

PROYECTO 2

ENTORNOS DE PROGRAMACIÓN

INTEGRANTES

Sergio Hernando Barón Rivera (2201885)

Andrés Felipe Muños Aguilar (2210087)

Ricardo Svensson Jaimes Estupiñan (2202007)



Introducción

El presente informe presenta el análisis y la especificación de los requerimientos funcionales y no funcionales, así como las historias de usuario, de la aplicación **EntornosUISBook**, una aplicación web en la que las personas pueden publicar libros, compartirlos o archivarlos. Además de contar con una comunidad para preguntas, opiniones y respuestas acerca del libro ofrecido por la persona.

El backend será desarrollado con **Springboot** y para garantizar la integridad y seguridad de los datos de la API se implementará **Spring Security** con JWT, utilizando Maven para la gestión de dependencias y Tomcat como servidor de aplicaciones.

Se usará **Angular** en el frontend de la aplicación, implementando auth guards para mejorar la seguridad y para el diseño se usará Bootstrap.

Justificación (Pertinencia de la temática)

La elección de tecnologías para el desarrollo de EntornosUISBook, responde directamente a las necesidades de una red social centrada en la interacción literaria. Spring Boot permite construir un backend robusto y escalable, mientras que la implementación de Spring Security con JWT garantiza una autenticación segura y controlada. Angular, por su parte, facilita una experiencia de usuario dinámica y modular, ideal para una interfaz interactiva como la que requiere EntornosUISBook. Bootstrap agiliza el diseño responsivo, asegurando accesibilidad desde distintos dispositivos. En conjunto, estas herramientas son pertinentes para construir una plataforma segura, moderna y centrada en la comunidad lectora.

Mundo del problema

En la actualidad, millones de personas consumen, recomiendan y comentan libros a través de diversas plataformas digitales. Sin embargo, muchas de estas plataformas carecen de un enfoque integral que combine la publicación, el intercambio y el debate alrededor de libros dentro de una misma comunidad especializada. Además, los lectores suelen dispersarse entre redes sociales generales, foros o tiendas virtuales sin encontrar un espacio exclusivo para compartir sus lecturas, descubrir nuevas obras a través de otros usuarios o recibir retroalimentación significativa. Esta falta de centralización limita la experiencia social y el crecimiento del hábito lector colaborativo.

Por otro lado, en un entorno donde la seguridad de los datos y la privacidad del usuario son cada vez más importantes, muchas soluciones existentes no ofrecen mecanismos modernos de autenticación o autorización robusta. Por tanto, se hace necesaria una plataforma web especializada que permita a los usuarios no solo publicar y compartir libros, sino también participar activamente en una comunidad segura que fomente el diálogo literario, el intercambio de ideas y la construcción colectiva del conocimiento lector.

Requerimientos funcionales

Los requerimientos funcionales del proyecto EntornosUISBook son los siguientes:

➤ Gestión de libros

1. Crear libro
2. Actualizar libro
3. Obtener lista de libro
4. Archivar libro

Id	Nombre	Prioridad	Descripción	Precondición	Postcondición
RF-1	Agregar Libro	Alta	Permite a los usuarios registrar un nuevo libro en la plataforma.	El usuario debe estar autenticado.	El libro queda disponible en la base de datos.
RF-2	Actualizar Libro	Media	Permite modificar la información de un libro previamente publicado.	El libro debe existir y el usuario debe ser su creador.	La información del libro es actualizada correctamente.
RF-3	Obtener lista de Libros	Baja	Permite visualizar una lista de todos los libros registrados.	El usuario debe estar autenticado.	Se muestra la lista de libros disponibles.

