Eso es casi imposible de implementar cuando monto el diagrama en donde localiza en la fase 2. Se me hizo difícil de implementar almacén del libro hasta que contacte el tutor privado llamado Andy Sterkowitz y me dio sugerencias que tiene que hacer con tips y tricks usar es crear Libary.java ponerse el objeto de Book.java con arreglo iniciar para que salga los libros de almacén incluir añadir, borrar etc. Gracias a él lo resolví yo mismo implementar poco a poco hasta que completé el problema. También me dio otra dificultad sobre cómo hacer el username y password cuando cumpla todos los requisitos de orientadas de objetos para cumplir todos los requisitos. Lo único que resolví es crear método estático de llamado Login en objetos de User sin usar new porque este método en el Test. También resolví con control statement para contraseña solamente que si contraseña es cero registra usuario, y lo pones password cualquier número que no sea el cero entran el administrador usando new admin con paréntesis también en el usuario. La última dificultad es como usar el menú que no tengo ni idea como usar hasta que pedí otra cita de Andy es poner this selected[] con corchete de arreglo para seleccionar el menú que empieza desde números 0. Solamente entra el número, si lo entras las letras no números va a salir un error out of bound error que solamente entra el número. Aprendí nuevos trucos como usar el arreglo de menú sin usar swich case ni nested if que this selected incluye el brackets y semicolon gracias al tutor aumenten nueva experiencia.

- b) Indicar lo alcanzado en el proyecto versus lo esperado inicialmente (objetivos).

Los objetivos alcance con toda programación de orientada de objetos y ganar experiencia de aplicaciones que use registrar la cuenta y crear dos menús de usuario y el administrados por sobreescritura de métodos que son los únicos que me faltan hasta que cumplir este proyecto final. Me tomo mucho tiempo y estrés que me dio porque programe esto hasta 5:00am 4 veces al día para completar el programa hasta que logre todo el requisito en Java.

## **7. Referencias y apéndices (incluir aquí el “source code”)**

Referencias:

1. <https://www.youtube.com/watch?v=Ah8JaZoWTxl> java getter and setter
2. [https://www.w3schools.com/java/java\\_arrays.asp](https://www.w3schools.com/java/java_arrays.asp) array tutorials
3. <https://stackoverflow.com/questions/23458712/how-to-remove-an-element-previously-added-into-an-array> delete books
4. <https://coderanch.com/t/676193/java/array-book-objects> array books ideas.
5. [https://go.oncehub.com/AndySternkowitz?utm\\_medium=description&utm\\_campaign=GvLjhHKZUkg&utm\\_source=youtube](https://go.oncehub.com/AndySternkowitz?utm_medium=description&utm_campaign=GvLjhHKZUkg&utm_source=youtube) tutorías de Andy Sternkowitz.

## Book.java

```
1
2 public class Book {
3
4     //instance field private with boolean
5     private String title;
6     private String author;
7     private int isbn;
8     private double price;
9     private double quantity;
10    private String releaseDate;
11    private boolean Borrow;
12
13    //constructor method
14    public Book() {
15
16        title = "";
17        author = "";
18        isbn = 0;
19        price = 0;
20        quantity = 0;
21        releaseDate = "";
22    }
23
24    //method with arguments overloading
25    public Book(String title, String author, double price, int isbn, double quantity, String releaseDate) {
26        this.title = title;
27        this.author = author;
28        this.isbn = isbn;
29        this.price = price;
30        this.quantity = quantity;
31        this.releaseDate = releaseDate;
32    }
33
34    //get title
35    public String getTitle() {
36        return title;
37    }
38
39    //setter title
40    public void setTitle(String title) {
41        this.title = title;
42    }
43
44    //get author
45    public String getAuthor() {
46        return author;
47    }
48
49    //set author
50    public void setAuthor(String author) {
51        this.author = author;
52    }
53
54    //get price
55    public double getPrice() {
56        return price;
57    }
58
59    //set price
60    public void setPrice(double price) {
61        this.price = price;
62    }
63
64 }
```

```

//set price
public void setPrice(double price) {
    this.price = price;
}

//get isbn
public int getISBN() {
    return isbn;
}

//set isbn
public void setISBN(int isbn) {
    this.isbn = isbn;
}

//get boolean the borrow
public boolean getBorrow() {
    return Borrow;
}

//set boolean the borrow
public void setBorrow(boolean borrow) {
    Borrow = borrow;
}

//method the sub total price
public double subTotal() {

    return price * quantity;
}

//calculate the total price
public double totalPrice() {

    //return the method for total price
    return subTotal();
}

//Overrriding method to String
public String toString() {

    return "Book Title: " + title + ", " + " author: " + author + ", " + " price: " + totalPrice() +
    ", " + " ISBN: " + isbn + ", " + " release date: " + releaseDate + " " + (Borrow ? "Not available":"Avaiable")+" ";
}
}

