



CultSfera – Aplicación para gestión bibliotecaria

Sergi Torres Maiquez
Grado de Ingeniería Informática

Consultor: Antoni Oller Arcas

26-06-2023



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-CompartirIgual [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-sa/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	CultSfera - Aplicación para gestión bibliotecaria
Nombre del autor:	Sergi Torres Maiquez
Nombre del consultor:	Antoni Oller Arcas
Fecha de entrega (mm/aaaa):	06/2023
Área del Trabajo Final:	Java EE
Titulación:	<i>Grado Ingeniería Informática</i>

Resumen del Trabajo (máximo 250 palabras):

Este es un documento que presenta el proyecto de una aplicación para la gestión bibliotecaria desarrollada con Java EE, concretamente mediante el framework Spring. La aplicación sigue el patrón arquitectónico MVC, donde la vista ha sido implementada con el framework Angular.

CultSfere es una aplicación de gestión bibliotecaria que permite la creación de nuevos recursos, ya sean libros, películas, revistas o álbumes de música. Donde el usuario con un rol de bibliotecario puede crear las copias necesarias de cada recursos, realizar préstamos y reservas, y donde los usuarios pueden consultar los recursos existentes, ver sus fichas, o realizar comentarios sobre los diversos recursos.

El proyecto se ha realizado mediante los conocimientos adquiridos durante el grado de Ingeniería Informática, especialmente en el itinerario de Ingeniería del Software. Algunos de estos conocimientos aplicados han sido: El modelo y las fichas de caso de uso, y los diagramas de clase aprendidos en las asignaturas de Ingeniería del Software y Ingeniería de Requisitos, el prototipo de pantallas aprendido en la asignatura Interacción Persona Ordenador, el modelo relacional de datos aprendido en la asignatura Diseño de Base de Datos.

Asimismo, la implementación en Java con el uso del framework Spring se ha realizado aplicando los conocimientos adquiridos en las asignaturas Diseño y Programación Orientada a Objetos, Fundamentos de Programación, Prácticas de Programación, Diseño de Estructuras de Datos, Ingeniería del Software de Componentes y Sistemas Distribuidos y de Proyecto de Desarrollo de Software

Abstract (in English, 250 words or less):

This is a document that presents the project of an application for library management developed with Java EE, using the Spring framework.

This application follows MVC architectural pattern, where the view has been implemented with the Angular framework.

CultSfere is a library management application that allows the creation of new resources, like books, magazines, movies, or music albums. The user with a librarian role can create the required copies of each resource, make loans and bookings. Users can consult the available resources, view resources card or post their comments about the various resources.

The project has been done using the knowledge acquired during the Computer Engineering degree, especially in the Software Engineering itinerary.

Some of this applied knowledge has been: the use case model, use case scenario and the class diagram learned in Software Engineering and Requirements Engineering subjects, the wireframe learned in Human Computer Interaction subject and the relational data model learned in Database Design subject.

Also, the implementation in Java with the Spring framework has been done by applying the knowledge acquired in the programming subjects like Object Oriented Programming and Design, Programming Fundamentals, Programming Practices, Data Structures Design, Software Engineering of Components and

Distributed Systems and the Software Development Project.

Palabras clave (entre 4 y 8):

Java EE, Spring, framework, base de datos, Angular

Índice

1. Introducción	1
1.1 Contexto y justificación del Trabajo	1
1.2 Objetivos del Trabajo.....	2
1.3 Enfoque y método seguido.....	2
1.4 Planificación del Trabajo	3
1.5 Breve resumen de productos obtenidos	6
1.6 Breve descripción de los otros capítulos de la memoria	6
2. Requerimientos, Análisis y Diseño	7
2.1 Modelo de casos de uso.....	7
2.2 Fichas de caso de uso.....	10
2.3 Prototipo de pantallas.....	21
2.4 Modelo relacional de la base de datos	33
2.5 Diagrama de clases.....	36
2.6 Diagrama de arquitectura	36
3. Implementación	42
3.1 Tecnologías utilizadas	42
3.2 Repositorios del proyecto	44
3.3 Características de la aplicación.....	44
3.4 Características pendientes de implementar y posibles mejoras	59
4. Conclusiones	61
5. Glosario	63
6. Bibliografía.....	65
7. Anexos.....	67
7.1 Manual de procedimiento para instalación del proyecto	67

Lista de figuras

1 - Planificación detallada en Diagrama de Gantt	3
2 - Diagrama de Gantt: Planificación PEC1	4
3 - Diagrama de Gantt: Planificación PEC2	4
4 - Diagrama de Gantt: Planificación PEC3	5
5 - Diagrama de Gantt: Planificación de la Memoria	5
6 – Modelo de casos de uso del Rol Administrador	7
7 - Modelo de casos de uso del Rol Bibliotecario	8
8 - Modelo de casos de uso del Rol Usuario	9
9 - Prototipo de Pantalla: Alta de Usuario	22
10 - Prototipo de Pantalla: Listar Usuarios	22
11 - Prototipo de Pantalla: Nuevo Libro	23
12 - Prototipo de Pantalla: Nueva Película	23
13 - Prototipo de Pantalla: Nueva Revista	24
14 - Prototipo de Pantalla: Nuevo Álbum	24
15 - Prototipo de Pantalla: Gestión de Copias	25
16 - Prototipo de Pantalla: Buscar Libro	25
17 - Prototipo de Pantalla: Ficha de un Libro	26
18 - Prototipo de Pantalla: Consultar Histórico de un Recurso	26
19 - Prototipo de Pantalla: Consultar Histórico de un Usuario	27
20 - Prototipo de Pantalla: Gestión de Préstamos	27
21 - Prototipo de Pantalla: Ficha de un Usuario	28
22 - Prototipo de Pantalla: Nuevo Autor	28
23 - Prototipo de Pantalla: Consultar Reservas	29
24 - Prototipo de Pantalla: Comentar un Recurso	29
25 - Prototipo de Pantalla: Consultar un Recurso	30
26 - Prototipo de Pantalla: Consultar Histórico de Préstamos	30
27 - Prototipo de Pantalla: Consultar Préstamos de un Usuario	31
28 - Prototipo de Pantalla: Modificar los Datos Personales de un Usuario	31
29 - Prototipo de Pantalla: Consultar Comentarios de un Usuario	32
30 - Prototipo de Pantalla: Inicio de Sesión	32
31 - Modelo Relacional de la Base de Datos	33
32 - Diagrama de Clases	36
33 - Diagrama de Componentes	38
34 - Pantalla Principal de Admin	45
35 - Crear Usuario	45
36 - Lista de los usuarios existentes	45
37 - Modificar un Usuario	45
38 - Pantalla Principal del Bibliotecario	46
39 - Creación de un Nuevo Libro	46
40 - Listado de los Libros Existentes	47
41 - Consulta Ficha del Libro	47
42 - Añadir una Copia	48
43 - Añadir una Nueva Copia a un Libro	48
44 - Mirar Listado de Copias de un Libro	48
45 - Listar Recursos con sus Copias	49
46 - Realizar Préstamo de una Copia de un Libro a un Usuario	49

47 - Estado de Copia tras Realizar Préstamo	50
48 - Reserva de Libro en Préstamo a un Usuario	50
49 - Correo de Aviso Recibido por el Usuario tras Devolución de un Libro que tiene en Reserva	50
50 - Debug del Correo Enviado	50
51 - Libro Devuelto y en Reserva hasta que el Usuario lo Recoja en Préstamo	51
52 - Estado Actual de Préstamos y Reservas	51
53 - Ver Histórico de Préstamos de un Usuario	52
54 - Ver Préstamos en Curso de un Usuario	52
55 - Ver Préstamos en Curso de un Libro	52
56 - Ver Histórico de Préstamos de un Libro	53
57 - Listado de Usuarios	53
58 - Creando una nueva sanción al usuario	54
59 - Listado de sanciones del usuario	54
60 - Pantalla Principal de un Usuario	55
61 - Histórico de Préstamos de un Usuario	55
62 - Préstamos en Curso de un Usuario	55
63 - Consulta de Reservas Vigentes de un Usuario	55
64 - Realizar Reseña de un Libro	56
65 - Mirar Histórico de Reseñas de un Usuario	56
66 - Búsqueda de Libros por Diferentes Criterios	56
67 - Búsqueda de un Libro: Posibilidad de Mirar Ficha o Realizar Comentario	57
68 - Mirar Ficha de un Libro	58
69 - Pantalla Modificación de Datos de un Usuario	58

1. Introducción

1.1 Contexto y justificación del Trabajo

Este trabajo de final de grado consiste en el desarrollo de una aplicación para la gestión de recursos de una biblioteca privada sin ánimo de lucro, que tiene como objetivo fomentar el acceso a la cultura a toda la población y ofrecer una amplia gama de recursos, incluyendo: libros, revistas, películas y álbumes de música.

Se desea cubrir la gestión de los recursos de forma eficiente, y que facilite a los usuarios la búsqueda y acceso a los recursos, así como a los bibliotecarios la tarea de seguimiento de préstamos, reservas y altas o bajas de recursos.

Aunque se trata de un tipo de aplicación bastante común y extendido en el mercado y hasta cierto punto trivial, no podemos obviar que la cultura es una parte fundamental de la educación y a su vez de nuestra sociedad, y es importante realizar una buena gestión de los recursos de una biblioteca para mejorar la experiencia de los usuarios, motivar el uso de dichos recursos a los usuarios y facilitar el trabajo a los bibliotecarios.

Se quiere conseguir una aplicación que permita la gestión de usuarios, con diferentes grados de permisos (administradores, bibliotecarios y clientes). También se implementará la gestión del catálogo de la biblioteca (altas, bajas, introducción de información de cada recurso).

Además, se añadirá la gestión de préstamos, lo que posibilitará a los usuarios solicitar préstamos, hacer reservas o prorrogar el período de préstamo. Se prevé incluir un sistema de valoración del estado del recurso antes y después del préstamo. Asimismo, la implementación de un sistema de comentarios y valoraciones de los usuarios, que permitirá a los usuarios compartir sus opiniones sobre los recursos disponibles.

También se implementará un sistema de notificaciones que informará al usuario sobre la finalización del período de préstamo, la entrada de recursos que puedan ser de su interés, la disponibilidad de algún recurso en el cual esté interesado o la devolución del préstamo de algún recurso que se haya reservado previamente.

Finalmente, también tenemos previsto incluir un sistema de penalizaciones a usuarios por diversos motivos como demoras en devoluciones de préstamos, mal comportamiento, perder o dañar un recurso.

1.2 Objetivos del Trabajo

1. Desarrollar una aplicación web para una gestión eficiente de recursos de una biblioteca privada.
2. Implementar gestión de usuarios con diferentes roles y permisos: Administradores, bibliotecarios y usuarios normales.
3. Mantenimiento de una base de datos de recursos de la biblioteca, con su gestión de altas, bajas, información por parte de los administradores y de los bibliotecarios.
4. Implementación de la gestión de préstamos y devoluciones de los recursos del catálogo de la biblioteca.
5. Incluir un sistema de sanciones por diversos motivos como devolver préstamos con demora, mal comportamiento, dañar recursos, o extraviar dichos recursos.
6. Permitir a los usuarios compartir opiniones y valoraciones sobre los recursos utilizados a través de un sistema de comentarios y valoraciones con posibilidad de compartir en las redes sociales.
7. Establecer un sistema de notificaciones para informar a los usuarios sobre la finalización del período de préstamo, disponibilidad de recursos o la entrada de nuevos recursos de su interés.

1.3 Enfoque y método seguido

El plan de Trabajo seguido ha sido realizar un producto a partir de cero. Primero se ha realizado un plan de Trabajo donde se han establecido las fechas para la entregas de los principales hitos del proyecto.

A continuación, se han definido las bases y requisitos primordiales y concretos del proyecto. Se ha realizado un análisis de diseño, se ha establecido que actores va a tener la aplicación, los principales casos de uso, y el diseño del prototipo de pantallas, diagrama de clases, modelo relacional y diagrama de la arquitectura.

Después, se ha proseguido con la fase de implementación de la aplicación, de forma iterativa, implementando cada caso de uso establecido en la fase de análisis y diseño.

Por último, se ha realizado los documentos de memoria del proyecto y la presentación de este.

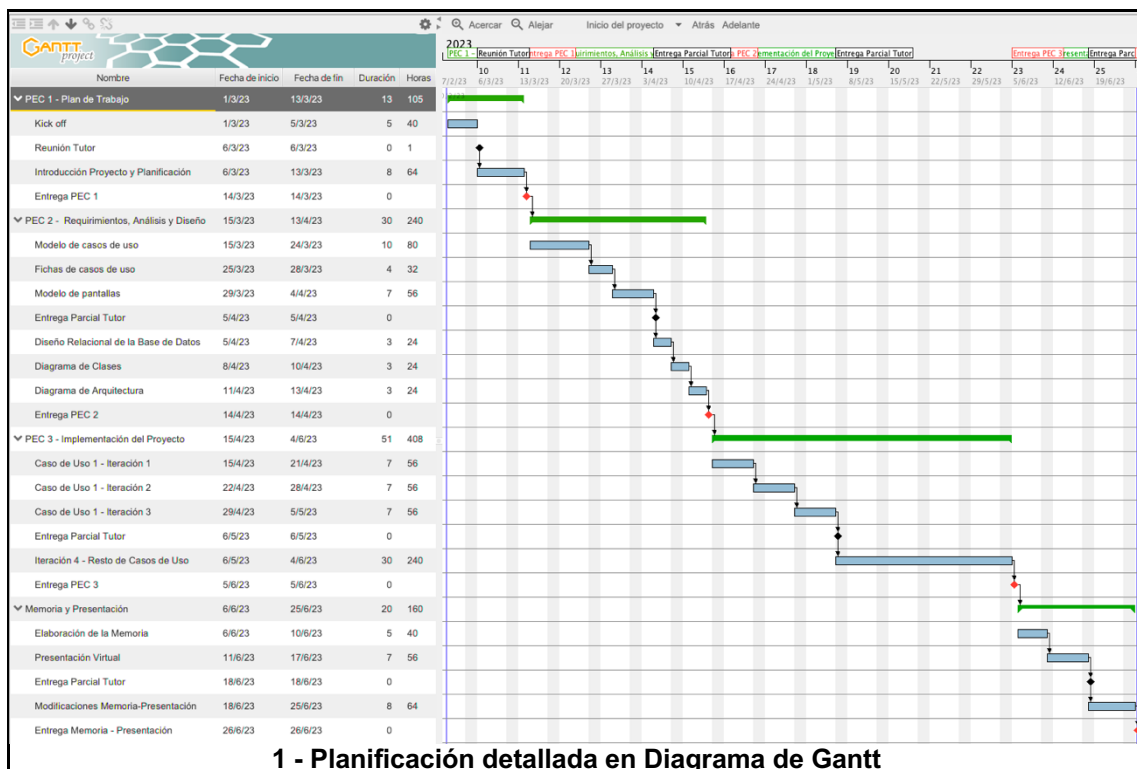
Se puede observar, que el método seguido es el modelo en cascada, donde se establecen las diferentes fases del proyecto, ejecutándolas en orden, y hasta que no se finaliza una etapa no se inicia la siguiente etapa.

1.4 Planificación del Trabajo

A continuación, presentamos un cuadro con las fechas de inicio y entrega de los principales hitos del proyecto.

	Hitos	Fecha Inicio	Fecha Entrega
PEC 1	Plan de trabajo	1-03-2023	14-3-2023
PEC 2	Requerimientos, Análisis y Diseño	15-3-2023	14-4-2023
PEC 3	Implementación	15-4-2023	5-6-2023
PEC 4	Memoria y Presentación	6-6-2023	26-6-2023

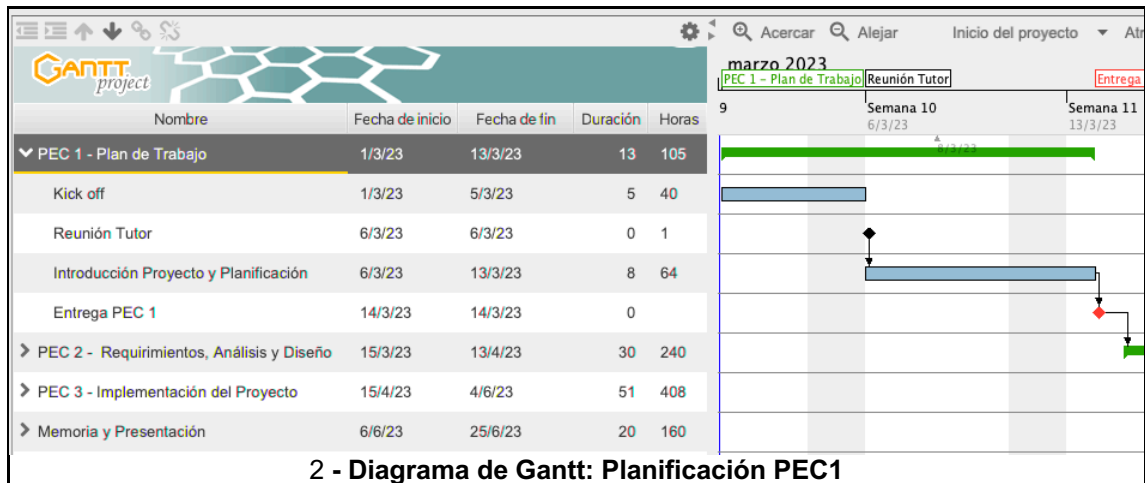
También adjuntamos la planificación detallado con un diagrama de Gantt.



Hay que mencionar que el número de horas es estimativo, ya que hemos calculado los días por 8h. Por otro lado, también contamos los fines de semana, ya que tenemos pensado sacar provecho de ellos.

Vamos a desglosar cada hito para una mejor visualización de las tareas, de sus fechas, días y horas.

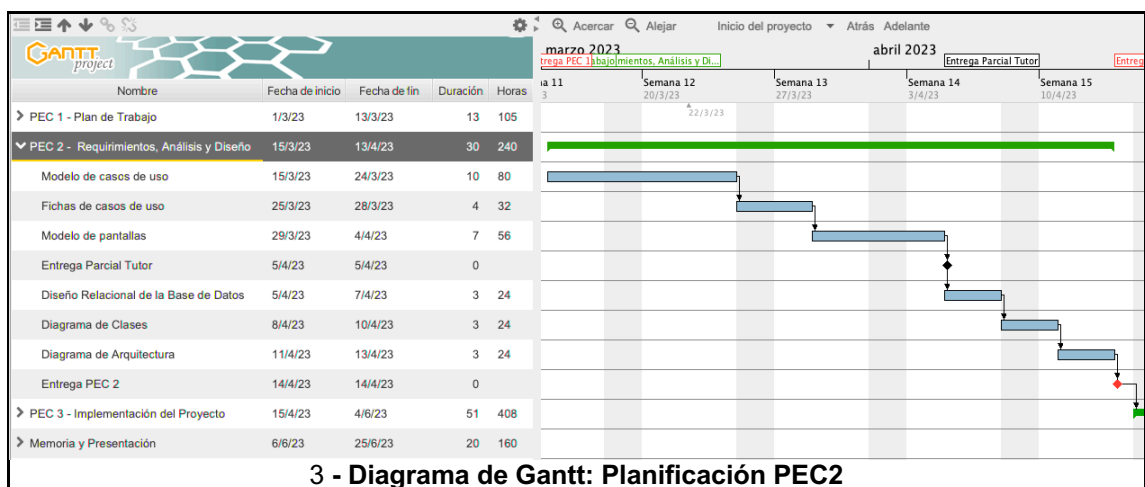
PEC 1 – Plan de Trabajo



En esta primera etapa, se establecerá una toma inicial con el proyecto y se reflexionará sobre la idea del proyecto y como se podría realizar, así como cuales serán las tecnologías necesarias. Luego con la reunión con nuestro tutor, la finalidad es realizar la presentación a nivel general de nuestra idea de proyecto para ser validada.

Por último, se realizará el contexto y justificación trabajo, sus objetivos y la planificación inicial.

PEC 2 – Requerimientos, Análisis y Diseño

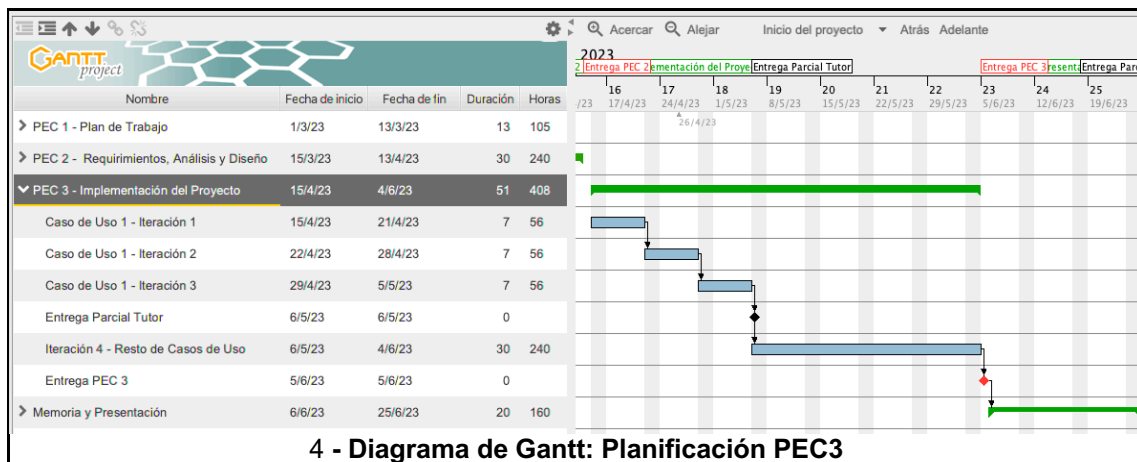


En esta segunda etapa, se comenzará con los requerimientos del proyecto con la elaboración de los casos de uso, así como las fichas de caso de uso para los más complejos. A continuación, se elaborará un modelado de pantallas con

Mockplus RP, y referencias con sus casos de uso. La idea es validar esta parte con una entrega parcial al tutor.

Después, ya se pasará a la parte de análisis y diseño con el diseño relacional de la base de datos, los diagramas de clases con las clases más importantes, y el diagrama de arquitectura, el cual es un diagrama con los componentes más importantes de nuestro proyecto.

PEC 3 – Implementación del Proyecto



4 - Diagrama de Gantt: Planificación PEC3

En esta etapa se comenzará con la implementación del proyecto. En primer lugar, se escogerá un caso de uso para corroborar nuestra arquitectura, se realizarán tres iteraciones, tras lo cual se hará una entrega parcial al tutor para su validación.

Después ya se pasará a una cuarta iteración donde se implementarán el resto de los casos de uso de la aplicación.

Memoria y Presentación



5 - Diagrama de Gantt: Planificación de la Memoria

En esta última etapa, se recopilará toda la documentación elaborada en las anteriores etapas para la realización de la memoria. Además, se hará una presentación virtual consistente en un vídeo de la presentación oral del trabajo realizado, así como los resultados obtenidos.

1.5 Breve resumen de productos obtenidos

Aplicación del tipo SPA (*Single Page Application*), concretamente una aplicación web para gestión bibliotecaria de diferentes recursos como libros, películas, revistas y álbumes musicales.

Memoria del proyecto realizado, con información de las diferentes fases del proyecto y con anexo para la instalación del producto.

Documento de presentación del producto, así como vídeo demostrativo de la operativa básica de la aplicación.

