

UNIVERSIDADE DE SANTIAGO DE
COMPOSTELA



ESCOLA TÉCNICA SUPERIOR DE ENXEÑARÍA

Herramienta para la creación y edición colaborativa de documentos

Autor:

Pablo Tarrío Otero

Tutores:

Manuel Lama Penín

Juan Carlos Vidal Aguiar

Víctor José Gallego Fontenla

Grado en Ingeniería Informática

Julio 2022

Trabajo de Fin de Grado presentado en la *Escola Técnica Superior de Enxeñaría* de la *Universidade de Santiago de Compostela* para la obtención del Grado en Ingeniería Informática



D. Manuel Lama Penín, Profesor del Departamento de Electrónica y Computación de la *Universidade de Santiago de Compostela*, **D. Juan Carlos Vidal Aguiar**, Profesor del Departamento de Electrónica y Computación de la *Universidade de Santiago de Compostela*, y **D. Víctor José Gallego Fontenla**, Investigador Predoctoral FPU del *Ministerio de Ciencia, Innovación y Universidades*,

INFORMAN:

Que la presente memoria, titulada *Herramienta para la creación y edición colaborativa de documentos*, presentada por **D. Pablo Tarrío Otero** para superar los créditos correspondientes al Trabajo de Fin de Grado de la titulación de Grado en Ingeniería Informática, se realizó bajo nuestra tutoría en el Departamento de Electrónica y Computación de la *Universidade de Santiago de Compostela*.

Y para que así conste a los efectos oportunos, expiden el presente informe en Santiago de Compostela, a 26 de julio de 2022:

Tutor,

Cotutor,

Manuel Lama Penín

Juan Carlos Vidal Aguiar

Cotutor,

Alumno,

Víctor José Gallego Fontenla

Pablo Tarrío Otero

Agradecimientos

A mis tutores Manuel, Juan Carlos y Víctor, por haberme orientado durante estos meses a la hora de realizar el TFG. Sin ellos el resultado de este proyecto no habría sido tan satisfactorio para mí.

A mis amigos, que me han acompañado durante todos estos años y me han hecho pasar una etapa preciosa, que aunque ha sido corta, resultará inolvidable en mi vida.

A mi familia, por no dejar que me rindiese ante cualquier bache que se presentase, y por haberme animado durante toda mi vida desde que era un niño hasta el día de hoy.

Resumen

Este trabajo de fin de grado aborda el desarrollo de una herramienta colaborativa que permitirá la creación, edición y consolidación de documentos por parte de varios usuarios. Esta herramienta será integrada en una plataforma de consulta de legislación consolidada gallega, facilitando la importación de legislación desde el Diario Oficial de Galicia (DOG) [1], así como el mantenimiento con la aplicación en las sucesivas modificaciones que se vayan realizando.

De esta forma, los usuarios podrán importar documentos a la herramienta colaborativa a partir de leyes importadas del DOG. Por una parte, se permite la búsqueda de leyes y normas encontradas en el DOG filtrando según los parámetros deseados. Adicionalmente, podrá verse una previsualización en la página oficial del DOG. Dichos documentos encontrados en el DOG podrán ser importados a la herramienta colaborativa, donde se podrá proceder a su edición. En cuanto a la edición, se ofrece la posibilidad de proponer cambios a la norma que se va a editar, manipular las leyes vinculadas a esta, así como escribir posibles notas aclaratorias. Además, todo ello será gestionado por usuarios previamente registrados.

Índice general

1. Introducción	1
1.1. Descripción del problema	1
1.2. Objetivos	2
1.3. Estructura del documento	3
2. Especificación de requisitos	5
2.1. Requisitos funcionales	5
2.2. Requisitos no funcionales	6
2.3. Requisitos de información	6
2.4. Casos de uso	7
3. Diseño	8
3.1. Tecnologías empleadas	8
3.1.1. Tecnologías del frontend	8
3.1.2. Tecnologías del backend	9
3.2. Arquitectura del sistema	9
3.2.1. Arquitectura global	10
3.2.2. Arquitectura del servidor	11
3.2.3. Arquitectura del cliente	14
3.3. Patrones utilizados	15
3.3.1. Patrones de arquitectura	15
3.3.2. Patrones de diseño	17
3.4. Modelo de datos	18
3.4.1. User	18
3.4.2. FinalDocument	19
3.5. Diagramas de secuencia	20
3.5.1. Inicio de sesión	20
3.5.2. Buscar una ley en el portal de edición	22
3.5.3. Validar y publicar una ley en el portal de edición	23
3.6. Diseño de la interfaz	24

4. Pruebas	29
4.1. Pruebas unitarias	29
4.1.1. Casos de prueba	30
4.1.2. Matriz de trazabilidad	38
4.2. Pruebas de integración	39
4.3. Validación de la interfaz de usuario	43
4.3.1. Cuestionario SUS	44
4.3.2. Principios heurísticos de Nielsen	45
4.4. Validación de los resultados de las pruebas	47
5. Conclusiones y posibles ampliaciones	48
5.1. Conclusiones	48
5.2. Posibles ampliaciones	49
A. Manuales técnicos	51
A.1. Manual de despliegue	51
A.2. Versiones de las herramientas recomendadas	53
B. Manuales de usuario	55
B.1. Página de inicio de sesión	55
B.2. Página principal	56
B.3. Página de registro de usuarios	58
B.4. Pestaña para importar del DOG	58
B.5. Página de previsualización de una ley de lex.gal	60
B.6. Página de edición de una ley de lex.gal	61
B.7. Pestaña de edición de leyes vinculadas	66
B.8. Página no encontrada	66
C. Requisitos	69
C.1. Requisitos funcionales	69
C.2. Requisitos no funcionales	75
C.3. Requisitos de información	79
C.4. Casos de uso	80
D. Diagramas de secuencia	89
D.1. Inicio de sesión	89
D.2. Cierre de sesión	89
D.3. Registro de un usuario	90
D.4. Previsualizar una ley del DOG	90
D.5. Buscar una ley en el DOG	91
D.6. Importar una ley del DOG	91
D.7. Buscar una ley en lex.gal	92
D.8. Previsualizar una ley de lex.gal	92

