

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
INFORMÁTICA
GRADO EN INGENIERÍA DEL SOFTWARE

APLICACIÓN WEB PARA LAS BIBLIOTECAS ESCOLARES
WEB APPLICATION FOR SCHOOL LIBRARIES

Realizado por
Sihame El Ghazouani
Tutorizado por
Eduardo Guzmán De los Riscos
Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, Febrero de 2017

Fecha defensa:
El Secretario del Tribunal

Resumen:

La lectura en la etapa escolar es supremamente importante, y la biblioteca cumple una especial labor en la enseñanza y el aprendizaje, pues ésta ejerce una influencia sobre el logro académico en los estudiantes. Las bibliotecas sirven para propiciar, favorecer y estimular la lectura. Por todo lo anterior, hemos decidido implementar nuestra aplicación web para las bibliotecas escolares cuyos usuarios serán los bibliotecarios y los encargados de todo lo relativo a la gestión del catálogo de la biblioteca.

Para llevar a cabo el desarrollo de este software hemos seguido un desarrollo iterativo, una iteración inicial para un estudio global del problema y las tecnologías a utilizar, seguida de siete iteraciones donde al inicio de cada una, seleccionamos las funcionalidades que más valor aportan a nuestro sistema, de manera que al final de cada iteración obtenemos una versión funcional y demostrable.

Para la implementación, hemos optado por utilizar Spring Framework por su modularidad, flexibilidad y facilidad a la hora de implementar aplicaciones empresariales. Otro aspecto técnico es la utilización de MySQL como gestor de bases de datos.

Palabras clave: biblioteca, catálogo, fondos, Spring Framework, software, aplicación web, bibliotecario

Abstract:

Reading at school is supremely important, and the library plays a special role in teaching and learning, as it has an influence on academic achievement in students. Libraries serve to promote, encourage and stimulate reading. For all of those reasons, we have decided to implement our web application for school libraries whose users will be the librarians and those in charge of everything related to the management of the library catalog.

To carry out the development of this software, we have followed an iterative and incremental approach. The process was divided in several iterations: an initial one where we perform a global study to get an overview about the system, followed by seven more iterations, where we have evolved the product until the final result has been achieved. At the beginning of each iteration, we select the requirements most valued, so at the end of each one, we get a functional and deliverable version.

The implementation has been done using the Spring Framework due to its modularity, flexibility and the facility that it provides in developing enterprise applications. Another technical aspect is the use of MySQL as a database manager.

Keywords: library, catalog, funds, Spring Framework, software, web application, librarian

Contenido

Capítulo 1. Introducción	5
1.1. Objetivos.....	5
1.2. La aplicación de escritorio	5
1.3. La aplicación web.....	6
1.4. Tecnologías y herramientas.....	6
1.4.1. Spring Core	6
1.4.2. Spring Boot	7
1.4.3. Spring MVC	7
1.4.4. Spring Security	7
1.4.5. Spring Data JPA	7
1.4.6. JSP	7
1.4.7. JSTL	8
1.4.8. Maven.....	8
1.4.9. MySQL.....	8
1.4.10. Git.....	8
1.4.11. Eclipse Mars.....	8
Capítulo 2. Funcionalidad requerida.....	9
2.1. Actores	9
2.2. Requisitos funcionales	9
2.2.1. RF01 – Gestión de usuarios.....	9
2.2.2. RF02 – Gestión del depósito	10
2.2.3. RF03 – Gestión del catálogo.....	10
2.2.4. RF04 – Gestión de lectores.....	11
2.2.5. RF05 – Gestión de políticas de préstamo	12

2.2.6. RF06 – Gestión de préstamos.....	13
2.3. Requisitos no funcionales.....	13
Capítulo 3. Diseño de la aplicación.....	15
3.1. Actores	15
3.2. Diagramas de Casos de uso	16
3.2.1. Gestión de usuarios	16
3.2.2. Gestión del depósito.....	16
3.2.3. Gestión del catálogo	17
3.2.4. Gestión de lectores	17
3.2.5. Gestión de políticas de préstamo.....	18
3.2.6. Gestión de préstamos	18
3.3. Modelo Conceptual.....	19
3.4. Modelo Relacional	21
3.5. Diagrama de paquetes	22
3.6. Diagrama de despliegue	22
Capítulo 4. Implementación	23
4.1. Metodología de desarrollo	23
4.2. Iteración cero	24
4.2.1. Estudio inicial	24
4.2.2. Búsqueda de información	24
4.2.3. Instalar las herramientas.....	24
4.2.4. Creación de la base de datos.....	24
4.2.5. Creación de la estructura del proyecto	24
4.2.6. Creación del repositorio.....	27
4.2.7. Configuración del proyecto.....	27
4.3. Iteración I.....	28

4.3.1. Acceso a datos.....	29
4.3.2. Spring MVC	31
4.3.3. Paginación.....	32
4.4. Iteración II	34
4.4.1. Spring Security	34
4.5. Iteración III.....	36
4.6. Iteración IV	36
4.7. Iteración V	37
4.8. Iteración VI.....	37
4.8.1. Validación de datos	37
4.9. Iteración VII	40
Capítulo 5. Aspecto visual.....	43
5.1. Herramientas.....	43
5.2. Componentes principales	43
5.3. Aspectos tenidos en cuenta.....	45
Capítulo 6. Pruebas.....	47
Capítulo 7. Conclusiones y trabajo futuro.....	49
7.1. Tiempo empleado	49
7.2. Resultado obtenido	50
7.3. Metodología utilizada.....	50
7.4. Tecnologías utilizadas.....	50
7.5. Trabajo futuro	51
Anexo Técnico I.....	55
I.1. Casos de uso	55
I.1.1. CU01 – Gestión de usuarios	55

I.1.2. CU02 – Gestión del depósito.....	57
I.1.3. CU03 – Gestión del catálogo.....	59
I.1.4. CU04 – Gestión de lectores.....	62
I.1.5. CU05 – Gestión de políticas de préstamo.....	64
I.1.6. CU06 – Gestión de préstamos.....	68
I.2. Casos de prueba.....	70

Capítulo 1. Introducción

La presente memoria pertenece al Trabajo Fin de Grado titulado “Aplicación web para las bibliotecas escolares” que, como su nombre indica, es un sistema software para gestionar el catálogo de las bibliotecas escolares. El trabajo consiste en la migración de la aplicación de escritorio ABIES, desarrollada por el Ministerio de Educación de España, a una aplicación web.

A lo largo de esta memoria, se obtendrán los conocimientos para la total comprensión del desarrollo y la implementación de la aplicación.

1.1. Objetivos

Los principales objetivos de este trabajo son:

- Implementar una aplicación web que facilite la gestión de los fondos documentales de las bibliotecas escolares.
- Migración de una aplicación de escritorio a una aplicación web.
- Comprensión y aprendizaje de las metodologías y tecnologías utilizadas durante el diseño e implementación de la aplicación.

1.2. La aplicación de escritorio

ABIES es la aplicación informática destinada a la gestión de bibliotecas escolares desarrollada por el INTEF, unidad dependiente del Ministerio de Educación, Cultura y Deporte, en proyecto de colaboración con las Comunidades Autónomas.

Su principal cometido consiste en automatizar los procesos de la biblioteca escolar, favoreciendo el trabajo del profesorado implicado en la puesta en marcha o dinamización de la biblioteca debido a la simplificación de tareas rutinarias como la catalogación o el préstamo, y permitiendo, incluso, el trabajo colectivo de los miembros del grupo de biblioteca a través de una red de área

local. Asimismo, fomenta la autonomía del usuario facilitando el acceso y la localización de los fondos a través del catálogo de acceso público (OPAC).

1.3. La aplicación web

La aplicación web que queremos diseñar e implementar va a ser una primera versión resultado de migrar la aplicación de escritorio ABIES a una web. Por ello, hemos seleccionado las funcionalidades que creemos que son más importantes y prioritarias para incluirlas en esta versión.

Nuestro sistema permitirá a los bibliotecarios la organización de catálogos y de los recursos de las bibliotecas, realización préstamos y devoluciones de ejemplares, gestión de lectores, además de configurar las políticas de préstamos. También permitirá la consulta de los fondos documentales de un depósito e importación de los registros bibliográficos. El depósito es de una base de datos que contiene registros bibliográficos de cerca de 400.000 fondos. Se trata una fuente de recursos para la catalogación automatizada principalmente dirigida a las bibliotecas públicas.

Con este trabajo pretendemos ofrecer una serie de ventajas que tiene el uso de una aplicación web sobre una aplicación de escritorio:

- Disponibilidad de la aplicación a través de dispositivos que tengan un navegador. De esta manera mejoramos la accesibilidad al catálogo de la biblioteca y las gestiones de la biblioteca se realizarán sin tener que instalar ninguna aplicación.
- Con actualizar la aplicación en el servidor, todos los usuarios la tendrán actualizada en el momento.

Además, hoy en día, resulta más práctico y aconsejable el uso de aplicaciones web.

1.4. Tecnologías y herramientas

Como ya hemos mencionado, uno de los objetivos de este proyecto es familiarizarnos con el número máximo de tecnologías posible y obtener unos conocimientos prácticos de cada uno de ellas. Por ello vamos a detallar de manera introductoria las que vamos a utilizar.

1.4.1. Spring Core

Para la implementación de nuestra aplicación, hemos optado por utilizar Spring, que es un framework para el desarrollo de aplicaciones de código libre para la plataforma Java. Es un framework que se compone de múltiples módulos que abarcan una gran variedad de servicios que facilitan la implementación de una aplicación.

1.4.2. Spring Boot

Como hemos mencionado anteriormente, Spring tiene una amplia cantidad de módulos que implican multitud de configuraciones. Estas configuraciones pueden tomar mucho tiempo, pueden ser desconocidas para principiantes y suelen ser repetitivas. La solución de Spring es Spring Boot, que aplica el concepto de *Conversion over Configuration* (CoC).

1.4.3. Spring MVC

Spring MVC es uno de los módulos que nos ofrece Spring Framework basado en el patrón modelo-vista-controlador. Ofrece una división limpia entre el controlador, el modelo y la vista, debilitando el acoplamiento entre ellos. Además de ser muy flexible ya que implementa toda su estructura mediante interfaces.

1.4.4. Spring Security

La mayoría de las páginas web tienen sistema de usuarios, dando a cada usuario un perfil. Cada perfil tiene su nivel de acceso, no todos los usuarios pueden acceder a todas las secciones de la aplicación web. Además, cada usuario debe verificar su identidad. Spring Security ofrece medidas de autenticación y control de acceso para las aplicaciones Java.

1.4.5. Spring Data JPA

Spring Data es otro de los módulos de Spring que a su vez contiene una colección de submódulos asociados a tecnologías de acceso a datos. Sus módulos dan acceso a tecnologías nuevas como bases de datos no relacionales, servicios cloud, y frameworks map-reduce, pero algunos submódulos también ofrecen un soporte robusto sobre las tradicionales bases de datos relacionales.

En concreto, Spring Data JPA es un submódulo que permite implementar repositorios JPA de forma simple. Java Persistence API (JPA) es una API de persistencia desarrollada para Java que gestiona datos relacionales siguiendo el patrón ORM. Spring Data JPA permite implementar la capa ORM de una forma simple y reduce las líneas de código necesarias implementando automáticamente la capa de acceso a los datos en base a interfaces de repositorio escritas por el desarrollador.

1.4.6. JSP

Java Server Pages (JSP) es una de las tecnologías que se pueden utilizar para renderizar una vista a partir de modelos de Spring. JSP permite generar

páginas web dinámicas basadas en documentos estándar de páginas web como HTML o XML. Al ser basado en Java, tiene gran compatibilidad con Spring. Al ser código Java, se trata de código compilado, por lo tanto, ofrece ventajas de estabilidad y rapidez frente a alternativas de código interpretado. Se ejecuta en el lado del servidor, por lo tanto, ofrece facilidad y seguridad a la hora de acceder a bases de datos o hacer otros procesos complejos.

1.4.7. JSTL

JSP Estándar Tag Library (JSTL) es una agrupación de etiquetas JSP estándares y comúnmente utilizadas en aplicaciones web. JSTL abarca tareas estructurales como iteraciones y condicionales y tareas de manipulación de información como documentos XML, i18N y extracción de datos a través de SQL. La colección de etiquetas se puede dividir en una serie de grupos, cada uno de los grupos compone una librería. Para utilizar cualquiera de estas librerías, se puede añadir mediante una directiva 'taglib'.

1.4.8. Maven

Maven es una herramienta de software para la gestión y la construcción de proyectos. Nos permite simplificar los procesos de compilar y generar ejecutables a partir del código fuente, descargar las librerías que nos hagan falta en nuestro proyecto, gestionar las dependencias, etc.

1.4.9. MySQL

MySQL es un sistema de gestión de bases de datos relacional. Es uno de los gestores más utilizados hoy en día. Nos permite, a través de una serie de sentencias, tener una información almacenada en la base de datos y recuperarla en el momento en el que la necesitemos de una forma eficiente y rápida.

1.4.10. Git

Para el control de versiones de nuestro proyecto, hemos escogido Git, un software muy eficiente dada la rapidez con la que gestiona las diferencias entre archivos.

1.4.11. Eclipse Mars

Eclipse es un entorno de desarrollo integrado (IDE) potente y completo en el que encontramos todas las herramientas y funciones necesarias para nuestro trabajo, recogidas además en una atractiva interfaz que lo hace fácil de usar y agradable.

Capítulo 2. Funcionalidad requerida

En este capítulo vamos a tratar de exponer la funcionalidad que nuestro sistema debe llevar a cabo para alcanzar los objetivos trazados. Primero identificamos los actores principales que van a hacer uso de nuestro sistema y posteriormente analizamos todos los requisitos funcionales que se han seleccionado para dicho fin.

2.1. Actores

Principalmente hemos detectado los siguientes tipos de usuarios:

- Administrador
- Bibliotecario
- Encargado
- Invitado

2.2. Requisitos funcionales

Los requisitos funcionales seleccionados son los siguientes:

2.2.1. RF01 – Gestión de usuarios

Dado que tenemos diferentes tipos de usuarios, cada uno con un perfil diferente, es necesario que el sistema pueda llevar a cabo la gestión de éstos, pudiendo crear un nuevo usuario, modificar sus datos o eliminarlo.

RF01.01 – Registrar usuario

El administrador del sistema podrá crear un nuevo usuario asignando a éste un determinado perfil.

RF01.02 – Consultar usuarios

El administrador del sistema podrá consultar el listado de todos los usuarios del sistema.

RF01.03 – Editar usuario

El administrador del sistema podrá modificar los datos de un determinado usuario del sistema.

RF01.04 – Eliminar usuario

El administrador del sistema podrá eliminar un usuario registrado anteriormente.

2.2.2. RF02 – Gestión del depósito

El sistema podrá permitir la consulta de los datos bibliográficos de un listado de fondos que contiene un depósito.

RF02.01 – Consultar fondos

El usuario podrá consultar el listado de fondos del depósito.

RF02.02 – Detalle fondo

El usuario podrá consultar los detalles de un determinado fondo del depósito.

RF01.03 – Buscar fondo

El usuario podrá buscar un fondo en el depósito, introduciendo el título, nombre del autor, o el ISBN del fondo.

2.2.3. RF03 – Gestión del catálogo

Para la gestión del catálogo, el sistema podrá permitir la creación de nuevos fondos y ejemplares de los mismo, importando los registros bibliográficos del depósito. También permitirá la modificación de los datos de los ejemplares y la eliminación de éstos.

RF03.01 – Crear fondo

El administrador y el bibliotecario podrán crear un nuevo fondo importando los datos bibliográficos del depósito.

RF03.02 – Crear ejemplar

El bibliotecario y el administrador podrán crear ejemplares de un determinado fondo.

RF03.03 – Consultar ejemplares

El usuario podrá consultar el listado de los ejemplares que se encuentran en el catálogo incluyendo ver si el ejemplar se encuentra disponible o está prestado.

RF03.04 – Consultar fondos del catálogo

El bibliotecario y el administrador podrán consultar el listado de los fondos del catálogo.

RF03.05 – Buscar ejemplar

El usuario podrá buscar un determinado ejemplar introduciendo el título, nombre del autor, o el ISBN del ejemplar.

RF03.06 – Editar ejemplar

El administrador y el bibliotecario podrán modificar los datos de un ejemplar del catálogo.

RF03.07 – Eliminar ejemplar

El administrador del sistema podrá eliminar un ejemplar del catálogo.

Restricciones:

No se podrá eliminar un ejemplar que esté prestado.

2.2.4. RF04 – Gestión de lectores

Otro aspecto importante, es la gestión de los lectores, ya el lector, junto al ejemplar, son los componentes esenciales para la realización de los préstamos. Por lo tanto, la aplicación permitirá dar de alta a un nuevo lector, modificar sus datos, así como eliminarlo.

RF04.01 – Crear lector

El administrador y el bibliotecario podrán crear un nuevo lector.

RF04.02 – Consultar lectores

El administrador, el bibliotecario y el encargado podrán consultar el listado de todos los lectores dados de alta en el sistema.

RF04.03 – Editar lector

El administrador y el bibliotecario podrán modificar los datos de un lector.

RF04.04 – Eliminar lector

El administrador del sistema podrá eliminar un determinado lector.

Restricciones:

No se podrá eliminar un lector que tiene un préstamo pendiente.

2.2.5. RF05 – Gestión de políticas de préstamo

Para llevar a cabo la gestión de préstamo de ejemplares, es importante definir primero las políticas de préstamo. De modo que el número de préstamos permitidos para un lector depende de qué tipo de lector es; y la duración del préstamo de un ejemplar depende del tipo del lector y también del del ejemplar.

RF05.01 – Consultar políticas

El administrador del sistema podrá consultar las políticas de préstamo.

RF05.02 – Crear tipo lector

El administrador del sistema podrá crear un nuevo tipo de lector asignando a éste el número de préstamos permitidos.

RF05.03 – Editar tipo lector

El administrador del sistema podrá modificar un tipo de lector.

RF05.04 – Eliminar tipo lector

El administrador del sistema podrá eliminar un tipo lector creado anteriormente.

Restricciones:

No se puede eliminar un tipo lector si hay un lector de ese tipo.

RF05.05 – Crear tipo ejemplar

El administrador del sistema podrá crear un nuevo tipo de ejemplar.

RF05.06 – Editar tipo ejemplar

El administrador del sistema podrá modificar un tipo de ejemplar.

RF05.07 – Eliminar tipo ejemplar

El administrador del sistema podrá eliminar un tipo de ejemplar creado anteriormente.

Restricciones:

No se puede eliminar un tipo ejemplar si hay un ejemplar de ese tipo.

RF05.08 – Editar política

El administrador del sistema podrá editar la duración del préstamo de un tipo de ejemplar para un tipo de lector.

2.2.6. RF06 – Gestión de préstamos

Este módulo permitirá gestionar el préstamo de los ejemplares disponibles en el catálogo de la biblioteca, pudiendo prestar un determinado ejemplar a un determinado lector, ver los préstamos realizados, prorrogar préstamo, así como realizar la devolución de un ejemplar prestado.

RF04.01 – Prestar ejemplar

El administrador, el bibliotecario y el encargado podrán prestar un ejemplar a un lector que lo ha solicitado.

Restricciones:

- No se puede prestar un ejemplar que, para el tipo de lector seleccionado, la duración del préstamo según la política, es de cero días.
- No se puede realizar el préstamo si el lector ha superado el máximo número de préstamos permitidos.

RF04.02 – Consultar préstamos

El administrador, el bibliotecario y el encargado podrán consultar el listado de préstamos realizados.

RF04.03 – Prorrogar préstamo

El administrador, el bibliotecario y el encargado podrán prorrogar el préstamo de un ejemplar. La nueva fecha de devolución será calculada por el sistema atendiendo a la política de préstamo correspondiente.

RF04.04 – Devolver ejemplar

El administrador, el bibliotecario y el encargado podrán realizar la devolución de un ejemplar, quedando éste disponible para un posterior préstamo.