1.6 Breve descripción de los otros capítulos de la memoria

En el capítulo 2 hablamos sobre los requerimientos, análisis y diseño del proyecto. Concretamente mostramos el modelo de casos de uso de los diferentes roles de la aplicación, después mostramos las fichas de caso de uso, mostramos el prototipo de pantallas de la aplicación, seguimos con el modelo relacional de la base de datos y finalizamos con los diagramas de clases y de arquitectura.

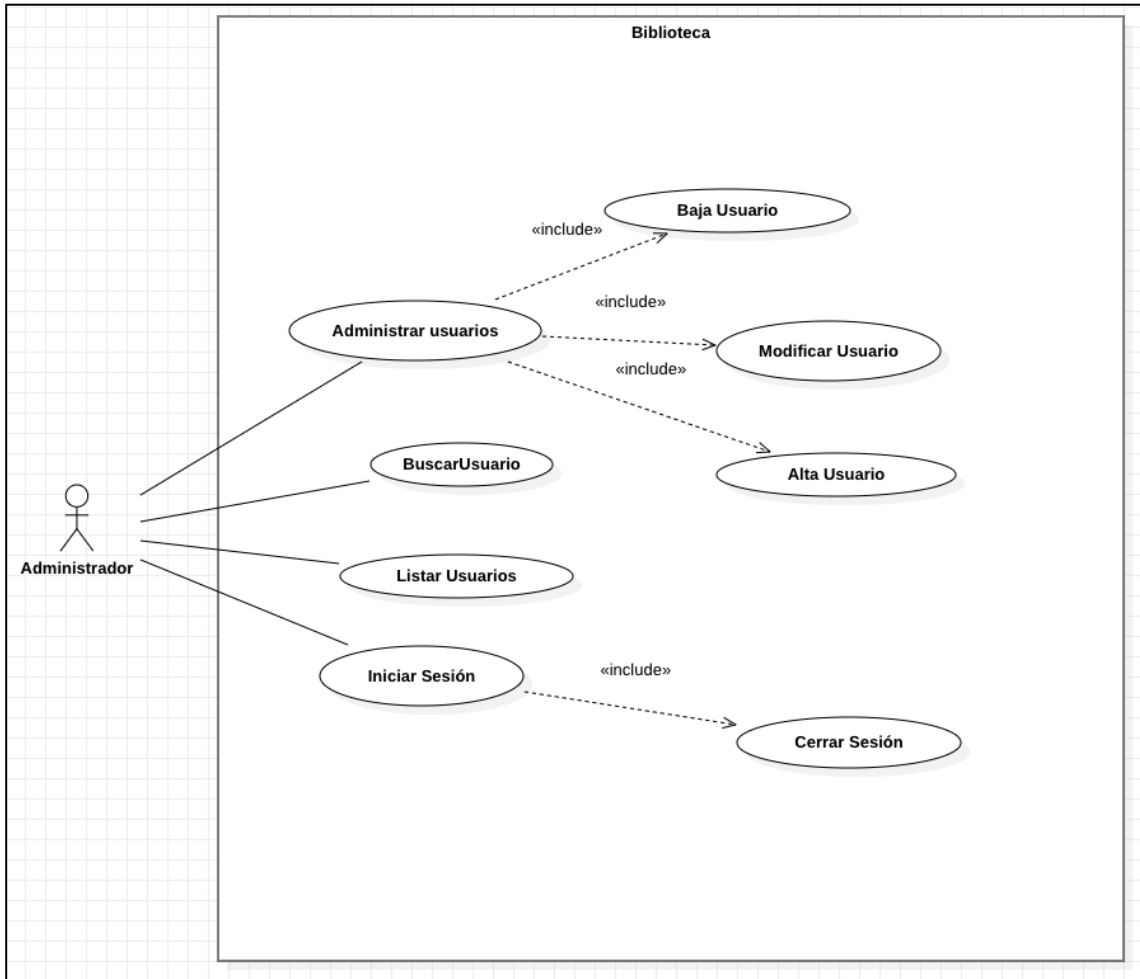
En el capítulo 3 nos centramos en la etapa de implementación donde hacemos un recuento sobre las diversas tecnologías empleadas en el proyecto, y hablamos sobre diferentes características de la aplicación donde también mostramos diversas capturas de pantalla de la aplicación en funcionamiento. Por último, comentaremos sobre aspectos pendientes de implementar y posibles mejoras a realizar.

En el capítulo 4 reflexionamos sobre las conclusiones obtenidas sobre el proyecto. En el capítulo 5 mostramos un glosario sobre algunos términos importantes empleados en la memoria. En el capítulo 6 mencionamos la bibliografía utilizada en el transcurso del proyecto y por último en el capítulo 7 de anexos mostramos el manual para la instalación del proyecto.

2. Requerimientos, Análisis y Diseño

2.1 Modelo de casos de uso

Rol – Administrador



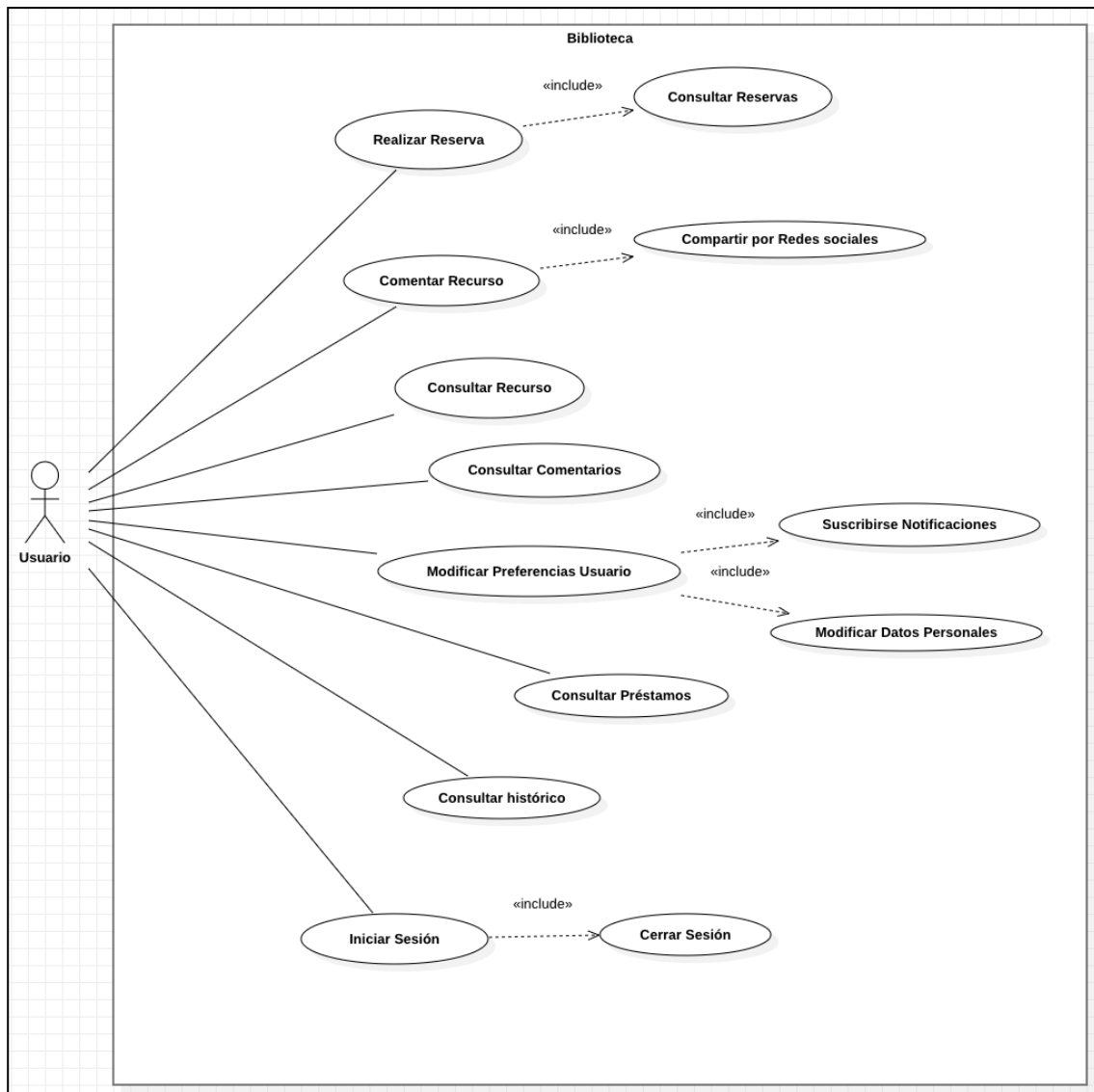
6 – Modelo de casos de uso del Rol Administrador

Rol – Bibliotecario



7 - Modelo de casos de uso del Rol Bibliotecario

Rol – Usuario



8 - Modelo de casos de uso del Rol Usuario

2.2 Fichas de caso de uso

Lista de casos de uso

Código	Caso de Uso	Actor
CU-001	Alta de Usuario	Administrador
CU-002	Listar Usuarios	Administrador
CU-003	Registrar Recurso	Bibliotecario
CU-004	Registrar Copia	Bibliotecario
CU-005	Buscar Recurso	Bibliotecario
CU-006	Consultar Histórico Recurso	Bibliotecario
CU-007	Consultar Histórico Usuario	Bibliotecario
CU-008	Realizar Reserva	Bibliotecario
CU-009	Realizar Préstamo	Bibliotecario
CU-010	Realizar Devolución	Bibliotecario
CU-011	Buscar Usuario	Administrador, Bibliotecario
CU-012	Registrar Autor	Bibliotecario
CU-013	Gestionar Sanción	Bibliotecario
CU-014	Realizar Reserva	Usuario
CU-015	Consultar Reservas	Usuario
CU-016	Comentar Recurso	Usuario
CU-017	Consultar Recurso	Usuario
CU-018	Consultar Histórico	Usuario
CU-019	Consultar Préstamos	Usuario
CU-020	Suscribirse Notificaciones	Usuario
CU-021	Modificar Datos Personales	Usuario
CU-022	Compartir por Redes Sociales	Usuario
CU-023	Consultar Comentarios	Usuario
CU-024	Iniciar Sesión	Todos
CU-025	Cerrar Sesión	Todos

Fichas de caso de uso

CU-001	Alta de Usuario
Actor principal	Administrador
Breve Descripción	El Administrador crea un nuevo usuario en el sistema
Precondición	Estar registrado como Administrador
Postcondición	El nuevo usuario quedará registrado en el sistema
Escenario principal de éxito	<ol style="list-style-type: none"> 1. El administrador elige dar de alta un nuevo usuario 2. El administrador introduce los datos del usuario (nombre, apellidos, domicilio, teléfono, email) 3. El sistema comprueba que los datos sean válidos. 4. El sistema registra el usuario en la base de datos.
Extensiones	<ol style="list-style-type: none"> 2a. El sistema indica que los datos de usuario ya existen en la base de datos 3a. El sistema indica que el tipo de datos introducido no es válido.

Para baja de usuario y modificar usuario ver el caso de uso CU-001

CU-002	Listar Usuarios
Actor principal	Administrador
Breve Descripción	El Administrador quiere consultar el listado de los usuarios registrados en la aplicación.
Precondición	Debe estar registrado como Administrador.
Postcondición	El sistema devuelve un listado de usuarios con las condiciones solicitadas.
Escenario principal de éxito	<ol style="list-style-type: none"> 1. El Administrador quiere sacar un listado de usuarios 2. El Administrador aplica algún filtro o condición a cumplir 3. El Sistema devuelve un listado de usuarios que cumplen con los filtros indicados
Extensiones	3a. El Sistema devuelve un mensaje de que no se han encontrado usuarios con los filtros aplicados

CU-003	Registrar Recurso
Actor principal	Bibliotecario
Breve Descripción	El Bibliotecario crea un nuevo recurso en el sistema
Precondición	El usuario debe estar registrado como bibliotecario
Postcondición	El nuevo recurso queda registrado en la base de datos
Escenario principal de éxito	<ol style="list-style-type: none"> 1. El bibliotecario elige dar de alta un nuevo recurso. 2. Elige el tipo de recurso que quiere introducir en el sistema 3. El bibliotecario introduce los datos del recurso. 4. El sistema comprueba que los datos sean válidos. 5. El sistema registra el recurso en la base de datos. 6. El sistema envía notificación a los usuarios interesados de la nueva alta de recurso
Extensiones	<ol style="list-style-type: none"> 3a. El sistema indica que los datos del recurso ya existen en la base de datos. 4a. El sistema informa que el tipo de datos introducido no es válido

Para modificar recurso o eliminar el recurso ver el caso de uso CU-003

CU-004	Registrar Copia
Actor principal	Bibliotecario
Breve Descripción	El Bibliotecario crea una nueva copia de un recurso en el sistema
Precondición	El recurso debe existir en el sistema El usuario debe estar registrado como bibliotecario
Postcondición	La nueva copia queda vinculada al recurso y registrada en la base de datos
Escenario principal de éxito	<ol style="list-style-type: none"> 1. El bibliotecario elige dar de alta una nueva copia de un recurso 2. El bibliotecario introduce los datos del recurso a buscar. 3. El sistema comprueba que los datos sean válidos. 4. El sistema muestra la ficha del recurso escogido 5. El bibliotecario introduce los datos de la nueva copia del recurso. 6. El sistema comprueba que los datos sean válidos. 7. El sistema registra la copia en el sistema y la vincula al recurso.
Extensiones	<ol style="list-style-type: none"> 3a.El sistema indica que el tipo de datos no es válido. 4a. El sistema indica el recurso introducido no existe. 6a.El sistema indica que el tipo de datos no es válido.

Para modificar copia o eliminar una copia ver el caso de uso CU-004

CU-005	Buscar Recurso
Actor principal	Bibliotecario
Breve Descripción	El Bibliotecario quiere buscar un recurso de la biblioteca aplicando filtros
Precondición	-
Postcondición	El sistema devuelve un listado de recursos que cumple las condiciones del usuario
Escenario principal de éxito	<ol style="list-style-type: none"> 1. El Bibliotecario quiere buscar uno o varios recursos. 2. El Bibliotecario introduce los datos y condiciones de los filtros para buscar recursos. 3. El Sistema devuelve listado del recurso o de los recursos que cumplen con las condiciones introducidas.
Extensiones	3a. El Sistema devuelve un mensaje de que no se han encontrado recursos con las condiciones introducidas

CU-006	Consultar Histórico Recurso
Actor principal	Bibliotecario
Breve Descripción	El Bibliotecario quiere consultar el histórico de un recurso
Precondición	El usuario debe estar registrado como bibliotecario
Postcondición	El sistema devuelve un histórico de préstamos de un recurso.
Escenario principal de éxito	1. El bibliotecario introduce el tipo de recurso y el título del recurso que quiere consultar 2. El sistema devuelve el histórico sobre el recurso solicitado.
Extensiones	2a. El Sistema devuelve un mensaje de que el recurso aún no tiene ningún préstamos.

CU-007	Consultar Histórico Usuario
Actor principal	Bibliotecario
Breve Descripción	El Bibliotecario quiere consultar el histórico de un usuario
Precondición	El usuario debe estar registrado como bibliotecario
Postcondición	El sistema devuelve un histórico de préstamos de un histórico de préstamos de un usuario
Escenario principal de éxito	1. El bibliotecario introduce el dni del usuario que quiere consultar. 2. El sistema devuelve el histórico sobre el usuario
Extensiones	2a. El Sistema devuelve un mensaje de que el usuario aún no ha realizado un préstamo.

CU-008	Realizar Reserva
Actor principal	Bibliotecario
Breve Descripción	El Bibliotecario realiza la reserva de un recurso para un determinado usuario
Precondición	El recurso debe existir. El recurso no debe encontrarse reservado
Postcondición	El recurso queda reservado para el usuario que introduzca el bibliotecario
Escenario principal de éxito	1. El bibliotecario busca el recurso a reservar 2. El sistema devuelve el recurso solicitado 3. El bibliotecario establece la reserva del recurso. 4. El sistema pregunta los datos del usuario 5. El sistema informa que el recurso ya está reservado a nombre del usuario introducido 6. El sistema envía notificación al usuario
Extensiones	3a El sistema informa que el recurso ya se encuentra reservado. 4a El usuario introducido no existe.

--	--

CU-009	Realizar Préstamo
Actor principal	Bibliotecario
Breve Descripción	El Bibliotecario realiza el préstamo de un recurso para un determinado usuario
Precondición	El recurso debe existir. El recurso no debe encontrarse reservado ni en préstamo. El usuario debe estar registrado como bibliotecario
Postcondición	El recurso queda en préstamo para el usuario introducido por el bibliotecario
Escenario principal de éxito	1. El bibliotecario busca el recurso a prestar 2. El sistema devuelve el recurso solicitado 3. El bibliotecario establece el préstamo del recurso 4. El sistema pregunta los datos del usuario 5. El sistema informa que el recurso ya está prestado a nombre del usuario introducido
Extensiones	4a El sistema informa que el recurso se encuentra reservado a un usuario diferente. 4b El sistema informa que el usuario no existe

CU-010	Realizar Devolución
Actor principal	Bibliotecario
Breve Descripción	El Bibliotecario realiza la devolución de un recurso en préstamo
Precondición	El recurso debe existir. El recurso debe estar en préstamo. El usuario debe estar registrado como bibliotecario
Postcondición	El recurso deja de estar en préstamo y vuelve a estar disponible
Escenario principal de éxito	1.El bibliotecario busca el usuario que devuelve el recurso 2. El sistema devuelve la ficha del usuario solicitado 3. El bibliotecario mira el listado de préstamos actuales del usuario 4. El sistema devuelve los recursos que el usuario tiene actualmente en préstamo 5. El bibliotecario selecciona el recurso que se devuelve y lo deja en disponible.
Extensiones	1a. El usuario no existe 3a. El sistema no devuelve ningún resultado o bien en el resultado devuelto no sale el recurso que se devuelve

CU-011	Buscar Usuario
Actor principal	Administrador,Bibliotecario
Breve Descripción	El Bibliotecario quiere buscar un usuario de la biblioteca por nombre y apellidos o por dni.
Precondición	El usuario debe estar registrado como administrador o como bibliotecario
Postcondición	El sistema devuelve el usuario que cumple con las condiciones de búsqueda.
Escenario principal de éxito	<ol style="list-style-type: none"> 1. El Bibliotecario quiere buscar un usuario 2. El Bibliotecario introduce los datos para buscar el usuario. 3. El Sistema valida que los datos introducidos sean correctos. 4. El Sistema devuelve la ficha del usuario.
Extensiones	<ol style="list-style-type: none"> 3a. El Sistema devuelve un mensaje diciendo que los datos introducidos no son válidos. 4a. El Sistema devuelve un mensaje de que no se han encontrado recursos con las condiciones introducidas

CU-012	Registrar Autor
Actor principal	Bibliotecario
Breve Descripción	El Bibliotecario registra un nuevo autor en la base de datos
Precondición	El usuario debe estar registrado como bibliotecario
Postcondición	El nuevo autor queda registrado en la base de datos
Escenario principal de éxito	<ol style="list-style-type: none"> 1. El bibliotecario elige dar de alta un nuevo autor. 2. El bibliotecario introduce los datos del autor. 3. El sistema comprueba que los datos sean válidos. 4. El sistema registra el autor en la base de datos.
Extensiones	<ol style="list-style-type: none"> 2a. El sistema indica que los datos del autor ya existen en la base de datos. 3a. El sistema informa que el tipo de datos introducido no es válido

Para modificar un autor o eliminar un autor ver el caso de uso CU-012

CU-013	Gestionar Sanción
Actor principal	Bibliotecario
Breve Descripción	El usuario quiere realizar una sanción a un usuario
Precondición	El usuario debe existir El usuario no debe tener ninguna sanción en curso
Postcondición	El usuario queda sanciona y no puede recibir préstamos ni reservas
Escenario principal de éxito	1.El bibliotecario busca al usuario deseado 2.El sistema le devuelve el resultado 3.El bibliotecario indica que quiere hacer sanción y introduce los datos de la sanción 4.El sistema le indica que la sanción ya está en curso y envía un correo de notificación al usuario
Extensiones	1a. El sistema le indica que el usuario no ha sido encontrado 3a. El sistema le indica que faltan datos para completar la sanción

CU-014	Realizar Reserva
Actor principal	Usuario
Breve Descripción	El usuario quiere realizar reserva de un recurso de la biblioteca
Precondición	El recurso debe existir El recurso no debe encontrarse reservado
Postcondición	El recurso queda reservado para el usuario
Escenario principal de éxito	1.El usuario busca el recurso que le interesa 2.El sistema le devuelve el resultado 3.El usuario entra en la ficha del recurso y selecciona reservar 4.El sistema le indica que tiene el recurso reservado a su nombre
Extensiones	1a. El sistema le indica que el recurso no ha sido encontrado 3a. El sistema le indica que el recurso ya está reservado

CU-015	Consultar Reservas
Actor principal	Usuario
Breve Descripción	El Usuario quiere consultar las reservas que tiene en curso
Precondición	-
Postcondición	El sistema devuelve un listado de reservas en curso que tiene el usuario
Escenario principal de éxito	<ol style="list-style-type: none"> 1. El Usuario quiere consultar las reservas que tiene en curso 2. El Usuario elige en consultar mis reservas 3. El Sistema devuelve el listado de las reservas actuales del usuario.
Extensiones	3a. El Sistema devuelve un mensaje al usuario conforme no tiene reservas en curso

CU-016	Comentar Recurso
Actor principal	Usuario
Breve Descripción	El usuario quiere hacer un comentario sobre un recurso.
Precondición	<p>El recurso debe existir</p> <p>El usuario no puede tener un comentario realizado anteriormente.</p>
Postcondición	El recurso tiene el nuevo comentario del usuario en su lista de comentarios
Escenario principal de éxito	<ol style="list-style-type: none"> 1. El usuario consulta su histórico de préstamos 2. El sistema le muestra la lista de sus préstamos 3. El usuario selecciona el recurso del que está interesado en realizar un comentario 4. El usuario escribe el comentario, y en caso de desearlo incluye una valoración numérica y lo envía al acabar.
Extensiones	3a. El recurso seleccionado muestra el comentario del usuario con anterioridad

CU-017	Consultar Recurso
Actor principal	Usuario
Breve Descripción	El usuario quiere consultar un recurso de la biblioteca aplicando filtros
Precondición	-
Postcondición	El sistema devuelve un listado de recursos que cumple las condiciones del usuario
Escenario principal de éxito	1. El usuario quiere hacer una consulta de uno o varios recursos 2. El usuario introduce los datos y condiciones a cumplir 3. El Sistema devuelve uno o varios recursos que cumplen con las condiciones introducidas.
Extensiones	3a. El Sistema devuelve un mensaje de que no se han encontrado recursos con las condiciones introducidas

CU-018	Consultar Histórico
Actor principal	Usuario
Breve Descripción	El Usuario quiere consultar el histórico de sus préstamos
Precondición	-
Postcondición	El sistema devuelve un histórico de préstamos del usuario
Escenario principal de éxito	1. El usuario escoge consultar el histórico de sus préstamos 2. El sistema devuelve el histórico de préstamos del usuarios
Extensiones	2a. El Sistema devuelve un mensaje de que el usuario aún no ha realizado un préstamo

CU-019	Consultar Préstamos
Actor principal	Usuario
Breve Descripción	El Usuario quiere consultar los préstamos que tiene en curso
Precondición	-
Postcondición	El sistema devuelve un listado de préstamos en curso que tiene el usuario
Escenario principal de éxito	1. El Usuario quiere consultar los préstamos que tiene en curso 2. El Usuario elige en consultar mis préstamos 3. El Sistema devuelve el listado de préstamos actuales del usuario
Extensiones	3a. El Sistema devuelve un mensaje al usuario conforme no tiene préstamos en curso

CU-020	Suscribirse Notificaciones
Actor principal	Usuario
Breve Descripción	El usuario quiere recibir notificaciones de los nuevos recursos de la biblioteca
Precondición	-
Postcondición	El sistema registra al usuario en la lista de suscriptores
Escenario principal de éxito	<ol style="list-style-type: none"> 1. El usuario quiere recibir notificaciones de las novedades de la biblioteca 2. El usuario elige modificar preferencias de usuario. 3. El usuario elige suscribirse a las notificaciones para novedades 4. El sistema registra al usuario en la lista de suscriptores
Extensiones	-

CU-021	Modificar Datos Personales
Actor principal	Usuario
Breve Descripción	El usuario quiere modificar sus datos personales
Precondición	-
Postcondición	El sistema modifica los datos personales introducidos por el usuario
Escenario principal de éxito	<ol style="list-style-type: none"> 1. El usuario quiere modificar algunos de sus datos personales 2. El usuario elige modificar preferencias de usuario 3. El usuario elige modificar datos personales 4. El sistema le muestra los datos personales del usuario 5. El usuario modifica los datos que desea 6. El Sistema comprueba que los datos sean válidos 7. El Sistema registra los nuevos datos en la base de datos
Extensiones	6a. El Sistema devuelve un mensaje informando que los datos introducidos no son válidos

CU-022	Compartir por redes sociales
Actor principal	Usuario
Breve Descripción	El usuario quiere compartir por redes sociales el comentario que ha hecho sobre un recurso
Precondición	El recurso debe tener comentario por parte del usuario. El usuario debe tener redes sociales
Postcondición	El usuario tiene compartido en sus redes sociales el comentario o valoración de un recurso que ha tomado en préstamo
Escenario principal de éxito	<ol style="list-style-type: none"> 1. El usuario acaba de hacer el comentario de un recurso. 2. El sistema informa al usuario que acaba de realizar el comentario 3. El usuario selecciona el botón de compartir en redes sociales 4. El sistema le muestra diferentes redes sociales donde compartir el comentario. 5. El usuario selecciona la red social donde desea enviar el comentario. 6. El sistema enlaza con la API de la red social y muestra el mensaje a compartir 7. El usuario da conformidad y comparte el mensaje
Extensiones	6a. El usuario configura el acceso a la red social por primera vez

CU-023	Consultar Comentarios
Actor principal	Usuario
Breve Descripción	El usuario quiere consultar el histórico de sus comentarios
Precondición	-
Postcondición	El sistema devuelve el listado de comentarios del usuario
Escenario principal de éxito	<ol style="list-style-type: none"> 1. El usuario quiere hacer una consulta del histórico de sus comentarios. 2. El Sistema devuelve el histórico de los comentarios del usuario.
Extensiones	2a. El Sistema devuelve un mensaje conforme a no existen comentarios por parte del usuario.