D.9. Cargar una ley de lex.gal	92
D.10. Validar y publicar una ley de lex.gal	93
D.11. Guardar como borrador una ley de lex.gal	94
D.12. Eliminar una ley de lex.gal	94
E. Plantillas	95
E.1. Plantilla de requisitos funcionales	95
E.2. Plantilla de requisitos no funcionales	95
E.3. Plantilla de requisitos de información	96
E.4. Plantilla de casos de uso	96
E.5. Plantilla de pruebas unitarias	97
E.6. Plantilla de pruebas de integración	97
F. Licencia	99
Bibliografía	101

Índice de figuras

3.1. Arquitectura global del sistema.	10
3.2. Arquitectura del servidor.	11
3.3. API del servidor.	14
3.4. Arquitectura del cliente.	14
3.5. Patrón de arquitectura SPA.	16
3.6. Patrón de diseño Composite.	17
3.7. Patrón de diseño Front Controller.	17
3.8. Documento User.	18
3.9. Documento FinalDocument.	19
3.10. Diagrama de secuencia del inicio de sesión.	20
3.11. Diagrama de secuencia de la búsqueda de una ley en la aplicación.	22
3.12. Diagrama de secuencia de la validación y publicación de una ley en el portal.	23
3.13. Diseño inicial de la página de búsqueda de leyes en lex.gal.	25
3.14. Diseño final de la página de búsqueda de leyes en lex.gal.	25
3.15. Diseño inicial de la página de edición de una ley de lex.gal.	26
3.16. Diseño final de la página de edición de una ley de lex.gal - I.	26
3.17. Diseño final de la página de edición de una ley de lex.gal - II.	27
3.18. Diseño de la pestaña de edición de leyes vinculadas.	28
B.1. Página de inicio de sesión.	55
B.2. Página principal/búsquedas.	56
B.3. Menú de usuarios.	57
B.4. Página de registro de usuarios.	58
B.5. Pestaña para importar del DOG - I.	58
B.6. Pestaña para importar del DOG - II.	59
B.7. Previsualización de ley del DOG.	59
B.8. Página de previsualización de una ley de lex.gal.	60
B.9. Página de edición de una ley de lex.gal.	61
B.10. Datos de cabecera de una ley de lex.gal.	61
B.11. Menú de edición de un párrafo.	62
B.12. Pestaña de edición de un párrafo.	62
B.13. Pestaña con un cambio introducido.	63
B.14. Pestaña para añadir una nota a un párrafo.	63

B.15.Pestaña con una nota introducida.	64
B.16.Apartado de leyes vinculadas.	64
B.17.Mensaje de cambios guardados.	65
B.18.Mensaje de ley eliminada.	65
B.19.Pestaña de edición de leyes vinculadas.	66
B.20.Página no encontrada.	67
D.1. Diagrama de secuencia 01. Inicio de sesión.	89
D.2. Diagrama de secuencia 02. Cierre de sesión.	89
D.3. Diagrama de secuencia 03. Registro de un usuario.	90
D.4. Diagrama de secuencia 04. Previsualizar una ley del DOG.	90
D.5. Diagrama de secuencia 05. Buscar una ley en el DOG.	91
D.6. Diagrama de secuencia 06. Importar una ley del DOG.	91
D.7. Diagrama de secuencia 07. Buscar una ley en lex.gal.	92
D.8. Diagrama de secuencia 08. Previsualizar una ley de lex.gal.	92
D.9. Diagrama de secuencia 09. Cargar una ley de lex.gal.	92
D.10.Diagrama de secuencia 10. Validar y publicar una ley de lex.gal. . .	93
D.11.Diagrama de secuencia 11. Guardar como borrador una ley de lex.gal. .	94
D.12.Diagrama de secuencia 12. Eliminar una ley de lex.gal.	94

Índice de cuadros

4.1. Prueba unitaria 01. Inicio de sesión con parámetros correctos. . .	30
4.2. Prueba unitaria 02. Inicio de sesión con parámetros incorrectos. .	30
4.3. Prueba unitaria 03. Cierre de sesión.	30
4.4. Prueba unitaria 04. Registro de usuario con parámetros válidos. .	31
4.5. Prueba unitaria 05. Registro de usuario con parámetros inválidos.	31
4.6. Prueba unitaria 06. Búsqueda de leyes en el DOG con filtros válidos.	31
4.7. Prueba unitaria 07. Búsqueda de leyes en el DOG sin introducir texto.	32
4.8. Prueba unitaria 08. Previsualizar una ley en el DOG.	32
4.9. Prueba unitaria 09. Importar ley del DOG en lex.gal.	32
4.10. Prueba unitaria 10. El usuario busca una ley en lex.gal.	33
4.11. Prueba unitaria 11. Previsualizar una ley en lex.gal.	33
4.12. Prueba unitaria 12. Cargar los datos de una ley de lex.gal.	33
4.13. Prueba unitaria 13. Proponer un cambio en la ley principal de lex.gal.	34
4.14. Prueba unitaria 14. Descartar una selección de cambios de lex.gal.	34
4.15. Prueba unitaria 15. Descartar todos los cambios propuestos a una ley de lex.gal.	34
4.16. Prueba unitaria 16. Añadir una anotación sobre un párrafo de la ley principal de lex.gal.	35
4.17. Prueba unitaria 17. Descartar una selección de anotaciones de una ley de lex.gal.	35
4.18. Prueba unitaria 18. Descartar todas las anotaciones de una ley de lex.gal.	35
4.19. Prueba unitaria 19. Añadir un comentario sobre cualquier anotación.	36
4.20. Prueba unitaria 20. Cargar las leyes vinculadas a una ley de lex.gal.	36
4.21. Prueba unitaria 21. Añadir una ley vinculada a la lista de leyes vinculadas a una ley.	36
4.22. Prueba unitaria 22. Ir al párrafo donde se modifica una sección de la ley vinculada.	37
4.23. Prueba unitaria 23. Proponer un cambio sobre la ley vinculada. .	37
4.24. Prueba unitaria 24. Descartar un cambio sobre la ley vinculada. .	37
4.25. Prueba unitaria 25. Eliminar una ley de lex.gal.	38
4.26. Prueba unitaria 26. Validar y publicar una ley de lex.gal.	38

