



ESCUELA DE INGENIERÍA DE FUENLABRADA

**INGENIERÍA EN SISTEMAS AUDIOVISUALES Y
MULTIMEDIA**

TRABAJO FIN DE GRADO

**PLATAFORMA WEB PARA PROYECTOS DE
ARQUITECTURA**

Autor: Bayas Pancho, Karol Joseth

Tutor: Robles Martínez, Gregorio

Cotutora: De Jorge Huertas, Virginia

Curso académico 2023/2024

Trabajo Fin de Grado

Plataforma WEB para Proyectos de Arquitectura

Autor : Bayas Pancho, Karol Joseth

Tutor : Robles Martínez, Gregorio

La defensa del presente Proyecto Fin de Carrera se realizó el día _____ de _____ de 2024, siendo calificada por el siguiente tribunal:

Presidente:

Secretario:

Vocal:

y habiendo obtenido la siguiente calificación:

Calificación:

Fuenlabrada, a de de 2024

*Dedicado a
mi familia*

Agradecimientos

Con la finalización de este proyecto doy por finalizada mi etapa universitaria por lo que antetodo me gustaría presentar la gratitud que siento hacia los profesores de la EIF. A lo largo de estos años, su orientación y sabiduría me han guiado en mi camino académico. En particular, quiero expresar mi sincero agradecimiento a Gregorio Robles, quien no solo nos introdujo en el mundo del desarrollo web, sino que también creyó en mí para la realización de este proyecto, junto con Virginia de Jorge Huertas. Su constante disponibilidad para orientarme y su apoyo inquebrantable han sido invaluables.

También deseo expresar mi amor y gratitud hacia mi familia; a mis padres y hermanos que siempre han estado a mi lado. Su apoyo incondicional y constante ha sido mi roca en los momentos de dificultad. Sus palabras de aliento siempre me impulsaron a seguir adelante incluso cuando el camino era difícil. Su amor y apoyo han sido la columna vertebral de todos mis logros y por eso, estoy eternamente agradecido.

Y a todos los que de alguna manera han contribuido a este trabajo, muchas gracias. Cada palabra de aliento, cada gesto de apoyo, ha sido fundamental para llegar hasta aquí. Este logro es tan suyo como mío.

Resumen

Este proyecto es la culminación de mi grado en Ingeniería en Sistemas Audiovisuales y Multimedia, y es la construcción de una plataforma web diseñada especialmente para los amantes de la arquitectura.

Usando el framework Flask de Python, la plataforma busca potenciar la colaboración y comunicación entre los estudiantes y profesores de arquitectura. La finalidad es proporcionar un espacio en el que los usuarios, ya sean estudiantes o profesores, puedan interactuar y compartir información relevante. De esta manera, una vez realizado el registro, pueden llevarse a cabo varias tareas, como añadir eventos, libros, revistas, enlaces interesantes y mucho más.

La plataforma cuenta con un administrador que tiene acceso a todas las partes del sistema y puede manejar la base de datos. De esta manera, si algo sale mal o se requiere una modificación de algún dato, puede ser realizado por el administrador sin ningún problema.

Además la plataforma permite la carga y gestión de archivos, incluyendo imágenes, vídeos, documentos PDF, entre otros. Estos archivos están a menudo vinculados a los libros de arquitectura presentes en la plataforma, proporcionando así una valiosa fuente de conocimiento e inspiración para cualquiera que esté interesado en el tema.

Esta plataforma está pensada para ser accesible desde cualquier navegador, por lo que puedes acceder a ella desde dispositivos móviles, tabletas o cualquier otro dispositivo.

Además, diferentes tipos de usuarios son considerados, de manera que si se trata de un alumno o un profesor, se les ofrecen diferentes funcionalidades. Esto se lleva a cabo para asegurar que cada usuario tenga una experiencia adaptada a sus necesidades y pueda aprovechar al máximo la plataforma.

En resumen, este proyecto no es solo una plataforma que facilita la colaboración entre alumnos y profesores. Es también una biblioteca virtual de conocimiento arquitectónico y una potente herramienta para mejorar la experiencia educativa en el campo de la arquitectura.

Summary

This project is the culmination of my degree in Audiovisual and Multimedia Systems Engineering, and it is the construction of a web platform designed especially for architecture enthusiasts.

Using the Flask framework of Python, the platform aims to enhance collaboration and communication among architecture students and professors. The purpose is to provide a space where users, whether students or professors, can interact and share relevant information. In this way, once registered, various tasks can be carried out, such as adding events, books, magazines, interesting links, and much more.

The platform features an administrator who has access to all parts of the system and can manage the database. Thus, if something goes wrong or a modification of any data is required, it can be carried out by the administrator without any problem.

Additionally, the platform allows for the uploading and management of files, including images, videos, PDF documents, among others. These files are often linked to the architecture books present on the platform, thus providing a valuable source of knowledge and inspiration for anyone interested in the topic.

This platform is designed to be accessible from any browser, so you can access it from mobile devices, tablets, or any other device.

Furthermore, different types of users are considered, so if it's a student or a professor, different functionalities are offered to them. This is done to ensure that each user has an experience tailored to their needs and can make the most of the platform.

In summary, this project is not just a platform that facilitates collaboration between students and professors. It is also a virtual library of architectural knowledge and a powerful tool to enhance the educational experience in the field of architecture.

Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Estructura de la memoria	2
2. Objetivos	5
2.1. Objetivo general	5
2.2. Objetivos específicos	7
2.3. Planificación temporal	9
3. Estado del arte	11
3.1. Tecnologías y Herramientas	11
3.1.1. Python	11
3.1.2. GitHub	12
3.1.3. phpMyAdmin	12
3.1.4. JavaScript	12
3.1.5. Docker	13
3.1.6. LaTeX	14
3.2. Librerías	14
3.2.1. Flask	14
3.2.2. Flask-Mail	15
3.2.3. Werkzeug	15
3.2.4. mysql-connector-python	15
3.2.5. os, random, string	16
3.2.6. Decorators y Config	16

3.2.7. Flask-Login	17
3.2.8. Flask-MySQL y Flask-MySQLdb	17
3.2.9. MySQLClient y PyMySQL	17
3.2.10. FullCalendar	18
3.2.11. Itsdangerous	18
3.2.12. MarkupSafe	18
3.2.13. Flask-Mail	19
3.2.14. PIP	19
3.2.15. Jinja2	19
3.2.16. MySQL	20
3.2.17. Bootstrap	20
3.2.18. jQuery	20
3.2.19. Google Translate API	21
4. Diseño e implementación	23
4.1. Arquitectura general	23
4.2. Arquitectura específica	26
4.2.1. Programa principal	26
4.2.2. Configuración	27
4.2.3. Decoradores	28
4.2.4. Rutas del Sitio	29
4.2.5. Rutas del Administrador	30
4.2.6. Base de Datos	33
4.3. Vista de Usuario	35
4.3.1. Vista de Usuario - No Registrado	36
4.3.2. Vista de Usuario - Registrado	39
5. Experimentos y validación	43
5.1. Pruebas con la aplicación	43
5.1.1. Pruebas Funcionales	43
5.1.2. Validación de Formularios	44
5.1.3. Pruebas de Compatibilidad e Interfaz de Usuario (UI)	44

ÍNDICE GENERAL	XI
5.1.4. Pruebas de la Base de Datos	45
6. Resultados	47
6.1. Descripción de las pruebas realizadas	47
6.1.1. Realizar Login o Registro de Usuarios	47
6.1.2. Recuperación de Contraseña	48
6.1.3. Gestión de Eventos: Añadir, Eliminar y Editar	48
6.1.4. Gestión de Enlaces y Revistas: Añadir y Eliminar	52
6.1.5. Gestión de Libros: Publicar, Eliminar y Editar	52
6.1.6. Comentar en Publicaciones	54
6.1.7. Dar Permisos a Usuarios	54
6.2. Conclusiones de los resultados	56
7. Conclusiones	59
7.1. Conclusión del Proyecto	59
7.2. Aplicación de lo aprendido	60
7.3. Lecciones aprendidas	61
7.4. Trabajos futuros	62
A. Manual de usuario	63
A.1. Bienvenida	63
A.2. Registro de Usuario	63
A.3. Inicio de Sesión	63
A.4. Gestión de Archivos	64
A.5. Calendario	64
A.6. Comentarios	64
A.7. Contacto	65
Bibliografía	67

Índice de figuras

2.1. Diagrama de Gantt	9
4.1. Estructura del proyecto	24
4.2. Arquitectura del sistema	24
4.3. Permisos de los Usuarios	35
4.4. Vista de página principal	36
4.5. Vista de la página de libros	37
4.6. Vista de la página de links	37
4.7. Vista de la página de revistas	37
4.8. Vista de la página de calendario	38
4.9. Acceso de a la plataforma	41
4.10. Calendario desde admin	41
5.1. Mensajes de error	44
5.2. Alerta: Faltan datos en el formulario	44
5.3. Vista desde un dispositivo móvil	45
5.4. Base de datos	46
6.1. Correo de Registro Correcto	48
6.2. Mensaje: Esperando confirmación	49
6.3. Mensaje: Username not valid	50
6.4. Restablecer contraseña	51
6.5. Calendario desde Administrador	51
6.6. Links - Journals desde Administrador	53
6.7. Books desde la vista admin	55

6.8. Vista de Comentarios desde el perfil de un alumno	56
6.9. Vista de Permisos desde el Administrador	56

Capítulo 1

Introducción

En el terreno educativo de la arquitectura, la colaboración eficiente y la comunicación fluida están siendo convertidas en elementos indispensables. Esta necesidad ha quedado plasmada en el proyecto que es presentado hoy: una plataforma web construida con la finalidad de potenciar la interacción entre estudiantes y profesores del mundo arquitectónico.

La plataforma está diseñada para el intercambio de información relevante y para facilitar la comunicación a lo largo del calendario académico.

Promoviendo activamente la dinámica académica, este proyecto no es sólo una recopilación del conocimiento técnico adquirido durante el grado, sino que también representa un proyecto para mejorar la educación arquitectónica.

1.1. Motivación

En el trasfondo de la educación arquitectónica, la firme convicción de que la tecnología puede desempeñar un papel transformador al mejorar la colaboración y la comunicación entre estudiantes y profesores. Inspirada por mis tutores Virginia y Gregorio en el deseo de superar las barreras existentes en la interacción educativa, he decidido emprender el desarrollo de esta plataforma web.

La motivación central detrás de este trabajo de fin de grado es la necesidad de proporcionar a la comunidad educativa de arquitectura una herramienta digital que simplifique y enriquezca su experiencia de aprendizaje. Observando las limitaciones en la comunicación y el intercambio de información, se ha diseñado un espacio virtual donde la colaboración se vuelve intuitiva y

donde la información relevante fluye de manera efectiva.

Con el objetivo de fomentar la participación activa, la plataforma busca ir más allá de ser simplemente un repositorio de datos. Pretende ser un medio dinámico donde los estudiantes pueden compartir ideas, los profesores pueden proporcionar orientación y todos pueden estar al tanto de los eventos educativos cruciales. La creación de esta plataforma no solo es un ejercicio técnico, sino una contribución tangible a la mejora de la calidad educativa en el ámbito de la arquitectura.

En la creencia firme de que, al proporcionar un entorno digital eficiente y fácil de usar, esta plataforma puede marcar la diferencia, fomentando la comunicación. Esta motivación arraigada en la mejora continua y la innovación tecnológica impulsa mi dedicación a la creación de esta herramienta que aspira a enriquecer la experiencia educativa para estudiantes y profesores por igual.

1.2. Estructura de la memoria

A continuación, se presenta una descripción de la estructura de la memoria, detallando el contenido de cada uno de los capítulos para proporcionar una guía organizada del trabajo de fin de grado:

- **Capítulo 1: Introducción**

En este capítulo, se ha presentado el contexto del trabajo, se ha definido el problema de investigación y se han destacado los objetivos del estudio. Además, se ha ofrecido una breve descripción de la metodología utilizada y se ha justificado la relevancia del tema.

- **Capítulo 2: Objetivos de Estudio**

En este capítulo se proporciona información sobre los objetivos esperados del proyecto, las características que se pretende obtener y los objetivos esperados.

- **Capítulo 3: Tecnologías Utilizadas**

En este capítulo se proporciona información detallada sobre el diseño, las características y los usos de cada una de las tecnologías usadas en el proyecto.

- **Capítulo 4: Funcionamiento de la Aplicación**

En este capítulo, se entra en detalle en el funcionamiento de la aplicación.

■ Capítulo 5: Experimentos y Pruebas

Se presentan los diferentes experimentos realizados y se prueba el correcto funcionamiento de la aplicación.

■ Capítulo 6: Resultados Obtenidos

En este capítulo, se explican los resultados obtenidos al poner a prueba varios aspectos de la aplicación.

■ Capítulo 7: Conclusiones

Se indican los resultados conseguidos y la solución a los objetivos no conseguidos.

Capítulo 2

Objetivos

2.1. Objetivo general

El propósito de este proyecto es desarrollar un sistema de administración integral destinado a gestionar no solo libros y documentos, sino también eventos, revistas y enlaces de interés orientado al área de arquitectura de la EIF. El objetivo principal es mejorar significativamente la experiencia de los usuarios al proporcionar un acceso más fácil y eficiente a la información a través de una interfaz intuitiva y amigable. Para garantizar la seguridad y la privacidad de los usuarios, se implementarán funcionalidades como el inicio de sesión seguro y la opción de restablecimiento de contraseña en caso de olvido.

En esta plataforma, nos enfocamos en potenciar la interacción entre los usuarios para la gestión de eventos, la administración de libros y la exploración de revistas y enlaces relevantes. Además de estas funciones principales, hemos integrado herramientas administrativas que facilitarán a los usuarios la interacción de manera eficiente y efectiva con el sistema. Nuestro objetivo es crear una experiencia lo más amigable posible para todos los usuarios que utilicen este sistema, promoviendo una colaboración activa, una comunicación fluida y un acceso fácil a los recursos educativos y de interés en el campo de la arquitectura.

- Facilitar la Colaboración:**

Se plantea la creación de un entorno digital diseñado específicamente para fomentar la colaboración entre estudiantes y profesores. Este entorno busca simplificar y agilizar el intercambio de información y recursos educativos relevantes para la disciplina de la ar-

quitectura. Se pretende que esta plataforma proporcione un espacio donde los usuarios puedan compartir ideas, proyectos, documentos y otros materiales de forma eficiente, promoviendo así una mayor interacción y cooperación.

■ **Centralizar Recursos:**

Se plantea el desarrollo de una plataforma digital que funcione como un repositorio centralizado para almacenar una amplia variedad de recursos relacionados con la arquitectura. Esta plataforma tendría como objetivo principal mejorar tanto el acceso como la gestión de la información educativa dentro del ámbito de la arquitectura. Al centralizar estos recursos, se facilitaría a estudiantes y profesores la búsqueda y el acceso a libros, archivos y otros materiales relevantes para su formación y desarrollo académico. Además, se buscaría optimizar la organización y clasificación de estos recursos para garantizar una experiencia de usuario fluida y eficiente.