CU-024	Iniciar Sesión
Actor principal	Administrador, Bibliotecario, Usuario
Breve Descripción	El usuario quiere iniciar sesión en el sistema
Precondición	El usuario debe estar registrado en el sistema
Postcondición	El usuario estará con una sesión iniciada en el sistema con los permisos inherentes a su perfil de usuario
Escenario principal de éxito	1. El usuario introduce su correo electrónico y su contraseña en los campos de input correspondientes y hace clic en el botón ok. 2. El sistema verifica que los datos de usuario existen en el sistema y da acceso al usuario en el sistema.
Extensiones	2a. El sistema indica que los datos de acceso son incorrectos.

CU-025	Cerrar Sesión
Actor principal	Administrador, Bibliotecario, Usuario
Breve Descripción	El usuario quiere cerrar la sesión en el sistema
Precondición	El usuario debe estar con su sesión iniciada en el sistema
Postcondición	El usuario ha cerrado su sesión en el sistema
Escenario principal de éxito	1. El usuario selecciona la opción cerrar la sesión 2. El sistema graba los datos pendientes y limpiar los datos temporales y cierra la sesión e informa al usuario que se cerró la sesión.
Extensiones	-

2.3 Prototipo de pantallas

No hemos puesto pantallas para todos los casos de uso, sino que hemos puesto las pantallas que consideramos más significativas para hacernos una idea de como puede quedar el diseño final. Ponemos el título de cada pantalla con el código del caso de uso con el que se relaciona.

Alta de Usuario (CU-001)

Alta de usuario

Nombre

Apellidos

NIF

Dirección

Teléfono

Email

Usuario

Rol

9 - Prototipo de Pantalla: Alta de Usuario

Listar Usuarios (CU-002)

Listado de Usuarios

DNI	Usuario	Fecha Alta			
48145623K	jperez	12-10-2022			
71234567J	lfernandez	12-10-2022			
27789012F	amartinez	12-10-2022			
93456789P	mrodriguez	12-10-2022			
56891234R	lgarcia	12-10-2022			
34567890S	plopez	12-10-2022			
90123456H	jrodriguez	12-10-2022			
12345678T	dfernandez	12-10-2022			
78901234G	cgonzalez	12-10-2022			
45678901L	lmartin	12-10-2022			

Página 3 de 30 << < 1 2 3 4 > >>

10 - Prototipo de Pantalla: Listar Usuarios

Nuevo Libro (CU-003)

The screenshot shows a web interface for adding a new book. At the top, there is a navigation bar with the following items: 'Gestión Recursos', 'Gestión Copias', 'Consultar Histórico', 'Gestión Prestamos', 'Consultar Recursos', and 'Bibliotecario'. The main heading is 'Nuevo Libro'. The form contains the following fields:

- Título: Text input field.
- Autor: Text input field with a '+' icon.
- Género: Dropdown menu with 'Ciencia-Ficción' selected.
- ISBN: Text input field.
- Editorial: Dropdown menu with 'Planeta' selected.
- Año de Publicación: Text input field.
- Sinopsis: Text area.

At the bottom of the form, there are two buttons: 'OK' and 'Cancelar'.

11 - Prototipo de Pantalla: Nuevo Libro

Nueva Película (CU-003)

The screenshot shows a web interface for adding a new movie. At the top, there is a navigation bar with the following items: 'Gestión Recursos', 'Gestión Copias', 'Consultar Histórico', 'Gestión Prestamos', 'Consultar Recursos', and 'Bibliotecario'. The main heading is 'Nueva Película'. The form contains the following fields:

- Título: Text input field.
- Director: Text input field with a '+' icon.
- Reparto: Text input field with a '+' icon.
- Género: Dropdown menu with 'Comedia' selected.
- Duración: Text input field.
- Año Lanzamiento: Text input field.
- Sinopsis: Text area.

At the bottom of the form, there are two buttons: 'OK' and 'Cancelar'.

12 - Prototipo de Pantalla: Nueva Película

Nueva Revista (CU-003)

Gestión Recursos Gestión Copias Consultar Histórico Gestión Prestamos Consultar Recursos Bibliotecario

Nueva Revista

Título

Mes Publicación

Editorial

Temática

Número Páginas

13 - Prototipo de Pantalla: Nueva Revista

Nuevo Álbum (CU-003)

Gestión Recursos Gestión Copias Consultar Histórico Gestión Prestamos Consultar Recursos Bibliotecario

Nuevo Álbum

Título

Artista

Género

Año Lanzamiento

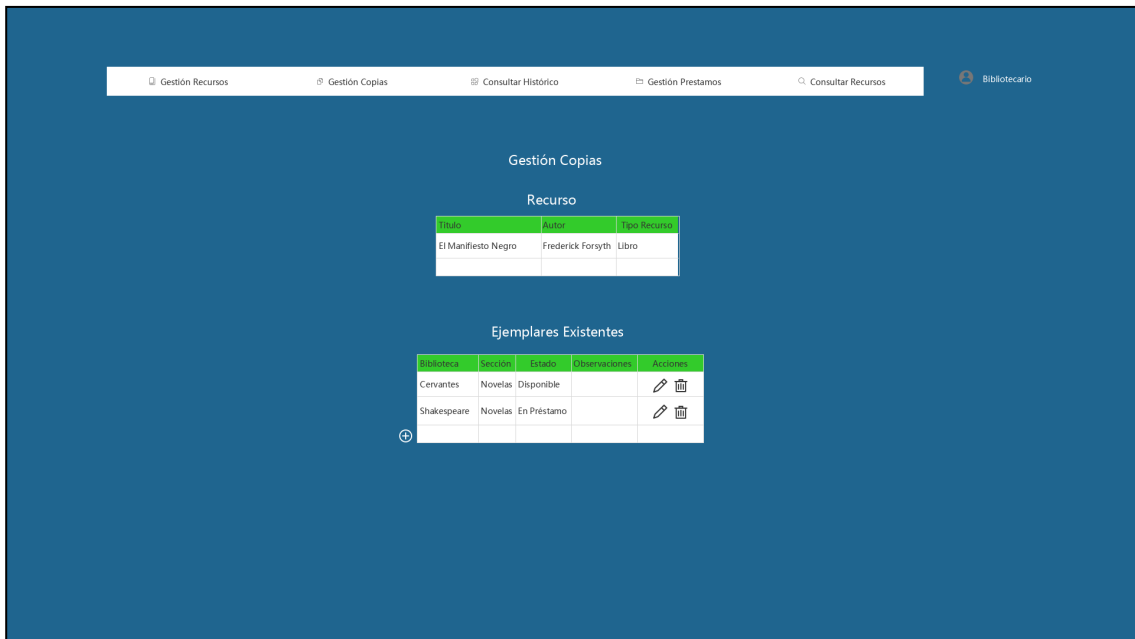
Duración

Soporte

Discográfica

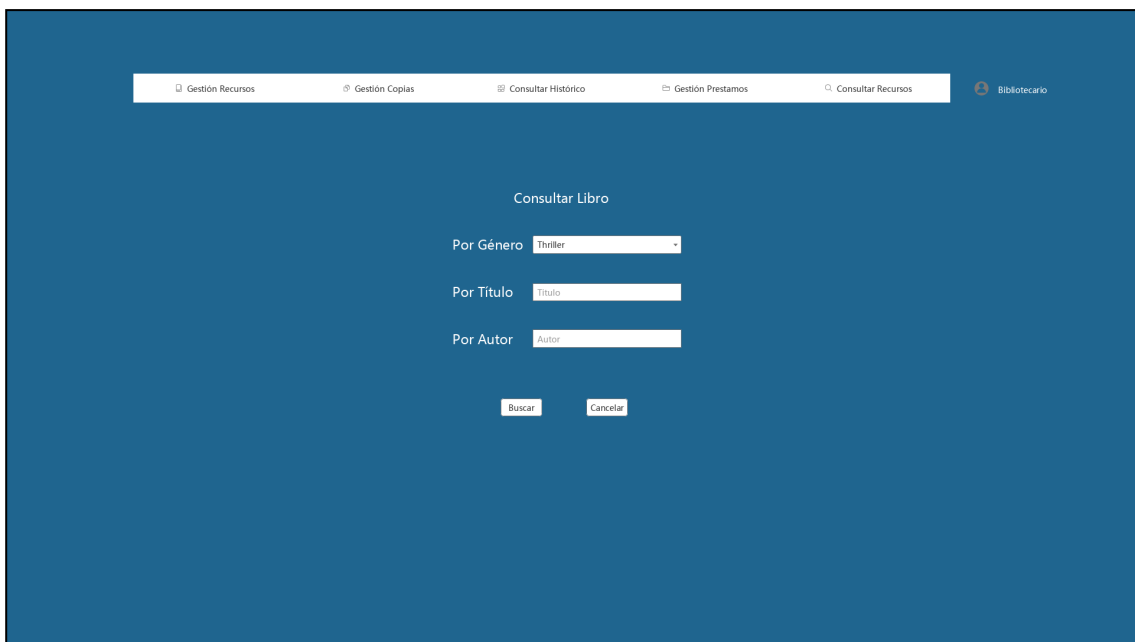
14 - Prototipo de Pantalla: Nuevo Álbum

Gestión de Copias (CU-004)



15 - Prototipo de Pantalla: Gestión de Copias

Buscar Libro (CU-005)



16 - Prototipo de Pantalla: Buscar Libro

Ficha Libro (CU-005)

Ficha Libro

Título: El Manifiesto Negro
Autor: Frederick Forsyth
Género: Thriller
ISBN: 9788497597715
Editorial: Planeta
Año de Publicación: 1996
Páginas: 380

Sinopsis: El Manifiesto Negro es Forsyth en toda su plenitud: intrincado realismo político, intriga a raudales y una trama que corta el aliento. «Era el verano en que el precio de una barra de pan alcanzó el millón de rublos.»

Copias

Biblioteca	Sección	Estado	Observaciones	Acciones
Cervantes	Novelas	Disponible		
Shakespeare	Novelas	En Préstamo		

17 - Prototipo de Pantalla: Ficha de un Libro

Consultar Histórico Recurso (CU-006)

Histórico Préstamos

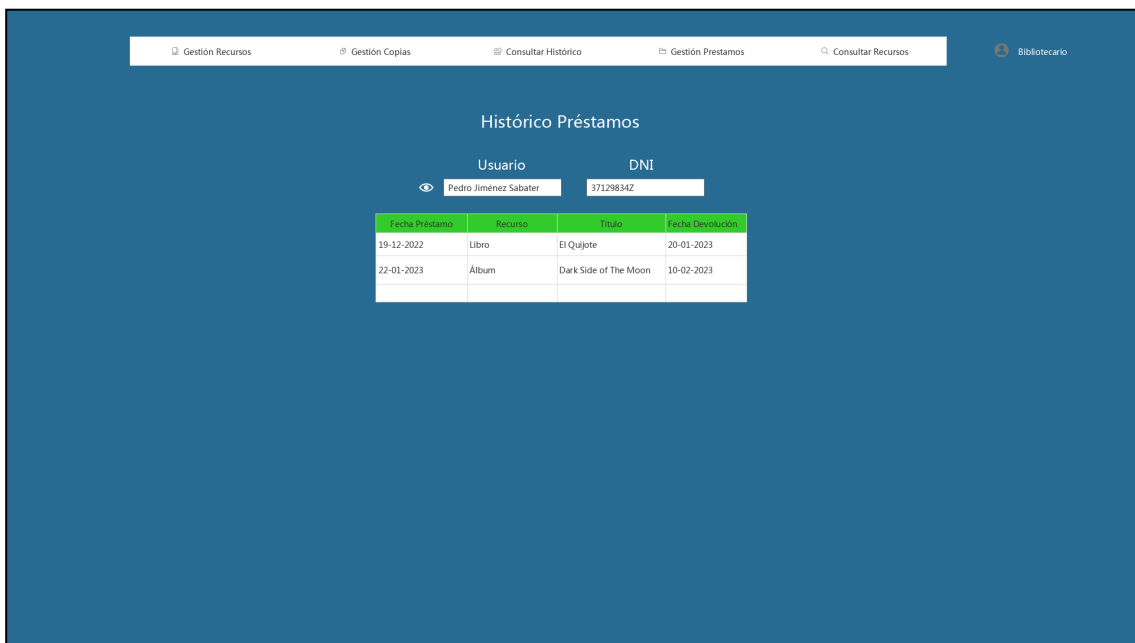
Recurso: El Manifiesto Negro
Tipo: Libro
Copia Biblioteca Cervantes

Fecha Préstamo	Fecha Devolución	Usuario
19-12-2022	20-01-2023	Andrés Gutiérrez Mas
22-01-2023	10-02-2023	Ana Montes Pérez

Página 5 de 5

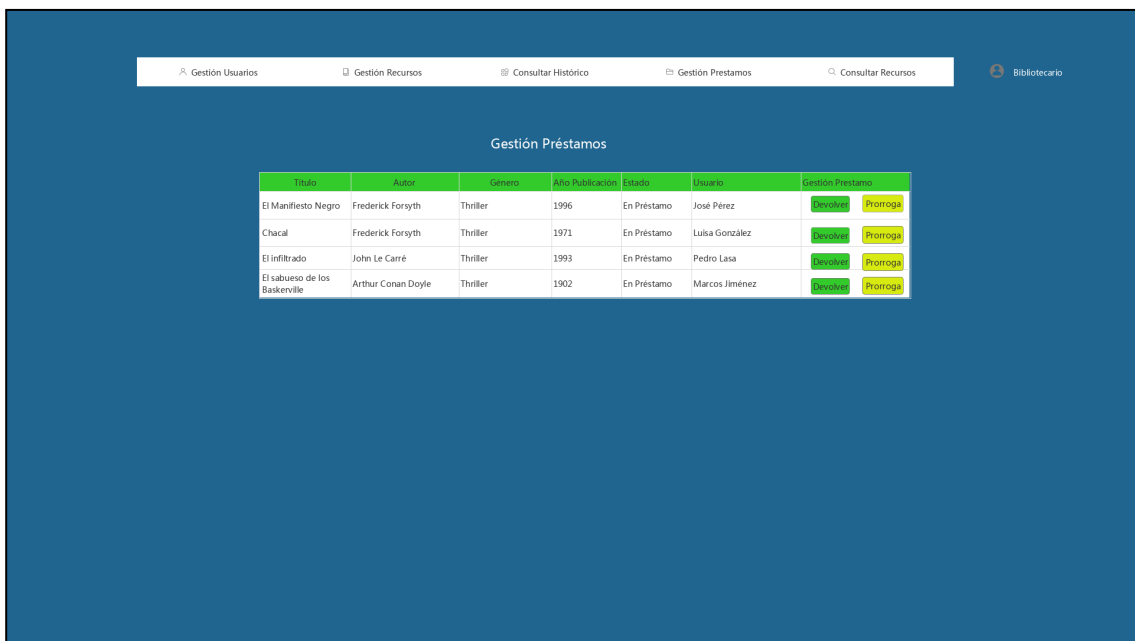
18 - Prototipo de Pantalla: Consultar Histórico de un Recurso

Consultar Histórico Usuario (CU-007)



19 - Prototipo de Pantalla: Consultar Histórico de un Usuario

Gestión Préstamos (CU-009)



20 - Prototipo de Pantalla: Gestión de Préstamos

Ficha Usuario (CU-011)

The screenshot shows a web interface with a dark blue background. At the top, there is a navigation bar with five menu items: 'Gestión Recursos', 'Gestión Copias', 'Consultar Histórico', 'Gestión Prestamos', and 'Consultar Recursos'. On the right side of the navigation bar, there is a user profile icon labeled 'Bibliotecario'. The main content area is titled 'Ficha Usuario' and contains a form with the following fields and values:

Nombre	Pedro
Apellido	Jiménez Sabater
Email	pjimenez@gmail.com
NIF	37129834Z
Domicilio	Gran Via, 312 4º 1ª
Teléfono	651283104

Below the form, there are three buttons: 'Ver Prestamos', 'Ver Penalizaciones', and 'Ver Reservas'.

21 - Prototipo de Pantalla: Ficha de un Usuario

Nuevo Autor (CU-012)

The screenshot shows a web interface with a dark blue background. At the top, there is a navigation bar with five menu items: 'Gestión Recursos', 'Gestión Copias', 'Consultar Histórico', 'Gestión Prestamos', and 'Consultar Recursos'. On the right side of the navigation bar, there is a user profile icon labeled 'Bibliotecario'. The main content area is titled 'Nuevo Autor' and contains a form with the following fields and values:

Nombre	Título
Apellido	Autor
Rol	Intérprete

Below the form, there are two buttons: 'OK' and 'Cancelar'.

22 - Prototipo de Pantalla: Nuevo Autor

Consultar Reservas (CU-014)

Título	Orden	
El Quijote	1	Cancelar
Dark Side of The Moon	3	Cancelar

23 - Prototipo de Pantalla: Consultar Reservas

Comentar Recurso (CU-015)

Título: El Manifiesto Negro

Comentar: Un libro muy interesante...

Valoración: ★★★★★

Enviar Cancelar

24 - Prototipo de Pantalla: Comentar un Recurso

Consultar Recurso (CU-016)

The screenshot shows a web interface with a navigation bar at the top containing three menu items: 'Consultar Histórico', 'Consultar Préstamos', and 'Consultar Recursos'. A user profile icon labeled 'Usuario' is in the top right corner. The main content area is titled 'Resultados Búsqueda' and contains a table with the following data:

Título	Autor	Género	Año Publicación	Estado	
El Manifiesto Negro	Frederick Forsyth	Thriller	1996	Disponible	Reservar
Chacal	Frederick Forsyth	Thriller	1971	En Préstamo	Reservar
Odessa	Frederick Forsyth	Thriller	1972	Disponible	Reservar
El Cuarto Protocolo	Frederick Forsyth	Thriller	1984	Disponible	Reservar

25 - Prototipo de Pantalla: Consultar un Recurso

Consultar Histórico (CU-017)

The screenshot shows a web interface with a navigation bar at the top containing three menu items: 'Consultar Histórico', 'Consultar Préstamos', and 'Consultar Recursos'. A user profile icon labeled 'Usuario' is in the top right corner. The main content area is titled 'Histórico Préstamos' and contains a table with the following data:

Fecha Préstamo	Recurso	Título	Fecha Devolución
19-12-2022	Libro	El Quijote	20-01-2023
22-01-2023	Álbum	Dark Side of The Moon	10-02-2023

26 - Prototipo de Pantalla: Consultar Histórico de Préstamos

Consultar Préstamos (CU-018)

The screenshot shows a web interface with a dark blue background. At the top, there is a navigation bar with three links: 'Consultar Histórico', 'Consultar Préstamos', and 'Consultar Recursos'. On the right side of the navigation bar, there is a user profile icon labeled 'Usuario'. Below the navigation bar, the main heading is 'Mis Préstamos Actuales'. Underneath this heading is a table with five columns: 'Titulo', 'Autor', 'Genero', 'Año Publicación', and 'Estado'. The table contains three rows of data.