4.27. Prueba unitaria 27. Guardar como borrador una ley de lex.gal. . .	38
4.28. Matriz de trazabilidad de Pruebas unitarias-Requisitos funcionales.	39
4.29. Prueba de integración 01. Inicio de sesión.	40
4.30. Prueba de integración 02. Registro de usuario.	40
4.31. Prueba de integración 03. Búsqueda de una ley en el DOG.	40
4.32. Prueba de integración 04. Importar una ley del DOG a lex.gal. . .	41
4.33. Prueba de integración 05. Búsqueda de una ley en lex.gal.	41
4.34. Prueba de integración 06. Previsualizar una ley de lex.gal.	41
4.35. Prueba de integración 07. Comenzar a editar una ley de lex.gal. .	42
4.36. Prueba de integración 08. Comenzar a editar una ley vinculada a otra ley de lex.gal.	42
4.37. Prueba de integración 09. Validar y publicar una ley de lex.gal. . .	42
4.38. Prueba de integración 10. Guardar como borrador una ley de lex.gal.	43
4.39. Prueba de integración 11. Eliminar una ley de lex.gal.	43
4.40. Resultados del cuestionario SUS.	45
C.1. Requisito funcional 01. Iniciar sesión.	69
C.2. Requisito funcional 02. Cerrar sesión.	69
C.3. Requisito funcional 03. Registrar usuarios.	70
C.4. Requisito funcional 04. Búsqueda de leyes en el DOG.	70
C.5. Requisito funcional 05. Previsualizar leyes en el DOG.	70
C.6. Requisito funcional 06. Importar leyes del DOG.	71
C.7. Requisito funcional 07. Búsqueda de leyes en lex.gal.	71
C.8. Requisito funcional 08. Previsualizar leyes en lex.gal.	71
C.9. Requisito funcional 09. Cargar leyes de lex.gal.	72
C.10. Requisito funcional 10. Proponer cambios en una ley de lex.gal. .	72
C.11. Requisito funcional 11. Añadir anotaciones sobre párrafos de una ley de lex.gal.	72
C.12. Requisito funcional 12. Cargar leyes vinculadas a una ley de lex.gal.	73
C.13. Requisito funcional 13. Añadir leyes vinculadas a una ley de lex.gal.	73
C.14. Requisito funcional 14. Proponer cambios en leyes vinculadas a una ley de lex.gal.	73
C.15. Requisito funcional 15. Eliminar una ley de lex.gal.	74
C.16. Requisito no funcional 01. Tiempo de respuesta < 2 segundos. . .	75
C.17. Requisito no funcional 02. Los datos sensibles del usuario se alma- cenan encriptados.	75
C.18. Requisito no funcional 03. No permitir que un rol realice operacio- nes de otro.	75
C.19. Requisito no funcional 04. Permanecer operativo durante el horario de trabajo.	76
C.20. Requisito no funcional 05. Tratamiento de documentos en lengua- jes de etiquetas.	76

C.21.Requisito no funcional 06. Tiempo de aprendizaje por el usuario < 1h.	76
C.22.Requisito no funcional 07. Proporcionar mensajes de apoyo al usuario.	77
C.23.Requisito no funcional 08. Uso de tecnologías web y servicios REST-ful.	77
C.24.Requisito no funcional 09. Base de datos NoSQL.	77
C.25.Requisito no funcional 10. Generación de tokens JWT.	78
C.26.Requisito de información 01. Usuario.	79
C.27.Requisito de información 02. Documento.	79
C.28.Caso de uso 01. Iniciar sesión.	80
C.29.Caso de uso 02. Cerrar sesión.	80
C.30.Caso de uso 03. Registrar usuario.	81
C.31.Caso de uso 04. Importar ley del DOG.	82
C.32.Caso de uso 05. Previsualizar una ley del DOG.	83
C.33.Caso de uso 06. Previsualizar una ley en lex.gal.	84
C.34.Caso de uso 07. Validar y publicar una ley y sus leyes vinculadas en lex.gal.	85
C.35.Caso de uso 08. Guardar como borrador una ley y sus leyes vinculadas en lex.gal.	86
C.36.Caso de uso 09. Eliminar una ley en lex.gal.	87
E.1. Plantilla de requisitos funcionales.	95
E.2. Plantilla de requisitos no funcionales.	95
E.3. Plantilla de requisitos de información.	96
E.4. Plantilla de casos de uso.	96
E.5. Plantilla de pruebas unitarias.	97
E.6. Plantilla de pruebas de integración.	97

Capítulo 1

Introducción

1.1. Descripción del problema

Los portales de consulta de legislación son extremadamente importantes en cualquier ámbito, ya que permiten consultar cualquier resolución o ley publicada por parte de todos los públicos. Actualmente existen portales donde poder consultar las leyes publicadas por parte de las instituciones públicas, como puede ser el DOG, a nivel gallego, o el BOE, a nivel nacional. En todos los casos, debe ser necesario poder acceder a la ley consolidada, es decir, la última versión de la ley donde poder ver si se han realizado cambios, correcciones, añadido anotaciones, etc. Por ejemplo, en el BOE existe un portal [2] donde poder consultar dichas leyes. Es por ello que a nivel autonómico es necesaria la existencia de un sistema donde publicar las leyes consolidadas.

