



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**TFG GII 23.23 Web de libros
y sistema de clasificación
Documentación Técnica**



Presentado por Daniel Fernández Fernández
en Universidad de Burgos
a 11 de junio de 2024
Tutora: Ana Serrano Mamolar

Índice general

Índice general	i
Índice de figuras	iii
Índice de tablas	v
Apéndice A Plan de Proyecto Software	1
A.1. Introducción	1
A.2. Planificación temporal	2
A.3. Estudio de viabilidad	10
Apéndice B Especificación de Requisitos	15
B.1. Introducción	15
B.2. Objetivos generales	15
B.3. Catálogo de requisitos	16
B.4. Especificación de requisitos	20
Apéndice C Especificación de diseño	42
C.1. Introducción	42
C.2. Diseño del modelo datos	42
C.3. Diseño procedimental	44
C.4. Diseño de interfaces	48
C.5. Diseño de la arquitectura	51
Apéndice D Documentación técnica de programación	56
D.1. Introducción	56
D.2. Estructura de directorios	56

<i>Índice general</i>	II
D.3. Manual del programador	59
D.4. Pruebas del sistema	66
Apéndice E Documentación de uso	68
E.1. Introducción	68
E.2. Requisitos de uso	68
E.3. Instalación	68
E.4. Manual del usuario/a sin cuenta	69
E.5. Manual del usuario/a con cuenta y permisos de administración	74
Apéndice F Anexo de sostenibilización curricular	89
F.1. Introducción	89
F.2. Objetivo número 4: Educación de calidad	89
F.3. Objetivo número 5: Igualdad de Género	90
F.4. Objetivo número 13: Acción por el clima	91
F.5. Conclusión	91
Bibliografía	92

Índice de figuras

A.1. Inicio de sesión	4
A.2. Gráfico de horas de sprints	10
B.1. Diagrama conexión de funciones con autenticación	21
B.2. Diagrama de casos de uso	22
C.1. Diagrama Relacional	44
C.2. Diagrama de flujo Web scraping <i>Frontend</i>	45
C.3. Diagrama de flujo Web scraping <i>Backend</i>	46
C.4. Diagrama de flujo permisos <i>Frontend</i>	47
C.5. Diagrama de flujo permisos <i>Backend</i>	48
C.6. Prototipo catálogo	49
C.7. Prototipo catálogo (Vista de administrador)	49
C.8. Prototipo Más Información	50
C.9. Prototipo Login	50
C.10. Módulos <i>backend</i>	51
C.11. Funciones gestionLibros	52
C.12. Patrón repositorio	53
C.13. ComponentesAngular	53
C.14. Servicio de libros	54
D.1. Página de instalación de Visual Studio Code	59
D.2. Extensiones instaladas Visual Studio Code	61
D.3. Instalador de Python	61
D.4. Instalación existosa de NodeJS	62
D.5. Apertura de terminal Git	63
D.6. Comandos de Usuario y correo de Git	64
D.7. Clonación del repositorio	64

Índice de figuras IV

D.8. Activar entorno virtual	65
E.1. Búsqueda de libro en catálogo	70
E.2. Más información de un libro	71
E.3. Formulario de sugerencia de libro	72
E.4. Resultado de valoración	73
E.5. Guardar valoración	73
E.6. Manual de Inicio de sesión	74
E.7. Cambio de botón Inicio de sesión	75
E.8. Estadísticas de administración	76
E.9. Operaciones en gestión de usuarios/as	77
E.10.Operación agregar rol en gestión de roles	78
E.11.Permisos de un rol	79
E.12.Panel de gestión de colaboradores	80
E.13.Agregar libro	81
E.14.Modal para agregar automáticamente	82
E.15.Selección de fuente de web scraping	83
E.16.Agregar libro combinando fuentes	84
E.17.Pantalla gestión del catálogo	85
E.18.Gestión de actividades	87
E.19.Listado de valoraciones guardadas	88

Índice de tablas

A.1. Resumen de licencias de las bibliotecas utilizadas en el proyecto	13
A.2. Resumen de licencias utilizadas	14
B.1. CU-1 Gestión de catálogo.	24
B.2. CU-2 Agregar libro manual.	25
B.3. CU-3 Agregar libro automático.	26
B.4. CU-4 Gestión de roles.	27
B.5. CU-5 Gestión de usuarios.	28
B.6. CU-5.5 Cambiar contraseña.	29
B.7. CU-6 Importar catálogo.	30
B.8. CU-7 Exportar catálogo.	31
B.9. CU-8 Gestionar actividades.	32
B.10.CU-8.5 Obtener actividades.	33
B.11.CU-9 Iniciar sesión.	34
B.12.CU-10 Buscar libro.	35
B.13.CU-11 Obtener catálogo.	36
B.14.CU-12 Obtener información de libro.	37
B.15.CU-13 Gestionar valoraciones.	38
B.16.CU-14 Obtener estadísticas.	39
B.17.CU-15 Generar valoraciones.	40
B.18.CU-16 Crear sugerencia.	41

Apéndice A

Plan de Proyecto Software

A.1. Introducción

Para todo proyecto que se va a producir, es fundamental realizar un estudio acerca de cuales son los los objetivos que se quieren alcanzar, una planificación temporal y los recursos disponibles. En este apartado se va a tratar sobre este establecimiento de metas y la temporalización de las mismas utilizando dos apartados principales:

- Planificación temporal
- Estudio de viabilidad
 - Viabilidad legal
 - Viabilidad económica

Debido a que se han utilizado metodologías ágiles durante el desarrollo de este proyecto, desde un inicio ha quedado muy claro cómo se gestionaban las tareas a realizar para poder ir generando un aporte de valor continuo tal como se muestra en la sección de planificación temporal.

Por otra parte, en el estudio de viabilidad se van a comentar tanto los aspectos económicos que conciernen al buen desarrollo del proyecto como el marco legal para tener garantizado que en un futuro no aparezcan inconvenientes legales

A.2. Planificación temporal

La planificación temporal del proyecto se ha estructurado en *sprints* de aproximadamente dos semanas. Esta metodología ágil permite dividir el trabajo en ciclos cortos y manejables, facilitando la adaptación a los cambios y la mejora continua del desarrollo. Cada sprint incluye la planificación, ejecución y revisión de un conjunto de tareas específicas, asegurando que el proyecto avance de manera constante y controlada.

Primer *sprint* 29-11-2023 / 20-12-2023

Antes de la selección de herramientas y tecnologías, se llevaron a cabo una serie de reuniones con el cliente para capturar y entender los requisitos del proyecto. Estas reuniones son cruciales para asegurar que todas las necesidades y expectativas del cliente se consideren desde el principio. Como primeros pasos para la creación del proyecto, se realiza un estudio inicial para considerar cuales van a ser los lenguajes de programación utilizados durante el proyecto. Debido a mi poca experiencia diseño web, opto por utilizar el *framework* de python [Flask](#) ya que de una manera sencilla me permite crear páginas web en un lenguaje de programación ya trabajado durante la carrera y tener una integración sencilla entre *backend* y *frontend*. Además de esta elección, en esta tarea se realizan labores de investigación y lectura de documentación para iniciar el proyecto con las habilidades suficientes para empezar a dar pequeños pasos en el diseño web. Algún ejemplo de la documentación utilizada es la página oficial de Flask [?] y la documentación de Bootstrap5 [?].

Como otra tarea a realizar durante este sprint, se determina como idea inicial obtener los datos de los libros a través de la realización de peticiones a una API . Esto permitía una manera simple de obtener unos datos alojados en internet y poder generar un filtro para obtener el resultado esperado. Tal como se menciona en la memoria, se inclina la balanza a favor de la API de Google, por lo que inicio la lectura de documentación asociada para informarme sobre cómo realizar las llamadas a este servicio.

Mientras se realizan esas tareas se toman apuntes para incluir la información de la API en la memoria.

Tras definir las tareas a realizar en este *sprint*, se estimó que la duración de la realización de las tareas son de 10 horas, pero finalmente las tareas se completan en 7 horas.

Segundo *sprint* 20-12-2023 / 16-01-2024

Tras tomar las decisiones iniciales del anterior *sprint*, comienzo a investigar la estructura del Framework y los lenguajes HTML y CSS realizando pequeños ejercicios, ya que mi experiencia anterior con estas herramientas era muy limitada.

Una vez realizada esa toma de contacto creo una ejemplo inicial de como podría estructurarse la web para debatirlo en la reunión de fin de *sprint*. Para este inicialmente creo un catálogo básico con una base de datos que permite realizar estas operaciones de manera manual:

- Barra de navegación que permite cambiar de página.
- Sistema de Login para gestionar la aparición de botones ocultos.
- Consultar todos los libros existentes en la base datos.
- Realizar búsquedas de los libros existentes en la base de datos en base al título o al ISBN.
- Agregar un libro a la base de datos.
- Editar un libro de la base de datos.
- Eliminar un libro de la base de datos.

El sistema de Login se estableció para no permitir que ningún usuario que no tuviese credenciales pudiera realizar modificaciones de la base de datos. En el caso de que ejecutaran la URL exacta de la función, la protección de seguridad les llevaría a la pantalla de Inicio de Sesión.

Para poder tener más libertad de elementos para integrar dentro de la web, se decide utilizar la biblioteca Bootstrap 5, la cual tiene elementos predefinidos con estilos, generando una programación más eficiente y limpia.

Mientras continuo realizando apuntes para incluirlos en la memoria posteriormente.

Realizar estas tareas tuvo una estimación inicial de 45 horas, pero finalmente fueron 40 horas.

El formulario de inicio de sesión es un cuadro rectangular con un efecto de sombra. En el encabezado dice "Iniciar Sesión". La sección "Usuario" contiene un campo de texto para "Nombre de usuario". La sección "Contraseña" contiene un campo de texto para "Contraseña". Abajo de los campos hay un botón azul con el texto "Iniciar Sesión".

Figura A.1: Inicio de sesión

Tercer *sprint* 16-01-2024 / 30-01-2024

Una vez presentado y debatido este primer prototipo de página web en el *sprint*, se decide continuar de esta base agregando más elementos para añadirle funcionalidades. De manera adicional, se comienza a realizar un prototipo web para ir documentando el esquema de secciones existente en este proyecto. Tal como se comenta en la sección de Técnicas y Herramientas, el prototipo web se realiza con el programa Justinmind.

Además, se agrega otra funcionalidad considerada como esencial, una página que al seleccionar el libro te muestre toda la información disponible en la base de datos para todo usuario interesado en obtener datos. Para ello, en cada libro mostrado en el catálogo se pone a disposición un botón que te redirige a esta página.

Durante esa reunión se propone la idea de incluir un apartado específico para los responsables de la administración que permite importar y exportar en un archivo CSV todos los datos de cada libro contenidos en la base de datos. Esta idea se concibe principalmente por dos grandes motivos

- Tener la oportunidad de volcar datos más rápido desde el archivo CSV y posteriormente cargar el contenido del fichero en la base de datos.
- Poder tener un duplicado de la base de datos por si el contenido de la misma quedase inaccesible o se realizara un cambio indeseado, lo cuál se solucionaría restableciendo el catálogo con el archivo de seguridad descargado.

Realizar estas tareas tuvo una estimación inicial de 40 horas, pero finalmente fueron 36 horas.

Cuarto *sprint* 30-1-2024 / 13-02-2024

En este *sprint* se corrigieron un error relativo a la implementación de la importación y la exportación. El error consistía en que un administrador pudiera colocar el signo "," para separar dos elementos como por ejemplo dos ISBN o que se encontrase en el contenido de la descripción del libro, lo cual generaba errores a la hora de transformar los campos a columnas al tener ese mismo signo como delimitador. Por ese motivo se propuso la idea de cambiar el delimitador por defecto a el signo ";".

Tras realizar estas correcciones de errores iniciamos el proceso de obtener la información de los libros de forma automática. Aunque inicialmente se valoró la opción de utilizar únicamente la API de Google Books, se consideró interesante para dar más fuentes de información y aumentar la dificultad técnica del proyecto la utilización de Web scraping. El web scraping realizado se aplica sobre 2 páginas web las cuales tienen un catálogo de libros muy amplio. Estas webs son [Amazon](#) y [Agapea](#). Estos tres proveedores nos permiten que en base a un ISBN o un Título, internamente se lancen unos procesos que busquen en los proveedores con las diferentes técnicas mostradas para que se pueda seleccionar el proveedor que nos de la información más precisa. En el caso de que se quisiera agregar un libro automáticamente pero se desee personalizar y usar una mezcla de los 3 proveedores, se dispone de un sistema de desplegables que nos permite realizar esta acción.

Durante el desarrollo en este espacio de tiempo, se considera que sería necesario implementar un sistema que nos permitiese proteger el catálogo de una importación que pueda encontrarse corrupta o con datos erróneos. Por lo que antes de realizar la importación, aparece una ventana modal que recuerda al usuario que debería de exportar el catálogo actual por seguridad.

Para este *sprint* se estimo inicialmente unas 30 horas pero finalmente se realizó en 35.

Quinto *sprint* 13-02-2024 / 27-02-2024

Tras hacer las pruebas pertinentes de que la técnica de Web scraping es satisfactoria en y una gran mayoría de casos devuelve el libro correcto, (Los casos en los que falla es debido a la inexistencia de ese título en el catálogo de la plataforma en la que se busca), se pone como tareas en este

sprint integrarlo a la página web en un apartado específico de búsqueda automática y acompañarlo de la llamada a la API de Google books. Para indicar al usuario que se está ejecutando la búsqueda de libros, se añade una animación de carga.

Finalmente se pone como tarea el tratamiento de los datos obtenidos de esas fuentes, indicando con botones cuál de los resultados de las fuentes es el deseado para agregar al catálogo. En el caso de que se quisiera agregar un libro automáticamente pero se desee personalizar y usar una mezcla de los 3 proveedores, se dispone de un sistema de desplegables que nos permite realizar esta acción.

Durante el desarrollo en este espacio de tiempo, se considera que sería necesario implementar un sistema que nos permitiese proteger el catálogo de una importación que pueda encontrarse corrupta o con datos erróneos. Por lo que antes de realizar la importación, aparece una ventana modal que recuerda al usuario que debería de exportar el catálogo actual por seguridad.

Este *sprint* recibió una estimación inicial de 30 horas pero se completaron todas las tareas en 20 horas.

Sexto *sprint* 27-02-2024 / 2-04-2024

En la reunión de inicio de este *sprint* se presentó la idea acerca de la posibilidad de cambiar la parte de *frontend* hasta ahora implementada utilizando Flask a un framework más potente llamado Angular. Esto se debe a que durante mis prácticas curriculares obtuve conocimientos de este framework, que además de utilizar HTML y CSS utiliza TypeScript para el desarrollo de las aplicaciones. Tras hablarlo decidimos que este *sprint* se dedicaría a investigar la viabilidad de una refactorización completa del *frontend* a Angular mientras se realizaban desarrollos en la aplicación ya creada. De manera adicional, para poder realizar un intercambio de información óptima el *backend* de Flask tendría que modificarse para convertirse en una API. Por lo que se realizó este estudio también.

Como desarrollo en paralelo en la aplicación actual se propuso generar un formulario de ejemplo acerca de la predicción de valoración de un libro.

El tiempo estimado inicial es de 20 horas pero las tareas se completan finalmente en 30 horas.

Séptimo *sprint* 13-03-2024 / 2-04-2024

Durante este *sprint* se definieron las tareas a refactorizar al nuevo framework, después de un resultado positivo en la investigación de viabilidad, por lo que numerosas pantallas generadas en flask se refactorizaron y modificaron para adaptarla al nuevo framework. Como tarea adicional, se definió nuevas pantallas que permitían tener un entorno de usuario más limpio y clasificado para todos los usuarios de la web. Un ejemplo de esto es la creación de un panel de administración. Para poder agregar aún más elementos y variedad a la web, se incluyeron las siguientes bibliotecas que brindan un catálogo de elementos extenso:

- PrimeNG
- Sweetalert2

Por la parte del *backend*, se modificaron todos los métodos para poder utilizarse como API y trabajar con archivos de formato JSON. Esto permite tener separado el *frontend* y el *backend*.

Estas tareas tuvieron una estimación de 80 horas y se completaron en 100 horas

Octavo *sprint* 2-04-2024 / 16 -4-2024

Durante este *sprint* se completaron de refactorizar los apartados no contemplados en el *sprint* anterior y se decidió mejorar la personalización de la página web para que los responsables de la administración de la misma tuvieran el mayor control posible y aumentar la complejidad técnica de la web. Algunas de estas decisiones fueron implementar un sistema de usuarios personalizados con un sistema de roles totalmente gráficos, y cada uno de esos roles contiene una serie de permisos personalizables y ajustables, limitando de manera dinámica a los usuarios con cuenta ciertos parámetros del panel de administrador.

Otra tarea perteneciente a este *sprint* consiste en aplicar esa misma filosofía de personalización, permitiendo tanto guardar las estimaciones realizadas por los usuarios como modificar ciertos parámetros de elección en ese apartado de estimaciones.

Finalmente, por agregar funcionalidades distintas se creó una tarea de creación de un apartado de sugerencias las cuales mandarían el contenido

de esa sugerencia automáticamente a una cuenta de correo expresamente creada para ese fin.

Este *sprint* ha tenido una estimación inicial de 50 horas y se dedicaron 70.

Noveno *sprint* 16-04-2024 / 30-04-2024

En este *sprint* se propone ir cerrando todas las tareas de desarrollo restantes para poder desplegar la aplicación completa y comenzar con el testing. Se han definido tareas como un apartado de estadísticas para dar una visión a los usuarios con cuenta de las estadísticas actuales de los libros y de otros parámetros. Estas tareas engloban tanto el desarrollo del *backend* como la implementación de los resultados en el *frontend*.

De manera adicional, se van corrigiendo errores de la página para que sea completamente adaptativa al tamaño de la pantalla, así como errores de importación y exportación de CSV y la inclusión del formato Excel.

Finalmente, también se crean unos ejemplos de como sería el formato de las páginas pendientes de llenar.

Este *sprint* ha tenido una estimación inicial de 20 horas, y se finalizó en un tiempo de 15 horas.