Titulo	Autor	Genero	Año Publicación	Estado
El Manifiesto Negro	Frederick Forsyth	Thriller	1996	En Préstamo
Chacal	Frederick Forsyth	Thriller	1971	En Préstamo
Odesa	Frederick Forsyth	Thriller	1972	En Préstamo

27 - Prototipo de Pantalla: Consultar Préstamos de un Usuario

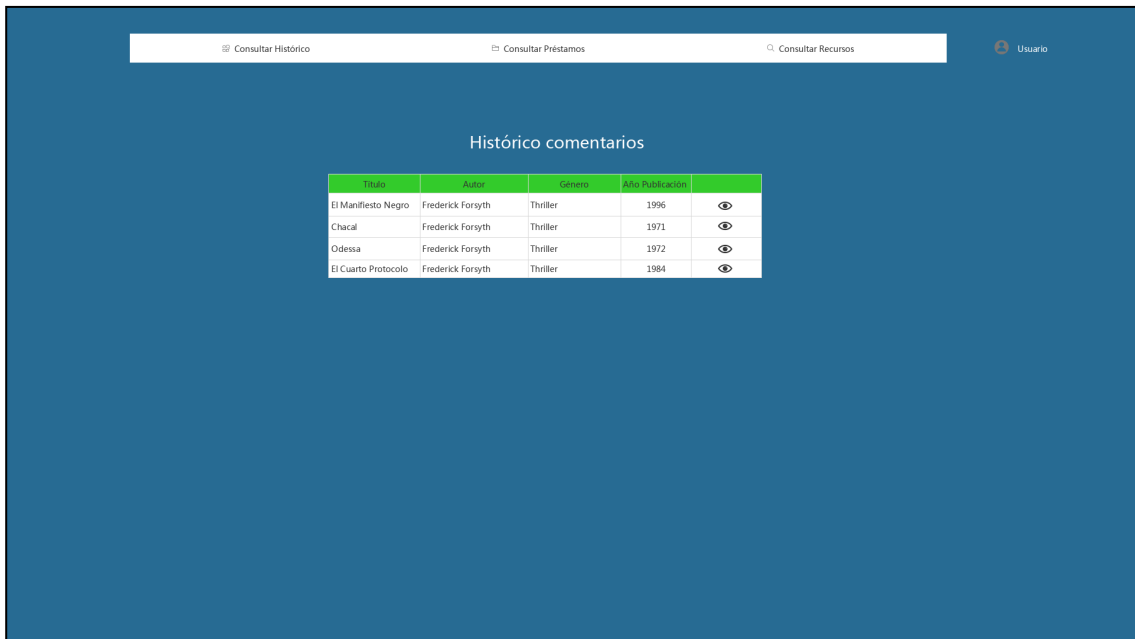
Modificar Datos Personales (CU-020)

The screenshot shows a web interface with a dark blue background. At the top, there is a navigation bar with three links: 'Consultar Histórico', 'Consultar Préstamos', and 'Consultar Recursos'. On the right side of the navigation bar, there is a user profile icon labeled 'Usuario'. Below the navigation bar, the main heading is 'Modificar Datos'. Underneath this heading is a form with several input fields. The fields are labeled 'Nombre', 'Apellido', 'Email', 'NIF', 'Domicilio', and 'Teléfono'. Each field contains a value. Below the form, there are two buttons: 'Alta Suscripción' and 'Cambiar Password'.

Nombre	Pedro
Apellido	Jiménez Sabater
Email	pjimenez@gmail.com
NIF	37129834Z
Domicilio	Gran Via, 312.4º. 1ª
Teléfono	651283104

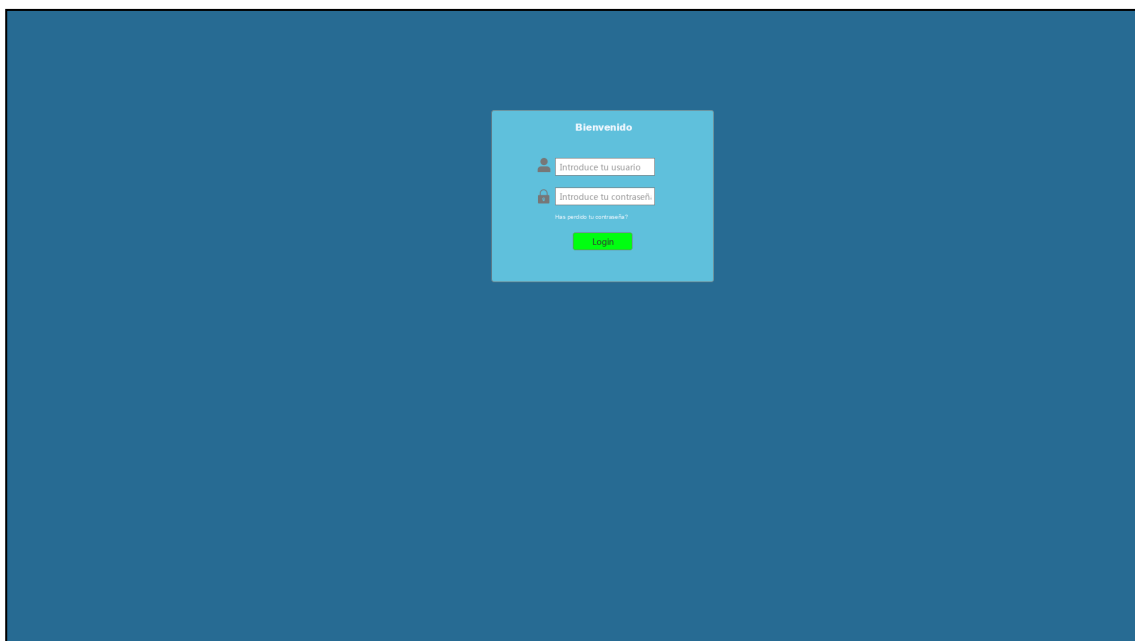
28 - Prototipo de Pantalla: Modificar los Datos Personales de un Usuario

Consultar Comentarios (CU-022)



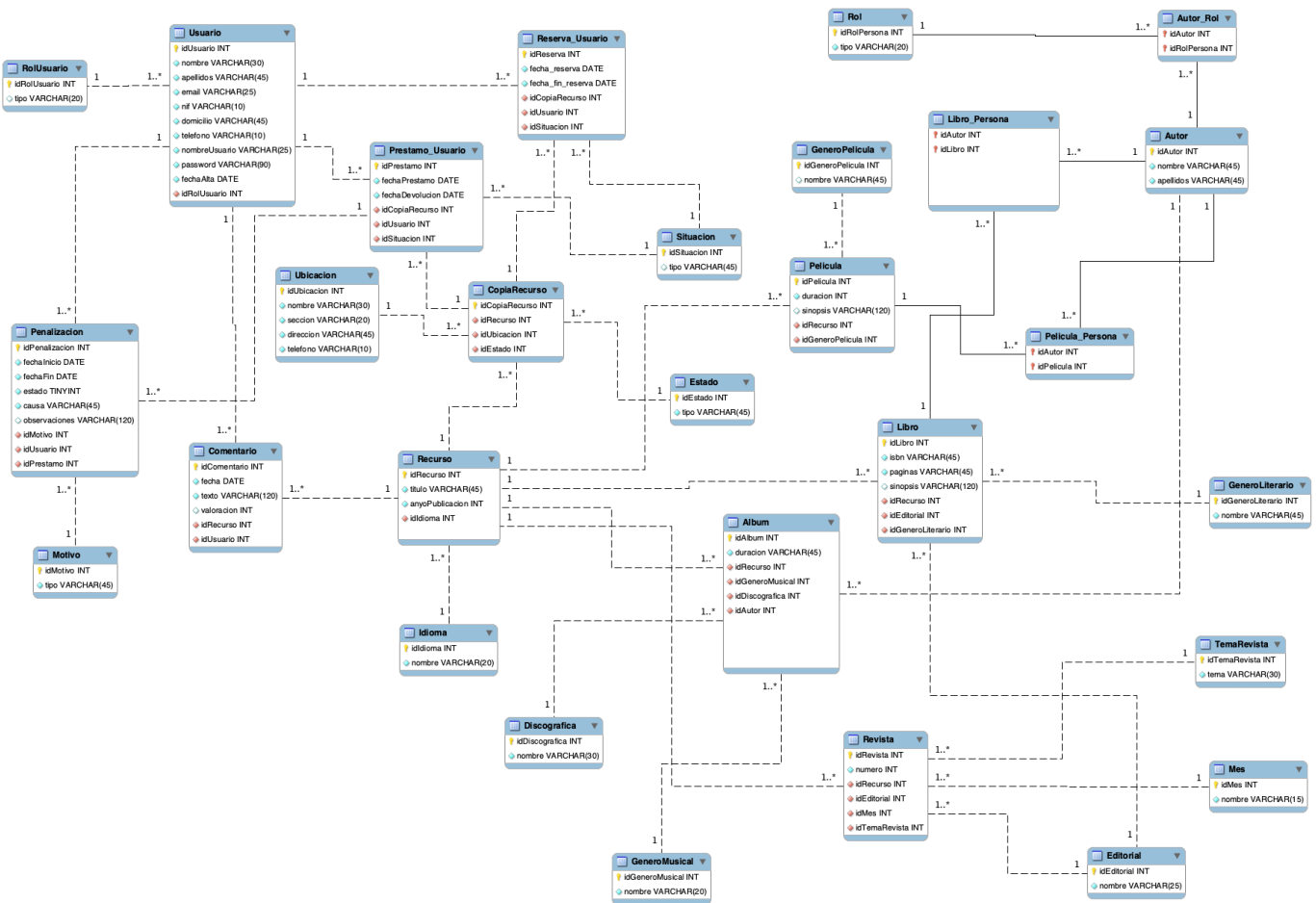
29 - Prototipo de Pantalla: Consultar Comentarios de un Usuario

Iniciar sesión (CU-023)



30 - Prototipo de Pantalla: Inicio de Sesión

2.4 Modelo relacional de la base de datos



31 - Modelo Relacional de la Base de Datos

resource (id,published_year,title)

copy (id,dismissal_date,registration_date,id_location,id_resource,id_status)

{id_location} es clave foránea de location

{id_resource} es clave foránea de resource

{id_status} es clave foránea de status_copy

location (id,address,name,phone)

status_copy (id,type)

album (id,duration,id_resource,id_genre,id_musician,id_record_company)

{id_resource} es clave foránea de resource

{id_genre} es clave foránea de music_genre

{id_musician} es clave foránea de autor

{id_record_company} es clave foránea de record_company

record_company (id,name)

music_genre (id,name)

review (id,comment,date,rating,title,id_resource,id_user)

{id_resource} es clave foránea de resource

{id_user} es clave foránea de user

user(id,address,email,name,nif,password,phone,registration,surnames,
username,suspended, id_user_role)

{id_user_role} es clave foránea de user_role

user_role (id,type)

penalty (id,comments,end_date,start_date,status,id_loan,id_reason,
id_user)

{id_loan} es clave foránea de transaction

{id_reason} es clave foránea de penalty_reason

{id_user} es clave foránea de user

penalty_reason (id,reason)

transaction (id,type,end_date,start_date,id_copy,id_status,id_user)

{id_copy} es clave foránea de copy

{id_status} es clave foránea de transaction_status

{id_user} es clave foránea de user

transaction_status (id,type)

movie (id,duration,synopsis,id_resource,id_director,id_genre,id_language)

{id_resource} es clave foránea de resource

{id_director} es clave foránea de autor

{id_genre} es clave foránea de movie_genre

{id_language} es clave foránea de language

movie_genre (id,name)

movie_actor (id_movie,id_actor)

{id_movie} es clave foránea de movie

{id_actor} es clave foránea de author

language (id,name)

book (id,isbn,pages,synopsis,id_resource,
id_genre,id_language,id_publisher)

{id_resource} es clave foránea de resource

{id_genre} es clave foránea de literary_genre

{id_language} es clave foránea de language

{id_publisher} es clave foránea de book_publisher

book_publisher (id,**name**)

book_writer (**id_book**,**id_writer**)

{id_book} es clave foránea de book

{id_writer} es clave foránea de autor

literary_genre (id,**name**)

author (id,**id_role**,**name**,surnames)

{id_role} es clave foránea de role

role (id,**type**)

magazine (id,**number**,**pages**,**id_resource**,
id_language,**id_month**,**id_publisher**,**id_subject**)

{id_resource} es clave foránea de resource

{id_language} es clave foránea de language

{id_month} es clave foránea de month

{id_publisher} es clave foránea de magazine_publisher

{id_subject} es clave foránea de magazine_subject

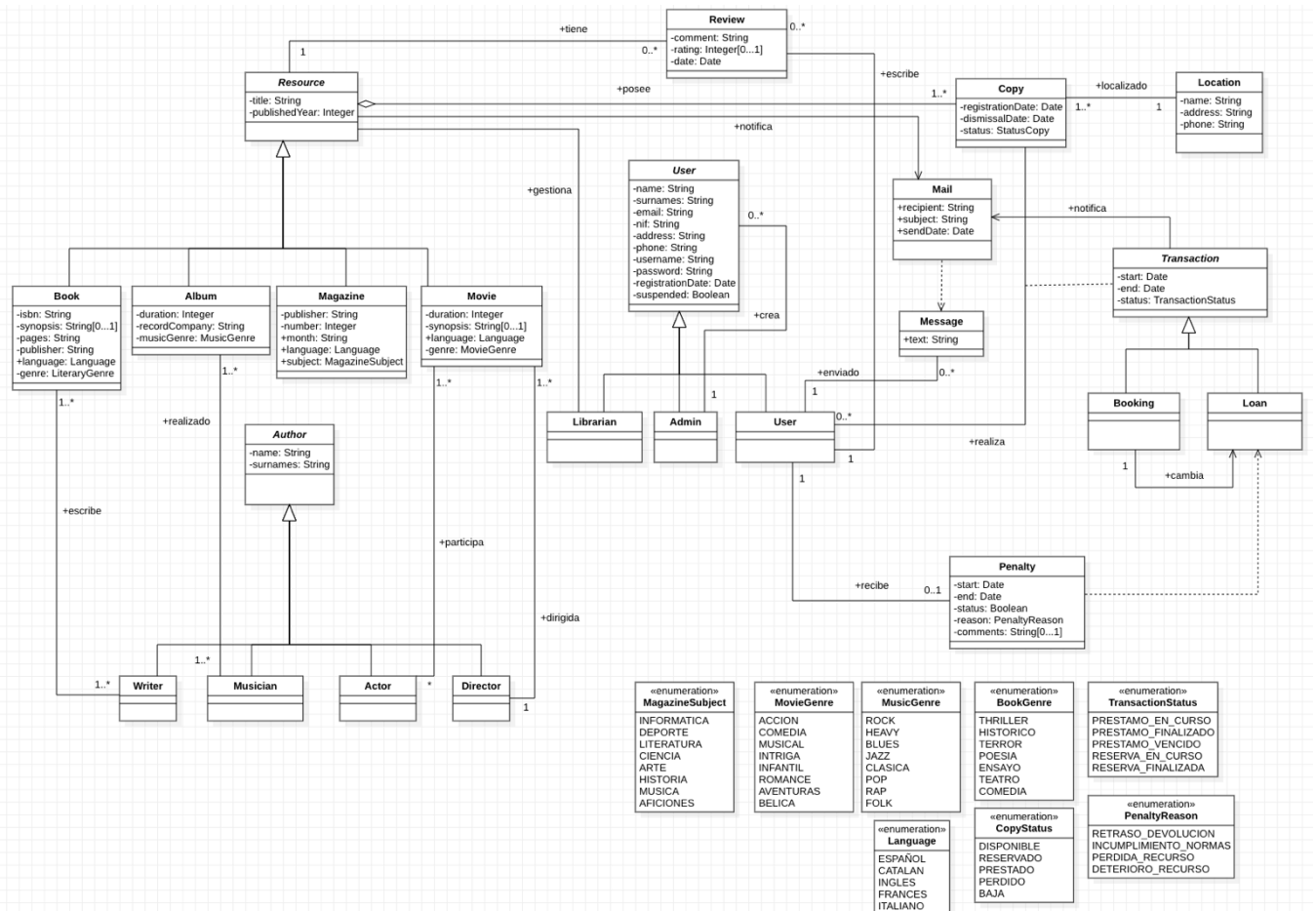
month (id,**name**)

magazine_publisher (id,**name**)

magazine_subject (id,**name**)

Los campos en negrita son campos que no permiten valores nulos.

2.5 Diagrama de clases



32 - Diagrama de Clases

Restricciones

Un usuario con penalización no puede tomar recursos prestados ni reservarlos.
 Un usuario no puede tomar prestados más de 5 recursos de forma simultánea.

2.6 Diagrama de arquitectura

La aplicación seguirá el modelo arquitectónico por capas. Uno de los principales motivos para utilizar este patrón es separar las lógicas de la parte de la interfaz gráfica de la aplicación de la lógica de la parte de negocio y de persistencia de la aplicación.

De esta forma, tendremos diferentes capas las cuales se encargarán de diferentes funciones, además estarán totalmente desacopladas, lo cual facilita el mantenimiento de la aplicación en el futuro, puesto que tenemos pensado utilizar el framework Angular para realizar la vista, pero si en el futuro queremos

cambiar a otro framework como Vue, React o cualquier otro que pueda surgir en el futuro, esto no afectaría al resto de capas de la aplicación.

La arquitectura por capas se estructura en capas horizontales, y cada una de ellas se encarga de una función concreta dentro de la aplicación. Habitualmente se utilizan 4 capas: Presentación, Negocio, Persistencia y Base de Datos, aunque en muchos casos estas dos últimas capas se juntan en una sola, debido a la utilización de lógica de persistencia como HSQL o JSQL está incorporada en los componentes de la capa de negocio. Hay que destacar que la comunicación entre capas es inmediata, es decir, la capa de presentación solo se comunica con la de negocio y viceversa, la capa de negocio solo se comunica con la de persistencia, y viceversa.

Presentación: Se ocupa de las vistas a mostrar al usuario, así como de las interacciones y solicitudes del usuario las cuales debe trasladar a la capa de negocio.

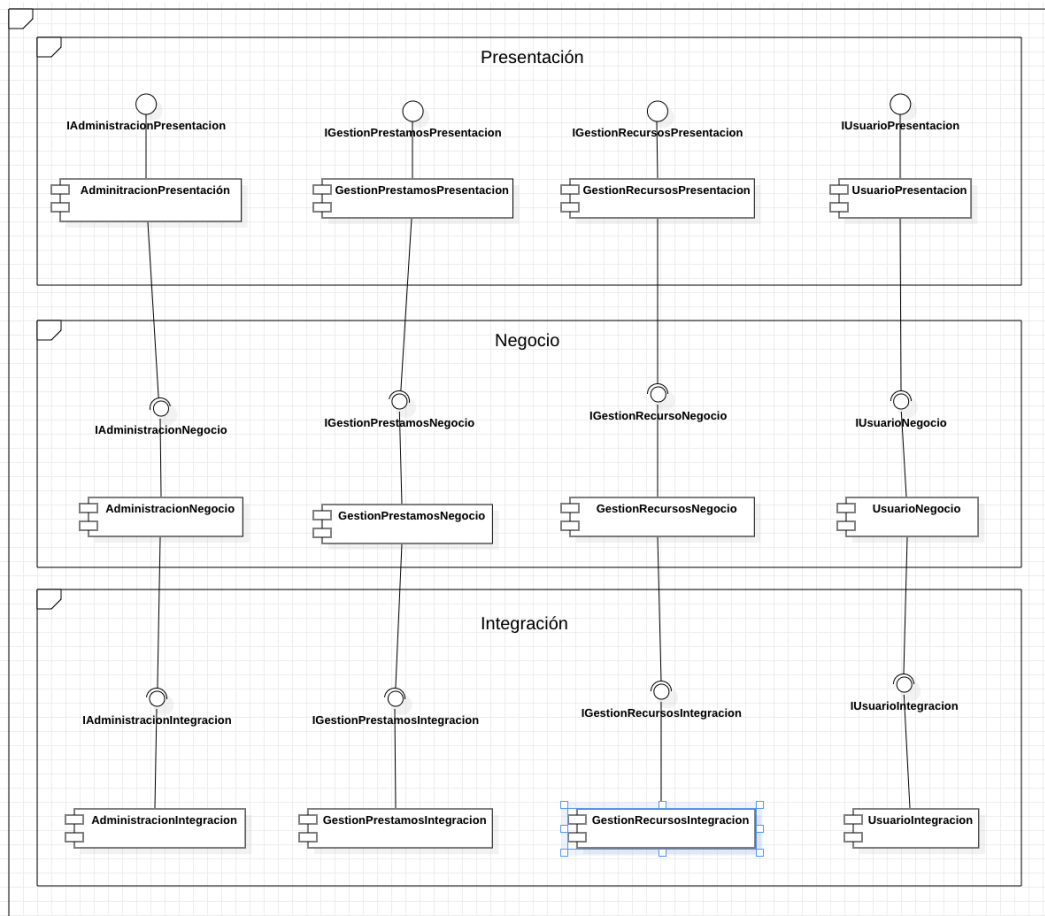
Negocio: Esta capa se encarga de toda la lógica de negocio y de trasladar sus peticiones para la obtención o envío de datos con la capa de persistencia.

Persistencia: Esta capa se ocupa de gestionar los datos y de guardarlos en la base de datos.

Otro motivo para la elección de este patrón arquitectónico es que es relativamente sencillo de implementar y una optima elección para aplicaciones pequeñas o para aplicaciones web, como es nuestro caso, que tenemos pensado hacer una aplicación pequeña del tipo SPA (*Single Page Application*).

Por último, el hecho de tener un plazo de tiempo ajustado para el diseño e implementación de la aplicación hace ideal a esta arquitectura de capas para aplicarla a nuestro proyecto.