2.3. Requisitos no funcionales

En esta sección se detallan los requisitos no funcionales que hemos considerado necesarios.

RNF01 – Control de acceso

El sistema permite el control de acceso mediante usuario y contraseña para las funcionalidades que lo requieran.

RNF02 – Funcionalidades visibles según el perfil de usuario

El sistema mostrará diversas funcionalidades atendiendo al perfil del usuario conectado al sistema.

RNF03 – Acceso simultáneo

El sistema permitirá el acceso simultáneo de varios usuarios.

RNF04 – Usabilidad

El sistema ofrecerá una interfaz gráfica intuitiva para facilitar la navegación y la realización de las operaciones.

RNF05 – Manual de usuario

Los usuarios podrán consultar el manual de usuario al que tendrán acceso en cualquier momento.

RNF06 – Soporte de cualquier navegador

El sistema será accesible vía web desde cualquier navegador.

Capítulo 3. Diseño de la aplicación

Después de definir la funcionalidad requerida para nuestro sistema, nos disponemos a describir cómo se ha diseñado la aplicación.

3.1. Actores

Tal y como hemos expuesto en el capítulo anterior, tenemos los siguientes cuatro actores: Administrador, bibliotecario, encargado e invitado. De tal modo que diferentes actores pueden realizar diferentes acciones. Por lo tanto, tenemos que el administrador tiene un control total del sistema, seguido por el bibliotecario que puede realizar las mismas acciones que el administrador menos el borrado de fondos. El encargado puede realizar préstamos y devoluciones y las consultas del catálogo y del depósito. Y, finalmente, el invitado sólo puede realizar las consultas del catálogo y del depósito. Para reflejar esta separación entre usuarios, hemos establecido la jerarquía que podemos contemplar en la Ilustración 3.1.

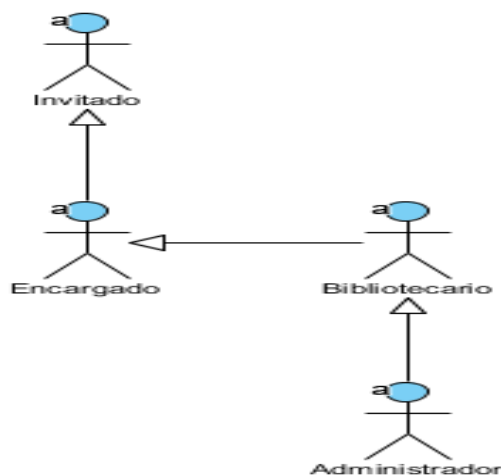


Ilustración 3.1. Jerarquía de usuarios.

3.2. Diagramas de Casos de uso

En esta sección, vamos a definir los casos de uso de nuestro sistema, así como los actores asociados a ellos.

3.2.1. Gestión de usuarios

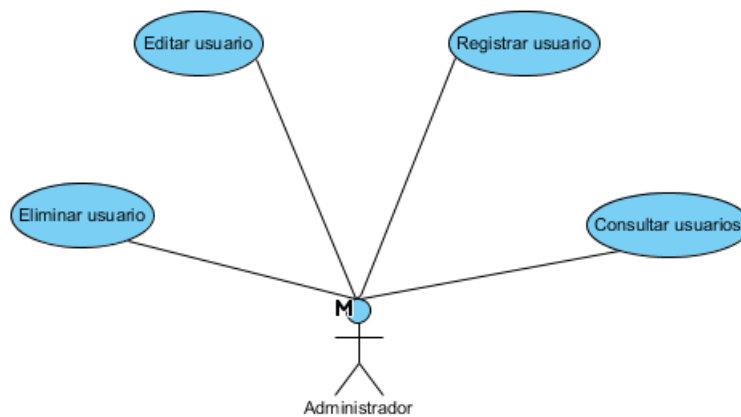


Ilustración 3.2. Gestión de usuarios.

3.2.2. Gestión del depósito

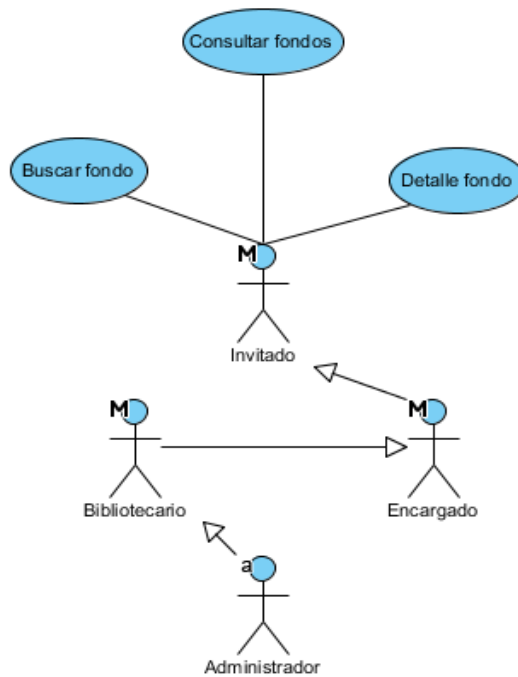


Ilustración 3.3. Gestión del depósito.

3.2.3. Gestión del catálogo

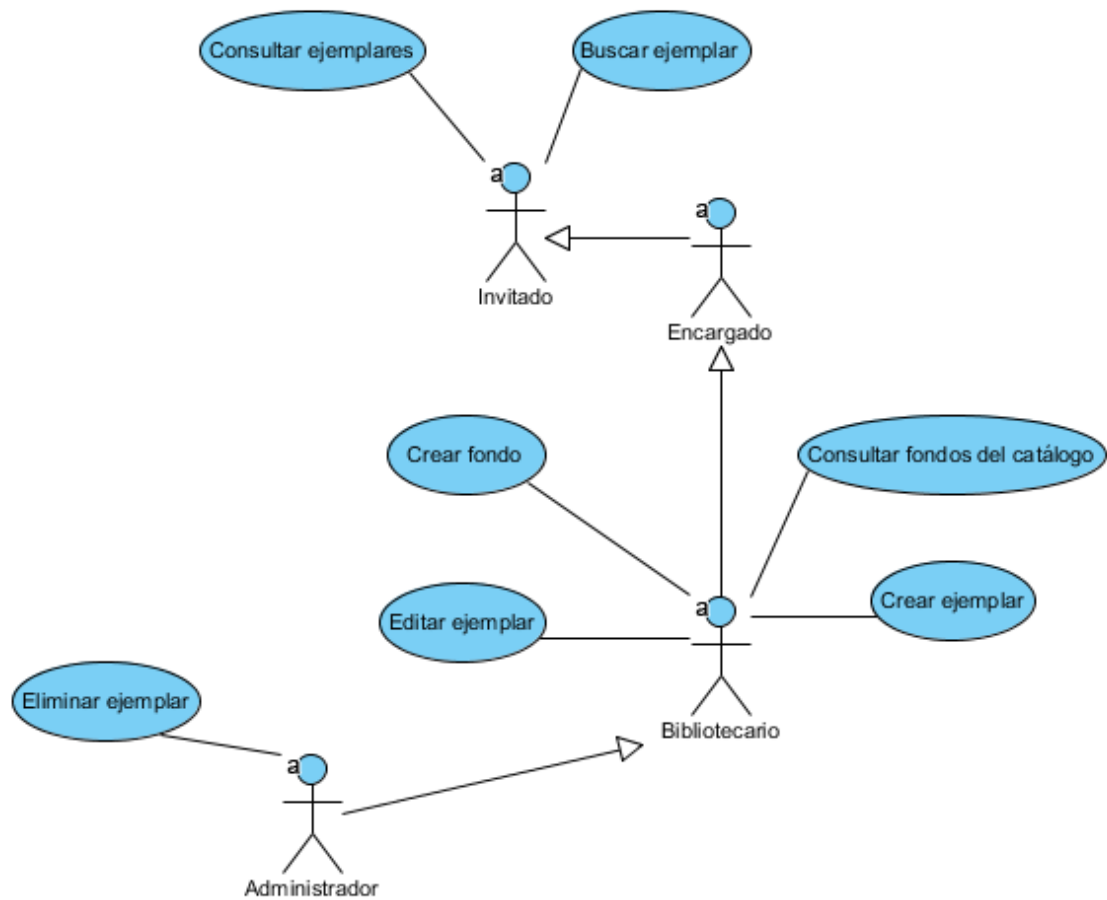


Ilustración 3.4. Gestión del catálogo.

3.2.4. Gestión de lectores

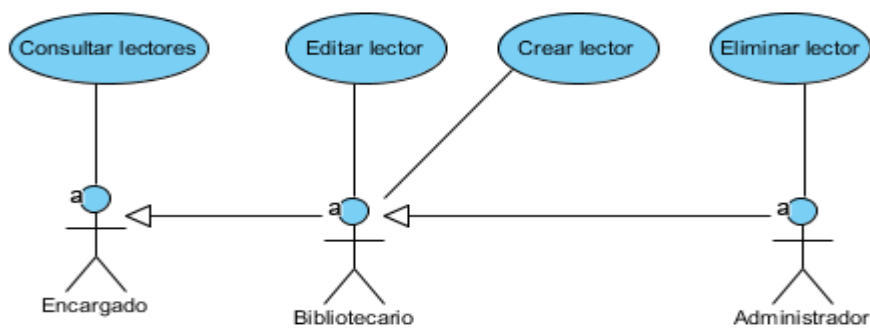


Ilustración 3.5. Gestión de lectores.

3.2.5. Gestión de políticas de préstamo

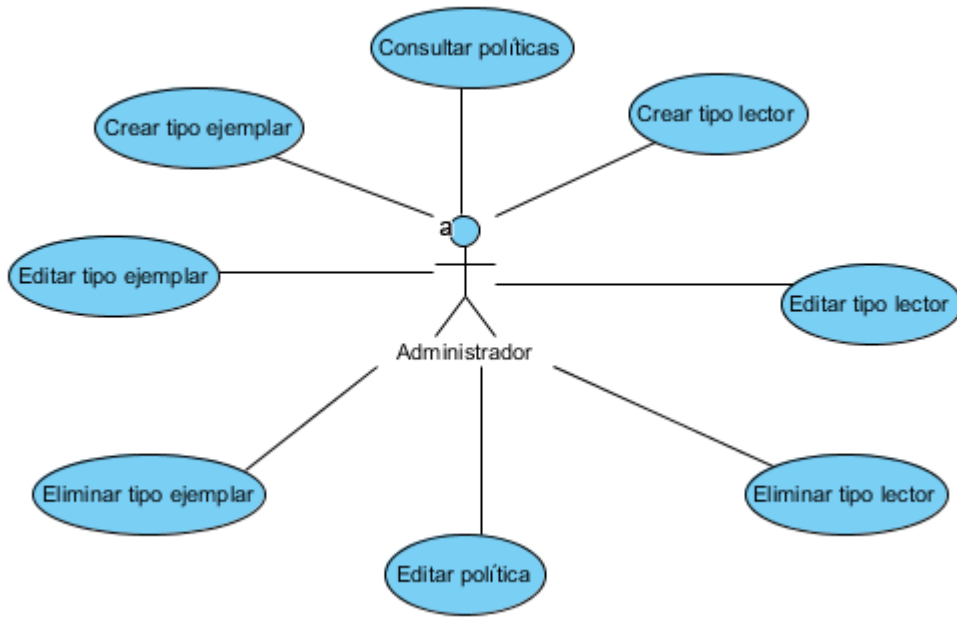


Ilustración 3.6. Gestión de políticas de préstamo.

3.2.6. Gestión de préstamos

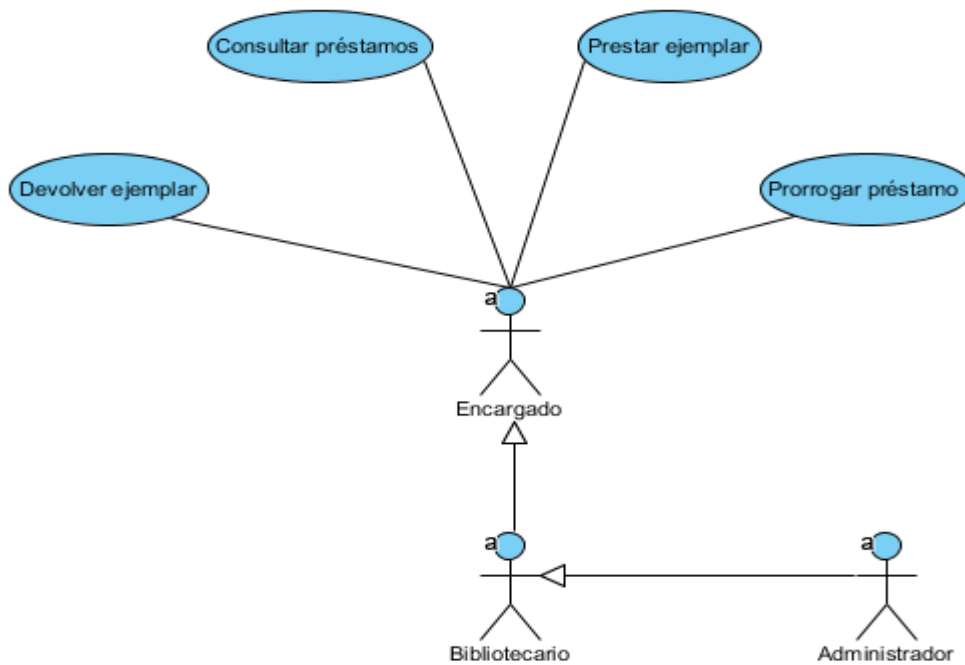


Ilustración 3.7. Gestión de préstamos.

3.3. Modelo Conceptual

En este apartado mostramos el diagrama de clases que representa a modo conceptual el dominio de la aplicación. Se trata de un modelo abstracto diseñado con el objetivo de ayudar a comprender el contexto del problema.

ABIES

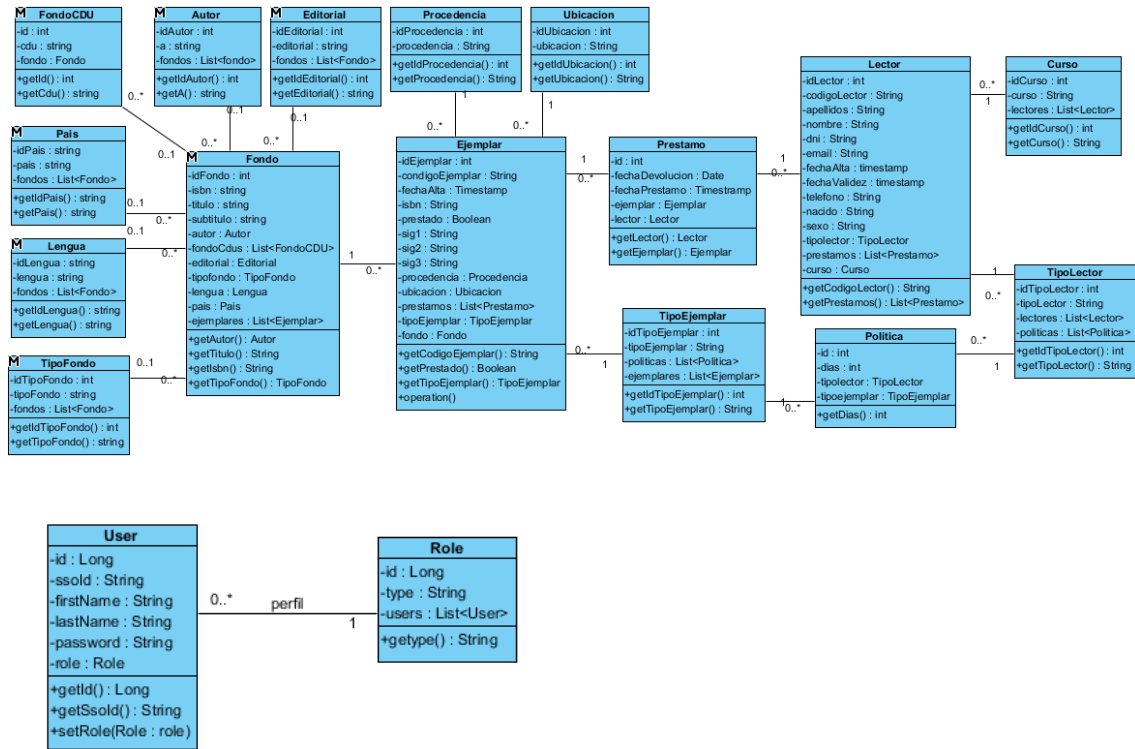


Ilustración 3.8. Diagrama de clases de Abies.

Depósito

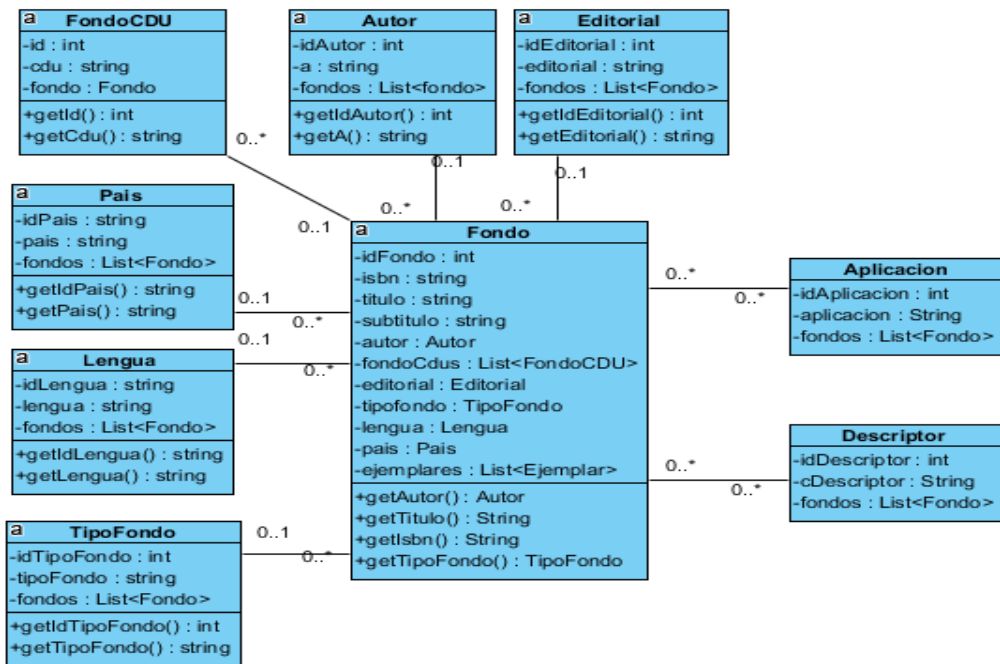


Ilustración 3.9. Diagrama de clases del depósito.

