

UNIVERSIDAD AUTÓNOMA DE COAHUILA
FACULTAD DE SISTEMAS

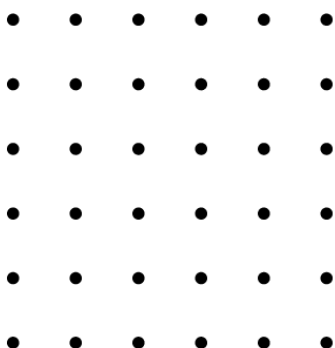
SISTEMA DE APARTADO **PARA LIBRERÍA**

MATERIA :ADMINISTRACIÓN DE
PROYECTOS DE SOFTWARE

RICARDO GABRIEL RODRÍGUEZ GONZÁLEZ
MATRÍCULA 17001433
OCTAVO SEMESTRE
ING SISTEMAS COMPUTACIONALES

SALTILLO COAH, 02 DE MAYO DE 2023

CARGANDO





Contenido

FreeLancer.....	4
1. FreeLancer.....	4
2 ApexWebCube.....	4
3. Script para capturar datos de una web.....	6
Workana.....	6
1. Aplicación Móvil para pedidos y mandarlos a una base de datos SQL Server.....	6
2. Implementar en HTML sitio web.....	8
Upwork.....	9
Video Editing Tools Consultant (Subtitle Application Experience Preferred).....	9
Document processing with OpenAI API.....	10
Tutor for Data Analytics (Python and SQL).....	10
Servicios de Hosting y de LMS.....	12
Servicios de Hosting.....	12
Plataformas de LMS.....	13
Análisis FODA.....	14
Diagrama de Gantt.....	15
23-ene al 7-mar.....	15
7mar al 15-abr.....	15
Proyecto Final.....	16
FlatLaf.....	16
Página principal.....	17
Prestamos de un libro.....	18
Devoluciones.....	19
Usuarios.....	19
Registrar usuarios.....	20
Libros.....	20
Agregar un nuevo libro.....	21
Registro.....	21
Base de Datos.....	22
Modelos de Usuarios, Prestamos y Libros.....	23
Usuarios.....	23
Libros.....	25



Prestamos..... 28

Interfaces Usuarios, Libros y Prestamos..... 29

 Usuarios..... 29

 Prestamos..... 30

 Libros 30

Implementación de las interfaces Usuarios, Prestamos y Libros 31

 Usuarios..... 31

 Prestamos..... 34

 Libros 37

Utilidad 41

 Utils 41

Dashboard Principal (Codigo)..... 43

Vistas (views)..... 53

Conclusión 54

Referencias bibliográficas..... 54



FreeLancer

1. FreeLancer

Busco experto en SEO para mejorar mis sitios web en WordPress

Job Description:

¡Hola Freelancers! Estoy buscando un experto en SEO para mejorar el posicionamiento de mis sitios web en WordPress.

Mis sitios web tienen un diseño atractivo y moderno, pero necesito mejorar el SEO para poder atraer más tráfico y aumentar las ventas. Algunos de mis sitios utilizan Woocommerce, por lo que es importante que el candidato tenga experiencia en este plugin.

Busco a alguien que tenga experiencia demostrable en SEO y que esté familiarizado con herramientas como Semrush, Metricool y/o Yoast SEO. Espero que el candidato tenga un conocimiento sólido en la optimización On-page y Off-page.

El proyecto incluirá la revisión y análisis de los sitios web, la identificación de las palabras clave adecuadas, la optimización de los metadatos y la estructura del sitio, la creación de contenido relevante, y la implementación de estrategias de link building. Es importante que el candidato tenga una buena comunicación y que esté dispuesto a trabajar en colaboración conmigo para lograr los objetivos.

Si eres un experto en SEO apasionado y estás listo para ayudarme a mejorar mis sitios webs, entonces este es el proyecto adecuado para ti. ¡Espero con interés trabajar contigo y ver resultados impresionantes juntos! Habilidades: SEO, Construcción de enlaces, Marketing por Internet, WordPress, SEMrush

PRESUPUESTO \$30-250 USD

2 ApexWebCube

\$8 USD / hour

FEATURED REVIEW



"Let me start off by saying thank you! Great job to Apex Web Cube it has been a complete pleasure working with you. Its nice when you can message a freelancer and them get back to you within a couple minuets to help with whatever you are working on and in specific Apex Web Cube helps in a lot of other ways as well. Not just SEO they have no problem going above and beyond when asked about anything else to do with my website. I usually get a detailed answer of my question within a couple hours which is great. for someone who has little experience on websites to have Apex Web Cube behind you really helps when you need something done in a reasonable manner. Also Apex Web Cube does my blog posts they are very good blog writers and I am lucky to have found them so that I only have one freelancer to go to with multiple things. So if you are looking for a freelancer that can do your SEO and blog posts and help with tons of other things I.E (go above and beyond) on your website. Have confidence when choosing Apex Web Cube I am so lucky to have found you and thank you for your hard work you are awesome and I look forward to working with you for quite some time. "

- Samuel S.

White Hat SEO | Social Media Marketing | Google Adwords | Web Development

Web Design and Development, High-Quality Link Building, Technical SEO audits, Comprehensive Niche Analysis, Keyword plans and Promotional Campaigns like Google Ads, SMM (Social Media Marketing).

Have enhanced the ranks and reputation of a wide range of websites during the previous ten years. My portfolio currently has almost 2900+ projects.

Dedicated to accomplishing company objectives.

- ▶ SEO (on-page and off-page optimizations)
- ▶ WordPress Development
- ▶ Web Design



- ▶ Page Speed Optimization
- ▶ Google ads Campaign Management
- ▶ Competitor Analysis
- ▶ Website Audit (technical, content and link audits)
- ▶ Link building on High DA and PA based site to bring traffic
- ▶ Proper setup of Google Webmaster tool and Analytic
- ▶ SMM Campaign Management

I understand all of the processes to a successful website marketing, and quality is always my first concern.

[3. Script para capturar datos de una web](#)

Job Description: Realizar un script para saber el puesto de votación de una página web , tomando el número de identificación de una base de datos y exportar los datos que arroja la búsqueda a otra base de datos. La página es <https://wsp.registraduria.gov.co/censo/consultar> Habilidades: PHP, JavaScript, Web Scraping, MySQL, HTML PRESUPUESTO \$30.00 – 250.00 USD

Workana

[1. Aplicación Móvil para pedidos y mandarlos a una base de datos SQL Server](#)

Sobre este proyecto:

Necesito una aplicación móvil para levantar pedidos con tablets o celulares por lo práctico,

no se usarán funciones de los dispositivos.

Desde la aplicación móvil solo se darán de alta los pedidos, asignan perfiles de usuarios y

Control de permisos de usuarios.



Los catálogos de clientes, productos y vendedores se dan de alta desde el punto de venta con base de datos en SQL Server.

Estos datos deben estar disponibles para la aplicación web y se estén actualizando mientras tengan conexión a esta base de datos

También se pueda guardar datos en el móvil cuando se queden sin conexión.

También se pueda guardar datos en el móvil cuando se queden sin conexión. Una vez tomados los pedidos y cuando tengan conexión que se manden los pedidos a esta base de datos.

Categoría: Programación y Tecnología

Subcategoría: Programación de Apps. Android, iOS y otros

¿Cuál es el alcance del proyecto? Crear una aplicación nueva

¿Es un proyecto o una posición? Un proyecto

Actualmente tengo las especificaciones

Disponibilidad requerida: Según se necesite

Necesidad específica: Rediseñar un sitio web

Plataformas requeridas: Android, iPhone, iPad

Necesito: Crear una app

Tipo de app: Gestión - Aplicación de gestión (turnos, stock de mercadería, etc.)

Necesita backend: Sí

Procesador de pago: No

Plazo de Entrega: 15 mayo, 2023

Habilidades necesarias: CSS, HTML, Responsive Web Design, HTML5, JavaScript, SQL

Server, Android, iPhone, Dart, Flutter



Presupuesto: USD 250 – 500

2. Implementar en HTML sitio web

Sobre este proyecto:

Tenemos un sitio web ya diseñado, archivos fuente en formato psd.

Requiero que se genere archivos HTML y css necesarios para publicación del sitio.

La conversión debe ser idéntica a los diseños conceptuales.