```

## Library.java

```
public class Library{

    //intance field with the object array the book capacity
    private Book[] books = new Book[20];

    //size for getter and setter and print the storage book array
    private int size = 0;

    //constructor method
    public Library() {

        //objects array
        //Title, author, price, isbn, quantity, release date
        //this book you can add whetever you want
        books[0] = new Book("Eragon", "Crishopher Paolini",25, 254352, 2,"26/08/2002");
        books[1] = new Book("Sword art online", "Reki Kawahara", 34, 4023452, 2,"10/04/2009");
        books[2] = new Book("Narnia", "C.S.LEWIS", 8, 4023452, 2,"16/10/1950");

        size = 3;
    }

    //get method with arguments index with Book objects
    public Book getBooks(int index) {
        return books[index];
    }

    //set method with arguments index including book class
    public void setBooks(int index,Book books) {

        //set books index array using books
        this.books[index] = books;
    }

    //the get side will not include paramenter but return the instance variable to print de index of the book.
    public int getsie() {
        return size;
    }

    //this size will use for adding or deleting the book, search boook or show all books
    public void setsize(int count) {
        this.size = this.size+count;
    }
}
```



## Abstract class Menu.java

```
2 //input the Scanner
3 import java.util.Scanner;
4
5 //Abstract class
6 public abstract class Menu {
7
8     //the array interface protected instance field
9
10    //protected method selected menu interface including array
11    protected Selected[] selected;
12
13    //create scanner input
14    Scanner sc = new Scanner(System.in);
15
16    //mensaje is method static void
17    public static void mensaje() {
18
19        System.out.println("Este es mi proyecto final\n");
20
21    }
22
23    //abstract menu with integer
24    public abstract int menu();
25
26    //void method using choice in for Menu driver and both Library declaring with do selected
27    public void doSelected(int choice, Library library){
28
29        // selected menu include index the choice and classes the implements with parenthesis library class
30        selected[choice].classes(library);
31
32    }
33 }
34 }
```

## Interface Selcted.java

```
//import scanner input
import java.util.Scanner;

public interface Selected {

    //Creating input
    Scanner sc = new Scanner(System.in);

    //method void to apply other class
    void classes(Library library);
}
```

## AddBooks.java

```
//implements Selected
public class Addbooks implements Selected {

    //complete

    public void classes(Library library) {

        //instance field with variables getsize library class
        int size = library.getsize();

        //input title
        System.out.println("Please enter the title");
        String title = sc.nextLine();

        //input author
        System.out.println("Please enter the author");
        String author = sc.nextLine();

        //input isbn
        System.out.println("Please Enter number isbn");
        int isbn = sc.nextInt();

        //input price
        System.out.println("Please enter price");
        double price = sc.nextDouble();

        //input quantity
        System.out.println("Please enter quantity");
        double quantity = sc.nextDouble();

        //input release date
        System.out.println("Please enter the release date");
        String date = sc.next();

        //The book object with selected parenthesis objects
        Book book = new Book(title,author,price,isbn, quantity,date);

        //set book to write size and book using library class
        library.setBooks(size,book);

        //adding a size of book using library get size
        library.setsize(1);

        //print with the add book
        System.out.println("Added successfully");
    }
}
```

