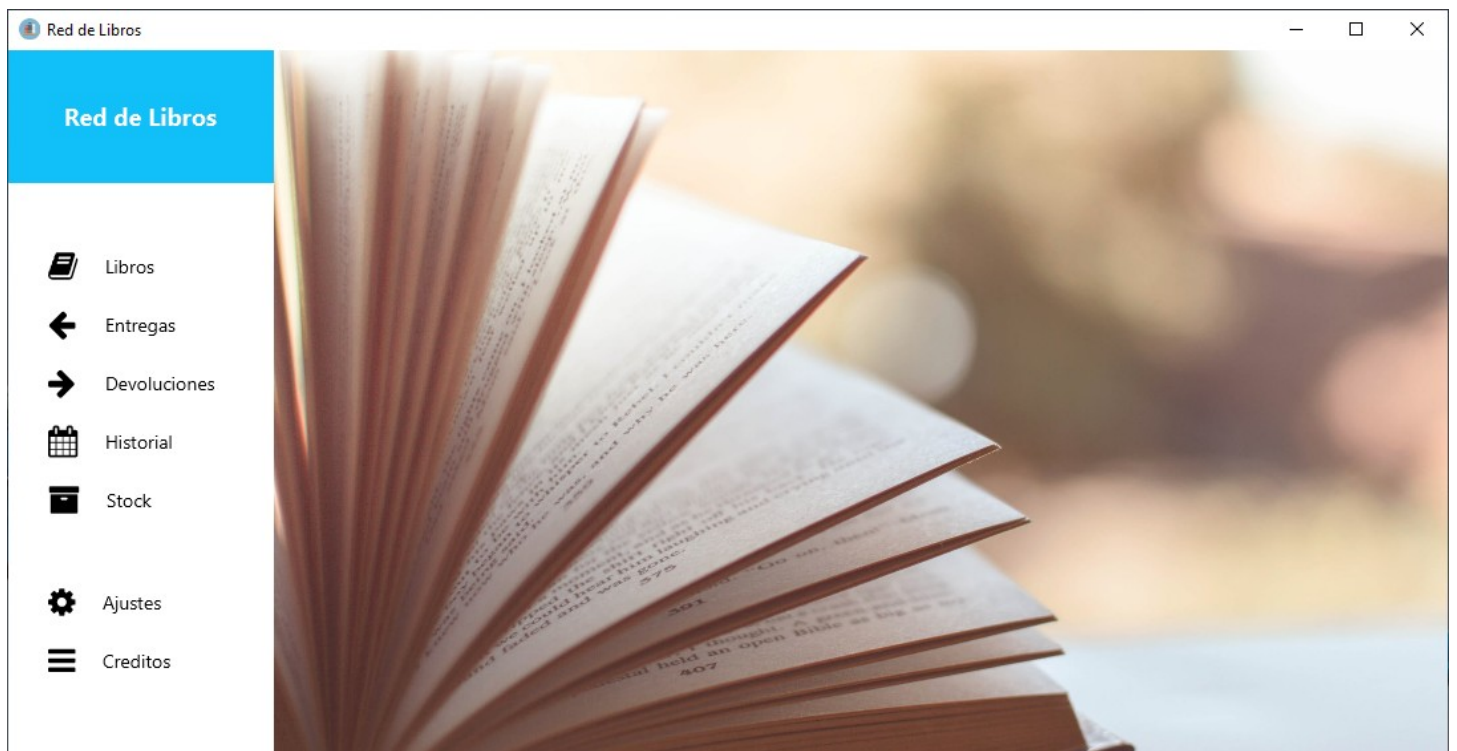


Proyecto de final de ciclo de
Desarrollo de Aplicaciones Multiplataforma

Red de Libros



Jose Galán Simó
Raúl Boix Palomares

Índice

1. Descripción del proyecto
2. Diagramas
 - ➔ Diagrama E/R
 - ➔ Diagrama de casos de uso
 - ➔ Diagrama de la base de datos
 - ➔ Diagrama temporal
3. Tecnologías usadas
4. Desarrollo de la aplicación
5. Dificultades encontradas
6. Propuesta de mejoras

1. Descripción del proyecto

Red de Libros es una aplicación para gestionar y tener informatizado los libros en un centro educativo, en especial, los libros que se le entregan a los alumnos.

En la aplicación puedes tener un control de ejemplares de libros que tiene el centro, los que ya has entregado a los alumnos y los que tienes para entregar, además cuenta con un historial en el que puedes consultar fechas de entrega o devolución, estado de los ejemplares y que alumno lo posee actualmente, también cuenta con un control de stock de libros, donde puedes consultar cuantos ejemplares necesitas para el curso, en base a los alumnos matriculados en esa asignatura.

Los libros en la aplicación están relacionados con su respectiva asignatura, la cual está relacionada con los alumnos matriculados a ella, para así tener el control de los ejemplares que se van a necesitar y el listado de alumnos para el control de ejemplares.

Una de las herramientas más útiles de la aplicación, es su sistema de códigos de barras, en el que cada ejemplar tiene un código de barras único, estos códigos de barras se pueden imprimir para que, por ejemplo se le vaya a entregar a un alumno, pasando el código por el programa y así encontrar el ejemplar más rápido, gracias a este sistema se puede ahorrar mucho tiempo al entregar o devolver ejemplares.