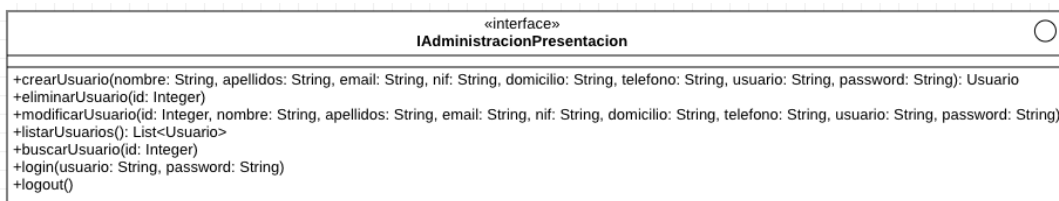
Por tanto, el diagrama de componentes nos queda de la siguiente manera:

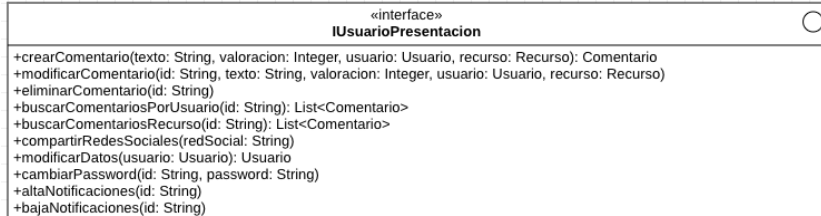
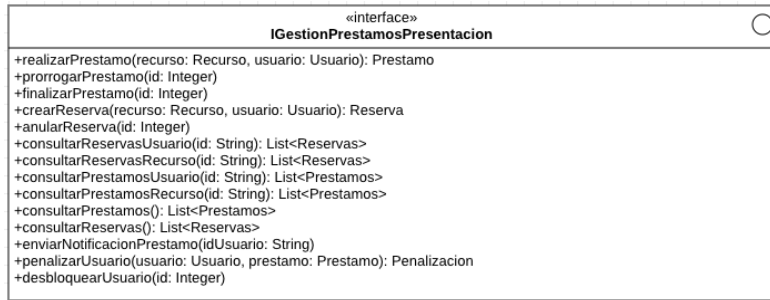


33 - Diagrama de Componentes

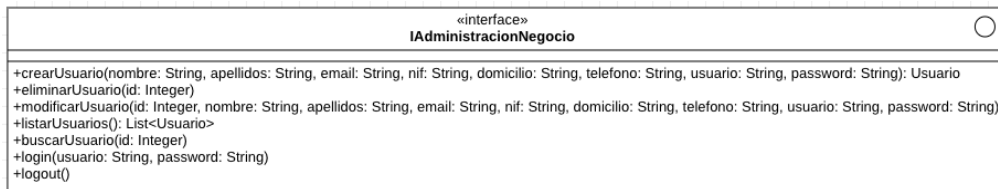
Ahora, vamos a mostrar el contenido de las diferentes interfaces:

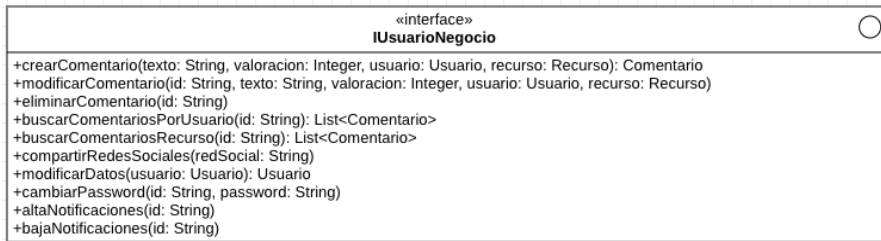
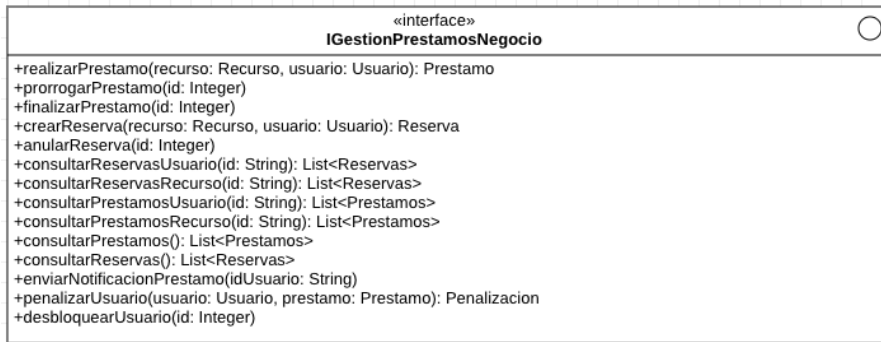
Capa de Presentación



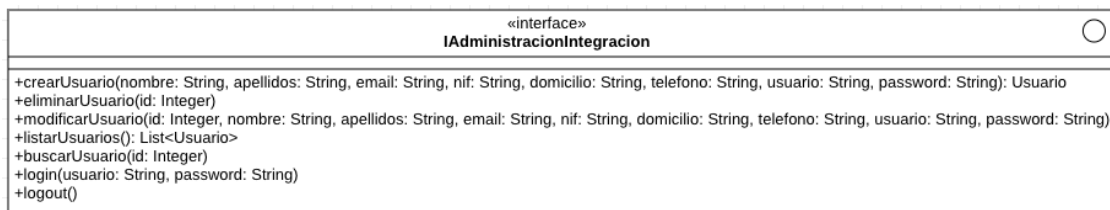


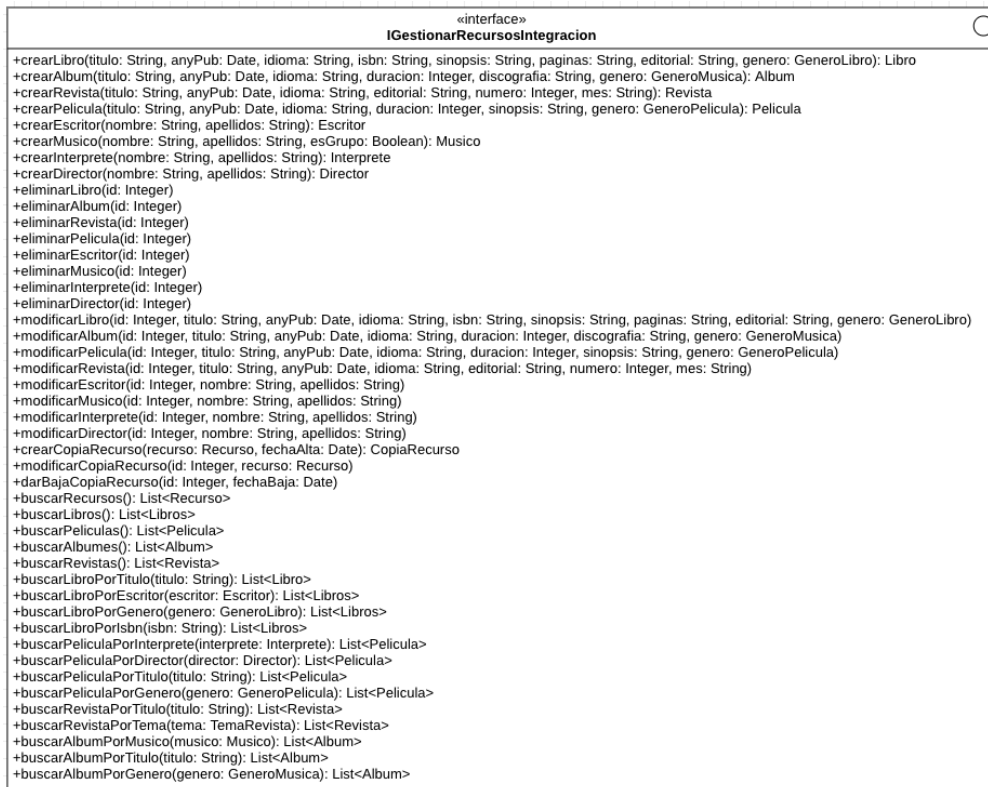
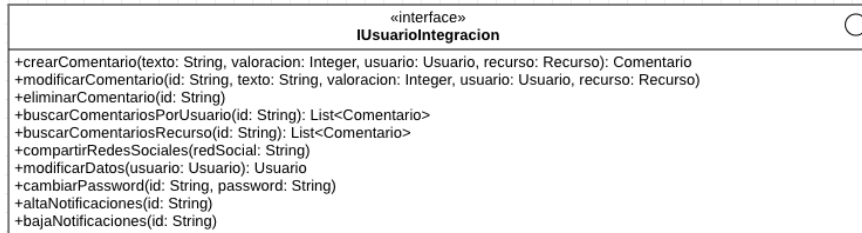
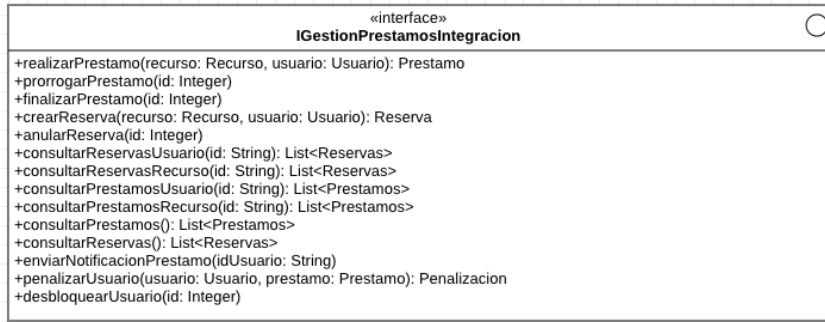
Capa de Negocio





Capa de Integración





3. Implementación

3.1 Tecnologías utilizadas

A continuación, mencionamos en un pequeño borrador algunos detalles de las tecnologías utilizadas en nuestro proyecto.

Frontend

Angular

Es un framework para aplicaciones web SPA (Single Page Application). Es de código abierto y ha sido desarrollado por Google.

Se basa en la utilización de componentes reutilizables, la inyección de dependencias y en el uso del patrón MVC (Modelo-Vista-Controlador).

Aunque se considera un Framework de JavaScript, se utiliza como código de desarrollo TypeScript que es un superset o extensión de JavaScript, cuya principal característica es ofrecer el tipado estricto. Además, acepta por supuesto, el uso de CSS o HTML, aunque ofrece gran cantidad de directivas para incluir en el código HTML y facilitar el trabajo, directivas como ngFor o ngIf entre otras.

Su web oficial es:

<https://angular.io/>

Prime NG

Es una biblioteca de componentes de código abierto. Nos ofrece muchos componentes para nuestra interfaz de usuario, ahorrándonos mucho trabajo de código y facilitándonos el uso de componentes como listas, tablas, componentes calendario, listas desplegables (dropdown), botones de opciones (Radio Button), barras de navegación (BarMenu) y otros componentes bastante habituales en este tipo de aplicaciones web. Nos ha servido para hacer una interfaz que de otra manera no habría sido posible, ya que habría requerido mucho más nivel de conocimientos HTML, CSS y Javascript, además de una cantidad de horas mayor.

Su web oficial es:

<https://primeng.org/>

Backend

Spring

Es un framework de Java, que nos facilita bastante la vida ofreciendo una serie de componentes y utilidades. Algunas de sus características principales son:

La inyección de dependencias, el uso del patrón MVC (Modelo-Vista-Controlador), el uso de ORM con Hibernate para el acceso de datos y persistencia JPA o el uso de inversión de control. Permite el uso de gestores de dependencias como Maven o Gradle.

Spring Boot

Es una extensión de Spring que sirve para facilitar el proceso de instalación y configuración de los proyectos de Spring. Permite ahorrar mucho tiempo y automatizar la configuración, lo que permite centrarse en el desarrollo de aplicaciones con lo cual nos permite empezar sin más dilación el proyecto.

Hibernate

Es un framework ORM o mapeo objeto-relacional que sirve para simplificar el mapeo entre la base de datos relacional y el modelo de objetos de una aplicación Java. Hay que destacar que este framework se basa en JPA.

Esto facilita mucho el trabajo y permite evitar el uso de SQL para la realización de consultas y ahorrándonos tener que trabajar con la base de datos a bajo nivel.

Gracias a esto se simplifica la implementación de las operaciones CRUD en nuestra aplicación de Spring.

Jasypt

Es una librería de java que nos permite implementar operaciones de encriptado y desencriptado en nuestra aplicación. Es posible utilizar diversos algoritmos de encriptación en una aplicación Java gracias a esta librería. Nosotros lo hemos utilizado para el cifrado de las contraseñas de los usuarios a la hora de registrarse y de autenticarse en la aplicación.

MySQL

Es un SGDB para bases de datos relacionales de los más utilizados por los desarrolladores. Es el que hemos usado para gestionar la base de datos de nuestro proyecto.

Spring Boot Starter Mail

Es una dependencia de Spring Boot que nos permite implementar la funcionalidad de avisos de correo electrónico para los usuarios. Concretamente nos permite utilizar la interface JavaMailSender que en realidad nos posibilita utilizar la librería JavaMail que permite enviar y recibir correo electrónico mediante nuestro código Java empleando los distintos protocolos existentes: SMTP, POP3 y IMAP.

En nuestra aplicación lo hemos utilizado para avisar por correo al usuario cuando queda 1 día para vencer el préstamo de un recurso, o bien para avisar cuando un usuario tiene una reserva y el recurso ya ha sido devuelto para que pueda ir a recoger el recurso a la biblioteca.

Swagger 3 (OpenAPI3)

Es una herramienta de código abierto que permite documentar servicios RESTful. Lo hemos utilizado para probar de forma más rápida y fácil nuestros puntos de acceso de la aplicación.

Se puede acceder mediante navegador con la siguiente url:

[http://localhost:8080/swagger-ui/index.html/](http://localhost:8080/swagger-ui/index.html#/)

Puesto que hemos instalado la dependencia en nuestro proyecto Spring.

Lombok

Es una librería que nos sirve mediante anotaciones para simplificar el código en nuestras clases de Java. En concreto, nos ahorra crear los constructores, los getter y setter, y métodos típicos como equals o toString.

3.2 Repositorios del proyecto

Estos son los repositorios del código del proyecto cultusfera.

Repositorio Frontend:

<https://github.com/pcharlesmorphy/cultusfera-frontend>

Repositorio Backend:

<https://github.com/pcharlesmorphy/cultusfera-backend>

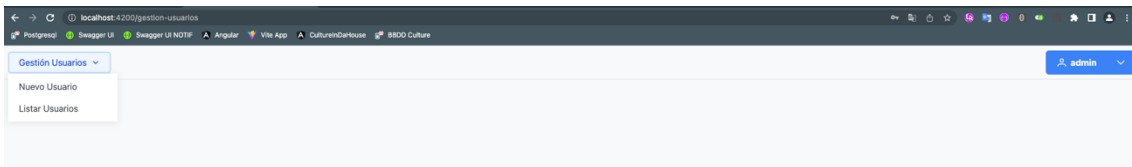
3.3 Características de la aplicación

Mencionamos a grandes rasgos las características implementadas en la aplicación.

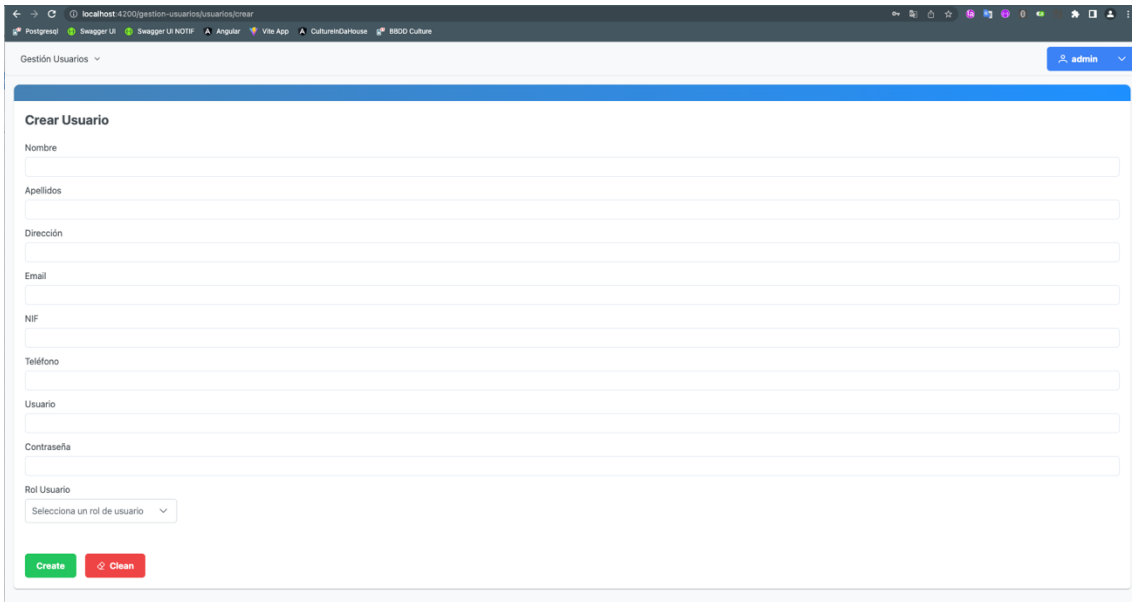
Hay que remarcar que existen tres tipos de roles de usuario en la aplicación, y esto es lo que determina las funcionalidades disponibles:

Rol Admin (Administrador)

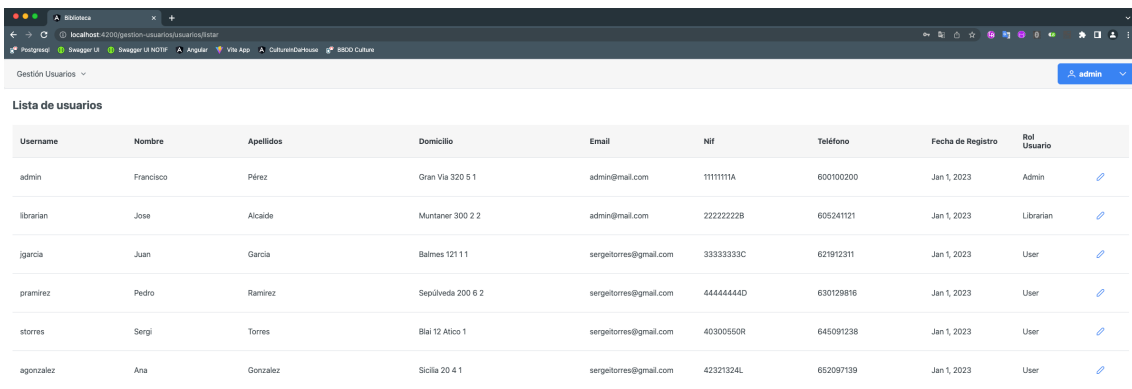
Hemos implementado las funcionalidades de crear usuarios, modificar usuarios y listar usuarios. No hemos implementado la opción de eliminar usuarios, ya que no queremos eliminar de la base de datos los usuarios. Quedaría implementar en un futuro una funcionalidad para marcar un usuario como inactivo.



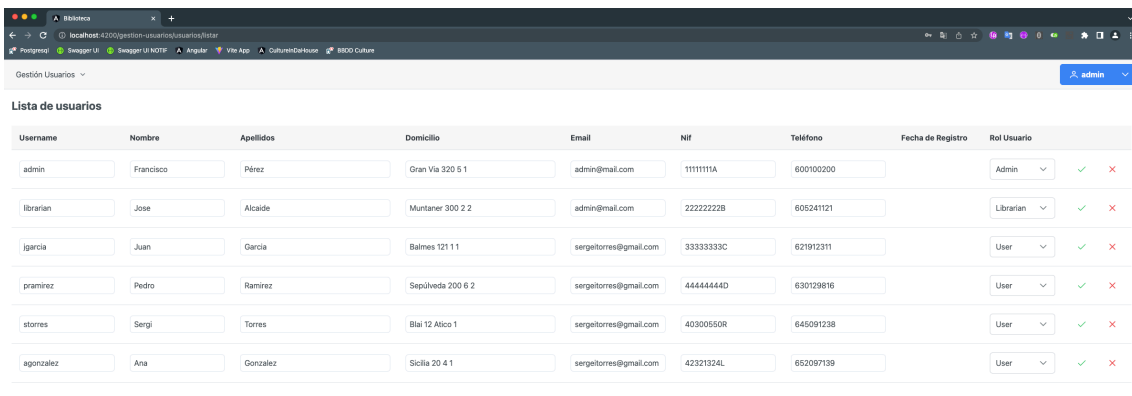
34 - Pantalla Principal de Admin



35 - Crear Usuario



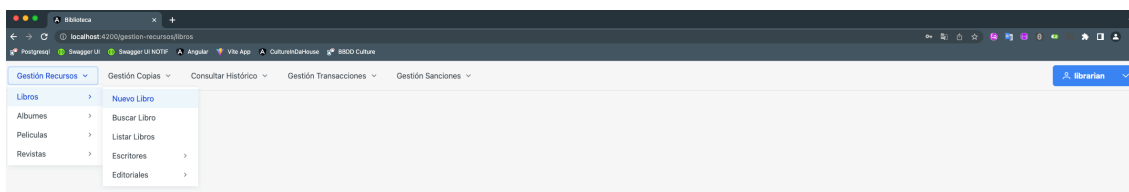
36 - Lista de los usuarios existentes



37 - Modificar un Usuario

Rol Librarian (Bibliotecario)

Es el rol con más funcionalidades de la aplicación. De hecho, es el corazón de la aplicación.



38 - Pantalla Principal del Bibliotecario

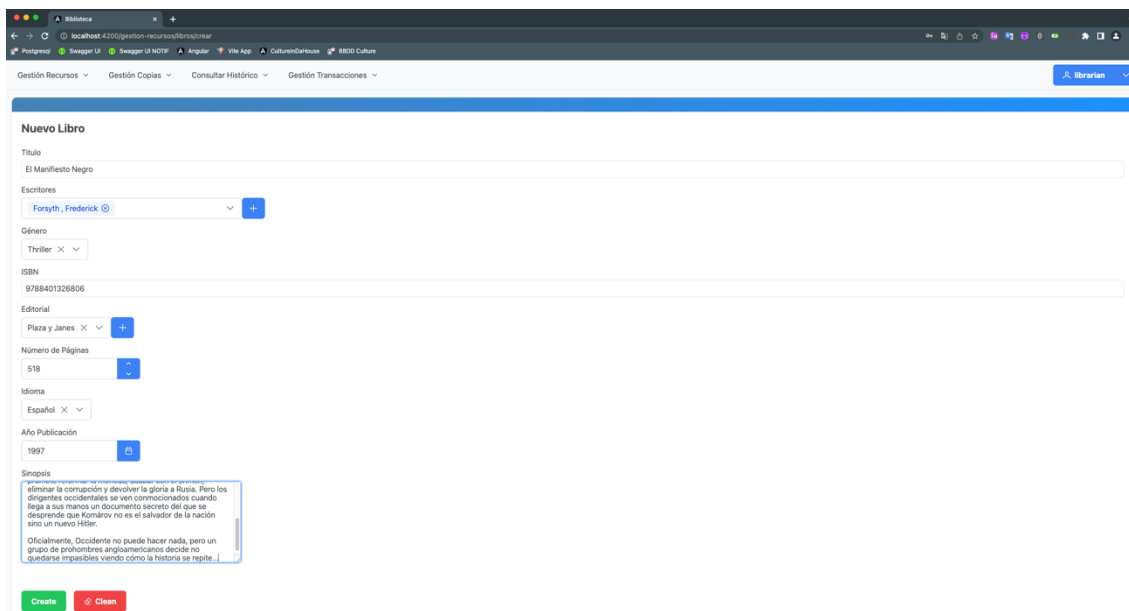
Las funcionalidades implementadas son:

Gestión de recursos:

Existen 4 recursos: Libros, Películas, Álbumes y Revistas.

Tenemos implementado el CRUD completo para crear, modificar, borrar y listar estos recursos. Además, también tenemos CRUD completo para hacer lo propio con escritores, directores, actores, músicos, editoriales de los libros y revistas, y de las discográficas.

Por otra parte, también tenemos implementada la búsqueda de estos recursos por diferente criterios como título, autor, género, idioma.



39 - Creación de un Nuevo Libro

Título	Autor	ISBN	Páginas	Editorial	Género	Idioma
Cien años de soledad	García Márquez, Gabriel	18298192	200	Planeta	Novela	Español
El Manifiesto Negro	Forsyth, Frederick	9788401326806	518	Plaza y Janes	Thriller	Español

Showing 1 to 2 of 2 entries

40 - Listado de los Libros Existentes

Ficha Libro

El Manifiesto Negro

Escritor: Frederick Forsyth
 Género: Thriller
 Páginas: 518
 Año Publicación: 1997
 Idioma: Español
 Editorial: Plaza y Janes
 ISBN: 9788401326806
 Rating: ☆☆☆☆☆

Sinopsis

«Era el verano en que el precio de una barra de pan alcanzó el millón de rublos...» Esta frase nos introduce en la convulsa Rusia de 1999. Esta gran superpotencia zozobra en un mar de hiperinflación, caos económico y criminalidad. Las elecciones se aproximan y una sola voz carismática resuena en todo el país: Igor Komárov, líder derechista, promete reformar la moneda, acabar con el crimen, eliminar la corrupción y devolver la gloria a Rusia. Pero los dirigentes occidentales se ven conmovidos cuando llega a sus manos un documento secreto del que se desprende que Komárov no es el salvador de la nación sino un nuevo Hitler. Ojalá, Occidente no puede hacer nada, pero un grupo de prohombres angloamericanos decide no quedarse impasibles viendo cómo la historia se repite...