Décimo *sprint* 30-04-2024 / 14-05-2024

En este *sprint* se realiza el despliegue tanto en netlify como en render para front y back respectivamente, con esto ya se pueden realizar pruebas más estrictas para terminar de perfilar los detalles y comenzar a localizar errores.

Además de eso, se comienza a separar todos los archivos del *backend* en módulos para facilitar la independencia, mantenibilidad y escalabilidad.

De manera adicional, se van corrigiendo errores del apartado responsive para que se pueda ver la web correctamente en dispositivos móviles y se integra JWT para evitar llamadas a la API de usuarios no autorizados.

Este *sprint* ha tenido un estimación inicial de 50 horas y finalmente se dedicaron 60 debido a problemas en el despliegue.

Undécimo *sprint* 14-05-2024 / 28-04-2024

Este *sprint* se dedica íntegramente a realizar la memoria e ir localizando errores que se van localizando. Un error muy significativo encontrado al desplegar ha sido la persistencia de los datos en la base de datos, por lo que se decidió pagar la suscripción mínima para poder tener esa persistencia y poder realizar pruebas con base de datos.

Otro de los errores importantes tratados durante este *sprint* ha sido el apartado de estadísticas, el cual si se importaban libros fallaba la generación de estadísticas y no se tenían en cuenta los libros ya existentes en el catálogo antes de realizar la importación.

La estimación de este apartado ha sido de 70 horas pero se pudo completar en 50 horas.

Duodécimo *sprint* 28-04-2024 / 4-04-2024

Durante la realización de este *sprint*, se realizó una reunión para evaluar la viabilidad de realizar nuevos desarrollos y cambios menores, los cuales se revisaron, se adaptaron y se aceptaron.

Uno de esos cambios es la realización de un nuevo apartado en la web que describa la intencionalidad de la página web y muestre una tabla con todos los colaboradores que han realizado aportaciones verídicas e interesantes en el apartado de valoración, los cuales sirven de guía para los responsables de la administración para poder generar una valoración más profunda.

Por otra parte, se continúa con el desarrollo de la memoria y los anexos y se sigue continuando con las pruebas de la página web, intentando localizar más errores o cambios mínimos como el idioma del calendario del selector de rango de las estadísticas.

Este *sprint* tiene una valoración temporal de 30 horas pero se han dedicando 45 horas.

Resumen de horas estimadas y dedicadas

Se puede observar en la gráfica inferior un resumen visual de las horas estimadas junto con las horas realizadas. Si se analiza este gráfico, se puede concluir que las horas estimadas y las horas dedicadas a la realización del proyecto siempre han tenido una relación aproximada (sin grandes variaciones), por lo que aún apareciendo problemas y soluciones más eficientes

a lo largo de los *sprints*, todos han tenido una estimación relativamente cercana al esfuerzo real que se ha llevado a cabo.

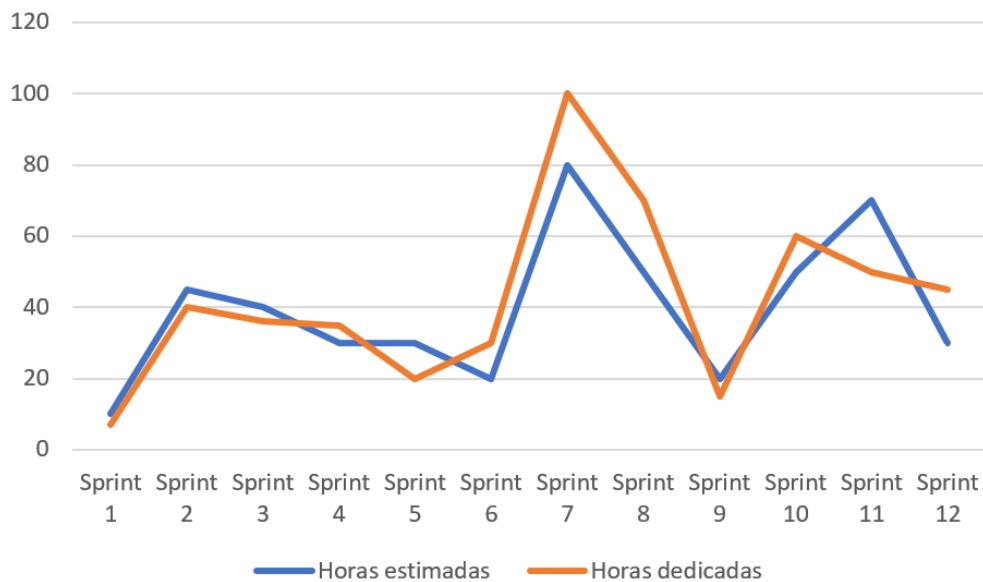


Figura A.2: Gráfico de horas de sprints

A.3. Estudio de viabilidad

Dentro de este apartado se analiza la viabilidad del proyecto desde varios puntos de vista: la viabilidad económica y la viabilidad legal. A continuación se explica más detalladamente cada uno de los subapartados.

Viabilidad económica

La viabilidad económica es un aspecto fundamental en todo proyecto software debido a que estudia tanto los costes como los beneficios que van a producirse a lo largo del desarrollo.

El análisis de costes y beneficios se ha realizado suponiendo que el proyecto se ha llevado a cabo en un entorno empresarial real en el ámbito español. Esto incluye la consideración de los salarios, impuestos y contribuciones a la seguridad social según la normativa vigente.

Costes de Personal

Teniendo en cuenta que se han dedicado 472 horas en total para el desarrollo del proyecto y que el salario bruto medio en Castilla y León es 21773€¹, el salario bruto correspondiente a ese tiempo es de 2950€

Descripción	Coste Bruto
Sueldo Bruto	2,950€
IRPF (11,64 %)	-344€
Sueldo Neto	2,606€

Tal como se puede observar, se ha realizado un cálculo simulado incluyendo la retención del IRPF, se obtiene un valor final de 2,606 euros netos de salario.

Costes de Hardware

Descripción	Coste Total	Amortización Anual
Ordenador de desarrollo	1,000€	200€
Periféricos (monitor, teclado, ratón)	300€	60€
Total	1,300€	260€

En este apartado tenemos en cuenta el valor del hardware que se ha utilizado, tanto el ordenador como sus periféricos. Para que los cálculos se correspondan con la realidad, se realiza una amortización a 5 años de los elementos utilizados.

Costes de Software

Descripción	Coste Total	Amortización Anual
Licencia de Sistema Operativo (Windows 10 Pro)	200€	40€
Servidor de Render (pago anual)	84€	84€
Total	284€	124€

Los costes de software son reducidos ya que se ha trabajado lo máximo posible con bibliotecas y software gratuito que no necesita recibir ningún tipo de pago, exceptuando el host del *backend* por motivos de persistencia de datos. Debido al motivo expuesto, solo se tiene en cuenta el sistema operativo del ordenador utilizado con una amortización similar a los elementos de

¹Bankinter: <https://www.bankinter.com/blog/mercados/salario-medio-espana-comparativa>

hardware (Duración de 5 años), y el servicio de host de pago mensual y con un valor anual mostrado en la tabla.

Costes Varios

Descripción	Coste Anual
Internet y comunicaciones	600€
Electricidad y otros suministros de oficina	300€
Coste curso formación angular	200€
Total	1100€

Estos costes van referidos a todos aquellos externos al sistema en el que se desarrolla el proyecto pero son esenciales para una correcta continuidad y trabajo. Actualmente de este apartado solo ha sido necesario tener en cuenta servicios como internet y electricidad ya que no se han usado elementos físicos extra para el desarrollo. De manera adicional, para la creación de este proyecto también se ha tenido en cuenta el coste de los cursos realizados.

Total de Costes Directos

Descripción	Coste
Personal (Neto)	2,606€
Hardware (Amortización)	260€
Software (Amortización)	124€
Varios	1100€
Total Anual	4,090€

Esta tabla muestra un resumen de la suma de los gastos anuales de todos los apartados anteriormente mencionados, por lo que anualmente este proyecto cuesta realizarle 4,090 euros.

Beneficios

Este proyecto no ha tenido en cuenta una retribución por el uso o distribución del programa ya que su objetivo es ser una fuente fiable de información y gratuita para todos las personas usuarias, y especialmente enfocado a fines didácticos y sociales.

Viabilidad legal

Para poder analizar la viabilidad legal de este proyecto es inicialmente necesario comprobar todas las bibliotecas externas utilizadas en las funcio-

nalidades. En la tabla de debajo se listan las dependencias principales con sus respectivas licencias, las cuales nos pueden llegar a limitar los permisos y las restricciones que podemos implantar en los usuarios.

Biblioteca	Descripción	Licencia
Flask	Microframework web para Python	BSD
Flask-Bcrypt	Proporciona utilidades Bcrypt para Flask	MIT
Flask-Cors	Manejo de CORS en aplicaciones Flask	MIT
Flask-JWT-Extended	Extensiones JWT para Flask	MIT
Flask-Login	Manejo de sesiones de usuario en Flask	MIT
Flask-RESTful	Extensión para construir APIs en Flask	BSD
Flask-SQLAlchemy	Extensión de Flask para SQLAlchemy	MIT
SQLAlchemy	Toolkit SQL y ORM para Python	MIT
openpyxl	Librería para leer/escribir archivos Excel	MIT
requests	Librería para realizar peticiones HTTP	Apache License 2.0
gunicorn	Servidor HTTP WSGI para UNIX	MIT
beautifulsoup4	Librería para parsear documentos HTML	MIT
PrimeNG	Librería para elementos Angular	MIT
SweetAlert2	Librería para modales	MIT

Tabla A.1: Resumen de licencias de las bibliotecas utilizadas en el proyecto

Una vez obtenidas las licencias, las ordenamos de más restrictivas a menos:

1. Apache License 2.0

- Descripción: Esta licencia de software libre y de código abierto es menos restrictiva que la GPL y permite a los usuarios del software la libertad de usar, modificar, y distribuir el software como lo vean conveniente, siempre que se incluyan las notas de derechos de autor y las licencias de todas las copias o versiones sustanciales del software. [4]
- Ejemplo de biblioteca: `requests`

2. MIT License

- Descripción: La licencia MIT es una de las licencias de software más permisivas. Permite el uso, copia, modificación, fusión, publicación, distribución, sublicencia, y/o venta de copias del software, sin restricciones siempre que se mantenga el aviso de derechos de autor original.[9]

- Ejemplos de bibliotecas: **Flask-Bcrypt**, **Flask-Cors**, **Flask-JWT-Extended**, **Flask-Login**, **Flask-SQLAlchemy**, **gunicorn**, **openpyxl**

3. BSD License

- Descripción: Las licencias BSD son otra familia de licencias de software libre permisivas que tienen menos restricciones comparadas con Apache, permitiendo el uso y redistribución del software con o sin modificación. No se requiere la redistribución del código fuente.[\[10\]](#)
- Ejemplos de bibliotecas: **Flask**, **Flask-RESTful**

Teniendo en cuenta toda esta información, la decisión más acertada para este proyecto sería una licencia de **MIT**. Esto se debe a que es una licencia simple que permite a las personas usuarias realizar operaciones y modificaciones por cuenta propia al programa siempre que se incluya el aviso de derechos de autor. Esto permitiría que más allá de las competencias del cliente en cuestión se pueda desarrollar y adaptar para otras necesidades educativas.

En cuanto a la documentación y las imágenes, todo lo incluido de terceros cuenta con Creative Commons. Como el programa en sí tiene una licencia que permite modificaciones, la elección es **Creative Commons**, concretamente la licencia **CC BY**, la cuál permite su uso siempre que se de crédito al autor original. [\[3\]](#)

Elemento	Licencia
Código fuente	MIT
Documentación	CC BY
Imágenes	CC BY

Tabla A.2: Resumen de licencias utilizadas

Apéndice B

Especificación de Requisitos

B.1. Introducción

Este apartado de los anexos contempla la especificación de los requisitos del proyecto desarrollado. Este apartado es de gran importancia debido a su utilidad para dejar unas pautas claras acerca de qué funcionalidades se esperan y para definir una buena planificación de tareas que asegure un desarrollo satisfactorio del proyecto.

B.2. Objetivos generales

Este proyecto tiene como objetivos principales los siguientes:

- Proporcionar una herramienta web que ofrezca un catálogo de libros de prehistoria aptos para la infancia en base a los hallazgos científicos respecto a los roles de género, y mejorar así la educación o al menos asegurar que no se perpetúan los sesgos de género que ya han sido clarificados por la ciencia.
- Aportar una herramienta didáctica para que cualquier persona pueda evaluar la adecuación de un libro, mostrando para ello la metodología de evaluación seguida en el catálogo.
- Brindar herramientas al personal de gestión del catálogo para poder modificar los datos internos de la aplicación mediante una interfaz gráfica.

- Almacenar los libros mostrados de manera segura con opciones de carga y descarga de los datos.

B.3. Catálogo de requisitos

En este apartado se muestran todos los requisitos funcionales y no funcionales recogidos en el proyecto.

Requisitos Funcionales

1. RF-1: Gestión de catálogo

- a) **RF-1.1: Listar libros** Permitir a la persona con rol de administración ver una lista completa de libros disponibles.
- b) **RF-1.2: Editar libro** Permitir a la persona con rol de administración modificar la información de un libro existente.
- c) **RF-1.3: Eliminar libro** Permitir a la persona con rol de administración borrar un libro del catálogo.
- d) **RF-1.4: Selección de libro** Permitir a la persona con rol de administración seleccionar libros específicos.

2. RF-2: Agregar libro manual

- a) **RF-2.1:** Permitir a la persona con rol de administración agregar nuevos libros al catálogo de forma manual.

3. RF-3: Agregar libro automático

- a) **RF-3.1: Agregar libro automático** Permitir a la persona con rol de administración agregar libros al catálogo de forma automática desde diversas fuentes.
- b) **RF-3.2: Combinar fuentes** Unificar datos de libros provenientes de diferentes fuentes y guardarlos como uno.

4. RF-4: Gestión de roles

- a) **RF-4.1: Crear rol** Permitir a la persona con rol de administración definir nuevos roles de usuario.
- b) **RF-4.2: Editar rol** Permitir a la persona con rol de administración modificar roles existentes.

- c) **RF-4.3: Eliminar rol** Permitir a la persona con rol de administración eliminar roles.
- d) **RF-4.4: Listar roles** Mostrar una lista de roles disponibles.
- e) **RF-4.5: Gestionar permisos** Asignar permisos específicos a los roles.

5. RF-5: Gestión de usuarios

- a) **RF-5.1: Crear usuario** Permitir a la persona con rol de administración agregar nuevos usuarios al sistema.
- b) **RF-5.2: Editar usuario** Permitir a la persona con rol de administración modificar la información de usuarios existentes.
- c) **RF-5.3: Eliminar usuario** Permitir a la persona con rol de administración borrar usuarios.
- d) **RF-5.4: Listar usuarios** Mostrar una lista de usuarios registrados en el sistema.
 - 1) **RF-5.5: Cambiar contraseña** Permitir a los usuarios registrados cambiar su contraseña.

6. RF-6: Importar catálogo

- a) **RF-6.1:** Permitir a la persona con rol de administración importar un catálogo de libros desde un archivo externo.

7. RF-7: Exportar catálogo

- a) **RF-7.1:** Permitir a la persona con rol de administración exportar el catálogo de libros a un archivo.

8. RF-8: Gestionar actividades incluidas en la evaluación de libros

- a) **RF-8.1: Crear actividad** Permitir a la persona con rol de administración definir nuevas actividades.
- b) **RF-8.2: Editar actividad** Permitir a la persona con rol de administración modificar actividades existentes.
- c) **RF-8.3: Eliminar actividad** Permitir a la persona con rol de administración borrar actividades.
- d) **RF-8.4: Listar actividades** Mostrar una lista de actividades disponibles.

- e) **RF-8.5: Obtener actividades** Obtener actividades disponibles del *backend*.

9. **RF-9: Iniciar sesión**

- a) **RF-9.1:** Permitir a los usuarios autenticarse en el sistema para acceder a sus funcionalidades.

10. **RF-10: Buscar libro**

- a) **RF-10.1:** Permitir a los usuarios buscar libros específicos en el catálogo.

11. **RF-11: Obtener catálogo**

- a) **RF-11.1:** Permitir a los usuarios obtener el catálogo completo de libros.

12. **RF-12: Obtener información de libro**

- a) **RF-12.1:** Permitir a los usuarios ver la información detallada de un libro específico.

13. **RF-13: Gestionar valoraciones**

- a) **RF-13.1: Eliminar valoraciones** Permitir a los responsables de la administración eliminar valoraciones.
- b) **RF-13.2: Listar valoraciones** Mostrar una lista de valoraciones realizadas por los usuarios.

14. **RF-14: Obtener estadísticas**

- a) **RF-14.1:** Realizar los cálculos necesarios y obtener las estadísticas guardadas en base de datos y mostrarlo

15. **RF-15: Generar valoraciones**

- a) **RF-15.1:** Permitir la generación de valoraciones basadas en parámetros definidos.

16. **RF-16: Crear sugerencia**

- a) **RF-16.1:** Permitir a los usuarios enviar sugerencias sobre libros.

Requisitos No Funcionales

1. RNF-1: Seguridad

- a) **RNF-1.1:** Implementar autenticación y autorización mediante JWT para proteger el acceso a la API.
- b) **RNF-1.2:** Asegurar que los datos personales de los usuarios estén encriptados y protegidos contra accesos no autorizados.

2. RNF-2: Rendimiento

- a) **RNF-2.1:** La aplicación debe ser capaz de manejar un gran número de solicitudes simultáneas sin degradar su rendimiento.
- b) **RNF-2.2:** Optimizar las consultas a la base de datos para asegurar tiempos de respuesta rápidos.

3. RNF-3: Usabilidad

- a) **RNF-3.1:** La interfaz de usuario debe ser intuitiva y fácil de usar para asegurar una experiencia agradable.
- b) **RNF-3.2:** El diseño debe ser adaptativo para asegurar que la interfaz se vea y funcione correctamente en distintos dispositivos y tamaños de pantalla.