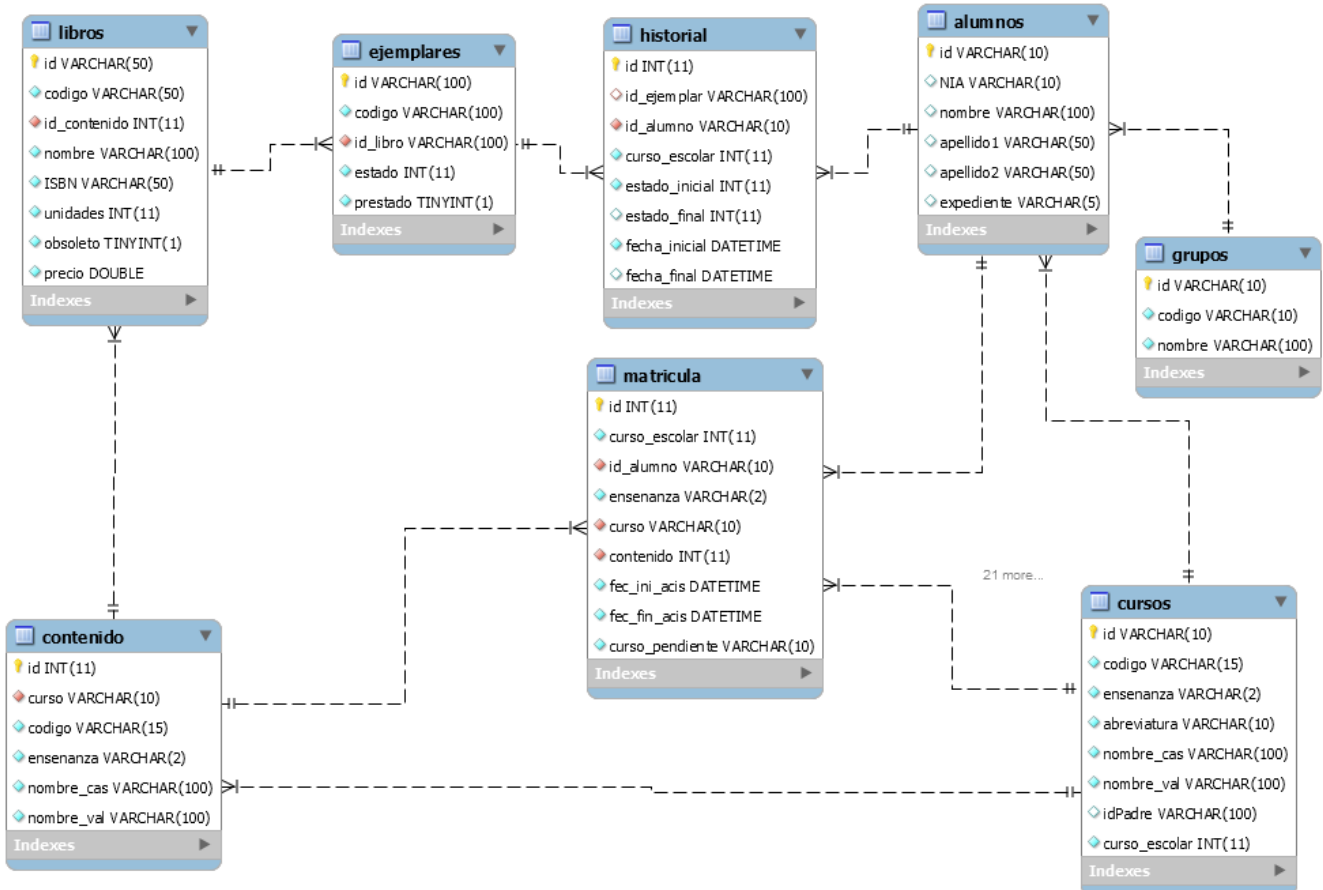
Para imprimir los códigos de barras, podrás seleccionar cuales quieres imprimir y en que posición de la página (fila y columna), así como la distribución de los códigos en la página, así se puede ahorrar papel, utilizando el máximo espacio de las hojas posible.

Por supuesto, el diseño de la aplicación es completamente adaptable, para así poder trabajar de la manera que más te guste: en ventana o a pantalla completa. Tiene un tamaño mínimo para que quepan todos los elementos en la ventana, pero suficiente para que quepa en cualquier resolución de pantalla.

La información de la base de datos como alumnos, cursos o asignaturas, se puede importar a través de archivos XML.

2. Diagramas

- Diagrama E/R



*. la tabla **alumnos** no está completa para ahorrar espacio, ya que tiene muchos campos