En cuanto a la creación de contenido, esta constituye uno de los problemas más importantes a los que se enfrentan muchos dominios de aplicación, entre ellos, la educación, la formación, generación de noticias, producción científica o la legislación. En estos dominios, el objetivo en la creación de los contenidos es doble: por una parte, creación de contenido de calidad, entendida como la fiabilidad, de modo que se minimicen los errores y que, si existen, sean sencillos de detectar; y, por, otra parte, la adaptación del contenido a los usuarios que los van a consumir, proporcionando herramientas que faciliten la comprensión de dicho contenido por parte de los usuarios de la aplicación.

Este Trabajo Fin de Grado (TFG) está centrado en el ámbito de la legislación y, más específicamente, en la creación y el consumo de textos legislativos (leyes o decretos) por parte de especialistas en el ámbito del dominio. Por lo tanto, la principal restricción es la generación de contenido fiable, ya que los usuarios a los que van destinados los documentos son del mismo ámbito y nivel de especialización y/o formación que los usuarios que los crean. Esta es una característica

clave que se tendrá muy en cuenta a la hora de diseñar el sistema.

Por otra parte, desde hace varias décadas se han propuesto una gran cantidad de estrategias o métodos para garantizar la fiabilidad de los textos, pero quizás la edición colaborativa de documentos ha sido la que ha tenido un mayor éxito y aceptación por parte de los usuarios [4]. Un ejemplo claro de ello es la Wikipedia [5], en la que cualquier usuario registrado puede proponer y generar cambios que, no obstante, deberán ser aceptados por los administradores a partir de los comentarios que se introducen en un panel de discusión al que podrán acceder todos los usuarios que hayan participado en la edición del contenido en cuestión [6]. Este tipo de edición ha proporcionado a la Wikipedia una gran calidad y fiabilidad como modo de transmisión de conocimiento [7].

El contexto en el que surge este TFG es el desarrollo de una herramienta para el portal de publicación de leyes consolidadas a nivel gallego, *lex.gal* [3]. La solución que se utiliza actualmente en este portal resulta muy tediosa para los usuarios. Algunos problemas que se presentan son la escasa usabilidad, el trabajo que deben realizar los usuarios manualmente es excesivo, siendo este susceptible a errores; necesidad de publicar la ley para poder consultar que los cambios se aplican correctamente (siendo necesario borrar la ley junto con sus cambios del portal en caso de haber cometido un fallo), etc.

Para solucionar todos los problemas posibles que se presentan en la aplicación que se utiliza hoy en día, en este TFG se desarrollará una herramienta colaborativa que permite la edición de documentos en el ámbito de la legislación. En este contexto, la colaboración se entiende como la edición por parte de varios usuarios, de modo que las modificaciones de unos se comuniquen a otros con el objeto de que puedan confirmarlas y/o discutir si esa modificación es pertinente o no. La implementación de este tipo de edición colaborativa tendrá en cuenta las características de los documentos legislativos, que tienen una estructura muy definitiva, a modo de artículos y apartados, en los que, además, se cruza mucha información de otros textos legislativos. En todo caso, la edición por parte de varios usuarios no será simultánea. Además, también se permitirá realizar una búsqueda de documentos en el DOG para poder importar las leyes deseadas, y así poder proceder a la edición del propio documento.

1.2. Objetivos

El objetivo principal del proyecto es el desarrollo de una herramienta de edición colaborativa de textos jurídicos. De forma más concreta, este objetivo general se puede dividir en los siguientes subobjetivos:

- **OBJ-1: Desarrollar una capa de servicios de gestión de textos jurídicos** que permita la creación, edición, indexado y anotación de documentos y que tenga en cuenta el acceso en modo colaborativo a dichos documentos por parte de varios usuarios.
- **OBJ-2: Desarrollo de una interfaz gráfica** que permita la creación avanzada de textos jurídicos a través de estrategias de edición, colaboración y generación de eventos que informen sobre las modificaciones que tengan lugar por parte de los usuarios.
- **OBJ-3: Validación de la interfaz gráfica** por parte de usuarios especialistas en edición y creación de textos jurídicos.
- **OBJ-4: Desarrollo de una capa de gestión de usuarios** donde se deben restringir las operaciones según los roles de un usuario, así como permitir el uso de la herramienta solamente a los usuarios que estén registrados en la aplicación.

1.3. Estructura del documento

La presente memoria recoge toda la documentación del proyecto realizado en este TFG. Esta se encuentra dividida en una serie de capítulos, los cuales pueden ser resumidos de la siguiente forma:

- **Capítulo 1: Introducción.** En este capítulo se aborda la descripción del problema y motivación del proyecto, los objetivos generales del mismo y la estructura de la memoria.
- **Capítulo 2: Especificación de Requisitos.** Esta sección del documento define los requisitos existentes que garantizarán la aceptación del proyecto por parte de los usuarios especialistas en textos jurídicos.
- **Capítulo 3: Diseño.** El diseño incluye el propio diseño de la aplicación y la comunicación entre sus diferentes componentes, la arquitectura del sistema, patrones de arquitectura y de diseño, explicaciones acerca de diagramas UML realizados y el diseño de la interfaz de usuario.
- **Capítulo 4: Pruebas.** Este capítulo contiene un informe acerca de las pruebas unitarias y de integración. También se ha validado la interfaz de usuario mediante un cuestionario SUS y los principios de Nielsen.
- **Capítulo 5: Conclusiones y posibles ampliaciones.** En este último capítulo se extraen diferentes conclusiones obtenidas tras la realización del TFG, así como posibles ampliaciones que se podrán realizar en un futuro.