4. RNF-4: Escalabilidad

- a) **RNF-4.1:** El sistema debe ser escalable para poder manejar un creciente número de usuarios y datos.
- b) **RNF-4.2:** El sistema debe ser escalable en términos funcionales, permitiendo la adición de nuevas funcionalidades sin comprometer el rendimiento.

5. RNF-5: Compatibilidad

- a) **RNF-5.1:** La aplicación debe ser compatible con los principales navegadores web.
- b) **RNF-5.2:** Asegurar que la aplicación funcione correctamente en diferentes dispositivos, incluyendo móviles y tablets.

6. RNF-6: Mantenimiento

- a) **RNF-6.1:** El código debe estar bien documentado y estructurado para facilitar su mantenimiento y actualización.

7. RNF-7: Disponibilidad

- a) **RNF-7.1:** La aplicación debe estar disponible el 99.9 % del tiempo, asegurando un mínimo de tiempo de inactividad.
- b) **RNF-7.2:** Implementar estrategias de backup y recuperación para asegurar la disponibilidad de los datos en caso de fallos.

B.4. Especificación de requisitos

Antes de comenzar con los casos de uso, se muestra el diagrama para un entendimiento rápido y visual. Debido a la complejidad y al gran número de funcionalidades, se han separado en dos diagramas para su mejor comprensión.

La figura B.1 representa el conjunto de todas las zonas restringidas de la aplicación que contienen un lazo de dependencia con la funcionalidad de iniciar sesión.

La figura B.2 muestra un diagrama de casos de uso tradicional en el que se cuenta con dos roles, un administrador que tiene acceso a todas las funcionalidades y un usuario que solo puede acceder a elementos limitados.

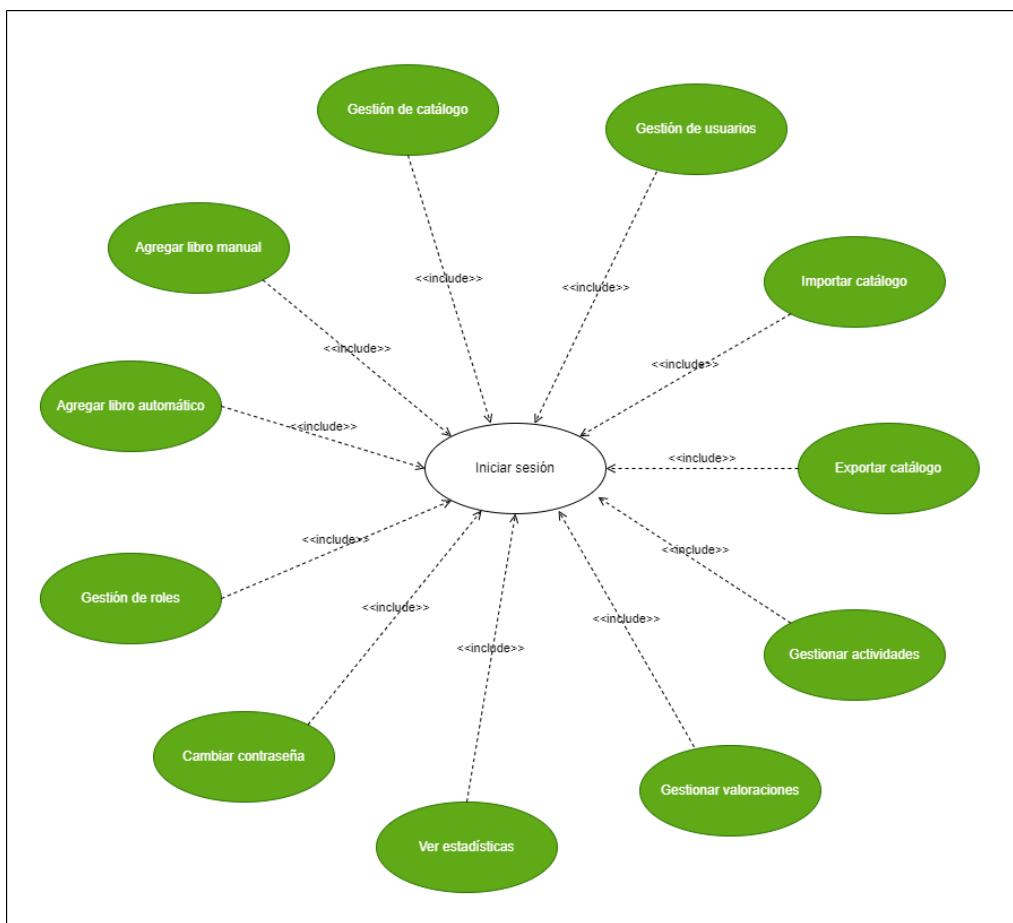


Figura B.1: Diagrama conexión de funciones con autenticación

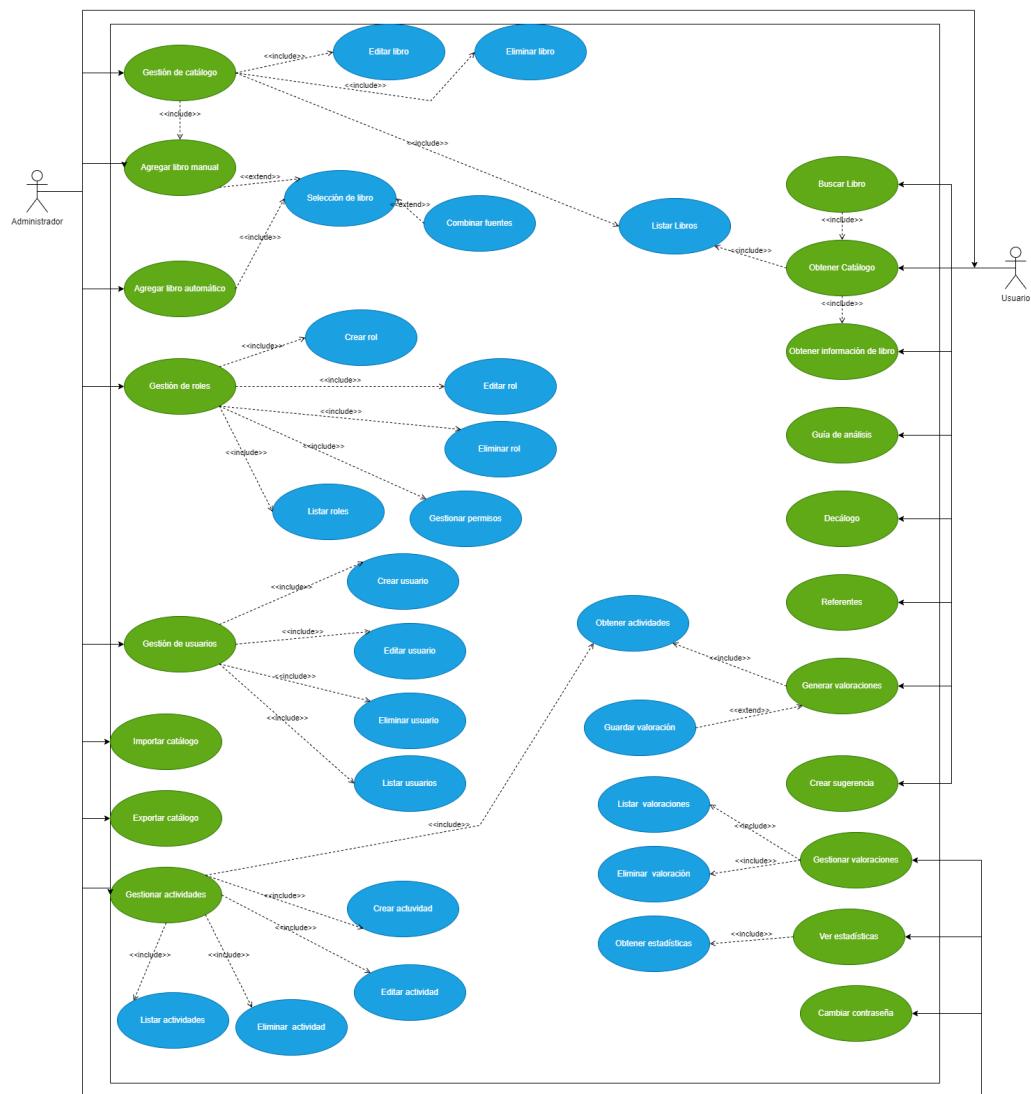


Figura B.2: Diagrama de casos de uso

En esta sección, se presentan los diferentes casos de uso del proyecto. Cada caso de uso describe una secuencia de interacciones entre un usuario/a y el sistema, con el objetivo de alcanzar un resultado específico.

CU-1	Gestión de catálogo
Versión	1.0
Autor	Daniel Fernández Fernández
Roles	Administrador
Requisitos asociados	RF-1.1, RF-1.2, RF-1.3, RF-1.4
Descripción	Permitir a la persona con rol de administración gestionar el catálogo de libros.
Precondición	El administrador debe haber iniciado sesión.
Acciones	<ol style="list-style-type: none"> 1. El administrador selecciona Gestión de catálogo en el menú principal. 2. El sistema muestra la lista completa de libros. 3. El sistema muestra las opciones de edición, eliminación o más información de cada libro. 4. El administrador realiza la acción deseada. 5. El sistema guarda los cambios y actualiza el catálogo.
Postcondición	El catálogo se actualiza con la acción realizada.
Excepciones	El sistema muestra un mensaje de error si la acción no puede completarse.
Importancia	Alta

Tabla B.1: CU-1 Gestión de catálogo.

CU-2	Agregar libro manual
Versión	1.0
Autor	Daniel Fernández Fernández
Roles	Administrador
Requisitos asociados	RF-2.1
Descripción	Permitir a la persona con rol de administración agregar nuevos libros al catálogo de forma manual.
Precondición	El administrador debe haber iniciado sesión.
Acciones	<ol style="list-style-type: none"> 1. El administrador selecciona Agregar libro manual en el menú. 2. El sistema muestra un formulario para agregar un nuevo libro. 3. El administrador completa el formulario con la información del libro. 4. El administrador envía el formulario. 5. El sistema guarda el nuevo libro en el catálogo.
Postcondición	El libro nuevo se agrega al catálogo.
Excepciones	El sistema no deja agregar si faltan datos obligatorios.
Importancia	Alta

Tabla B.2: CU-2 Agregar libro manual.

CU-3	Agregar libro automático
Versión	1.0
Autor	Daniel Fernández Fernández
Roles	Administrador
Requisitos asociados	RF-3.1, RF-3.2
Descripción	Permitir a la persona con rol de administración agregar libros al catálogo de forma automática desde diversas fuentes.
Precondición	El administrador debe haber iniciado sesión.
Acciones	<ol style="list-style-type: none"> 1. El administrador selecciona Agregar libro automáticamente en el menú. 2. El sistema muestra opciones de fuentes de datos disponibles. 3. El administrador selecciona la fuente de datos o la combinación de los mismos. 4. El sistema genera un formulario de los datos de los libros con las fuentes seleccionadas. 5. El sistema agrega el libro generado al catálogo.
Postcondición	Los libros nuevos se agregan al catálogo.
Excepciones	El sistema muestra un mensaje de error si no se pueden obtener datos de las fuentes.
Importancia	Media

Tabla B.3: CU-3 Agregar libro automático.

CU-4	Gestión de roles
Versión	1.0
Autor	Daniel Fernández Fernández
Roles	Administrador
Requisitos asociados	RF-4.1, RF-4.2, RF-4.3, RF-4.4, RF-4.5
Descripción	Permitir a la persona con rol de administración gestionar los roles de usuario y sus permisos.
Precondición	El administrador debe haber iniciado sesión.
Acciones	<ol style="list-style-type: none"> 1. El administrador selecciona Gestión de roles en el menú. 2. El sistema muestra la lista de roles existentes. 3. El administrador selecciona la opción de crear, editar, eliminar o modificar permisos de un rol. 4. El sistema muestra el formulario correspondiente. 5. El administrador completa el formulario y guarda los cambios.
Postcondición	Los cambios en los roles se actualizan en el sistema.
Excepciones	El sistema muestra un mensaje de error si la acción no puede completarse.
Importancia	Alta

Tabla B.4: CU-4 Gestión de roles.

CU-5	Gestión de usuarios
Versión	1.0
Autor	Daniel Fernández Fernández
Roles	Administrador
Requisitos asociados	RF-5.1, RF-5.2, RF-5.3, RF-5.4
Descripción	Permitir a la persona con rol de administración gestionar los usuarios del sistema.
Precondición	El administrador debe haber iniciado sesión.
Acciones	<ol style="list-style-type: none"> 1. El administrador selecciona Gestión de usuarios en el menú. 2. El sistema muestra la lista de usuarios existentes. 3. El administrador selecciona la opción de crear, editar o eliminar un usuario. 4. El sistema muestra el formulario correspondiente. 5. El administrador completa el formulario y guarda los cambios.
Postcondición	Los cambios en los usuarios se actualizan en el sistema.
Excepciones	El sistema muestra un mensaje de error si la acción no puede completarse.
Importancia	Alta

Tabla B.5: CU-5 Gestión de usuarios.

CU-5.5	Cambiar contraseña
Versión	1.0
Autor	Daniel Fernández Fernández
Roles	Administrador
Requisitos asociados	RF-5.5
Descripción	Permitir a los usuarios registrados cambiar su contraseña.
Precondición	El usuario debe haber iniciado sesión.
Acciones	<ol style="list-style-type: none"> 1. El usuario selecciona Cambiar contraseña en el menú. 2. El sistema muestra un formulario para cambiar la contraseña. 3. El usuario ingresa la nueva contraseña y la confirma. 4. El usuario envía el formulario. 5. El sistema actualiza la contraseña del usuario en la base de datos.
Postcondición	La contraseña del usuario se actualiza correctamente.
Excepciones	El sistema muestra un mensaje de error si la contraseña antigua no es correcta.
Importancia	Alta

Tabla B.6: CU-5.5 Cambiar contraseña.

CU-6	Importar catálogo
Versión	1.0
Autor	Daniel Fernández Fernández
Roles	Administrador
Requisitos asociados	RF-6.1
Descripción	Permitir a la persona con rol de administración importar un catálogo de libros desde un archivo externo.
Precondición	El administrador debe haber iniciado sesión.
Acciones	<ol style="list-style-type: none"> 1. El administrador selecciona Importar en el menú. 2. El sistema muestra una advertencia de seguridad. 3. El sistema muestra una interfaz para seleccionar el archivo de importación. 4. El administrador selecciona el archivo y confirma la importación. 5. El sistema procesa el archivo y agrega los libros al catálogo.
Postcondición	Los libros del archivo se agregan al catálogo.
Excepciones	El sistema muestra un mensaje de error si no se puede procesar.
Importancia	Media

Tabla B.7: CU-6 Importar catálogo.

CU-7	Exportar catálogo
Versión	1.0
Autor	Daniel Fernández Fernández
Roles	Administrador
Requisitos asociados	RF-7.1
Descripción	Permitir a la persona con rol de administración exportar el catálogo de libros a un archivo.
Precondición	El administrador debe haber iniciado sesión.
Acciones	<ol style="list-style-type: none"> 1. El administrador selecciona Exportar en el menú. 2. El sistema muestra una interfaz para seleccionar el formato de exportación. 3. El administrador selecciona el formato y confirma la exportación. 4. El sistema genera el archivo y lo pone disponible para descarga.
Postcondición	El archivo exportado está disponible para descarga.
Excepciones	El sistema muestra un mensaje de error si la exportación falla.
Importancia	Media

Tabla B.8: CU-7 Exportar catálogo.

CU-8	Gestionar actividades
Versión	1.0
Autor	Daniel Fernández Fernández
Roles	Administrador
Requisitos asociados	RF-8.1, RF-8.2, RF-8.3, RF-8.4
Descripción	Permitir a la persona con rol de administración gestionar las actividades de las estimaciones.
Precondición	El administrador debe haber iniciado sesión.
Acciones	<ol style="list-style-type: none"> 1. El administrador selecciona Gestión de actividades en el menú. 2. El sistema muestra la lista de actividades existentes. 3. El administrador selecciona la opción de crear, editar o eliminar una actividad. 4. El sistema muestra el formulario correspondiente. 5. El administrador completa el formulario y guarda los cambios.
Postcondición	Los cambios en las actividades se actualizan en el sistema.
Excepciones	El sistema muestra un mensaje de error si la acción no puede completarse.
Importancia	Media

Tabla B.9: CU-8 Gestionar actividades.

CU-8.5	Obtener actividades
Versión	1.0
Autor	Daniel Fernández Fernández
Roles	Administrador y usuarios
Requisitos asociados	RF-8.5
Descripción	Permitir a los usuarios obtener la lista de actividades disponibles en el apartado de valoración.
Precondición	El usuario debe encontrarse en el valorador.
Acciones	<ol style="list-style-type: none"> 1. El usuario selecciona despliega los selectores de actividades en el valorador. 2. El sistema muestra la lista de actividades disponibles.
Postcondición	El usuario ve la lista de actividades disponibles.
Excepciones	El sistema no muestra ninguna actividad en la lista.
Importancia	Media

Tabla B.10: CU-8.5 Obtener actividades.

CU-9	Iniciar sesión
Versión	1.0
Autor	Daniel Fernández Fernández
Roles	Administrador
Requisitos asociados	RF-9.1
Descripción	Permitir a los usuarios autenticarse en el sistema para acceder a sus funcionalidades.
Precondición	El usuario debe estar registrado en el sistema.
Acciones	<ol style="list-style-type: none"> 1. El usuario selecciona Iniciar sesión en la página principal. 2. El sistema muestra el formulario de inicio de sesión. 3. El usuario ingresa su nombre de usuario y contraseña. 4. El usuario envía el formulario. 5. El sistema verifica las credenciales y autentica al usuario.
Postcondición	El usuario autenticado puede acceder a las funcionalidades del sistema.
Excepciones	El sistema muestra un mensaje de error si las credenciales son incorrectas.
Importancia	Alta

Tabla B.11: CU-9 Iniciar sesión.