- Diagrama de casos de uso

Diagrama de casos de uso 1

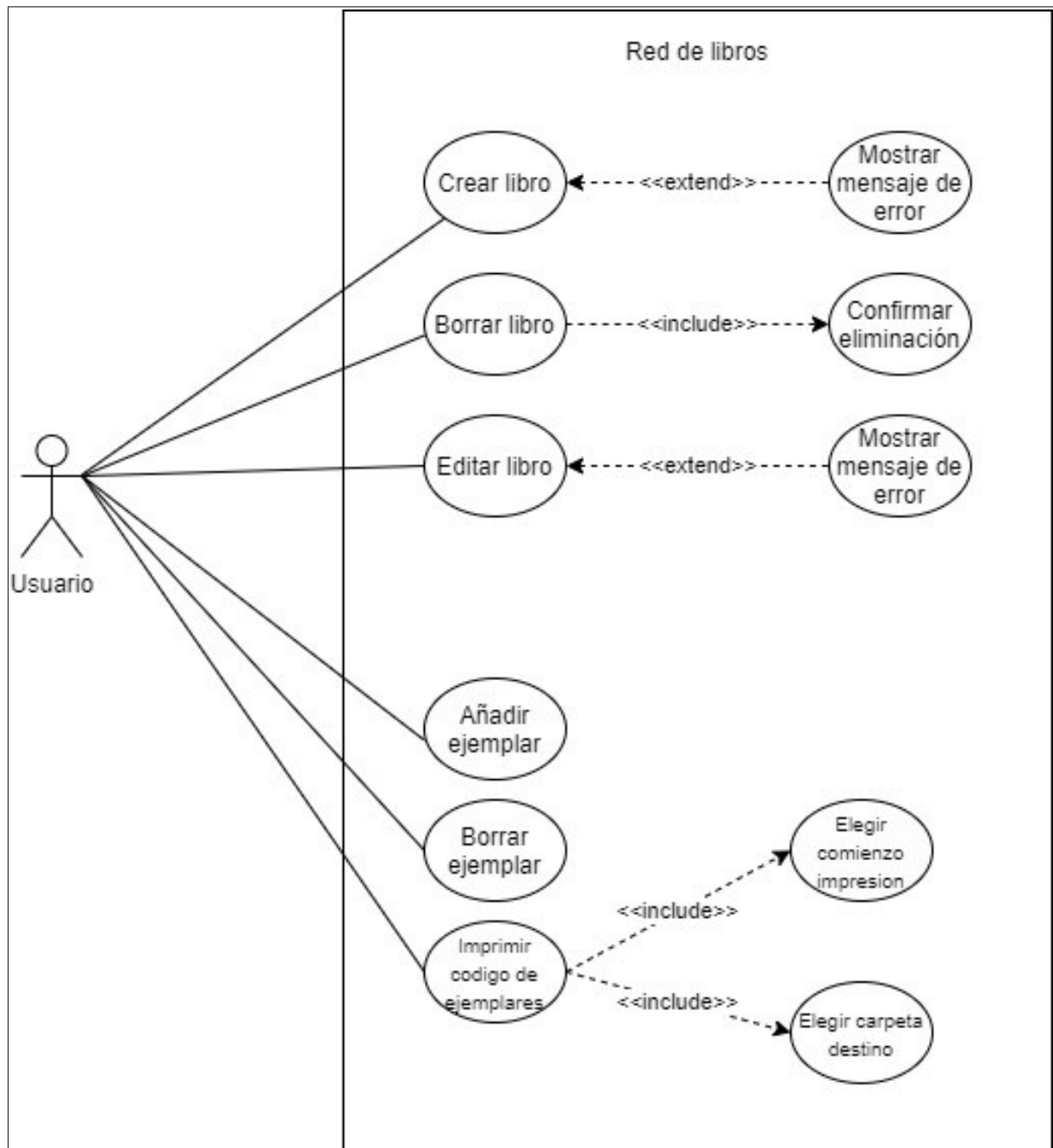
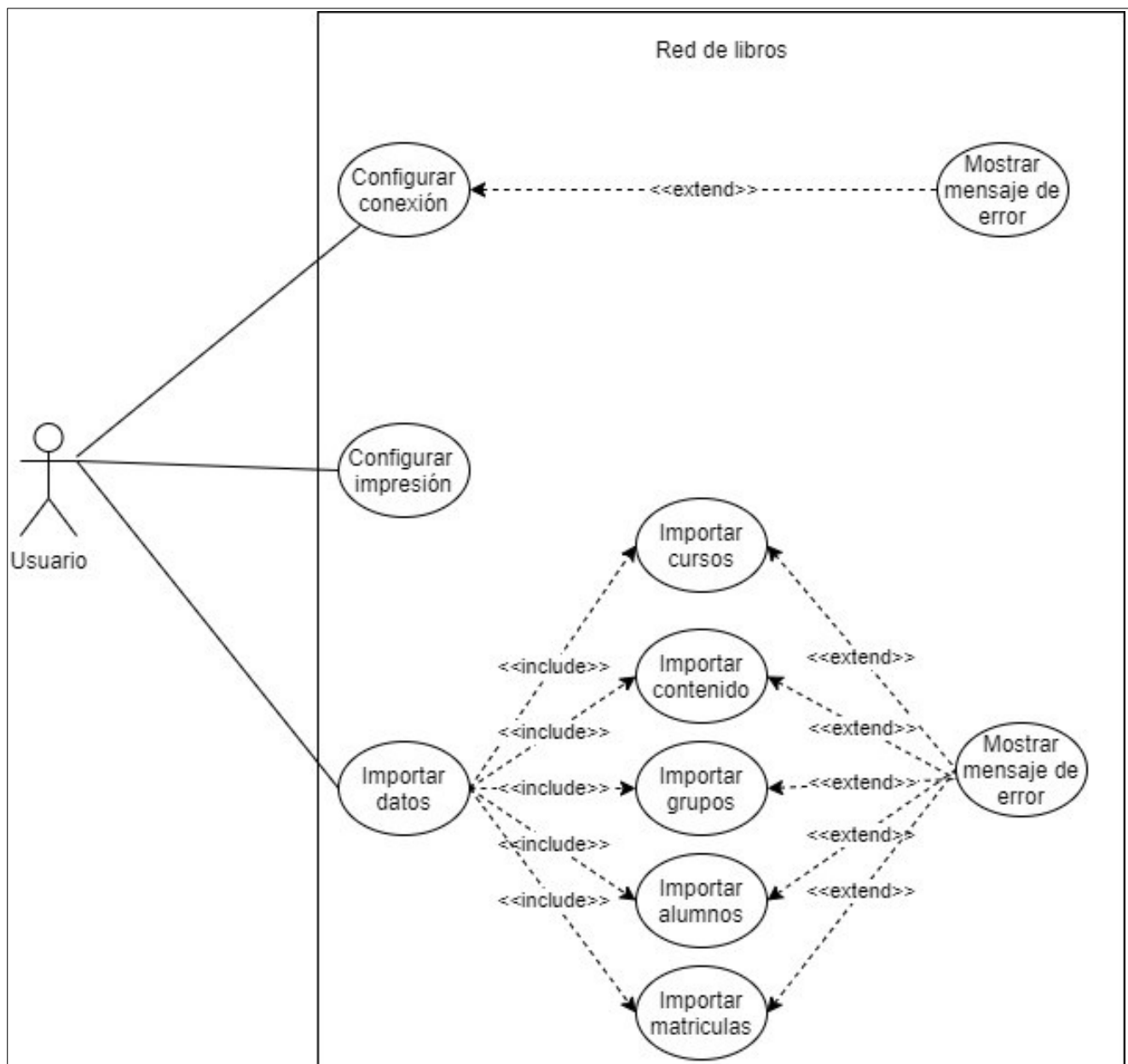