➤ Gestión de Préstamo de Libros

1. Compartir Libro
2. Archivar Libro
3. Solicitar préstamo del libro
4. Seguimiento del estado de un libro
5. Evitar préstamo simultáneo

RF-5	Compartir Libro	Media	Permite compartir un libro dentro o fuera de la plataforma.	El libro debe existir.	El sistema genera un enlace o realiza la acción de compartir.
RF-6	Archivar Libro	Media	Permite que el usuario guarde un libro en su biblioteca personal.	El usuario debe estar autenticado.	El libro se añade a la biblioteca personal del usuario.
RF-7	Solicitar préstamo del Libro	Alta	Permite a un usuario solicitar el préstamo de un libro disponible.	El libro debe estar disponible para préstamo.	El libro queda marcado como "prestado" al usuario solicitante.
RF-8	Seguimiento del estado de un Libro	Media	Permite visualizar si un libro está disponible, prestado o archivado.	El libro debe existir en el sistema.	El sistema muestra el estado actual del libro.
RF-9	Evitar préstamo simultáneo	Alta	El sistema debe impedir que dos o más usuarios tomen prestado el mismo libro de forma simultánea.	El libro debe estar disponible.	Si el libro ya fue tomado, la solicitud posterior es rechazada.

➤ Autenticación de usuario

1. Registro de usuario
2. Funcionalidad del login
3. Autenticación del usuario con JWT
4. Cierre de sesión

Id	Nombre	Prioridad	Descripción	Precondición	Postcondición
RF-10	Registro de usuario	Alta	Permite a un nuevo usuario crear una cuenta en la plataforma.	El usuario no debe estar autenticado.	El usuario queda registrado y puede iniciar sesión.
RF-11	Funcionalidad del Login	Alta	Permite al usuario autenticarse en la plataforma ingresando sus credenciales.	El usuario debe estar registrado.	El sistema verifica las credenciales y permite el acceso.

RF-12	Autenticación del usuario con JWT	Alta	Genera un token JWT al momento del login para gestionar sesiones seguras.	El login debe ser exitoso.	Se genera y entrega un token JWT válido al usuario.
RF-13	Cierre de sesión	Media	Permite al usuario cerrar sesión, invalidando el token o eliminándolo del cliente.	El usuario debe estar autenticado.	El token es eliminado y el acceso del usuario finaliza.

Requerimientos no funcionales

Id	Nombre	Descripción	Precondiciones	Postcondiciones
RNF-1	Seguridad de acceso	El sistema debe pedir inicio de sesión para usar las funciones principales cómo agregar libros o prestar.	El sistema debe estar funcionando correctamente.	El usuario puede acceder solo si está autenticado.
RNF-2	Encriptación de contraseñas	Las contraseñas deben guardarse cifradas para proteger la información de los usuarios.	El usuario se registra o inicia sesión.	La contraseña queda almacenada de forma segura.
RNF-3	Tiempo de respuesta	El sistema debe responder en menos de 2 segundos al realizar acciones como agregar o editar un libro.	El usuario hace una acción desde la interfaz.	El sistema responde rápidamente sin errores.
RNF-4	Diseño adaptable	La plataforma debe verse bien en	El usuario accede desde cualquier	La interfaz se adapta correctamente al

		celulares, tabletas y computadoras.	dispositivo.	tamaño de pantalla.
RNF-5	Registro de préstamos	Se debe guardar un historial de los libros prestados, fechas y usuarios involucrados.	Al menos un libro debe estar prestado.	El préstamo queda registrado en el sistema.
RFN-1 1	Control de devoluciones	El sistema debe permitir poner fechas límite y restricciones para la devolución de libros.	Un libro ha sido prestado.	Se registra la fecha límite y se notifica si hay retraso.
RFN-2 2	Separación por roles	Los usuarios solo pueden editar o eliminar sus propios libros, no los de otros.	El usuario ha iniciado sesión.	Solo puede ver o modificar sus propios libros.
RFN-6	Recuperación ante fallos.	Si el sistema se cae o falla, debe poder reiniciarse sin perder información importante como préstamos o registros.	El sistema ha tenido un error.	Al reiniciar, los datos importantes siguen disponibles.