Son 5 archivos psd correspondientes al homepage y 4 páginas internas

Categoría: Programación y Tecnología

Subcategoría: Programación Web

¿Cuál es el alcance del proyecto? Crear un nuevo sitio personalizado

¿Es un proyecto o una posición? Un proyecto

Actualmente tengo el diseño

Disponibilidad requerida: Según se necesite

Roles necesarios: Programador

Plazo de Entrega: No definido

Habilidades: JavaScript, CSS, HTML, HTML5, Responsive Web Design

Presupuesto: USD 100 – 250



Upwork

Video Editing Tools Consultant (Subtitle Application Experience Preferred)

Search more Full Stack Development Jobs

We are seeking a knowledgeable consultant with experience in video editing tools, especially subtitle applications, to provide guidance and advice for our SaaS web product, SubtitleO. Our project is built on Node.js and utilizes FFMPEG for subtitles. However, we are open to exploring solutions in Python as well.

One of the primary areas where we need assistance is in styling subtitles. We have found limited resources and community support online for this specific aspect. Your expertise will be invaluable in helping us create various subtitle templates with unique styling to cater to different user preferences.

Requirements:

Proven experience with video editing tools and applications

Expertise in subtitle application development is highly preferred

Strong understanding of web development technologies, particularly Node.js and Python

Experience with FFMPEG for subtitles

Specific knowledge of subtitle styling and templating

Excellent problem-solving skills and the ability to troubleshoot complex technical issues

Strong communication and interpersonal skills

Presupuesto: \$10.00-\$20.00

Duration: <1 month

Level: Expert

Remote Job.



Document processing with OpenAI API

This project involves

- 1.- setup, installation of existing back end code, open source (Apache 2.0 license) software in cloud (Azure, AWS)
- 2.- testing existing code and its integration
- 3.- write Javascript, Node.JS demo front ends that will take simple input from the user and displaying results from the back end.
- 4.- write a front end in Javascript and Python to manage configuration

Presupuesto: \$15.00-\$20.00 dollars for day

Duration < 1 month

Level: Expert Level

Tutor for Data Analytics (Python and SQL)

estamos buscando un analista de datos experto que pueda brindar clases de enseñanza.

preferencias: femenino

hablante nativo de inglés

de la región de la Estados Unidos

the lessons will be delivered during the day (CET) daily, 09:00-12:30

<https://www.womenaiacademy.com/home-eng>

kind regards

naz

Presupuesto: \$40.00-\$80.00 for Hourly

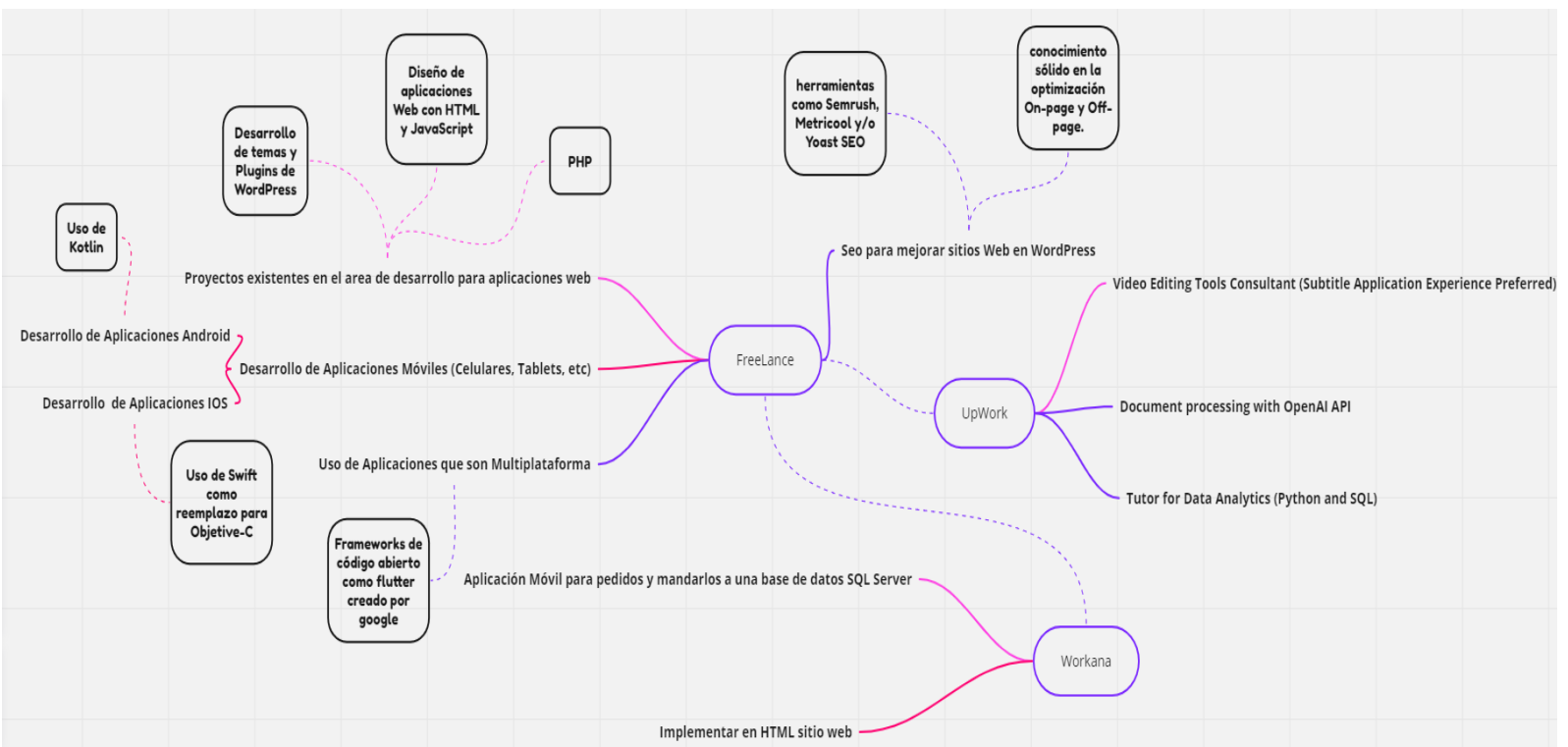
Less than 30 hrs/week

Hourly

Duration: 3-6 months

Remote Job.

Level: Expert





Servicios de Hosting y de LMS

Servicios de Hosting

Aquí están 15 servicios de hosting con una breve descripción de cada uno:

1. Bluehost: Un servicio de hosting de alta calidad con una interfaz fácil de usar y un gran soporte al cliente.
2. HostGator: Un servicio de hosting popular con una amplia gama de opciones de hosting y una interfaz intuitiva.
3. InMotion Hosting: Un servicio de hosting con una gran velocidad y disponibilidad, y una amplia gama de opciones de alojamiento.
4. A2 Hosting: Un servicio de hosting con un enfoque en la velocidad y una amplia gama de opciones de alojamiento.
5. SiteGround: Un servicio de hosting con una gran velocidad y disponibilidad, y un enfoque en la seguridad.
6. WP Engine: Un servicio de hosting especializado en WordPress con un enfoque en la velocidad y la seguridad.
7. DreamHost: Un servicio de hosting con una amplia gama de opciones de alojamiento y un enfoque en la simplicidad.
8. GreenGeeks: Un servicio de hosting con un enfoque en la sostenibilidad y una gran velocidad y disponibilidad.
9. Hostinger: Un servicio de hosting de bajo costo con una gran velocidad y disponibilidad.
10. Namecheap: Un servicio de hosting con un enfoque en el precio y una amplia gama de opciones de alojamiento.
11. TMD Hosting: Un servicio de hosting con una amplia gama de opciones de alojamiento y un enfoque en la velocidad y la seguridad.
12. Liquid Web: Un servicio de hosting con un enfoque en la velocidad y la disponibilidad, y una amplia gama de opciones de alojamiento.
13. DigitalOcean: Un servicio de hosting en la nube con un enfoque en la simplicidad y una gran escalabilidad.
14. Linode: Un servicio de hosting en la nube con una gran escalabilidad y una amplia gama de opciones de alojamiento.



15. Vultr: Un servicio de hosting en la nube con una gran escalabilidad y un enfoque en la velocidad y la disponibilidad.