## DeleteBooks.java

```
public class DeleteBook implements Selected{

    //Complete

    //method implements
    public void classes(Library library) {

        //output the delete
        System.out.println("Please enter the title of the book to be deleted");

        //input the title for delete the space

        //instance field the String
        String title = sc.nextLine();

        //instance field
        int i = 0;

        //this loop use for delete book with specific line the array
        for (; i < library.getsize(); i++) {

            //if use get book including parenthesis index with title including equals
            if(library.getBooks(i).getTitle().equals(title)){

                //break to avoid the buffer
                break;
            }
        }

        //control flow the book input validation.
        if(i >= library.getsize()){

            System.out.println("This book does not exist or the name is not specific");
            //adding return the menu
            return;
        }

        //else use the title was found.
        else {

            //using for loop for delete the book
            for (int j = i; j < library.getsize()-1; j++) {

                //declaring set books for locating especific index with sum 1 for location
                library.setBooks(j,library.getBooks(j+1));
            }

            //The result will use outside the loop because inside the loop for delete wont work
            System.out.println("Book was remove");
            library.setsize(-1);
        }
    }
}
```

## DisplayAllBooks.java

```
public class DisplayAllBooks implements Selected {  
  
    //implements the classes  
    public void classes(Library library) {  
  
        //for loops with getsize for disply all books of array  
        for (int i = 0; i < library.getsize(); i++) {  
  
            //print using index  
            System.out.println(library.getBooks(i));  
        } //end loop  
  
    } // end implement method  
} //end class
```

## Quit.java

```
public class Quit implements Selected {  
  
    //this is the exit system in java  
  
    //implement Salaected book  
    public void classes(Library library) {  
  
        //Library log out  
        System.out.println("You quit the program ");  
        |  
        //exit function which cause JVM to exit.  
        System.exit(0);  
  
    }  
}
```

## SearchBook.java

```
public class SearchBook implements Selected{

    //search book will use nested if

    public void classes(Library library) {
        //asking the output
        System.out.println("Enter the title of the book that you want to search");

        //input the title of the book
        String title = sc.nextLine();

        //index 0 for for loop
        int i =0;

        //loop the storage book
        for(; i < library.getsize(); i++) {

            //control flow the if statement if
            if(library.getBooks(i).getTitle().equals(title))

                //break loop will use for only selected one
                break;
        }

        if(i >= library.getsize()){
            System.out.println("This book does not exist or type specific program");
        }

        else
        {
            System.out.println(library.getBooks(i));
        }

    }

} //end method
} //end class
```

## Getbook.java

```
public class GetBook implements Selected{

    public void classes(Library library) {

        //output asking
        System.out.println("Selected the book you want to pick?");

        //input title of the book
        String title = sc.next();

        //instance field de index for for loop using techniques to obtain getter and setter
        int i =0;

        for (; i < library.getSize(); i++) {
            if(library.getBooks(i).getTitle().equals(title)){

                //using break to display one book get only
                break;
            }
        }

        //input validation part 1
        if(i >= library.getSize()){
            System.out.println("This book does not exist or you type without Capital letter");

        }

        //input validation for book not available
        else if(library.getBooks(i).getBorrow()){
            System.out.println("This book has been taken");

        }

        else {

            //The book obtain result with boolean True
            library.getBooks(i).setBorrow(true);
            System.out.println("the book has sucess you have two weeks to return");

        }
    }
}
```