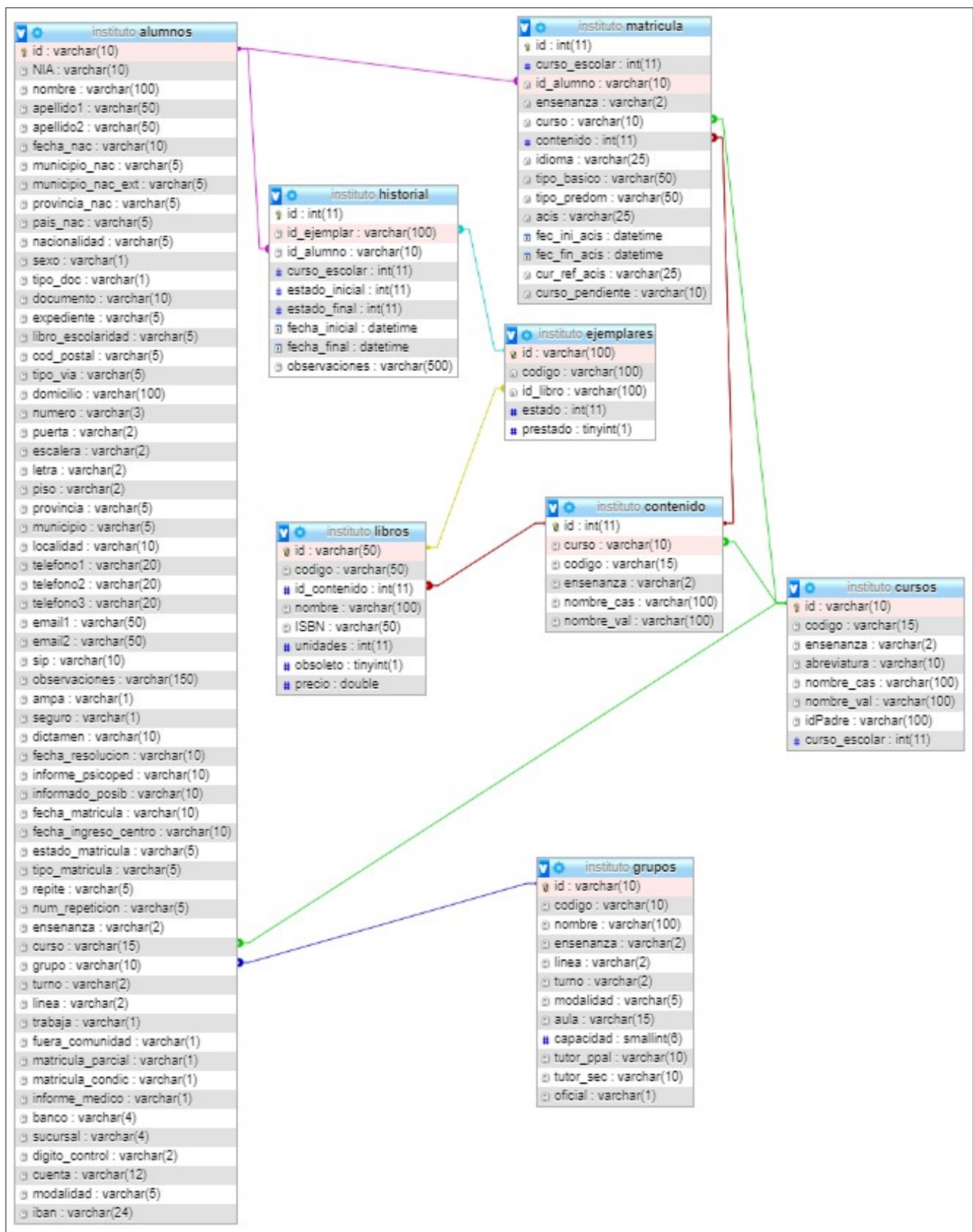
Diagrama de casos de uso 2



Diagrama de casos de uso 3



- Diagrama de base de datos



- Diagrama temporal

Actividad	Inicio	Final	09/03/2020	10/03/2020	11/03/2020	12/03/2020	13/03/2020	14/03/2020	15/03/2020	16/03/2020	17/03/2020	18/03/2020	19/03/2020	20/03/2020	21/03/2020	22/03/2020	23/03/2020	24/03/2020	25/03/2020	26/03/2020	27/03/2020
Diseño de la base de datos	09/03/2020	09/03/2020																			
Diagrama E/R	09/03/2020	09/03/2020																			
Base de datos	09/03/2020	-																			
Vista principal	11/03/2020	24/03/2020																			
Diseño de interfaz	10/03/2020	-																			
Creditos	12/03/2020	12/03/2020																			
Ajustes	12/03/2020	-																			
Pojos	14/03/2020	25/03/2020																			
Hibernate	15/03/2020	-																			
Libros y ejemplares	14/03/2020	20/05/2020																			
Manejo de errores	11/03/2020	-																			
DAOs	27/03/2020	-																			