■ **Mejorar la Comunicación:**

Se propone la implementación de herramientas destinadas a mejorar la comunicación entre los usuarios de la plataforma. Esto incluye la organización de eventos y la creación de foros de discusión, con el objetivo de fomentar una interacción más activa y un intercambio de ideas más fluido entre los participantes. Estas herramientas permitirían a los usuarios compartir experiencias, plantear preguntas, discutir temas relevantes y colaborar en proyectos de manera colaborativa. El enfoque está en promover una comunicación efectiva que enriquezca la experiencia educativa de los usuarios y fortalezca la comunidad de aprendizaje en el ámbito de la arquitectura.

■ **Optimizar la Experiencia del Usuario:**

Se plantea el diseño de una interfaz de usuario que sea intuitiva y fácil de usar, adaptada a las necesidades particulares de los estudiantes y profesores del campo. El objetivo es mejorar significativamente la experiencia de navegación y participación dentro de la plataforma. Se buscaría crear un entorno digital que sea accesible para usuarios de todos los niveles de habilidad tecnológica y que ofrezca una experiencia fluida y agradable. Esto implica la implementación de un diseño claro y estructurado, la inclusión de funciones de búsqueda eficientes, así como la disposición de herramientas y recursos relevantes de manera fácilmente accesible.

En resumen, se persigue la creación de un espacio digital que potencie la interacción y el aprendizaje en el ámbito de la arquitectura mediante una experiencia de usuario óptima.

2.2. Objetivos específicos

A continuación se detallan los objetivos específicos que guiarán el desarrollo y la implementación del sistema de administración:

1. Desarrollar Vistas de Administración:

Crear interfaces administrativas intuitivas y funcionales para la gestión de archivos, eventos y usuarios, proporcionando herramientas eficientes para realizar tareas de administración.

2. Desarrollar Vistas de Página Pública:

Diseñar una interfaz intuitiva y funcional que permita el acceso a la página web para cualquier persona, incluso si no forma parte de la comunidad registrada. Esta vista pública proporcionará una experiencia de navegación informativa y atractiva, sin acceso a funcionalidades de administración reservadas para usuarios registrados.

3. Implementar Sistema de Autenticación:

Desarrollar un sistema de autenticación robusto y seguro que permita el acceso protegido a las funcionalidades administrativas, garantizando la autenticidad y la autorización de los usuarios.

4. Diseñar Páginas de Visualización:

Diseñar páginas de visualización atractivas y de fácil navegación para la presentación de libros, eventos y comentarios, asegurando una experiencia de usuario óptima para todos los usuarios.

5. Integrar Sistema de Comentarios:

Incorporar un sistema de comentarios interactivo que permita a los usuarios expresar sus opiniones y compartir comentarios sobre los libros, fomentando la participación y el intercambio de ideas.

6. Gestionar Eventos:

Implementar funciones de gestión de eventos que posibiliten la visualización, edición y organización eficiente de eventos, proporcionando información relevante y actualizada.

7. Administrar Libros:

Desarrollar funcionalidades completas para la gestión de libros, incluyendo la capacidad de agregar, editar y eliminar libros, así como la asociación adecuada con los autores correspondientes.

8. Administrar Enlaces y Revistas:

Implementar funcionalidades para la gestión de revistas y enlaces, que permitan a los usuarios agregar y eliminar tanto revistas como enlaces de interés

9. Implementar Herramientas Administrativas:

Integrar herramientas administrativas avanzadas para la gestión de usuarios, roles y permisos, permitiendo una administración eficaz y segura del sistema.

10. Optimizar la Seguridad:

Mejorar la seguridad del sistema mediante la implementación de técnicas avanzadas de almacenamiento y recuperación de contraseñas, garantizando la protección de la información confidencial.

11. Aplicar Restricciones de Acceso:

Establecer restricciones de acceso basadas en roles para controlar qué usuarios pueden acceder a determinadas funcionalidades, asegurando un uso adecuado y autorizado del sistema.

12. Garantizar la Integridad de los Datos:

Implementar medidas para asegurar la integridad de los datos almacenados en la base de datos, evitando inconsistencias y errores que puedan comprometer la fiabilidad y la precisión de la información.

Estos objetivos específicos serán clave para el desarrollo exitoso del sistema de administración, garantizando su funcionalidad, seguridad y eficiencia en la gestión de recursos y usuarios.

Cronograma del Proyecto

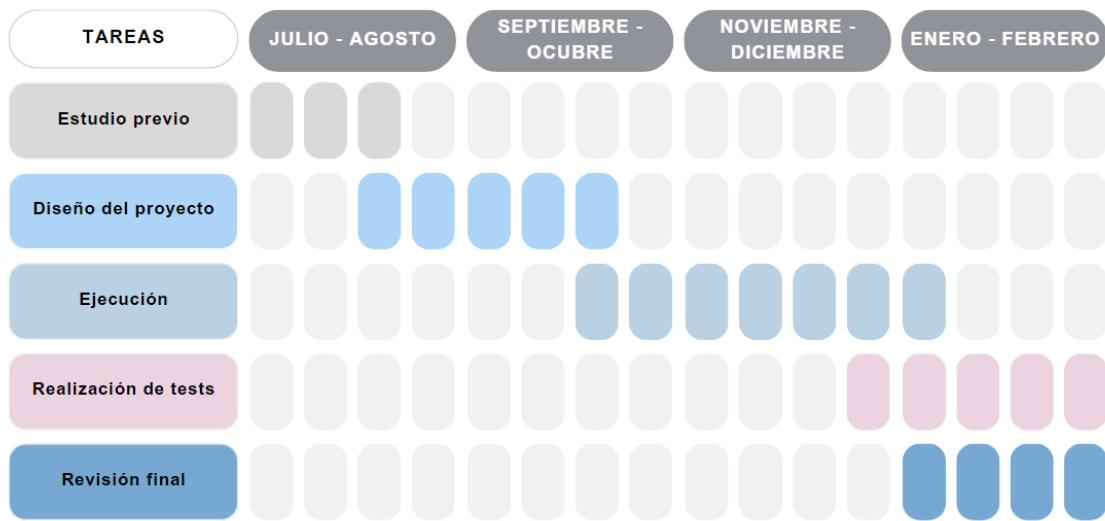


Figura 2.1: Diagrama de Gantt

2.3. Planificación temporal

La ejecución de este proyecto de Trabajo de Fin de Grado correspondió a un período de ocho meses, desde julio de 2023 hasta febrero de 2024. A continuación se hace, un breve desglose de la planificación temporal del proyecto:

- **Julio 2023:** Este mes marca el inicio del proyecto, con la presentación del mismo bajo la guía del profesor Gregorio Robles Martínez. Además, comienzan las reuniones con Virginia, mi codirectora de Trabajo de Fin de Grado (TFG), donde juntos empezamos a esbozar los elementos básicos que queríamos incorporar en el sitio web.
 - **Agosto 2023:** Este mes fue dedicado a una serie de consultas repetidas con Virginia para afinar la visión del proyecto. Realizamos una exploración exhaustiva de diversas páginas web centradas en arquitectura, con el objetivo de infundir ideas innovadoras en nuestra propia plataforma.
 - **Agosto - Octubre 2023:** Durante este período, todo el tiempo fue dedicado a la fase de implementación y desarrollo de la plataforma web. Construí páginas dedicadas tanto para usuarios registrados como para visitantes. Durante las sesiones tutoriales, nos centramos

en aspectos técnicos como el manejo de permisos de usuarios, la incorporación de un sistema de admisión de usuarios y la funcionalidad de recuperación de contraseñas.

- **Noviembre - Diciembre 2023:** Durante estos meses, se llevaron a cabo importantes adiciones a la plataforma. Integré un calendario y una sección de libros para enriquecer la experiencia del usuario. Además, se habilitaron los comentarios, aunque solo para usuarios registrados, con el fin de fomentar la participación y el intercambio de ideas.
- **Enero - Febrero 2024:** Durante los últimos dos meses, nos reunimos para finalizar detalles pendientes y abordar los últimos requisitos del proyecto. Además, comenzamos a redactar la memoria del proyecto y corregimos ciertos detalles en el código según las necesidades identificadas. Este período fue crucial para garantizar la calidad y la coherencia del trabajo realizado antes de su presentación final.

Capítulo 3

Estado del arte

En este capítulo se presentarán las herramientas y librerías utilizadas en el Trabajo de Fin de Grado. Esto nos proporcionará una visión detallada de las tecnologías empleadas a lo largo del proyecto, lo cual es fundamental para comprender la infraestructura y el entorno de desarrollo utilizados en la implementación de la plataforma.

3.1. Tecnologías y Herramientas

3.1.1. Python

Python¹, desarrollado por Guido van Rossum a finales de los 80 en los Países Bajos, es un lenguaje de programación con un enfoque en la legibilidad del código y la productividad del programador. Desde su primera versión en 1991, ha experimentado un crecimiento significativo y se ha convertido en uno de los lenguajes más populares y utilizados en la actualidad. Su sintaxis simple, clara y precisa facilita la escritura de código limpio, mientras que su extensa gama de bibliotecas estándar acelera el proceso de desarrollo. Su adaptabilidad y compatibilidad con diversos paradigmas de programación han permitido su uso en una amplia variedad de aplicaciones, desde el desarrollo de software hasta el análisis de datos y machine learning.

¹<https://www.python.org/>

3.1.2. GitHub

GitHub² es un servicio en la nube que aloja repositorios de un sistema de control de versiones, Git, lo que permite a los desarrolladores colaborar en proyectos compartidos manteniendo un seguimiento detallado de su progreso. Funciona como una plataforma web para el desarrollo colaborativo de software, facilitando la colaboración y el almacenamiento de proyectos de código abierto o privado. En GitHub, los usuarios pueden trabajar juntos en proyectos, compartiendo ideas y soluciones. Su principal ventaja es la posibilidad de trabajar en un proyecto en cualquier lugar y momento, ya que se realiza un seguimiento del desarrollo de forma remota. Además, proporciona otras funciones vitales a los desarrolladores, como bibliotecas y funciones para la gestión de versiones de software, la gestión de problemas y la revisión de código [1].

3.1.3. phpMyAdmin

PhpMyAdmin³ es una herramienta creada en PHP que facilita la administración de bases de datos MySQL a través de una interfaz de usuario en un navegador web. Entre sus funciones se incluyen[4]: la creación, modificación y eliminación de bases de datos y tablas; la gestión de campos y el ajuste de privilegios; y la capacidad de exportar datos en varios formatos. Esta herramienta multilingüe está disponible bajo la licencia GPL Versión 2. Desde su creación en 1998, phpMyAdmin ha experimentado continuos avances y adaptaciones a las máquinas con servidores web y soporte de PHP y MySQL.

3.1.4. JavaScript

JavaScript⁴ es un lenguaje de programación de alto nivel, interpretado y orientado a objetos, conocido por su capacidad para ofrecer interactividad a las páginas web. Es un lenguaje del lado del cliente, por lo que se ejecuta en el navegador del usuario y puede interactuar con el contenido de la página, el Document Object Model (DOM) y otros elementos del navegador. Puede manipular el contenido de la página y responder a eventos del usuario. Además, es

²<https://github.com/>

³<http://localhost/phpmyadmin/>

⁴<https://lenguajejs.com/javascript/>

un lenguaje asincrónico, lo que le permite realizar tareas sin bloquear la ejecución del código. Es esencial para la manipulación del DOM, permitiendo agregar, eliminar o modificar elementos dinámicamente. También maneja eventos del usuario, facilita la creación de aplicaciones web interactivas y utiliza tecnologías como XMLHttpRequest o la interfaz Fetch para realizar solicitudes asíncronas al servidor. Dispone de diversos frameworks y bibliotecas que facilitan el desarrollo de aplicaciones web complejas. Además, sigue las especificaciones definidas por ECMAScript y es pluriparadigmático, admitiendo varios estilos de programación.

3.1.5. Docker

Docker⁵ es una plataforma de código abierto que permite automatizar el despliegue, la escalabilidad y la operación de aplicaciones dentro de contenedores de software. Estos contenedores permiten empaquetar una aplicación junto con todas sus dependencias en una unidad estandarizada para el desarrollo de software, lo que facilita su portabilidad entre distintos entornos.

Algunos de los puntos clave de Docker son:

1. **Portabilidad:** Docker permite empaquetar y ejecutar aplicaciones en prácticamente cualquier entorno, lo que facilita la migración de aplicaciones y sistemas.
2. **Aislamiento:** Los contenedores de Docker permiten el aislamiento de recursos, lo que garantiza que cada contenedor funcione como una unidad autónoma y que los problemas en un contenedor no afecten a otros.
3. **Eficiencia:** Docker es muy eficiente en términos de rendimiento ya que los contenedores que utiliza consumen menos recursos que las máquinas virtuales equivalentes.
4. **Escalabilidad:** Docker facilita la escalabilidad de las aplicaciones, ya que permite crear y desplegar contenedores rápidamente y en gran medida.
5. **Integración continua y entrega continua (CI/CD):** Docker se integra bien con los sistemas de CI/CD, lo que permite probar las aplicaciones en contenedores en distintas etapas del desarrollo y despliegue de forma automatizada.

⁵<https://www.docker.com/>

6. **Descentralización del desarrollo:** Docker ayuda a los equipos de desarrollo a trabajar de manera descentralizada y colaborativa, ya que cualquier miembro del equipo puede crear y desplegar contenedores con su propio trabajo.

3.1.6. LaTeX

LaTeX⁶ es un sistema de composición de textos basado en TeX, desarrollado por Leslie Lamport en 1984, que automatiza la generación de documentos científicos, técnicos y académicos de alta calidad. Al aprovechar las funciones avanzadas de TeX, LaTeX permite crear documentos con una amplia variedad de elementos, como fórmulas matemáticas, imágenes y texto programático, utilizando una sintaxis de comandos para controlar la estructura y el formato del documento. Ampliamente utilizado en la comunidad científica y académica debido a su capacidad para producir documentos profesionales con facilidad y eficiencia, LaTeX es una herramienta esencial para la creación de contenido complejo y detallado.

3.2. Librerías

En el proyecto se emplearon diversas tecnologías para desarrollar una plataforma web robusta y escalable. Se utilizó HTML, CSS y JavaScript para la interfaz de usuario, Bootstrap y jQuery para el diseño y la funcionalidad, Flask en el lado del servidor con Python para la lógica y la gestión de la base de datos, MySQL y phpMyAdmin para la gestión de la base de datos, y Git para el control de versiones del código. Estas herramientas permitieron cumplir con los requisitos del proyecto de manera efectiva.

3.2.1. Flask

Flask [3] es un marco de trabajo para desarrollo web en Python. Las funciones que se mencionan a continuación son partes esenciales de Flask:

- Blueprint: Permite organizar rutas en módulos separados para una mejor organización del código.

⁶<https://es.overleaf.com/>

- `render_template`: Se utiliza para renderizar archivos de plantillas y generar una respuesta HTTP a partir de ellos.
- `redirect`: Redirige al navegador a otra URL.
- `session`: Permite almacenar información que es persistente entre las solicitudes, como datos de usuario.
- `request`: Proporciona información sobre la solicitud HTTP del cliente, como parámetros de formulario.
- `flash`: Se usa para enviar mensajes a la próxima solicitud, lo que resulta útil para proporcionar feedback al usuario.
- `url_for`: Genera una URL a una vista basada en un nombre y argumentos, lo que facilita la generación de enlaces dinámicos.