CU-10	Buscar libro
Versión	1.0
Autor	Daniel Fernández Fernández
Roles	Administrador y usuarios
Requisitos asociados	RF-10.1
Descripción	Permitir a los usuarios buscar libros específicos en el catálogo.
Precondición	El usuario debe estar en la página principal del catálogo o en la de inicio.
Acciones	<ol style="list-style-type: none"> 1. El usuario ingresa un término de búsqueda en el campo de búsqueda. 2. El usuario envía la búsqueda. 3. El sistema procesa la búsqueda y muestra los resultados correspondientes.
Postcondición	El usuario ve los resultados de la búsqueda.
Excepciones	El sistema muestra el catálogo vacío si no encuentra resultados.
Importancia	Alta

Tabla B.12: CU-10 Buscar libro.

CU-11	Obtener catálogo
Versión	1.0
Autor	Daniel Fernández Fernández
Roles	Administrador y usuarios
Requisitos asociados	RF-11.1
Descripción	Permitir a los usuarios obtener el catálogo completo de libros.
Precondición	El usuario debe estar en la página principal del catálogo.
Acciones	<ol style="list-style-type: none"> 1. El usuario selecciona Catálogo en el menú. 2. El sistema muestra la lista completa de libros disponibles.
Postcondición	El usuario ve la lista completa de libros.
Excepciones	El sistema muestra un mensaje si no hay libros disponibles.
Importancia	Alta

Tabla B.13: CU-11 Obtener catálogo.

CU-12	Obtener información de libro
Versión	1.0
Autor	Daniel Fernández Fernández
Roles	Administrador y usuarios
Requisitos asociados	RF-12.1
Descripción	Permitir a los usuarios ver la información detallada de un libro específico.
Precondición	El usuario debe estar en la página del catálogo. El administrador puede acceder desde el gestor del catálogo
Acciones	<ol style="list-style-type: none">1. El usuario selecciona un libro de la lista.2. El sistema muestra la información detallada del libro seleccionado.
Postcondición	El usuario ve la información detallada del libro.
Importancia	Alta

Tabla B.14: CU-12 Obtener información de libro.

CU-13	Gestionar valoraciones
Versión	1.0
Autor	Daniel Fernández Fernández
Roles	Administrador
Requisitos asociados	RF-13.1, RF-13.2
Descripción	Mostrar una lista de valoraciones realizadas por los usuarios y sus acciones.
Precondición	El usuario debe haber iniciado sesión.
Acciones	<ol style="list-style-type: none"> 1. El administrador selecciona valoraciones guardadas en el menú. 2. El sistema muestra la lista de valoraciones disponibles. 3. El administrador puede borrar valoraciones.
Postcondición	El usuario ve la lista de valoraciones realizadas actualizada.
Excepciones	El sistema muestra un listado vacío o un mensaje de error si no borra correctamente.
Importancia	Media

Tabla B.15: CU-13 Gestionar valoraciones.

CU-14	Obtener estadísticas
Versión	1.0
Autor	Daniel Fernández Fernández
Roles	Administrador
Requisitos asociados	RF-14.1
Descripción	Permitir a los usuarios obtener estadísticas relacionadas con la actividad y el catálogo.
Precondición	El usuario debe haber iniciado sesión.
Acciones	<ol style="list-style-type: none"> 1. El administrador debe de encontrarse en el panel de administrador. 2. El sistema muestra las estadísticas calculadas.
Postcondición	El usuario ve las estadísticas relacionadas con la actividad y el catálogo.
Excepciones	El sistema muestra unos gráficos vacíos si no se pueden obtener las estadísticas.
Importancia	Media

Tabla B.16: CU-14 Obtener estadísticas.

CU-15	Generar valoraciones
Versión	1.0
Autor	Daniel Fernández Fernández
Roles	Administrador y usuarios
Requisitos asociados	RF-15.1
Descripción	Permitir la generación de valoraciones basadas en parámetros definidos.
Precondición	El usuario debe de encontrarse en valoraciones.
Acciones	<ol style="list-style-type: none"> 1. El usuario selecciona valoraciones en el menú. 2. El sistema muestra una interfaz para definir los parámetros de la valoración. 3. El usuario ingresa los parámetros y confirma la generación. 4. El sistema genera las valoraciones basadas en los parámetros introducidos.
Postcondición	Las valoraciones generadas calculan y se muestran.
Excepciones	El sistema muestra un mensaje de error si la valoración falla.
Importancia	Media

Tabla B.17: CU-15 Generar valoraciones.

CU-16	Crear sugerencia
Versión	1.0
Autor	Daniel Fernández Fernández
Roles	Administrador y usuarios
Requisitos asociados	RF-16.1
Descripción	Permitir a los usuarios enviar sugerencias sobre libros.
Precondición	El usuario debe estar en el catálogo.
Acciones	<ol style="list-style-type: none"> 1. El usuario selecciona el botón de sugerencia en el menú. 2. El sistema muestra un formulario para ingresar la sugerencia. 3. El usuario completa el formulario y envía la sugerencia. 4. El sistema manda la sugerencia por correo al personal de administración.
Postcondición	La sugerencia se envía correctamente.
Excepciones	El sistema muestra un mensaje de error si la sugerencia no puede ser enviada.
Importancia	Media

Tabla B.18: CU-16 Crear sugerencia.

Apéndice C

Especificación de diseño

C.1. Introducción

En este apartado se va a tratar la especificación de diseño desarrollado a lo largo de este proyecto, dividido en diferentes apartados considerados clave para el éxito del desarrollo de esta web. Una buena especificación de diseño nos ha permitido, partiendo de la idea inicial de cliente, transformar sus necesidades en un producto final completo apoyándonos en un sistema lo más estructurado y eficaz posible.

C.2. Diseño del modelo datos

La aplicación web contiene una serie de entidades que aseguran el cumplimiento de los requerimientos:

- **Libros:** Esta entidad almacena información sobre los libros disponibles en el sistema. Incluye detalles como el título, ISBN, editorial, descripción, año de publicación, puntuaciones y ubicaciones del estudio.
- **LibrosAutomaticos:** Almacena información de manera temporal acerca los libros obtenidos mediante web scraping, incluyendo detalles como título, ISBN, editorial y descripción.
- **Fecha_modificacion:** Registra la última fecha y hora de modificación de los datos del catálogo, indicando así a los usuarios si han existido modificaciones y cuando.

- **GestionEstimacion:** Contiene detalles de las actividades de producción, poder y mantenimiento tanto de hombres como de mujeres, necesarias para realizar valoraciones de libros.
- **Roles:** Define los diferentes roles disponibles en el sistema, como administrador, usuario, etc. Cada rol puede estar asociado con múltiples usuarios.
- **Usuarios:** Almacena información sobre los usuarios del sistema, incluyendo su nombre, correo electrónico, contraseña encriptada y el rol asignado.
- **Botones:** Contiene información sobre los diferentes botones de la interfaz del sistema y los roles autorizados para utilizarlos, permitiendo un ajuste dinámico de permisos en base a roles.
- **Estimacion:** Registra las estimaciones realizadas por los usuarios, incluyendo detalles sobre género, actividades, ubicación, título e ISBN del estudio, y los resultados obtenidos.
- **EstadisticasPorMes:** Almacena estadísticas mensuales de los libros, como el número de libros, visitas totales, estimaciones, usuarios y el libro más visitado.
- **EstadisticasPorMesAuxiliar:** Similar a EstadisticasPorMes, pero utilizada como una entidad auxiliar durante la importación de catálogo y almacenamiento temporal.
- **Colaboradores:** Almacena información sobre los colaboradores de valoraciones de libros, incluyendo su nombre, apellido e institución.

Tras mencionar todas las entidades existentes, cabe destacar que todas las entidades son independientes unas de otras excepto dos, la entidad Usuarios y la entidad Roles, ya que todo usuario debe de tener asignado un solo rol, pero un rol puede estar asignado para varios roles. Esto se ve de manera más visual en el diagrama relacional de la figura C.1:

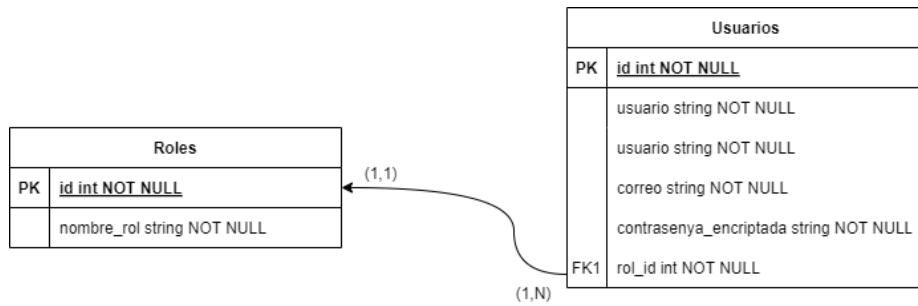
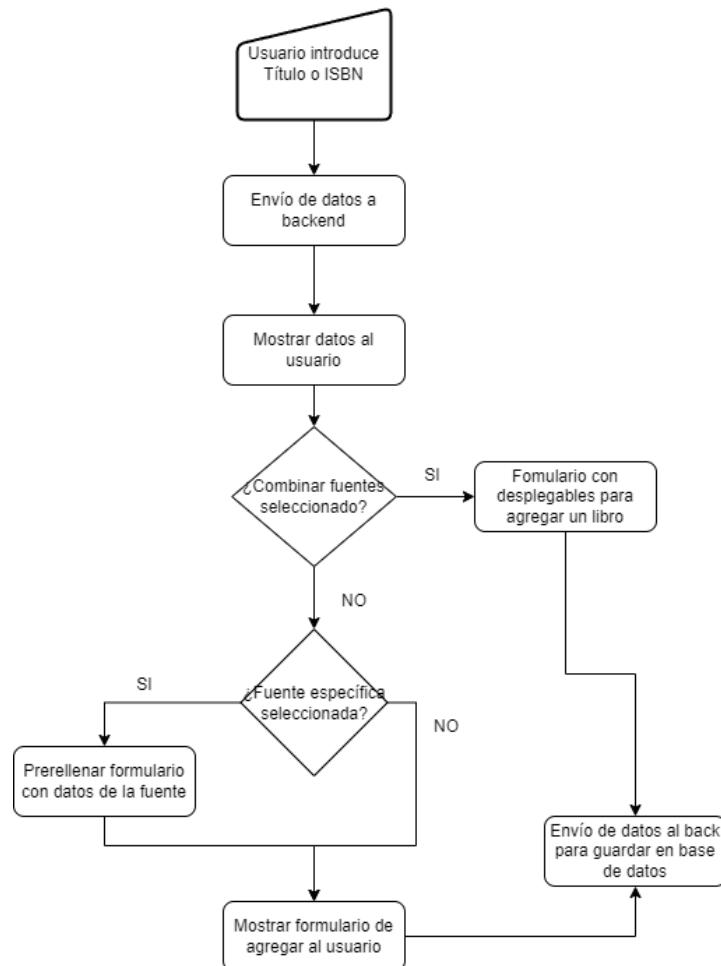


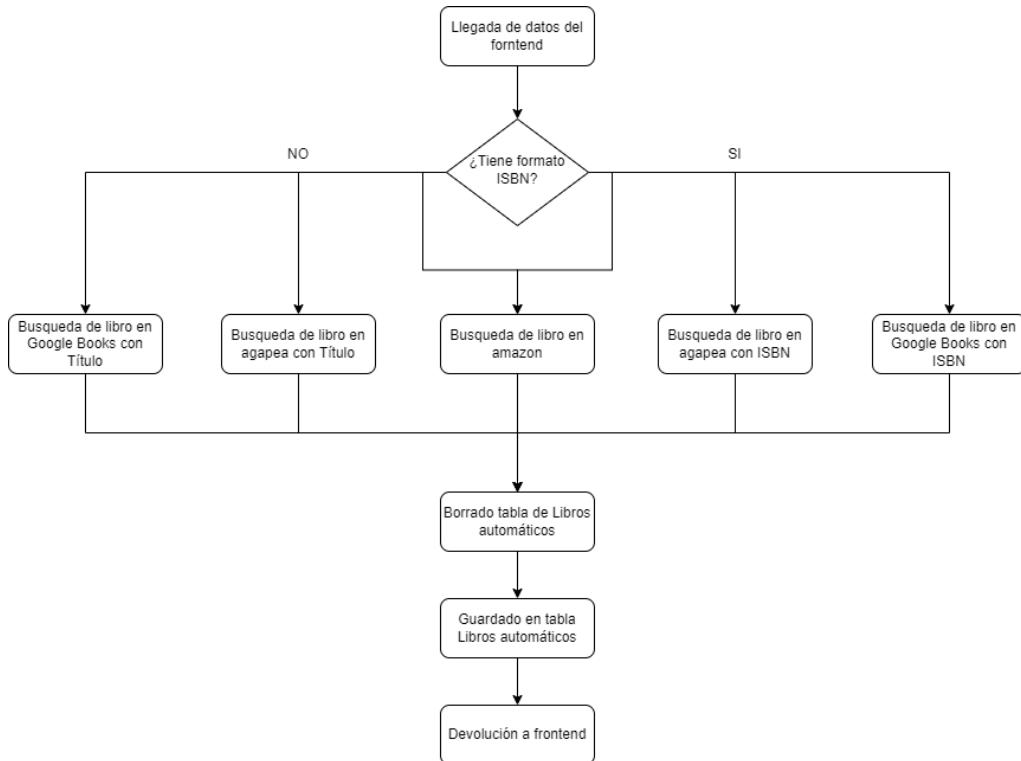
Figura C.1: Diagrama Relacional

C.3. Diseño procedimental

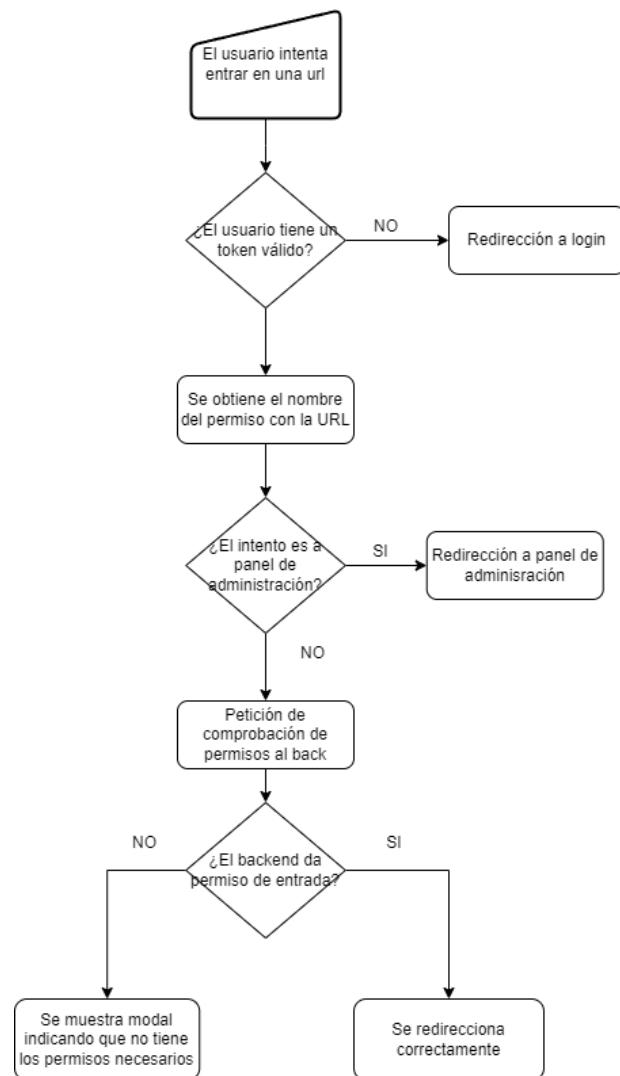
En este apartado se van a mostrar mediante diagramas de flujo algunos de los procedimientos internos más interesantes y de mayor complejidad de este proyecto.

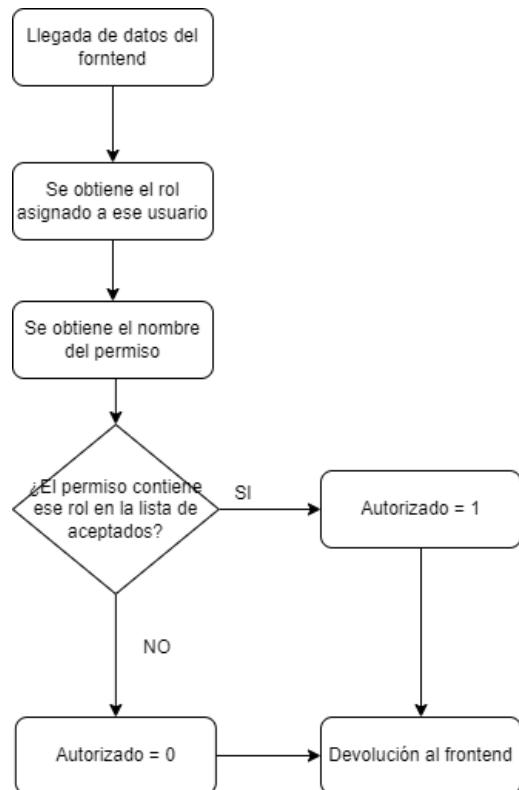
En el diagrama de flujo de la figura C.2 podemos apreciar todos los pasos que se realizan cuando un usuario quiere obtener un libro a través del web scraping desde el *frontend*. En la figura C.5 se aprecia la parte del *backend*, donde se ejecuta toda la lógica

Frontend:Figura C.2: Diagrama de flujo Web scraping *Frontend*

Backend:Figura C.3: Diagrama de flujo Web scraping *Backend*

En las figuras C.4 y C.5 muestra el sistema que permite a la web bloquear o permitir la entrada dinámicamente a los usuarios a ciertas partes de la web utilizando permisos, los cuales no son estáticos y son editables.

Frontend:Figura C.4: Diagrama de flujo permisos *Frontend*

Backend:Figura C.5: Diagrama de flujo permisos *Backend***C.4. Diseño de interfaces**

Para realizar un primer prototipado del diseño de interfaz de la web se utilizó la herramienta Justimind, que ha permitido el diseño y la estructura de las funciones básicas de este proyecto.