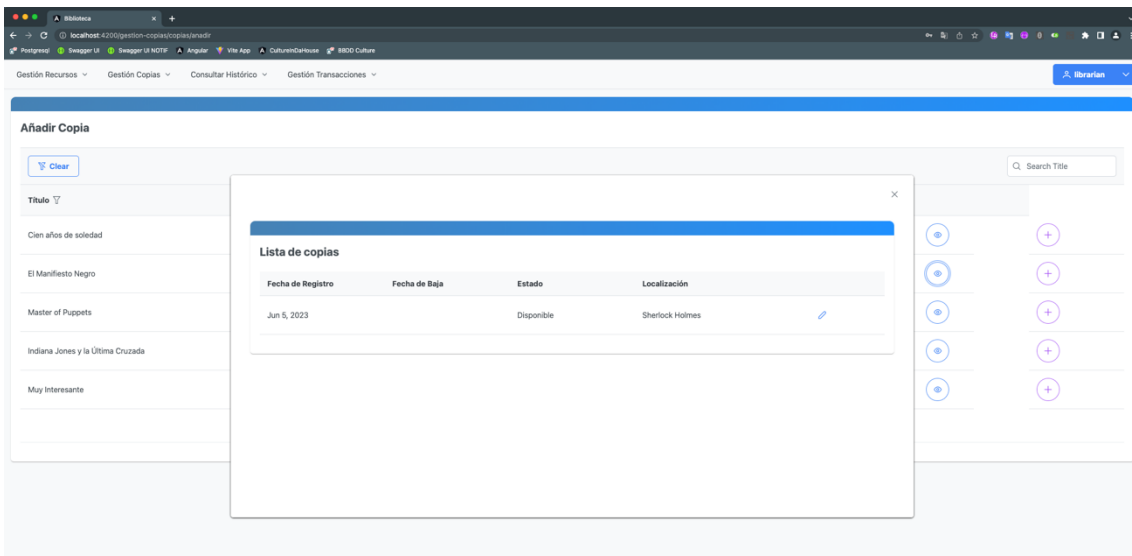
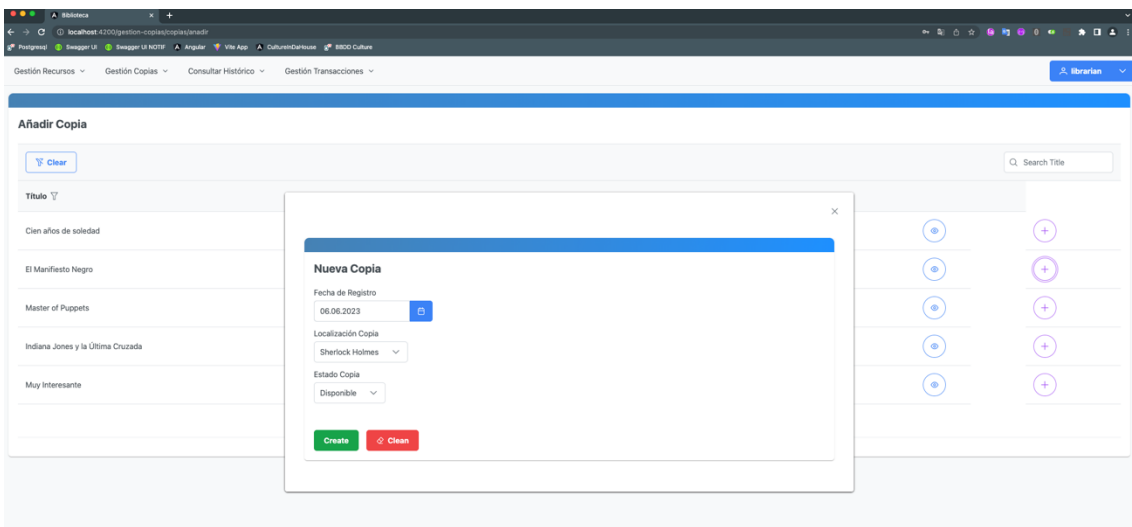
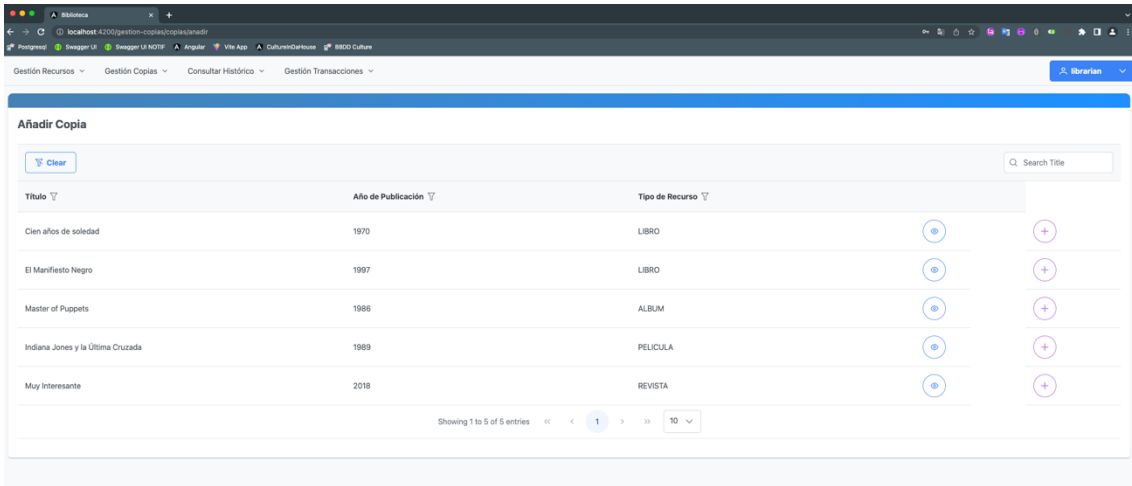
Copias

Id	Fecha Alta	Fecha Baja	Estado	Localización
----	------------	------------	--------	--------------

41 - Consulta Ficha del Libro

Gestión de copias:

Aquí tenemos implementado por una parte la posibilidad de añadir copias de cada uno de los recursos que tengamos, además de tener la posibilidad de editar los datos de cada copia.



Gestión de Transacciones

Aquí hemos implementado que se vea el listado de recursos y a su vez sus copias, para poder hacer las diferentes operaciones en cada copia: Reservar una copia, de momento solo permitimos realizar una reserva cuando un recurso esta en préstamos, pero sería interesante en un futuro implementar una cola de reservas.

Por otro lado, también implementamos los préstamos, con un máximo de 5 por usuario, además no permitimos reservar para un usuario si este es el titular del préstamo en curso.

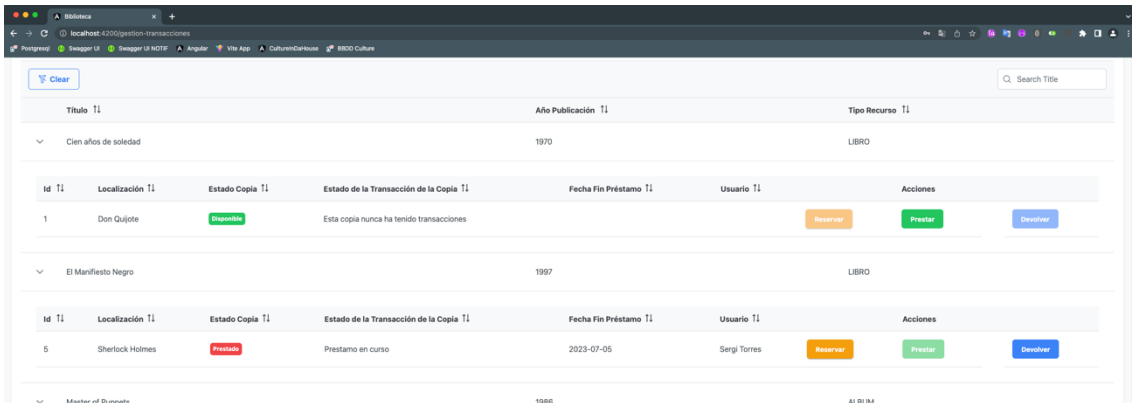
También ofrecemos la implementación de cancelar la reserva y de devolver el recurso, donde se comprueba al devolver el recurso si existe una reserva de la copia, y en caso afirmativo se envía un correo de aviso al usuario que tiene la reserva.

Título	Año Publicación	Tipo Recurso				
Cien años de soledad	1970	LIBRO				
Id	Localización	Estado Copia	Estado de la Transacción de la Copia	Fecha Fin Préstamo	Usuario	Acciones
1	Don Quijote	Disponible	Esta copia nunca ha tenido transacciones			Reservar Prestar Devolver
El Manifiesto Negro	1997	LIBRO				
5	Sherlock Holmes	Disponible	Esta copia nunca ha tenido transacciones			Reservar Prestar Devolver
Master of Puppets	1986	ALBUM				
2	Hamlet	Disponible	Esta copia nunca ha tenido transacciones			Reservar Prestar Devolver
Indiana Jones y la Última Cruzada	1989	PELICULA				

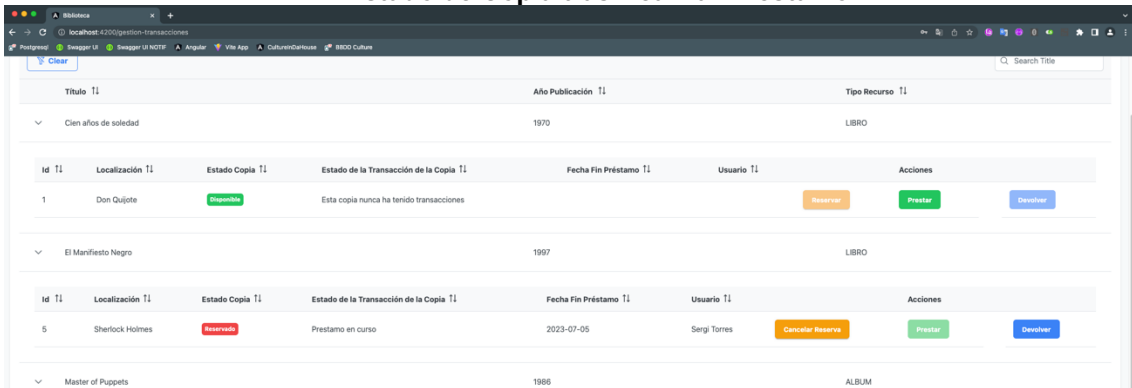
45 - Listar Recursos con sus Copias

Título	Año Publicación	Tipo Recurso				
Cien años de soledad	1970	LIBRO				
Id	Localización	Estado Copia	Estado de la Transacción de la Copia	Fecha Fin Préstamo	Usuario	Acciones
1	Don Quijote	Disponible	Esta copia nunca ha tenido transacciones			Reservar Prestar Devolver
El Manifiesto Negro	1997	LIBRO				
5	Sherlock Holmes	Disponible	Esta copia nunca ha tenido transacciones			Reservar Prestar Devolver
Master of Puppets	1986	ALBUM				
2	Hamlet	Disponible	Esta copia nunca ha tenido transacciones			Reservar Prestar Devolver
Indiana Jones y la Última Cruzada	1989	PELICULA				
3	Don Quijote	Disponible	Esta copia nunca ha tenido transacciones			Reservar Prestar Devolver

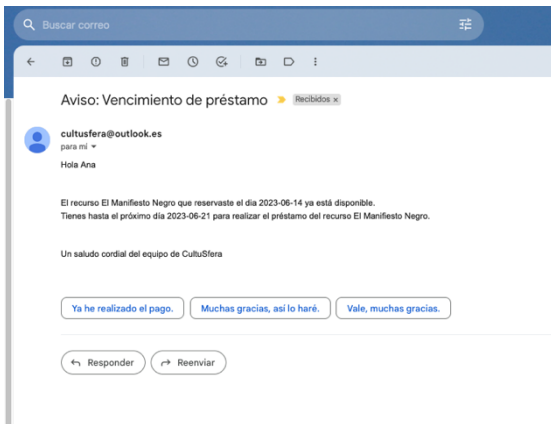
46 - Realizar Préstamo de una Copia de un Libro a un Usuario



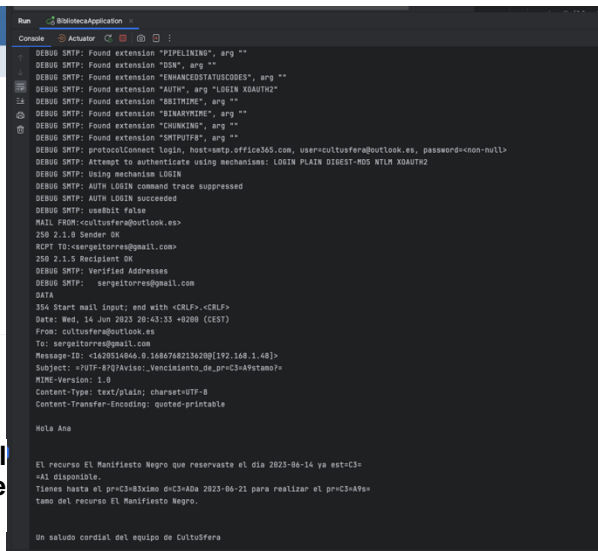
47 - Estado de Copia tras Realizar Préstamo



48 - Reserva de Libro en Préstamo a un Usuario



49 - Correo de Aviso Recibido por el Usuario tras Devolución de un Libro que tiene en Reserva



50 - Debug del Correo Enviado

Título	Año Publicación	Tipo Recurso				
Cien años de soledad	1970	LIBRO				
Id	Localización	Estado Copia	Estado de la Transacción de la Copia	Fecha Fin Préstamo	Usuario	Acciones
1	Don Quijote	En reserva	Esta copia nunca ha tenido transacciones			Reservar, Prestar, Devolver
El Manifiesto Negro	1997					
Id	Localización	Estado Copia	Estado de la Transacción de la Copia	Fecha Fin Préstamo	Usuario	Acciones
5	Sherlock Holmes	Reservado	Préstamo finalizado	2023-07-05	Sergi Torres	Cancelar Reserva, Prestar, Devolver
Master of Puppets	1986					

51 - Libro Devuelto y en Reserva hasta que el Usuario lo Recoja en Préstamo

Aún así , pensamos que la gestión de estas transacciones es mejorable y se puede pulir para que tenga un mejor funcionamiento.

También se ha implementado, una función automatizada para que cada día se compruebe si existen préstamos a punto de finalizar. En caso, de existir préstamos que van a finalizar al siguiente día, se envía un correo electrónico de recordatorio al usuario conforme debe devolver el recurso en préstamo.

Quedaría por implementar la gestión de las copias en caso de baja o de pérdida.

Consultar Histórico:

Aquí tenemos implementado por un lado ver el histórico de préstamos por usuario o bien por recurso, y con un checkbox, permitimos si el bibliotecario desea, ver solo la lista de préstamos en curso por usuario o por recurso.

Título	Año Publicación	Tipo Recurso				
Cien años de soledad	1970	LIBRO				
Id	Localización	Estado Copia	Estado de la Transacción de la Copia	Fecha Fin Préstamo	Usuario	Acciones
1	Don Quijote	Préstado	Préstamo en curso	2023-07-05	Sergi Torres	Reservar, Prestar, Devolver
El Manifiesto Negro	1997					
Id	Localización	Estado Copia	Estado de la Transacción de la Copia	Fecha Fin Préstamo	Usuario	Acciones
5	Sherlock Holmes	Reservado	Préstamo finalizado	2023-07-05	Sergi Torres	Cancelar Reserva, Prestar, Devolver
Master of Puppets	1986					

52 - Estado Actual de Préstamos y Reservas

Historico de Préstamos de Sergi Torres

Inserta el texto
Torres, Sergi

Ver sólo los Préstamos en curso

Buscar Clean

Título Copia	Año Publicación	Tipo Recurso	Localización Copia	Fecha Inicio	Fecha Fin	Usuario	Estado
El Manifiesto Negro	1997	Libro	Sherlock Holmes	2023-06-14	2023-07-05	Sergi Torres	Prestamo finalizado

Showing 1 to 1 of 1 entries

53 - Ver Histórico de Préstamos de un Usuario

Préstamos en curso de Sergi Torres

Inserta el texto
Torres, Sergi

Ver sólo los Préstamos en curso

Buscar Clean

Título Copia	Año Publicación	Tipo Recurso	Localización Copia	Fecha Inicio	Fecha Fin	Usuario	Estado
Cien años de soledad	1970	Libro	Don Quijote	2023-06-14	2023-07-05	Sergi Torres	Prestamo en curso

Showing 1 to 1 of 1 entries

54 - Ver Préstamos en Curso de un Usuario

Préstamos en curso de Cien años de soledad

Inserta el texto
Cien años

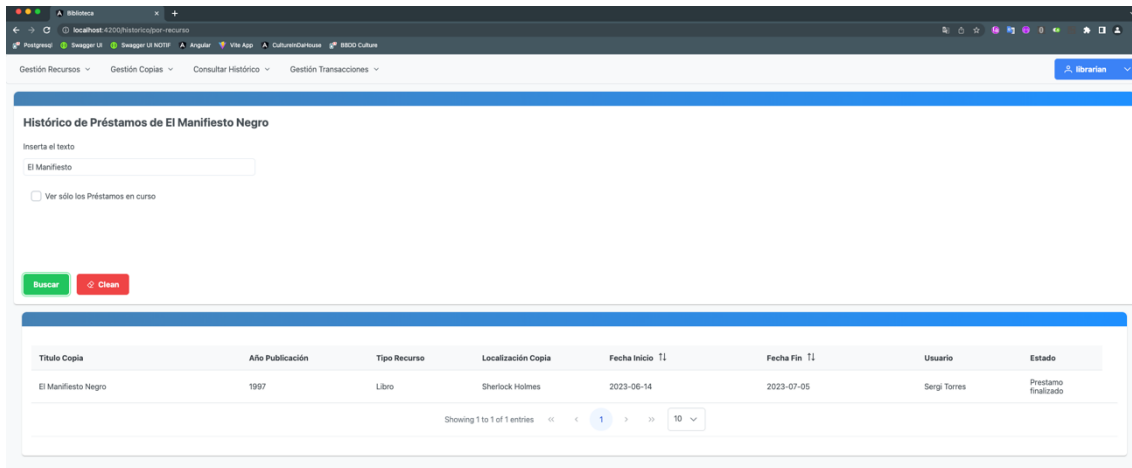
Ver sólo los Préstamos en curso

Buscar Clean

Título Copia	Año Publicación	Tipo Recurso	Localización Copia	Fecha Inicio	Fecha Fin	Usuario	Estado
Cien años de soledad	1970	Libro	Don Quijote	2023-06-14	2023-07-05	Sergi Torres	Prestamo en curso

Showing 1 to 1 of 1 entries

55 - Ver Préstamos en Curso de un Libro

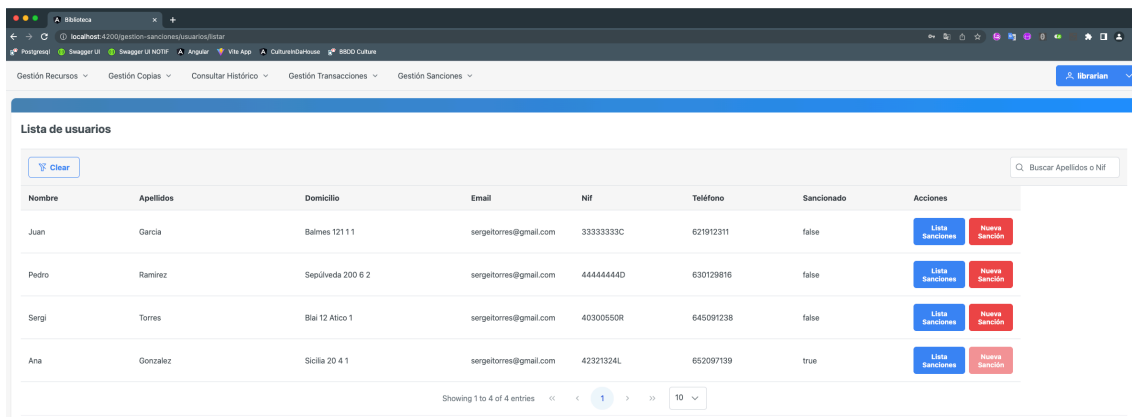


56 - Ver Histórico de Préstamos de un Libro

Gestión de sanciones:

Aunque aún se trata de una implementación básica que debe ser mejorada, y relacionarla con el préstamo relacionado, si es posible sancionar a un usuario, y con ello, no permitir que pueda tomar prestados recursos, ni reservar recursos. En un futuro, está previsto vincular sanciones con un préstamo determinado, e incluso un sistema de puntos o inclusión de multas. Ahora mismo, el único castigo es dejar al usuario sin tomar préstamos durante 30 días.

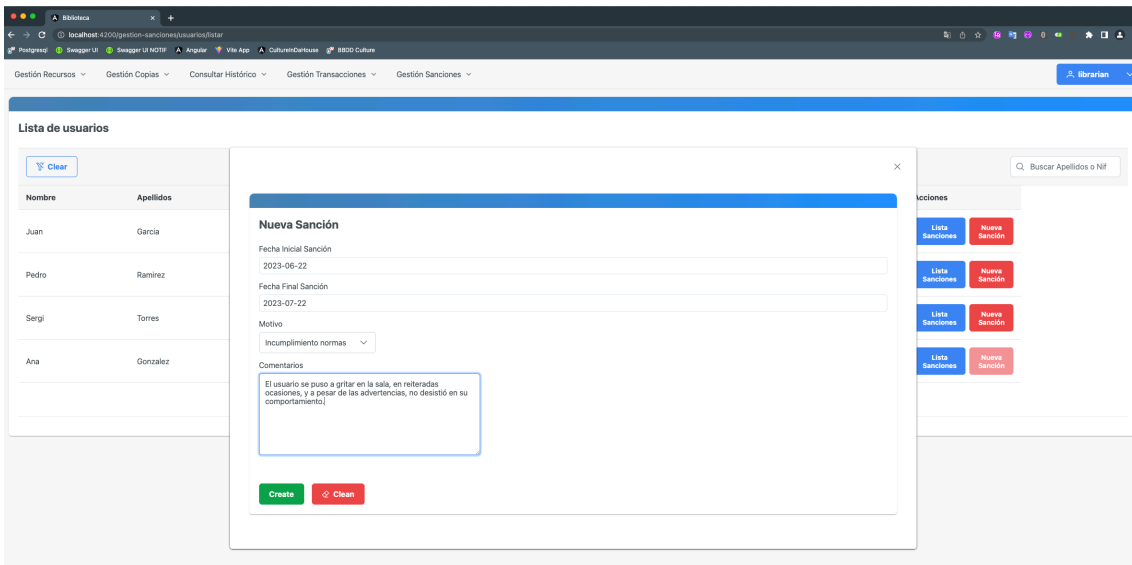
Se ha creado un método para que el sistema compruebe cada día las sanciones en curso y en caso de encontrarse en el día final, revertir la sanción y mandar un aviso por correo al usuario.



57 - Listado de Usuarios

Se pueden realizar dos acciones básicas. Por un lado, aplicar una nueva sanción. Hay que destacar, que en caso de que el usuario tenga una sanción vigente, ya se desactiva el botón ya que no es posible aplicar dos sanciones simultáneas al mismo usuario.

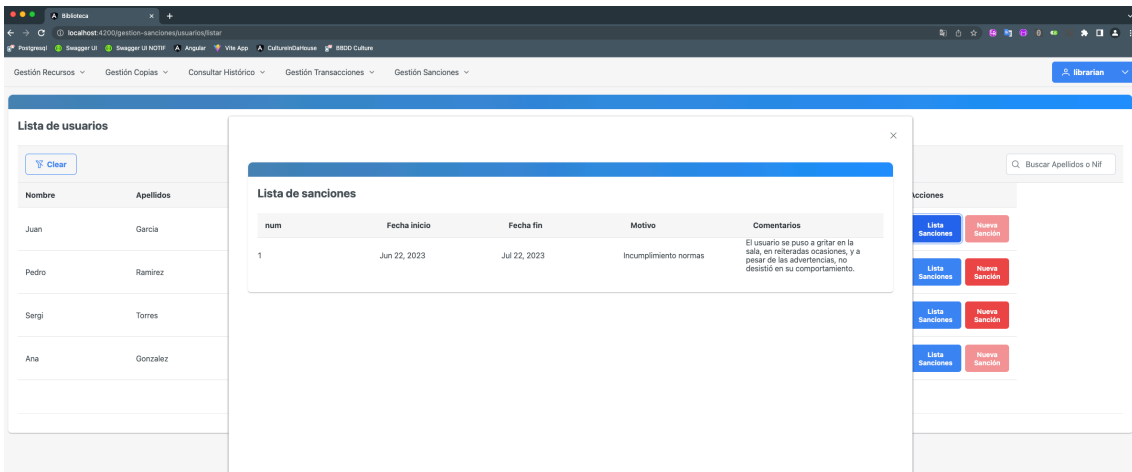
Al crear una sanción, las fechas de inicio y fin se establecen automáticamente, además se incluye el motivo y comentarios si es pertinente incluirlos.