■ Apéndices:

- Apéndice A: Manuales técnicos. Incluye un manual que contiene una explicación acerca de cómo instalar las diferentes dependencias de la aplicación, así como una guía para poder desplegarla correctamente en cualquier máquina.
- Apéndice B: Manuales de usuario. En este apéndice se describe cómo un usuario podrá utilizar la aplicación web que ha sido desarrollada.
- Apéndice C: Requisitos. Se muestran todos los detalles de los requisitos en sus respectivas tablas.
- Apéndice D: Diagramas de secuencia. Apéndice donde se pueden consultar las ilustraciones de todos los diagramas de secuencia del sistema.
- Apéndice E: Plantillas. Plantillas empleadas para la redacción de las tablas en la presente memoria.
- Apéndice F: Licencia. Se trata de la licencia de uso del software y documentación del propio TFG.
- Bibliografía. La lista de bibliografía empleada para la realización de la memoria de este TFG.

Capítulo 2

Especificación de requisitos

Para proceder a especificar los requisitos se ha realizado un profundo análisis de los objetivos de la aplicación. Estos requisitos se han buscado teniendo en cuenta aspectos funcionales, de rendimiento, de seguridad, de información, etc.

Las plantillas empleadas para las tablas de los requisitos y casos de uso se encuentran en el apéndice E.Plantillas. Estas plantillas están inspiradas en las tablas empleadas en el REM [8] para la gestión de requisitos.

La importancia de los requisitos será medida según el grado de la necesidad del requisito en la aplicación, siendo en el caso de este proyecto “Vital” para los requisitos de mayor importancia, y, “Quedaría bien” para los menos importantes. Por su parte, la urgencia, indicará aquellas partes que se han de implementar con mayor rapidez, siendo “Inmediatamente” las que tienen una mayor prioridad, y, “Puede esperar”, para los requisitos que no son prioritarios.

2.1. Requisitos funcionales

Los requisitos funcionales [9] son aquellos que definen una funcionalidad del sistema de software o sus componentes, tanto para las entradas de datos como para el comportamiento en situaciones específicas.

En el caso de esta aplicación se han localizado dos grupos de requisitos funcionales: los asociados con operaciones de usuarios, y los que se relacionan con operaciones asociadas a leyes.

En cuanto a requisitos establecidos para usuarios, encontramos tres requisitos funcionales: inicio de sesión, cerrar sesión y registrar usuarios. De todos ellos, resulta de vital importancia el requisito de iniciar sesión, pues es necesario para realizar operaciones en el sistema.

Por parte de los requisitos asociados a leyes, encontramos requisitos relacionados con búsquedas, importación, edición, previsualización y borrado. Los requisitos más importantes en este caso son los relacionados con la edición de un documento, ya que es la base de la herramienta realizada en este TFG.

Los requisitos funcionales pueden ser consultados en el Apéndice C.1. Requisitos funcionales.

2.2. Requisitos no funcionales

Con respecto a los requisitos no funcionales [9], estos son los requisitos que especifican criterios que pueden usarse para juzgar la operación de un sistema, es decir, las características de funcionamiento. Se engloban en este caso los que afectan al rendimiento, seguridad, restricciones, etc.

En cuanto a los requisitos no funcionales de este proyecto, se han localizado requisitos en cuanto a encriptación de datos, rendimiento del sistema, restricción de operaciones según los roles de un usuario, tecnologías a emplear y ayudas visuales para el usuario.

Los requisitos no funcionales pueden ser consultados en el Apéndice C.2. Requisitos no funcionales.

2.3. Requisitos de información

Por último, los requisitos de información [9], son aquellos requisitos que indican algún tipo de información, la cual es necesaria para el correcto funcionamiento del sistema.

En el caso de esta aplicación, se han definido dos: usuario y documento. El primero de ellos, almacenará toda la información relacionada con el usuario que utilice el sistema, incluyendo sus roles. El segundo, por su parte, almacena todos los datos relacionados con una ley, ya sean datos de cabecera, cambios aplicados o anotaciones hechas sobre esta.

Los requisitos de información pueden ser consultados en el Apéndice C.3. Requisitos de información.

2.4. Casos de uso

Un caso de uso [10] es cualquier acción o actividad que se realiza en el sistema. Representan diferentes operaciones que se realizan en la aplicación.

Los casos de uso identificados son operaciones relacionadas con los requisitos funcionales definidos anteriormente. Se dividen en tres grupos:

- Casos de uso relacionados con operaciones de usuario: iniciar sesión, cerrar sesión y registrar usuario.
- Casos de uso relacionados con operaciones del DOG: previsualizar una ley del DOG e importar ley del DOG.
- Casos de uso relacionados con el portal de edición: previsualizar, validar y publicar, guardar como borrador y eliminar leyes de lex.gal.

Los casos de uso pueden ser consultados en el Apéndice C.4. Casos de uso.

Capítulo 3

Diseño

En el apartado de diseño se hace mención a distintos aspectos de la aplicación relacionados con las tecnologías utilizadas, arquitectura del sistema, patrones arquitectónicos y de diseño, modelos de datos, diagramas de secuencia de las principales operaciones de la aplicación, así como el diseño de la interfaz de usuario. Gracias a todo ello, se podrá realizar una representación del software a desarrollar.

3.1. Tecnologías empleadas

En esta sección se realiza un breve resumen acerca de las diferentes tecnologías empleadas durante el desarrollo de la aplicación. Todas ellas las podemos dividir entre las tecnologías utilizadas para el *frontend*, como aquellas usadas en el *backend* [11].