A continuación, se muestran algunas de las interfaces realizadas con la herramienta:

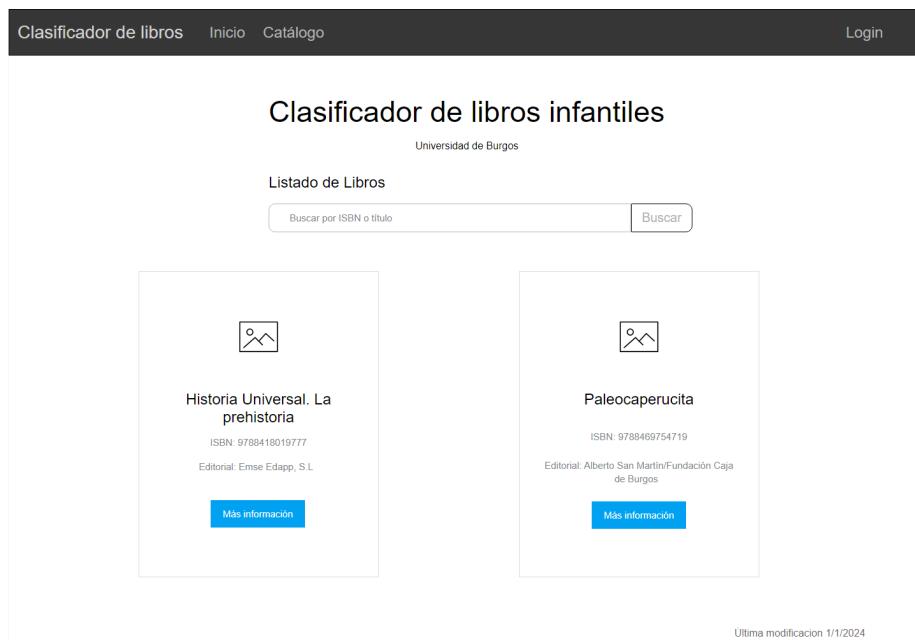


Figura C.6: Prototipo catálogo

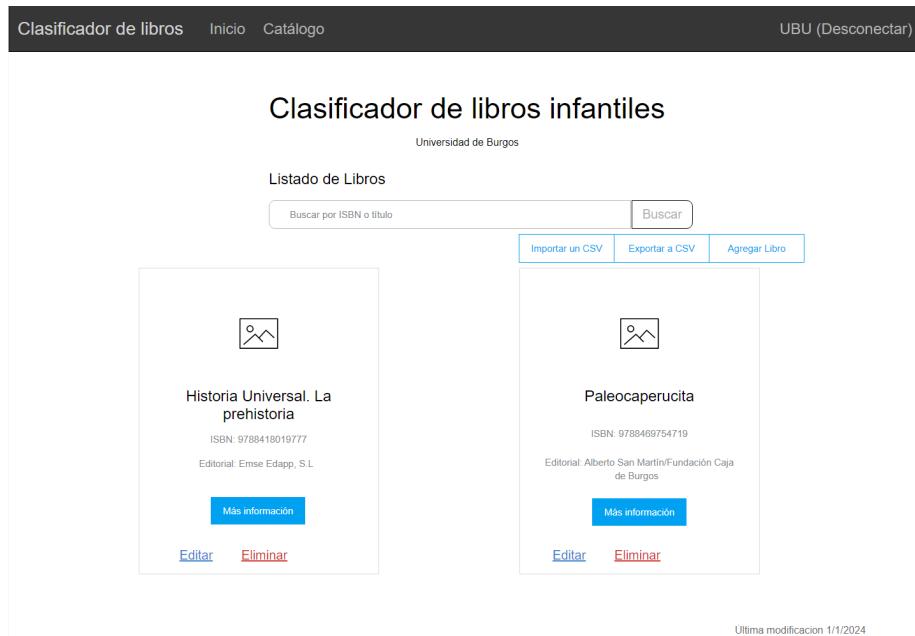


Figura C.7: Prototipo catálogo (Vista de administrador)



Descripción del libro

Correspondiente descripción

Puntuación obtenida

Valor de la puntuación

Figura C.8: Prototipo Más Información



Clasificador de libros Infantiles

Universidad de Burgos

Iniciar sesión

Usuario	<input type="text"/>
Nombre de usuario	
Contraseña	<input type="password"/>
Contraseña	
<input type="button" value="Iniciar Sesión"/>	

Figura C.9: Prototipo Login

C.5. Diseño de la arquitectura

Dentro de este apartado se exponen los patrones de diseño utilizados durante el desarrollo de esta aplicación web. Para una mejor comprensión se separan los patrones de *frontend* y *backend* y finalmente la combinación de ambos.

Patrones del *backend*

Patrón modular [7]

El patrón modular tiene como objetivo principal separar las funcionalidades del código en módulos separados y lo más independientes posibles. Cada módulo consiste en una o varias funcionalidades relacionadas entre sí. Esto contiene la gran ventaja de poder reutilizar el código y tener una escalabilidad y mantenibilidad muy simple y eficaz.

A nivel de ejemplo del *backend*, en la figura C.10 se puede apreciar cómo las funcionalidades principales se encuentran separadas en carpetas con sus correspondientes archivos, utilizando uno general para conectarlos entre si y poder usar todos los componentes.

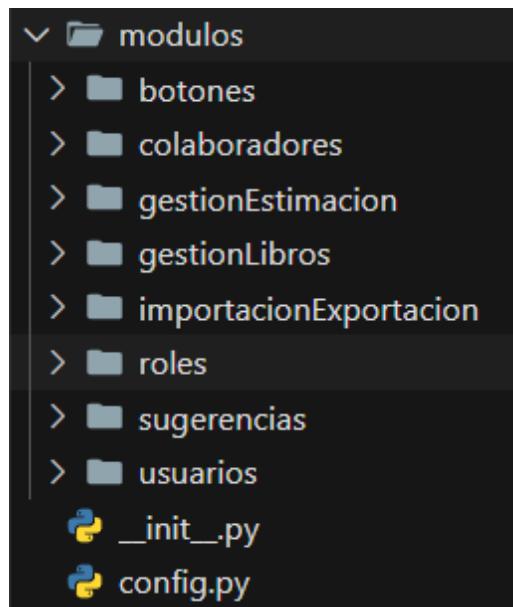
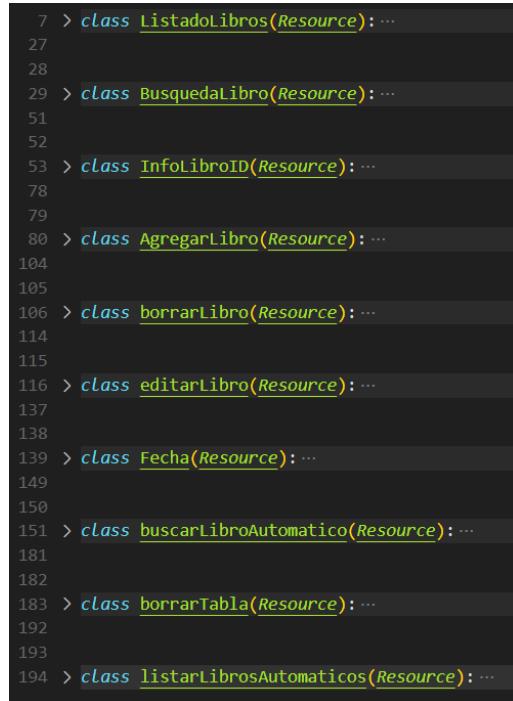


Figura C.10: Módulos *backend*

Un ejemplo más concreto de estos módulos es la carpeta `gestionLibros`, la cual es la encargada de toda la lógica relacionada con todas las operaciones del catálogo.



```
7 > class ListadoLibros(Resource): ...
27
28
29 > class BusquedaLibro(Resource): ...
51
52
53 > class InfoLibroID(Resource): ...
78
79
80 > class AgregarLibro(Resource): ...
104
105
106 > class borrarLibro(Resource): ...
114
115
116 > class editarLibro(Resource): ...
137
138
139 > class Fecha(Resource): ...
149
150
151 > class buscarLibroAutomatico(Resource): ...
181
182
183 > class borrarTabla(Resource): ...
192
193
194 > class listarLibrosAutomaticos(Resource): ...
```

Figura C.11: Funciones `gestionLibros`

Patrón repositorio[8]

El patrón repositorio es utilizado para permitir la separación entre el acceso a los datos de la base de datos con la lógica de negocio. Al realizar esta acción podemos tener grandes ventajas como son la limpieza, la mantenibilidad y la posibilidad de realización de pruebas unitarias de manera sencilla.

A nivel de ejemplo esto se puede apreciar en la estructura del proyecto al existir una separación entre todos los módulos de funcionalidades y el apartado de modelos.

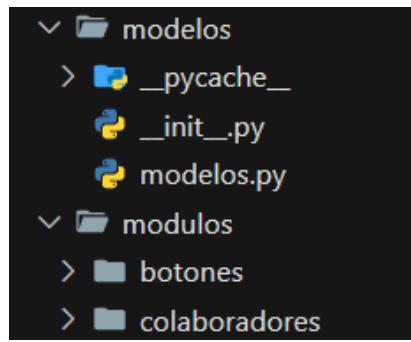


Figura C.12: Patrón repositorio

Patrones del *frontend*

Patrón de componentes [5]

El patrón de componentes consiste en dividir la interfaz del usuario en elementos totalmente independientes y reutilizables. Cada componente existente en el proyecto contiene su propia lógica y su propia y diseño, lo cual permite tener una modularización completa y muy fácilmente escalable. Este patrón es esencial para poder construir vistas muy complejas utilizando componentes muy pequeños.

Para exemplificarlo, en la figura C.13 se puede observar algunos de los componentes creados para la aplicación web, donde se observa la separación completa y los elementos de cada componente.

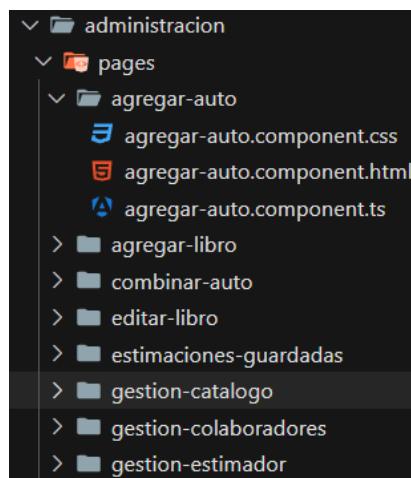


Figura C.13: ComponentesAngular

Patrón de servicios [6]

El patrón de servicios consiste en generar archivos que contengan la lógica que sea compartida por varios de los componentes. En el caso de este proyecto, los servicios se han utilizado para realizar las llamadas necesarias al *backend*, definiendo así una serie de funciones que pueden compartir varios de los componentes existentes, en base a sus necesidades. Para poder utilizar correctamente estos elementos, han de inyectarse en los constructores de todos aquellos componentes que quieran interactuar con ellos.

A modo de ejemplo, en la figura C.14 se puede ver una parte de uno de los servicios existentes (en este caso es el servicio de los libros), donde entre todos los elementos que contiene, se pueden observar las funciones que realizan las llamadas al *backend* y una función donde se establecen las cabeceras de todas las llamadas.

```
16 > private getHeaders() { ...
24   }
25
26 > conexionLibroInfo(libro: any) { ...
28   }
29
30 > listarLibros(): Observable<any> { ...
32   }
33
34 > buscarLibro(busqueda: string): Observable<any> { ...
36   }
37
38 > obtenerInfoLibro(id: number): Observable<any> { ...
40   }
41
42 > agregarLibro(libro: any): Observable<any> { ...
44   }
45
46 > editarLibro(id: number, Libro: any): Observable<any> { ...
48   }
```

Figura C.14: Servicio de libros

Patrones conjuntos

Patrón cliente-servidor [1]

Este último patrón es uno de los más importantes de este proyecto y se genera al existir de manera separada el *backend* y el *frontend*. La forma en la que se muestra este patrón es de la siguiente manera:

- Cliente: El cliente es el responsable de la interfaz de usuario, y en este proyecto recaería sobre la parte de Angular.
- Servidor: Son los responsables de devolver los recursos pedidos por los clientes, por lo que esta parte recaería sobre el *backend* Flask.

El funcionamiento de este patrón se realiza mediante solicitudes HTTP por parte del cliente al servidor, los cuáles recibe el *backend*, que procesa la información pedida utilizando parámetros recibidos del cliente (si existieran), y devuelve un resultado para que el cliente lo procese.

Apéndice D

Documentación técnica de programación

D.1. Introducción

Este anexo contiene toda la información técnica de programación, incluyendo la estructura de directorios, las instalaciones necesarias, el proceso de arranque de la aplicación y los test generados. Esto es de gran utilidad para todo desarrollador/a que quiera realizar modificaciones al proyecto actual.

D.2. Estructura de directorios

Estructura de directorios del *frontend*

- `frontendAngular/`
 - `.angular/` – Archivos de configuración específica de Angular.
 - `dist/` – Archivos generados después de la construcción del proyecto.
 - `node_modules/` – Dependencias de Node.js instaladas.
 - `src/` – Directorio principal de código fuente.
 - `app/`
 - ◊ `administracion/` – Páginas específicas de la administración y su módulo.

- ◊ **aplicacion/** – Páginas específicas de la administración y su módulo.
- ◊ **auth/** – Módulo y componentes de autenticación.
- ◊ **prime-ng/** – Configuración del módulo PrimeNG.
- ◊ **services/** – Servicios que manejan la lógica de negocio y comunicación con la API.
- ◊ **shared/** – Componentes compartidos en varios componentes.
- ◊ **app-routing.module.ts** – Configuración de rutas de la aplicación.
- ◊ **app.component.css** – Estilos del componente principal de la aplicación.
- ◊ **app.component.html** – Plantilla HTML del componente principal.
- ◊ **app.component.ts** – Lógica del componente principal.
- ◊ **app.module.ts** – Módulo principal de la aplicación.
- **assets/** – Recursos estáticos de la aplicación (Imágenes).
 - ◊ **.redirects** – Configuración de redirecciones para Netlify.
 - ◊ **index.html** – Página principal de la aplicación.
 - ◊ **main.ts** – Punto de entrada principal de la aplicación.
 - ◊ **styles.css** – Estilos globales de la aplicación.
 - ◊ **.editorconfig** – Configuración del editor.
- **.gitignore** – Archivos y directorios a ignorar por Git.
- **angular.json** – Configuración del proyecto Angular.
- **package-lock.json** – Descripción detallada de las dependencias instaladas.
- **package.json** – Configuración y dependencias del proyecto.
- **README.md** – Documentación del proyecto.
- **tsconfig.app.json** – Configuración de TypeScript para la aplicación.
- **tsconfig.json** – Configuración global de TypeScript.
- **tsconfig.spec.json** – Configuración de TypeScript para pruebas.

Estructura de directorios del *backend*

■ API PYTHON/

- **app/** – Directorio principal del código fuente de la aplicación.
 - **Funciones_Auxiliares/** – Funciones auxiliares utilizadas en la aplicación.
 - ◊ `automatizar_correo.py` – Script para automatizar el envío de correos.
 - ◊ `funciones_webscraping.py` – Funciones para realizar web scraping.
 - **modelos/** – Definición de modelos de datos.
 - ◊ `__init__.py` – Inicialización del paquete de modelos.
 - ◊ `modelos.py` – Definición de los modelos de la base de datos.
 - **modulos/** – Módulos de la aplicación, organizados por funcionalidad.
 - ◊ **botones/** – Módulo con la funcionalidad de botones y permisos.
 - ◊ **colaboradores/** – Módulo con la gestión de colaboradores.
 - ◊ **gestionEstimacion/** – Módulo con la gestión de estimaciones.
 - ◊ **gestionLibros/** – Módulo con la gestión de libros.
 - ◊ **importacionExportacion/** – Módulo con la importación y exportación del catálogo.
 - ◊ **roles/** – Módulo con la gestión de roles de usuarios.
 - ◊ **sugerencias/** – Módulo con la gestión de sugerencias.
 - ◊ **usuarios/** – Módulo con la gestión de usuarios.
 - ◊ `__init__.py` – Inicialización del paquete de módulos.
 - **config.py** – Archivo de configuración de la aplicación.
- **instance/** – Directorio de la base de datos local.
- **tests/** – Directorio que contiene las pruebas unitarias.
- **requirements.txt** – Archivo que lista las dependencias del proyecto.
- **run.py** – Script para iniciar la aplicación.

D.3. Manual del programador

Este manual tiene como objetivo formar a todo aquel que desee realizar desarrollos o modificaciones a este proyecto. En él se explican las instalaciones necesarias para generar el entorno de desarrollo, obtener el código, y una guía para ejecutarlo.

Instalaciones necesarias

Para este proyecto se requieren realizar las siguientes instalaciones:

- Visual Studio Code
- Git
- NodeJS 20.11.1
- Python 3.12.1

Visual Studio Code

Este a sido el editor de código elegido para realizar este proyecto, por lo que es recomendable usarle por su facilidad de configuración y variedad de extensiones útiles. Para instalar este editor de código debemos dirigirnos a la página oficial [Visual Studio Code](#) y descargarnos la última versión estable.