Plataformas de LMS

1. Blackboard: plataforma de aprendizaje en línea que brinda soluciones personalizadas para instituciones educativas.
2. Canvas: plataforma de aprendizaje en línea intuitiva y fácil de usar con una amplia gama de funciones.
3. Moodle: plataforma de aprendizaje de código abierto utilizada por instituciones educativas y empresas en todo el mundo.
4. Schoology: plataforma de aprendizaje en línea que combina recursos de clase y colaboración en un solo lugar.
5. TalentLMS: plataforma de aprendizaje en línea simple y fácil de usar, diseñada para empresas y formadores independientes.
6. Thinkific: plataforma de aprendizaje en línea para crear y vender cursos en línea.
7. D2L Brightspace: plataforma de aprendizaje en línea que ofrece personalización y flexibilidad para instituciones educativas y empresas.
8. Edmodo: plataforma de aprendizaje en línea que permite a los profesores y estudiantes conectarse y colaborar en línea.
9. Google Classroom: plataforma de aprendizaje en línea gratuita que permite a los profesores crear y gestionar clases en línea.
10. Kajabi: plataforma de aprendizaje en línea que permite crear y vender cursos en línea y ofrecer una experiencia de membresía.

Teachable: plataforma de aprendizaje en línea que permite crear y vender cursos en línea de manera fácil y efectiva.



Análisis FODA

	Útil	Perjudicial
Interno	Fortalezas <ul style="list-style-type: none">• Poder organizar todo el contenido de una librería para así eficientar el stock de la misma.• Intuitiva y de fácil funcionamiento.• Una mejor administración para el préstamo de libros.• Fácil visualización de un registro de préstamo de libros en la librería.• Fácil contacto con el administrador del sistema de librería• Feedback de parte del administrador	Debilidades <ul style="list-style-type: none">• Poco conocimiento sobre el tema de administración de una librería.• Tiempo de desarrollo del proyecto para obtener un mejor feedback del cliente• Falta de tiempo para algunas opciones de desarrollo por parte del cliente• Distancia para el despliegue del sistema
	Oportunidades <ul style="list-style-type: none">• Tener la oportunidad de mejorar el desempeño del negocio de cliente• Poder interactuar más con este cliente para más oportunidades de negocio• Adquisición de experiencia en el desarrollo de sistemas de inventario• Oportunidad de aprender a desplegar conexiones VPN para el correcto funcionamiento del sistema	Amenazas <ul style="list-style-type: none">• Falta de experiencia trabajando con sistemas de inventario (lo veo más como un reto).• Distancia entre ciudades para despliegue del sistema.• Falta de tiempo para hacer pruebas adecuadas de testing de la aplicación• Falta de conocimiento del usuario sobre el tema de tecnologías de la computación



Proyecto Final

Para comenzar con una explicación breve del proyecto, es una aplicación para la sección de apartados en la librería llamada Fray servando Teresa de Mier, este proyecto se realizo en su totalidad con Java, y lo novedoso del mismo, fue intentar aplicar una metodología de diseño moderna utilizando diversas librerías, para esto investigue como poder realizar un diseño moderno en este lenguaje que como todos sabemos es bastante antiguo.

También me concentre en hacer que el programa fuera responsivo, sumándole valor para poder ejecutarlo en pantallas de diferentes tamaños sin problemas.

Posteriormente me concentre en trabajar los apartados de los botones y todo lo relacionado a la funcionalidad del proyecto en cuanto a lo visible, para que sea mas atractivo para la vista y mas moderno teniendo en cuenta que java es un lenguaje de programación que se caracteriza por tener interfaces antiguas y feas.

Seleccione este lenguaje de programación principalmente porque es donde tengo mas experiencia resolviendo problemas y inconformidades que surgen durante el desarrollo de la aplicación, tuve relativamente poco tiempo para desarrollar la aplicación me quise ir a un ambiente seguro donde conozco mas sobre el mismo lenguaje.

Este proyecto me lo encargo mi hermano mayor al cual le estoy ayudando a agregar el sistema de apartado en su librería y poder implementar este mismo servicio.

FlatLaf

Flatlaf fue la librería que utilice para estilizar todo lo que tiene que ver con mi proyecto, todos sabemos que java swing es bastante feo en cuanto a un diseño grafico se habla, así que utilice esta librería, y con ayuda de la documentación que



ofrece en su pagina web pude mejorar bastante de muchas maneras todas las letras, títulos y diseños de la aplicación de una manera muy buena.

Adjunto imágenes del diseño realizado en el proyecto

Página principal



En esta imagen se ve claramente todo el HUD y patrón de diseño que estoy llevando en todo el proyecto, este patrón de diseño es constante.

En este apartado tenemos la bienvenida al usuario, que terminaría siendo el cajero o administrador de prestamos de la librería, como podemos observar en este apartado solamente hay un mensaje de bienvenida, la fecha del día que está corriendo, y una breve explicación del programa.



Prestamos de un libro

La imagen muestra una interfaz web para el sistema de préstamos de la Librería Fray servando Teresa de mier. El encabezado incluye un mensaje de bienvenida y la fecha actual, miércoles 26 de abril de 2023. A la izquierda hay un menú lateral con opciones como Principal, Préstamos, Devoluciones, Usuarios, Libros y Reportes. El área principal contiene un formulario para realizar un préstamo, con campos para el folio del usuario y el ID del libro, y un botón 'Prestar'. En el fondo del formulario hay una ilustración de una biblioteca con un mostrador y dos figuras estilizadas.

Bienvenido al sistema de prestamos de La Libreria Fray servando Teresa de mier!

Administración/Control/Libreria

Hoy es miércoles 26 de abril de 2023

Libreria

- Principal
- Préstamos
- Devoluciones
- Usuarios
- Libros
- Reportes

Biblioteca

Folio Usuario

Ingrese el folio del usuario.

Libro ID

Ingrese el ID del Libro a prestar.

Prestar

En este apartado podemos ver que solamente nos pide el folio del usuario, que terminaría siendo el ID del usuario y el ID de libro que vamos a prestar, el administrador de caja puede acceder a estos datos en cualquier momento que él quiera moviéndose por el menú lateral izquierdo y acceder directamente a el apartado de usuarios o libros, puede buscar el dato solicitado y regresar a esta ventana para poder realizar el préstamo.



Devoluciones

Bienvenido al sistema de prestamos de La Libreria Fray servando Teresa de mier!

Administración/Control/Libreria

Hoy es miércoles 26 de abril de 2023

Devolución de Libro

Folio Usuario

Ingrese el folio del usuario.

Libro ID

Ingrese el ID del Libro a devolver.

Devolver

A la hora de devolver el libro, el administrador debe teclear el ID del usuario y el ID del libro que se desea devolver, estos datos también los podemos conseguir del apartado antes mencionado.

Usuarios

Bienvenido al sistema de prestamos de La Libreria Fray servando Teresa de mier!

Administración/Control/Libreria

Hoy es miércoles 26 de abril de 2023

Usuarios

Ingrese el nombre de usuario a buscar. **Buscar**

ID	Nombre	Apellido P.	Apellido M.	Domicilio	Teléfono
----	--------	-------------	-------------	-----------	----------

Nuevo **Editar** **Borrar**



En esta sección podemos dar de alta a nuevos usuarios que se vayan a registrar en el sistema, podemos editar algún error que haya cometido el administrador, o simplemente un cambio de domicilio, etc. Y también podemos borrar usuarios que dado determinado tiempo ya no se hayan presentado.

Registrar usuarios

La interfaz muestra un menú lateral con opciones: Principal, Préstamos, Devoluciones, Usuarios, Libros y Reportes. El contenido principal está dividido en dos columnas. La columna izquierda contiene los campos de texto para el registro de un usuario: 'Nombre' (con el placeholder 'Ingrese el nombre del usuario.'), 'Apellido Paterno' (con el placeholder 'Ingrese el apellido paterno del usuario.'), 'Apellido Materno' (con el placeholder 'Ingrese el apellido materno del usuario.'), 'Domicilio' (con el placeholder 'Ingrese el domicilio del usuario.'), y 'Teléfono' (con el placeholder 'Ingrese el teléfono del usuario.'). La columna derecha contiene un botón azul con el texto 'Registrar'. Encabezando la sección principal, se muestra el mensaje 'Bienvenido al sistema de prestamos de La Librería Fray servando Teresa de mieri!' y la fecha 'Hoy es jueves 27 de abril de 2023'.

En este apartado están todos los campos que me pidieron registrar para tener información de los usuarios a los que se les va a prestar el libro.