3.1.1. Tecnologías del frontend

El *frontend* es la parte del sitio web con la que interactúan los usuarios, es decir, el lado del cliente. Entre las tecnologías que se han utilizado en esta parte, encontramos las siguientes:

- **React.js** [12]: uno de los frameworks más populares de JavaScript. Permite el desarrollo de interfaces de usuario interactivas de forma sencilla.
- **Tailwind** [13]: framework de CSS [14] que permite un desarrollo ágil basado en clases de utilidad. Se ha utilizado para estilizar de forma rápida y sencilla la interfaz del usuario.
- **Node.js** [15]: entorno de ejecución de JavaScript del proyecto.
- **Npm** [16]: gestor de dependencias de los paquetes del cliente. Entre las principales dependencias utilizadas destacan:

- **React Diff Viewer** [17]: componente de React que permite la visualización de las diferencias entre dos textos.
- **Heroicons** [18]: iconografía que permite ser gestionada mediante propiedades CSS de Tailwind.
- **Material UI** [19]: componentes e iconografía que han sido implementados con el objetivo de un desarrollo más rápido en páginas web. También constan de integración con Tailwind.

3.1.2. Tecnologías del backend

El *backend* es lo que conecta la base de datos con el servidor. Con todas estas tecnologías el usuario no tiene una interacción directa, si bien resultan imprescindibles para el correcto funcionamiento de la aplicación. Entre las tecnologías que se han utilizado en esta parte, encontramos las siguientes:

- **Spring Boot** [20]: framework que permite la creación de aplicaciones en Java de una forma más sencilla. Spring simplifica el proceso de gestión de las dependencias, permitiendo que nos centremos lo máximo posible en el desarrollo de la aplicación.
- **Gradle** [21]: gestor de dependencias en aplicaciones de Java. Permite automatizar además el proceso de compilación, permitiendo configuraciones personalizadas y flexibles.
- **MongoDB** [22]: base de datos NoSQL [23]. Este tipo de bases de datos se caracteriza por estar orientadas hacia documentos, almacenando y recuperando datos en formatos que no son tablas.
- **Swagger** [24]: utilizado para generar toda la documentación de la API RESTful implementada en el servidor. Expone los servicios implementados en el servidor, permitiendo ver posibles valores a introducir, y conocer posibles respuestas de cada servicio.

3.2. Arquitectura del sistema

En cuanto a la arquitectura del sistema, se explica en tres secciones. La primera de ellas hará hincapié en como se conecta todo con respecto al sistema globalmente. En la segunda sección, por su parte, se explicará como se ha arquitecturizado el servidor web. Por último, se explica la arquitectura del cliente.

3.2.1. Arquitectura global

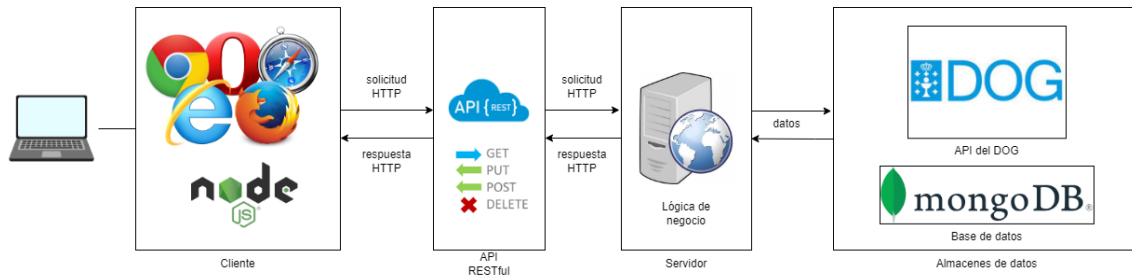


Figura 3.1: Arquitectura global del sistema.

En esta aplicación se utiliza una arquitectura basada en servicios REST [25]. Este tipo de arquitecturas se caracterizan por el acceso a recursos de forma simple mediante solicitudes HTTP, además de permitir un desacople entre interfaz y servidor. La arquitectura del sistema es ilustrada en la Figura 3.1.

El cliente está conformado por cualquier interfaz ejecutada en un navegador web, desde la que se realizan llamadas a la API del servidor mediante su propia interfaz. El usuario accede a cualquier navegador y puede utilizar la aplicación. No obstante, la interfaz ha sido desarrollada para las características de versiones modernas de JavaScript. Es por ello que para poder utilizar esta aplicación se han de utilizar navegadores evergreen [26] (que se actualizan automáticamente a sus futuras versiones).

Desde el cliente se realizan solicitudes HTTP hacia la API RESTful, que expone todos los servicios implementados en el servidor. Gracias al uso de la arquitectura REST, se permite el acceso a los recursos mediante URIs, las cuales pueden ser accedidas mediante los verbos HTTP (GET para obtener información, POST para crear objetos, PATCH y PUT para modificar información, y DELETE para eliminar objetos). Al cliente se le devuelve desde la API una respuesta HTTP con todas sus cabeceras. Y, además, en el caso de las operaciones GET, POST, PATCH y PUT, se devuelve el objeto correspondiente a la operación, ya se haya obtenido, creado o modificado.

La API, como se ha mencionado anteriormente, expone los servicios que implementan la lógica de negocio de la aplicación [27] y son ejecutados en el servidor.

En cuanto a la información que se utiliza en la aplicación, esta puede ser obtenida de dos fuentes diferentes. Por un lado, se obtiene información del DOG a través de su API REST, la cual se corresponde con las leyes que todavía no están almacenadas en la base de datos de la aplicación. El otro método para obtener

información es directamente de la base de datos de MongoDB, donde se encuentran las leyes consolidadas.

Con esta arquitectura se ha logrado el cumplimiento de dos requisitos no funcionales descritos, el **NFR-08: Uso de tecnologías web y servicios RESTful** y el **NFR-09: Base de datos NoSQL**.