3.4. Modelo Relacional

A continuación, mostramos el diagrama entidad-relación

ABIES

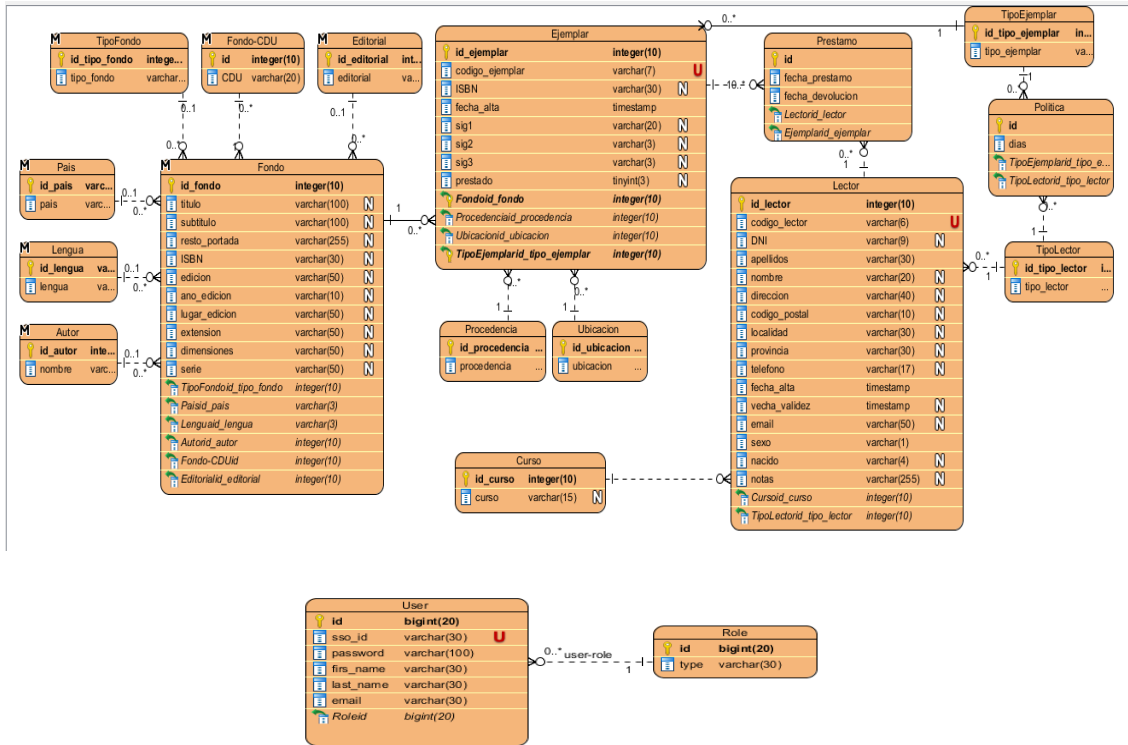


Ilustración 3.10. Diagrama entidad-relación de Abies.

Depósito

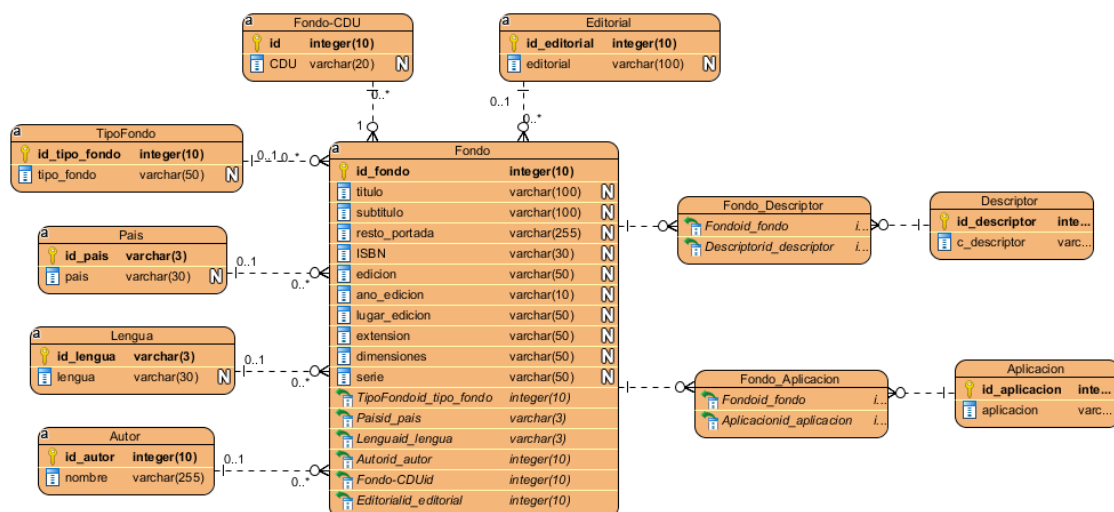


Ilustración 3.11. Diagrama entidad-relación del depósito.

3.5. Diagrama de paquetes

En esta sección, mostramos la estructura de los paquetes de la aplicación mediante el diagrama de paquetes.

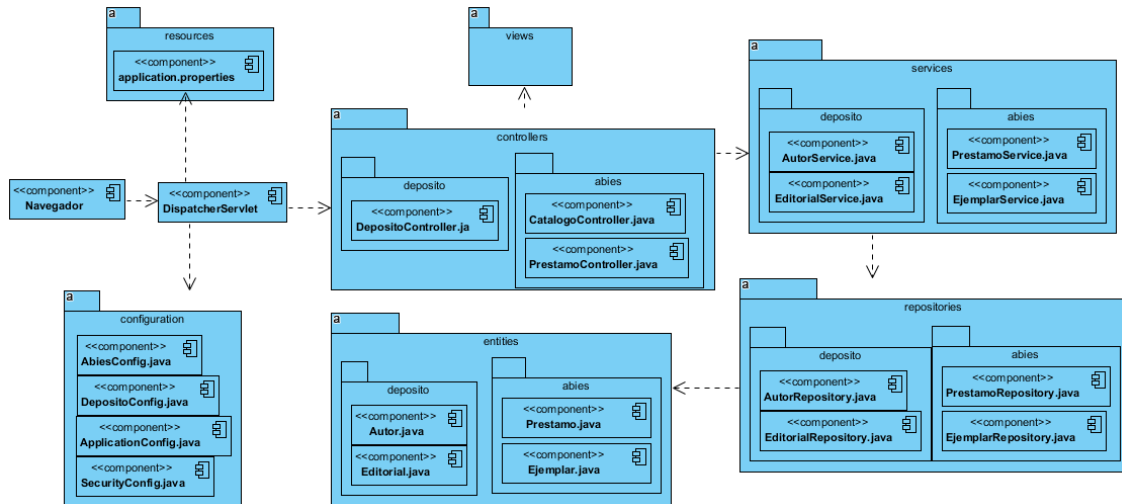


Ilustración 3.12. Diagrama de paquetes de la aplicación.

3.6. Diagrama de despliegue

A continuación, mostramos el diagrama de despliegue.

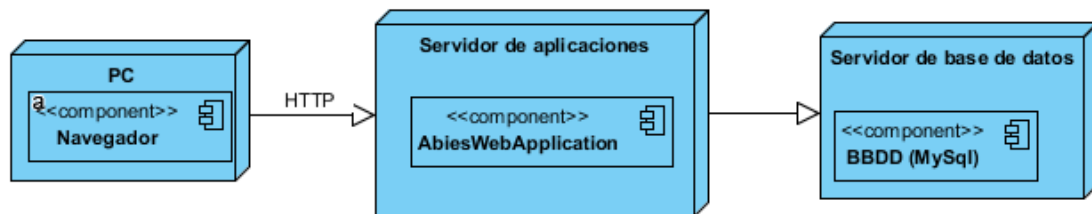


Ilustración 3.13. Diagrama de despliegue.

Capítulo 4. Implementación

El objetivo del presente capítulo es que, al final del mismo, el lector sea capaz de comprender el funcionamiento interno de la aplicación. De esta manera, cualquiera que lo desee podría continuar desarrollando funcionalidades que no hemos incluido en esta versión de la migración.

Por supuesto, no vamos a detallar absolutamente todos los casos de uso, pero sí que haremos un repaso general de la aplicación.

Antes de todo, primero hablaremos de la metodología de desarrollo que hemos aplicado para la realización del trabajo.

4.1. Metodología de desarrollo

Como metodología para el desarrollo de nuestro sistema, hemos escogido la metodología iterativa e incremental. En esta metodología el proyecto se planifica en diversos bloques temporales llamado iteraciones. En cada iteración se evoluciona el producto a partir de los resultados completados en las iteraciones anteriores, añadiendo nuevos objetivos/requisitos o mejorando los que ya fueron completados. De esta manera, cada nueva versión, aumenta la funcionalidad y mejora en calidad respecto a la versión anterior.

En todas las iteraciones se repite un proceso de trabajo similar para proporcionar un resultado completo sobre el producto final. En nuestro caso, este proceso lo hemos dividido en las siguientes tareas:

- Seleccionar los requisitos que implementaríamos durante la iteración.
- Transformar los requisitos seleccionados en casos de uso.
- Desarrollar los casos de usos.
- Crear casos de prueba correspondientes a los casos de uso desarrollados.
- Actualizar el modelo relacional.
- Implementar.

En los siguientes apartados, vamos a exponer qué casos de uso se han desarrollado en cada iteración, y detallar los aspectos más importantes de cada uno de ellos. La especificación de cada caso de uso se encuentra en el [Anexo I.1](#) de este documento.

4.2. Iteración cero

Es muy frecuente en las metodologías iterativas el uso de la llamada iteración cero, cuyo objetivo son los preparativos previos a comenzar el desarrollo.

En nuestro caso hemos aprovechado esta iteración para realizar diversas tareas que detallamos a continuación:

4.2.1. Estudio inicial

Puesto que el objetivo principal de nuestro trabajo es la migración de una aplicación de escritorio a una aplicación web, hemos tenido que hacer un estudio previo de la funcionalidad de la aplicación que vamos a migrar identificando los requisitos que vamos a implementar, así como las tecnologías y las herramientas necesarias para llevar a cabo dicho desarrollo.

4.2.2. Búsqueda de información

Una vez escogidas las tecnologías necesarias para desarrollar nuestra aplicación, hemos procedido a buscar información acerca del aprendizaje y la familiarización con las distintas tecnologías. Esta tarea se verá repetitiva en cada iteración.

4.2.3. Instalar las herramientas

Una vez hemos obtenido la información principal, procedemos a instalar las herramientas necesarias para comenzar el desarrollo de nuestra aplicación.

4.2.4. Creación de la base de datos

Cabe destacar que la base de datos de la aplicación de escritorio es una base de datos MS-ACCESS. Por facilidad de configuración en Spring, hemos preferido convertir la base de datos a MySQL. Para ello, hemos utilizado una aplicación externa (Bullzip MS Access to MySQL).

4.2.5. Creación de la estructura del proyecto

A la hora de crear el proyecto en nuestro IDE Eclipse, elegimos la opción de Spring Starter Project. Dado que es un proyecto Maven, éste se encarga de la construcción del proyecto con su correspondiente estructura.

Podemos ver en la ilustración 4.1 cómo queda la estructura del proyecto.

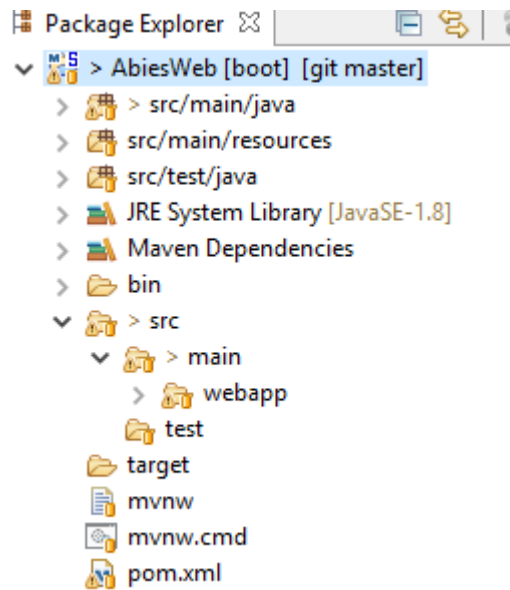


Ilustración 4.1. Estructura del proyecto.

- **src/main/java:** es un directorio donde se almacena todo el código java de la aplicación.
- **src/main/resources:** es un directorio donde se almacenan los recursos de la aplicación, como por ejemplo el fichero de propiedades de la aplicación.
- **src/test/java:** es un directorio donde se almacenan el código de testing de la aplicación.
- **src/main/webapp:** es un directorio donde se almacenan las vistas de la aplicación, sus recursos necesarios, y la configuración de la parte frontal.

Además de lo mencionado anteriormente, toda aplicación Maven dispone del archivo Pom (Project Object Model). Es un fichero xml que contiene información acerca de las dependencias, plugins utilizados, versiones, etc. Esto facilita mucho la labor de incluir librerías de terceros que vayamos a necesitar. En la Ilustración 4.2, podemos observar cómo hemos podido añadir fácilmente algunas de las librerías de Spring que vamos a necesitar.

```

<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>

  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-security</artifactId>
  </dependency>
</dependencies>

```

Ilustración 4.2. Dependencias Maven.

Para organizar mejor el proyecto, además de la estructura que crea Maven, hemos estructurado el proyecto en varios paquetes tal y como se puede observar en la ilustración 4.3.

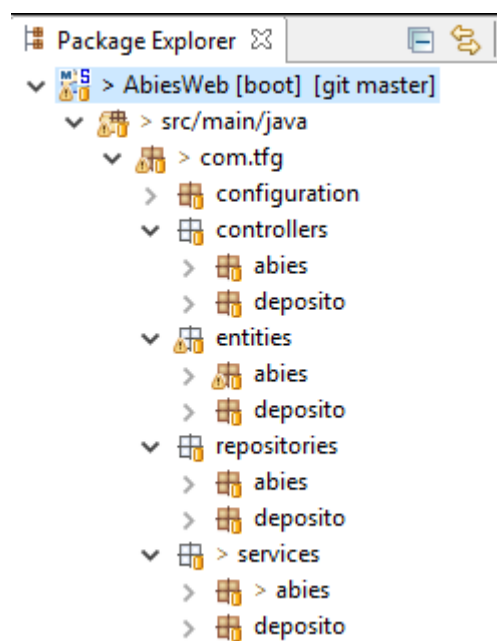


Ilustración 4.3. Organización de los paquetes.

De manera que en cada paquete guardamos los siguientes ficheros:

- **com.tfg.configuration:** almacenamos las clases relativas a la configuración de la aplicación.
- **com.tfg.controllers:** guardaremos los controladores.

- **com.tfg.entities:** guardaremos las clases mapeadas desde la base de datos.
- **com.tfg.repositories:** guardaremos las clases repositorio.
- **com.tfg.services:** guardaremos las clases service.

4.2.6. Creación del repositorio

Tal y como hemos mencionado anteriormente, hemos utilizado la herramienta Git para el control de las versiones de nuestro proyecto. Para ello hemos creado el repositorio y las distintas ramas que corresponden a las distintas iteraciones en las que se ha dividido en trabajo. (Ver ilustración 4.4)

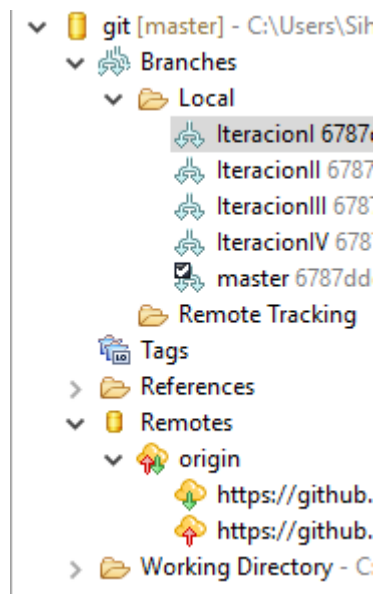


Ilustración 4.4. Gestión de versiones.

4.2.7. Configuración del proyecto

Llegados a este punto, antes de comenzar el desarrollo de la aplicación, es necesario configurar el proyecto, como por ejemplo configurar la conexión a la base de datos. Dado que nuestra aplicación se conectará a otra base de datos, el depósito, la configuración de cada conexión se implementará en dos clases diferentes. (Ver ilustración 4.5 y 4.6)

```

@Configuration
@PropertySource({ "classpath:application.properties" })
@EnableJpaRepositories(
    basePackages = "com.tfg.repositories.abies",
    entityManagerFactoryRef = "abiesEntityManager",
    transactionManagerRef = "abiesTransactionManager"
)
public class AbiesConfig {
    @Autowired
    @Bean
    @Qualifier(value = "abiesTransactionManager")
    public JpaTransactionManager transactionManager(EntityManagerFactory emf) {
        return new JpaTransactionManager(emf);
    }

    @Bean
    @Primary

```

Ilustración 4.5. Configuración de la base de datos de ABIES

```

@Configuration
@PropertySource({ "classpath:application.properties" })
@EnableTransactionManagement
@EnableJpaRepositories(
    basePackages = "com.tfg.repositories.deposito",
    entityManagerFactoryRef = "depositoEntityManager",
    transactionManagerRef = "depositoTransactionManager"
)
public class DepositoConfig {
    @Autowired
    @Bean
    @Qualifier(value = "depositoTransactionManager")
    public JpaTransactionManager transactionManager(EntityManagerFactory emf) {
        return new JpaTransactionManager(emf);
    }

    @Bean

```

Ilustración 4.6. Configuración de la base de datos del depósito

4.3. Iteración I

Una vez configurado el proyecto, comenzamos con la iteración uno. En esta iteración hemos seleccionado los requisitos funcionales mostrados en la siguiente tabla.

Requisitos funcionales	Casos de uso derivados
<u>RF01 – Gestión de usuarios</u>	<u>CU01 – Gestión de usuarios</u>
<u>RF02 – Gestión del depósito</u>	<u>CU02 – Gestión del depósito</u>

A continuación, vamos a explicar la implementación del caso de uso CU01.

4.3.1. Acceso a datos

Para la gestión de usuarios, necesitamos crear dos tablas en la base de datos, las que corresponden en el modelo relacional a User y Role. Una vez creadas las tablas, procedemos a mapearlas usando el estándar JPA.

Para empezar a utilizar Spring Data JPA es necesario añadir su módulo.

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
```

Ilustración 4.7. Módulo Spring Data JPA.

Al mapear las tablas, obtenemos las entidades correspondientes como se puede observar en la ilustración 4.8.

Todas las tablas se definen con la anotación `@Entity` para indicar que serán utilizadas por Spring Data.

```
@Entity
@NamedQuery(name = "User.findAll", query = "SELECT u FROM User u")
public class User implements Serializable {
    private static final long serialVersionUID = 1L;
    private long id;
    private String email;
    private String firstName;
    private String lastName;
    private String password;
    private String ssoId;
    private String state;
    private Role role;

    public User() {
    }

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    public long getId() {
        return this.id;
    }

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

Ilustración 4.8. Entidad User.

El siguiente paso es codificar un repositorio que enlace esta entidad. El repositorio será tan sólo una interfaz específica para cada entidad que JPA es capaz de implementar en tiempo de ejecución de forma automática.

El siguiente repositorio mostrado en la ilustración 4.9 funciona con entidades de tipo User.

```
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.query.Param;
import com.tfg.entities.abies.User;

public interface UserRepository extends JpaRepository<User,Long> {
    @Query("SELECT u FROM User u WHERE u.ssoId = :sso")
    User findBySso(@Param("sso")String sso);
}
```

Ilustración 4.9. Interfaz UserRepository.

UserRepository extiende de la interfaz JpaRepository, una extensión de Repository específica para JPA. Los parámetros son User y Long, que indican el tipo de la entidad y del ID respectivamente. Debido a la herencia, UserRepository ya tiene los métodos básicos tipo CRUD para encontrar, guardar y borrar entidades de tipo User.

Si queremos realizar consultas más específicas, hay distintas formas de hacerlo, una de ellas es escribiendo la consulta en la anotación @Query tal y como se ve en la Ilustración 4.9. Si la consulta en cuestión modifica el estado de la base de datos, tenemos que añadir la etiqueta @Modifying.

Una vez creadas las entidades y su correspondiente repositorio, procedemos a codificar la capa de service, que son las que se encargan de realizar las acciones solicitadas por los controladores, procesando la información necesaria y hacerla llegar a los repositorios. Son clases anotadas con la anotación @Service en las que podemos declarar una interfaz repositorio y acceder a ella. (Ver ilustración 4.10).

```
import java.util.List;
@Service("UsuarioService")
@Transactional(value = "abiesTransactionManager")
public class UsuarioServiceImpl implements UsuarioService {

    @Resource
    private UserRepository userRepository;

    @Override
    public List<User> findAll() {
        return userRepository.findAll();
    }

    @Override
    public void save(User user) {
        userRepository.save(user);
    }
}
```

Ilustración 4.10. Implementación del servicio usuario.