Libros

La interfaz muestra el mismo menú lateral. El contenido principal incluye un campo de búsqueda con el placeholder 'Ingrese el título del libro a buscar.' y un botón azul 'Buscar'. Debajo de la barra de búsqueda, se encuentra una tabla con las siguientes columnas: ID, Título, Fecha de Pub., Autor, Categoría, Edición, Idioma, Páginas, Descripción, Ejemplares, Stock y Disponibles. La tabla está actualmente vacía. En la parte inferior de la sección, hay tres botones azules: 'Nuevo', 'Editar' y 'Borrar'. Encabezando la sección principal, se muestra el mensaje 'Bienvenido al sistema de prestamos de La Librería Fray servando Teresa de mieri!' y la fecha 'Hoy es jueves 27 de abril de 2023'.



En esta sección tenemos los libros disponibles a préstamo en la librería, el usuario administrador tiene la oportunidad de dar de alta a los nuevos libros que vayan llegando, editarlos por cualquier motivo, y hasta eliminarlos si así se desea.

Agregar un nuevo libro

Bienvenido al sistema de prestamos de La Libreria Fray servando Teresa de mier!

Administración/Control/Libreria

Hoy es jueves 27 de abril de 2023

Subir nuevo Libro

Título
Ingrese el título del libro

Idioma
Ingrese el idioma del libro.

Páginas
Ingrese las páginas totales del libro.

Descripción
Ingrese la descripción del libro.

Stock
Ingrese el stock total del libro.

Disponibles
Ingrese el número de libros di...

Ejemplares
Ingrese el número de ejemplares del libro.

Subir

Fecha de Publicación
Ingrese la de publicación del libro.

Autor
Ingrese el autor del libro.

Categoría
Ingrese la categoría del libro.

Edición
Ingrese la editorial del libro.

Esta es la ventana para agregar un nuevo libro a la base de datos.

Registro.

Bienvenido al sistema de prestamos de La Libreria Fray servando Teresa de mier!

Administración/Control/Libreria

Hoy es jueves 27 de abril de 2023

Reportes

User ID	Book ID	Fecha de Salida	Fecha de Entrega
---------	---------	-----------------	------------------

Actualizar



Este es el apartado de Registro, en este apartado podremos observar de manera fácil todos los préstamos que están en curso, la fecha de salida del libro y la fecha de entrega, con esta información, el administrador determinara si el libro se entrego en tiempo y forma y así determinar si esto mismo lleva una multa o no.

Todos los cobros de esta aplicación, o aportación monetaria de cualquier tipo a la hora de emitir un préstamo o recibir una multa, etc. Todo esto se hace de forma análoga, próximamente se estará trabajando con un sistema de punto de venta para poderlo vincular con este sistema y trabajar ese apartado. Por falta de tiempo no se incluyeron todos esos cambios en este proyecto.

Base de Datos

```
CREATE SCHEMA `libreria` ;

CREATE TABLE `libreria`.`books` (
  `id` int(11) NOT NULL,
  `title` varchar(255) NOT NULL,
  `date` varchar(255) NOT NULL,
  `author` varchar(255) NOT NULL,
  `category` varchar(255) NOT NULL,
  `edit` varchar(255) NOT NULL,
  `lang` varchar(255) NOT NULL,
  `pages` varchar(255) NOT NULL,
  `description` varchar(255) NOT NULL,
  `ejemplares` varchar(255) NOT NULL,
  `stock` int(11) NOT NULL,
  `available` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `libreria`.`lendings` (
  `id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `book_id` int(11) NOT NULL,
  `date_out` varchar(255) NOT NULL,
  `date_return` varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `libreria`.`users` (
  `id` int(11) NOT NULL,
  `name` varchar(30) NOT NULL,
```



```
`last_name_p` varchar(30) NOT NULL,  
`last_name_m` varchar(30) NOT NULL,  
`domicilio` varchar(250) DEFAULT NULL,  
`tel` varchar(25) DEFAULT NULL,  
`sanctions` int(11) DEFAULT '0',  
`sanc_money` int(11) NOT NULL DEFAULT '0'  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
ALTER TABLE `books`  
  ADD PRIMARY KEY (`id`);  
  
ALTER TABLE `lendings`  
  ADD PRIMARY KEY (`id`);  
  
ALTER TABLE `users`  
  ADD PRIMARY KEY (`id`);  
  
ALTER TABLE `books`  
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;  
  
ALTER TABLE `lendings`  
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;  
  
ALTER TABLE `users`  
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
```

Este es el Query que se utilizo para crear toda la base de datos del sistema.

Esta base de datos se encuentra en MySQL, sencillamente por la facilidad de funcionamiento con el mismo lenguaje de programación java, y por la facilidad que tiene usar esta base de datos en cuanto a conexiones.

Modelos de Usuarios, Prestamos y Libros

Usuarios

```
package com.mycompany.models;  
  
public class Users {  
    private int id;  
    private String name;
```



```
private String last_name_p;  
private String last_name_m;  
private String domicilio;  
private String tel;  
private int sanctions;  
private int sanc_money;  
  
public void setId(int id) {  
    this.id = id;  
}  
  
public void setName(String name) {  
    this.name = name;  
}  
  
public void setLast_name_p(String last_name_p) {  
    this.last_name_p = last_name_p;  
}  
  
public void setLast_name_m(String last_name_m) {  
    this.last_name_m = last_name_m;  
}  
  
public void setDomicilio(String domicilio) {  
    this.domicilio = domicilio;  
}  
  
public void setTel(String tel) {  
    this.tel = tel;  
}  
  
public void setSanctions(int sanctions) {  
    this.sanctions = sanctions;  
}  
  
public void setSanc_money(int sanc_money) {  
    this.sanc_money = sanc_money;  
}  
  
public int getId() {  
    return id;  
}  
  
public String getName() {  
    return name;  
}
```




```
}

public String getLast_name_p() {
    return last_name_p;
}

public String getLast_name_m() {
    return last_name_m;
}

public String getDomicilio() {
    return domicilio;
}

public String getTel() {
    return tel;
}

public int getSanctions() {
    return sanctions;
}

public int getSanc_money() {
    return sanc_money;
}
}
```

Ese es el modelo que utilice para la creación de los usuarios, y esto es básicamente toda la información que tiene nuestra misma base de datos en su tabla de usuarios.

Libros

```
package com.mycompany.models;

public class Books {
    private int id;
    private String title;
    private String date;
    private String author;
    private String category;
    private String edit;
    private String lang;
}
```



```
private String pages;
private String description;
private String ejemplares;
private int stock;
private int available;

public void setId(int id) {
    this.id = id;
}

public void setTitle(String title) {
    this.title = title;
}

public void setDate(String date) {
    this.date = date;
}

public void setAuthor(String author) {
    this.author = author;
}

public void setCategory(String category) {
    this.category = category;
}

public void setEdit(String edit) {
    this.edit = edit;
}

public void setLang(String lang) {
    this.lang = lang;
}

public void setPages(String pages) {
    this.pages = pages;
}

public void setDescription(String description) {
    this.description = description;
}

public void setEjemplares(String ejemplares) {
    this.ejemplares = ejemplares;
}
```



```
public void setStock(int stock) {
    this.stock = stock;
}

public void setAvailable(int available) {
    this.available = available;
}

public int getId() {
    return id;
}

public String getTitle() {
    return title;
}

public String getDate() {
    return date;
}

public String getAuthor() {
    return author;
}

public String getCategory() {
    return category;
}

public String getEdit() {
    return edit;
}

public String getLang() {
    return lang;
}

public String getPages() {
    return pages;
}

public String getDescription() {
    return description;
}
```



```
public String getEjemplares() {  
    return ejemplares;  
}  
  
public int getStock() {  
    return stock;  
}  
  
public int getAvailable() {  
    return available;  
}  
}
```

Esto es lo que utilizamos para el modelo de los libros, igual que en usuarios, estos son todos los datos que están involucrados en la base de datos.

Prestamos

```
package com.mycompany.models;  
  
public class Lendings {  
    private int id;  
    private int user_id;  
    private int book_id;  
    private String date_out;  
    private String date_return;  
  
    public void setId(int id) {  
        this.id = id;  
    }  
  
    public void setUser_id(int user_id) {  
        this.user_id = user_id;  
    }  
  
    public void setBook_id(int book_id) {  
        this.book_id = book_id;  
    }  
  
    public void setDate_out(String date_out) {  
        this.date_out = date_out;  
    }  
  
    public void setDate_return(String date_return) {
```



```
        this.date_return = date_return;
    }

    public int getId() {
        return id;
    }

    public int getUser_id() {
        return user_id;
    }

    public int getBook_id() {
        return book_id;
    }

    public String getDate_out() {
        return date_out;
    }

    public String getDate_return() {
        return date_return;
    }
}
```

Por último, este es el modelo que se siguió para los préstamos, todos sus valores son los mismos que las tablas de prestamos en la base de datos.