58 - Creando una nueva sanción al usuario

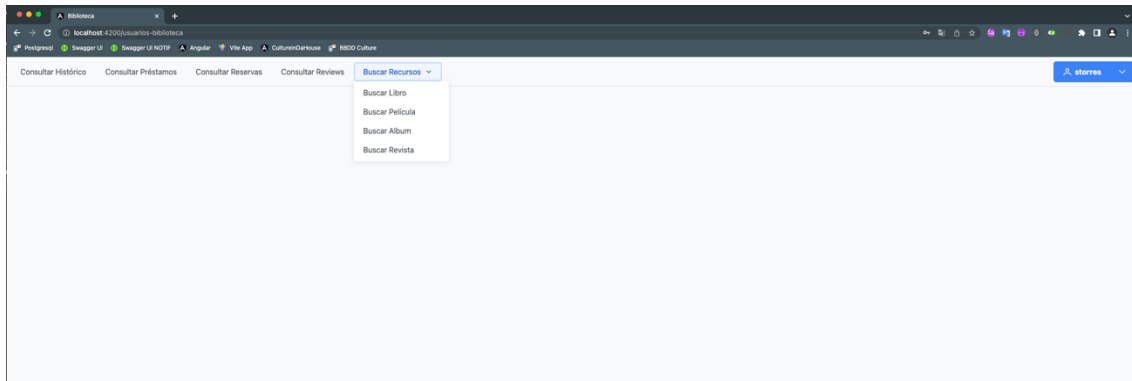
Una vez creada la sanción, el sistema envía un correo electrónico al usuario informándole de la sanción, las fechas de inicio y fin, así como el motivo.

Por otro lado, es posible ver la lista de sanciones de un usuario, aunque esto se tiene previsto mejorarlo en el futuro, ya que se encuentra en un estado primario, ya que sería interesante mejorar la visualización de cada sanción.



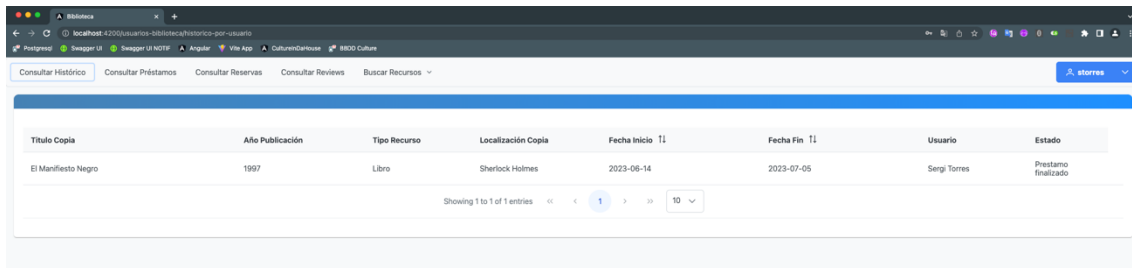
59 - Listado de sanciones del usuario

Rol User (Usuario)

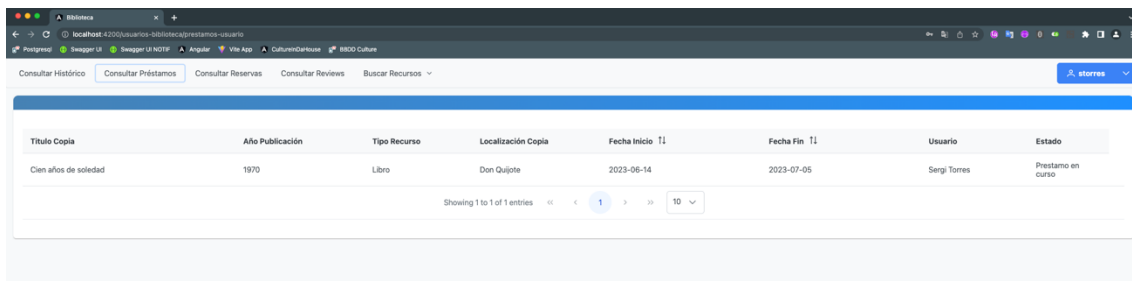


60 - Pantalla Principal de un Usuario

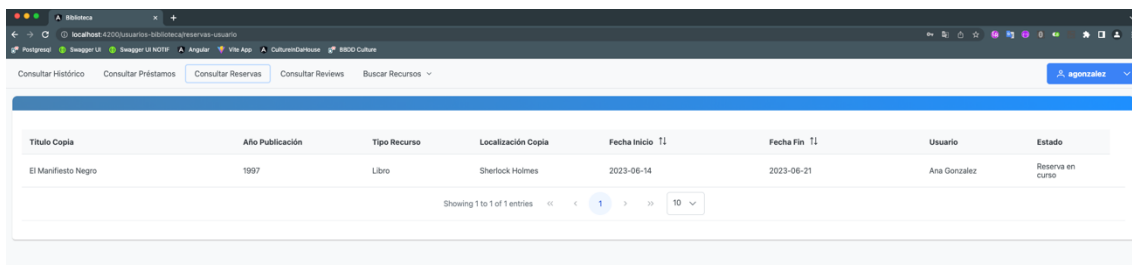
Hemos implementado las consultas de históricos y préstamos vigentes, así como las reservas que el usuario tenga vigentes.



61 - Histórico de Préstamos de un Usuario

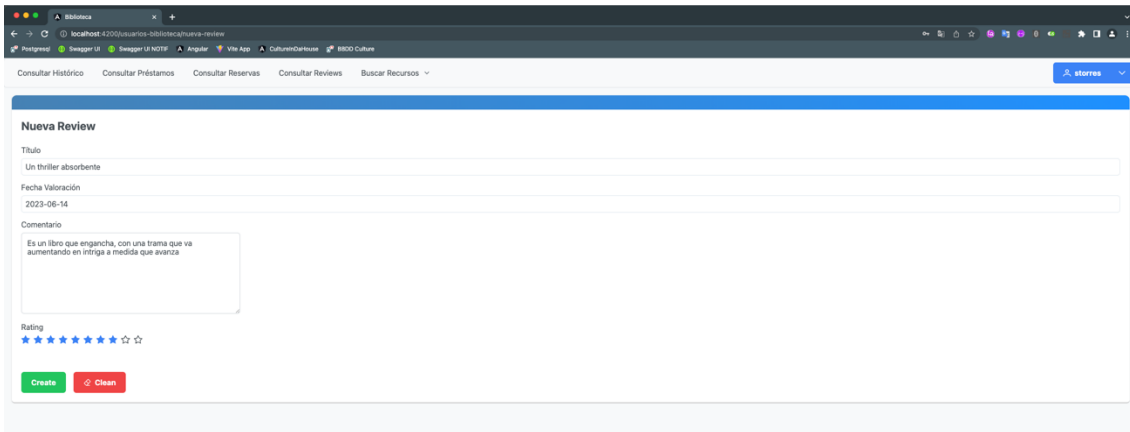


62 - Préstamos en Curso de un Usuario

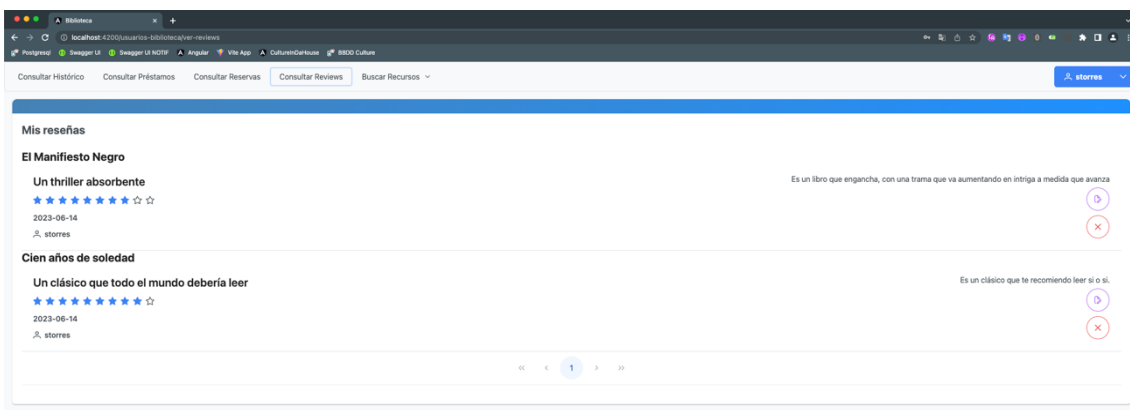


63 - Consulta de Reservas Vigentes de un Usuario

También hemos implementado la consulta de reviews, donde el usuario puede visualizar todas las reseñas que hay realizado.

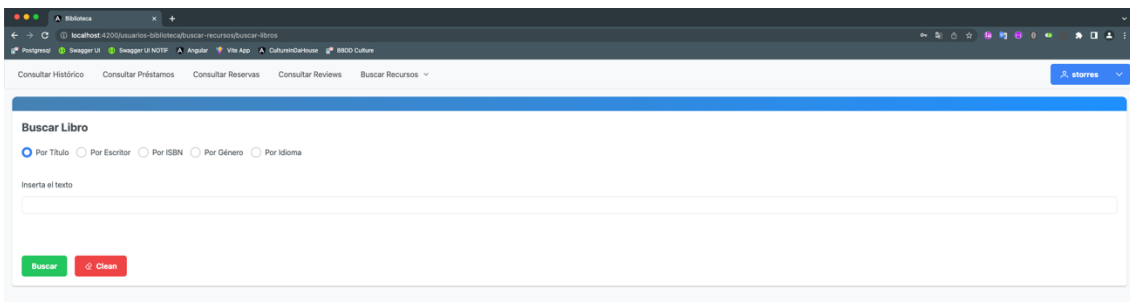


64 - Realizar Reseña de un Libro

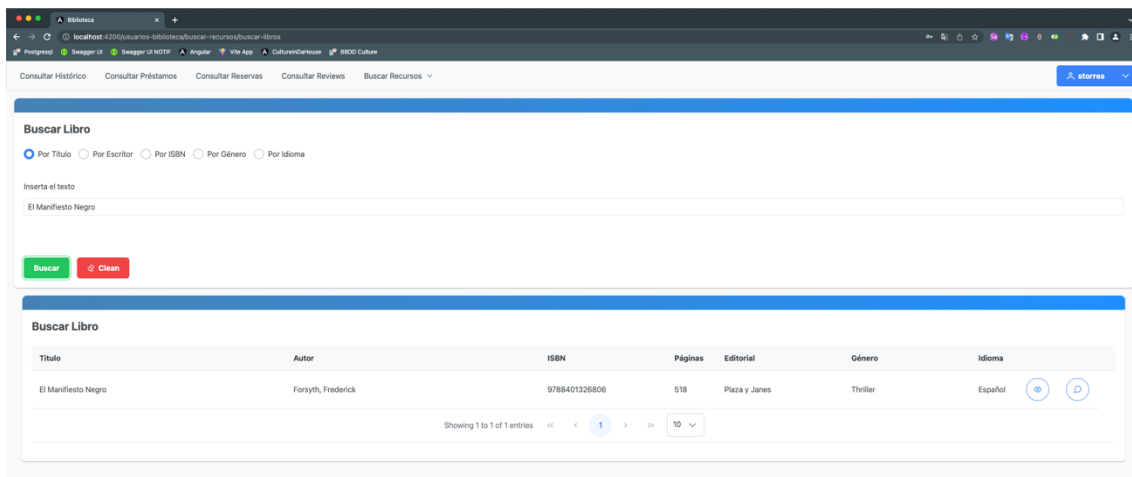


65 - Mirar Histórico de Reseñas de un Usuario

Por otro lado, se ha implementado las búsquedas de recursos por diferentes criterios, tal como se hizo para el bibliotecario.



66 - Búsqueda de Libros por Diferentes Criterios



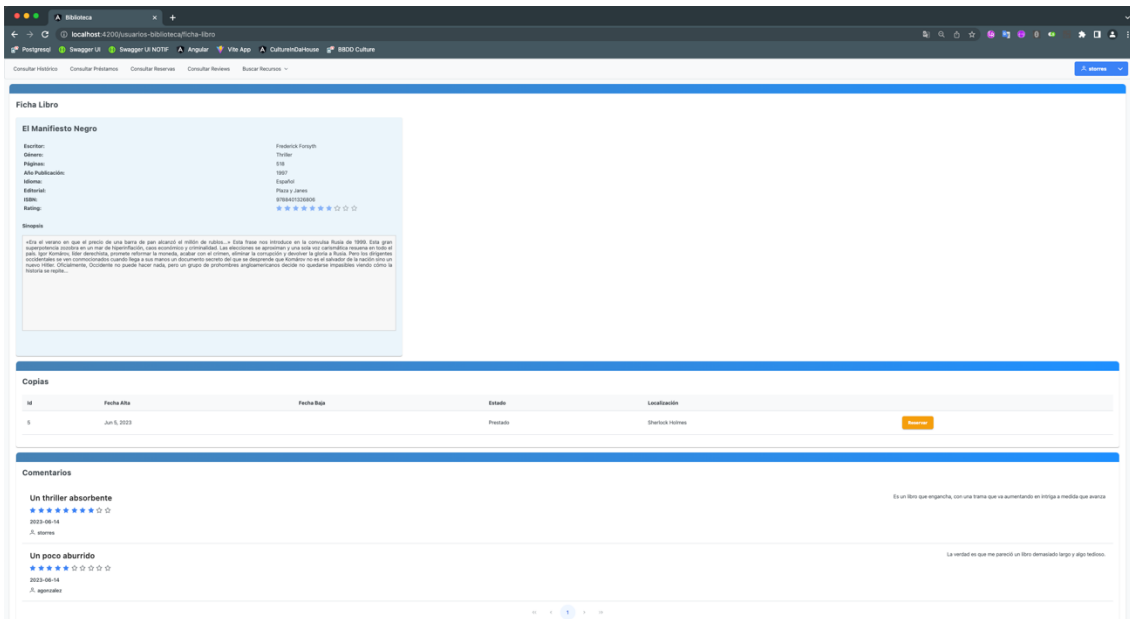
67 - Búsqueda de un Libro: Posibilidad de Mirar Ficha o Realizar Comentario

Aunque aquí con el resultado de las búsquedas, el usuario puede visualizar el recurso, donde se muestra una pantalla con los datos del recurso, la lista de sus copias, donde el usuario puede hacer la reserva o cancelarla, puesto que no permitimos que el usuario haga préstamos por si mismo, solo pueden ser realizados por el bibliotecario.

Además, también se puede ver la lista de reseñas del recurso. Por otra banda, se ha implementado la opción de realizar un comentario, donde el usuario puede realizar el comentario y poner la puntuación con estrellas del 1 al 10. Se ha implementado, de tal manera, que no permite al usuario realizar más de una reseña para cada recurso.

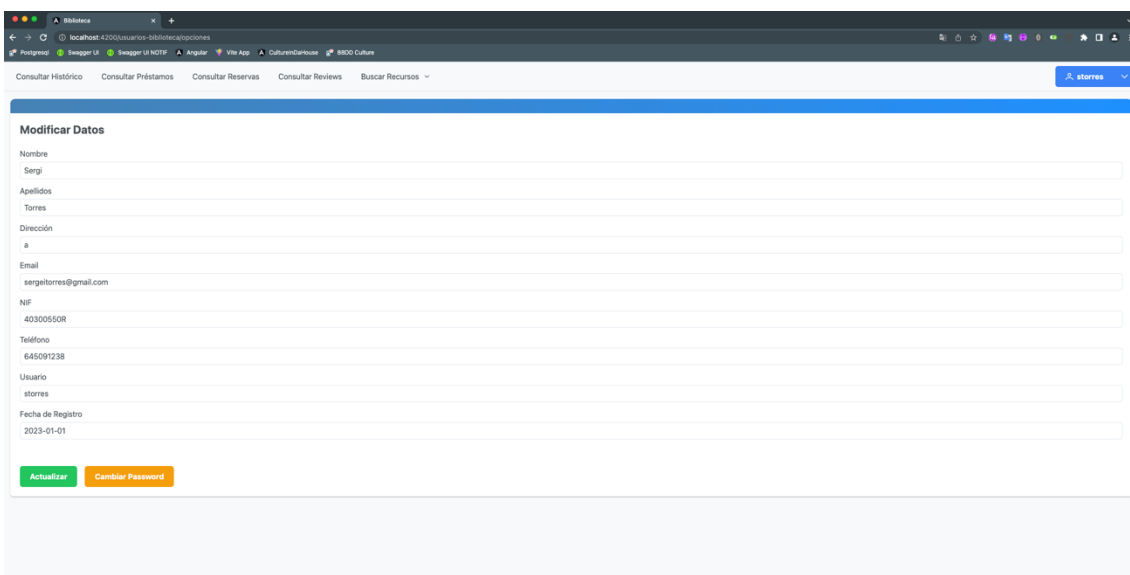
También se ha implementado la edición y la eliminación de la reseña, a estas opciones se accede desde el listado de reseñas obtenido tras hacer el usuario la consulta de las reseñas que ha realizado.

No hemos implementado, el cálculo de la media de puntuaciones de las reseñas de cada recurso es una característica que quedaría pendiente de implementar.



68 - Mirar Ficha de un Libro

Por último, al lado del desplegable para cerrar sesión, permitimos al usuario modificar sus datos, concretamente, aunque muestra todos sus datos, solo permitimos que modifique la dirección, el email, su teléfono y le damos posibilidad de cambiar su contraseña.



69 - Pantalla Modificación de Datos de un Usuario

Nos quedaría implementar en este punto la posibilidad de que el usuario pueda subir los comentarios o notificar sus préstamos en redes sociales. También quedaría por implementar, la opción de registrarse a las notificaciones por correo electrónico, de las nuevas altas de recursos de su interés.

3.4 Características pendientes de implementar y posibles mejoras

En este punto hablaremos de las funcionalidades que han quedado pendientes por implementar y además de algunas propuestas para mejorar la aplicación

Pendiente de implementar

- Posibilidad de que el usuario pueda elegir sus preferencias e intereses para recibir notificaciones de novedades relacionadas con dichos intereses.
- Compartir comentarios o préstamos realizados por redes sociales.
- Implementar la capa de seguridad a nivel backend con Spring Security, puesto que solo se ha implementado una capa básica de seguridad a nivel frontend mediante el uso de los Guards de Angular.
- Mejorar el sistema de notificaciones de errores para que sean más precisos.
- Documentar completamente a todo detalle la API en Swagger.
- Realización de batería de pruebas automatizadas.

Posibles mejoras de la aplicación

- Pulir el sistema de préstamos y reservas. Seguramente a causa de mis limitados conocimientos de frontend tengo la sensación de que el sistema resulta algo tosco.
- Implementar una cola de reservas, de momento solo un usuario puede reservar un recurso, pero sería conveniente que se pudiese reservar un recurso en cualquier momento, aunque ya existan reservas.
- Que un usuario pueda prorrogar un préstamo de forma fácil y eficiente. Relacionado con el primer punto, queda pulir y hacer más intuitivo el manejo de préstamos. Seguramente en la lista de préstamos en curso del usuario incluir la opción donde se debería tener en cuenta que no existan reservas del recurso, y además que falten un máximo de 2 o 3 días para el vencimiento.
- Incluir más información en las fichas de los recursos como biografías de los autores, más información sobre los recursos y el contexto en el que fueron concebidos, visualización de las carátulas o portadas de cada elemento, o incluir elementos multimedia como los tráiler de una película, documentales relacionados, o incluso muestras de audio de cada álbum de música.
- Sistema de puntos para las sanciones de los usuarios, vincular las sanciones a un préstamo determinado en caso de ser necesario o incluir un sistema de multas a los usuarios.

- Implementar un sistema de valoración de los recursos a nivel físico, para controlar su estado y probable deterioro, así como posibles daños recibidos. Esto podría consistir en un histórico vinculado con los préstamos del recurso, comentando el estado entre préstamos y si es necesario acompañarlo de fotos del recurso.
- Incluir otros recursos como los videojuegos o cómics.
- Posibilidad de recuperar el password por parte del usuario mediante reseteo del password y envío de nuevo password por correo electrónico.
- Mejora general de detalles y usabilidad de la interfaz gráfica.

4. Conclusiones

Es complicado plasmar en palabras todas las sensaciones y pensamientos que he tenido durante todo el proyecto: Alegría, frustración, angustia, tranquilidad... La principal conclusión que he sacado del proyecto es que aún estoy verde para afrontar con garantías un proyecto de esta envergadura y que tengo mucho por aprender.

Tanto en la tanto en la planificación, debo tener una perspectiva más amplia de todos los puntos de diseño y requisitos que se deben plasmar desde un principio, han ido surgiendo ideas o cosas a cambiar constantemente por no haberlo pensado desde un principio.

Asimismo, incapacidad para concebir propuestas novedosas Soy consciente que tengo poca imaginación para tener ideas nuevas y originales para realizar un proyecto, lo sé, un proyecto sobre una aplicación de gestión bibliotecaria no es nada original ni novedoso. He visto ideas realmente originales por parte de otros compañeros en los TFG del área Java que he podido ojear.

Por otro lado, en la implementación y parte técnica, aún estoy verde, y aunque durante el grado he adquirido una base sólida, aún debo aprender muchas cosas, he tenido contratiempos constantemente con problemas técnicos a la hora de manejar los datos, con el ORM, con el manejo de DTO. Todo esto, aunque se consulten libros, cursos, artículos en la web, no hay otra forma de mejorarlo más que con práctica.

Y complementando lo anterior, en la parte de diseño de interfaces, y manejo de la tecnología web, con HTML, CSS y Javascript, así como con el framework Angular, he gastado innumerables horas, ya que nunca había visto nada referente a diseño web. Si bien he aprendido mucho, el diseño de la interfaz gráfica, así como en la parte de usabilidad y experiencia de usuario hay mucho que mejorar.

Otra conclusión que he sacado es lo útil que son los conocimientos adquiridos en el grado de ingeniería informática, especialmente a nivel de diseño y planificación, y comprobar lo importante y la utilidad que tiene por ejemplo crear los casos de uso, el diagrama de clases o el modelo relacional de la base de datos.

Todo esto nos lleva a concluir que, si bien creo haber conseguido cumplir con el objetivo primordial de implementar una aplicación funcional, no obstante, aún se trata de una beta lejos de ser una versión lista para producción que tiene margen de mejora, además de algunas funcionalidades previstas inicialmente que no se han implementado aún. Pienso los motivos han sido por un lado mi falta de pericia técnica, creo que me falta más nivel para ser más rápido y eficaz en la implementación de código, sobretodo a nivel frontend.

Por otro lado, la falta de tiempo, seguramente con más tiempo para la implementación, podría haber mejorado en todos los aspectos y haber incluido

todas las funcionalidades previstas inicialmente, así como tener solventados posibles bugs que pudiesen existir.