## ReturnBook.java

```
public class ReturnBook implements Selected{

    //implemnt selected method
    public void classes(Library library) {

        //output asking the title of the book
        System.out.println("Enter the title of the book you want to return:");

        //input the title of the book
        String title = sc.nextLine();

        //index instance field to apply the control flow
        int i = 0;

        //for loop the array library (storage book) including break for display only one book
        for (; i < library.getsize(); i++) {

            //control flow for title of the book break the index to avoid all dispaly.
            if(library.getBooks(i).getTitle().equals(title)) {
                break;
            }
        }

        //input validation 1
        if (i >= library.getsize()) {
            System.out.println("This book does not exist ");
        }

        //input validation the book has not return using false Bool
        else if (library.getBooks(i).getBorrow() == false) {

            System.out.println("This book has not return or is still available");

        }

        //displaying result the book return
        else {
            library.getBooks(i).setBorrow(false);

            System.out.println("Return succeeded!");
        }
    } //end implement class
} //end class
```

## User.java

```
1 public class User extends Menu {
2
3     //extending abstract class menu
4
5
6     //complete
7
8     //constructor method
9     public User() {
10
11         //this method array with adding bracket like using method
12         this.selected = new Selected[] {
13
14             //array with new objects menu selected with coma selecting index 0
15             new Quit(),
16             new DisplayAllBooks(),
17             new SearchBook(),
18             new GetBook(),
19             new ReturnBook(),
20
21         }; //addingg semicolon
22     } //end constructor
23
24
25     //abstract class menu
26     public int menu(){
27
28         //Menu mention
29         System.out.println();
30         System.out.println("For the user Selected Menu");
31         System.out.println("1. View all books");
32         System.out.println("2. Search books");
33         System.out.println("3. Get books");
34         System.out.println("4. Return books");
35         System.out.println("0. Quit");
36
37         //variables the input
38         int choice = sc.nextInt();
39
40         //return method
41         return choice;
42     } //end method menu
43
44
45 } // end class
```



## Admin.java

```
1
2 public class Admin extends User {
3
4     //extending abstract class menu
5
6
7     //complete
8
9     //constructor method
10    public Admin() {
11
12        //this method array with adding bracket like using method
13        this.selected = new Selected[] {
14
15            //array with new objects menu selected with coma selecting index 0
16            new Quit(),
17            new Addbooks(),
18            new DisplayAllBooks(),
19            new DeleteBook(),
20            new SearchBook(),
21
22        }; //adding semicolon
23
24    } //end constructor
25
26
27    //OVERRIDING METHOD
28    public int menu(){
29
30        //Menu mention
31        System.out.println();
32        System.out.println("Welcome to the Torres Library for the administrator selected the menu");
33        System.out.println("1. Add books");
34        System.out.println("2. View all books");
35        System.out.println("3. Delete books");
36        System.out.println("4. Search books");
37        System.out.println("0. Exit the system");
38
39        //variables the input
40        int choice = sc.nextInt();
41
42        //return method
43        return choice;
44
45    } //end method menu
46
47
48
49
50
51 } // end class
52
```

## Test.java

```
1 import java.util.Scanner; //creatin input for login method static
2
3 public class Test {
4
5     //main mehtod
6     public static void main(String[] args) {
7
8         //objects the user menu
9         User x = Login();
10
11
12         //objects the library
13         Library y = new Library();
14
15         //the loop statement using only true to get infinite loop until you quit the menu whatever is user or admin.
16         while(true) {
17             //declaring variables with the menu
18             int choice = x.menu();
19
20             //this include user with selected
21             x.doSelected(choice,y);
22
23         }
24     }
25
26     //Public static method for the user
27     public static User login() {
28
29         //creating scanner
30         Scanner sc = new Scanner(System.in);
31
32         //asking output the username
33         System.out.print("Username: ");
34
35         //input the username
36         String username = sc.next();
37
38         System.out.print("Enter you password: ");
39         int choice = sc.nextByte();
40
41         //the number password with control statement
42         if (choice == 0){
43
44             //abstract mensaje
45             Menu.mensaje();
46
47             //Display for the username library
48             System.out.println("Welcome " + username + " to the Torres Library");
49
50             //the user will include new and return
51             return new User();
52         }
53
54         else {
55
56             //if not enter the password different numer not 0 will enter the administrator account.
57             return new Admin();
58         }
59     }
60 }
61
62 //end method login
63 }
64
65 // end class
```