3.2.2. Flask-Mail

Flask-Mail es una extensión que facilita el envío de correos electrónicos desde una aplicación Flask. Permite crear y enviar mensajes de correo electrónico de manera sencilla.

```
from flask_mail import Message
```

3.2.3. Werkzeug

Werkzeug es una biblioteca WSGI utilizada en Flask. `secure_filename` se utiliza para asegurar que los nombres de archivo proporcionados por los usuarios sean seguros para usar en el sistema de archivos, evitando posibles ataques de inyección de directorios.

```
from werkzeug.utils import secure_filename
```

3.2.4. mysql-connector-python

Esta biblioteca es la implementación oficial de MySQL para Python. Proporciona un conjunto de operaciones de base de datos y manejo de errores compatibles con el estándar DB API. En este caso, `mysql` es una instancia de conexión a la base de datos MySQL configurada en

un archivo de configuración (`config.py`), que se importa para interactuar con MySQL en la aplicación Flask.

```
from config import mysql
```

3.2.5. os, random, string

Estos son módulos estándar de Python:

- `os`: Proporciona una manera de utilizar funcionalidades dependientes del sistema operativo, como la manipulación de archivos.
- `random`: Se utiliza para generar números aleatorios o seleccionar elementos aleatorios de una secuencia.
- `string`: Contiene operaciones comunes para trabajar con cadenas de texto, como la generación de caracteres aleatorios.

3.2.6. Decorators y Config

`decorators` y `config` son módulos personalizados creados específicamente para la aplicación.

- `config`: Contiene configuraciones de la aplicación, como la cadena de conexión a la base de datos y otras variables de entorno. Es común utilizar este tipo de archivos para centralizar todas las opciones de configuración de la aplicación y facilitar su gestión.
- `decorators`: Contiene decoradores que son una característica la cual permite modificar o extender el comportamiento de funciones o métodos de manera transparente. En el contexto de las aplicaciones web, los decoradores se utilizan comúnmente para aplicar acciones adicionales a las rutas de la aplicación, como la autenticación de usuarios o la gestión de permisos.

```
Decorator
def login_required(func):
    @wraps(func)
```

```
def wrapper(*args, **kwargs):  
    if 'login' not in session:  
        return redirect('/admin/login')  
    return func(*args, **kwargs)  
    return wrapper
```

Colocamos decorador
`@admin_bp.route('/calendar')`
`@login_required`
`def admin_calendar():`
`# Código`

3.2.7. Flask-Login

Flask-Login es una extensión de Flask que proporciona funcionalidades relacionadas con la gestión de sesiones de usuarios. Ofrece funciones esenciales para manejar la autenticación de usuarios, tales como iniciar sesión, cerrar sesión, recordar usuarios y proteger ciertas vistas para que solo los usuarios autenticados puedan acceder a ellas.

3.2.8. Flask-MySQL y Flask-MySQLdb

Flask-MySQL y Flask-MySQLdb son extensiones de Flask que proporcionan soporte para MySQL en aplicaciones de Flask. Estas bibliotecas facilitan la interacción con bases de datos MySQL.

3.2.9. MySQLClient y PyMySQL

MySQLClient y PyMySQL son interfaces de Python para conectarse con MySQL y realizar operaciones en la base de datos.

3.2.10. FullCalendar

FullCalendar es una potente librería de JavaScript que permite crear calendarios interactivos y personalizables para aplicaciones web. Esta librería ha sido desarrollada para facilitar la visualización y gestión de eventos en calendarios en entornos web, ofreciendo una gran variedad de características y opciones de configuración.

3.2.11. Itsdangerous

Itsdangerous es una biblioteca de Python que se utiliza para garantizar la seguridad en las aplicaciones. Esta herramienta nos brinda la capacidad de generar y validar tokens seguros, firmar y cifrar información sensible, y manejar cookies de manera segura. En el contexto de nuestro proyecto, estamos utilizando Flask-Login para gestionar la autenticación de usuarios, Itsdangerous se convierte en una parte fundamental. Por ejemplo, en la función `olvide_contrasena`, utilizamos Itsdangerous para generar un token seguro que luego se envía al usuario por correo electrónico, permitiéndole restablecer su contraseña de forma segura. De manera similar, en la función `restablecer_contrasena`, Itsdangerous se encarga de verificar la validez del token recibido antes de permitir al usuario realizar el cambio de contraseña.

3.2.12. MarkupSafe

MarkupSafe es una biblioteca de Python que se utiliza en aplicaciones web, para crear cadenas seguras en HTML y XML evitando así ataques de inyección de código al escapar automáticamente caracteres especiales. En el código HTML, MarkupSafe se integra con frameworks como Flask o Jinja para escapar de los datos dinámicos, previniendo ataques XSS y garantizando la seguridad del contenido mostrado. Además, MarkupSafe protege la integridad de los formularios al asegurar que los datos ingresados por los usuarios estén correctamente formateados, previniendo así ataques de inyección de HTML y asegurando una interacción segura con la aplicación web.

3.2.13. Flask-Mail

Flask-Mail es una extensión de Flask que simplifica el envío de correos desde aplicaciones de Flask. Proporciona una interfaz sencilla para la configuración de correos y envío de mensajes, facilitando la implementación de funcionalidades relacionadas con correos, como la confirmación de cuentas o la recuperación de contraseñas.

3.2.14. PIP

Pip es un sistema de administración de paquetes para Python y se utiliza para instalar y administrar paquetes de software escritos en Python.

3.2.15. Jinja2

Jinja2 es un motor de plantillas para Python que permite la generación de contenido HTML dinámico. Se utiliza en conjunto con Flask para generar páginas web dinámicas y se distingue por su simplicidad y flexibilidad.

Algunas características clave de Jinja2:

1. **Sintaxis de plantillas:** Jinja2 proporciona una sintaxis clara y concisa para la definición de plantillas, permitiendo a los desarrolladores generar contenido dinámico de manera eficiente y legible.
2. **Herencia de plantillas:** Permite la reutilización de código a través de la herencia de plantillas, donde una plantilla puede heredar bloques de contenido de una plantilla padre.
3. **Filtros:** Proporciona una serie de filtros útiles que transforman variables para su visualización. Algunos ejemplos son filtros para cambiar a mayúsculas o minúsculas, truncar texto, modificar fechas, entre otros.
4. **Tags de control de flujo:** Ofrece tags para estructuras de control, como bucles y condicionales. Con ellos, es posible realizar operaciones como recorrer listas o verificar condiciones para mostrar u ocultar partes de la plantilla.
5. **Escapado de texto:** Se encarga automáticamente del escapado de texto, protegiendo las aplicaciones de ataques como Cross Site Scripting (XSS).

6. Alto rendimiento: Compila las plantillas en bytecode de Python, lo que significa que la ejecución de las plantillas es rápida y eficiente.

En el código HTML, Jinja2 se utiliza para incluir fragmentos de código HTML con la directiva `include`, para iterar sobre rangos de números con la directiva `for`, y para insertar valores de variables con las llaves dobles (`{ variable }`).

3.2.16. MySQL

MySQL⁷ es un sistema de gestión de bases de datos relacional (RDBMS) de código abierto que utiliza el lenguaje de consulta estructurado (SQL) para manipular datos. Destaca por su escalabilidad para adaptarse de aplicaciones pequeñas a sistemas empresariales de gran tamaño, su soporte para transacciones, su compatibilidad con diversas plataformas, y su robusta comunidad de usuarios. Ofrece diferentes motores de almacenamiento, siendo InnoDB uno de los más populares por su soporte de transacciones ACID. Asimismo, proporciona variadas herramientas y utilidades para la administración y monitoreo de bases de datos.

3.2.17. Bootstrap

Bootstrap⁸ es un framework de desarrollo frontend basado en HTML, CSS y JavaScript, que facilita la creación de interfaces web responsivas. Proporciona un sistema de rejilla para organizar el diseño en filas y columnas, un conjunto de componentes predefinidos reutilizables (botones, formularios, navegación, alertas, etc.), estilos predeterminados para la tipografía y funcionalidades interactivas mediante JavaScript. Aunque ofrece un conjunto completo de estilos y componentes, Bootstrap es altamente personalizable. Cuenta con una documentación detallada y una amplia comunidad de desarrolladores.

3.2.18. jQuery

jQuery es una biblioteca de JavaScript ampliamente utilizada que proporciona una forma simplificada de manipular el DOM, manejar eventos, crear animaciones y realizar llamadas AJAX. Es reconocida por su facilidad de uso y su soporte multiplataforma.

⁷<https://www.mysql.com/>

⁸<https://getbootstrap.com/>

Las características claves de jQuery son las siguientes:

1. **Manipulación del DOM:** jQuery proporciona una interfaz fácil y eficiente para manipular el árbol de elementos DOM en una página web.
2. **Facilidad de uso:** jQuery presenta una API fácil de usar que funciona en una multitud de navegadores. Esto incluye una manera sencilla de navegar por un documento, crear animaciones, manejar eventos y desarrollar aplicaciones con AJAX.
3. **Llamadas AJAX:** jQuery simplifica mucho el proceso de envío de solicitudes AJAX y la manipulación de las respuestas.
4. **Compatibilidad entre navegadores:** jQuery es conocida por manejar de manera eficiente muchas de las incompatibilidades entre diferentes versiones de navegadores y sus respectivos motores de JavaScript.
5. **Extensibilidad:** jQuery puede ser extendida de diversas maneras para crear, por ejemplo, nuevos métodos que pueden ser reutilizados en más de un proyecto jQuery.

En el código proporcionado se utiliza jQuery para ejecutar una serie de funciones después de que el documento HTML esté cargado y listo.

```
$ (document) .ready (function () { /código/ }) ;
```

Además, se utiliza para localizar elementos en el DOM, vincular eventos a elementos, enviar solicitudes AJAX y manipular la interfaz del usuario.

3.2.19. Google Translate API

La API de Google Translate se utiliza para integrar la funcionalidad de traducción en aplicaciones, sitios web, y servicios. Esta API permite la traducción de texto en tiempo real a varios idiomas y se basa en los mismos algoritmos de aprendizaje automático que utiliza Google Translate.

Las características claves de Google Translate API son:

1. **Detección de idioma:** La API puede detectar automáticamente el idioma de la entrada y traducirlo al idioma de la salida requerido.

2. **Diversidad de idiomas:** Soporta más de 100 idiomas.
3. **Traducción en tiempo real:** Permite la traducción de texto en tiempo real, lo que permite a los usuarios obtener traducciones instantáneas mientras escriben.
4. **Simplicidad de uso:** Con solo algunas líneas de código, los desarrolladores pueden integrar fácilmente la API en sus aplicaciones o sitios web.

En el archivo "translation.js", Google Translate API se utiliza para inicializar un nuevo objeto de traducción con una configuración específica y para traducir el contenido de una página a un idioma especificado.

Capítulo 4

Diseño e implementación

A continuación, se expondrán con detalle los métodos y herramientas empleados en la ejecución del proyecto, ofreciendo una perspectiva minuciosa sobre el desarrollo de la iniciativa que tiene como finalidad la creación de una página web para el almacenamiento de toda la documentación relevante. Este capítulo ahonda en las diversas etapas del proyecto, detallando las decisiones adoptadas y las justificaciones vinculadas a las metodologías seleccionadas. Asimismo, se expone el diseño e implementación del proyecto con la intención de proporcionar a otros la capacidad de contribuir, adaptar o ampliar el sistema. Esta sección brinda una visión completa de cómo se estructuró y llevó a cabo el proyecto, abarcando las distintas fases de su creación.

4.1. Arquitectura general

En esta sección, es presentada la organización de los archivos y directorios tiene un propósito específico que soporta la funcionalidad, el mantenimiento, y la escalabilidad del proyecto. Entender esta estructura facilita navegar por el código y contribuye al desarrollo del proyecto.

La estructura del proyecto se puede ver en la Figura 4.1 que se describe a continuación:

- `static/`: En este directorio se encuentran los archivos estáticos que son solicitados directamente por el cliente, como las hojas de estilo CSS, los scripts JavaScript, las imágenes, archivos y otros recursos multimedia. Esto permite una carga rápida y eficiente de los elementos visuales de la aplicación en el navegador del usuario.

```

TFG-Karol
|
|--- venv (entorno virtual de python)
|--- static (Recursos estáticos utilizados por las plantillas HTML)
|   |--- archivos
|   |   |--- fondos
|   |   |--- imágenes
|   |   |--- pdf
|   |   |--- videos
|
|   |--- css (Hojas de estilo)
|   |--- imágenes
|   |--- js (scripts JavaScript)
|
|--- templates (Plantillas HTML usadas por Flask para renderizar páginas del sitio)
|   |--- admin (Plantillas para el panel de administración)
|   |--- sitio (Plantillas para el sitio web público)
|   |--- base.html
|   |--- detalle_libro.html
|   |--- error.html
|
|--- __init__.py (Contiene la creación de la aplicación Flask)
|--- admin_routes.py (Las rutas para el panel de administración)
|--- app.py (Punto de entrada principal para iniciar la aplicación)
|--- config.py (Archivo de configuración del proyecto)
|--- decorators.py (Decoradores utilizados en todo el proyecto)
|--- Dockerfile (Archivo para crear un contenedor Docker para el proyecto)
|--- requirements.txt (Dependencias del proyecto)
|--- sitio_routes.py (Las rutas para el sitio web público)

```

Figura 4.1: Estructura del proyecto

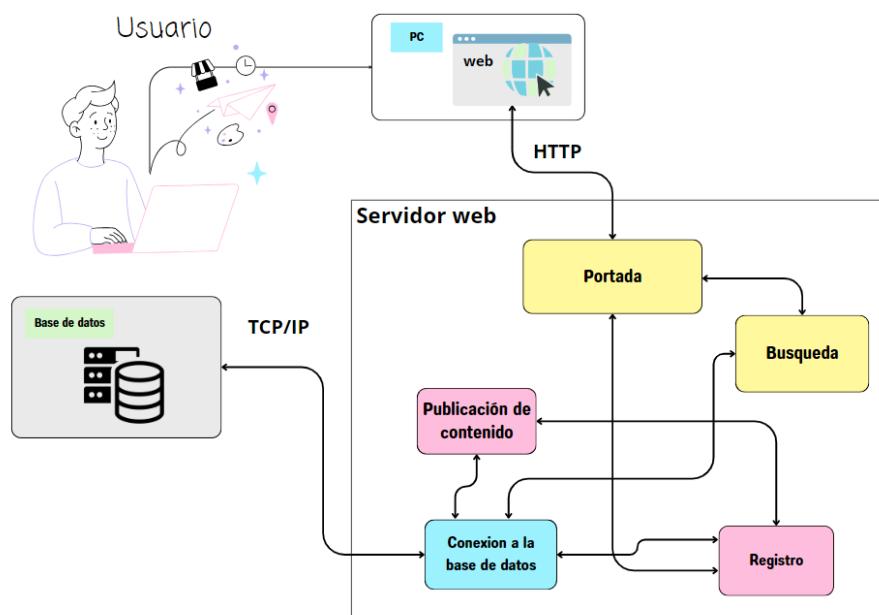


Figura 4.2: Arquitectura del sistema

- `templates/`: Aquí residen las plantillas HTML utilizadas por Flask para generar las páginas web dinámicas. Estas plantillas incluyen el código HTML junto con expresiones y directivas específicas de Flask que permiten la inserción de datos dinámicos en las páginas. Separar las plantillas del código Python facilita la gestión del contenido y la presentación de la aplicación.
- `venv/`: Este directorio alberga el entorno virtual de Python utilizado para instalar y gestionar las dependencias del proyecto. El uso de un entorno virtual garantiza que las dependencias del proyecto se instalen de forma aislada, evitando conflictos con otras versiones de paquetes instalados globalmente en el sistema.
- `__init__.py`: Este archivo especial indica a Python que el directorio debe tratarse como un paquete. Aunque puede estar vacío, comúnmente se utiliza para inicializar el paquete o para importar funciones y clases que se utilizarán en otros módulos del proyecto.
- `admin_routes.py, sitio_routes.py`: Estos módulos definen las rutas de la aplicación, que asignan las URL a las funciones encargadas de manejar las solicitudes del cliente. Separar las rutas en distintos archivos según su funcionalidad ayuda a mantener el código organizado y facilita su mantenimiento y escalabilidad.
- `app.py`: Este archivo sirve como punto de entrada de la aplicación Flask. Aquí se inicializa la aplicación Flask y se integran los distintos componentes del proyecto, como las rutas, las configuraciones y las extensiones. Es el archivo principal que ejecuta la aplicación y gestiona su funcionamiento.
- `config.py`: Contiene los valores de configuración de la aplicación, como las claves secretas y la información de conexión a la base de datos. Separar la configuración en un archivo dedicado facilita su gestión y permite ajustar la configuración según el entorno de desarrollo, pruebas o producción.
- `decorators.py`: En este archivo se definen los decoradores personalizados utilizados en el proyecto. Los decoradores son funciones que envuelven otras funciones y permiten agregar funcionalidades adicionales, como la autenticación de usuarios o el control de acceso. Su uso ayuda a mantener el código limpio y modular, facilitando la implementación de políticas de seguridad y autorización.