4.3.2. Spring MVC

Llegados a este punto, vamos a crear la capa controladora. Los controladores son los responsables de preparar un modelo y elegir el nombre de la vista. El modelo es una interfaz que permite abstraerse del tipo de tecnología que se está utilizando para la vista, transformándose al formato adecuado cuando se solicita. Las vistas, en nuestro caso, se generan utilizando la tecnología JSP junto a JSTL.

La clase controladora es anotada con `@Controller`. Los métodos de esta clase llevan la anotación `@RequestMapping`. En ella, se pasa por parámetro el valor, que es la URI que debe mapearse con ese método (la petición a la que se va a responder) y el tipo de peticiones HTTP a la que responde (GET, POST, PUT, DELETE). El método devuelve un objeto de tipo `ModelAndView` en que pasaremos los datos que vamos a necesitar en la vista, así como el nombre de la propia vista que se va a cargar.

Es necesario declarar `usuarioService` y `rolService` para poder acceder a las funcionalidades que encapsulan. (Ver ilustración 4.11).

```
@Controller
@RequestMapping(value = "/usuarios")
public class UsuariosController {

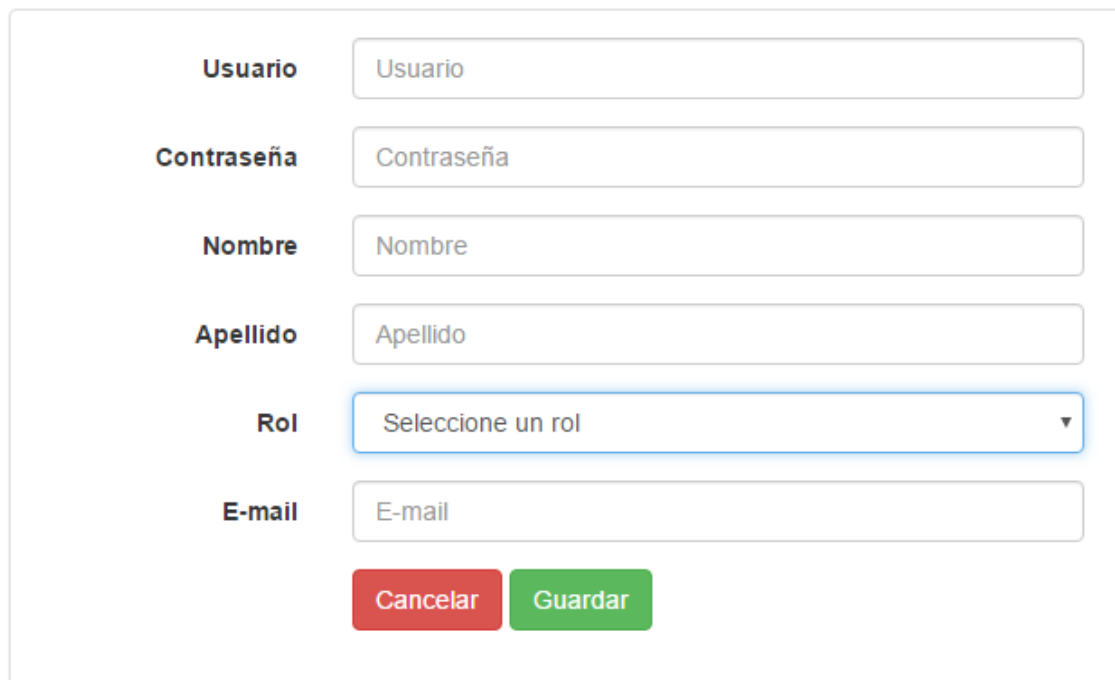
    @Autowired
    public UsuarioService usuarioService;
    @Autowired
    public RolService rolService;

    @RequestMapping(value = { "/nuevo" }, method = RequestMethod.GET)
    public ModelAndView usuarioNuevo() {
        ModelAndView mav = new ModelAndView("/usuarios/usuarios-new");
        User user = new User();
        List<Role> roles = rolService.findAll();
        mav.addObject("roles", roles);
        mav.addObject("user", user);
        return mav;
    }
}
```

Ilustración 4.11. La clase controladora.

Por ejemplo, en el caso crear un nuevo usuario, se carga la vista correspondiente para rellenar los datos solicitados. (Ver ilustración 4.12).

Nuevo usuario



Formulario de creación de nuevo usuario con los siguientes campos:

- Usuario:** Campo de texto con el valor "Usuario".
- Contraseña:** Campo de texto con el valor "Contraseña".
- Nombre:** Campo de texto con el valor "Nombre".
- Apellido:** Campo de texto con el valor "Apellido".
- Rol:** Selector de lista desplegable con el valor "Seleccione un rol".
- E-mail:** Campo de texto con el valor "E-mail".

Botones de acción:

- Cancelar:** Botón rojo.
- Guardar:** Botón verde.

Ilustración 4.12. Vista de crear nuevo usuario.

4.3.3. Paginación

Puesto que el depósito contiene una gran cantidad de fondos, es necesario utilizar la paginación para poder mostrarlos en la web.

En este caso, la interfaz repositorio de la entidad en cuestión, extiende de `PagingAndSortingRepository` para indicar que puede ser paginable (ver ilustración 4.13).

```
public interface FondoCrRepository extends PagingAndSortingRepository<FondoCr, Integer>{  
  
    @Query("SELECT f FROM FondoCr f WHERE f.titulo = :tit")  
    List<FondoCr>findByTitulo(@Param("tit") String titulo);  
  
    @Query("SELECT f FROM FondoCr f WHERE f.autorCr = :aut")  
    List<FondoCr>findByAutor(@Param("aut") AutorCr autor);  
  
    @Query("SELECT f FROM FondoCr f WHERE f.isbn = :cod")  
    List<FondoCr>findByISBN(@Param("cod") String isbn );  
  
}
```

Ilustración 4.13. Interfaz FondoRepository.

El siguiente paso es implementar un nuevo método en el service como se puede ver en la ilustración 4.14. En este caso, en cada página se cargan 50

fondos. Con `Sort.Direction.ASC` indicamos que los fondos se ordenan con un orden ascendente y el parámetro del orden es el ID del fondo.

```
private static final int PAGE_SIZE = 50;
@Override
public Page<FondoCr> listAllByPage(Integer pageNumber) {
    // TODO Auto-generated method stub
    PageRequest request =
        new PageRequest(pageNumber - 1, PAGE_SIZE, Sort.Direction.ASC, "idFondo");
    return fondoCrRepository.findAll(request);
}
```

Ilustración 4.14. Método de paginación.

A continuación implementamos el controlador. Podemos ver en la ilustración 4.15 que el método implementado anteriormente en el service nos devuelve la página cargada del número de fondos que le hemos indicado. Por lo tanto, creamos una nueva lista en la que almacenamos los fondos y se la pasamos a la vista mediante el objeto `ModelAndView`.

```
@RequestMapping(value = "/pages/{pageNumber}", method = RequestMethod.GET)
public ModelAndView getFondoCrPage(@PathVariable Integer pageNumber, HttpServletRequest request) {
    ModelAndView mav = new ModelAndView("/deposito/deposito-list");
    Page<FondoCr> page = fondoCrService.listAllByPage(pageNumber);
    int current = page.getNumber() + 1;
    int begin = Math.max(1, current - 5);
    int end = Math.min(begin + 10, page.getTotalPages());
    mav.addObject("paginas", page);
    List<FondoCr> fondos = new ArrayList<FondoCr>();
    mav.addObject("beginIndex", begin);
    mav.addObject("endIndex", end);
    mav.addObject("currentIndex", current);
    for (FondoCr fondo : page) {
        fondos.add(fondo);
    }
    mav.addObject("control", "deposito");
    mav.addObject("fondos", fondos);
    return mav;
}
```

Ilustración 4.15. Método de la clase controladora.

En la ilustración 4.16 podemos ver cómo ha quedado el listado de fondos una vez utilizada la paginación.

Fondos del depósito

	Título	País	Lengua	Edición	Lugar de edición
QVer	El señor de la coca	España	Español	1ª ed.	Barcelona
QVer	Hardwired	España	Español		Barcelona
QVer	Año de nieves, año de bienes	España	Español		Barcelona
QVer	¡Dí que no!	España	Español		Barcelona
QVer	Las aventuras del caballero Tirant	España	Español		Barcelona
QVer	Evaluación de programas educativos, centros y profesores	España	Español		Madrid

Ilustración 4.16. Vista de listado de fondos del depósito.

4.4. Iteración II

Una vez terminada y comprobada a la primera iteración, comenzamos con la segunda. Los requisitos que hemos seleccionado esta vez son los siguientes.

[RFN01 – Control de acceso](#)

[RFN02 – Funcionalidades visibles según el perfil](#)

[RFN03 – Acceso simultáneo](#)

4.4.1. Spring Security

Como ocurre con el resto de módulos de Spring, Spring Security se adapta a los estándares de seguridad para que su configuración se base principalmente en especificar requisitos propios de la aplicación. Spring Boot ofrece una configuración inicial que permite utilizar Spring Security al instante.

Para integrar Spring Security con Maven en un proyecto Spring Boot, podemos integrar su módulo preconfigurado como podemos ver en la ilustración 4.17.

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-security</artifactId>
</dependency>
```

Ilustración 4.17. Módulo de Spring Security.

Con el módulo cargado, codificamos una clase de configuración con las especificaciones de acceso para nuestra aplicación. (Ver ilustración 4.18)


```

@Configuration
@EnableWebSecurity
public class SecurityConfiguration extends WebSecurityConfigurerAdapter {

    @Autowired
    @Qualifier("customUserDetailsService")
    UserDetailsService userDetailsService;

    @Autowired
    public void configureGlobalSecurity(AuthenticationManagerBuilder auth) throws Exception {
        auth.userDetailsService(userDetailsService);
    }
}

```

Ilustración 4.18. Configuración de Spring Security.

La anotación `@Configuration` indica que se trata de una clase de configuración y `@EnableWebSecurity` integra esta clase de configuración de seguridad con `spring-web-security`.

`UserDetailsService` es una interfaz que nos proporciona Spring Security a modo de DAO. Tenemos que implementarla para poder acceder a los usuarios que tenemos almacenados en nuestra base de datos. (Ver Ilustración 4.19)

```

@Service("customUserDetailsService")
public class CustomUserDetailsService implements UserDetailsService{

    @Autowired
    private UsuarioService userService;

    @Transactional(readOnly=true)
    public UserDetails loadUserByUsername(String ssoId)
        throws UsernameNotFoundException {
        User user = userService.findBySso(ssoId);
        System.out.println("User : "+user);
        if(user==null){
            System.out.println("User not found");
            throw new UsernameNotFoundException("Username not found");
        }
        return new org.springframework.security.core.userdetails.User(user.getSsoId(), user.ge
            user.getState().equals("Active"), true, true, true, getGrantedAuthorities(user));
    }
}

```

Ilustración 4.19. Configuración del acceso de usuarios.

Con este método comprobamos si un determinado usuario existe en base de datos y si es activo y cargamos sus permisos. Estos permisos se pueden establecer de diferentes formas, según sea conveniente.

Spring genera un objeto de la clase `HttpSecurity` y lo envía como parámetro en el método `configure` (`HttpSecurity http`) (ver ilustración 4.20) donde se especifica qué acciones de seguridad deben aplicarse en esta página web. Se indican las peticiones que están autorizadas mediante el método `authorizeRequests()` del objeto de la clase `HttpSecurity`. Se especifican URLs básicas mediante `antMatchers`, que mapea patrones de URL de tipo Ant. De esta manera restringimos el acceso a las URLs de nuestra aplicación.

```

@Override
protected void configure(HttpSecurity http) throws Exception {
    http.authorizeRequests()
        .antMatchers("/principal").permitAll()
        .antMatchers("/usuarios/**").access("hasRole('ADMIN')")
        .antMatchers("/politica/**").access("hasRole('ADMIN')")
        .antMatchers("/lectores/list").access("hasAnyRole('ADMIN', 'BIBLIOTECARIO', 'ENCARGADO')")
        .antMatchers("/prestamos/**").access("hasAnyRole('ADMIN', 'BIBLIOTECARIO', 'ENCARGADO')")
        .antMatchers("/catalogo/ejemplar/**").access("hasAnyRole('ADMIN', 'BIBLIOTECARIO')")
}

```

Ilustración 4.20. Restringido de las URLs.

Spring security ofrece su propio taglib que permite a los archivos JSP acceder a información de seguridad de la aplicación y permite aplicar restricciones de seguridad. Para integrar el taglib, añadimos la siguiente dependencia (ver ilustración 4.21)

```

<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-taglibs</artifactId>
</dependency>

```

Ilustración 4.21. Spring Security taglibs.

De esta manera, podemos controlar la visibilidad de algunas partes de la página JSP. Por ejemplo, tenemos en la barra de navegación secciones que son visibles dependiendo del tipo de usuario autenticado. Con el tag <sec:authorize> estamos indicando que todo el contenido que encierra, será visible únicamente por los perfiles indicados en el atributo access. (ver ilustración 4.22)

```

<sec:authorize
    access="hasAnyRole('ADMIN', 'BIBLIOTECARIO', 'ENCARGADO')">
    <li><a href="/lectores/list"><i class="fa fa-users fa-lg"
        aria-hidden="true"></i> Lectores</a></li>
    <li><a href="/prestamos/list"><i
        class="fa fa-check-square-o fa-lg" aria-hidden="true"></i>Préstamos</a></li>
</sec:authorize>
<sec:authorize access="hasRole('ADMIN')">
    <li><a href="/usuarios/list"><i class="fa fa-ban fa-lg"
        aria-hidden="true"></i> Control de acceso</a></li>
</sec:authorize>

```

Ilustración 4.22. Control de visibilidad en Spring Security.

4.5. Iteración III

En esta iteración, implementamos todas las funcionalidades relacionadas con la gestión del catálogo.

Requisitos funcionales	Casos de uso derivados
<u>RF03 – Gestión del catálogo</u>	<u>CU03 – Gestión del catálogo</u>

4.6. Iteración IV

Para esta iteración, hemos seleccionado los siguientes requisitos funcionales.

Requisitos funcionales	Casos de uso derivados
RF04 - Gestión de lectores	CU04 - Gestión de lectores

4.7. Iteración V

Antes de proceder a la realización de gestión de préstamos, es necesario una iteración anterior en la que implementamos las políticas de préstamo.

Requisitos funcionales	Casos de uso derivados
RF05 - Gestión de políticas de préstamo	CU05 - Gestión de políticas de préstamo

4.8. Iteración VI

Una vez terminada y validada la iteración anterior, comenzamos la implementación de la gestión de préstamos.

Requisitos funcionales	Casos de uso derivados
RF06 - Gestión de préstamos	CU06 - Gestión de préstamos

4.8.1. Validación de datos

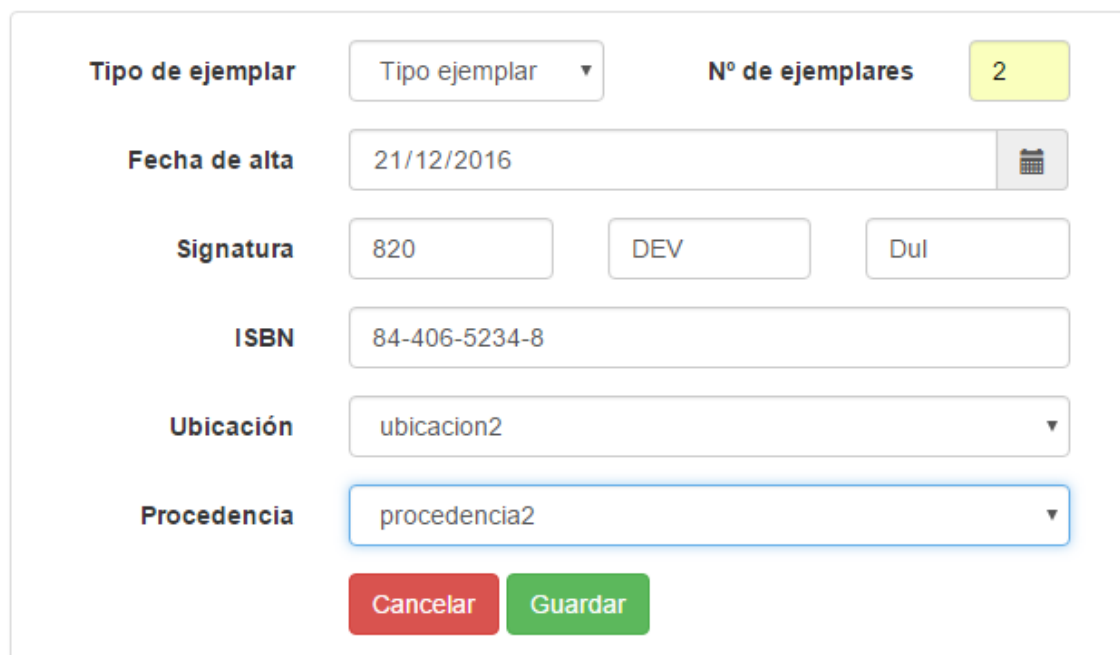
Cuando disponemos de formularios que han de ser rellenados por el usuario, es importante realizar la validación de los datos introducidos, antes de proceder a su manipulación, ya que éstos podrían no ser los adecuados y producirían problemas innecesarios. Para ello, vamos a necesitar las dos siguientes librerías:

```
<dependency>
  <groupId>javax.validation</groupId>
  <artifactId>validation-api</artifactId>
  <version>1.1.0.final</version>
</dependency>
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-validator</artifactId>
  <version>5.3.3.Final</version>
</dependency>
```

Ilustración 4.23. Librerías de validación de datos.

A continuación, vamos a explicar cómo funciona la validación de datos utilizando un ejemplo del formulario de añadir un nuevo ejemplar al catálogo de la biblioteca, suponiendo que el usuario no selecciona el tipo de ejemplar del ejemplar que quiere añadir.

Nuevo ejemplar




Tipo de ejemplar	Tipo ejemplar ▾	N° de ejemplares	2
Fecha de alta	21/12/2016 		
Signatura	820	DEV	Dul
ISBN	84-406-5234-8		
Ubicación	ubicacion2 ▾		
Procedencia	procedencia2 ▾		
Cancelar Guardar			

Ilustración 4.24. Vista de crear nuevo ejemplar.

Como podemos ver en el modelo relacional, la entidad Ejemplar tiene una relación con la entidad TipoEjemplar (N.1), Por tanto, a la hora de mapear las tablas obtenemos las siguientes clases con los atributos y sus respectivos getters y setters.

```
@Entity
@NamedQuery(name="Ejemplar.findAll", query="SELECT e FROM Ejemplar e")
public class Ejemplar implements Serializable {
    private static final long serialVersionUID = 1L;
    private int idEjemplar;
    private Tipoejemplar tipoejemplar;
    private String codigoEjemplar;
    private Timestamp fechaAlta;
```

Ilustración 4.25. Entidad Ejemplar.

```
@Entity
@NamedQuery(name="Tipoejemplar.findAll", query="SELECT t FROM Tipoejemplar t")
public class Tipoejemplar implements Serializable {
    private static final long serialVersionUID = 1L;
    private int idTipoJemplar;
    private String tipoEjemplar;
    private List<Ejemplar> ejemplars;
    private List<Politica> politicas;
```

Ilustración 4.26. Entidad TipoEjemplar.

Como el campo que vamos a validar es de un tipo de otra clase, entonces utilizamos grupos de validación.

El primer paso es crear una interfaz vacía en la clase Ejemplar.

```
public interface createEjemplar{  
  
}
```

Ilustración 4.27. Interfaz createEjemplar.

A continuación, ponemos las anotaciones en método getTipoEjemplar() de la clase Ejemplar y en el método getIdTipoEjemplar() de la clase TipoEjemplar y los asignamos al mismo grupo de validación.