3.2.2. Arquitectura del servidor

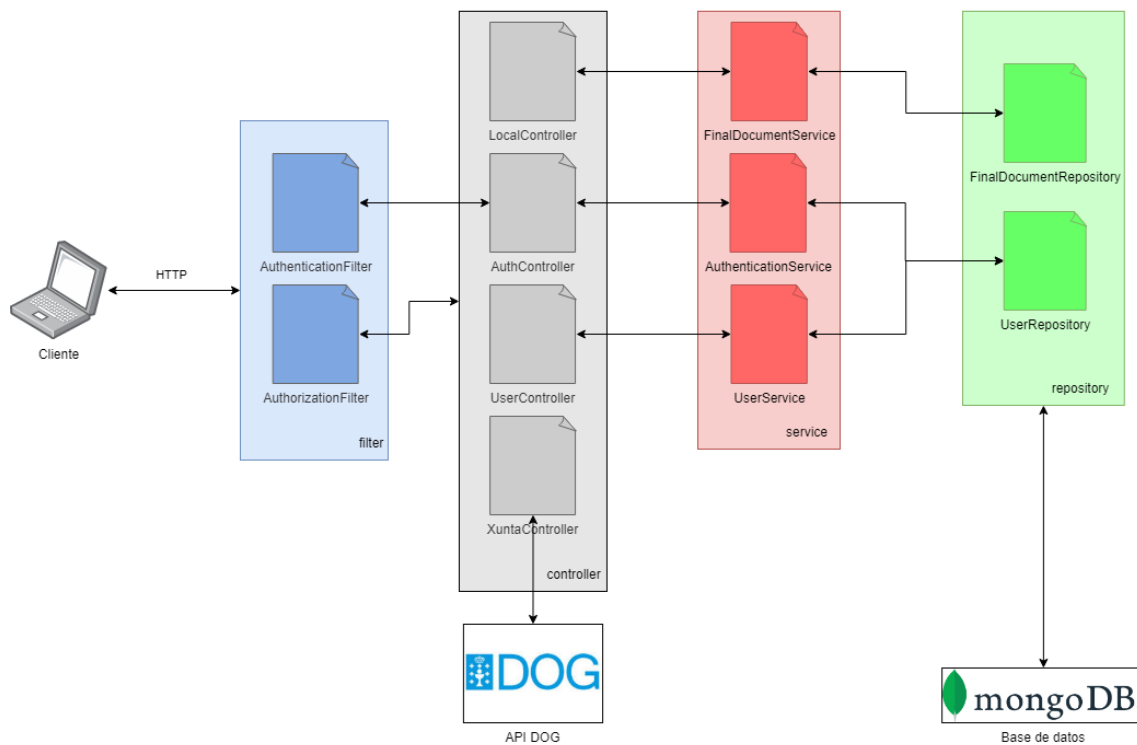


Figura 3.2: Arquitectura del servidor.

La arquitectura del servidor se muestra en la Figura 3.2, y en ella se pueden ver a simple vista los diferentes componentes que la forman.

El usuario realiza desde el cliente las llamadas a la API mediante solicitudes HTTP (ya sean GET, POST, ...), de las que posteriormente obtendrá una respuesta. Una vez llegan las solicitudes al servidor, se hacen pasar por los filtros (*filter*). Encontramos por un lado el componente **AuthenticationFilter**, que se encarga de gestionar el inicio de sesión, y, por otra parte, **AuthorizationFilter**, que comprueba los permisos del usuario, así como si el token de la sesión es válido.

Una vez pasan los filtros, la información se redirige hacia los controladores (*controller*). Estos controladores redirigen las llamadas a los servicios necesarios (*service*), que es donde se implementa la lógica de negocio en la aplicación. Por último, los servidores invocan a los métodos de los repositorios (*repository*), que son los que se encargan del acceso y almacenamiento en la base de datos.

Teniendo en cuenta la división en componentes realizada, las funcionalidades de los distintos controladores, servicios y repositorios se pueden resumir en:

- **LocalController, FinalDocumentService y FinalDocumentRepository:** se encargan de gestionar las operaciones relacionadas con las leyes almacenadas en la base de datos de la aplicación. Desde LocalController se llama a los servicios necesarios de FinalDocumentService, el cual finalmente llama a FinalDocumentRepository para que maneje los datos de las leyes en la base de datos.
- **UserController, UserService y UserRepository:** se encargan de gestionar las operaciones relacionadas con los usuarios, los cuales se almacenan en la base de datos de la aplicación. Desde UserController se llama a los servicios necesarios de UserService, el cual finalmente llama a UserRepository para que maneje los datos de los usuarios en la base de datos.
- **AuthController, AuthenticationService y AuthenticationFilter:** se encargan de gestionar las operaciones relacionadas con aspectos de seguridad respecto a los usuarios, como el inicio de sesión. Acceden también a los usuarios almacenados en la base de datos local. Desde AuthController se llama a los servicios necesarios de AuthenticationService, el cual finalmente llama a AuthenticationFilter para que maneje los datos de las leyes en la base de datos. En este caso es necesario utilizar el filtro de AuthenticationFilter, pues se encarga del inicio de sesión.
- **XuntaController:** se encarga de gestionar las llamadas a la API del DOG. De aquí el controlador XuntaController obtiene las diferentes leyes encontradas en el DOG.