- `Dockerfile`: Contiene las instrucciones para construir una imagen Docker de la aplicación. Docker es una herramienta de virtualización que permite encapsular la aplicación y todas sus dependencias en un contenedor independiente, lo que facilita su despliegue en diferentes entornos de producción de manera consistente y reproducible.
- `requirements.txt`: En este archivo se listan todas las dependencias de Python necesarias para ejecutar el proyecto. Estas dependencias pueden instalarse fácilmente utilizando la herramienta pip, lo que simplifica la gestión de las dependencias y garantiza que todas las bibliotecas necesarias estén disponibles en el entorno de ejecución.

Esta estructura organizada y coherente de archivos y directorios ayuda a mantener separados los distintos aspectos de la aplicación, facilitando su desarrollo, mantenimiento y escalabilidad a medida que el proyecto evoluciona.

4.2. Arquitectura específica

4.2.1. Programa principal

El archivo `app.py` es el que inicializa la aplicación Flask y con el cual se establece la configuración principal de la misma. Las configuraciones incluyen la conexión a la base de datos MySQL, la inicialización del correo y la generación de una clave secreta para la sesión. Con el fin de proveer mayor seguridad, se implementa un decorador `login_required`, el cual se encarga de verificar si el usuario ha iniciado sesión antes de poder acceder a ciertas rutas.

A continuación, se encuentran varias rutas esenciales del proyecto con sus respectivas funciones de manejo:

- `/`: Redirige al usuario a la página principal.
- `/ver_libro()`: Con esta función se puede visualizar el detalle de cada libro, la función realiza una consulta a la base de datos para obtener la información específica de un libro. También se manejan los errores en caso de no encontrar el libro en cuestión.

- `/update_event()`: Esta ruta se utiliza para actualizar los detalles de un evento específico, donde se recogen los datos del evento a través de una consulta POST y se actualizan en la base de datos.
- `/insert_event()`: Esta función maneja la inserción de un nuevo evento en la base de datos al recibir la información relevante del evento a través de una consulta POST.
- `/ajax_delete()`: Esta función realiza la eliminación de un evento específico cuando se recibe una solicitud POST que contiene el id del evento a eliminar.

Finalmente, el registro de los blueprints se realiza al final. Los blueprints son plantillas que permiten la organización de las rutas y proporcionan modularidad a la aplicación. En este caso, se tienen dos blueprints: `admin_bp` y `sitio_bp` los cuales agrupan las rutas asociadas a las funcionalidades del administrador y del sitio respectivamente.

La aplicación se inicia únicamente si `app.py` se ejecuta como el módulo principal de Python, proporcionando un punto de entrada para iniciar la aplicación. En este caso, la aplicación se ejecuta en modo depuración, lo que significa que Flask proporcionará una página con información de depuración detallada si se produce un error.

4.2.2. Configuración

El archivo `config.py` contiene la configuración esencial de la plataforma web. Este archivo es fundamental en cualquier aplicación Flask, ya que su principal objetivo es mantener todos los valores de configuración en un solo lugar para facilitar la administración y el mantenimiento.

En este archivo, se define una clase `Config` que contiene las siguientes configuraciones:

- **MYSQL_HOST**: Es la dirección del servidor de la base de datos MySQL que se utilizará. Se proporcionan varias opciones comentadas para poder cambiar rápidamente de host.
- **MYSQL_USER**: Es el nombre de usuario para acceder a la base de datos MySQL.
- **MYSQL_PASSWORD**: Es la contraseña del usuario de la base de datos MySQL.
- **MYSQL_DB**: Es el nombre de la base de datos a la cual se establecerá la conexión.

- **MAIL_SERVER**: Es el servidor de correo que se utiliza para enviar correos electrónicos desde la aplicación.
- **MAIL_PORT**: Es el puerto al que se conectará para enviar correos electrónicos.
- **MAIL_USE_TLS**: Es una opción de configuración para utilizar TLS para la conexión.
- **MAIL_USERNAME**: Es la dirección de correo electrónico que se utilizará para enviar correos.
- **MAIL_PASSWORD**: Es la contraseña de la cuenta de correo electrónico que se utilizará para enviar correos.

Todos estos valores se almacenan como variables de clase dentro de la clase `Config` y pueden ser accedidos una vez que se crea una instancia de la misma.

Posteriormente, se crean dos objetos para manejar las conexiones a MySQL y al servicio de correo. Estos objetos serán utilizados más adelante para interactuar con la base de datos y con el servidor de correo, respectivamente.

4.2.3. Decoradores

En Python, un decorador es una función que toma otra función y extiende su comportamiento sin modificarla explícitamente. Los decoradores proporcionan una forma sencilla y legible de agregar funcionalidades adicionales a las funciones y métodos existentes.

El archivo `decorators.py` contiene la definición de un decorador personalizado:

- **login_required**: Este decorador se utiliza para garantizar que el usuario esté autenticado antes de poder acceder a ciertas rutas de la aplicación web. Específicamente, verifica si el valor 'login' está presente en el objeto de sesión de Flask. Si no es así, redirige al usuario al formulario de inicio de sesión. Si el usuario está autenticado, la función decorada se ejecuta como se espera.

Con este decorador, la comprobación de autenticación se puede realizar de manera sencilla y consistente en todas las rutas que requieran que el usuario esté autenticado. Simplemente es necesario incluir `@login_required` antes de la definición de la función de la ruta para aplicar esta comprobación.

Este uso de los decoradores ayuda a mantener el código limpio y legible, y refuerza que cada pieza de código tiene un propósito específico.

El decorador `@wraps(func)` se usa para mantener el nombre y la documentación de las funciones originales cuando se usan con decoradores. Esto es útil para la depuración y también para trabajar con la documentación de ayuda (`help`) incorporada en Python.

4.2.4. Rutas del Sitio

El archivo `sitio_routes.py` es un módulo que contiene la definición de las rutas y las funciones de manejo asociadas al sitio principal.

Comienza con la creación de una instancia de `Blueprint`, que se utiliza para organizar las rutas en la aplicación Flask. Tener las rutas en un `Blueprint` permite la modularidad y la reutilización del código.

Las rutas específicas del 'sitio' y sus funciones de manejo son las siguientes:

- `/books()`: Esta función se encarga de obtener todos los libros de la base de datos y renderizar la plantilla `books.html` para mostrarlos.
- `/buscar_libros()`: Con esta función, se permite buscar libros en base al año, área y autor. La función construye una consulta SQL en base a los parámetros de búsqueda ingresados y obtiene los resultados correspondientes de la base de datos.
- `/calendar()`: La ruta del calendario[2] se utiliza para mostrar los eventos. La función maneja el acceso a la base de datos para obtener todos los eventos y luego los muestra en la plantilla `calendar.html`.
- `/links()`: Esta ruta se encarga de recopilar todos los enlaces disponibles en la base de datos y renderizar la plantilla `links.html` para mostrarlos.
- `/link_detail()`: Esta función maneja la visualización detallada de un único enlace. La función toma un parámetro `link_id` de la URL, realiza una consulta a la base de datos para obtener los detalles de ese enlace en particular y los muestra en la plantilla `link_detail.html`.

- /journals (): Para las derivaciones de revistas, esta función se encarga de obtener todas las revistas de la base de datos y renderizar la plantilla journals.html para mostrarlas.

Parte del código para obtener libros desde la base de datos:

```
@sitio_bp.route('/books')
def books():
    try:
        with mysql.connection.cursor() as cur:
            cur.execute("SELECT * FROM `libros`")
            libros = cur.fetchall()
    except Exception as e:
        print(f"Error fetching data from database: {e}")
        libros = []
    return render_template('sitio/books.html', libros=libros)
```

Este módulo de rutas del 'sitio' se encarga de manejar todas las peticiones con respecto a la visualización de libros, enlaces y revistas en el sitio principal, así como la búsqueda de libros y la visualización del calendario.

4.2.5. Rutas del Administrador

El archivo admin_routes.py contiene la definición de las rutas asociadas a las funcionalidades del administrador en la plataforma web.

1. Verificación de Inicio de Sesión

- /admin_index (): Página de inicio del administrador, requiere autenticación.
- /admin_login () y /admin_login_post (): Manejan el inicio de sesión del administrador.
- /admin_login_cerrar (): Cierra la sesión del usuario administrador.
- /admin_registro (): Registra un nuevo usuario administrador.

- `/olvide_contrasena()` y `/restablecer_contrasena()`: Proceso de restablecimiento de contraseña.

2. Manejo de Libros

- `/admin_books()`: Operaciones de libros en la interfaz de administración.
- `/guardar_archivo()`: Maneja el guardado de archivos cargados por el usuario.
- `/admin_libros_guardar()`: Maneja la lógica para guardar un nuevo libro en la base de datos.
- `/admin_books_delete()`: Elimina un libro existente de la base de datos.

3. Manejo de Permisos de Usuarios

- `/admin_permisos()`: Permite a los administradores gestionar los permisos de otros usuarios.
- `/admin_permisos_editar()`: Actualiza los campos de permisos existentes en la tabla de usuarios.
- `/admin_permisos_eliminar()` y `/admin_permisos_aceptar()`: Eliminan usuarios y aceptan solicitudes de registro pendientes, respectivamente.

4. Manejo de Comentarios

- `/agregar_comentario()`: Agrega comentarios a un libro en la base de datos.
- `/eliminar_comentario()`: Elimina un comentario seleccionado de un libro en la base de datos.

5. Manejo de Enlaces

- `/admin_links()`: Permite a los administradores gestionar los enlaces en la plataforma.
- `/admin_links_guardar()`: Guarda un nuevo enlace en la base de datos.
- `/admin_links_delete()`: Elimina un enlace de la base de datos.

6. Manejo de Diarios

- /admin_journals(): Permite a los administradores gestionar los diarios en la plataforma.
- /admin_journals_guardar(): Guarda un nuevo diario en la base de datos.
- /admin_journals_delete(): Elimina un diario de la base de datos.

7. Manejo del Calendario

- /admin_calendar(): Muestra los eventos almacenados en la base de datos en un formato de calendario. Verifica los roles de los usuarios para determinar si tienen permiso para acceder.

8.

Parte del código para eliminar libros desde la base de datos:

```
@admin_bp.route('/books/delete', methods=['POST'])
def admin_books_delete():
    # Check for authorization
    if 'login' not in session:
        return redirect('/admin/login')
    book_id = request.form['txtID']

    try:
        with mysql.connection.cursor() as cur:
            cur.execute("START TRANSACTION")
            cur.execute("DELETE FROM comentarios
                        WHERE libro_id = %s", (book_id,))
            cur.execute("DELETE FROM libros WHERE id = %s", (book_id,))
            cur.execute("COMMIT")

        return redirect('/admin/books')
    except Exception as e:
```

```

    cur.execute("ROLLBACK")
    flash('Error deleting book.', 'error')
    print(f"Error deleting book: {e}")
    return redirect('/admin/books')

```

En resumen, el archivo `admin_routes.py` maneja todos los aspectos relacionados con la autenticación, el manejo de sesiones y las funcionalidades administrativas de la plataforma web.

4.2.6. Base de Datos

Con phpMyAdmin, las tablas se pueden visualizar, administrar y estructurar fácilmente mediante una interfaz gráfica sencilla de utilizar. Esta herramienta permite realizar operaciones como ver, agregar, modificar y eliminar registros en cada tabla, así como ejecutar consultas SQL directamente en la base de datos. Esto facilita la gestión completa de la base de datos de la aplicación de gestión de contenido, desde la administración de usuarios y libros hasta la gestión de eventos y comentarios. Además, phpMyAdmin proporciona una interfaz intuitiva para realizar tareas como la creación de nuevas tablas y la modificación de la estructura de las existentes, lo que simplifica el desarrollo y mantenimiento de la base de datos.

■ Tabla usuarios

- `id_usuario`: identificador único para cada usuario.
- `username`, `password`, `email`, `id_rol`: información básica del usuario.
- `registro_pendiente`: indica si hay acciones de registro pendientes para el usuario.

■ Tabla links

- `id`: identificador único para cada enlace.
- `nombre`, `url`: detalles del enlace.
- `user_id`: referencia al `id_usuario` en la tabla usuarios.

■ Tabla journals

- `id, title, year, link, cover, user_id`: detalles de la revista, `user_id` es una clave foránea.

■ Tabla **comentarios**

- `id, comentario, user_id, libro_id`: detalles del comentario. `user_id` y `libro_id` son claves foráneas.

■ Tabla **events**

- `id, title, place, start_event, end_event`: detalles del evento.

■ Tabla **libros**

- Contiene detalles sobre los libros como `id, nombre, imagen, url, year, area_id, autor_id, descripcion, video, pdf, imagen_secundaria, id_usuario`.

■ Tabla **reset_tokens**

- `id, email, token, created_at`: detalles para restablecer la contraseña del usuario.

■ Tabla **role**

- `id_rol, descripcion`: detalles del rol del usuario.