Historias de usuario

HU-1 - Registro de usuario

Como usuario nuevo, quiero poder registrarme en la plataforma, para crear una cuenta y usar las funciones del sistema.

Criterios de aceptación:

-El formulario de registro debe tener campos para nombre, correo, contraseña.

-Si algún campo está vacío, se debe mostrar un mensaje de error.

-Si el correo ya está registrado, debe mostrarse un aviso.

-Al registrarse correctamente, el usuario debe ser redirigido al login o estar automáticamente autenticado.

HU-2 - Agregar libro

Como usuario registrado, quiero poder agregar un libro a mi perfil, para llevar un registro de los libros que tengo.

Criterios de aceptación:

-El formulario debe incluir campos como título, autor y estado del libro (disponible o no).

-Al guardar el libro, debe aparecer en la lista de "mis libros".

-El libro debe estar solo visible para su dueño, a menos que esté marcado como "disponible para préstamo".

HU-21 - Editar información de libro

Como usuario, **quiero** editar los datos de mis libros, **para** mantener mi colección actualizada.

Criterios de aceptación:

-Solo el dueño del libro puede editarlo.

-El formulario debe permitir cambiar el título, autor y disponibilidad.

-Los cambios deben guardarse y reflejarse en la lista del usuario.

-Si se deja algún campo requerido vacío, debe mostrarse un mensaje de error.

HU-22 - Ver mis libros

Como usuario, **quiero** ver la lista de libros que he registrado, **para** tener control sobre mi colección.

Criterios de aceptación:

- El sistema debe mostrar todos los libros registrados por el usuario.
- Cada libro debe mostrar título, autor y disponibilidad.
- Debe haber opciones para editar o eliminar los libros.

HU-31 - Archivar libros

Como usuario, **quiero** archivar un libro de mi colección, **para** mantener un registro pero sin que esté activo para préstamos.

Criterios de aceptación:

- El usuario puede marcar un libro como “archivado” desde la lista de libros.
- Los libros archivados deben moverse a una sección aparte (“archivados”).
- Un libro archivado no debe estar visible para otros usuarios.
- El usuario puede desarchivar el libro en cualquier momento.

HU-32 - Marcar libro como disponible o no para préstamo

Como usuario, quiero activar o desactivar la opción de préstamo en un libro, para controlar cuándo quiero prestarlo o no.

Criterios de aceptación:

- Desde la vista del libro, debe haber un botón o interruptor para activar/desactivar disponibilidad para préstamo.
- Cuando un libro está marcado como “no disponible”, no debe aparecer en las búsquedas de préstamo.

-El cambio de estado debe guardarse inmediatamente y mostrarse correctamente en la interfaz.

-No se puede cambiar la disponibilidad de un libro que está actualmente prestado.

HU-33 - Solicitar préstamo de un libro

Como usuario, **quiero** poder pedir prestado un libro de otro usuario, **para** poder leerlo sin tener que comprarlo.

Criterios de aceptación:

- Solo se pueden pedir libros marcados como disponibles.
- Al solicitar el libro, se debe notificar al dueño.
- El estado del libro debe cambiar a "pendiente de aprobación".
- El sistema debe registrar la fecha de solicitud.

HU-4 - Registrar devolución libro

Como usuario, **quiero** registrar que me devolvieron un libro, **para** cerrar el préstamo y liberar el libro.

Criterios de aceptación:

- Solo el dueño puede marcar un libro como devuelto.
- El libro debe cambiar de estado a "disponible".
- El sistema debe registrar la fecha de devolución.
- Si el libro se devuelve después de la fecha límite, debe registrarse como "devolución tardía".

HU-41 - Aceptación de devolución

Como usuario propietario del libro, quiero verificar que haya sido devuelto en buenas condiciones por lo que podrá aprobar o denegar la entrega por malas condiciones o mal tiempo de devolución.

Criterios de aceptación:

- Debe haber un apartado para seleccionar si aprueba la entrega o no.
- En caso de no aprobar la entrega, se notifica al otro usuario.