Interfaces Usuarios, Libros y Prestamos.

Para esto utilizare el famoso patrón de diseño DAO (Data Access Object), con el cual ya estoy familiarizado gracias a una clase que lleve sobre el manejo de base de datos.

Usuarios

```
package com.mycompany.interfaces;

import com.mycompany.models.Users;
import java.util.List;

public interface DAOUsers {
    public void registrar(Users user) throws Exception;
}
```



```
public void modificar(Users user) throws Exception;
public void sancionar(Users user) throws Exception;
public void eliminar(int userId) throws Exception;
public List<Users> listar(String name) throws Exception;
public Users getUserById(int userId) throws Exception;
}
```

Aquí esta el apartado de registrar, modificar, sancionar y eliminar, aquí mismo listamos y obtenemos el usuario en base a su Id para las búsquedas.

Prestamos

```
package com.mycompany.interfaces;

import com.mycompany.models.Books;
import com.mycompany.models.Lendings;
import com.mycompany.models.Users;
import java.util.List;

public interface DAOLendings {
    public void registrar(Lendings lending) throws Exception;
    public void modificar(Lendings lending) throws Exception;
    public Lendings getLending(Users user, Books book) throws Exception;
    // public void eliminar(Lendings user) throws Exception;
    public List<Lendings> listar() throws Exception;
}
```

Aquí tenemos lo mismo que en el anterior, pero no tomamos en cuenta la de eliminar prestamos ya que no es útil esta función teniendo en cuenta que queremos llevar todos los prestamos sin eliminarlos.

Libros

```
package com.mycompany.interfaces;

import com.mycompany.models.Books;
import java.util.List;

public interface DAOBooks {
    public void registrar(Books book) throws Exception;
    public void modificar(Books book) throws Exception;
    public void eliminar(int bookId) throws Exception;
    public List<Books> listar(String title) throws Exception;
}
```



```
    public Books getBookById(int bookId) throws Exception;  
}
```

Aquí sería lo mismo que en usuarios, ya que aquí si queremos registrar, modificar eliminar, listar, y buscar.

Implementación de las interfaces Usuarios, Prestamos y Libros

En esta parte podemos ver mas claramente toda la parte de los query para la base de datos, y donde se estarán ejecutando todos los scripts y a su vez cerrando la conexión con la base de datos.

Usuarios

```
package com.mycompany.libreria;  
  
import com.mycompany.db.Database;  
import com.mycompany.interfaces.DAOUsers;  
import com.mycompany.models.Users;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.util.ArrayList;  
import java.util.List;  
  
public class DAOUsersImpl extends Database implements DAOUsers {  
  
    @Override  
    public void registrar(Users user) throws Exception {  
        try {  
            this.Conectar();  
            PreparedStatement st = this.conexion.prepareStatement("INSERT  
INTO users(name, last_name_p, last_name_m, domicilio, tel)  
VALUES(?,?,?,?,?);");  
            st.setString(1, user.getName());  
            st.setString(2, user.getLast_name_p());  
            st.setString(3, user.getLast_name_m());  
            st.setString(4, user.getDomicilio());  
            st.setString(5, user.getTel());  
            st.executeUpdate();  
            st.close();  
        } catch(Exception e) {  
            throw e;  
        } finally {  
            this.Cerrar();  
        }  
    }  
}
```



```
    }  
}  
  
@Override  
public void modificar(Users user) throws Exception {  
    try {  
        this.Conectar();  
        PreparedStatement st = this.conexion.prepareStatement("UPDATE  
users SET name = ?, last_name_p = ?, last_name_m = ?, domicilio = ?, tel = ?  
WHERE id = ?");  
        st.setString(1, user.getName());  
        st.setString(2, user.getLast_name_p());  
        st.setString(3, user.getLast_name_m());  
        st.setString(4, user.getDomicilio());  
        st.setString(5, user.getTel());  
        st.setInt(6, user.getId());  
        st.executeUpdate();  
        st.close();  
    } catch (Exception e) {  
        throw e;  
    } finally {  
        this.Cerrar();  
    }  
}  
  
@Override  
public void eliminar(int userId) throws Exception {  
    try {  
        this.Conectar();  
        PreparedStatement st = this.conexion.prepareStatement("DELETE  
FROM users WHERE id = ?");  
        st.setInt(1, userId);  
        st.executeUpdate();  
        st.close();  
    } catch (Exception e) {  
        throw e;  
    } finally {  
        this.Cerrar();  
    }  
}  
  
@Override  
public List<Users> listar(String name) throws Exception {  
    List<Users> lista = null;  
    try {
```




```
        this.Conectar();
        String Query = name.isEmpty() ? "SELECT * FROM users;" : "SELECT
* FROM users WHERE name LIKE '%" + name + "%'";
        PreparedStatement st = this.conexion.prepareStatement(Query);

        lista = new ArrayList();
        ResultSet rs = st.executeQuery();
        while(rs.next()) {
            Users user = new Users();
            user.setId(rs.getInt("id"));
            user.setName(rs.getString("name"));
            user.setLast_name_p(rs.getString("last_name_p"));
            user.setLast_name_m(rs.getString("last_name_m"));
            user.setDomicilio(rs.getString("domicilio"));
            user.setTel(rs.getString("tel"));
            user.setSanctions(rs.getInt("sanctions"));
            user.setSanc_money(rs.getInt("sanc_money"));
            lista.add(user);
        }
        rs.close();
        st.close();
    } catch(Exception e) {
        throw e;
    } finally {
        this.Cerrar();
    }
    return lista;
}

@Override
public Users getUserById(int userId) throws Exception {
    Users user = null;

    try {
        this.Conectar();
        PreparedStatement st = this.conexion.prepareStatement("SELECT *
FROM users WHERE id = ? LIMIT 1;");
        st.setInt(1, userId);

        ResultSet rs = st.executeQuery();
        while(rs.next()) {
            user = new Users();
            user.setId(rs.getInt("id"));
            user.setName(rs.getString("name"));
            user.setLast_name_p(rs.getString("last_name_p"));

```



```
        user.setLast_name_m(rs.getString("last_name_m"));
        user.setDomicilio(rs.getString("domicilio"));
        user.setTel(rs.getString("tel"));
        user.setSanctions(rs.getInt("sanctions"));
        user.setSanc_money(rs.getInt("sanc_money"));
    }
    rs.close();
    st.close();
} catch(Exception e) {
    throw e;
} finally {
    this.Cerrar();
}
return user;
}

@Override
public void sancionar(Users user) throws Exception {
    try {
        this.Conectar();
        PreparedStatement st = this.conexion.prepareStatement("UPDATE
users SET sanctions = ?, sanc_money = ? WHERE id = ?");
        st.setInt(1, user.getSanctions());
        st.setInt(2, user.getSanc_money());
        st.setInt(3, user.getId());
        st.executeUpdate();
        st.close();
    } catch(Exception e) {
        throw e;
    } finally {
        this.Cerrar();
    }
}
}
```

Aquí está la implementación completa de toda la interfaz antes creada para usuarios. Es un extends de la implementación DAOUsuarios.