Actividad	Inicio	Final	07/04/2020	08/04/2020	09/04/2020	10/04/2020	11/04/2020	12/04/2020	13/04/2020	14/04/2020	15/04/2020	16/04/2020	17/04/2020	18/04/2020	19/04/2020	20/04/2020	21/04/2020	22/04/2020	23/04/2020	24/04/2020	25/04/2020
Diseño de interfaz	-	-																			
Manejo de errores	-	-																			
DAOs	-	-																			
Base de datos	-	16/04/2020																			
Hibernate	-	25/04/2020																			
Barcodes y PDF	25/04/2020	-																			

Actividad	Inicio	Final	27/04/2020	28/04/2020	29/04/2020	30/04/2020	01/05/2020	02/05/2020	03/05/2020	04/05/2020	05/05/2020	06/05/2020	07/05/2020	08/05/2020	09/05/2020	10/05/2020	11/05/2020	12/05/2020	13/05/2020	14/05/2020	15/05/2020
Diseño de interfaz	-	-																			
Manejo de errores	-	-																			
DAOs	-	15/05/2020																			
Barcodes y PDF	-	-																			
Libros y ejemplares	-	-																			

Actividad	Inicio	Final	16/05/2020	17/05/2020	18/05/2020	19/05/2020	20/05/2020	21/05/2020	22/05/2020	23/05/2020	24/05/2020	25/05/2020	26/05/2020	27/05/2020	28/05/2020	29/05/2020	30/05/2020	31/05/2020	01/06/2020	02/06/2020	03/06/2020
Diseño de interfaz	-	23/05/2020																			
Manejo de errores	-	23/05/2020																			
Historial	-	18/05/2020																			
Barcodes y PDF	-	23/05/2020																			
Libros y ejemplares	-	20/05/2020																			
Bugs	21/05/2020	23/05/2020																			
Devoluciones	-	24/05/2020																			
Ajustes de complicacion	24/05/2020	26/05/2020																			

3. Tecnologías usadas

Java	JavaFX	CSS	MYSQL	Hibernate
				

Java 11

Hemos utilizado **Java** como lenguaje de programación, ya que es el lenguaje que más experiencia tenemos y mas cómodo es para nosotros trabajar con el, con java, hemos podido hacer uso de **JavaFX** para desarrollar la parte gráfica. Para programar utilizamos Eclipse IDE, en su versión 2019-12.

Sobre la versión, elegimos **Java 11** por que actualmente es la versión actual más usada y la que tiene más soporte (hasta 2026).

Para el proyecto hemos usado el patrón de diseño modelo-vista-controlador (MVC) para separar los ficheros y tenerlo todo organizado y accesible.

Además nos hemos apoyado de archivos DAO (Objeto de acceso a datos) para tener separado del resto de código la comunicación con la base de datos, que es a través de hibernate.

También hemos usado archivos Service, para la ejecución de código específico como puede ser imprimir o importar.

JavaFX y CSS

Para desarrollar la interfaz gráfica utilizamos **JavaFX** a pesar de que en el curso mayoritariamente estudiamos Swing, hemos preferido utilizar este porque tiene mayor potencial y mas facilidad de uso.

Con **JavaFX**, hemos hecho uso de **CSS**, que realmente no es para nada necesario pero nosotros decidimos darle a la interfaz gráfica un estilo “Material Design” para el cual necesitábamos editar los estilos de con **CSS**.

Para diseñar con **JavaFX** utilizamos la herramienta SceneBuilder en su versión 11.0.

MySQL

En la base de datos elegimos **MySQL**, en este tipo de aplicación necesitábamos una base de datos relacional y tenemos ya experiencia con **MySQL** con lo cual nos venia perfecto.

Para utilizar **MySQL** usamos varias formas como puede ser MySQL Workbench o phpMyAdmin.

Hibernate

Para la conexión con la base de datos usamos **Hibernate**, nos parece una tecnología con un gran potencial y de la que hemos sacado provecho para facilitar el desarrollo de la aplicación.

Además tenemos experiencia previa y así hemos combinado el uso de hibernate con archivos DAO para una mayor organización del proyecto.

4. Desarrollo de la aplicación

Base de datos

El desarrollo de la aplicación empezó con el diseño de la base de datos, su diagrama E/R y su implementación en MySQL ya que necesitábamos datos específicos como las columnas de la tabla alumnos que nos daría el tutor.

Para empezar a desarrollar la aplicación tuvimos que elegir que tecnología usar para diseñar las vistas, y decidimos usar JavaFX por su facilidad de uso y comodidad al crear los controladores.

Vista y diseño

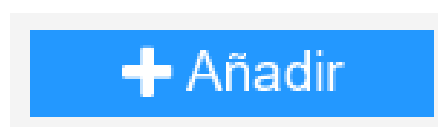
Empezamos a diseñar la vista principal, nosotros queríamos un tipo de aplicación en el que siempre estas en la misma ventana, y por eso JavaFX nos permitía de manera muy cómoda el cambio de “componentes” dentro de la misma ventana. También nosotros queríamos un estilo diferente a las ventanas convencionales de Windows y decidimos aplicar a través de CSS un estilo simulando “Material Design” para que quede mucho mas bonito.

Por ejemplo: los ‘botones’ de la aplicación, no son botones como tal, si no que son “contenedores” con propiedades especiales (color, efectos...).