```
@NotNull(groups = {Ejemplar.createEjemplar.class}, message="campo obligatorio")  
@Valid  
public Tipoejemplar getTipoejemplar() {  
    return this.tipoejemplar;  
}  
  
@Min(value = 1, groups = Ejemplar.createEjemplar.class, message="campo obligatorio")  
@Valid  
public int getIdTipoJemplar() {  
    return this.idTipoJemplar;  
}
```

Ilustración 4.28. Validación de datos.

La anotación @Valid obliga a que se valide el campo en cuestión cumpliendo con la anotación que tiene por encima.

En la JSP, necesitamos indicar que los campos del formulario necesitan ser validados. Esto lo indicamos con un tag adicional (<form:errors>), que nos mostrará el mensaje en caso de error.

```
<form:select path="tipoejemplar.idTipoJemplar"  
    class="form-control">  
    <form:option value="-1" label="Tipo ejemplar" />  
    <form:options items="{tiposEjemplar}"  
        itemValue="idTipoJemplar" itemLabel="tipoEjemplar" />  
</form:select>  
<form:errors path="tipoejemplar.idTipoJemplar" cssClass="text-danger" />
```

Ilustración 4.29. Validación de datos. Mensaje de error.

Una vez introducidas las anotaciones necesarias, hay que configurar el método del controlador que recibe este objeto para que lo valide. El primer paso es añadir la anotación @Validated delante del modelo que se quiere validar. Una vez hecho esto, se puede acceder al objeto BindingResult para consultarlo si ha encontrado algún error.

```

@RequestMapping(value = { "/ejemplar/nuevo/{id}" }, method = RequestMethod.POST)
public ModelAndView ejemplarNuevo(
    @ModelAttribute("ejemplar") @Validated(Ejemplar.createEjemplar.class) Ejemplar ejemplar,
    BindingResult result, @PathVariable Integer id, HttpServletRequest request) throws ParseException {
    ModelAndView mav = new ModelAndView();
    if (result.hasErrors()) {