Prestamos

```
package com.mycompany.libreria;
```



```
import com.mycompany.db.Database;
import com.mycompany.interfaces.DAOlendings;
import com.mycompany.models.Books;
import com.mycompany.models.Lendings;
import com.mycompany.models.Users;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;

public class DAOLendingsImpl extends Database implements DAOLendings {

    @Override
    public void registrar(Lendings lending) throws Exception {
        try {
            this.Conectar();
            PreparedStatement st = this.conexion.prepareStatement("INSERT
INTO lendings(user_id, book_id, date_out) VALUES(?,?,?);");
            st.setInt(1, lending.getUser_id());
            st.setInt(2, lending.getBook_id());
            st.setString(3, lending.getDate_out());
            st.executeUpdate();
            st.close();
        } catch(Exception e) {
            throw e;
        } finally {
            this.Cerrar();
        }
    }

    @Override
    public void modificar(Lendings lending) throws Exception {
        try {
            this.Conectar();
            PreparedStatement st = this.conexion.prepareStatement("UPDATE
lendings SET user_id = ?, book_id = ?, date_out = ?, date_return = ? WHERE
id = ?");
            st.setInt(1, lending.getUser_id());
            st.setInt(2, lending.getBook_id());
            st.setString(3, lending.getDate_out());
            st.setString(4, lending.getDate_return());
            st.setInt(5, lending.getId());
            st.executeUpdate();
            st.close();
        }
```



```
    } catch(Exception e) {
        throw e;
    } finally {
        this.Cerrar();
    }
}

@Override
public Lendings getLending(Users user, Books book) throws Exception {
    Lendings lending = null;

    try {
        this.Conectar();
        PreparedStatement st = this.conexion.prepareStatement("SELECT *
FROM lendings WHERE user_id = ? AND book_id = ? AND date_return IS NULL
ORDER BY id DESC LIMIT 1");
        st.setInt(1, user.getId());
        st.setInt(2, book.getId());

        ResultSet rs = st.executeQuery();
        while (rs.next()) {
            lending = new Lendings();
            lending.setId(rs.getInt("id"));
            lending.setUser_id(rs.getInt("user_id"));
            lending.setBook_id(rs.getInt("book_id"));
            lending.setDate_out(rs.getString("date_out"));
            lending.setDate_return(rs.getString("date_return"));
        }

        st.close();
        rs.close();
    } catch (Exception e) {
        throw e;
    } finally {
        this.Cerrar();
    }

    return lending;
}

@Override
public List<Lendings> listar() throws Exception {
    List<Lendings> lista = null;
    try {
        this.Conectar();
```



```
        PreparedStatement st = this.conexion.prepareStatement("SELECT *
FROM lendings ORDER BY id DESC");

        lista = new ArrayList();
        ResultSet rs = st.executeQuery();
        while(rs.next()) {
            Lendings lending = new Lendings();
            lending.setId(rs.getInt("id"));
            lending.setUser_id(rs.getInt("user_id"));
            lending.setBook_id(rs.getInt("book_id"));
            lending.setDate_out(rs.getString("date_out"));
            lending.setDate_return(rs.getString("date_return"));
            lista.add(lending);
        }
        rs.close();
        st.close();
    } catch(Exception e) {
        throw e;
    } finally {
        this.Cerrar();
    }
    return lista;
}
}
```

Aquí está la implementación de la interfaz antes creada para presamos, como podemos ver, no está implementada la parte de eliminar, algo que ya habíamos explicado antes el motivo.

Libros

```
package com.mycompany.libreria;

import com.mycompany.db.Database;
import com.mycompany.interfaces.DAOBooks;
import com.mycompany.models.Books;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;

public class DAOBooksImpl extends Database implements DAOBooks {
```



```
@Override
public void registrar(Books book) throws Exception {
    try {
        this.Conectar();
        PreparedStatement st = this.conexion.prepareStatement("INSERT
INTO books(title, date, author, category, edit, lang, pages, description,
ejemplares, stock, available) VALUES(?,?,?,?,?,?,?,?,?,?,?,?);");
        st.setString(1, book.getTitle());
        st.setString(2, book.getDate());
        st.setString(3, book.getAuthor());
        st.setString(4, book.getCategory());
        st.setString(5, book.getEdit());
        st.setString(6, book.getLang());
        st.setString(7, book.getPages());
        st.setString(8, book.getDescription());
        st.setString(9, book.getEjemplares());
        st.setInt(10, book.getStock());
        st.setInt(11, book.getAvailable());
        st.executeUpdate();
        st.close();
    } catch(Exception e) {
        throw e;
    } finally {
        this.Cerrar();
    }
}

@Override
public void modificar(Books book) throws Exception {
    try {
        this.Conectar();
        PreparedStatement st = this.conexion.prepareStatement("UPDATE
books SET title = ?, date = ?, author = ?, category = ?, edit = ?, lang = ?,
pages = ?, description = ?, ejemplares = ?, stock = ?, available = ? WHERE
id = ?");
        st.setString(1, book.getTitle());
        st.setString(2, book.getDate());
        st.setString(3, book.getAuthor());
        st.setString(4, book.getCategory());
        st.setString(5, book.getEdit());
        st.setString(6, book.getLang());
        st.setString(7, book.getPages());
        st.setString(8, book.getDescription());
        st.setString(9, book.getEjemplares());
```



```
        st.setInt(10, book.getStock());
        st.setInt(11, book.getAvailable());
        st.setInt(12, book.getId());
        st.executeUpdate();
        st.close();
    } catch(Exception e) {
        throw e;
    } finally {
        this.Cerrar();
    }
}

@Override
public void eliminar(int bookId) throws Exception {
    try {
        this.Conectar();
        PreparedStatement st = this.conexion.prepareStatement("DELETE
FROM books WHERE id = ?;");
        st.setInt(1, bookId);
        st.executeUpdate();
        st.close();
    } catch(Exception e) {
        throw e;
    } finally {
        this.Cerrar();
    }
}

@Override
public List<Books> listar(String title) throws Exception {
    List<Books> lista = null;
    try {
        this.Conectar();
        String Query = title.isEmpty() ? "SELECT * FROM books;" :
"SELECT * FROM books WHERE title LIKE '%" + title + "%'";
        PreparedStatement st = this.conexion.prepareStatement(Query);

        lista = new ArrayList();
        ResultSet rs = st.executeQuery();
        while(rs.next()) {
            Books book = new Books();
            book.setId(rs.getInt("id"));
            book.setTitle(rs.getString("title"));
            book.setDate(rs.getString("date"));
            book.setAuthor(rs.getString("author"));
        }
    }
}
```



```
        book.setCategory(rs.getString("category"));
        book.setEdit(rs.getString("edit"));
        book.setLang(rs.getString("lang"));
        book.setPages(rs.getString("pages"));
        book.setDescription(rs.getString("description"));
        book.setEjemplares(rs.getString("ejemplares"));
        book.setStock(rs.getInt("stock"));
        book.setAvailable(rs.getInt("available"));
        lista.add(book);
    }
    rs.close();
    st.close();
} catch(Exception e) {
    throw e;
} finally {
    this.Cerrar();
}
return lista;
}

@Override
public Books getBookById(int bookId) throws Exception {
    Books book = null;

    try {
        this.Conectar();
        PreparedStatement st = this.conexion.prepareStatement("SELECT *
FROM books WHERE id = ? LIMIT 1;");
        st.setInt(1, bookId);

        ResultSet rs = st.executeQuery();
        while(rs.next()) {
            book = new Books();
            book.setId(rs.getInt("id"));
            book.setTitle(rs.getString("title"));
            book.setDate(rs.getString("date"));
            book.setAuthor(rs.getString("author"));
            book.setCategory(rs.getString("category"));
            book.setEdit(rs.getString("edit"));
            book.setLang(rs.getString("lang"));
            book.setPages(rs.getString("pages"));
            book.setDescription(rs.getString("description"));
            book.setEjemplares(rs.getString("ejemplares"));
            book.setStock(rs.getInt("stock"));
            book.setAvailable(rs.getInt("available"));
        }
    }
}
```




```
        }  
        rs.close();  
        st.close();  
    } catch (Exception e) {  
        throw e;  
    } finally {  
        this.Cerrar();  
    }  
    return book;  
}  
}
```

Por último, la implementación total de los libros, que es muy parecido a usuarios.

Utilidad

Utils

```
package com.mycompany.utils;  
  
import java.text.DateFormat;  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.Date;  
  
public class Utils {  
  
    public static boolean isNumeric(String strNum) {  
        if (strNum == null) {  
            return false;  
        }  
        try {  
            double d = Integer.parseInt(strNum);  
        } catch (NumberFormatException nfe) {  
            return false;  
        }  
        return true;  
    }  
  
    public static String getFechaActual() {  
        Date ahora = new Date();  
        SimpleDateFormat formateador = new SimpleDateFormat("dd-MM-yyyy");  
        return formateador.format(ahora);  
    }  
}
```



```
//Diferencias entre dos fechas
//@param fechaInicial La fecha de inicio
//@param fechaFinal La fecha de fin
//@return Retorna el numero de dias entre dos fechas
public static synchronized int diferenciasDeFechas(Date fechaInicial,
Date fechaFinal) throws ParseException {

    DateFormat df = DateFormat.getDateInstance(DateFormat.MEDIUM);
    String fechaInicioString = df.format(fechaInicial);
    try {
        fechaInicial = df.parse(fechaInicioString);
    } catch (ParseException ex) {
    }

    String fechaFinalString = df.format(fechaFinal);
    fechaFinal = df.parse(fechaFinalString);

    long fechaInicialMs = fechaInicial.getTime();
    long fechaFinalMs = fechaFinal.getTime();
    long diferencia = fechaFinalMs - fechaInicialMs;
    double dias = Math.floor(diferencia / (1000 * 60 * 60 * 24));
    return ((int) dias);
}

//Devuele un java.util.Date desde un String en formato dd-MM-yyyy
//@param La fecha a convertir a formato date
//@return Retorna la fecha en formato Date
public static synchronized java.util.Date stringToDate(String fecha) {
    SimpleDateFormat formatoDelTexto = new SimpleDateFormat("dd-MM-
yyyy");
    Date fechaEnviar = null;
    try {
        fechaEnviar = formatoDelTexto.parse(fecha);
        return fechaEnviar;
    } catch (ParseException ex) {
        ex.printStackTrace();
        return null;
    }
}
}
```



Estas utilerías, son prácticamente métodos que se pueden utilizar en varias clases del proyecto, esto para evitar duplicar el código y tener una mayor orden en el código y evitar lo mas posible la redundancia.