Estilo normal

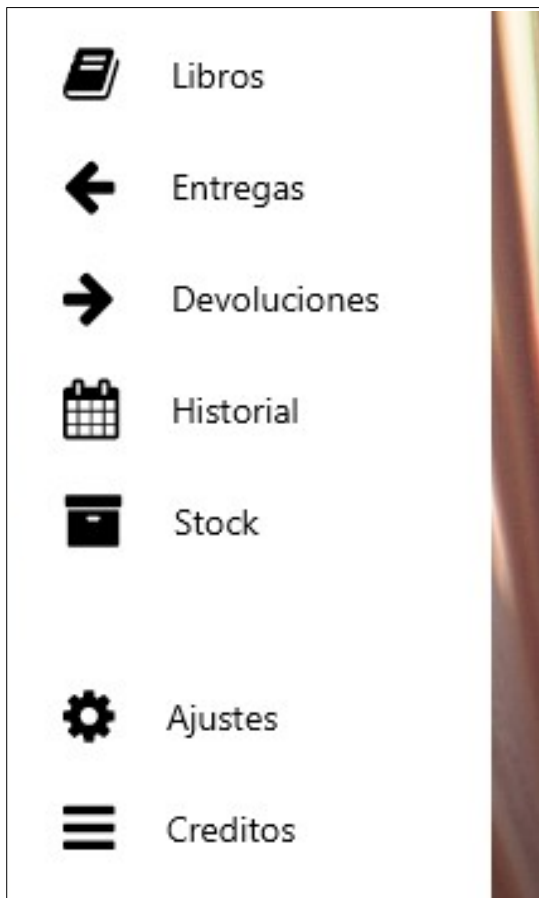


Estilo personalizado



Estos cambios de estilo también están en elementos como listas de columnas, listas en árbol, campos de texto, menús...

Menú principal



Una vez ya teníamos en mente el aspecto que queríamos teníamos que diseñar el menú principal, decidimos ponerlo siempre a la vista, en la izquierda de la aplicación para no tener incomodidades al navegar entre páginas.

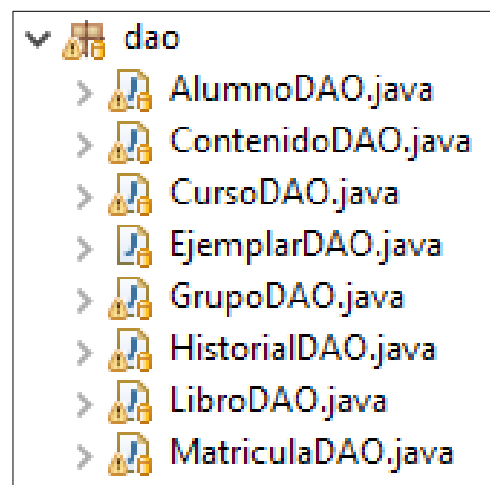
De esta forma el usuario final en cualquier momento puede ir a otro sitio rápidamente.

Este diseño está inspirado por aplicaciones o páginas como Discord, Telegram , WhatsApp, Youtube y muchas mas...

Estructura del proyecto

Para organizar el proyecto seguimos el patrón de diseño modelo-vista-controlador (MVC), para así separar todos los ficheros y tenerlo mas organizado.

Por otra parte decidimos utilizar hibernate, en su versión 5.1 para la conexión con la base de datos, además decidimos organizar el código de hibernate, en archivos DAO (Data Access Object), de esta manera tenemos una estructura mas organizada y métodos más cortos y simples.



Ajustes

La primera página que decidimos incorporar fue la de Ajustes, ya que esta no requiere información de la base de datos para iniciarse.

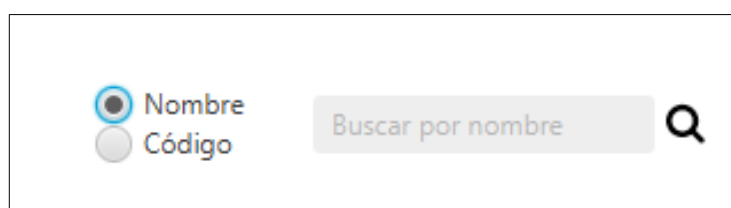


La estructura de ficheros de esta pagina sería un archivo .FXML (vista de JavaFX), su controlador y algunas clases java que hemos denominado “Service”, porque prestan un servicio, en este caso guardar la información en JSON o la importación de datos.

Libros

Para el componente de Libros, implementamos un TreeView (una lista) para que te liste los libros que hay en la base de datos.

También añadimos filtros de búsqueda para encontrar más rápido los libros, por nombre o por código, que puedes cambiar a través de un radiobutton.



Jose Galán Simó
Raúl Boix Palomares

Cuando haces click en un libro se muestra un detalle de este libro, donde ya puedes ver toda su información y los ejemplares que tiene. El detalle tiene su respectivo controlador que, reutilizamos al igual que la vista para visualizar el detalle, editar información y añadir un nuevo libro.

The screenshot shows a web application window titled 'Red de Libros'. On the left is a sidebar with navigation links: Libros, Entregas, Devoluciones, Historial, Stock, Ajustes, and Creditos. The main content area displays the details for a book named 'Programacion'. It includes fields for ISBN (12334), Código (PRO), Curso escolar (2018), Curso (1CFS - Primero), and Asignatura (0485 - Programación). The price is 0.0, and there is an 'Obsoleto' checkbox. Below this is a table of 'Ejemplares' (copies) with columns for 'Codigo', 'Prestado', and 'Estado'. The table shows two copies: PRO001 (No prestado, Regular) and PRO006 (No prestado, Perfecto). To the right of the table is a barcode for PRO001. At the bottom of the table area, it says 'Unidades totales: 2' and has buttons for 'Imprimir Selección', '+', and a trash icon. At the very bottom of the window are buttons for 'Editar', 'Imprimir', 'Guardar', and 'Borrar'.

Codigo	Prestado	Estado
PRO001	No prestado	Regular
PRO006	No prestado	Perfecto

La vista de ejemplares, el sistema de códigos y imprimir fue lo ultimo que añadimos.