En conclusión, si bien no estoy contrariado con el proyecto realizado, y creo que hemos conseguido cumplir con la mayoría de los objetivos propuestos inicialmente, pienso que hay margen de mejora. Asimismo, sería conveniente para mi progreso plantearme más proyectos de estas características para mejorar mis habilidades. No obstante, estoy contento por lo que me ha aportado a nivel de experiencia y de poner a prueba los conocimientos adquiridos durante el grado para la realización de este proyecto.

5. Glosario

Caso de uso: Técnica utilizada en la ingeniería de requisitos para reflejar la interacción entre el sistema y sus diversos actores en diferentes situaciones o escenarios principales en la aplicación

Diagrama de clases: Es un tipo de diagrama utilizado en la Ingeniería del Software y concretamente bajo el lenguaje de modelado UML en la cual se muestra la organización estática de una aplicación indicando las diferentes clases (atributos y métodos) así como sus relaciones dentro del sistema.

Prototipo de pantallas: Es una técnica en la cual se muestra diferentes pantallas en forma de boceto o borrador correspondientes a la interfaz gráfica de una aplicación con la finalidad de visualizar la posible estructura de las diferentes opciones y elementos.

Java: Lenguaje de programación desarrollado por la empresa Sun Microsystems y publicado por primera vez en 1995. Sus principales características son el fuerte tipado, cuyo paradigma está orientada a objetos y que es multiplataforma.

JavaScript: Lenguaje de programación diseñado originalmente por Netscape, el cual es multiparadigma y con tipado débil y dinámico. Su principal finalidad es permitir el desarrollo de software en la tecnología web. Es muy utilizado para dotar de interactividad y dinamismo a las páginas web.

HTML: Son las siglas de *HyperText Markup Language* y es un lenguaje de marcado y etiquetas utilizado para la organizar y mostrar el contenido de una página web.

CSS: Son las siglas de *Cascading Style Sheets* y es un lenguaje de marcado utilizado para definir el diseño y estilo de una página web. Sirve como complemento al lenguaje HTML.

ORM: Son las siglas de *Object Relational Mappings*, y es un marco de programación que sirve para proporcionar una capa de abstracción y facilitar el mapeo, comunicación y persistencia entre los diferentes objetos y clases de un lenguaje orientado a objetos y las tablas de una base de datos relacional.

JPA: Son las siglas de *Java Persistence API*, y es una API de persistencia desarrollada para utilizarla en aplicaciones Java EE. Concretamente se utiliza como ORM, y podríamos decir que es el ORM estándar para Java.

SGDB: Son las siglas de Sistema Gestor de Base de Datos y se trata de un software para la administración y gestión de bases de datos. Estos sistemas proporcionan diversas herramientas para la creación de esquemas de datos, manipulación y consulta de datos, mantener la integridad de los datos,

proporcionar seguridad al acceso de los datos o el control del acceso concurrente a los datos.

SPA: Son las siglas de *Single Page Application* es una aplicación web que se realiza o presenta en una sola página para simular la apariencia y comportamiento de una aplicación de escritorio con la finalidad de ofrecer una experiencia de usuario ágil, fluida y satisfactoria.

Diagrama de Gantt: Es una técnica para planificar proyectos que muestra de forma gráfica las diversas tareas, sus plazos, los hitos y recursos utilizados en un proyecto.

Frontend: Hace referencia al desarrollo de parte del cliente, centrándose en la realización de la interfaz gráfica de la aplicación web que interactuará con el usuario.

Backend: Hace referencia al desarrollo en la parte del servidor, centrándose en la implementación de toda la lógica de negocio de la aplicación web o en la gestión de las bases de datos y de manejar las peticiones provenientes del cliente.

CRUD: Son las siglas de *Create, Read, Update y Delete* y reflejan las diversas operaciones básicas que se pueden realizar sobre los datos almacenados en una base de datos.

API: Son las siglas de *Application Programming Interface* y se puede decir que son un conjunto de normas y procedimientos que posibilitan la relación y comunicación entre diversos componentes de software. Ofrece una interfaz detallada y documentada de los diversos métodos y datos que ofrece para ser utilizado por otras aplicaciones.

API RESTful: Es una interfaz que sigue los principios del estilo arquitectónico REST (*Representational State Transfer*) y permite la comunicación entre cliente y servidor mediante métodos HTTP como GET, POST, PUT o DELETE, los cuales permiten la operaciones CRUD de los datos contenidos en el servidor.

Framework: Es un conjunto de librerías predeterminadas que deben personalizarse para usarlas como base para el desarrollo de aplicaciones lo cual permite ahorrar mucho tiempo. De hecho, ofrece una serie de características genéricas implementadas y validadas que posibilitan al programador centrarse en los aspectos propios de la aplicación que tiene que implementar.

6. Bibliografía

Casas, J. Diseño Conceptual de Base de Datos. UOC

Burgués, X. Diseño Lógico de base de datos. UOC

Moreno,N.;Vallecillo,A.; Romero, J.R.;Durán, F.J. Arquitectura del programari.UOC

Moreno,N.;Vallecillo,A.; Romero, J.R.;Durán, F.J. Desenvolupament de programari basat en components.UOC

Pradel, J.; Raya, J.Introducción a los patrones. UOC

Pradel, J.; Raya, J. Requisitos. UOC

Pradel, J.; Raya, J. Análisis UML. UOC

Richards, M.;Ford, N. (2020) Fundamentals of Software Architecture. O'Reilly
TFG's del Repositorio de la UOC.

Herrera, F. – Angular: De cero a experto – Edición 2023

<https://www.udemy.com/course/angular-fernando-herrera/>

Guzmán, A.J – Spring Framework & Spring Boot desde cero a experto

<https://www.udemy.com/course/spring-framework-5/>

jonmircha – Curso Javascript

<https://www.youtube.com/watch?v=2SetvwBV-SU&list=PLvq-ilkSeTUZ6QgYYO3MwG9EMqC-KoLXA>

Haverbeke, M. – Eloquent Javascript: A Modern Introduction to Programming – 3ª Edición.

Jansen, T. Inheritance Strategies with JPA and Hibernate [Consulta: Abril 2023]

<https://thorben-janssen.com/complete-guide-inheritance-strategies-jpa-hibernate/>

Almeida, N. Cascade en JPA/Hibernate [Consulta: Abril 2023]

<https://www.nestoralmeida.com/cascade-en-jpa-hibernate/>

Romero, J. Hibernate – OneToOne, OneToMany, ManyToOne y ManyToMany [Consulta: Abril 2023]

<https://www.adictosaltrabajo.com/2020/04/02/hibernate-onetoone-onetomany-manytoone-y-manymany/>

Baeldung – Documenting a Spring REST API. Using OpenAPI 3.0 [Consulta: Abril 2023]

<https://www.baeldung.com/spring-rest-openapi-documentation>

Open API3 – springdoc-openapi v.1.7.0 [Consulta: Abril 2023]

<https://springdoc.org/#getting-started>

Baeldung - Overview of JPA/Hibernate Cascade Types [Consulta: Abril 2023]

<https://www.baeldung.com/jpa-cascade-types>

Bezkoder – Spring Boot + Swagger example [Consulta: Abril 2023]

<https://www.bezkoder.com/spring-boot-swagger-3/>

Rodriguez Calle, N. Tipos de Cascade en Hibernate – JPA [Consulta: Abril 2023]

<https://refactorizando.com/tipos-cascade-hibernate-jpa/>

Peiretti, G. How to send emails with Spring Boot [Consulta: Junio 2023]

<https://gustavopeiretti.com/spring-boot-send-emails/>

Baeldung – Guide to Spring Email [Consulta: Junio 2023]

<https://www.baeldung.com/spring-email>

Baeldung – The Scheduled Annotation in Spring [Consulta: Abril 2023]

<https://www.baeldung.com/spring-scheduled-tasks>

Baeldung – Spring Scheduling Annotations [Consulta: Abril 2023]

<https://reflectoring.io/spring-scheduler/>

IBM – Expresiones CRON para definir frecuencias [Consulta: Junio 2023]

<https://www.ibm.com/docs/es/urbancode-release/6.1.0?topic=interval-cron-expressions-defining-frequency>

7. Anexos

7.1 Manual de procedimiento para instalación del proyecto

En esta fase del proyecto, como aún no tenemos ningún proveedor de servicios para tener alojados nuestra base de datos, el ejecutable de Spring con el servidor de Apache embebido o el frontend de Angular.

De momento, tenemos que conformarnos con la instalación y ejecución del proyecto en local.

En primer lugar, debemos ir a los repositorios del código de frontend y backend:

Repositorio Frontend:

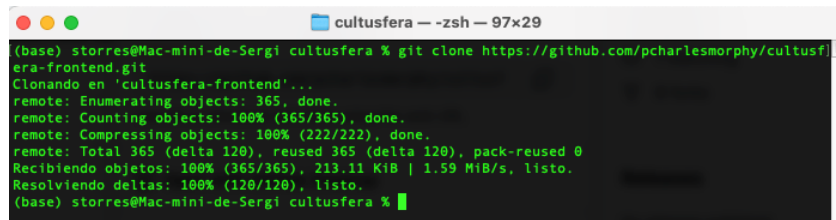
<https://github.com/pcharlesmorphy/cultusfera-frontend>

Repositorio Backend:

<https://github.com/pcharlesmorphy/cultusfera-backend>

Empezaremos con el Frontend

Abrimos el terminal



```
cultusfera -- zsh -- 97x29
(base) storres@Mac-mini-de-Sergi cultusfera % git clone https://github.com/pcharlesmorphy/cultusfera-frontend.git
Clonando en 'cultusfera-frontend'...
remote: Enumerating objects: 365, done.
remote: Counting objects: 100% (365/365), done.
remote: Compressing objects: 100% (222/222), done.
remote: Total 365 (delta 120), reused 365 (delta 120), pack-reused 0
Recibiendo objetos: 100% (365/365), 213.11 KiB | 1.59 MiB/s, listo.
Resolviendo deltas: 100% (120/120), listo.
(base) storres@Mac-mini-de-Sergi cultusfera %
```

Se habrá creado la carpeta cultusfera-frontend.

Como se trata de un proyecto de frontend implementado en Angular, esto implica tener que instalar Node.js

<https://nodejs.org/es>

Tras instalar Node.js, debemos instalar el Angular CLI

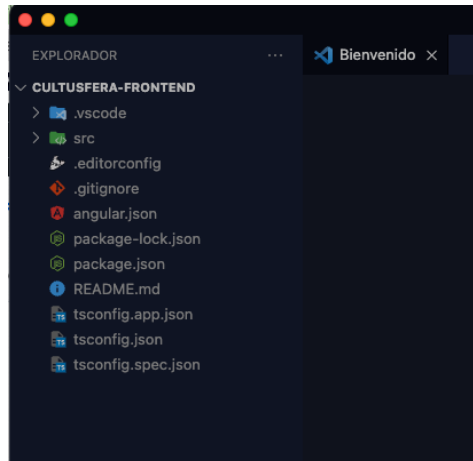
<https://angular.io/cli>

En el terminal debemos escribir

```
npm install -g @angular/cli
```

Tras instalar Node.js y Angular CLI, ya estamos en disposición de utilizar Angular en nuestro equipo.

El siguiente paso sería abrir la carpeta del proyecto con un editor, nosotros recomendamos Visual Studio Code, pero esto no es obligatorio. Abrimos la carpeta y nos saldrá el proyecto:



Se puede comprobar que nos faltan los módulos de Node en el proyecto, para poder ejecutarlo, así que en la carpeta del proyecto desde terminal ejecutamos el siguiente comando: npm install

```
(base) storres@Mac-mini-de-Sergi cultusfera-frontend % npm install
npm WARN deprecated @npmcli/move-file@2.0.1: This functionality has been moved to @npmcli/fs
added 982 packages, and audited 983 packages in 12s

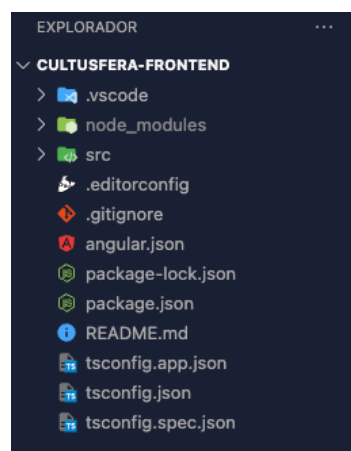
98 packages are looking for funding
  run `npm fund` for details

2 vulnerabilities (1 moderate, 1 high)

To address all issues, run:
  npm audit fix

Run `npm audit` for details.
(base) storres@Mac-mini-de-Sergi cultusfera-frontend %
```

Ahora observamos que ya tenemos los módulos de node instalados.



Es importante, también configurar la url del backend, para esto tenemos un archivo en la carpeta src/app/environments llamado environment.ts, por defecto esta la url localhost:8080 que es la que utiliza por defecto Spring. Si se cambia esta url, hay que tener en cuenta que se debe cambiar en el proyecto de Angular.

Una vez instalados los modulos de node en el proyecto ya se puede iniciar el proyecto con ng serve -o en el terminal , dentro de la carpeta del proyecto.

```
(base) storres@Mac-mini-de-Sergi cultusfera-frontend % ng serve -o
✓ Browser application bundle generation complete.

Initial Chunk Files                                         Names                               Raw Size
vendor.js                                                  vendor                             5.59 MB
main.js                                                    main                               1.56 MB
styles.css, styles.js                                     styles                             1.23 MB
polyfills.js                                              polyfills                          314.27 kB
runtime.js                                                runtime                             12.62 kB

Initial Total                                              8.70 MB

Lazy Chunk Files                                           Names                               Raw Size
src_app_user-library-client_user-library-client_module_ts.js  user-library-client-user-library-client-module  325.65 kB
src_app_transaction-management_transaction-management_module_ts.js  transaction-management-transaction-management-module  189.35 kB
src_app_user-management_user-management_module_ts.js            user-management-user-management-module          188.53 kB
default-src_app_historic-search_historic-search_module_ts.js    historic-search-historic-search-module          95.93 kB
src_app_copy-management_copy-management_module_ts.js            copy-management-copy-management-module          69.57 kB
default-src_app_auth_auth_module_ts.js                          auth-auth-module                               16.14 kB
common.js                                                      common                                           5.45 kB

Build at: 2023-06-05T18:13:17.018Z - Hash: 676d5d74ffe5c63c - Time: 16352ms

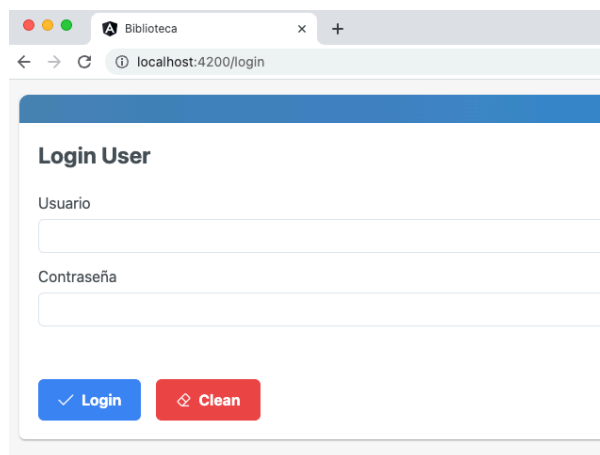
./src/styles.css - Warning: Module Warning (from ./node_modules/css-loader/dist/cjs.js):
Warning

(6:1) It looks like you didn't end your @import statement correctly. Child nodes are attached to it.

** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **

✓ Compiled successfully.
```

Tras esto se nos abrirá ya la pantalla de login de la aplicación en el navegador con la dirección: localhost:4200/login



Ahora pasamos a la parte de backend

```
(base) storres@Mac-mini-de-Sergi cultusfera % git clone https://github.com/pcharlesmorphy/cultusfera-backend.git
Clonando en 'cultusfera-backend'...
remote: Enumerating objects: 265, done.
remote: Counting objects: 100% (265/265), done.
remote: Compressing objects: 100% (158/158), done.
remote: Total 265 (delta 96), reused 265 (delta 96), pack-reused 0
Recibiendo objetos: 100% (265/265), 137.40 KiB | 1.72 MiB/s, listo.
Resolviendo deltas: 100% (96/96), listo.
(base) storres@Mac-mini-de-Sergi cultusfera %
```

En primer lugar, hay que instalar el JDK de Java

<https://www.oracle.com/es/java/technologies/downloads/>

Además, es conveniente instalar un IDE, nosotros recomendamos IntelliJ IDEA Ultimate, aunque existe la opción gratuita Spring Tool Suite 4.

Hay que instalar también el gestor de dependencias Maven, así como configurar las variables de entorno.

<https://maven.apache.org/download.cgi>

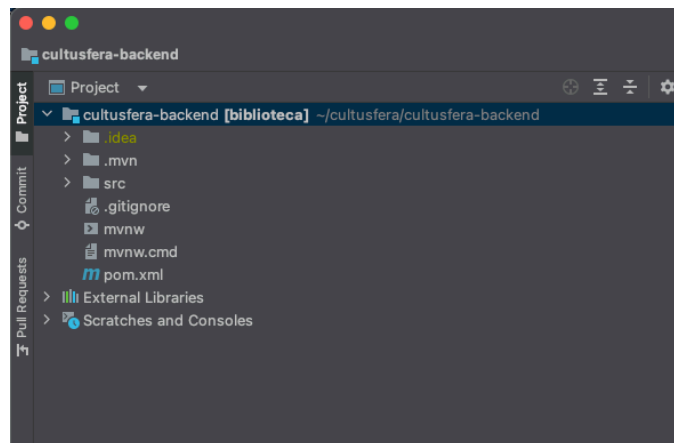
Por otro lado, debemos instalar el SGDB, en este caso MySQL, en concreto MySQL Community Server

<https://dev.mysql.com/downloads/mysql/>

Por otro lado, recomendamos un interface gráfico para trabajar con el servidor de base de datos, creemos que la mejor opción es MySQL Workbench

<https://dev.mysql.com/downloads/workbench/>

Una vez, tenemos el entorno preparado, abrimos la carpeta del proyecto con el IDE, y esto sería lo que obtenemos:



Importante ir al archivo de configuración application.properties

```
spring.datasource.url=jdbc:mysql://localhost:3306/biblioteca
spring.datasource.username=root
spring.datasource.password=
spring.main.allow-circular-references=true
```

Y configurar tanto la url, como el username y el password que se tenga en cada caso particular de la instalación de MySQL.

Lo ideal es crear un nuevo Schema, con el nombre que se desee para incluirlo en la url, el puerto 3306 es el que normalmente se dispone por defecto por parte de MySQL.

También es importante tener configurado el puerto en el que se ejecutará el proyecto backend de Spring, por defecto, tenemos localhost:8080.

Una vez tenemos esto configurado, nos vamos al terminal , a la carpeta del proyecto backend para crear el ejecutable del proyecto. Esto es opcional, en caso de no querer abrir siempre el IDE para ejecutar el backend.

En el terminal ejecutamos:

```
(base) storres@Mac-mini-de-Sergi cultusfera-backend % mvn clean package
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.example.biblioteca >-----
[INFO] Building biblioteca 0.0.1-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- maven-clean-plugin:3.2.0:clean (default-clean) @ biblioteca ---
[INFO] Deleting /Users/storres/cultusfera/cultusfera-backend/target
[INFO]
[INFO] --- maven-resources-plugin:3.3.1:resources (default-resources) @ biblioteca ---
[INFO] Copying 2 resources from src/main/resources to target/classes
[INFO] Copying 1 resource from src/main/resources to target/classes
[INFO]
[INFO] --- maven-compiler-plugin:3.10.1:compile (default-compile) @ biblioteca ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 226 source files to /Users/storres/cultusfera/cultusfera-backend/target/classes
[INFO] /Users/storres/cultusfera/cultusfera-backend/src/main/java/com/example/biblioteca/controlles unchecked or unsafe operations.
[INFO] /Users/storres/cultusfera/cultusfera-backend/src/main/java/com/example/biblioteca/controlles
[INFO]
[INFO] --- maven-resources-plugin:3.3.1:testResources (default-testResources) @ biblioteca ---
[INFO] skip non existing resourceDirectory /Users/storres/cultusfera/cultusfera-backend/src/test/
[INFO]
[INFO] --- maven-compiler-plugin:3.10.1:testCompile (default-testCompile) @ biblioteca ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to /Users/storres/cultusfera/cultusfera-backend/target/test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.22.2:test (default-test) @ biblioteca ---
[INFO]
[INFO] -----
[INFO] T E S T S
[INFO] -----
[INFO] Running com.example.biblioteca.BibliotecaApplicationTests
11:16:37.730 [main] DEBUG org.springframework.boot.test.context.SpringBootTestContextBootstrapper
```

Una vez obtenido el ejecutable tras un “BUILD SUCCESS”, se obtiene el jar


```

Hibernate: alter table transaction add constraint FK2bf21b32im48dxm8kd3usmuy8 foreign key (id_user) referenc
Hibernate: alter table user add constraint FK2d97cq5t6lj83mdqxfpccdec5 foreign key (id_user_role) referenc
2023-06-05T21:34:04.604+02:00 INFO 42597 --- [main] o.h.e.t.j.p.i.JtaPlatformInitiator : H
2023-06-05T21:34:04.609+02:00 INFO 42597 --- [main] j.LocalContainerEntityManagerFactoryBean : I
2023-06-05T21:34:05.644+02:00 WARN 42597 --- [main] JpaBaseConfiguration$JpaWebConfiguration : S
.jpa.open-in-view to disable this warning
2023-06-05T21:34:06.084+02:00 INFO 42597 --- [main] c.e.b.BibliotecaApplicationTests : S
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 5.289 s - in com.example.biblioteca.B
2023-06-05T21:34:06.211+02:00 INFO 42597 --- [ionShutdownHook] j.LocalContainerEntityManagerFactoryBean : C
2023-06-05T21:34:06.212+02:00 INFO 42597 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource : H
2023-06-05T21:34:06.217+02:00 INFO 42597 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource : H
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] --- maven-jar-plugin:3.3.0:jar (default-jar) @ biblioteca ---
[INFO] Building jar: /Users/storres/cultusfera/cultusfera-backend/target/biblioteca-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- spring-boot-maven-plugin:3.0.6:repackage (repackage) @ biblioteca ---
[INFO] Replacing main artifact with repackaged archive
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 11.195 s
[INFO] Finished at: 2023-06-05T21:34:07+02:00
[INFO] -----

```

Vamos a la ruta indicada donde se generó el archivo jar:

Y en el terminal ponemos `java -jar biblioteca-0.0.1-SNAPSHOT.jar`

Y ya tenemos la aplicación ejecutándose en localhost: 8080 con el servidor Apache Tomcat embebido.

Ahora tan solo quedaría ir al frontend que tenemos en ejecución

Existen tres Roles en la aplicación:

Admin, Librarian y User

Cada una de ellas tiene una serie de funcionalidades disponibles:

Admin: Puede crear nuevos usuarios, así como editar los existentes.

Librarian: Es el perfil de bibliotecario, puede crear, editar, eliminar recursos, gestionar las copias de los recursos, gestionar préstamos entre otras cosas.

User: Este perfil es el de un usuario normal.

Los usuarios y password para poder entrar con cada usuario son:

Admin

User: admin

Password: 12345

Librarian

User: librarian

Password: 12345

De usuarios tenemos 4.

User:jgarcia

User:pramirez

User:storres

User:agonzalez

Todos con el mismo password 12345

Entramos con el usuario librarian

Y nos vamos a Gestión Recursos-Libros-Listar Libros

Si toda la instalación ha sido correcta nos debería salir