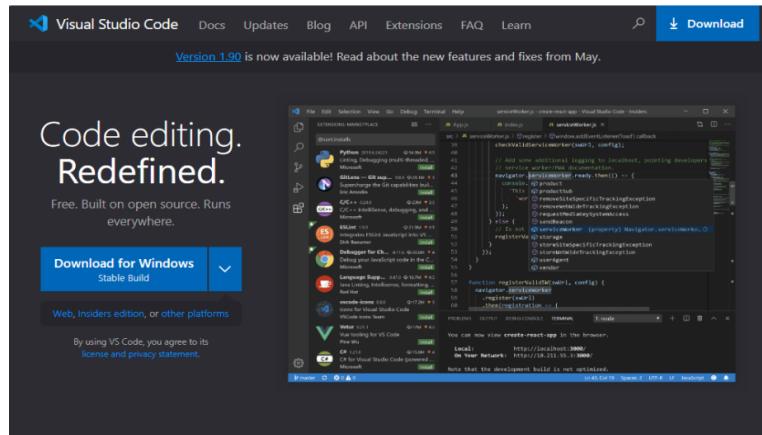


Figura D.1: Página de instalación de Visual Studio Code

Una vez se nos descarga el instalador se mantienen las opciones por defecto y se procede a la instalación. Una vez instalado es recomendable dirigirse al apartado de extensiones e instalar las siguientes:

- Angular Language Service
- Angular 17 Snippets
- Angular Schematics
- angular2-inline
- TypeScript Importer
- Python (Automáticamente instala las dos siguientes)
- Pylance
- Python Debugger

Además de esas, existen extensiones opcionales pero muy recomendables:

- Error Lens
- Better Comments
- Material Icon Theme
- Auto Close Tag
- Activitus Bar

Si todos estos pasos se han realizado correctamente, deberíamos tener el Visual Studio iniciado correctamente y 13 extensiones instaladas.

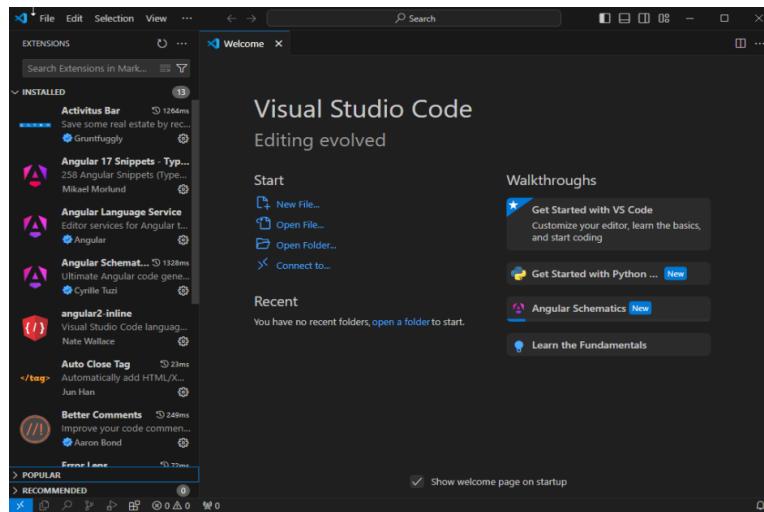


Figura D.2: Extensiones instaladas Visual Studio Code

Python

Este lenguaje es el usado para el *backend* con la versión 3.12.1, por lo que para instalar esta versión debemos dirigirnos a la web oficial de Python y seleccionar esa versión ([Versión 3.12.1 de python](#)). En la parte de abajo de esta página se encuentran todas las opciones de instalación, en este caso se va a escoger el instalador de Windows 64x ya que es el sistema operativo utilizado. Una vez abierto el instalador, es recomendable activar la checkbox que agrega python.exe al PATH, ya que si no se encuentra activo puede acarrear problemas en el futuro. Una vez seleccionamos esa opción, pulsamos en instalar ahora para que proceda con la instalación.

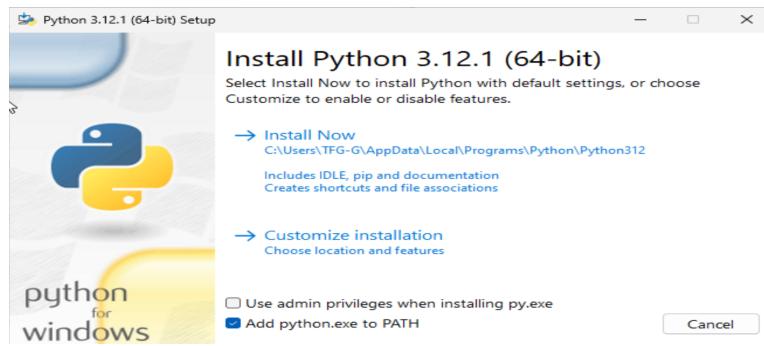


Figura D.3: Instalador de Python

NodeJS

NodeJS es el encargado de compilar el lenguaje TypeScript utilizado en Angular así como el administrados de dependencias del proyecto en la parte *frontend*. En Prehistoria en Igualdad se ha utilizado la versión 20.11.1, la cual se instala desde esta URL [Versión 20.11.1 de node](#)

Ya dentro de la web de NodeJS bajamos en la página y encontramos la zona de instaladores. En este caso se instala la versión de Windows 64 bit en formato instalador.

Una vez abierto el instalador, se va dando al botón de siguiente aceptando las condiciones y el resto de campos se dejan por defecto. Si tras la instalación, no sale en el instalador que se ha instalado correctamente, no es necesario realizar más pasos.

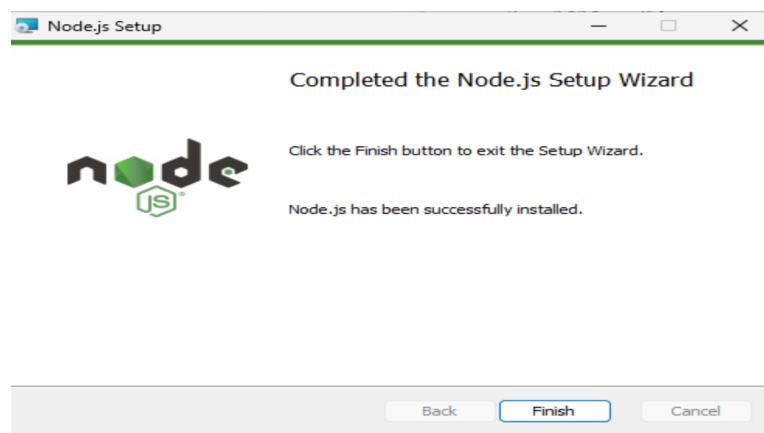


Figura D.4: Instalación existosa de NodeJS

Git

Git es una herramienta de control de versiones que se ha utilizado en este proyecto para ir actualizando los cambios en GitHub y que en este manual se va a utilizar para obtener el proyecto del repositorio para poder ejecutarlo desde local. La instalación de Git comienza en la [página oficial](#), donde descargaremos su instalador con la versión más reciente en la versión *Standalone*.

Una vez descargado el instalador, nos pedirá por permisos de administración el poder realizar cambios. Una vez aceptado, se dejan todas las opciones por defecto y se da al botón de continuar para que se ejecute la instalación.

Si se ha instalado correctamente, al cerrar el instalador nos dará la opción de ver las últimas modificaciones de esta versión.

Obtención del código fuente y Ejecución del código

Para obtener el código del proyecto existen varias vías, pero en este caso se va a utilizar la herramienta ya instalada Git.

Con los comandos que se muestran a continuación se podrá descargar de manera rápida todos los ficheros.

- Abrimos una terminal de Git en el directorio en el que queramos guardar el proyecto.

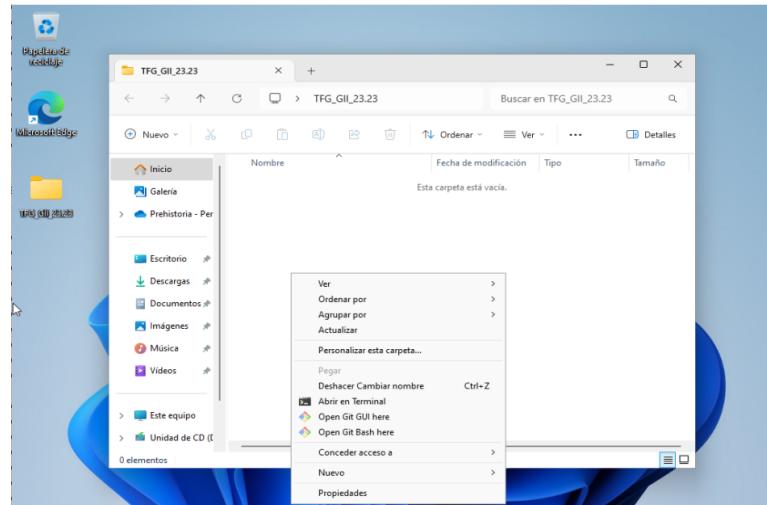


Figura D.5: Apertura de terminal Git

- Introducimos los siguientes comandos con la información personal:
 1. git config --global user.name "Tu Nombre"
 2. git config --global user.email "tuemail@example.com"

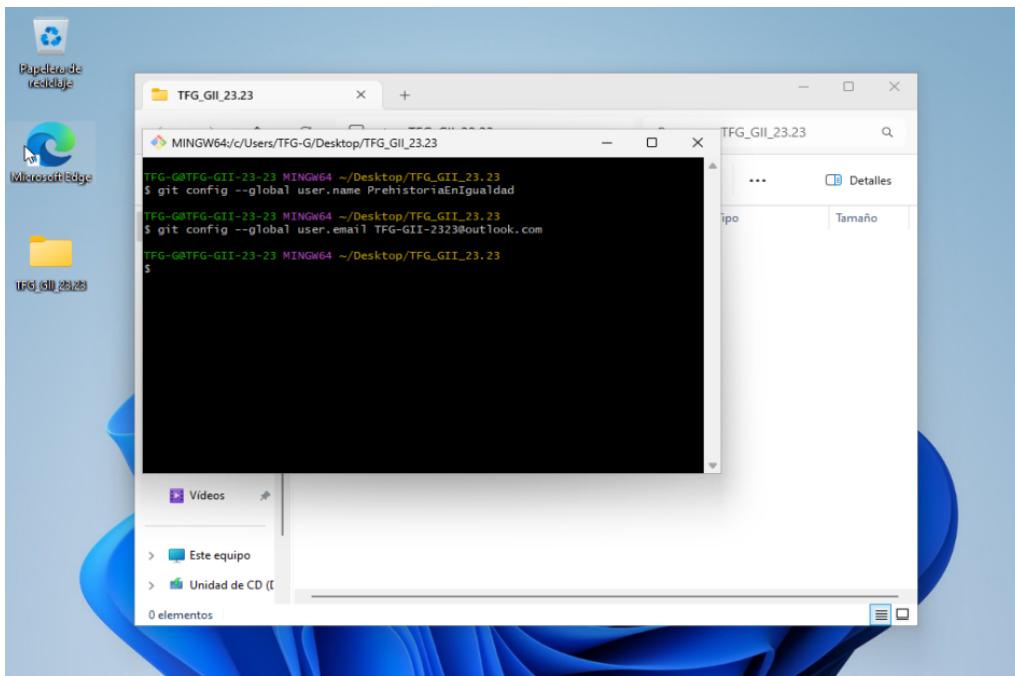


Figura D.6: Comandos de Usuario y correo de Git

- Ejecutamos el comando que clona el repositorio en local.

1. git clone https://github.com/DanielFernandezFdez/TFG_GII_23.23.git

```
MINGW64:/c/Users/TFG-G/Desktop/TFG_GII_23.23
$ git config --global user.name PrehistoriaEnIgualdad
$ git config --global user.email TFG-GII-2323@outlook.com
$ git clone https://github.com/DanielFernandezFdez/TFG_GII_23.23.git
Cloning into 'TFG_GII_23.23'...
remote: Enumerating objects: 2150, done.
remote: Counting objects: 100% (446/446), done.
remote: Compressing objects: 100% (294/294), done.
remote: Total 2150 (delta 207), reused 317 (delta 143), pack-reused 1704
Receiving objects: 100% (2150/2150), 36.02 MiB | 22.49 MiB/s, done.
Resolving deltas: 100% (1293/1293), done.
Updating files: 100% (276/276), done.
$
```

Figura D.7: Clonación del repositorio

Una vez importado correctamente, desde visual le damos a abrir carpeta y seleccionamos el proyecto que acabamos de obtener.

Antes de comenzar con la ejecución, han de realizarse unos pasos previos para finalizar la instalación de dependencias.

- Instalación de dependencias del *backend*

1. Dentro de la carpeta 'API PYTHON' ejecutar el siguiente comando para generar un entorno virtual:

```
python -m venv [Nombre del entorno]
```

Si es satisfactoria la creación, dentro de la carpeta API PYTHON se generará una carpeta con el nombre dado.

2. Activamos el entorno virtual para cargar las dependencias:

```
[Nombre del entorno virtual]\Scripts\Activate
```

Si se ha ejecutado correctamente nos tendría que aparecer el nombre en la línea de comandos entre paréntesis. En el caso de la figura D.8 se puede observar en la parte izquierda en color verde el nombre EntornoVirtual.



Figura D.8: Activar entorno virtual

3. Con el entorno virtual activo volcamos las dependencias del archivo requirements.txt con el comando:

```
py -m pip install -r requirements.txt
```

4. Con todos estos pasos realizados, con ejecutar el comando *python run.py* se iniciaría el backend. Es importante recordar que hay que activar el entorno virtual antes de iniciar la API.

- Instalación de dependencias del *frontend*

1. Situarnos en la carpeta frontendAngular con la consola.

2. Ejecutar el comando *npm install* para instalar las dependencias y el framework Angular.

3. Una vez terminada la descarga, podemos arrancar el frontend con el comando *npm start*

Cambiar la configuración de producción a local

- En la parte del *backend*, en el fichero config.py cambiar la variable db_path a la comentada.
- En la parte del *frontend*, en los ficheros service, cambiar la variable apiURL a la comentada.

D.4. Pruebas del sistema

En esta última sección se describen las pruebas disponibles para comprobar si los cambios han generado resultados inesperados, y así poder solucionarlos antes de realizar el despliegue.

Ejecutar

1. Abre una terminal o línea de comandos en el Visual Studio Code.
2. Navega al directorio de la carpeta `tests`.
3. Ejecuta el siguiente comando para ejecutar todas las pruebas unitarias:
`pytest [nombre del test]`
Este comando ejecutará todas las pruebas definidas en el archivo y generará un reporte con los resultados. Si se pone simplemente `pytest ./` y ejecutará todos los test de la carpeta.

Resultados de las Pruebas

Después de ejecutar las pruebas, `pytest` generará un reporte en la terminal mostrando qué pruebas pasaron y cuáles fallaron. Si alguna prueba falla, `pytest` proporcionará detalles sobre la causa del fallo, permitiéndote identificar y corregir el problema.

Pruebas Manuales con Postman

Además de las pruebas unitarias, se recomienda realizar pruebas manuales utilizando Postman para verificar el comportamiento de las APIs del sistema. Postman es una herramienta potente para probar y documentar APIs. A continuación se describe cómo utilizar Postman para realizar estas pruebas.

Configuración de Postman

Si aún no tienes Postman instalado, puedes descargarlo e instalarlo desde [Postman](#). Una vez instalado, sigue estos pasos:

1. Abrir Postman.
2. Agregar una nueva solicitud a la búsqueda con la URL local. Configurar el tipo de solicitud (GET, POST, PUT, DELETE).
3. Configurar los parámetros necesarios, como headers y cuerpo en formato JSON.
4. Envíar la solicitud y verificar la respuesta. Postman mostrará el código de estado HTTP y el cuerpo de la respuesta recibida del servidor.

Validación de Respuestas

Para cada solicitud, hay que asegurarse que la respuesta recibida es la esperada. Esto incluye verificar:

- El código de estado HTTP (por ejemplo, 200 OK, 404 Not Found, 500 Internal Server Error).
- El contenido del cuerpo de la respuesta, asegurando que los datos devueltos sean correctos y estén en el formato esperado.
- Los headers de la respuesta, si es necesario.

Apéndice E

Documentación de uso

E.1. Introducción

En este manual se detallan los requisitos de la aplicación web para poder utilizarla correctamente así como su manual completo de uso de la aplicación dividido en dos partes. La primera de ellas muestra las opciones disponibles para una persona que no tenga cuenta en la aplicación, y la segunda muestra toda la parte que puede utilizar el personal de administración.

E.2. Requisitos de uso

Los requisitos mínimos de la aplicación web son los siguientes:

- Conexión a internet
- Un navegador web para entrar a la siguiente URL: **Prehistoria en igualdad**

E.3. Instalación

En este caso no se necesita realizar ningún tipo de instalación, ya que en todos los dispositivos ya existe por defecto un navegador con el que poder entrar a la aplicación web.

E.4. Manual del usuario/a sin cuenta

En este apartado se recogen todas las acciones que puede realizar un usuario que no se encuentre registrado en la aplicación web.

Consultar en el catálogo un libro

Una de las funciones principales de esta aplicación es mostrar información a docentes y familias acerca de los libros analizados. Existen dos opciones para realizar esta operación.

1. Desde el inicio.

- Rellenar el campo de búsqueda con un ISBN o con un Título.
- Automáticamente se redirige al catálogo con la búsqueda completada mostrando el libro deseado al hacer click en el botón de buscar.

2. Realizando la búsqueda desde el catálogo.

- Hacer click en el menú superior en el botón catálogo
- Rellenar el campo de búsqueda con ISBN o Título.
- Hacer click en la lupa para realizar la búsqueda.
- Se mostrarán los libros que coincidan con los parámetros de la búsqueda

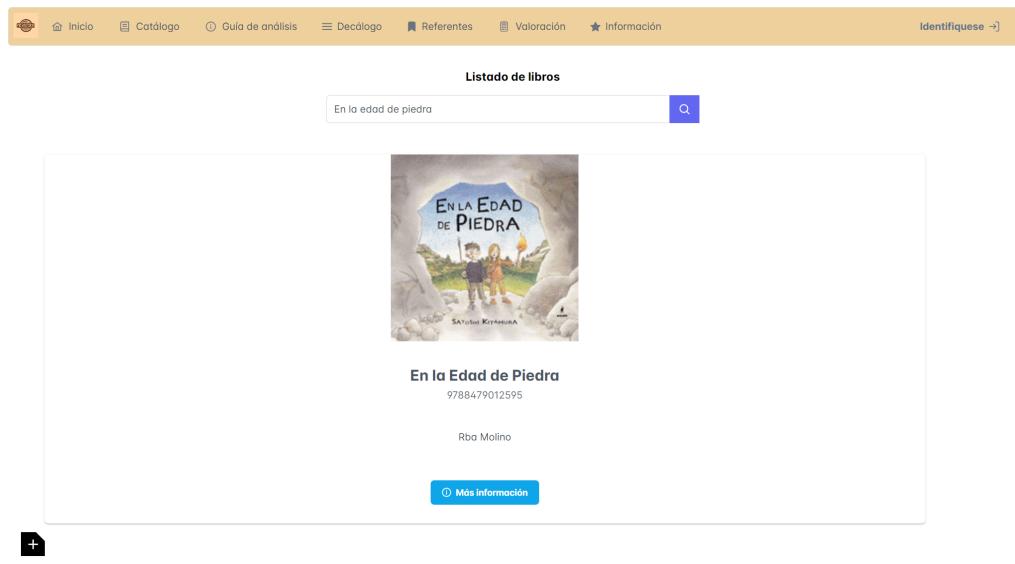


Figura E.1: Búsqueda de libro en catálogo

Consultar la información de un libro

Si un usuario/a conoce un libro sobre el que quiere consultar toda su información, se han de realizar los siguientes pasos.