Por temas de seguridad, decidimos poner una ventana de confirmación al borrar un libro, ya que si borras este, también se borrarán sus ejemplares y si algún alumno lo prestado, también se le eliminará.

A confirmation dialog box with a light gray background. It contains the text '¿Estás seguro?' in a large, bold font, followed by '(Se borrarán todos los ejemplares de este libro)' in a smaller font. At the bottom, there are two buttons: a blue button with a left arrow icon and the text 'Cancelar', and a red button with a trash icon and the text 'Borrar'.

Entregas y Devoluciones

La página de Entregas y Devoluciones siguen el mismo principio, una lista de alumnos que puedes filtrar, por curso, por expediente y por NIA.

Entregas				
Filtrar por curso:			<input checked="" type="radio"/> NIA <input type="radio"/> Expediente	
Año <input type="text"/> Curso <input type="text"/>			Buscar por NIA <input type="text"/>	
Primer apellido	Segundo apellido	Nombre	NIA	Expediente
GALÁN	SIMÓ	JOSE ANTONIO	11472312	3334
BOIX	PALOMARES	RAÚL	11472312	3334

El código de el filtro de curso es un tanto especial respecto a los otros porque tienes que llamar a otro DAO que te liste los años y cursos y luego compararlos.

El detalle es lo que cambia, en entregas disponemos de dos tablas, a la derecha los libros disponibles y a la izquierda los libros entregados, decidimos que esta es la mejor manera de representarlo y la mas intuitiva.

Al final decidimos añadir títulos arriba de las tablas para mas claridad.

Red de Libros

Libros

Entregas

Devoluciones

Historial

Stock

Ajustes

Creditos

Nombre: JOSE ANTONIO GALÁN SIMÓ

NIA: 11472312

Expediente: 3334

Libros entregados

Asignatura	Ejemplar	Fecha Inicial
Programación	PRO001	2020-06-02 13:04...

Libros disponibles

Escane el código del ejemplar

Bases de Datos

Sistemas informáticos

Tutoría Primero

Inglés Técnico I-S / Horario reservado para la docencia en inglés

Entornos de desarrollo

Programación

Formación y orientación laboral

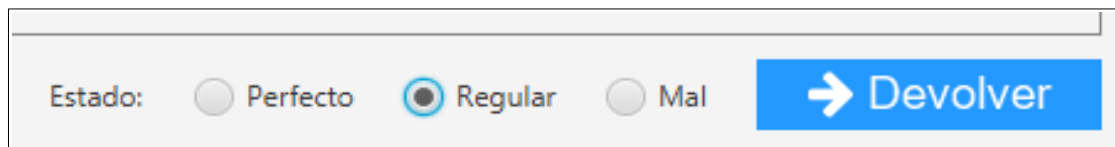
Lenguajes de marcas y sistemas de gestión de información

Entregar

Jose Galán Simó
Raúl Boix Palomares

Hay un TextField arriba de la tabla de Libros disponibles, aquí es donde puedes escanear los códigos y solo con pulsar Enter o el icono de la lupa, si el libro está disponible se entregara automáticamente, si no, saldrá su respectivo mensaje de error.

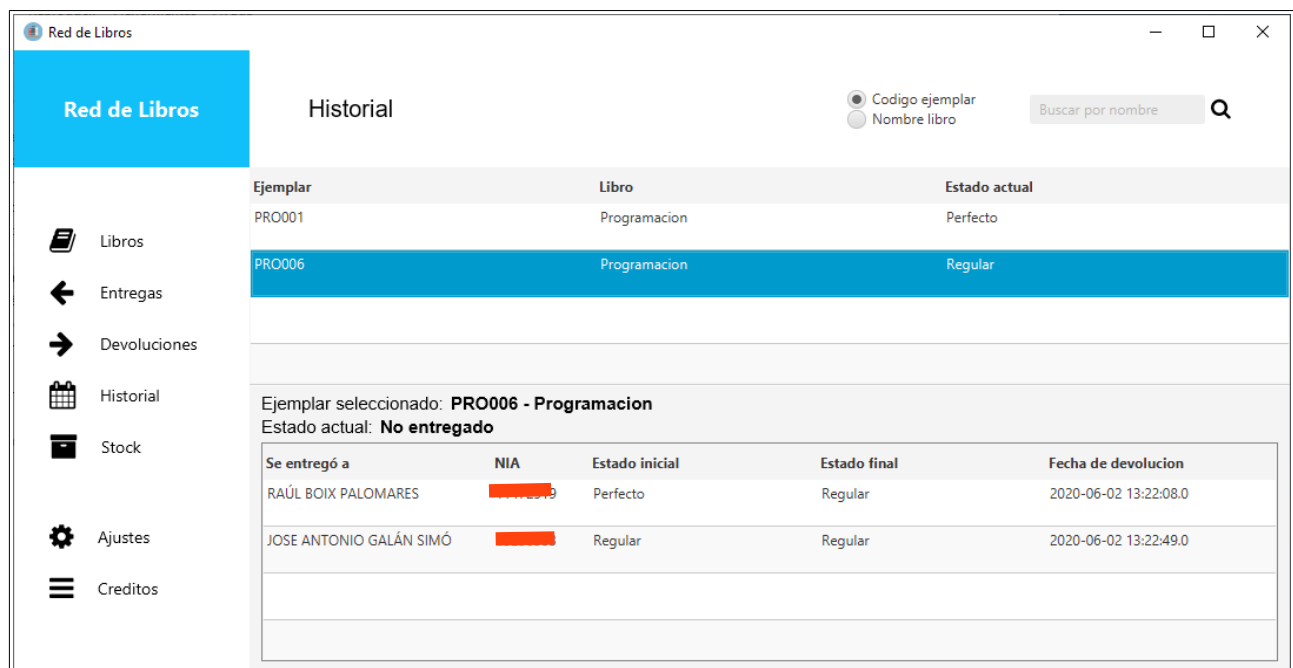
En devoluciones, no es muy diferente, simplemente es una tabla con los libros que tiene entregados y un botón de devolver, pero antes tienes que determinar el estado en el que esta el libro al devolverlo a través de un “radiobutton”, si no pulsas nada, por defecto está “Perfecto”.