Entre estas tablas (ver Figura 5.4), las claves foráneas crean las siguientes relaciones:

- La tabla **libros** tiene una relación con la tabla **usuarios** a través del campo `id_usuario`.
- Las tablas **journals**, **comentarios** y **links** tienen relaciones con la tabla **usuarios** a través del campo `user_id`.
- La tabla **libros** tiene relaciones con las tablas de **autores** y **áreas** a través de los campos `autor_id` y `area_id`, respectivamente.
- La tabla **usuarios** tiene una relación con la tabla **roles** a través del campo `id_rol`.
- La tabla **reset_tokens** esta relacionada con la tabla **usuarios** a través del campo `email`.

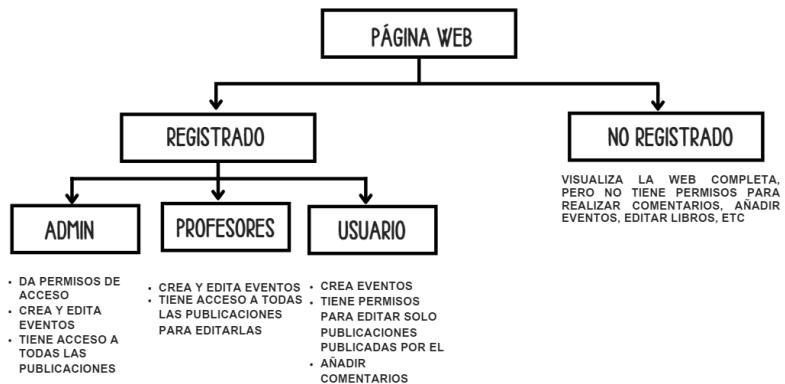


Figura 4.3: Permisos de los Usuarios

Esta estructura de relaciones permite administrar eficientemente libros, revistas, comentarios, enlaces y eventos en la base de datos. Además, facilita la gestión de la autenticación de usuarios y el restablecimiento de contraseñas en la aplicación.

4.3. Vista de Usuario

La Figura (ver Figura 4.3) muestra un esquema general de la estructura de la página web. Esta página es accesible tanto para usuarios registrados como para visitantes, aunque la principal distinción entre estos dos roles radica en los privilegios que poseen.

Como visitante, se tiene acceso a la vista pública de la página, permitiendo visualizar el contenido ya publicado. No obstante, la interacción es limitada, ya que los visitantes no pueden comentar, subir nuevo contenido o interactuar con las publicaciones existentes.

Por otro lado, los usuarios registrados disfrutan de permisos adicionales que les permiten una interacción amplia con la página. Además de ver el contenido, pueden participar activamente a través de comentarios, subir nuevo contenido y interactuar con las publicaciones existentes.

Este diseño permite a los visitantes explorar la web y, al mismo tiempo, incita a un mayor compromiso a través del registro, permitiendo así aprovechar todas las funciones que ofrece la plataforma.

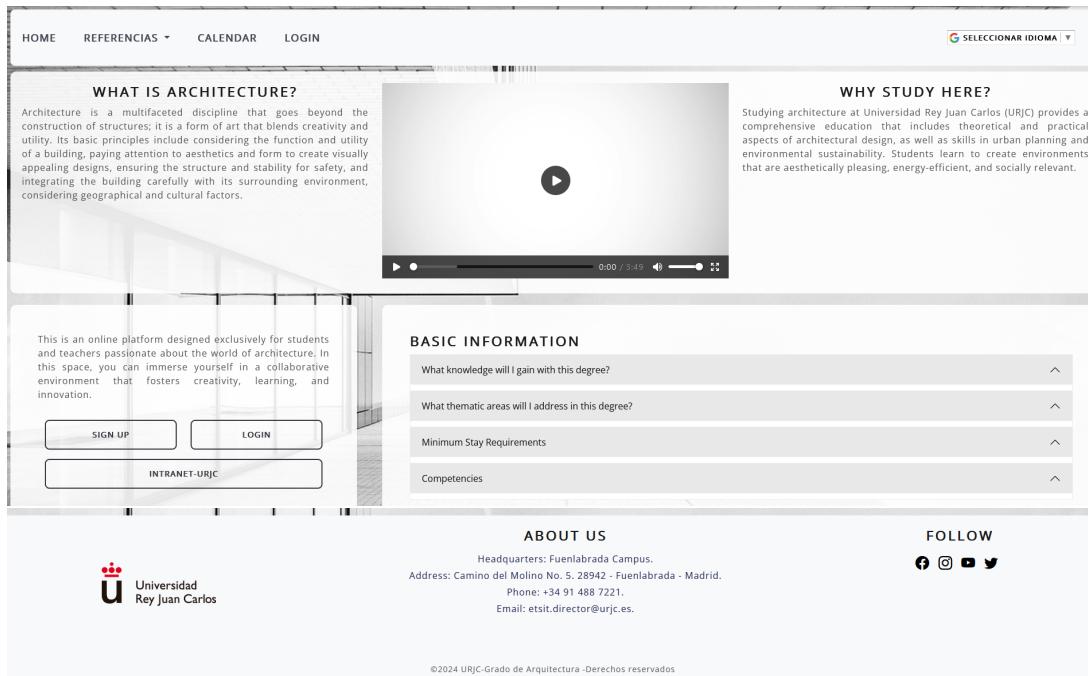


Figura 4.4: Vista de página principal

4.3.1. Vista de Usuario - No Registrado

En esta sección se detallan las diferentes páginas y archivos relacionados con la experiencia del usuario no registrado en la plataforma:

- `sitio/top_of_page.html`: Este archivo contiene la barra de navegación principal de la aplicación, que proporciona enlaces a diversas páginas y funcionalidades del sitio web.
- `sitio/index.html`: Se trata de la página de inicio de la plataforma (ver Figura 4.4), donde los usuarios pueden comenzar su navegación.
- `sitio/books.html`: Esta página muestra la estructura y propiedades de la sección de libros (ver Figura 4.5), donde los usuarios pueden explorar los libros disponibles en la plataforma y obtener más detalles sobre ellos.
- `sitio/calendar.html`: Aquí se presenta el calendario (ver Figura 4.8), mostrando eventos presentes y futuros relevantes para los usuarios.
- `sitio/filterForm.html`: Ofrece a los usuarios la posibilidad de filtrar el contenido según ciertos criterios establecidos.

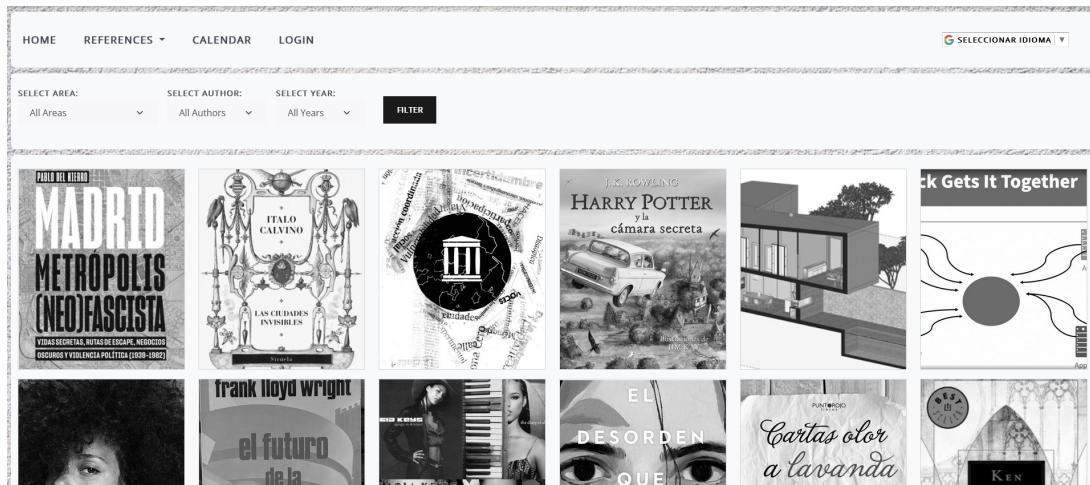


Figura 4.5: Vista de la página de libros

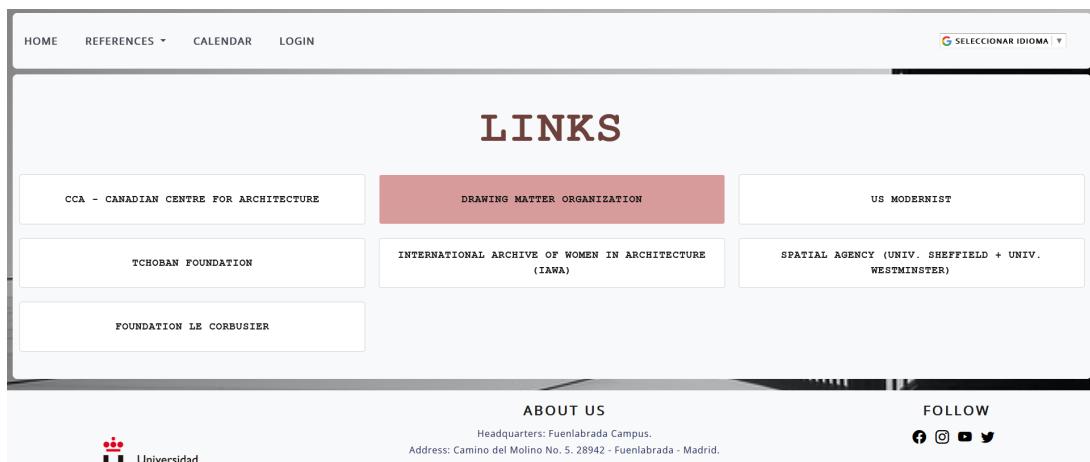


Figura 4.6: Vista de la página de links

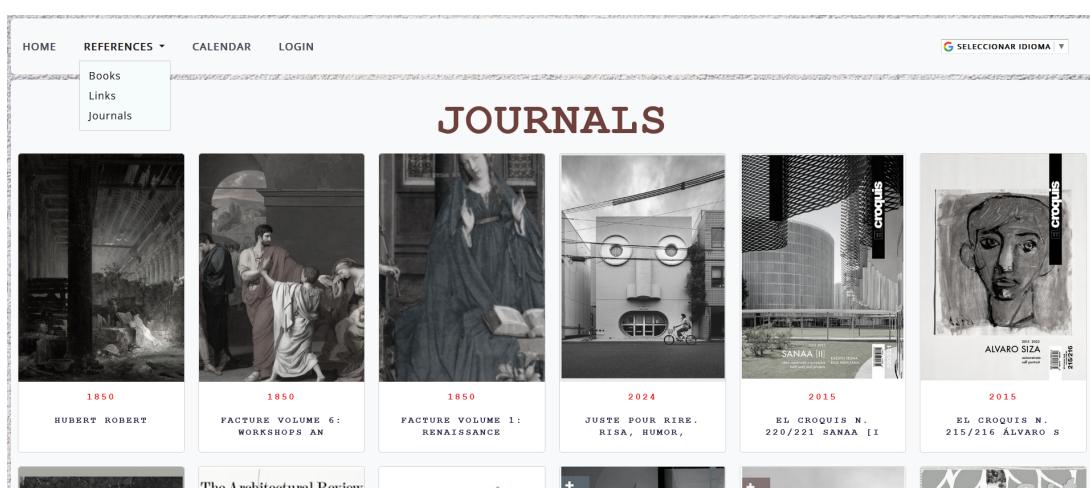


Figura 4.7: Vista de la página de revistas

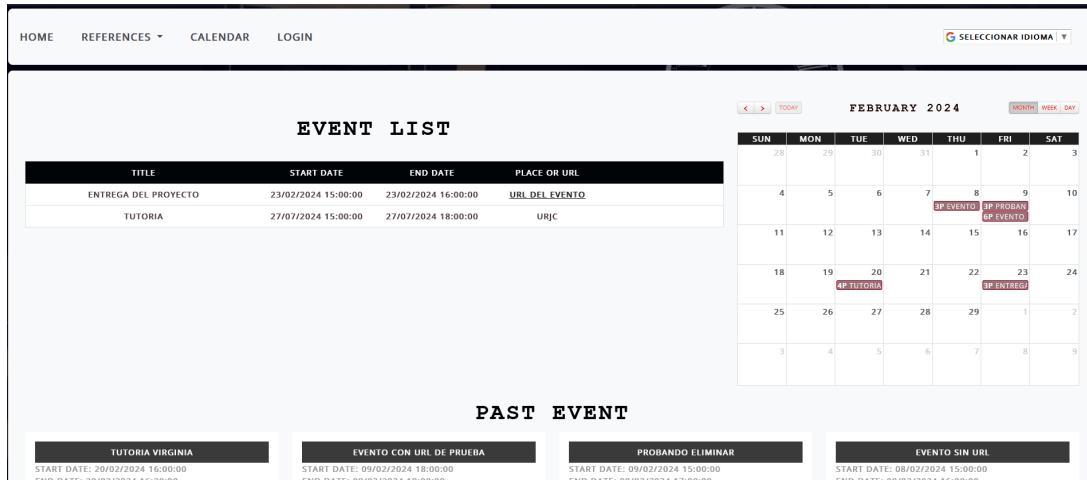


Figura 4.8: Vista de la página de calendario

- `sitio/journals.html`: Define la estructura y características de la sección de revisas 4.7, donde se muestran las revistas disponibles en la plataforma.
- `sitio/links.html`: En esta página se muestran los enlaces (ver Figura 4.6) disponibles para los usuarios.
- `sitio/footer.html`: Es el pie de página que se muestra en todas las páginas de la plataforma, proporcionando información adicional y enlaces útiles.
- `base.html`: Define la apariencia general común a todas las páginas, permitiendo que cada página hija llene bloques específicos con su título y contenido.
- `detalle_libro.html`: Presenta detalles específicos de un libro, permitiendo a los usuarios agregar comentarios sobre el libro y al administrador eliminar comentarios si es necesario.
- `error.html`: Proporciona una interfaz amigable para mostrar mensajes de error a los usuarios en caso de que ocurra algún problema, ofreciéndoles la opción de volver a la página anterior.

Estos archivos constituyen los elementos fundamentales que componen la experiencia de usuario para aquellos que aún no se han registrado en la plataforma.

4.3.2. Vista de Usuario - Registrado

En esta sección se detallan las diferentes páginas y archivos relacionados con la experiencia de usuarios registrados en la plataforma:

- `admin/top_of_page.html`: Este archivo define la barra de navegación en la sección de administración del sitio web. Incluye enlaces para las páginas de inicio, libros, links, revistas, calendario y cierre de sesión. Además, muestra un enlace adicional para la página de permisos si la 'id_rol' del usuario en sesión es igual a 1. También proporciona un traductor de Google para cambiar el idioma del contenido.
- `admin/calendar.html`: En esta página se presenta el calendario (ver Figura 4.10), que permite la creación, edición y eliminación de eventos pasados, presentes y futuros, dependiendo del rol del usuario identificado por su `id_rol`.
- `admin/index.html`: Esta es la vista de la página de inicio del administrador en el sitio web. Ofrece un mensaje de bienvenida, información sobre la comunidad de arquitectura y enumera las expectativas para los miembros de la comunidad.
- `admin/books.html`: Aquí se encuentra una interfaz para cargar libros al sitio desde la sección de administración. Incluye un formulario con campos para ingresar detalles como el título del libro, imagen de portada, nombre y apellido del autor, URL, descripción, video, archivo PDF... Además, permite seleccionar el año y el área. También muestra los documentos existentes en la plataforma.
- `admin/documents.html`: Esta vista muestra una tabla para gestionar los libros cargados en la página de administración del sitio web. Proporciona una lista de todos los libros mostrando su título y autor. Para cada libro, se ofrecen acciones para eliminar, visualizar y editar los detalles de los libros.
- `admin/editar_libro.html`: Proporciona una interfaz para editar los detalles de un libro (Previamente cargados de la base de datos). Incluye un formulario con campos para el título, imagen de portada, URL, año de publicación, área de estudio, nombre y apellido del autor, descripción, y opción de agregar o eliminar archivos multimedia como vídeo, archivo PDF o imagen secundaria.