        List<Tipoejemplar> tiposEjemplar = tipoEjemplarService.findAll();
        List<Ubicacion> ubicaciones = ubicacionService.findAll();
        List<Procedencia> procedencias = procedenciaService.findAll();
        mav.addObject("ubicaciones", ubicaciones);
        mav.addObject("procedencias", procedencias);
        mav.addObject("tiposEjemplar", tiposEjemplar);
        mav.addObject("ejemplar", ejemplar);
        mav.setViewName("/catalogo/ejemplar-nuevo");
        return mav;
    }
}

```

Ilustración 4.30. Validación de datos. El controlador.

En este caso, si se detecta el error con `result.hasErrors()`, se devuelve la misma vista del formulario. Esta vez el formulario tendrá los datos del último envío. Además, el modelo tendrá el campo de error relacionado con el atributo que falta.

Nuevo ejemplar

The screenshot shows a web form titled "Nuevo ejemplar". It contains several input fields and buttons:

- Tipo de ejemplar:** A dropdown menu with "Tipo ejemplar" selected. Below it, the text "campo obligatorio" is displayed in red.
- Nº de ejemplares:** A text input field containing the number "2".
- Fecha de alta:** A date picker showing "21/12/2016".
- Signatura:** A text input field containing "820".
- ISBN:** A text input field containing "84-406-5234-8".
- Ubicación:** A dropdown menu with "ubicacion2" selected.
- Procedencia:** A dropdown menu with "procedencia2" selected.
- Buttons:** Two buttons at the bottom: "Cancelar" (red) and "Guardar" (green).

Ilustración 4.31. Validación de datos. La vista.

4.9. Iteración VII

La última iteración la hemos dedicado a mejorar aspectos varios del sistema como por ejemplo el aspecto visual, la usabilidad, aspectos técnicos, así como elaborar un manual de usuario integrado en la propia aplicación.

[RNF04 – Usabilidad](#)

[RNF05 – Manual de usuario](#)



Ilustración 4.32. Manual de usuario.

Capítulo 5. Aspecto visual

En este capítulo vamos a tratar cuestiones relativas al aspecto visual de la aplicación, framework, herramientas, estilo, etc.

Nuestro objetivo a la hora de diseñar la interfaz gráfica era que nuestra aplicación contara con una interfaz sencilla e intuitiva que consiguiera que el manejo de la misma fuera fácil.

5.1. Herramientas

Para conseguir el aspecto visual obtenido, se han utilizado las siguientes herramientas:

- Bootstrap v3.3.7. Un framework que simplifica el proceso de creación de diseños web. La mayor ventaja es que podemos crear interfaces web que se adapten a los distintos navegadores apoyándonos en un framework potente con numerosos componentes webs que nos ahorran mucho esfuerzo y tiempo.
- CSS. Hojas de estilo para centralizar los aspectos visuales y mantenerlos separados del contenido.
- JavaScript.
- Fontawesome. Una librería que cuenta con una gran variedad de iconos.

5.2. Componentes principales

Para desarrollar la interfaz gráfica de usuario, hemos considerado un diseño minimalista y una paleta de colores homogénea. El diseño lo forman tres componentes principales:

- **Una barra superior.** En ella se muestran las secciones principales de la aplicación, así como la opción de desconectarse.

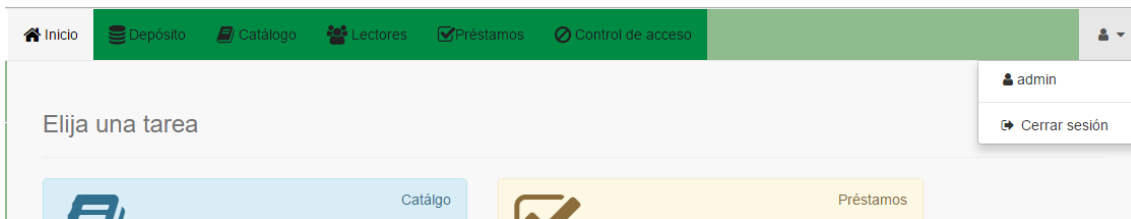


Ilustración 5.1. Barra superior

- **Una barra lateral.** Contiene las funcionalidades de cada sección de la barra superior.

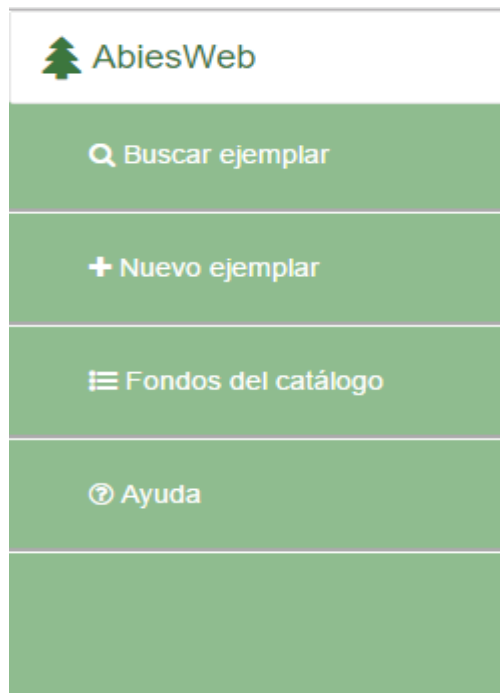


Ilustración 5.2. Barra lateral.

- **Un contenedor central.** En el cual se muestra el contenido de cada funcionalidad. (Ver Ilustración 5.3 y 5.4)

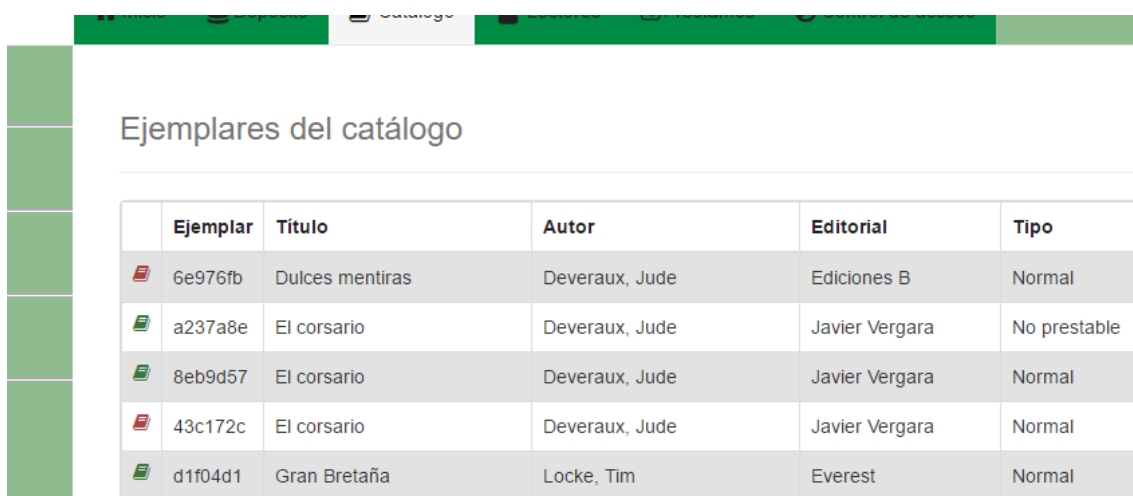


Ilustración 5.3. Contenedor central. Vista de ejemplares del catálogo.

The image shows a web form titled "Nuevo ejemplar". It contains the following fields and controls:

- Tipo de ejemplar:** A dropdown menu with the text "Tipo ejemplar" and a downward arrow.
- Nº de ejemplares:** A text input field containing the text "num".
- Fecha de alta:** A date input field with the placeholder "dd/mm/aaaa" and a calendar icon on the right.
- Signatura:** Three text input fields containing the values "914", "ARN", and "Lon".
- ISBN:** A text input field containing the value "84-7577-368-0".
- Ubicación:** A dropdown menu with the text "Seleccione la ubicación" and a downward arrow.
- Procedencia:** A dropdown menu with the text "Seleccione la procedencia" and a downward arrow.
- Buttons:** Two buttons at the bottom: a red "Cancelar" button and a green "Guardar" button.

Ilustración 5.4. Contenedor central. Vista de nuevo ejemplar.

5.3. Aspectos tenidos en cuenta

A la hora de implementar una interfaz gráfica de usuario, hay muchos aspectos a tener en cuenta. Para nuestro caso, el principio que hemos tenido en mente en todo momento, es presentar al usuario una interfaz fácil de usar que apenas requiera aprendizaje. Lo que en el mundo de UX (*User Experience*) se denomina usabilidad. Para conseguir este objetivo hemos considerado los siguientes detalles:

- Presentar un diseño minimalista sin recargar con información innecesaria.
- Toda la funcionalidad disponible visible en todo momento en las barras superior y lateral.
- Accesos rápidos (shortcuts) en la página de inicio. (Ver Ilustración 5.5)
- Usar iconos descriptivos. (Ver Ilustración 5.1 y 5.2)
- Interacción con el usuario mediante distintos tipos de mensajes. (Ver Ilustración 5.6)

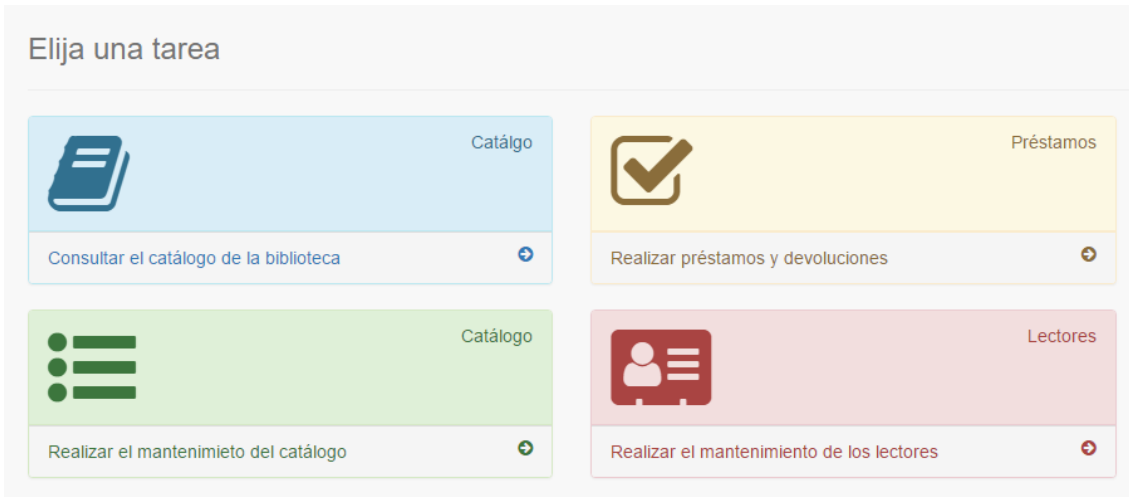


Ilustración 5.5. Accesos directos.

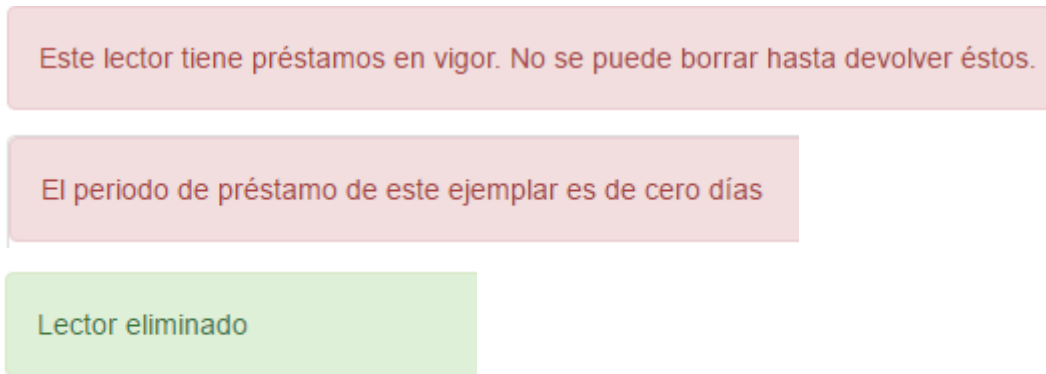


Ilustración 5.6. Mensajes de información.

Capítulo 6. Pruebas

Las pruebas de software son una fase muy importante en el ciclo de desarrollo de software. Tan importante que debería introducirse al inicio del proceso para que los errores no se acumulen en la fase final de la entrega. Por limitaciones de tiempo no se ha podido dedicar el tiempo suficiente a esta parte.

El tipo de pruebas que se han realizado son las pruebas de sistema. Se trata de evaluar el sistema durante el desarrollo para determinar si satisface los requisitos iniciales. En el [Anexo I.2](#), encontraremos algunos casos de pruebas que se han especificado para validar algunas de las funcionalidades.

La validación de estos casos de prueba se ha realizado de forma manual

Capítulo 7. Conclusiones y trabajo futuro

Una vez finalizado el diseño y la implementación de nuestro Trabajo Fin de Grado, recopilamos en este capítulo las conclusiones obtenidas durante el proceso de desarrollo del proyecto.

7.1. Tiempo empleado

En la siguiente tabla mostramos el tiempo empleado para desarrollar el proyecto. Podemos observar que las primeras iteraciones son las que han requerido más tiempo debido a la fase de aprendizaje y búsqueda de información. Una vez adquirida una cierta soltura con las tecnologías y herramientas necesarias, vemos que en las siguientes iteraciones se ha necesitado menos tiempo.

	Tareas realizadas	Número de horas
Iteración 0	Estudio Inicial	12
	Búsqueda de información	100
	Instalación de herramientas	4
	Creación de la base de datos	3
	Creación del repositorio	1
	Estructura del proyecto	
	Configuración del proyecto	30
Iteración I	CU01	40
	CU02	60
	Interfaz gráfica	20
Iteración II	Spring Security	12
	Control de la seguridad	25
Iteración III	CU03	36
Iteración IV	CU04	24
Iteración V	CU05	24

Iteración VI	CU06	36
	Validación de datos	24
Iteración VII	Mejorar aspectos varios	10
	Manual de usuario	8
	Memoria	60
Total horas		529

7.2. Resultado obtenido

Al principio de la memoria detallábamos los objetivos de este Trabajo Fin de Grado, y en ello, especificábamos todos los requisitos que pretendíamos cubrir. Ahora, podemos decir que, en general, hemos cumplido con dichos requisitos.

El resultado es una aplicación web que facilita llevar a cabo la gestión de los fondos documentales de las bibliotecas escolares ofreciendo un entorno intuitivo y de sencillo manejo para la organización de catálogos, préstamos, consulta de los fondos documentales, etc.

7.3. Metodología utilizada

Tal y como se ha mencionado en apartados anteriores, para llevar a cabo el proyecto, se ha intentado seguir una metodología iterativa e incremental, una metodología conocida pero nunca utilizada, por lo que consideramos este trabajo el primer contacto con ella.

Es destacable mencionar, que ha sido todo un reto poner en práctica esta metodología, sobre todo a la hora de empezar, o en el momento de priorizar, o cuando en iteraciones avanzadas se requiere volver a tocar funcionalidades implementadas en iteraciones anteriores.

El resultado final ha sido gratificante, ya que una vez arrancado el proceso y superadas las dos primeras iteraciones, el flujo de trabajo ha avanzado de una manera mucho más fluida, y el resultado de una iteración guiaba en gran medida el contenido de la siguiente.

7.4. Tecnologías utilizadas

Otro de los objetivos a alcanzar durante el desarrollo e implementación de este Trabajo Fin de Grado era conocer el funcionamiento de las tecnologías utilizadas.

Durante el estudio de las tecnologías a utilizar y la fase de implementación, tuvimos el placer de trabajar con Spring, Maven, JSP, JSTL, Bootstrap, CSS, etc. La experiencia con estas tecnologías ha sido, cuanto menos gratificante, ya que, nuestra finalidad personal era dominar su uso y esto, en mayor o menor

medida dependiendo de la tecnología de la que hablemos, lo hemos conseguido.

Puesto que la tecnología del proyecto es Spring Framework, cabe destacar algunas conclusiones:

El punto más fuerte de este framework, desde mi punto de vista, es su modularidad. Esto hace que sea muy fácil de integrar con otras herramientas enfocadas, cada una en un área particular, como pueden ser la capa de la presentación, la persistencia, etc.

Otro aspecto destacable es que brinda muchas facilidades a la hora de escribir código, diseñar sistemas, etc. proporcionando con ello una estructura sólida y consistente.

Como aspecto negativo, la curva de aprendizaje que no es especialmente rápida, ya que se requiere invertir gran cantidad de tiempo para poder comprender y utilizar el framework correctamente.

7.5. Trabajo futuro

Como hemos mencionado anteriormente, el proyecto es una versión inicial de migrar una aplicación de escritorio a una aplicación web, con lo cual, debido al tiempo limitado, hemos escogido las funcionalidades más importantes. A continuación, vamos a listar algunas de las funcionalidades que no se han incluido en esta versión y que podrían ser interesantes de implementar a corto plazo:

- Consultas avanzadas.
- Módulo de estadísticas.
- Módulo de auxiliares.
- Realización de informes.
- Exportación de datos.

Puesto que el lector no es un usuario de la aplicación, sería interesante incluirlo, y con ello implementar las funcionalidades que podría realizar. Por ejemplo:

- Darse de alta.
- Reservar libros.
- Consultar el historial de préstamo.
- Prorrogar préstamo.

Bibliografía

- Spring Boot: <https://projects.spring.io/spring-boot/#quick-start>
- Spring Web MVC: <http://docs.spring.io/spring/docs/current/spring-framework-reference/html/mvc.html>
- Spring Security: <https://spring.io/guides/gs/securing-web/>
- Spring Data JPA: <http://www.baeldung.com/spring-data-jpa-multiple-databases>
- Dependencias Maven: <https://repository.sonatype.org/index.html#welcome>
- JSTL: https://www.tutorialspoint.com/jsp/jsp_standard_tag_library.htm
- Bootstrap: <http://getbootstrap.com/>
- Fontawesome: <http://fontawesome.io/icons/>
- Además de consultas puntuales en otras páginas, manuales y foros de Internet.

Anexo Técnico I

En esta capítulo se recogen todos los anexos técnicos relacionados con apartados anteriores.

I.1. Casos de uso

A continuación, se exponen todos los casos de uso especificados.

I.1.1. CU01 – Gestión de usuarios

CU01.01 Registrar usuario

Título	CU01.01 Registrar usuario
Descripción	Permite dar de alta a un nuevo usuario de la aplicación.
Actor	Administrador.
Pre-condición	El usuario debe estar autenticado y con perfil administrador.
Post-condición	Usuario dado de alta.
Escenario principal	
<ol style="list-style-type: none">1. El usuario pulsa sobre la sección Control de acceso.2. El usuario pulsa sobre la opción Nuevo usuario.3. El sistema muestra un formulario para introducir los datos necesarios.4. El usuario rellena los datos solicitados.5. El usuario pulsa el botón Guardar.6. El sistema redirige a la página de listado de usuarios.	
Escenario alternativo	
<ol style="list-style-type: none">1. El usuario cancela la registración2. El sistema redirige a la página de listado de usuarios	

CU01.02 Consultar usuarios

Título	CU01.02 Consultar usuarios
---------------	----------------------------

Descripción	Permite ver todos los usuarios registrados en la aplicación.
Actor	Administrador.
Pre-condición	El usuario debe estar autenticado y con perfil administrador.
Post-condición	Usuarios consultados.
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario pulsa sobre la sección Control de acceso 2. El sistema muestra un listado de los usuarios que están dados de alta. 	
Escenario alternativo	

CU01.03 Editar usuario

Título	CU01.03 Editar usuario
Descripción	Permite modificar los datos de un usuario.
Actor	Administrador.
Pre-condición	El usuario debe estar autenticado y con perfil administrador.
Post-condición	Datos de usuario modificados.
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario pulsa sobre la sección Control de acceso. 2. El sistema muestra un listado de los usuarios que están dados de alta. 3. El usuario pulsa la opción Editar de un terminado usuario. 4. El sistema muestra un formulario relleno con los datos actuales del usuario. 5. El usuario modifica los datos. 6. El usuario pulsa el botón Guardar. 7. El sistema redirige a la página de listado de usuarios. 	
Escenario alternativo	
<ol style="list-style-type: none"> 1. El usuario cancela la modificación. 2. El sistema redirige a la página de listado de usuarios. 	

CU01.04 Eliminar usuario

Título	CU01.04 Eliminar usuario
Descripción	Permite eliminar usuario dado de alta en la aplicación.
Actor	Administrador.
Pre-condición	El usuario debe estar autenticado y con perfil administrador.
Post-condición	Usuario eliminado.
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario pulsa sobre la sección Control de acceso. 2. El sistema muestra un listado de los usuarios que están dados de alta. 	

<ol style="list-style-type: none"> 3. El actor pulsa el botón Eliminar. 4. El sistema muestra una ventana para confirmar la acción. 5. El usuario pulsa el botón Eliminar. 6. El sistema redirige a la página de listado de usuarios.
Escenario alternativo
<ol style="list-style-type: none"> 1. El usuario cancela la eliminación. 2. El sistema redirige a la página de listado de usuarios.

I.1.2. CU02 – Gestión del depósito

CU02.01 Consultar fondos

Título	CU02.01 Consultar fondos
Descripción	Permite consultar el listado de fondos disponibles en el depósito.
Actor	Administrador, Bibliotecario, Encargado, Invitado.
Pre-condición	
Post-condición	Fondos consultados.
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario pulsa en la sección Depósito. 2. El sistema muestra un listado de los fondos del depósito. 	
Escenario alternativo	

CU02.02 Detalle fondo

Título	CU02.02 Detalle fondo
Descripción	Permite consultar el listado de fondos disponibles en el depósito.
Actor	Administrador, Bibliotecario, Encargado, Invitado.
Pre-condición	
Post-condición	Fondos consultados.
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario pulsa en la sección Depósito. 2. El sistema muestra un listado de los fondos del depósito. 3. El usuario pulsa en la opción Ver. 4. El sistema muestra una ventana con datos principales. 5. El usuario pulsa en la pestaña Datos adicionales. 6. El sistema muestra los datos adicionales del fondo. 	
Escenario alternativo	

CU02.03 Buscar fondo

Título	CU02.03 Buscar fondo
Descripción	Permite hacer una búsqueda en el depósito.
Actor	Administrador, Bibliotecario, Encargado, Invitado.
Pre-condición	
Post-condición	Fondo buscado.
Escenario principal	
<ol style="list-style-type: none">1. El usuario pulsa en la sección Depósito.2. El sistema muestra un listado de los fondos.3. El usuario pulsa sobre la opción Buscar fondo.4. El sistema muestra un formulario para filtrar los fondos.5. El usuario introduce el parámetro del filtro.6. El usuario introduce el valor del filtro y pulsa el botón buscar7. El sistema muestra el listado de los fondos encontrados.	
Escenario alternativo	
<ol style="list-style-type: none">1. El sistema muestra un mensaje indicando que el resultado de la búsqueda es nulo.	

CU03.01 Crear fondo

Título	CU03.01 Crear fondo
Descripción	Permite añadir un nuevo fondo al catálogo.
Actor	Administrador, Bibliotecario.
Pre-condición	El usuario debe estar autenticado con perfil Administrador o Bibliotecario.
Post-condición	Nuevo fondo añadido.
Escenario principal	
<ol style="list-style-type: none">1. El usuario pulsa en la sección Catálogo.2. El usuario pulsa en la opción Fondos del catálogo.3. El sistema muestra el listado de los fondos del catálogo.4. El usuario pulsa en la opción Nuevo fondo.5. El sistema muestra un formulario para buscar el fondo en base de datos externa.6. El usuario introduce el parámetro del filtro.7. El usuario introduce el valor del filtro y pulsa el botón buscar.8. El sistema muestra el listado de los fondos encontrados.9. El usuario pulsa en la opción de Añadir de un determinado fondo.10. El sistema muestra un formulario para introducir los datos necesarios.11. El usuario rellena los datos necesarios.12. El usuario pulsa el botón Guardar13. El sistema redirige a la ventana de listado de fondos del catálogo.	
Escenario alternativo	

1. El usuario cancela la creación.
2. El sistema redirige a la ventana de listado de fondos del catálogo.

I.1.3. CU03 – Gestión del catálogo

CU03.01 Crear fondo

Título	CU03.01 Crear fondo
Descripción	Permite añadir un nuevo fondo al catálogo.
Actor	Administrador, Bibliotecario.
Pre-condición	El usuario debe estar autenticado con perfil Administrador o Bibliotecario.
Post-condición	Nuevo fondo añadido.
Escenario principal	
<ol style="list-style-type: none"> 14. El usuario pulsa en la sección Catálogo. 15. El usuario pulsa en la opción Fondos del catálogo. 16. El sistema muestra el listado de los fondos del catálogo. 17. El usuario pulsa en la opción Nuevo fondo 18. El sistema muestra un formulario para buscar el fondo en base de datos externa. 19. El usuario introduce el parámetro del filtro. 20. El usuario introduce el valor del filtro y pulsa el botón buscar. 21. El sistema muestra el listado de los fondos encontrados. 22. El usuario pulsa en la opción de Añadir de un determinado fondo. 23. El sistema muestra un formulario para introducir los datos necesarios. 24. El usuario rellena los datos necesarios. 25. El usuario pulsa el botón Guardar. 26. El sistema redirige a la ventana de listado de fondos del catálogo. 	
Escenario alternativo	
<ol style="list-style-type: none"> 3. El usuario cancela la creación. 4. El sistema redirige a la ventana de listado de fondos del catálogo. 	

CU03.02 Crear ejemplar

Título	CU03.02 Crear ejemplar
Descripción	Permite añadir ejemplares de un fondo al catálogo.
Actor	Administrador, Bibliotecario.
Pre-condición	El usuario debe estar autenticado con perfil Administrador o Bibliotecario.
Post-condición	Nuevo ejemplar añadido.
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario pulsa en la sección Catálogo. 	

<ol style="list-style-type: none"> 2. El sistema muestra un listado de los ejemplares del catálogo. 3. El usuario pulsa en la opción Nuevo ejemplar. 4. El sistema muestra una ventana con los fondos disponibles. 5. El usuario pulsa en la opción de Añadir ejemplar de un determinado fondo. 6. El sistema muestra un formulario para introducir los datos necesarios. 7. El usuario rellena los datos solicitados. 8. El usuario pulsa el botón Guardar. 9. El sistema redirige a la ventana de listado de ejemplares del catálogo.
Escenario alternativo
<ol style="list-style-type: none"> 1. El usuario cancela la creación. 2. El sistema redirige a la ventana de listado de ejemplares del catálogo.

CU03.03 Consultar ejemplares

Título	CU03.03 Consultar ejemplares
Descripción	Permite ver todos los ejemplares disponibles en el catálogo.
Actor	Administrador, Bibliotecario, Encargado, Invitado.
Pre-condición	
Post-condición	Ejemplares consultados.
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario pulsa en la sección Catálogo. 2. El sistema muestra un listado de todos los ejemplares del catálogo. 	
Escenario alternativo	

CU03.04 Consultar fondos del catálogo

Título	CU03.04 Consultar fondos del catálogo
Descripción	Permite ver todos los fondos disponibles en el catálogo.
Actor	Administrador, Bibliotecario.
Pre-condición	El usuario debe estar autenticado con perfil Administrador o Bibliotecario.
Post-condición	Fondos consultados.
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario pulsa en la sección Catálogo. 2. El usuario pulsa en la opción Fondos del catálogo. 3. El sistema muestra el listado de los fondos del catálogo. 	
Escenario alternativo	

CU03.05 Buscar ejemplar

Título	CU03.05 Buscar ejemplar
Descripción	Permite buscar un ejemplar en el catálogo.
Actor	Administrador, Bibliotecario, Encargado, Invitado.
Pre-condición	
Post-condición	Ejemplar buscado.
Escenario principal	
<ol style="list-style-type: none">1. El usuario pulsa en la sección Catálogo.2. El sistema muestra un listado de los ejemplares del catálogo.3. El usuario pulsa en la opción Buscar ejemplar.4. El sistema muestra un formulario para filtrar los ejemplares.5. El usuario introduce el parámetro del filtro.6. El usuario introduce el valor del filtro y pulsa el botón buscar.7. El sistema muestra el listado de los ejemplares encontrados.	
Escenario alternativo	
<ol style="list-style-type: none">1. El sistema muestra un mensaje indicando que el resultado de la búsqueda es nulo.	

CU03.06 Editar ejemplar

Título	CU03.06 Editar ejemplar
Descripción	Permite modificar los datos de ejemplar.
Actor	Administrador, Bibliotecario.
Pre-condición	El usuario debe estar autenticado con perfil Administrador o Bibliotecario.
Post-condición	Ejemplar modificado.
Escenario principal	
<ol style="list-style-type: none">1. El usuario pulsa en la sección Catálogo.2. El sistema muestra un listado de los ejemplares del catálogo.3. El usuario pulsa en la opción Editar de un determinado ejemplar.4. El sistema muestra un formulario relleno con los datos actuales del ejemplar.5. El usuario modifica el formulario.6. El usuario pulsa el botón Guardar.7. El sistema redirige a la ventana de listado de ejemplares del catálogo.	
Escenario alternativo	
<ol style="list-style-type: none">1. El usuario cancela la modificación.2. El sistema redirige a la ventana de listado de ejemplares del catálogo.	

CU03.07 Eliminar ejemplar

Título	CU03.07 Eliminar ejemplar
---------------	---------------------------

Descripción	Permite eliminar un ejemplar del catálogo
Actor	Administrador.
Pre-condición	El usuario debe estar autenticado con perfil Administrador.
Post-condición	Ejemplar eliminado del catálogo.