Dashboard Principal (Codigo)

```
package com.mycompany.libreria;

import com.formdev.flatlaf.FlatLightLaf;
import com.formdev.flatlaf.intellijthemes.materialthemeuilite.FlatMaterialLighterIJ
Theme;
import com.mycompany.views.*;
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Insets;
import java.time.DayOfWeek;
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.util.Locale;
import javax.swing.JPanel;
import javax.swing.UIManager;
import org.netbeans.lib.awtextra.AbsoluteConstraints;

/**
 *
 * @author ricardo_rg
 */
public class Dashboard extends javax.swing.JFrame {

    /**
     * Creates new form Dashboard
     */
    public Dashboard() {
        initComponents();
        InitStyles();
        SetDate();
        InitContent();
    }

    private void InitStyles() {
        mensaje.putClientProperty("FlatLaf.style", "font: 14 $light.font");
        mensaje.setForeground(Color.black);
        navText.putClientProperty("FlatLaf.style", "font: bold
$h3.regular.font");
    }
}
```



```
        navText.setForeground(Color.white);
        dateText.putClientProperty("FlatLaf.style", "font: 24 $light.font");
        dateText.setForeground(Color.white);
        appName.putClientProperty("FlatLaf.style", "font: bold
$h1.regular.font");
        appName.setForeground(Color.white);
    }

    private void SetDate() {
        LocalDate now = LocalDate.now();
        Locale spanishLocale = new Locale("es", "ES");
        dateText.setText(now.format(DateTimeFormatter.ofPattern("'Hoy es'
EEEE dd 'de' MMMM 'de' yyyy", spanishLocale)));
    }

    private void InitContent() {
        ShowJPanel(new Principal());
    }

    public static void ShowJPanel(JPanel p) {
        p.setSize(750, 430);
        p.setLocation(0,0);

        content.removeAll();
        content.add(p, BorderLayout.CENTER);
        content.revalidate();
        content.repaint();
    }

    /**
     * This method is called from within the constructor to initialize the
    form.
     * WARNING: Do NOT modify this code. The content of this method is
    always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    BEGIN: initComponents
    private void initComponents() {

        header = new javax.swing.JPanel();
        navText = new javax.swing.JLabel();
        dateText = new javax.swing.JLabel();
        menu = new javax.swing.JPanel();
    }
```



```
appName = new javax.swing.JLabel();
jSeparator1 = new javax.swing.JSeparator();
btn_prin = new javax.swing.JButton();
btn_lends = new javax.swing.JButton();
btn_returns = new javax.swing.JButton();
btn_users = new javax.swing.JButton();
btn_books = new javax.swing.JButton();
btn_reports = new javax.swing.JButton();
content = new javax.swing.JPanel();
mensaje = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setMinimumSize(new java.awt.Dimension(1020, 670));

header.setBackground(new java.awt.Color(25, 118, 210));
header.setPreferredSize(new java.awt.Dimension(744, 150));

navText.setText("Administración/Control/Librería");

dateText.setText("Hoy es {dayname} {day} de {month} de {year}");

javax.swing.GroupLayout headerLayout = new
javax.swing.GroupLayout(header);
header.setLayout(headerLayout);
headerLayout.setHorizontalGroup(
    headerLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(headerLayout.createSequentialGroup()
            .addGap(66, 66, 66)
            .addGroup(headerLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                .addComponent(navText,
                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
                    Short.MAX_VALUE)
                .addComponent(dateText,
                    javax.swing.GroupLayout.DEFAULT_SIZE, 434, Short.MAX_VALUE))
            .addGap(262, Short.MAX_VALUE))
        );
headerLayout.setVerticalGroup(
    headerLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(headerLayout.createSequentialGroup()
            .addGap(33, 33, 33)
```



```
.addComponent(navText,
javax.swing.GroupLayout.PREFERRED_SIZE, 33,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(18, 18, 18)
    .addComponent(dateText,
javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addContainerGap(31, Short.MAX_VALUE))
);

menu.setBackground(new java.awt.Color(13, 71, 161));
menu.setPreferredSize(new java.awt.Dimension(270, 640));

appName.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
appName.setText("Libreria");

jSeparator1.setPreferredSize(new java.awt.Dimension(50, 5));

btn_prin.setBackground(new java.awt.Color(21, 101, 192));
btn_prin.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
btn_prin.setForeground(new java.awt.Color(255, 255, 255));
btn_prin.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/home-outline.png"))); //
NOI18N
    btn_prin.setText("Principal");
    btn_prin.setBorder(javax.swing.BorderFactory.createMatteBorder(1,
13, 1, 1, new java.awt.Color(0, 0, 0)));
    btn_prin.setBorderPainted(false);
    btn_prin.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
    btn_prin.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
    btn_prin.setIconTextGap(13);
    btn_prin.setInheritsPopupMenu(true);
    btn_prin.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btn_prinActionPerformed(evt);
        }
    });

btn_lends.setBackground(new java.awt.Color(21, 101, 192));
btn_lends.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
btn_lends.setForeground(new java.awt.Color(255, 255, 255));
btn_lends.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/calendar-plus.png"))); //
NOI18N
```



```
        btn_lends.setText("Préstamos");
        btn_lends.setBorder(javax.swing.BorderFactory.createMatteBorder(1,
13, 1, 1, new java.awt.Color(0, 0, 0)));
        btn_lends.setBorderPainted(false);
        btn_lends.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        btn_lends.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
        btn_lends.setIconTextGap(13);
        btn_lends.setInheritsPopupMenu(true);
        btn_lends.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btn_lendsActionPerformed(evt);
            }
        });

        btn_returns.setBackground(new java.awt.Color(21, 101, 192));
        btn_returns.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
        btn_returns.setForeground(new java.awt.Color(255, 255, 255));
        btn_returns.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/calendar-multiple-
check.png"))); // NOI18N
        btn_returns.setText("Devoluciones");
        btn_returns.setBorder(javax.swing.BorderFactory.createMatteBorder(1,
13, 1, 1, new java.awt.Color(0, 0, 0)));
        btn_returns.setBorderPainted(false);
        btn_returns.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        btn_returns.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
        btn_returns.setIconTextGap(13);
        btn_returns.setInheritsPopupMenu(true);
        btn_returns.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btn_returnsActionPerformed(evt);
            }
        });

        btn_users.setBackground(new java.awt.Color(21, 101, 192));
        btn_users.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
        btn_users.setForeground(new java.awt.Color(255, 255, 255));
        btn_users.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/account-multiple.png"))); //
NOI18N
        btn_users.setText("Usuarios");
        btn_users.setBorder(javax.swing.BorderFactory.createMatteBorder(1,
13, 1, 1, new java.awt.Color(0, 0, 0)));
```



```
        btn_users.setBorderPainted(false);
        btn_users.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        btn_users.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
        btn_users.setIconTextGap(13);
        btn_users.setInheritsPopupMenu(true);
        btn_users.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btn_usersActionPerformed(evt);
            }
        });

        btn_books.setBackground(new java.awt.Color(21, 101, 192));
        btn_books.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
        btn_books.setForeground(new java.awt.Color(255, 255, 255));
        btn_books.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/book-open-page-
variant.png"))); // NOI18N
        btn_books.setText("Libros");
        btn_books.setBorder(javax.swing.BorderFactory.createMatteBorder(1,
13, 1, 1, new java.awt.Color(0, 0, 0)));
        btn_books.setBorderPainted(false);
        btn_books.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        btn_books.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
        btn_books.setIconTextGap(13);
        btn_books.setInheritsPopupMenu(true);
        btn_books.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btn_booksActionPerformed(evt);
            }
        });

        btn_reports.setBackground(new java.awt.Color(21, 101, 192));
        btn_reports.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
        btn_reports.setForeground(new java.awt.Color(255, 255, 255));
        btn_reports.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/file-chart.png"))); // NOI18N
        btn_reports.setText("Reportes");
        btn_reports.setBorder(javax.swing.BorderFactory.createMatteBorder(1,
13, 1, 1, new java.awt.Color(0, 0, 0)));
        btn_reports.setBorderPainted(false);
        btn_reports.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        btn_reports.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
```




```
btn_reports.setIconTextGap(13);
btn_reports.setInheritsPopupMenu(true);
btn_reports.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btn_reportsActionPerformed(evt);
    }
});

javax.swing.GroupLayout menuLayout = new
javax.swing.GroupLayout(menu);
menu.setLayout(menuLayout);
menuLayout.setHorizontalGroup(
    menuLayout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
        .addGroup(menuLayout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addComponent(appName,
javax.swing.GroupLayout.PREFERRED_SIZE, 250,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(menuLayout.createSequentialGroup()
                .addGap(40, 40, 40)
                .addComponent(jSeparator1,
javax.swing.GroupLayout.PREFERRED_SIZE, 190,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(btn_lends, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(btn_users, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(btn_prin, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(btn_returns, javax.swing.GroupLayout.DEFAULT_SIZE,
270, Short.MAX_VALUE)
                .addComponent(btn_books, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(btn_reports, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            );
menuLayout.setVerticalGroup(
    menuLayout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
        .addGroup(menuLayout.createSequentialGroup()
            .addGap(52, 52, 52)
            .addComponent(appName,
javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)
```



```
.addGap(4, 4, 4)
.addComponent(jSeparator1,
javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addGap(20, 20, 20)
.addGroup(menuLayout.createParallelGroup(javax.swing.GroupLa
yout.Alignment.LEADING)
.addGroup(menuLayout.createSequentialGroup())
.addGap(50, 50, 50)
.addComponent(btn_lends,
javax.swing.GroupLayout.PREFERRED_SIZE, 52,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addGroup(menuLayout.createSequentialGroup())
.addGap(150, 150, 150)
.addComponent(btn_users,
javax.swing.GroupLayout.PREFERRED_SIZE, 52,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addComponent(btn_prin,
javax.swing.GroupLayout.PREFERRED_SIZE, 52,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addGroup(menuLayout.createSequentialGroup())
.addGap(100, 100, 100)
.addComponent(btn_returns,
javax.swing.GroupLayout.PREFERRED_SIZE, 52,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addGroup(menuLayout.createSequentialGroup())
.addGap(200, 200, 200)
.addComponent(btn_books,
javax.swing.GroupLayout.PREFERRED_SIZE, 52,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addGroup(menuLayout.createSequentialGroup())
.addGap(250, 250, 250)
.addComponent(btn_reports,
javax.swing.GroupLayout.PREFERRED_SIZE, 52,
javax.swing.GroupLayout.PREFERRED_SIZE))))
);

content.setBackground(new java.awt.Color(255, 255, 255));
content.setLayout(new java.awt.BorderLayout());

mensaje.setText("Bienvenido al sistema de prestamos de la Libreria
Fray servando Teresa de mier!");

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
```



```
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(menu, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(header,
                    javax.swing.GroupLayout.DEFAULT_SIZE, 762, Short.MAX_VALUE)
                .addComponent(content,
                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
                    Short.MAX_VALUE)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(layout.createSequentialGroup()
                        .addGap(10, 10, 10)
                        .addComponent(mensaje,
                            javax.swing.GroupLayout.PREFERRED_SIZE, 501,
                            javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addContainerGap()))
                    .addGap(10, 10, 10)
                ));
        .addGroup(layout.createSequentialGroup()
            .addComponent(menu, javax.swing.GroupLayout.DEFAULT_SIZE, 670,
                Short.MAX_VALUE)
            .addGroup(layout.createSequentialGroup()
                .addGap(27, 27, 27)
                .addComponent(mensaje,
                    javax.swing.GroupLayout.PREFERRED_SIZE, 26,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(6, 6, 6)
                .addComponent(header,
                    javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(0, 0, 0)
                .addComponent(content, javax.swing.GroupLayout.DEFAULT_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addGap(1, 1, 1))
            .addGap(1, 1, 1)
        ));
pack();
setLocationRelativeTo(null);
```



```
    }// </editor-fold>//GEN-END:initComponents

    private void btn_prinActionPerformed(java.awt.event.ActionEvent evt)
    {//GEN-FIRST:event_btn_prinActionPerformed
        ShowJPanel(new Principal());
    }//GEN-LAST:event_btn_prinActionPerformed

    private void btn_lendsActionPerformed(java.awt.event.ActionEvent evt)
    {//GEN-FIRST:event_btn_lendsActionPerformed
        ShowJPanel(new Lendings());
    }//GEN-LAST:event_btn_lendsActionPerformed

    private void btn_returnsActionPerformed(java.awt.event.ActionEvent evt)
    {//GEN-FIRST:event_btn_returnsActionPerformed
        ShowJPanel(new Returns());
    }//GEN-LAST:event_btn_returnsActionPerformed

    private void btn_usersActionPerformed(java.awt.event.ActionEvent evt)
    {//GEN-FIRST:event_btn_usersActionPerformed
        ShowJPanel(new Users());
    }//GEN-LAST:event_btn_usersActionPerformed

    private void btn_booksActionPerformed(java.awt.event.ActionEvent evt)
    {//GEN-FIRST:event_btn_booksActionPerformed
        ShowJPanel(new Books());
    }//GEN-LAST:event_btn_booksActionPerformed

    private void btn_reportsActionPerformed(java.awt.event.ActionEvent evt)
    {//GEN-FIRST:event_btn_reportsActionPerformed
        ShowJPanel(new Reports());
    }//GEN-LAST:event_btn_reportsActionPerformed

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        FlatMaterialLighterIJTheme.setup();

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Dashboard().setVisible(true);
            }
        });
    }
}
```



```
}  
  
// Variables declaration - do not modify//GEN-BEGIN:variables  
private javax.swing.JLabel appName;  
private javax.swing.JButton btn_books;  
private javax.swing.JButton btn_lends;  
private javax.swing.JButton btn_prin;  
private javax.swing.JButton btn_reports;  
private javax.swing.JButton btn_returns;  
private javax.swing.JButton btn_users;  
private static javax.swing.JPanel content;  
private javax.swing.JLabel dateText;  
private javax.swing.JPanel header;  
private javax.swing.JSeparator jSeparator1;  
private javax.swing.JLabel mensaje;  
private javax.swing.JPanel menu;  
private javax.swing.JLabel navText;  
// End of variables declaration//GEN-END:variables  
}
```