- Buscar un libro desde inicio o desde el catálogo.
- Seleccionar el botón de más información del libro deseado.
- Hacer click en el botón de más información correspondiente a ese libro

Las acciones que se pueden realizar dentro de este apartado son los siguientes:

1. Si se hace click en el símbolo de información, la web cargará la pestaña de la guía de análisis.
2. El apartado de la ubicación del estudio, si contiene información, es un hipervínculo que lleva a la dirección del estudio de ese libro

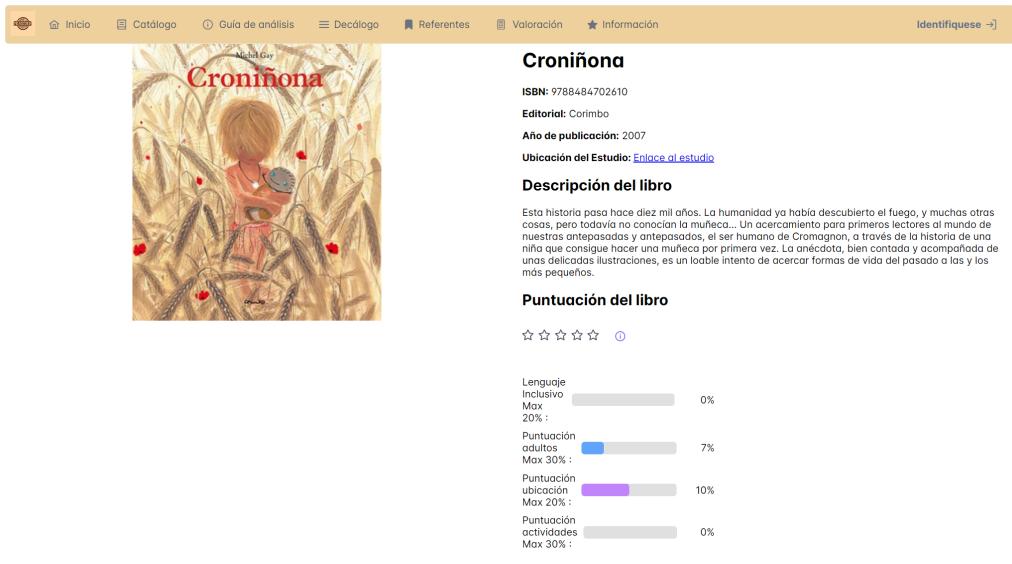


Figura E.2: Más información de un libro

Sugerir un libro

Prehistoria en Igualdad permite a los usuarios realizar recomendaciones al personal de administración para que agreguen nuevos libros al catálogo que puedan ser interesantes de cara a los objetivos de esta aplicación.

Para realizar una sugerencia de un libro:

- Entrar en el apartado de catálogo
- Pulsar el botón con el ícono de un '+' situado abajo a la izquierda.
- Rellenar el formulario con los campos solicitados y enviar.

The image shows a user interface for suggesting a book. At the top, the title "Sugerir libro" is displayed. Below it are five input fields: "Nombre" (Name), "Apellidos" (Last name), "Correo" (Email), "Título" (Title), and "ISBN". A purple "OK" button is located at the bottom of the form.

Figura E.3: Formulario de sugerencia de libro

Realizar una valoración

Otra de las partes fundamentales de esta aplicación web es la posibilidad que tienen los docentes y familias de realizar sus propias valoraciones de los libros que deseen. Los pasos para realizarla son los siguientes:

- Desde el menú acceder a la pestaña de valoración
- Rellenar los campos existentes utilizando como base la guía de análisis
- Pulsar el botón de calcular.
- Una vez se realiza el cálculo, aparecerá una modal con el resultado del cálculo y 3 opciones, navegar a la guía de análisis, no guardar la estimación, o guardarla



Figura E.4: Resultado de valoración

- En el caso de que se guarde la estimación, aparecerá un formulario para llenar con los datos personales (opcionales) por si es necesario ponerse en contacto con la persona que la ha realizado.

The screenshot shows a data entry form titled 'Ingrese sus datos'. A note below states: 'Solo es obligatorio el título y el ISBN. Puede enviar los datos sin identificarse.' The form contains six input fields: 'Nombre', 'Apellidos', 'Título', 'ISBN', 'Correo electrónico', and 'Institución'. At the bottom is a purple 'OK' button.

Figura E.5: Guardar valoración

E.5. Manual del usuario/a con cuenta y permisos de administración

Antes de continuar con los siguientes apartados, hay que tener en cuenta que todas las operaciones que se realizan por un usuario sin cuenta las puede realizar un usuario con cuenta, por lo que el manual de la sección anterior es válido también para usuarios y usuarias autenticadas. Además de esta aplicación web, está disponible una cuenta de correo electrónico donde se reciben todas las sugerencias realizadas por los usuarios.

Iniciar sesión

Para poder acceder a la zona de administración para realizar todas las operaciones de gestión, han de seguirse los siguientes pasos:

- Se selecciona en el menú superior el botón Identifíquese.
- Se rellena el campo de usuario y de contraseña para iniciar sesión.

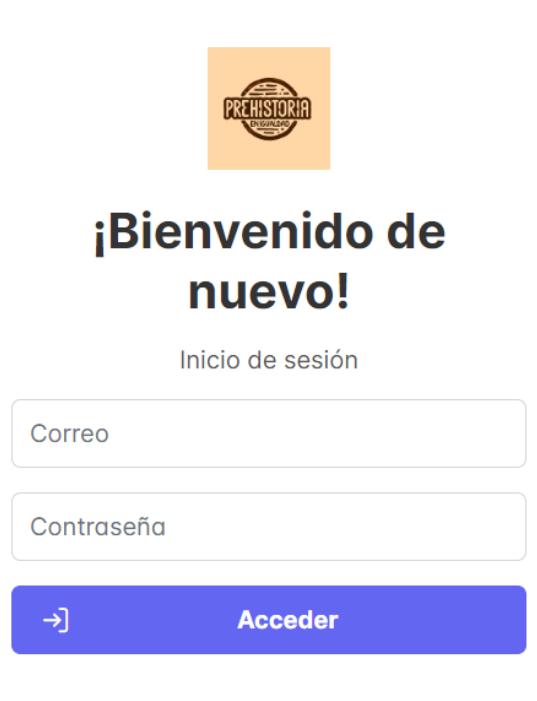


Figura E.6: Manual de Inicio de sesión

- Si se ha iniciado sesión correctamente se redirigirá al catálogo, y en el menú superior, donde se encontraba el botón de identifíquese aparecerá el nombre del usuario y un desplegable que permite acceder al panel de administración y cerrar sesión.

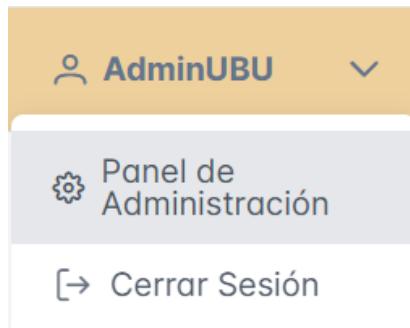


Figura E.7: Cambio de botón Inicio de sesión

Consultar estadísticas de la aplicación web

Actualmente todo el equipo de administración tiene a su disposición unas métricas mensuales que muestran distintos elementos informativos para consultar de manera gráfica el estado de la aplicación. A este apartado se accede desde el botón del panel de administrador.

La información que muestran estas estadísticas es la siguiente:

- El primer gráfico muestra el número de visitas totales que se ha tenido este mes entre todos los libros y las visitas del libro más popular.
- En el segundo gráfico se puede observar el número de usuarios/as registradas en la aplicación.
- El tercer gráfico muestra con gráficos de líneas el número de libros existentes, sugerencias enviadas y estimaciones realizadas cada mes.
- El cuarto elemento es una tarjeta donde se indica cuál ha sido el libro más popular.

Cabe destacar que existe la opción de filtrar los datos para acotar los datos en una franja temporal determinada. Por motivos de diseño el máximo de meses a mostrar son 12.



Figura E.8: Estadísticas de administración

Gestión de personas usuarias

Este apartado es el encargado de dar todas las herramientas que necesite el equipo de administración para tener un control total de quién puede formar parte o no de ese equipo.

Los pasos para realizar una operación en este apartado son los siguientes:

- Acceder al panel de administración desde el menú superior.
- Desplegar el menú lateral disponible en el menú de administración y seleccionar 'gestión de personas'.
- La aplicación muestra las cuentas existentes.
- Si se desea crear un nuevo usuario/a:
 1. Hacer click en el botón de nuevo usuario.
 2. Rellenar el formulario que se ha desplegado con un nombre, correo y el rol deseado. Si no se selecciona rol, se establece uno predefinido.
 3. Si se ha creado correctamente, aparecerá una modal indicando la contraseña temporal de esa cuenta.

- Si se desea modificar un usuario/a existente:
 1. Hacer click en el lápiz correspondiente a ese usuario/a.
 2. Modificar el formulario que se ha desplegado autocompletado con el nombre, correo y el rol asignado.
 3. Si se ha modificado correctamente, aparecerá una modal indicando éxito en la operación.
- Si se desea eliminar un usuario/a:
 1. Hacer click en la papelera correspondiente a ese usuario/a.
 2. Confirmar en la modal la operación por seguridad.
 3. Si se ha eliminado correctamente, aparecerá una modal indicando éxito en la operación.

Usuario	Correo	Rol
Administrador	admin@admin.com	Admin
Ana	asmamolar@ubu.es	Admin
testbas	valeparatodo@gmail.com	UsuarioBase
AdminUBU	adminUBU@ubu.es	Admin

Figura E.9: Operaciones en gestión de usuarios/as

Gestión de roles

Para poder gestionar la autorización de acceso a ciertas funcionalidades de la aplicación, se ha diseñado un sistema de permisos utilizando roles que limitan estas acciones de manera dinámica.

- Acceder al panel de administración desde el menú superior.
- Desplegar el menú lateral disponible en el menú de administración y seleccionar 'gestión de roles'.
- La aplicación muestra los roles existentes.

- Si se desea crear un nuevo rol:
 1. Hacer click en el botón de nuevo usuario.
 2. Introducir el nombre deseado.
 3. Si se ha creado correctamente, aparecerá una modal el éxito de la operación.
- Si se desea modificar un rol existente:
 1. Hacer click en el lápiz correspondiente a ese rol.
 2. Introducir el nuevo nombre de rol deseado.
 3. Si se ha modificado correctamente, aparecerá una modal indicando éxito en la operación.
- Si se desea eliminar un rol:
 1. Hacer click en la papelera correspondiente a ese rol.
 2. Confirmar en la modal la operación por seguridad.
 3. Si se ha eliminado correctamente, aparecerá una modal indicando éxito en la operación.

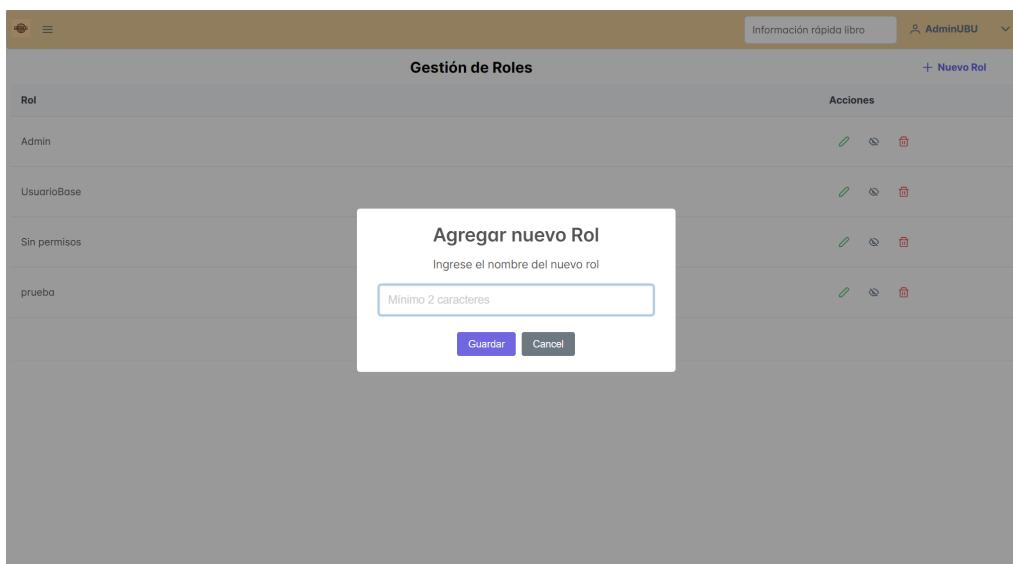


Figura E.10: Operación agregar rol en gestión de roles

- Si se desean gestionar los permisos correspondientes a un rol, ha de seleccionarse el icono del ojo en el rol deseado.
- Activar y desactivar los permisos asociados a ese rol y guardar cambios.

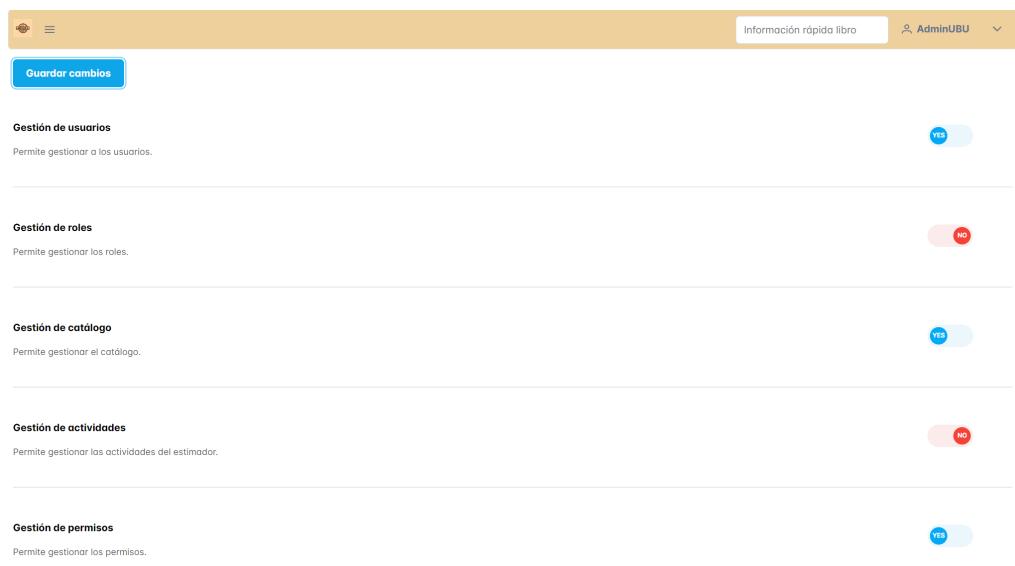


Figura E.11: Permisos de un rol

Gestión de colaboradores

Los colaboradores de esta aplicación web son todos aquellos usuarios/as que han realizado una valoración de un libro la cuál aporte datos objetivos y se considere que ese libro se puede incluir en el catálogo. Para gestionar los colaboradores existentes se han de realizar los siguientes pasos.

- Acceder al panel de administración desde el menú superior.
- Desplegar el menú lateral ubicado en el menú de administración y seleccionamos 'gestión de colaboradores'.
- La aplicación muestra los colaboradores existentes.
- Si se desea agregar un nuevo colaborador:
 1. Hacer click en el botón Nuevo colaborador.

2. Rellenar el desplegable con los campos pedidos.
 3. Guardar los cambios.
- Si se desea editar un colaborador:
 1. Hacer click en el lápiz correspondiente al colaborador.
 2. Modificar el desplegable con los campos deseados.
 3. Guardar los cambios.
 - Si se desea eliminar un colaborador:
 1. Hacer click en la papelera correspondiente.
 2. Confirmar en la modal la operación.

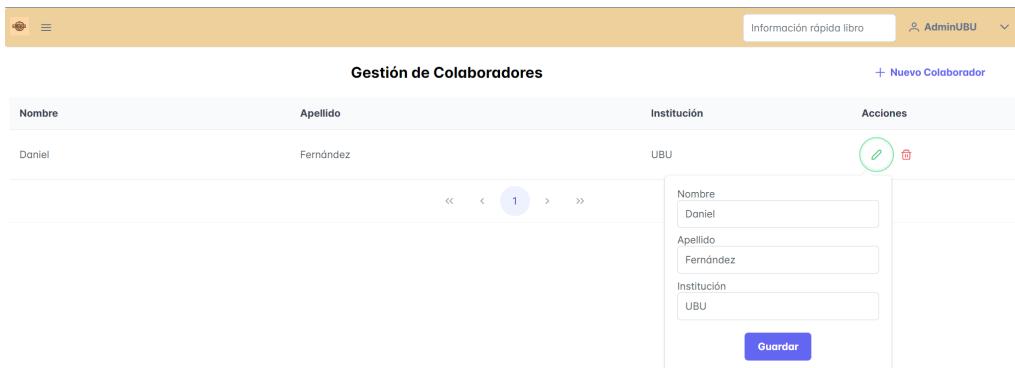


Figura E.12: Panel de gestión de colaboradores

Agregar un nuevo libro

Una de las piezas fundamentales de esta aplicación es el catálogo y todos sus elementos de gestión que lo rodean. Este apartado va dirigido a explicar las 2 maneras existentes en la aplicación de añadir libros en el catálogo.