HU-5 - Ver historial de préstamos

Como usuario, **quiero** ver el historial de préstamos que he hecho o recibido, **para** llevar control de mis libros.

Criterios de aceptación:

- El historial debe mostrar fechas de préstamo, devolución y usuarios involucrados.
- Debe incluir préstamos activos y antiguos.
- Los registros deben estar ordenados por fecha o estado.

DIAGRAMA DE LA BASE DE DATOS

Para el desarrollo del presente proyecto, se ha diseñado e implementado una base de datos relacional utilizando el sistema de gestión de bases de datos PostgreSQL. Esta elección se debe a las múltiples ventajas que ofrece esta tecnología en términos de integridad referencial, soporte a operaciones transaccionales, extensibilidad y manejo eficiente de grandes volúmenes de datos. PostgreSQL, al ser un sistema de código abierto y altamente robusto, proporciona además un entorno estable y seguro para el desarrollo de aplicaciones web modernas como la red social de libros que aquí se propone.

La base de datos modela los elementos esenciales de una plataforma social enfocada en la lectura, permitiendo a los usuarios compartir libros, dejar reseñas, gestionar su historial de transacciones de préstamo, y participar activamente en una comunidad de lectores. En el diagrama a continuación se representan las tablas, sus respectivas columnas y las relaciones entre ellas, empleando la notación estándar donde "1" indica una relación de uno y "M" una relación de muchos.

Estructura y Componentes de la Base de Datos

La base de datos está compuesta por las siguientes tablas principales:

- **book**: almacena la información básica de los libros compartidos en la plataforma. Sus columnas incluyen `id_book`, `title`, `author_name`, `isbn`, `synopsis`, `book_cover`, `archived` (estado del libro), y `shareable` (indicador de disponibilidad para compartir).
- **feedback**: contiene las reseñas o comentarios de los usuarios sobre los libros. Cada reseña está asociada a un libro específico mediante la columna `id_book`, y posee campos como `id_feedback`, `note` (calificación) y `comment`.
- **book_transaction_history**: registra el historial de préstamos y devoluciones de libros. Sus campos son `id_book_history`, `user_id`, `book_id`, `returned` (estado de devolución) y `return_approved` (aprobación del retorno del libro).
- **users**: representa a los usuarios registrados en la plataforma. Las columnas de esta tabla incluyen `id_user`, `first_name`, `last_name`, `date_of_birth`, `email`, `password`, `account_locked`, `enabled`, `created_date` y `last_modified_date`.
- **tokens**: se utiliza para la autenticación de usuarios mediante tokens de acceso. Incluye campos como `id_token`, `user_id`, `token`, `created_at`, `expires_at` y `validated_at`.
- **roles**: define los distintos tipos de roles que puede tener un usuario (por ejemplo, administrador, lector, moderador). Sus columnas son `id_role`, `name`, `created_date` y `last_modified_date`.
- **users_roles**: es una tabla intermedia que permite establecer relaciones de muchos a muchos entre `users` y `roles`. Contiene las columnas `user_id` y `role_id`.

Relaciones entre Tablas

Las relaciones entre tablas han sido cuidadosamente definidas para mantener la integridad de los datos y reflejar las reglas de negocio del sistema. Por ejemplo:

- Un **libro** (`book`) puede tener múltiples **reseñas** (`feedback`), pero cada reseña está asociada a un único libro (relación uno a muchos).
- Un **usuario** (`users`) puede haber realizado múltiples **transacciones** (`book_transaction_history`) con múltiples libros (Relación muchos a muchos) y puede tener múltiples **tokens** activos o expirados (Relación uno a muchos).
- La relación entre **usuarios** y **roles** es de muchos a muchos, implementada a través de la tabla intermedia `users_roles`.
- La tabla `book_transaction_history` también relaciona a los libros con los usuarios, representando un modelo de préstamo.