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario pulsa en la sección Catálogo. 2. El sistema muestra un listado de los ejemplares del catálogo. 3. El usuario pulsa en la opción Eliminar de un determinado ejemplar. 4. El sistema muestra una ventana para confirmar la acción. 5. El usuario pulsa el botón Eliminar. 6. El sistema redirige a la ventana de listado de ejemplares del catálogo. 	
Escenario alternativo	
<ol style="list-style-type: none"> 1. El sistema muestra un mensaje de error indicando que el ejemplar está prestado. 	

I.1.4. CU04 – Gestión de lectores

CU04.01 Crear lector

Título	CU04.01 Crear lector
Descripción	Permite dar de alta a un nuevo lector en el sistema
Actor	Administrador, Bibliotecario
Pre-condición	El usuario debe estar autenticado con perfil Administrador o Bibliotecario.
Post-condición	Lector creado.
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario pulsa en la sección Lectores. 2. El sistema muestra un listado de los lectores. 3. El usuario pulsa en la opción Nuevo lector. 4. El sistema muestra un formulario para introducir los datos necesarios. 5. El usuario rellena los datos solicitados. 6. El usuario pulsa el botón Guardar. 7. El sistema redirige a la ventana de listado de lectores. 	
Escenario alternativo	
<ol style="list-style-type: none"> 1. El usuario cancela la creación del lector. 	

CU04.02 Consultar lectores

Título	CU04.02 Consultar lectores
Descripción	Permite ver todos los lectores dados de alta en el sistema.
Actor	Administrador, Bibliotecario, Encargado

Pre-condición	El usuario debe estar autenticado con perfil Administrador, Bibliotecario o Encargado.
Post-condición	Lectores consultados.
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario pulsa en la sección Lectores. 2. El sistema muestra el listado de los lectores. 	
Escenario alternativo	

CU04.03 Editar lector

Título	CU04.03 Editar lector
Descripción	Permite modificar datos de lector.
Actor	Administrador, Bibliotecario
Pre-condición	El usuario debe estar autenticado con perfil Administrador o Bibliotecario.
Post-condición	Datos del lector modificados.
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario pulsa en la sección Lectores. 2. El sistema muestra un listado de los lectores. 3. El usuario pulsa en la opción Editar de un determinado lector. 4. El sistema muestra un formulario relleno que los datos actuales del lector. 5. El usuario modifica los datos. 6. El usuario pulsa el botón Guardar. 7. El sistema redirige a la ventana de listado de lectores. 	
Escenario alternativo	
<ol style="list-style-type: none"> 1. El usuario cancela la modificación de datos del lector. 	

CU04.04 Eliminar lector

Título	CU04.04 Eliminar lector
Descripción	Permite eliminar un lector datos de alta en el sistema.
Actor	Administrador.
Pre-condición	El usuario debe estar autenticado con perfil Administrador.
Post-condición	Lector eliminado.
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario pulsa en la sección Lectores. 2. El sistema muestra el listado de los lectores. 3. El usuario pulsa en la opción Eliminar de un determinado usuario. 4. El sistema muestra una ventana para confirmar la acción. 	

5. El usuario pulsa el botón Eliminar.
6. El sistema redirige a la ventana de listado de lectores.
Escenario alternativo
1. El sistema muestra un mensaje de error indicando que el lector tiene un préstamo pendiente.

I.1.5. CU05 – Gestión de políticas de préstamo

CU05.01 Consultar políticas

Título	CU05.01 Consultar políticas
Descripción	Permite consultar las políticas de préstamo.
Actor	Administrador.
Pre-condición	El usuario debe estar autenticado con perfil Administrador.
Post-condición	Política consultada.
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario pulsa en la sección Préstamos. 2. El usuario pulsa en la opción Políticas de préstamo. 3. El sistema redirige a la ventana donde se muestran los tipos de lectores del sistema, los tipos de ejemplares del sistema y las políticas configuradas. 	
Escenario alternativo	

CU05.02 Crear tipo lector

Título	CU05.02 Crear tipo lector
Descripción	Permite crear un nuevo tipo lector.
Actor	Administrador.
Pre-condición	El usuario debe estar autenticado con perfil Administrador.
Post-condición	Tipo lector creado.
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario pulsa en la sección Préstamos. 2. El usuario pulsa en la opción Política de préstamos. 3. El sistema redirige a la ventana donde se muestran los tipos de lectores del sistema, los tipos de ejemplares del sistema y la política configurada. 4. El usuario pulsa el botón de crear nuevo tipo lector. 5. El sistema muestra un formulario para introducir los datos necesarios. 6. El usuario rellena los datos solicitados. 7. El usuario pulsa el botón Guardar. 8. El sistema redirige a la ventana donde se muestran los tipos de lectores del sistema, los 	

tipos de ejemplares del sistema y la política configurada.
Escenario alternativo
1. El usuario cancela la creación de tipo lector

CU05.03 Editar tipo lector

Título	CU05.03 Editar tipo lector
Descripción	Permite modificar los datos del tipo lector.
Actor	Administrador.
Pre-condición	El usuario debe estar autenticado con perfil Administrador.
Post-condición	Tipo lector modificado.
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario pulsa en la sección Préstamos. 2. El usuario pulsa en la opción Política de préstamos. 3. El sistema redirige a la ventana donde se muestran los tipos de lectores del sistema, los tipos de ejemplares del sistema y la política configurada. 4. El usuario pulsa en la opción de Editar de un determinado tipo lector. 5. El sistema muestra un formulario relleno con los datos actuales del tipo lector. 6. El usuario modifica los datos. 7. El usuario pulsa el botón Guardar. 8. El sistema redirige a la ventana donde se muestran los tipos de lectores del sistema, los tipos de ejemplares del sistema y la política configurada. 	
Escenario alternativo	
1. El usuario cancela la modificación de tipo lector.	

CU05.04 Eliminar tipo lector

Título	CU05.04 Eliminar tipo lector
Descripción	Permite eliminar un tipo lector.
Actor	Administrador.
Pre-condición	El usuario debe estar autenticado con perfil Administrador.
Post-condición	Tipo lector eliminado.
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario pulsa en la sección Préstamos. 2. El usuario pulsa en la opción Política de préstamos. 3. El sistema redirige a la ventana donde se muestran los tipos de lectores del sistema, los tipos de ejemplares del sistema y la política configurada. 4. El usuario pulsa en la opción de Eliminar de un determinado tipo lector. 5. El sistema muestra una ventana para confirmar la acción. 6. El usuario pulsa el botón Eliminar. 7. El sistema redirige a la ventana donde se muestran los tipos de lectores del sistema, los tipos de ejemplares del sistema y la política configurada. 	

tipos de ejemplares del sistema y la política configurada.
Escenario alternativo

CU05.05 Crear tipo ejemplar

Título	CU05.05 Crear tipo ejemplar
Descripción	Permite crear un nuevo tipo ejemplar.
Actor	Administrador.
Pre-condición	El usuario debe estar autenticado con perfil Administrador.
Post-condición	Tipo ejemplar creado.
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario pulsa en la sección Préstamos. 2. El usuario pulsa en la opción Política de préstamos. 3. El sistema redirige a la ventana donde se muestran los tipos de lectores del sistema, los tipos de ejemplares del sistema y la política configurada. 4. El usuario pulsa el botón de crear nuevo tipo ejemplar. 5. El sistema muestra un formulario para introducir los datos necesarios. 6. El usuario rellena los datos solicitados. 7. El usuario pulsa el botón Guardar. 8. El sistema redirige a la ventana donde se muestran los tipos de lectores del sistema, los tipos de ejemplares del sistema y la política configurada. 	
Escenario alternativo	
<ol style="list-style-type: none"> 1. El usuario cancela la creación de tipo ejemplar 	

CU05.06 Editar tipo ejemplar

Título	CU05.06 Editar tipo ejemplar
Descripción	Permite modificar los datos del tipo ejemplar.
Actor	Administrador.
Pre-condición	El usuario debe estar autenticado con perfil Administrador.
Post-condición	Tipo ejemplar modificado.
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario pulsa en la sección Préstamos. 2. El usuario pulsa en la opción Política de préstamos. 3. El sistema redirige a la ventana donde se muestran los tipos de lectores del sistema, los tipos de ejemplares del sistema y la política configurada. 4. El usuario pulsa en la opción de Editar de un determinado tipo ejemplar. 5. El sistema muestra un formulario relleno con los datos actuales del tipo lector. 6. El usuario modifica los datos. 7. El usuario pulsa el botón Guardar. 	

8. El sistema redirige a la ventana donde se muestran los tipos de lectores del sistema, los tipos de ejemplares del sistema y la política configurada.
Escenario alternativo
1. El usuario cancela la modificación de tipo ejemplar.

CU05.07 Eliminar tipo ejemplar

Título	CU05.7 Eliminar tipo ejemplar
Descripción	Permite eliminar un tipo ejemplar.
Actor	Administrador.
Pre-condición	El usuario debe estar autenticado con perfil Administrador.
Post-condición	Tipo ejemplar eliminado.
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario pulsa en la sección Préstamos. 2. El usuario pulsa en la opción Política de préstamos. 3. El sistema redirige a la ventana donde se muestran los tipos de lectores del sistema, los tipos de ejemplares del sistema y la política configurada. 4. El usuario pulsa en la opción de Eliminar de un determinado tipo ejemplar. 5. El sistema muestra una ventana para confirmar la acción. 6. El usuario pulsa el botón Eliminar. 7. El sistema redirige a la ventana donde se muestran los tipos de lectores del sistema, los tipos de ejemplares del sistema y la política configurada. 	
Escenario alternativo	
1. El usuario cancela la eliminación de tipo ejemplar	

CU05.08 Editar política

Título	CU05.08 Editar política
Descripción	Permite editar la política de préstamo.
Actor	Administrador.
Pre-condición	El usuario debe estar autenticado con perfil Administrador.
Post-condición	Política modificada.
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario pulsa en la sección Préstamos. 2. El usuario pulsa en la opción Política de préstamos. 3. El sistema redirige a la ventana donde se muestran los tipos de lectores del sistema, los tipos de ejemplares del sistema y la política configurada. 4. El usuario pulsa en la opción Editar de una determinada política. 5. El sistema muestra un formulario relleno con los datos actuales de la política. 6. El usuario modifica los datos de la política. 7. El usuario pulsa el botón Guardar. 	

8. El sistema redirige a la ventana donde se muestran los tipos de lectores del sistema, los tipos de ejemplares del sistema y la política configurada.
Escenario alternativo
1. El usuario cancela la modificación de política.

I.1.6. CU06 – Gestión de préstamos

CU06.01 Prestar ejemplar

Título	CU06.01 Prestar ejemplar
Descripción	Permite realizar el préstamo de un determinado ejemplar a un determinado lector.
Actor	Administrador, Bibliotecario, Encargado
Pre-condición	El usuario debe estar autenticado con perfil Administrador, Bibliotecario o Encargado.
Post-condición	Ejemplar prestado.
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario pulsa en la sección Préstamos. 2. El sistema muestra el listado de los préstamos actuales. 3. El usuario pulsa en la opción Nuevo préstamo. 4. El sistema muestra una ventana para seleccionar el ejemplar. 5. El usuario selecciona el ejemplar. 6. El sistema muestra una ventana para seleccionar el lector. 7. El usuario selecciona el lector. 8. El sistema muestra una ventana con los datos del ejemplar, datos del lector, así como la fecha de devolución calculada y efectiva. 9. El usuario selecciona la fecha efectiva de la devolución. 10. El usuario pulsa el botón guardar. 11. El sistema redirige a la ventana de listado de préstamos. 	
Escenario alternativo I	
1. El sistema muestra mensaje de error indicando que el periodo de préstamo del tipo de ejemplar seleccionado para el tipo de lector seleccionado es de cero días.	
Escenario alternativo II	
1. El sistema muestra mensaje de error indicando que el lector seleccionado ha superado su número máximo de libros prestados.	
Escenario alternativo III	
1. El usuario cancela la realización del préstamo.	

CU06.02 Consultar préstamos

Título	CU06.02 Consultar préstamos
Descripción	Permite ver todos los préstamos realizados.

Actor	Administrador, Bibliotecario, Encargado
Pre-condición	El usuario debe estar autenticado con perfil Administrador, Bibliotecario o Encargado.
Post-condición	Préstamos consultados.
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario pulsa en la sección Préstamos. 2. El sistema muestra el listado de los préstamos actuales. 	
Escenario alternativo	

CU06.03 Prorrogar préstamo

Título	CU06.03 Prorrogar préstamo
Descripción	Permite prorrogar el préstamo de un ejemplar
Actor	Administrador, Bibliotecario, Encargado
Pre-condición	El usuario debe estar autenticado con perfil Administrador, Bibliotecario o Encargado
Post-condición	Préstamo prorrogado.
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario pulsa en la sección Préstamos. 2. El sistema muestra el listado de los préstamos actuales. 3. El usuario pulsa en la opción Prorrogar de un determinado préstamo. 4. El sistema muestra una ventana con los datos del préstamo y la nueva fecha prevista de devolución. 5. El usuario pulsa el botón Guardar. 6. El sistema redirige a la ventana de listado de préstamos. 	
Escenario alternativo	
<ol style="list-style-type: none"> 1. El usuario cancela la prórroga del préstamo. 	

CU06.04 Devolver ejemplar

Título	CU06.04 Devolver ejemplar
Descripción	Permite realizar la devolución de un ejemplar prestado anteriormente.
Actor	Administrador, Bibliotecario, Encargado
Pre-condición	El usuario debe estar autenticado con perfil Administrador, Bibliotecario o Encargado
Post-condición	Ejemplar disponible.
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario pulsa en la sección Préstamos. 	

<ol style="list-style-type: none"> 2. El sistema muestra el listado de los préstamos actuales. 3. El usuario pulsa en la opción Devolver de un determinado préstamo. 4. El sistema muestra una ventana para confirmar la acción. 5. El usuario pulsa el botón Devolver. 6. El sistema redirige a la ventana de listado de préstamos.
Escenario alternativo
<ol style="list-style-type: none"> 1. El usuario cancela la devolución del ejemplar.

I.2. Casos de prueba

En esta sección mostraremos los casos de prueba que hemos diseñado y probado en cada iteración para evaluar el correcto funcionamiento de nuestra aplicación.

Título		CP01.01 – Registrar usuario		
Caso de uso		CU01.01 – Registrar usuario		
Precondición		Usuario autenticado y con perfil Administrador		
ID	Estado	Entrada / acción del usuario	Salida esperada	Resultado
1	Pantalla de creación de nuevo usuario	<ol style="list-style-type: none"> 1. El usuario introduce el usuario. 2. El usuario introduce la contraseña. 3. El usuario introduce el nombre. 4. El usuario introduce los apellidos. 5. El usuario selecciona el perfil. 6. El usuario introduce el correo electrónico. 7. El usuario pulsa el botón guardar. 	<ol style="list-style-type: none"> 8. El sistema guardará el usuario creado y redirigirá el usuario a la página de listado de usuarios. 	OK
2	Pantalla de creación de nuevo usuario	<ol style="list-style-type: none"> 1. El usuario introduce el usuario. 2. El usuario introduce la contraseña. 3. El usuario introduce el nombre. 4. El usuario introduce los apellidos. 5. El usuario introduce el correo electrónico. 6. El usuario pulsa el botón guardar. 	<ol style="list-style-type: none"> 7. El sistema se mantendrá en la misma página indicando que hay que seleccionar el perfil del usuario a registrar. 	OK
3	Pantalla de	<ol style="list-style-type: none"> 1. El usuario pulsa el 	<ol style="list-style-type: none"> 2. El sistema redirigirá 	OK

	creación de nuevo usuario	botón cancelar.	el usuario a la página de listado de usuarios.	
--	---------------------------	-----------------	--	--

Título		CP01.02 – Consultar usuarios		
Caso de uso		CU01.02 – Consultar usuarios		
Precondición		Usuario autenticado y con perfil Administrador		
ID	Estado	Entrada / acción del usuario	Salida esperada	Resultado
1	Pantalla de inicio	1. El usuario pulsa sobre la sección control de acceso.	2. El sistema redirigirá el usuario a la página de listado de usuarios.	OK

Título		CP01.03 – Editar usuario		
Caso de uso		CU01.03 – Editar usuario		
Precondición		Usuario autenticado y con perfil Administrador		
ID	Estado	Entrada / acción del usuario	Salida esperada	Resultado
1	Pantalla de listado de usuarios	1. El usuario localiza el usuario que quiere editar. 2. El usuario pulsa sobre la opción de editar del usuario correspondiente. 4. El usuario cambia el perfil de usuario a “Bibliotecario”. 5. El usuario pulsa el botón guardar.	3. El sistema redirigirá el usuario a la página de editar usuario, mostrando un formulario con los datos actuales del usuario. 6. El sistema guardará los cambios y redirigirá el usuario a la página de listado de usuarios.	OK
2	Pantalla de listado de usuarios	1. El usuario localiza el usuario que quiere editar. 2. El usuario pulsa sobre la opción de editar del usuario correspondiente. 4. El usuario borra el campo de apellidos. 5. El usuario pulsa el botón guardar.	3. El sistema redirigirá el usuario a la página de editar usuario, mostrando un formulario con los datos actuales del usuario. 6. El sistema se mantendrá en la misma página indicando que el campo de apellidos es obligatorio.	OK
3	Pantalla de	1. El usuario localiza el	3. El sistema cargará	OK

	listado de usuarios	<p>usuario que quiere editar.</p> <p>2. El usuario pulsa sobre la opción de editar del usuario correspondiente.</p> <p>4. El usuario pulsa el botón cancelar.</p>	<p>un formulario con los datos actuales del usuario.</p> <p>5. El sistema redirigirá el usuario a la página de listado de usuarios.</p>	
--	---------------------	---	---	--

Título		CP01.04 – Eliminar usuario		
Caso de uso		CU01.04 – Eliminar usuario		
Precondición		Usuario autenticado y con perfil Administrador		
ID	Estado	Entrada / acción del usuario	Salida esperada	Resultado
1	Pantalla de listado de usuarios	<p>1. El usuario localiza el usuario que quiere eliminar.</p> <p>2. El usuario pulsa sobre la opción eliminar del usuario correspondiente</p> <p>4. El usuario pulsa el botón eliminar.</p>	<p>3. El sistema mostrará una ventana modal para que el usuario confirme la acción de eliminar.</p> <p>5. El sistema eliminará el usuario y redirigirá el usuario a la página de listado de usuarios.</p>	OK
2	Pantalla de listado de usuarios	<p>1. El usuario localiza el usuario que quiere eliminar.</p> <p>2. El usuario pulsa sobre la opción editar del usuario correspondiente</p> <p>4. El usuario pulsa el botón cancelar.</p>	<p>3. El sistema mostrará una ventana modal para confirmar la acción de eliminar.</p> <p>5. El sistema redirigirá el usuario a la página de listado de usuarios.</p>	OK

Título		CP02.01 – Consultar fondos		
Caso de uso		CU02.01 – Consultar fondos		
Precondición				
ID	Estado	Entrada / acción del usuario	Salida esperada	Resultado
1	Pantalla de listado de fondos	<p>1. El usuario pulsa sobre la página 2 del listado de fondos.</p>	<p>2. El sistema cargará los fondos de la página 2 en la tabla.</p>	OK
2	Pantalla de listado de	<p>1. El usuario pulsa sobre la opción de página</p>	<p>2. El sistema cargará los fondos de la</p>	OK

	fondos	anterior.	página 1 en la tabla.	
3	Pantalla de listado de fondos	1. El usuario pulsa sobre la opción de última página.	2. El sistema cargará los fondos de la última página en la tabla.	OK

Título		CP02.02 – Detalle fondo		
Caso de uso		CU02.02 – Detalle fondo		
Precondición				
ID	Estado	Entrada / acción del usuario	Salida esperada	Resultado
1	Pantalla de listado de fondos	1. El usuario localiza el fondo que quiere editar 2. El usuario pulsa sobre la opción de ver del fondo correspondiente.	3. El sistema mostrará una ventana modal con los detalles del fondo.	OK
2	Pantalla de listado de fondos	1. El usuario localiza el fondo que quiere editar 2. El usuario pulsa sobre la opción ver del fondo correspondiente. 4. El usuario pulsa sobre la pestaña datos principales.	3. El sistema mostrará una ventana modal con los detalles del fondo. 5. El sistema abrirá la pestaña de datos principales del fondo.	OK
3	Pantalla de listado de fondos	1. El usuario localiza el fondo que quiere editar 2. El usuario pulsa sobre la opción de ver del fondo correspondiente. 4. El usuario pulsa sobre la pestaña datos adicionales.	3. El sistema mostrará una ventana modal con los detalles del fondo. 5. El sistema abrirá la pestaña de datos adicionales del fondo.	OK

Título		CP02.03 – Buscar fondo		
Caso de uso		CU02.03 – Buscar fondo		
Precondición				
ID	Estado	Entrada / acción del usuario	Salida esperada	Resultado
1	Pantalla de búsqueda de fondos	1. El usuario selecciona ISBN como parámetro de búsqueda 2. El usuario introduce “84-239-7159-7”. 3. El usuario pulsa el botón buscar.	4. El sistema encontrará el libro “¡Peligro! playa radiactiva” y lo mostrará en la tabla.	OK

2	Pantalla de búsqueda de fondos	<ol style="list-style-type: none"> 1. El usuario selecciona Título como parámetro de búsqueda. 2. El usuario introduce “Dulces mentiras” 3. El usuario pulsa el botón buscar. 	4. El sistema encontrará el libro y lo mostrará en la tabla.	OK
3	Pantalla de búsqueda de fondos	<ol style="list-style-type: none"> 1. El usuario selecciona ISBN como parámetro de búsqueda. 2. El usuario pulsa el botón buscar 	3. El sistema mostrará una advertencia de que ningún registro cumple los requisitos pedidos.	OK

Título		CP03.01 – Crear fondo		
Caso de uso		CU03.01 – Crear fondo		
Precondición		Usuario autenticado con perfil Administrador o Bibliotecario		
ID	Estado	Entrada / acción del usuario	Salida esperada	Resultado
1	Pantalla de búsqueda de fondos en el depósito	<ol style="list-style-type: none"> 1. El usuario selecciona ISBN como parámetro de búsqueda 2. El usuario introduce “84-239-7159-7”. 3. El usuario pulsa el botón buscar. 5. El usuario selecciona la opción añadir del libro correspondiente. 8. El usuario consulta los datos bibliográficos que el sistema ha cargado del depósito. 9. El usuario pulsa el botón guardar. 	<ol style="list-style-type: none"> 4. El sistema encontrará el libro “¡Peligro! playa radiactiva” y lo muestra en la tabla. 6. El sistema redirigirá el usuario a la pantalla de añadir nuevo fondo al catálogo. 7. El sistema cargará los datos bibliográficos obtenidos del depósito en el formulario. 10. El sistema redirigirá el usuario a la pantalla de listado de fondos del catálogo. 	OK

Título		CP03.02 – Crear ejemplar		
Caso de uso		CU03.02 – Crear ejemplar		
Precondición		Usuario autenticado con perfil Administrador o Bibliotecario		
ID	Estado	Entrada / acción del usuario	Salida esperada	Resultado