Aquí está la mayoría de la funcionalidad del código ya que en este dashboard es donde se esta ejecutando en todo momento los diferentes menús que tenemos (views).

Vistas (views).



Books.java



Returns.java



Reports.java



UpBooks.java



Users.java



UpUsers.java



Principal.java



Lendings.java

Aquí están todas las vistas en código de todo el programa, no las pongo en forma de código para no alargar más el mismo documento.



Conclusión

Como conclusión, todavía falta mucho trabajo por recorrer, falta implementar todo el sistema de multas e implementar el sistema con el sistema de cobros que ya tienen ahí implementado,

Faltan aun muchos pasos para recorrer para que el programa fuera perfecto, pero ahora mismo es funcional y estamos por implementarlo. El fuerte de este sistema es que puede ser implementado no solamente para un sistema de librería, ahora se tiene una pauta a seguir al momento de poder hacer un punto de venta diferente, o cualquier sistema de inventario que pueda ser útil para alguna empresa o local, en un nicho más pequeño de locales donde aun no han podido ser alcanzadas por la modernización y esto tiene una fácil implementación para todos estos lugares.

Cabe resaltar, sobre todo, que me apoyé bastante de la ayuda que nos brinda Chat GPT para poder solucionar problemas y determinar cuales serían las mejores tecnologías para implementar en este proyecto, y aprovechando esta nueva inteligencia artificial que salió a relucir este año pude mejorar bastante la calidad de mi proyecto a comparación de proyectos realizados anteriormente.

Referencias bibliográficas

Flatlaf Maven: (Software, 2023)

Chat GPT: <https://openai.com/blog/chatgpt>