- `admin/journals.html`: Ofrece una interfaz para gestionar revistas en la sección de administración. Incluye un formulario para añadir una nueva revista con campos para título, año, enlace y portada. También muestra una tabla con todas las revistas existentes, proporcionando opciones para eliminarlas.
- `admin/links.html`: Proporciona una interfaz para administrar enlaces en la sección de administración del sitio web. Permite agregar nuevos enlaces a través de un formulario que solicita título y URL, y muestra una tabla con todos los enlaces existentes, con opciones para eliminarlos.
- `admin/login.html`: Es la interfaz de inicio de sesión (ver Figura 4.9) para el área de administración del sitio web. Permite a los usuarios crear una cuentas, iniciar sesión ingresando su nombre de usuario y contraseña tambien dipone de un enlace para recuperar la contraseña en caso de olvido.
- `admin/olvide_contrasena.html`: Proporciona una interfaz para que los usuarios recuperen sus contraseñas en caso de olvido. Solicita al usuario ingresar su dirección de correo electrónico y, al enviar el formulario, se envía una solicitud para restablecer la contraseña a esa dirección de correo electrónico.
- `admin/restablecer_contrasena.html`: Permite a los usuarios restablecer sus contraseñas después de solicitar un restablecimiento. Los usuarios pueden ingresar su nueva contraseña en un formulario y, al enviarlo, la aplicación tomará esa nueva contraseña y la establecerá como la contraseña actual del usuario.
- `admin/registro.html`: Proporciona un formulario de registro de usuario. Permite a los nuevos usuarios crear un perfil en el sitio web proporcionando un nombre de usuario, contraseña y dirección de correo electrónico. Después de la presentación exitosa, se envía un correo electrónico de confirmación a la dirección de correo electrónico proporcionada.
- `admin/permisos.html`: Ofrece una interfaz para gestionar usuarios y sus permisos en el dashboard de administración del sitio web. Muestra en una tabla el nombre de usuario, correo electrónico y rol de cada usuario registrado. El administrador puede cambiar el rol de un usuario y eliminar usuarios. También muestra el estatus de aprobación para el acceso del usuario y permite al administrador aceptar o rechazar la solicitud de acceso.

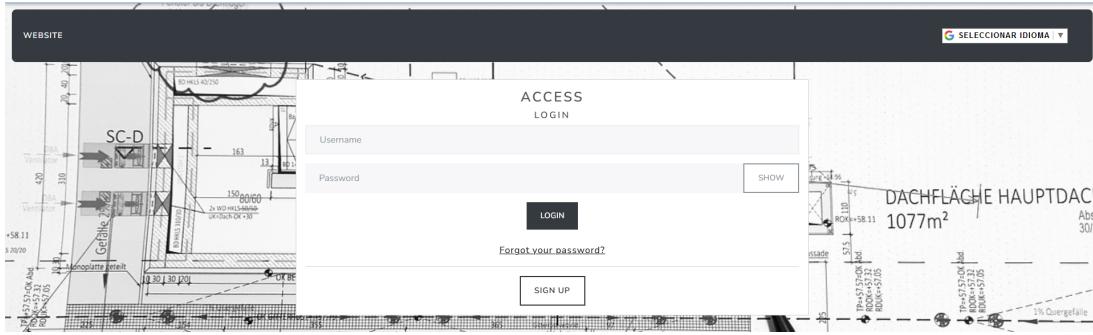


Figura 4.9: Acceso de a la plataforma

Title	Start Date	End Date	Place or URL	Action
entrega del proyecto	23/02/2024 15:00	23/02/2024 16:00	https://www.urjc.es/intranet-urjc	Delete
Tutoria	27/07/2024 15:00	27/07/2024 18:00	URJC	Delete

Figura 4.10: Calendario desde admin

Estos archivos constituyen los elementos fundamentales que componen la experiencia de usuarios que se han registrado o que están por registrarse en la plataforma.

Capítulo 5

Experimentos y validación

En este capítulo se explicarán las diferentes pruebas que se fueron realizando durante el desarrollo de este proyectos.

5.1. Pruebas con la aplicación

A lo largo del desarrollo del proyecto, se llevaron a cabo pruebas continuas para evaluar y garantizar el correcto funcionamiento del sistema en todas sus etapas. Estas pruebas se realizaron de manera periódica y sistemática, abarcando diferentes aspectos del proyecto, desde la funcionalidad básica hasta características más específicas. El objetivo principal de estas pruebas fue detectar y corregir cualquier problema o inconveniente lo antes posible, asegurando así la calidad y la fiabilidad del producto final. Este enfoque contribuyó significativamente a la mejora continua del proyecto y a la entrega de un proyecto final de alta calidad.

5.1.1. Pruebas Funcionales

Nuestro objetivo es garantizar el funcionamiento óptimo de todas las rutas y funciones del sistema. Para lograr esto, realizamos una exhaustiva validación de formularios y enlaces, asegurándonos de que cada elemento de la interfaz de usuario funcione de manera correcta y coherente. Esta revisión minuciosa nos permite identificar cualquier posible inconveniente o error en la navegación y la interacción del usuario (ver Figura 5.1), asegurando así una experiencia fluida y satisfactoria para todos los usuarios que ocupen la página web.



Figura 5.1: Mensajes de error

The image shows a web form titled 'ADD JOURNAL'. It has several input fields with validation messages:

- '**Title:' - An empty input field.
- '**Year:' - An input field containing '2010' with a validation message 'Completa este campo' (Fill this field) next to it.
- '**Link:' - An empty input field.
- '**Cover Image:' - A file selection input field with the placeholder 'Seleccionar archivo' (Select file) and 'Ninguno archivo selec.' (No file selected).

Figura 5.2: Alerta: Faltan datos en el formulario

5.1.2. Validación de Formularios

Se llevaron a cabo una serie de pruebas exhaustivas para asegurar el correcto funcionamiento de las validaciones del formulario en diversas acciones, como la subida, publicación o edición de un libro o evento, así como en el registro de usuarios. Durante este proceso, nos aseguramos de que cada aspecto del formulario se comporte como se espera (ver Figura 5.2). En caso de que se introduzcan datos incorrectos, verificamos que se muestren los mensajes de error correspondientes, proporcionando así una retroalimentación clara al usuario. Además, nos aseguramos de que cada elemento del formulario cumpla adecuadamente su función asignada, garantizando una experiencia fluida y sin contratiempos para el usuario final.

5.1.3. Pruebas de Compatibilidad e Interfaz de Usuario (UI)

Nuestra prioridad es garantizar la coherencia y la funcionalidad del sistema en una amplia variedad de entornos. Realizamos exhaustivas pruebas para verificar que el sistema se comporte de manera uniforme en diferentes navegadores y dispositivos (ver Figura 5.3). Nos esforzamos

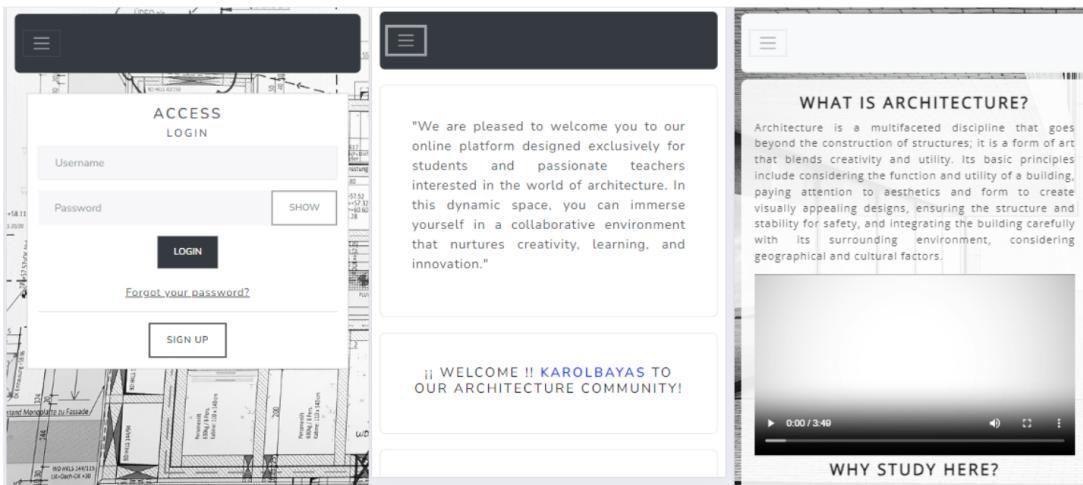
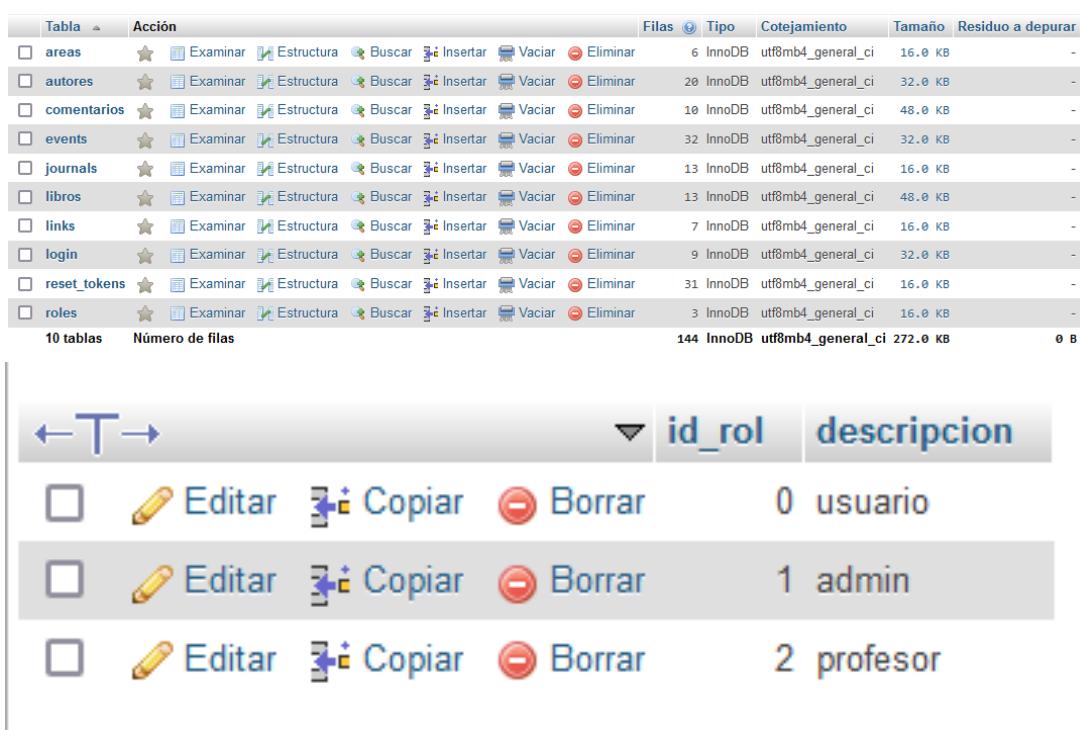


Figura 5.3: Vista desde un dispositivo móvil

por asegurar que cada página mantenga su apariencia y funcionalidad óptimas, sin importar la resolución de pantalla utilizada. De esta manera, nos aseguramos de proporcionar una experiencia de usuario consistente y satisfactoria en todas las circunstancias.

5.1.4. Pruebas de la Base de Datos

Se lleva a cabo una exhaustiva verificación para asegurar el correcto funcionamiento de las consultas a la base de datos (ver Figura 5.4). Se realizan pruebas detalladas para garantizar la precisión y la eficacia de cada interacción con la base de datos, asegurando así la integridad y la fiabilidad de los datos recuperados y procesados.



The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons for database management. Below the toolbar is a table listing 10 tables in the current schema:

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
areas	Examinar Estructura Buscar Insertar Vaciar Eliminar	6	InnoDB	utf8mb4_general_ci	16.0 KB	-
autores	Examinar Estructura Buscar Insertar Vaciar Eliminar	20	InnoDB	utf8mb4_general_ci	32.0 KB	-
comentarios	Examinar Estructura Buscar Insertar Vaciar Eliminar	10	InnoDB	utf8mb4_general_ci	48.0 KB	-
events	Examinar Estructura Buscar Insertar Vaciar Eliminar	32	InnoDB	utf8mb4_general_ci	32.0 KB	-
journals	Examinar Estructura Buscar Insertar Vaciar Eliminar	13	InnoDB	utf8mb4_general_ci	16.0 KB	-
libros	Examinar Estructura Buscar Insertar Vaciar Eliminar	13	InnoDB	utf8mb4_general_ci	48.0 KB	-
links	Examinar Estructura Buscar Insertar Vaciar Eliminar	7	InnoDB	utf8mb4_general_ci	16.0 KB	-
login	Examinar Estructura Buscar Insertar Vaciar Eliminar	9	InnoDB	utf8mb4_general_ci	32.0 KB	-
reset_tokens	Examinar Estructura Buscar Insertar Vaciar Eliminar	31	InnoDB	utf8mb4_general_ci	16.0 KB	-
roles	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	utf8mb4_general_ci	16.0 KB	-

Below the table, it says "10 tablas Número de filas" and "144 InnoDB utf8mb4_general_ci 272.0 KB 0 B".

Below this, there's a detailed view of the 'roles' table:

id_rol	descripcion
0	usuario
1	admin
2	profesor

The 'admin' row is currently selected.

Figura 5.4: Base de datos

Capítulo 6

Resultados

En el capítulo de resultados, se realizaron diversas pruebas para evaluar las funcionalidades de la plataforma en varios aspectos. Se centraron en la capacidad de cargar y eliminar libros, enlaces, revistas y eventos, asegurando que cada operación se llevara a cabo de manera adecuada y sin contratiempos.