The screenshot shows a web-based application interface for adding a new book. At the top, there's a navigation bar with icons for search, refresh, and menu, followed by links for 'Información rápida libro' and 'AdminUBU'. The main title 'Agregar Nuevo Libro' is centered above a form. The form consists of several input fields: 'Título', 'ISBN', 'Editorial', 'Descripción', 'Año de Publicación' (with a star rating scale below it), and 'Ubicación en el Estudio'. Below these are three rating scales: 'Puntuación Masculino Genérico' (scale from 0 to 5), 'Puntuación Menores' (scale from 0 to 5), and 'Puntuación Adultos' (scale from 0 to 5). The entire interface has a light orange header and a white body.

Figura E.13: Agregar libro

Agregar un libro manualmente

La forma de agregar un libro manualmente es agregando todos los campos de la información del libro a mano, por lo que se tiene que buscar toda la información. Los pasos son los siguientes:

- Acceder al panel de administración desde el menú superior.
- Desplegar el menú lateral ubicado en el menú de administración y seleccionamos la opción 'agregar libro'.
- La aplicación muestra un formulario completo con todos los datos que han de incluirse para que se reflejen en el catálogo.
- Se llenan todos los campos (Si ubicación del estudio se queda vacío el programa lo va a sustituir por no disponible).
- Si se guarda correctamente la aplicación mostrará una modal indicándolo

Agregar un libro automáticamente

Esta segunda opción ahorra al equipo de administradores el tiempo de encontrar todos los datos utilizando web scraping y la cantidad de datos a llenar es mucho menor. Los pasos son los siguientes:

- Acceder al panel de administración desde el menú superior.
- Desplegar el menú lateral ubicado en el menú de administración y seleccionamos 'agregar libro automáticamente'.
- La aplicación muestra una modal donde enviar un título o un ISBN.

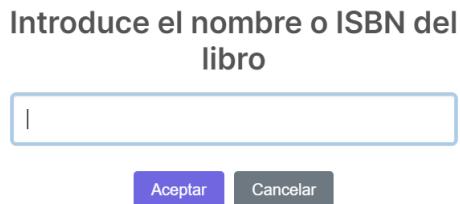


Figura E.14: Modal para agregar automáticamente

- Al enviar esa solicitud aparece una pantalla de carga mientras se buscan posibles libros.
- Al terminar la carga aparecerá la información más parecida a la petición realizada en base a 3 fuentes. Abajo del todo aparecerá un botón para ir a agregar un libro manualmente, un botón por cada una de las fuentes para escoger la información de la misma, o un apartado que permite combinar la información de las tres fuentes.

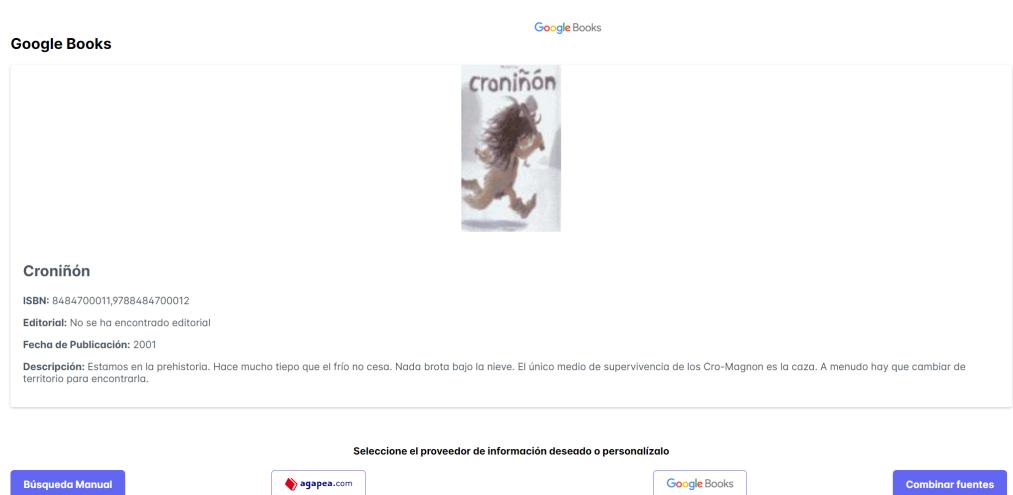


Figura E.15: Selección de fuente de web scraping

- Si se desea agregar un nuevo libro con fuente exclusiva:
 1. Se selecciona el botón de esa fuente.
 2. La aplicación se redirigirá al formulario de agregar libro pero con los campos obtenidos autocompletados.
 3. Guardar los cambios.
- Si se desea agregar un nuevo libro con fuentes combinadas:
 1. Hacer click en botón de combinar fuentes.
 2. Aparecerá un formulario con los mismos campos que agregar libro pero con desplegables, conteniendo la información de cada fuente, para así seleccionar en cada campo la deseada.
 3. Guardar los cambios.

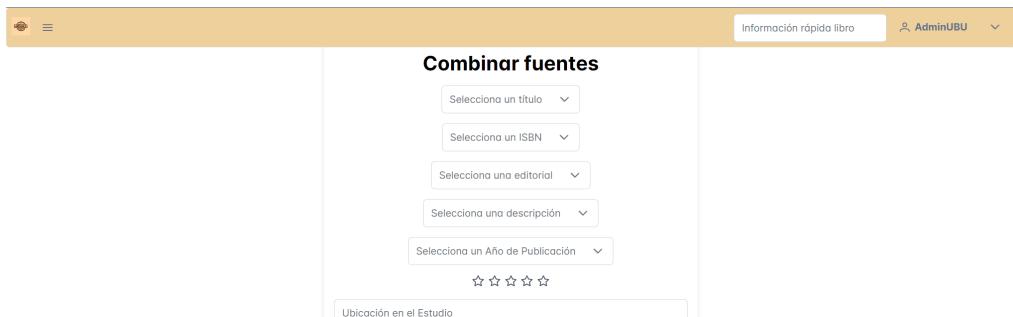


Figura E.16: Agregar libro combinando fuentes

Gestionar Catálogo

Además de la herramienta de agregar, el catálogo tiene opciones de editar y eliminar, los pasos para realizar estas operaciones son las siguientes:

- Acceder al panel de administración desde el menú superior.
- Desplegar el menú lateral ubicado en el menú de administración y seleccionamos 'gestión de catálogo'.
- Si se desea editar un libro:
 1. Hacer click en el lápiz correspondiente al libro.
 2. Modificar el libro con los campos deseados.
 3. Guardar los cambios.
- Si se desea eliminar un libro:
 1. Hacer click en la papelera correspondiente.
 2. Confirmar en la modal la operación.

Finalmente, existe un ícono de información en cada libro que redirige a la información del libro seleccionado.

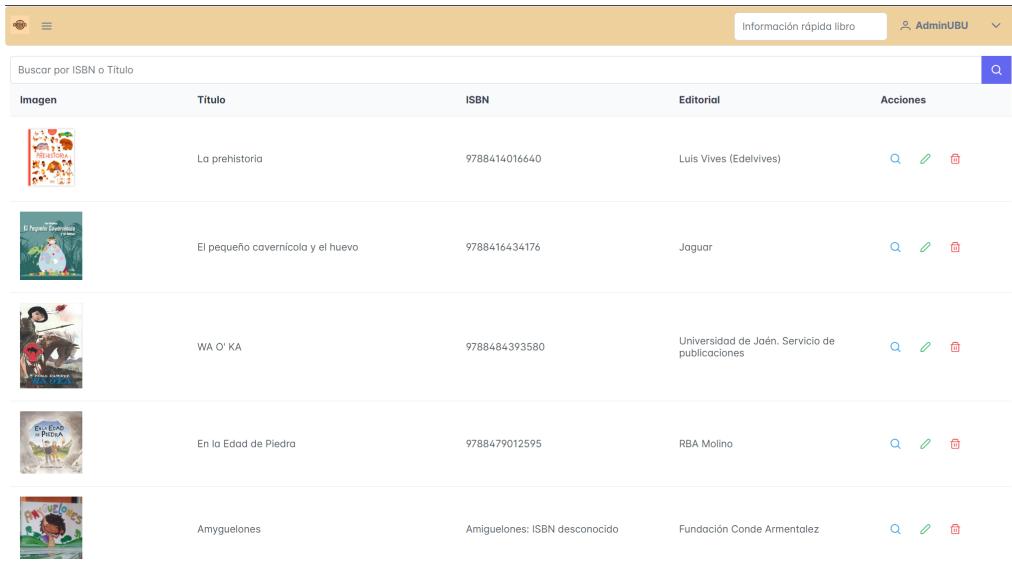


Imagen	Título	ISBN	Editorial	Acciones
	La prehistoria	9788414016640	Luis Vives (Edelvives)	
	El pequeño cavernícola y el huevo	9788416434176	Jaguar	
	WA O' KA	97888484393580	Universidad de Jaén. Servicio de publicaciones	
	En la Edad de Piedra	9788479012595	RBA Molino	
	Amyguelones	Amiguelones: ISBN desconocido	Fundación Conde Armentalez	

Figura E.17: Pantalla gestión del catálogo

Importar y exportar el catálogo

Para facilitar las labores de agregar libros ya analizados y poder mantener copias de seguridad, se ha facilitado una característica de la aplicación web que permite realizar estas operaciones. A continuación se detallan los pasos:

- Acceder al panel de administración desde el menú superior.
- Desplegar el menú lateral ubicado en el menú de administración y seleccionamos 'importación o exportación'.
- Si se desea importar:
 1. Se selecciona el archivo a cargar (Tiene que ser xlsx o csv con separador ;).
- Si se desea exportar:
 1. Seleccionar en la modal la extensión con la que se desea exportar.

Es importante mencionar que los archivos a cargar han de incluir todas las siguientes columnas (separadas por ;):

- ID

- Título
- ISBN
- Editorial
- Descripción
- Año de publicación
- Puntuación
- Ubicación del estudio
- URL de la imagen
- Visitas mensuales
- Visitas totales
- Mes de creación
- Año de creación
- Puntuación Lenguaje genérico
- Puntuación Menores
- Puntuación Adultos
- Puntuación Ubicación
- Puntuación Actividades

Es importante destacar que para que la importación/exportación funcione correctamente **el ISBN debe de ir entre comillas dobles y el campo ID debe de ser único.**

Gestión de actividades

En este apartado se permite al equipo de administración modificar según necesiten las actividades mostradas al generar una valoración. Los pasos son los siguientes:

- Acceder al panel de administración desde el menú superior.

- Desplegar el menú lateral disponible en el menú de administración y seleccionar 'gestión de actividades'.
- La aplicación muestra las actividades existentes.
- Si se desea crear una nueva actividad:
 1. Hacer click en el botón de nueva actividad.
 2. Introducir el nombre deseado y las categorías deseadas.
 3. Si se ha creado correctamente, aparecerá una modal el éxito de la operación.
- Si se desea modificar una actividad existente:
 1. Hacer click en el lápiz correspondiente a esa actividad.
 2. Introducir el nuevo nombre y las categorías a cambiar.
 3. Si se ha modificado correctamente, aparecerá una modal indicando éxito en la operación.
- Si se desea eliminar un rol:
 1. Hacer click en la papelera correspondiente a esa actividad.

Gestión de Actividades			
Nombre de la actividad	Categoría	Puntuación extra	Acciones
Cazar	Producción	Mujer	
Pescar	Producción	Mujer	
Producir herramientas	Producción	Mujer	
Producir bienes inmuebles	Producción	Mujer	

Figura E.18: Gestión de actividades

Gestionar valoraciones realizadas

Esta característica de la aplicación permite al equipo de administración observar y eliminar todas las valoraciones que han hecho los usuarios/as.

- Acceder al panel de administración desde el menú superior.
- Desplegar el menú lateral disponible en el menú de administración y seleccionar 'gestión de valoraciones'.
- La aplicación muestra las valoraciones existentes.
- Si se desea ver los datos de una valoración existente:
 1. Hacer click en el ícono de información correspondiente a esa actividad.
 2. Se muestra una modal con toda la información disponible.
- Si se desea eliminar una valoración:
 1. Hacer click en la papelera correspondiente a esa valoración.
- Si se desea descargar una valoración:
 1. Hacer click en el ícono de descargar correspondiente a esa valoración.
 2. Se abrirá una pestaña para ver la dirección de guardado del csv.

Título	ISBN	Resultado	Acciones
Prueba de evaluación	12345678	57	

Figura E.19: Listado de valoraciones guardadas

Apéndice F

Anexo de sostenibilización curricular

F.1. Introducción

Este proyecto se alinea significativamente con varios Objetivos de Desarrollo Sostenible (ODS) [2] definidos por Naciones Unidas en 2015. Se trata de unos objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo cuyas metas pretenden alcanzarse en 2030. A continuación, se muestran las formas en que este proyecto contribuye a los ODS seleccionados.

F.2. Objetivo número 4: Educación de calidad

La literatura infantil puede mediar en la educación para la igualdad de género. Este trabajo pretende facilitar la evaluación de literatura mediante el análisis de las atribuciones de género y los mecanismos de construcción identitaria en libros informativos y de ficción sobre prehistoria. Mediante la identificación de tareas desempeñadas por hombres y mujeres en el espacio social, la presencia o ausencia de rasgos conductuales referidos a la sumisión o a la dominación, conocer la frecuencia y grado de visibilidad o invisibilidad de hombres y mujeres en cada uno de los textos e imágenes, y el uso gramatical del masculino genérico para referirse a todas las personas que conforman un colectivo humano.

El proyecto ofrece un catálogo de libros infantiles, los cuales han sido evaluados en términos anteriormente comentados, lo que permite al personal docente y familias seleccionar libros que cumplan con un rigor científico acerca de los roles de género históricos o que al menos no perpetúan estereotipos de género que los estudios científicos ya han descartado desde hace tiempo. Al identificar y promover libros que representen de manera justa y precisa los roles de género, se fomenta una educación que refuerza los valores de igualdad y diversidad desde una edad temprana.

Impacto en la práctica

- Formación docente: Proporciona al personal de educación una herramienta para evaluar y escoger libros para mejorar la enseñanza utilizando bases científicas. Esto facilita la tarea al personal docente al seleccionar materiales que no perpetúan estereotipos de género
- Participación familiar: Se involucra a las familias en la elección de libros y su proceso de selección en términos de calidad.
- Actualización continua: La herramienta permite actualizaciones periódicas basadas en nuevos estudios y hallazgos, asegurando que los recursos educativos se mantengan relevantes y científicamente precisos.

F.3. Objetivo número 5: Igualdad de Género

La aplicación web creada se centra en detectar y evaluar los estereotipos de género en los libros infantiles de la prehistoria. Al realizar esta acción, contribuye a la promoción de la igualdad de género desde una edad temprana, educando a niños y niñas con ejemplos equitativos y sin sesgos.

Impacto en la práctica

- Eliminación de estereotipos: Al proporcionar análisis de los libros, esta herramienta contribuye a eliminar los estereotipos de género en la educación.
- Empoderamiento de niñas y mujeres: Este proyecto promueve una representación real y justa de las mujeres en la época de la prehistoria, lo que puede ayudar a inspirar a las niñas a interesarse por campos tradicionalmente ocupados por los hombres.

- Aumento de la sensibilización: Fomenta la concienciación sobre la importancia de la igualdad de género utilizando materiales educativos.
- Desarrollo de materiales educativos: Fomenta la creación de nuevos materiales educativos que sean inclusivos y equitativos, ayudando a reducir las brechas de género en la educación.

F.4. Objetivo número 13: Acción por el clima

Indirectamente, este proyecto contribuye a la acción por el clima al mejorar la eficiencia de la selección de los mejores materiales educativos. Al ayudar a identificar libros de calidad, tanto docentes como familias pueden escoger los mejores libros, evitando así comprar libros sin sesgos de género. Esto puede reducir el consumo de recursos y la huella de carbono asociada a su producción y distribución.

Impacto en la práctica

- Consumo responsable: Indirectamente se fomenta un consumo más responsable de recursos educativos, seleccionando en un menor número de intentos la elección del mejor libro para la educación.
- Reducción de desperdicios: Al elegir los libros correctos desde el principio, se evita el desperdicio de recursos naturales y energía utilizados en la producción de libros que no cumplen con los estándares educativos.

F.5. Conclusión

Tras todo lo mostrado anteriormente, el proyecto desarrollado no solo cumple con su objetivo principal de ayudar a mejorar la enseñanza de la prehistoria libre de perpetuar roles de género inadecuados, si no que también contribuye a 3 de los Objetivos de Desarrollo Sostenible. A través de su implementación cuidadosa, la aplicación web demuestra cómo los objetivos globales pueden ser avanzados mediante pequeñas iniciativas educativas.

Bibliografía

- [1] Modelo cliente servidor. <https://www.geeksforgeeks.org/client-server-model/>, 2024.
- [2] Objetivos de desarrollo sostenible. <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>, 2024.
- [3] Creative Commons. Cc by license. <https://creativecommons.org/licenses/by/4.0/>, 2001. [Internet; accessed 15-May-2024].
- [4] Apache Software Foundation. Apache license 2.0. <https://www.apache.org/licenses/LICENSE-2.0>, 2004. [Internet; accessed 15-May-2024].
- [5] Google. Patrón componentes. <https://angular.io/guide/architecture-components>, 2023.
- [6] Google. Patrón servicios. <https://v17.angular.io/guide/architecture-services>, 2023.
- [7] iluwatar. Patrón módulo. <https://java-design-patterns.com/es/patterns/module/>.
- [8] Microsoft. Patrón repositorio. <https://learn.microsoft.com/en-us/dotnet/architecture/microservices/microservice-ddd-cqrs-patterns/infrastructure-persistence-layer-design>, 2023.
- [9] Massachusetts Institute of Technology. Mit license. <https://opensource.org/licenses/MIT>, 1988. [Internet; accessed 15-May-2024].

- [10] Regents of the University of California. Bsd license. <https://opensource.org/licenses/BSD-3-Clause>, 1990. [Internet; accessed 15-May-2024].