1	Pantalla de listado de fondos del catálogo	<ol style="list-style-type: none"> 1. El usuario localiza el fondo del que quiere añadir ejemplares. 2. El usuario pulsa en la opción de añadir del ejemplar correspondiente. 4. El usuario selecciona el tipo de ejemplar "Normal" 5. El usuario introduce el "3" en el número de ejemplares que quiere añadir. 6. El usuario introduce "15/10/2016" en fecha de alta. 7. El usuario selecciona la ubicación. 8. El usuario selecciona la procedencia. 9. El usuario pulsa el botón guardar. 	<ol style="list-style-type: none"> 3. El sistema redirigirá el usuario a la pantalla de nuevo ejemplar. 10. El sistema guardará el ejemplar y redirigirá el usuario a la pantalla listado de ejemplares del catálogo. 	OK OK
2	Pantalla de listado de fondos del catálogo	<ol style="list-style-type: none"> 1. El usuario localiza el fondo del que quiere añadir ejemplares. 2. El usuario pulsa en la opción de añadir del ejemplar correspondiente. 4. El usuario introduce el "3" en el número de ejemplares que quiere añadir. 5. El usuario introduce "15/10/2016" en fecha de alta. 6. El usuario selecciona la ubicación. 7. El usuario selecciona la procedencia. 8. El usuario pulsa el botón guardar. 	<ol style="list-style-type: none"> 3. El sistema redirigirá el usuario a la pantalla de nuevo ejemplar. 9. El sistema se mantendrá en la misma pantalla mostrando un mensaje de advertencia de que el tipo de ejemplar es un campo obligatorio. 	OK

Título	CP03.03 – Consultar ejemplares
Caso de uso	CU03.03 – Consultar ejemplares

Precondición				
ID	Estado	Entrada / acción del usuario	Salida esperada	Resultado
1	Pantalla de inicio	1. El usuario pulsa sobre la sección catálogo	2. El sistema redirigirá el usuario a la página de listado de ejemplares del catálogo, mostrando en color rojo los ejemplares que están prestados y en color verde los ejemplares disponibles.	OK

Título				
CP03.04 – Consultar fondos del catálogo				
Caso de uso				
CU03.04 – Consultar fondos del catálogo				
Precondición				
Usuario autenticado con perfil Administrador o Bibliotecario				
ID	Estado	Entrada / acción del usuario	Salida esperada	Resultado
1	Pantalla de listado de ejemplares del catálogo	1. El usuario pulsa sobre la sección fondos del catálogo.	2. El sistema redirigirá el usuario a la página de listado de fondos del catálogo.	OK

Título				
CP03.05 – Buscar ejemplar				
Caso de uso				
CU03.05 – Buscar ejemplar				
Precondición				
ID	Estado	Entrada / acción del usuario	Salida esperada	Resultado
1	Pantalla de búsqueda de ejemplares	1. El usuario selecciona ISBN como parámetro de búsqueda 2. El usuario introduce “84-239-7159-7”. 3. El usuario pulsa el botón buscar.	4. El sistema encontrará el libro “¡Peligro! playa radiactiva” y lo mostrará en la tabla.	OK
2	Pantalla de búsqueda de ejemplares	1. El usuario selecciona Título como parámetro de búsqueda. 2. El usuario introduce “Dulces mentiras” 3. El usuario pulsa el botón buscar.	4. El sistema encontrará el libro y lo muestra en la tabla.	OK
3	Pantalla de búsqueda de	1. El usuario selecciona ISBN como parámetro	3. El sistema mostrará una advertencia de	OK

	ejemplares	de búsqueda. 2. El usuario pulsa el botón buscar	que ningún registro cumple los requisitos pedidos.	
--	------------	---	--	--

Título		CP03.06 – Editar ejemplar		
Caso de uso		CU03.06 – Editar ejemplar		
Precondición		Usuario autenticado con perfil Administrador o Bibliotecario		
ID	Estado	Entrada / acción del usuario	Salida esperada	Resultado
1	Pantalla de listado de ejemplares del catálogo	<ol style="list-style-type: none"> 1. El usuario localiza el ejemplar que quiere editar. 2. El usuario pulsa sobre la opción editar del ejemplar correspondiente. 4. El usuario cambia el tipo de ejemplar a “No prestable”. 5. El usuario pulsa el botón guardar 	<ol style="list-style-type: none"> 3. El sistema redirigirá el usuario a la página de editar ejemplar. 6. El sistema guardará la modificación de los datos y redirigirá el usuario a la página de listado de ejemplares del catálogo. 	OK
2	Pantalla de listado de ejemplares del catálogo	<ol style="list-style-type: none"> 1. El usuario localiza el ejemplar que quiere editar 2. El usuario pulsa sobre la opción editar del ejemplar correspondiente. 4. El usuario pulsa el botón cancelar. 	<ol style="list-style-type: none"> 3. El sistema redirigirá el usuario a la página de editar ejemplar. 5. El sistema redirigirá el usuario a la página de listado de ejemplares del catálogo. 	OK

Título		CP03.07 – Eliminar ejemplar		
Caso de uso		CU03.07 – Eliminar ejemplar		
Precondición		Usuario autenticado con perfil Administrador		
ID	Estado	Entrada / acción del usuario	Salida esperada	Resultado

1	Pantalla de listado de ejemplares del catálogo	<ol style="list-style-type: none"> 1. El usuario localiza el ejemplar que quiere eliminar. 2. El usuario pulsa sobre la opción eliminar del ejemplar correspondiente. 4. El usuario pulsa el botón eliminar. 	<ol style="list-style-type: none"> 3. El sistema mostrará una ventana modal para que el usuario confirme la acción de eliminar. 5. El sistema eliminará el ejemplar y redirigirá el usuario a la página de listado de ejemplares del catálogo. 	OK
2	Pantalla de listado de ejemplares del catálogo	<ol style="list-style-type: none"> 1. El usuario localiza un ejemplar que está prestado. 2. El usuario pulsa sobre la opción eliminar del ejemplar correspondiente. 4. El usuario pulsa el botón eliminar. 	<ol style="list-style-type: none"> 3. El sistema mostrará una ventana modal para que el usuario confirme la acción de eliminar. 5. El sistema mostrará un mensaje de error indicando de que el ejemplar está prestado. 	OK

Título		CP04.01 – Crear lector		
Caso de uso		CU04.01 – Crear lector		
Precondición		Usuario autenticado con perfil Administrador o Bibliotecario		
ID	Estado	Entrada / acción del usuario	Salida esperada	Resultado
1	Pantalla de nuevo lector	<ol style="list-style-type: none"> 1. El usuario introduce el DNI. 2. El usuario selecciona el tipo de lector. 3. El usuario introduce los apellidos. 4. El usuario introduce el nombre. 5. El usuario introduce la dirección. 6. El usuario introduce la localidad. 7. El usuario introduce la provincia. 8. El usuario introduce el código postal. 	17. El sistema guardará el usuario y redirigirá el usuario a la página de listado de lectores.	OK

		<p>9. El usuario introduce el correo electrónico.</p> <p>10. El usuario introduce el teléfono.</p> <p>11. El usuario introduce el año de nacimiento.</p> <p>12. El usuario selecciona el curso.</p> <p>13. El usuario selecciona el sexo.</p> <p>14. El usuario introduce la fecha de alta.</p> <p>15. El usuario introduce la fecha de validez.</p> <p>16. El usuario pulsa el botón guardar</p>		
2	Pantalla de nuevo lector	1. El usuario pulsa el botón guardar.	2. El sistema se mantendrá en la misma página indicando que los campos DNI, nombre y apellidos son obligatorios.	OK
3	Pantalla de nuevo lector	1. El usuario pulsa el botón cancelar.	2. El sistema redirigirá el usuario a la página de listado de lectores.	OK

Título		CP04.02 – Consultar lectores		
Caso de uso		CU04.02 – Consultar lectores		
Precondición		Usuario autenticado		
ID	Estado	Entrada / acción del usuario	Salida esperada	Resultado
1	Pantalla de inicio	1. El usuario pulsa sobre la sección lectores.	2. El sistema redirigirá el usuario a la página de listado de lectores.	OK

Título		CP04.03 – Editar lector		
Caso de uso		CU04.03 – Editar lector		
Precondición		Usuario autenticado con perfil Administrador o Bibliotecario		
ID	Estado	Entrada / acción del usuario	Salida esperada	Resultado
1	Pantalla de listado de	1. El usuario localiza el lector que quiere editar.	3. El sistema redirigirá el usuario a la	OK

	lectores	<ol style="list-style-type: none"> 2. El usuario pulsa sobre la opción de editar del lector correspondiente. 4. El usuario cambia el tipo de usuario a "Profesor". 5. El usuario pulsa el botón guardar. 	<p>página de editar lector, mostrando un formulario con los datos actuales del lector.</p> <ol style="list-style-type: none"> 6. El sistema guardará los cambios y redirigirá el usuario a la página de listado de lectores. 	
2	Pantalla de listado de lectores	<ol style="list-style-type: none"> 1. El usuario localiza el lector que quiere editar. 2. El usuario pulsa sobre la opción de editar del lector correspondiente. 4. El usuario pulsa el botón cancelar. 	<ol style="list-style-type: none"> 3. El sistema redirigirá el usuario a la página de editar lector, mostrando un formulario con los datos actuales del lector. 5. El sistema redirigirá el usuario a la página de listado de lectores. 	OK

Título		CP04.04 – Eliminar lector		
Caso de uso		CU04.04 – Eliminar lector		
Precondición		Usuario autenticado y con perfil Administrador		
ID	Estado	Entrada / acción del usuario	Salida esperada	Resultado
1	Pantalla de listado de lectores	<ol style="list-style-type: none"> 1. El usuario localiza el lector que quiere eliminar. 2. El usuario pulsa sobre la opción eliminar del lector correspondiente 4. El usuario pulsa el botón eliminar. 	<ol style="list-style-type: none"> 3. El sistema mostrará una ventana modal para que el usuario confirme la acción de eliminar. 5. El sistema eliminará el lector y redirigirá el usuario a la página de listado de lectores. 	OK
2	Pantalla de listado de lectores	<ol style="list-style-type: none"> 1. El usuario localiza el lector que quiere eliminar. 2. El usuario pulsa sobre la opción editar del lector correspondiente 4. El usuario pulsa el botón cancelar. 	<ol style="list-style-type: none"> 3. El sistema mostrará una ventana modal para que el usuario confirme la acción de eliminar. 5. El sistema redirigirá el usuario a la página de listado de lectores. 	OK

Título		CP05.01 – Consultar políticas		
Caso de uso		CU05.01 – Consultar políticas		
Precondición		Usuario autenticado y con perfil Administrador		
ID	Estado	Entrada / acción del usuario	Salida esperada	Resultado
1	Pantalla de listado de préstamos	1. El usuario pulsa sobre la sección políticas de préstamo.	2. El sistema redirigirá el usuario a la página de listado de políticas de préstamo.	OK

Título		CP05.02 – Crear tipo lector		
Caso de uso		CU05.02 – Crear tipo lector		
Precondición		Usuario autenticado y con perfil Administrador		
ID	Estado	Entrada / acción del usuario	Salida esperada	Resultado
1	Pantalla de políticas de préstamo	1. El usuario pulsa el botón nuevo tipo lector. 3. El usuario introduce “Alumno” en tipo lector. 4. El usuario introduce “3” en número de ejemplares. 5. El usuario pulsa el botón guardar.	2. El sistema redirigirá el usuario a la página de crear nuevo tipo lector. 6. El sistema guardará el tipo lector y creará una nueva política con el nuevo tipo lector. 7. El sistema redirigirá el usuario a la página de políticas de préstamo.	OK
2	Pantalla de políticas de préstamo	1. El usuario pulsa el botón nuevo tipo lector. 3. El usuario introduce “3” en número de ejemplares. 4. El usuario pulsa el botón guardar.	2. El sistema redirigirá el usuario a la página de crear nuevo tipo lector. 5. El sistema se mantendrá en la misma página indicando que el campo tipo lector es obligatorio.	OK
3	Pantalla de políticas de préstamo.	1. El usuario pulsa el botón nuevo tipo lector. 3. El usuario pulsa el botón cancelar.	2. El sistema redirigirá el usuario a la página de crear nuevo tipo lector. 4. El sistema redirigirá	OK

			el usuario a la página de políticas de préstamo.	
--	--	--	--	--

Título		CP05.03 – Editar tipo lector		
Caso de uso		CU05.03 – Editar tipo lector		
Precondición		Usuario autenticado y con perfil Administrador		
ID	Estado	Entrada / acción del usuario	Salida esperada	Resultado
1	Pantalla de políticas de préstamo	<ol style="list-style-type: none"> 1. El usuario localiza el tipo lector que quiere editar. 2. El usuario pulsa sobre la opción de editar del tipo lector correspondiente. 4. El usuario cambia en nombre de tipo lector. 5. El usuario pulsa el botón guardar. 	<ol style="list-style-type: none"> 3. El sistema redirigirá el usuario a la página de editar tipo lector. 6. El sistema guardará los cambios y redirigirá el usuario a la página de políticas de préstamo. 	OK
2	Pantalla de políticas de préstamo	<ol style="list-style-type: none"> 1. El usuario localiza el tipo lector que quiere editar. 2. El usuario pulsa sobre la opción de editar del tipo lector correspondiente. 4. El usuario cambia el número de ejemplares. 5. El usuario pulsa el botón guardar. 	<ol style="list-style-type: none"> 3. El sistema redirigirá el usuario a la página de editar tipo lector. 6. El sistema guardará los cambios y redirigirá el usuario a la página de políticas de préstamo. 	OK
3	Pantalla de políticas de préstamo	<ol style="list-style-type: none"> 1. El usuario localiza el tipo lector que quiere editar. 2. El usuario pulsa sobre la opción de editar del tipo lector correspondiente. 4. El usuario borra el campo tipo lector. 5. El usuario pulsa el botón guardar 	<ol style="list-style-type: none"> 3. El sistema redirigirá el usuario a la página de editar tipo lector. 6. El sistema se mantendrá en la misma página indicando que el campo tipo lector es obligatorio. 	OK

Título		CP05.04 – Eliminar tipo lector		
Caso de uso		CU05.04 – Eliminar tipo lector		

Precondición		Usuario autenticado y con perfil Administrador		
ID	Estado	Entrada / acción del usuario	Salida esperada	Resultado
1	Pantalla de políticas de préstamo	<ol style="list-style-type: none"> 1. El usuario localiza el tipo lector que quiere eliminar. 2. El usuario pulsa sobre la opción de eliminar del lector correspondiente. 4. El usuario pulsa el botón eliminar. 	<ol style="list-style-type: none"> 3. El sistema mostrará una ventana modal para que el usuario confirme la acción de eliminar. 5. El sistema eliminará el tipo lector y la política asociada a él y redirigirá el usuario a la página de políticas de préstamo. 	OK
2	Pantalla de políticas de préstamo	<ol style="list-style-type: none"> 1. El usuario localiza el tipo lector que quiere eliminar. 2. El usuario pulsa sobre la opción de eliminar del lector correspondiente. 4. El usuario pulsa el botón cancelar. 	<ol style="list-style-type: none"> 3. El sistema mostrará una ventana modal para que el usuario confirme la acción de eliminar. 5. El sistema redirigirá el usuario a la página de políticas de préstamo. 	OK

Título		CP05.05 – Crear tipo ejemplar		
Caso de uso		CU05.05 – Crear tipo ejemplar		
Precondición		Usuario autenticado y con perfil Administrador		
ID	Estado	Entrada / acción del usuario	Salida esperada	Resultado
1	Pantalla de políticas de préstamo	<ol style="list-style-type: none"> 1. El usuario pulsa el botón nuevo tipo ejemplar. 3. El usuario introduce “No prestable” en tipo ejemplar. 4. El usuario pulsa el botón guardar. 	<ol style="list-style-type: none"> 2. El sistema redirigirá el usuario a la página de crear nuevo tipo ejemplar. 5. El sistema guardará el tipo ejemplar y creará una nueva política con el nuevo tipo ejemplar 6. El sistema redirigirá el usuario a la página de políticas de préstamo. 	OK
2	Pantalla de políticas de préstamo	<ol style="list-style-type: none"> 1. El usuario pulsa el botón nuevo tipo lector. 3. El usuario pulsa el 	<ol style="list-style-type: none"> 2. El sistema redirigirá el usuario a la página de crear 	OK

		botón guardar.	nuevo tipo ejemplar. 4. El sistema se mantendrá en la misma página indicando que el campo tipo ejemplar es obligatorio.	
3	Pantalla de políticas de préstamo.	1. El usuario pulsa el botón nuevo tipo ejemplar. 3. El usuario pulsa el botón cancelar.	2. El sistema redirigirá el usuario a la página de crear nuevo tipo ejemplar. 4. El sistema redirigirá el usuario a la página de políticas de préstamo.	OK

Título		CP05.06 – Editar tipo ejemplar		
Caso de uso		CU05.06 – Editar tipo ejemplar		
Precondición		Usuario autenticado y con perfil Administrador		
ID	Estado	Entrada / acción del usuario	Salida esperada	Resultado
1	Pantalla de políticas de préstamo	1. El usuario localiza el tipo ejemplar que quiere editar. 2. El usuario pulsa sobre la opción de editar del tipo ejemplar correspondiente. 4. El usuario cambia en nombre de tipo ejemplar. 5. El usuario pulsa el botón guardar.	3. El sistema redirigirá el usuario a la página de editar tipo ejemplar. 6. El sistema guardará los cambios y redirigirá el usuario a la página de políticas de préstamo.	OK
2	Pantalla de políticas de préstamo	1. El usuario localiza el tipo ejemplar que quiere editar. 2. El usuario pulsa sobre la opción de editar del tipo ejemplar correspondiente. 4. El usuario borra el campo tipo ejemplar. 5. El usuario pulsa el botón guardar	3. El sistema redirigirá el usuario a la página de editar tipo ejemplar. 6. El sistema se mantendrá en la misma página indicando que el campo tipo ejemplar es obligatorio.	OK

Título		CP05.07 – Eliminar tipo ejemplar		
Caso de uso		CU05.07 – Eliminar tipo ejemplar		
Precondición		Usuario autenticado y con perfil Administrador		
ID	Estado	Entrada / acción del usuario	Salida esperada	Resultado
1	Pantalla de políticas de préstamo	<ol style="list-style-type: none"> 1. El usuario localiza el tipo ejemplar que quiere eliminar. 2. El usuario pulsa sobre la opción de eliminar del tipo ejemplar correspondiente. 4. El usuario pulsa el botón eliminar. 	<ol style="list-style-type: none"> 3. El sistema mostrará una ventana modal para que el usuario confirme la acción de eliminar. 5. El sistema eliminará el tipo ejemplar y la política asociada a él y redirigirá el usuario a la página de políticas de préstamo. 	OK
2	Pantalla de políticas de préstamo	<ol style="list-style-type: none"> 1. El usuario localiza el tipo ejemplar que quiere eliminar. 2. El usuario pulsa sobre la opción de eliminar del tipo ejemplar correspondiente. 4. El usuario pulsa el botón cancelar. 	<ol style="list-style-type: none"> 3. El sistema mostrará una ventana modal para que el usuario confirme la acción de eliminar. 5. El sistema redirigirá el usuario a la página de políticas de préstamo. 	OK

Título		CP05.08 – Editar política		
Caso de uso		CU05.08 – Editar política		
Precondición		Usuario autenticado y con perfil Administrador		
ID	Estado	Entrada / acción del usuario	Salida esperada	Resultado
1	Pantalla de políticas de préstamo	<ol style="list-style-type: none"> 1. El usuario localiza la política que quiere editar. 2. El usuario pulsa sobre la opción de editar de la política correspondiente. 4. El usuario cambia el campo días. 5. El usuario pulsa el botón guardar. 	<ol style="list-style-type: none"> 3. El sistema redirigirá el usuario a la página de editar política. 6. El sistema guarda los cambios y redirigirá el usuario a la página de políticas de préstamo. 	OK
2	Pantalla de políticas de préstamo	<ol style="list-style-type: none"> 1. El usuario localiza la política que quiere editar. 	<ol style="list-style-type: none"> 3. El sistema redirigirá el usuario a la página de editar 	OK

		<p>2. El usuario pulsa sobre la opción de editar de la política correspondiente.</p> <p>4. El usuario pulsa el botón cancelar.</p>	<p>política.</p> <p>5. El sistema redirigirá el usuario a la página de políticas de préstamo.</p>	
--	--	--	---	--

Título		CP06.01 – Prestar ejemplar		
Caso de uso		CU06.01 – Prestar ejemplar		
Precondición		Usuario autenticado		
ID	Estado	Entrada / acción del usuario	Salida esperada	Resultado
1	Pantalla de nuevo préstamo	<p>1. El usuario selecciona un ejemplar.</p> <p>3. El usuario seleccionar un lector.</p> <p>5. El usuario introduce la fecha efectiva de devolución.</p> <p>6. El usuario pulsa el botón guardar.</p>	<p>2. El sistema redirigirá el usuario a la página de seleccionar lector mostrando los datos del lector seleccionado anteriormente.</p> <p>4. El sistema redirigirá el usuario a la página de introducir la fecha efectiva de devolución, mostrando los datos del lector y el ejemplar seleccionados, así como la fecha calculada según la política correspondiente.</p> <p>7. El sistema redirigirá el usuario a la página de listado de préstamos.</p>	OK
2	Pantalla de nuevo préstamo	<p>1. El usuario selecciona un ejemplar de tipo “No prestable”.</p> <p>3. El usuario selecciona un lector de tipo “Alumno”.</p>	<p>2. El sistema redirigirá el usuario a la página de seleccionar lector, mostrando los datos del lector seleccionado anteriormente.</p> <p>4. El sistema se</p>	OK

			mantendrá en la misma página indicando el periodo de préstamo del ejemplar seleccionado para el tipo de lector “Alumno” es de cero días.	
3	Pantalla de nuevo préstamo	<ol style="list-style-type: none"> 1. El usuario selecciona un ejemplar. 3. El usuario selecciona un lector 	<ol style="list-style-type: none"> 2. El sistema redirigirá el usuario a la página de seleccionar lector mostrando los datos del lector seleccionado anteriormente. 4. El sistema se mantendrá en la misma página, indicando que el lector ha superado el número máximo de libros prestados. 	OK

Título		CP06.02 – Consultar préstamos		
Caso de uso		CU06.02 – Consultar préstamos		
Precondición		Usuario autenticado		
ID	Estado	Entrada / acción del usuario	Salida esperada	Resultado
1	Pantalla de inicio	<ol style="list-style-type: none"> 1. El usuario pulsa sobre la sección préstamos. 	<ol style="list-style-type: none"> 2. El sistema redirigirá el usuario a la página de listado de préstamos actuales. 	OK

Título		CP06.03 – Prorrogar préstamo		
Caso de uso		CU06.03 – Prorrogar préstamo		
Precondición		Usuario autenticado		
ID	Estado	Entrada / acción del usuario	Salida esperada	Resultado
1	Pantalla de listado de préstamos	<ol style="list-style-type: none"> 1. El usuario localiza el préstamo que quiere prorrogar. 2. El usuario pulsa sobre la opción de prorrogar 	<ol style="list-style-type: none"> 3. El sistema abrirá una ventana modal mostrando los datos del préstamo y la nueva fecha prevista 	OK

		del préstamo correspondiente. 4. El usuario pulsa el botón guardar.	para la devolución del ejemplar. 5. El sistema aplica los cambios y redirigirá el usuario a la página de listado de préstamos.	
2		1. El usuario localiza el préstamo que quiere prorrogar. 2. El usuario pulsa sobre la opción de prorrogar del préstamo correspondiente. 4. El usuario pulsa el botón cancelar.	3. El sistema abrirá una ventana modal mostrando los datos del préstamo y la nueva fecha prevista para la devolución del ejemplar. 5. El sistema redirigirá el usuario a la página de listado de préstamos.	OK

Título		CP06.03 – Devolver préstamo		
Caso de uso		CU06.03 – Devolver préstamo		
Precondición		Usuario autenticado		
ID	Estado	Entrada / acción del usuario	Salida esperada	Resultado
1	Pantalla de listado de préstamos	1. El usuario localiza el préstamo que quiere devolver. 2. El usuario pulsa sobre la opción de devolver del préstamo correspondiente. 4. El usuario pulsa el botón devolver.	3. El sistema mostrará una ventana modal para que el usuario confirme la acción de devolver. 5. El sistema aplica los cambios y redirigirá el usuario a la página de listado de préstamos.	OK
2		1. El usuario localiza el préstamo que quiere devolver. 2. El usuario pulsa sobre la opción de devolver del préstamo correspondiente. 4. El usuario pulsa el botón cancelar.	3. El sistema mostrará una ventana modal para que el usuario confirme la acción de devolver. 5. El sistema redirigirá el usuario a la página de listado de préstamos.	OK