6.1. Descripción de las pruebas realizadas

El capítulo de resultados presenta una serie de pruebas realizadas para evaluar diversas funcionalidades de la plataforma en diferentes aspectos. Aquí se detallan las pruebas realizadas y las conclusiones obtenidas:

6.1.1. Realizar Login o Registro de Usuarios

Se evaluó la capacidad de la plataforma para manejar el inicio de sesión de los usuarios registrados y el registro de nuevos usuarios, además de realizar las siguientes comprobaciones adicionales:

- **Envío de correo de registro:** Se verificó que el correo electrónico de confirmación de registro se enviara correctamente a los nuevos usuarios después de registrarse en la plataforma.(ver Figura 6.1)
- **Validación de nombre de usuario único:** Se aseguró de que no pudiera existir más de un usuario con el mismo nombre de usuario. Esta validación garantiza la unicidad de los

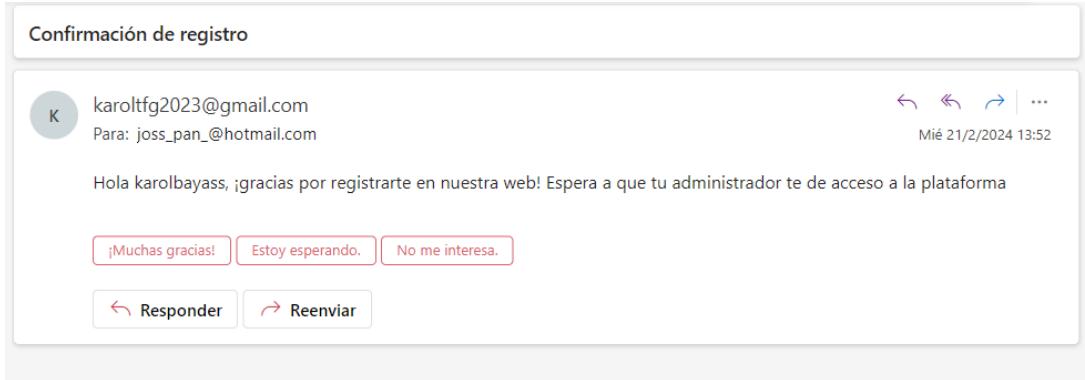


Figura 6.1: Correo de Registro Correcto

nombres de usuario en la plataforma. (ver Figura 6.3)

- **Acceso restringido hasta que se otorgue permiso:** Se comprobó que los usuarios no pudieran acceder a la aplicación hasta que se les otorgara acceso. Esto asegura que solo los usuarios autorizados puedan utilizar la plataforma una vez que se les haya dado permiso (ver Figura [reffig:waiting](#)).

6.1.2. Recuperación de Contraseña

Durante esta prueba, se evaluó la funcionalidad de recuperación de contraseña de la plataforma (ver Figura 6.4). Se verificó que los usuarios pudieran restablecer su contraseña en caso de olvidarla o necesitar cambiarla por motivos de seguridad. El proceso de recuperación de contraseña debe ser intuitivo y seguro, garantizando que solo el usuario legítimo pueda restablecer su contraseña. Para ello, se implementó un sistema que envía un correo electrónico con un enlace único y seguro que permite al usuario restablecer su contraseña. Se evaluaron aspectos como la recepción del correo electrónico de restablecimiento de contraseña, la seguridad de los enlaces de restablecimiento enviados por correo electrónico y la facilidad de uso del proceso en general. Esta funcionalidad es fundamental para la seguridad y la experiencia del usuario en la plataforma.

6.1.3. Gestión de Eventos: Añadir, Eliminar y Editar

Durante esta evaluación, pusimos a prueba la capacidad de la plataforma para administrar eventos (ver Figura 6.5). Nos aseguramos de que los usuarios con los permisos adecuados pu-

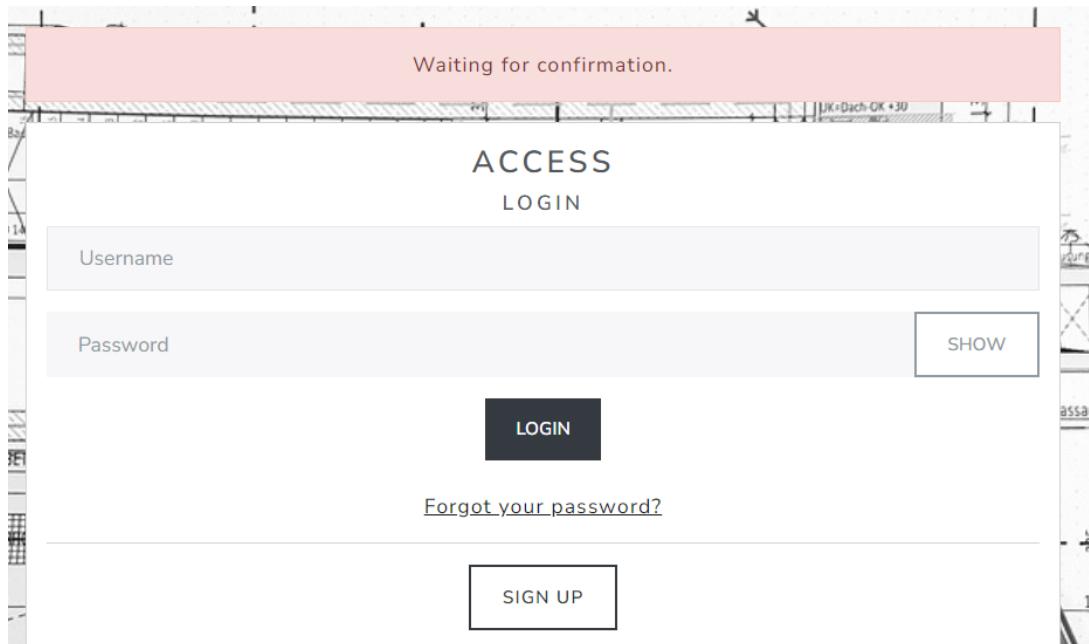


Figura 6.2: Mensaje: Esperando confirmación

dieran realizar tres funciones clave: añadir, eliminar y editar eventos, de manera eficiente y sin inconvenientes.

Para comenzar, nos centramos en el proceso de añadir un nuevo evento, asegurándonos de que fuera intuitivo y claro para los usuarios. Queríamos que pudieran ingresar fácilmente toda la información necesaria, como el título del evento, la fecha y la ubicación, sin complicaciones innecesarias.

Además, verificamos que los usuarios autorizados pudieran eliminar eventos existentes cuando fuera necesario. Esta función es vital para mantener actualizada la plataforma y garantizar que solo se muestren eventos relevantes.

La funcionalidad de edición de eventos también fue sometida a pruebas. Nos aseguramos de que los usuarios pudieran modificar con precisión los detalles de un evento, como el título, la fecha y la ubicación, sin cometer errores. Validamos cuidadosamente los datos introducidos durante la edición, evitando fechas incoherentes o información incorrecta.

En resumen, esta evaluación demostró que la plataforma puede gestionar eficazmente todo el ciclo de vida de los eventos, desde su creación inicial hasta cualquier modificación o eliminación necesaria, proporcionando una experiencia fluida y sin complicaciones para todos los usuarios.

USER REGISTRATION

USERNAME not valid

Username

Password

Email

SIGN UP

Figura 6.3: Mensaje: Username not valid

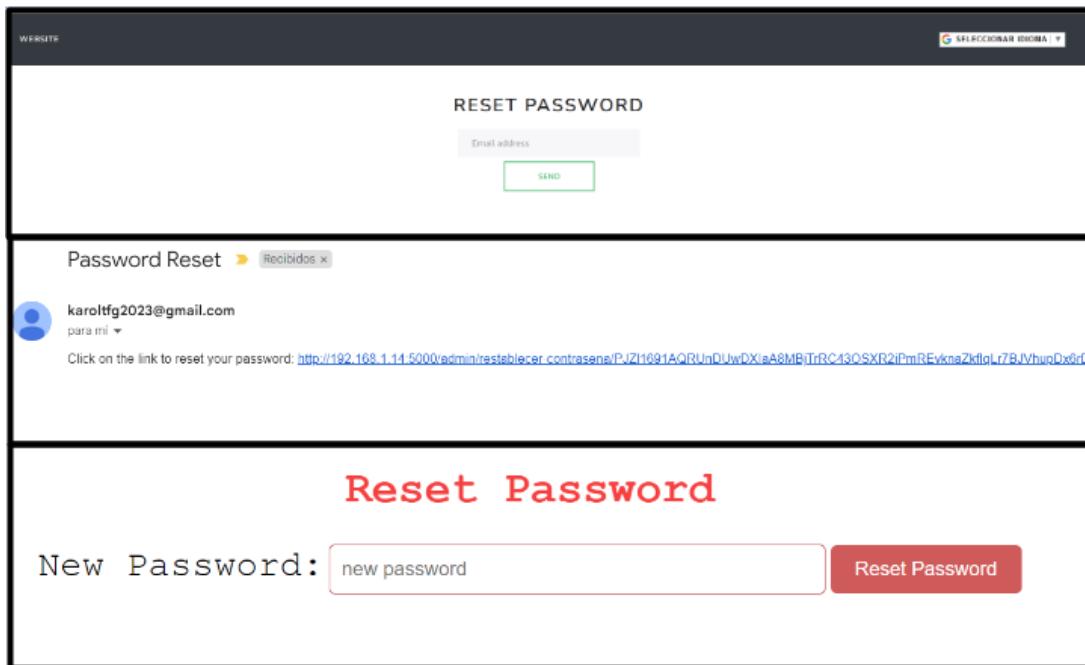


Figura 6.4: Restablecer contraseña

Title	Start Date	End Date	Place or URL	Action
entrega del proyecto	23/02/2024 15:00	23/02/2024 16:00	https://www.urjc.es/intranet-urjc	<button>Delete</button>
Tutoría	27/07/2024 15:00	27/07/2024 18:00	URJC	<button>Delete</button>

Figura 6.5: Calendario desde Administrador

6.1.4. Gestión de Enlaces y Revistas: Añadir y Eliminar

En esta fase de evaluación, nos concentraremos en la capacidad de la plataforma para administrar enlaces y revistas, centrándonos en las operaciones de añadir y eliminar (ver Figura 6.6).

Primero, nos aseguramos de que los usuarios autorizados pudieran añadir nuevos enlaces o revistas de manera sencilla y eficiente. Nos preocupamos por que el proceso de añadir fuera claro y directo, permitiendo a los usuarios ingresar la información necesaria de manera fluida, como el título, la URL (en el caso de los enlaces) o los detalles de la revista.

Seguidamente, evaluamos la función de eliminación para garantizar que los usuarios pudieran eliminar enlaces o revistas existentes según fuera necesario. Esta capacidad es crucial para mantener actualizada la plataforma y garantizar que solo se muestren recursos relevantes y válidos.

En resumen, durante esta evaluación confirmamos que la plataforma puede gestionar de manera efectiva la adición y eliminación de enlaces y revistas, proporcionando a los usuarios una experiencia sin complicaciones y garantizando la integridad de los recursos mostrados.

6.1.5. Gestión de Libros: Publicar, Eliminar y Editar

En esta fase de evaluación, nos centramos en las capacidades de la plataforma para la gestión de libros, abordando las operaciones de publicación, eliminación y edición. (ver Figura 6.7)

Primero, nos aseguramos de que los usuarios autorizados pudieran publicar nuevos libros de manera efectiva. Verificamos que el proceso de publicación fuera claro y accesible, permitiendo a los usuarios ingresar la información necesaria, como el título, el autor y la descripción del libro, de manera fluida y sin contratiempos.

Luego, evaluamos la función de eliminación para garantizar que los usuarios pudieran eliminar libros existentes según fuera necesario. Esta capacidad es esencial para mantener la plataforma actualizada y gestionar eficientemente el contenido disponible.

Por último, examinamos la función de edición para asegurar que los usuarios autorizados pudieran modificar la información de los libros de manera precisa y sin errores. Esto incluyó la capacidad de editar detalles como el título, el autor, la fecha, portada, añadir o eliminar imágenes, vídeos o PDF y la descripción del libro, asegurando que cualquier cambio realizado fuera coherente y válido.

ADD LINKS

**Title:

**Link:

LINKS

TITLE	LINK	ACTION
CCA - Canadian Centre for Architecture	https://www.cca.qc.ca/en/archives	ELIMINAR
Drawing Matter Organization	https://drawingmatter.org/	ELIMINAR

ADD JOURNAL

**Title:

**Year:

**Link:

**Cover Image:

JOURNALS

TITLE	YEAR	LINK	COVER IMAGE	ACTIONS
Hubert Robert	1850	http://www.rmg.org/features/sites/digital/hubert-robert-1733-1808.html#slide_1		ELIMINAR
Facture Volume 6: Workshops an	1850	http://www.rmg.org/features/publications/gallery-sections/workshops-studios.html		ELIMINAR

Figura 6.6: Links - Journals desde Administrador

En resumen, durante esta evaluación, confirmamos que la plataforma puede gestionar de manera efectiva la publicación, eliminación y edición de libros, proporcionando a los usuarios una experiencia sin complicaciones y garantizando la integridad del contenido mostrado.

6.1.6. Comentar en Publicaciones

Durante esta evaluación, nos centramos en la capacidad de la plataforma para permitir a los usuarios comentar en las publicaciones. Verificamos que los usuarios autorizados pudieran agregar comentarios de manera efectiva los libros. Nos aseguramos de que el proceso de comentario fuera claro y accesible, permitiendo a los usuarios expresar sus opiniones y participar en discusiones de manera significativa (ver Figura 6.8).

Además, evaluamos la funcionalidad de moderación de comentarios para garantizar que los administradores pudieran gestionar los comentarios de manera efectiva, incluida la capacidad de eliminar comentarios inapropiados o irrelevantes según fuera necesario.

En resumen, durante esta evaluación confirmamos que la plataforma puede facilitar la interacción de los usuarios a través de comentarios en los libros, promoviendo la participación activa y la comunicación entre todos.

6.1.7. Dar Permisos a Usuarios

En esta fase de evaluación, nos centramos en la capacidad de la plataforma para administrar los permisos de los usuarios (Solo el administrador podra otorgar permisoa). Verificamos que los administradores pudieran asignar y gestionar los permisos de los usuarios de manera efectiva, controlando qué funciones y se pueda acceder con los usuarios en función de sus roles y privilegios (ver Figura 6.9).

Nos aseguramos de que el proceso de asignación de permisos fuera claro y accesible, permitiendo a los administradores definir los roles de los usuarios y asignar permisos específicos de manera precisa y sin complicaciones.

Además, evaluamos la funcionalidad de control de acceso para garantizar que los usuarios solo pudieran acceder a las funciones y recursos para los que tienen los permisos adecuados, garantizando así la seguridad y la integridad de la plataforma.

En resumen, durante esta evaluación confirmamos que la plataforma puede gestionar de ma-

6.1. DESCRIPCIÓN DE LAS PRUEBAS REALIZADAS

55

The screenshot shows a web-based administration interface for managing books. At the top, there is a navigation bar with links: HOME, BOOKS, LINKS, REVISTAS, CALENDAR, WEBSITE, CLOSE, and PERMISOS. Below this is a section titled 'INSTRUCTIONS' containing a note about file upload requirements.

Below the instructions, there are several input fields for adding a new book:

- Title:** Write the file's title
- Cover Image:** A file selection input field showing 'Ninguno archivo selec.'
- Author Name:** Author's First Name
- Author Last Name:** Author's Last Name
- URL for additional information:** download url

Below these fields is a message: "Here you can view the uploaded books, modify them, delete them, or access them directly."