Estado: ☐ Perfecto ☒ Regular ☐ Mal

Al igual que en entregas, hay un TextField donde escaneando el código se puede devolver, pulsando Enter o con el botón de la lupa.

Historial



Ejemplar	Libro	Estado actual
PRO001	Programacion	Perfecto
PRO006	Programacion	Regular

Ejemplar seleccionado: PRO006 - Programacion
Estado actual: No entregado

Se entregó a	NIA	Estado inicial	Estado final	Fecha de devolucion
RAÚL BOIX PALOMARES	[REDACTED]	Perfecto	Regular	2020-06-02 13:22:08.0
JOSE ANTONIO GALÁN SIMÓ	[REDACTED]	Regular	Regular	2020-06-02 13:22:49.0

Historial, es una lista de ejemplares, donde al hacer click te muestra un detalle, pero sin cambiar de página de los registros.

Jose Galán Simó
Raúl Boix Palomares

En un principio no estaba estructurado igual, pero la conclusión final fue esta, la cual es muy intuitiva y simple de usar.

Stock

Stock	
Codigo	Nombre
BBDD	Bases de datos
PRO	Programacion
Ejemplares totales: 4	
Ejemplares que necesitamos (segun matricula): 2	
Ejemplares entregados: 1	
Ejemplares no entregados: 3	

En stock hay una lista de libros, pero sigue una interfaz parecida a la de historial, donde en la misma página puedes ver el detalle.

Puedes sacar información muy útil como los libros que necesitas para este curso, según la matricula, entre otras cosas.

El calculo de los campos, son simples calculos hechos a partir de contar dentro de un For, y con comprobaciones con If.

```
int ejemplaresEntregados = 0;
int ejemplaresNoEntregados = 0;

for (int i = 0; i < libroSeleccionado.getEjemplares().size(); i++) {
    if(libroSeleccionado.getEjemplares().get(i).getPrestado()==1) {
        ejemplaresEntregados++;
    }
    ejemplaresNoEntregados++;
}
ejemplaresNoEntregados = ejemplaresNoEntregados - ejemplaresEntregados;
```

5. Dificultades encontradas

A lo largo del desarrollo de la aplicación hemos tenido bastantes dificultades que afrontar, sobre todo con temas de diseño pero también muchas respecto al código.

Principales dificultades encontradas:

- 1) Problema de conexión con la base de datos en el segundo intento
Teníamos un Singleton para iniciar la conexión de hibernate con la base de datos, el problema era cuando iniciabas la aplicación y tenías los ajustes de la base de datos mal, entonces sale el mensaje de error de que no hay conexión, pero cuando pones los ajustes bien, e intentabas volver a acceder ya no podías porque el Singleton solo se ejecuta una vez.

Solución: **Cambiar el Singelton por un método normal.**

- 2) Problema con la distribución de los componentes
En un principio habían 3 componentes principales, el menu lateral derecho, el componente principal de la página y el un componete superior con el titulo y filtros.
El problema surge cuando hay que aplicar distintos filtros en el componente superior, como en el caso de Entregas, que es diferente al de libros.

Solución: **Quitar el componente superior y implementarlo individualmente en cada página.**

- 3) Problema al mostrar datos en una tabla de JavaFX
En la base de datos hay información que se almacena en números, pero en realidad son palabras, como en el caso de los ejemplares que tiene por ejemplo, la columna “estado” guarda 0, 1, o 2, el 0 significa Perfecto, el 1 significa Regular y el 2, Mal. Tuvimos un problema para mostrar estos datos en la tabla porque JavaFX lee directamente el pojo y no sabíamos cambiar el numero por un string.

Solución: **Crear pojos personalizados con la información deseada y unos métodos que transforman el pojo original en este.**

6. Propuesta de mejoras

1. Versión para dispositivos móviles.

Con unos cuantos meses más se podría haber desarrollado una versión para dispositivos móviles, lo complejo de esto es desarrollar una nueva interfaz que se adapte a un teléfono la cual sea intuitiva y fácil de usar. Si se pueden hacer aplicaciones “multiplataforma” escritorio-móvil con la misma interfaz en los dos pero sinceramente no queda bien tener una aplicación con las dimensiones de un móvil en un escritorio que se maneja con teclado y ratón y viceversa. Por eso muchas aplicaciones optan por tener diferentes interfaces en cada dispositivo.

2. Temas y apariencia.

Un gran característica que se puede poner es la opción de personalizar el diseño de la aplicación, como la imagen de fondo, el color de los botones, etc.

Esto no es realmente necesario pero puede ser un añadido muy atractivo.

3. Información de conexión con la base de datos

Una cosa interesante que se puede añadir es una indicativo en alguna esquina diciéndote de color verde que si hay conexión con la base de datos o en rojo si no la hay, esto puede que ralentice el programa porque tiene que estar haciendo comprobaciones cada rato pero puede ser un añadido interesante.

4. Modo auto-escaneo

Una característica que me he imaginado sería la de con un simple Textfield, le pasas un código de un ejemplar y automáticamente lo devuelve, o se entrega a un alumno que especifiques momentos después. Esto podría ahorrar algo de tiempo al usuario que gestiona los ejemplares.