A table titled 'ACTIONS' lists five books with their titles and authors:

ACTIONS	TITLE	AUTHOR
DELETE BOOKS EDITAR	MADRID, METRÓPOLIS (NEO)FASCISTA	PABLO DEL HIERRO
DELETE BOOKS EDITAR	LE CITTÀ INVISIBILI	ITALO CALVINO
DELETE BOOKS EDITAR	ARQUITECTURA DE MONTAJE	ALEJANDRA PANCHO
DELETE BOOKS EDITAR	HARRY POTTER Y LA CÁMARA SECRETA	J.K ROWLING
DELETE BOOKS EDITAR	vamos a realizar distintas pruebas	MELANI BAYAS

Below the table is an 'EDIT BOOK' form for the first book listed:

EDIT BOOK

Title: MADRID, METRÓPOLIS (NEO)FASCISTA

Cover Image: Ninguno archivo selec.

Download URL: <https://www.planetadelibros.com/libro-madrid-metropolis-neofascista/374505>

***Year:** 2023

***Area:** CONSTRUCCIÓN

Author Name: PABLO

Author Last Name: DEL HIERRO

Description:
paseos y recuerdos como peones de la nueva guerra de Madrid. Una España dividida por circunstancias convenció, en un momento u otro, en una misma geografía: Madrid. Así, mientras la España franquista intentaba borrar cualquier recuerdo de sus simpatías en favor de Hitler y Mussolini y realinearse como baluarte católico y anticomunista, la capital española ejercía de facto como la principal Metrópolis neofascista.

Video: Ninguno archivo selec.

PDF: Ninguno archivo selec.

Eliminar PDF:

Image: Ninguno archivo selec.

Eliminar Imagen Secundaria:

SAVE CHANGES **CANCEL**

Figura 6.7: Books desde la vista admin

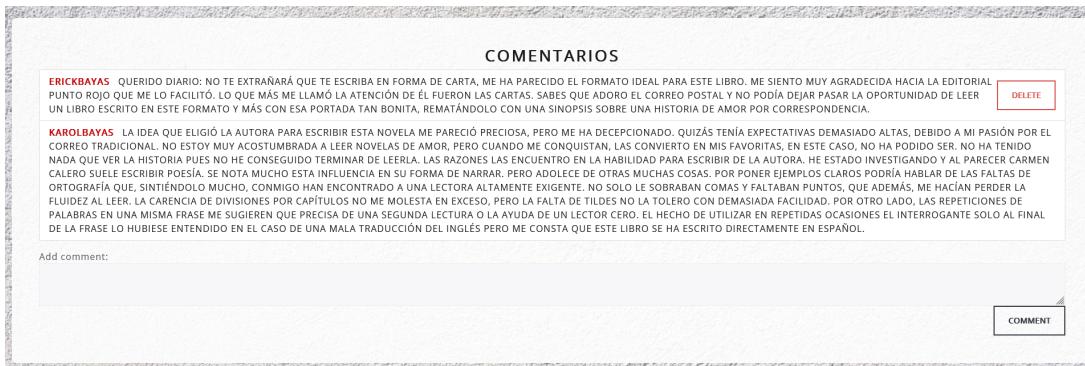


Figura 6.8: Vista de Comentarios desde el perfil de un alumno

USERNAME	EMAIL	ID_ROL	DELETE	ACCESS
karolbayas	karolbayas@gmail.com	1	SAVE	DELETE
prueba	isamkarol@gmail.com	0	SAVE	DELETE
virginia	virginia@gmail.com	2	SAVE	DELETE
erickbayas	erick_25hm@gmail.com	0	SAVE	DELETE
havetkarol	karolbaya@gmail.com			Accepted

Figura 6.9: Vista de Permisos desde el Administrador

nra efectiva los permisos de los usuarios, proporcionando un control granular sobre el acceso a las funciones y recursos, y garantizando la seguridad y la confidencialidad de la información.

6.2. Conclusiones de los resultados

Durante estas pruebas, se observó que los libros podían ser subidos y también editados para ajustar detalles como el título, el autor, el año, el área, la descripción entre otros datos. Además, se confirmó que la función de eliminación funcionaba de manera efectiva, permitiendo la eliminación exitosa de libros de la plataforma. Sin embargo, en el caso de los enlaces y las revistas, se constató que solo era posible cargarlos y eliminarlos, sin la opción de realizar ediciones en sus detalles.

Además, se realizó un análisis detallado para garantizar que los diferentes tipos de usuarios solo pudieran acceder a los elementos pertinentes y a los archivos que ellos mismos habían cargado. Por ejemplo, se verificó que los usuarios alumnos no pudieran acceder a las funciones

de administración ni ver documentos subidos por otros usuarios que no estuvieran disponibles para el acceso público. Mientras tanto, los administradores tuvieron acceso y pudieron realizar operaciones en todos los archivos, independientemente de su origen.

Estos resultados validan la robustez de la plataforma y su capacidad para gestionar las operaciones necesarias de manera segura y eficiente, cumpliendo con los estándares y requisitos establecidos previamente.

Capítulo 7

Conclusiones

7.1. Conclusión del Proyecto

Este proyecto ha sido una experiencia enriquecedora, enfocada en proporcionar una solución integral y satisfactoria para los usuarios, especialmente en el ámbito de la arquitectura. A lo largo de su desarrollo, hemos diseñado e implementado una plataforma versátil que cumple con las expectativas tanto de usuarios no registrados como de administradores del sistema.

Durante este proceso, nos hemos centrado en garantizar la calidad y la usabilidad del sistema. Realizamos pruebas exhaustivas en cada etapa del desarrollo para asegurarnos de que todas las funcionalidades, desde la validación de formularios hasta la gestión de usuarios, funcionen de manera óptima, brindando una experiencia fluida y sin problemas a los usuarios.

La inclusión de un panel de administración ha sido fundamental para permitir una gestión eficiente y controlada del sistema. Esto ha otorgado al administrador la capacidad de asignar roles, gestionar usuarios y controlar el acceso a diversas funcionalidades, lo que contribuye a la seguridad y la integridad del sistema en su conjunto.

En resumen, este proyecto representa un esfuerzo colaborativo y meticuloso orientado a satisfacer las necesidades y expectativas de los usuarios. A través de un diseño cuidadoso y una implementación diligente, hemos logrado crear un sistema que no solo cumple con los requisitos funcionales, sino que también ofrece una experiencia de usuario gratificante y satisfactoria.

7.2. Aplicación de lo aprendido

La aplicación de los conocimientos y habilidades adquiridos durante mi formación en el Grado en Ingeniería en Sistemas Audiovisuales y Multimedia ha sido fundamental para la realización de este Trabajo de Fin de Grado. A continuación, mencionaré los aprendizajes que me permitieron llevar a cabo la extracción, tratamiento y clasificación de datos para realizar el análisis de este proyecto:

1. **Informática I:** Esta asignatura me proporcionó las bases fundamentales de programación, sentando los cimientos necesarios para comprender y desarrollar código de manera efectiva.
2. **Informática II:** En esta asignatura, además de reforzar mis conocimientos previos de programación, adquirí una formación más avanzada en herramientas comúnmente utilizadas en el desarrollo de software. Además, mejoré en la estructuración del código y en la lógica necesaria para la programación.
3. **Protocolos para la Transmisión de Audio y Vídeo por Internet:** En este curso, obtuve conocimientos sobre el lenguaje de programación Python y aprendí cómo se estructura y para qué se utiliza la estructura de datos JSON. Estos conocimientos fueron fundamentales para la manipulación y tratamiento de datos en este proyecto.
4. **Construcción de Servicios y Aplicaciones Audiovisuales en Internet:** En este curso, adquirí habilidades específicas relacionadas con la construcción de servicios y aplicaciones en el ámbito audiovisual en internet. Aprendí sobre los principios fundamentales y las tecnologías utilizadas en el desarrollo de servicios y aplicaciones multimedia en la web, lo que me proporcionó una base sólida para la implementación de funcionalidades relacionadas con el contenido audiovisual en este proyecto.
5. **Laboratorio de Tecnologías Audiovisuales en la Web:** Durante este curso, tuve la oportunidad de aplicar los conocimientos teóricos adquiridos en la construcción práctica de aplicaciones y servicios multimedia en la web. A través de actividades de laboratorio y proyectos prácticos, desarrolté habilidades prácticas en el diseño, desarrollo y despliegue de tecnologías audiovisuales en entornos web. Estas habilidades fueron fundamentales para la implementación exitosa de funcionalidades multimedia en este proyecto.

En resumen, la combinación de estos conocimientos y habilidades adquiridos a lo largo de mi formación académica me ha proporcionado las herramientas necesarias para llevar a cabo con éxito la elaboración de este Trabajo de Fin de Grado y realizar un análisis profundo y significativo del proyecto.

7.3. Lecciones aprendidas

Durante la realización de este proyecto, tuve la oportunidad de adquirir nuevos conocimientos y habilidades que no había explorado durante mi formación académica en la universidad. En particular, destacaría tres aspectos fundamentales:

1. **Uso de Docker:** A través de este proyecto, aprendí a utilizar Docker, una herramienta de virtualización que facilita la creación, implementación y ejecución de aplicaciones en entornos aislados. Esta experiencia me permitió comprender mejor los conceptos relacionados con la virtualización y la gestión de contenedores, así como mejorar mis habilidades en el despliegue de aplicaciones en diferentes entornos.
2. **Gestión de Base de Datos:** Durante la realización de este proyecto, adquirí experiencia en la gestión de bases de datos, incluyendo la creación de esquemas, la inserción y consulta de datos, y la optimización del rendimiento de las consultas. Esta experiencia fue fundamental para el desarrollo de funcionalidades relacionadas con el almacenamiento y recuperación de información en el sistema.
3. **Uso del Framework Flask:** Uno de los aspectos más destacados de este proyecto fue aprender a utilizar el framework Flask para el desarrollo de aplicaciones web en Python. A través de la práctica y la experimentación, adquirí habilidades en el diseño y desarrollo de aplicaciones web dinámicas y escalables. El uso de Flask me permitió implementar funcionalidades complejas de manera eficiente y mantener una estructura organizada en el código del proyecto.

En resumen, la realización de este proyecto me brindó la oportunidad de explorar y adquirir nuevos conocimientos y habilidades que no había tenido la oportunidad de desarrollar durante mi formación académica en la universidad. El aprendizaje de Docker, la gestión de bases de

datos y el uso del framework Flask fueron aspectos fundamentales que enriquecieron mi experiencia y ampliaron mis horizontes en el campo de la ingeniería de sistemas audiovisuales y multimedia.

7.4. Trabajos futuros

Este proyecto no solo marca un hito en su estado actual, sino que también sienta las bases para futuros desarrollos y expansiones. Se pretende convertir esta aplicación en un proyecto a futuro, con la meta de alojarla en un servidor accesible para cualquier usuario. Además, se planifica ampliar su funcionalidad mediante la incorporación de nuevas características y tecnologías.

Entre los trabajos futuros planeados para esta aplicación se incluyen:

- **Añadir redes neuronales mediante conectores:** Se explorará la posibilidad de integrar redes neuronales en la aplicación mediante conectores especializados. Esto permitirá mejorar el análisis y la interpretación de datos, así como ofrecer funcionalidades avanzadas basadas en inteligencia artificial.
- **Ampliar la gestión de bases de datos:** Se considerará la incorporación de la capacidad para administrar múltiples bases de datos, lo que ofrecerá mayor flexibilidad y escalabilidad al sistema. Esto incluirá la posibilidad de trabajar con bases de datos relacionales y no relacionales, según las necesidades específicas de los usuarios.
- **Innovar para convertirla en una plataforma completa:** Se buscará continuamente la innovación y el desarrollo de nuevas funcionalidades para convertir esta aplicación en una plataforma completa que cumpla con todos los requisitos y necesidades de los usuarios. Esto implica mantenerse al tanto de las últimas tendencias tecnológicas y adaptar la aplicación en consecuencia.

En resumen, se proyecta un futuro emocionante para esta aplicación, con la visión de convertirla en una herramienta robusta, versátil y completa que continúe creciendo y evolucionando para satisfacer las demandas cambiantes del mercado y las necesidades de los usuarios.

Apéndice A

Manual de usuario

A.1. Bienvenida

¡Bienvenido a nuestra aplicación! Este manual tiene como objetivo proporcionarte toda la información necesaria para utilizar nuestra plataforma de manera efectiva y segura.

A.2. Registro de Usuario

Para acceder a todas las funcionalidades de la aplicación, es necesario registrarse. Sigue estos pasos:

1. Accede a la página de registro.
2. Completa el formulario con tu dirección de correo electrónico y un nombre de usuario único.
3. Una vez registrado, recibirás un correo electrónico de validación. Sigue las instrucciones para activar tu cuenta.

Recuerda que cada cuenta solo puede tener una dirección de correo electrónico y un nombre de usuario únicos. El acceso a la aplicación será otorgado por el administrador.

A.3. Inicio de Sesión

Una vez que tu cuenta esté activada, puedes iniciar sesión en la aplicación:

1. Ingresa tu correo electrónico y contraseña en el formulario de inicio de sesión.
2. Si olvidaste tu contraseña, puedes solicitar una nueva. Se te enviará un enlace único a tu correo electrónico para restablecerla.

A.4. Gestión de Archivos

Una vez dentro de la aplicación, podrás gestionar tus archivos subidos y compartidos de la siguiente manera:

- En la sección de libros, podrás editar la información rellenando todos los campos requeridos. También podrás eliminar documentos como videos, PDF y la foto extra si no es necesario.
- Los enlaces y revistas no son editables y siempre deben tener todos los datos llenados. Si por algún motivo algún dato es erróneo podrás eliminarlos y volver a crearlos.

A.5. Calendario

Utiliza el calendario para programar eventos, reuniones y fechas importantes:

- Solo los profesores y administradores pueden editar y eliminar eventos del calendario. Los alumnos solo pueden crear eventos.

A.6. Comentarios

Los comentarios en la aplicación están sujetos a las siguientes reglas:

- Los comentarios pueden ser gestionados solo por usuarios registrados. Los administradores tienen la capacidad de eliminar comentarios inapropiados.

A.7. Contacto

Si tienes alguna pregunta o problema técnico, no dudes en ponerte en contacto. Envía un correo electrónico a karolbayas@gmail.com estare encantada de poder ayudarte a resolver tus dudas.

¡Gracias por utilizar nuestra aplicación!

Bibliografía

- [1] K. J. Bayas. Tfg-karol. <https://github.com/kjbayas/TFG-Karol>. Repositorio de GitHub que contiene el código fuente y la documentación del Trabajo de Fin de Grado (TFG) de Karol J. Bayas.
- [2] FullCalendar. Fullcalendar. <https://github.com/fullcalendar/fullcalendar>. FullCalendar es un proyecto de código abierto alojado en GitHub que proporciona una librería de JavaScript para crear calendarios interactivos en aplicaciones web.
- [3] M. Grinberg. *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media, 2018.
- [4] phpMyAdmin. phpMyAdmin. <http://localhost/phpmyadmin/index.php?route=/database/structure&db=sitio>. phpMyAdmin es una herramienta de administración de bases de datos MySQL alojada en un servidor local.