Listado de servicios

Como se ha mencionado anteriormente, el servidor sigue una arquitectura basada en servicios REST, los cuales permiten el acceso a recursos mediante los verbos HTTP. La información completa de dichos servicios se puede consultar en la documentación de la API del servidor en la URI `/swagger-ui/index.html`. Por ejemplo, en el caso de desplegar el servidor localmente, el enlace sería `http://localhost:8080/swagger-ui/index.html`. Se expone un resumen de los servicios a continuación:

- **Xunta API:** existe una única operación GET, la cual se encarga de devolver el listado de leyes encontradas en el DOG. Para poder utilizar este servicio, es necesario estar autenticado. La URI de acceso es */xunta*.
- **Authentication API:** una única operación también, en este caso, un POST. Se encarga de gestionar el inicio de sesión por parte de un usuario, devolviendo un token JWT [28] para mayor seguridad (**NFR-10**). Es la única operación donde no es necesario estar autenticado y la URI de acceso es */login*.
- **Local API:** se encuentran aquí los servicios más importantes de la aplicación, y por ello tenemos muchos más que en el resto de controladores. Solamente es necesario estar autenticado para poder operar con estos servicios, y se explican a continuación:
 - Cuatro servicios GET. El primero de ellos, al cual se accede mediante la URI */local* devuelve todas las leyes encontradas. Hay otros dos (*/local/sumario* y */local/id*) que permiten buscar una ley concreta a partir de su sumario e id respectivamente. Por último, */local/sumario/htmlDoc*, que devuelve el documento HTML de una ley.
 - Un POST para crear objetos de leyes. El contenido del documento de una ley se almacena en formato HTML, cumpliendo con el requisito **NFR-05**. Se accede mediante la URI */local*.
 - Las operaciones PUT en la URI */local/id* y PATCH */local/sumario* para modificar los datos de documentos. La operación PUT se utiliza cuando es necesario modificar un documento en su totalidad, mientras que PATCH solo cambia los atributos que es necesario.
 - El servicio DELETE en la URI */local/id*, que se encarga de eliminar una ley de la base de datos.
- **User API:** dos operaciones para gestionar usuarios. La primera de ellas es un GET a la URI */users/id* para obtener los datos de un usuario, y solo es necesario estar autenticado para obtener dichos datos. La segunda, es un POST a la URI */users*. Esta operación inserta un usuario con la contraseña encriptada (**NFR-02**), así como solo puede ser realizada por usuarios que poseen el rol de administrador (**NFR-03**).

Xunta API Operacións realizadas sobre documentos recuperados da Xunta		
GET	/xunta	Obter a lista de normas dunha búsqueda no DOG.
Authentication API Operacións de autenticación.		
POST	/login	Inicio de sesión.
Local API Operacións realizadas sobre documentos almacenados localmente.		
PUT	/local/{id}	Modificar os datos dun documento.
DELETE	/local/{id}	Elimina un documento.
GET	/local	Obter as normas importadas a lex.gal.
POST	/local	Guardar os datos dun documento.
GET	/local/{sumario}	Obter os datos dun documento.
PATCH	/local/{sumario}	Modificar os datos dun documento.
GET	/local/{sumario}/htmlDoc	Obter o código HTML dun documento.
GET	/local/id/{id}	Obter os datos dun documento.
User API Operacións realizadas sobre usuarios.		
POST	/users	Registrar un usuario.
GET	/users/{id}	Obtén os datos dun usuario.

Figura 3.3: API del servidor.

En la Figura 3.3 se pueden observar los distintos servicios en la plataforma de Swagger.

3.2.3. Arquitectura del cliente

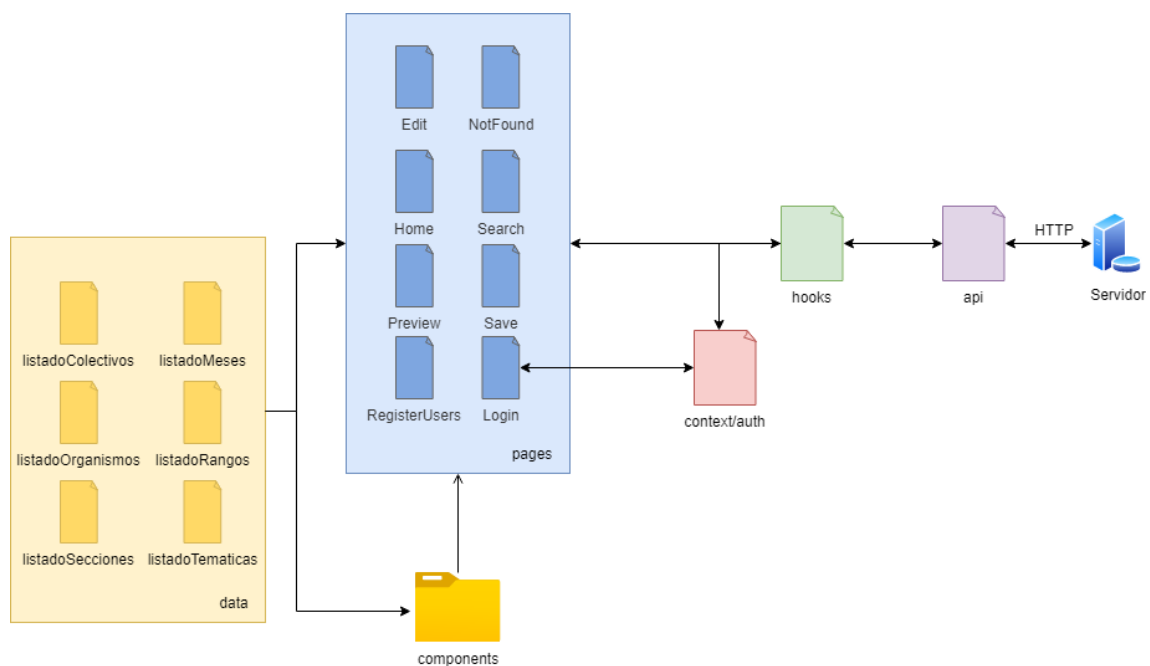


Figura 3.4: Arquitectura del cliente.

En la Figura 3.4 se ilustra una pequeña representación de la arquitectura del cliente, con las diferentes partes que lo componen.

En el cliente existen un total de ocho páginas: **Edit**, **NotFound**, **Home**, **Search**, **Preview**, **Save**, **RegisterUsers** y **Login**. Todas estas páginas son divididas en componentes, los cuales se almacenan en **components** (no se ilustran en esta imagen de la arquitectura porque son más de 50). Estos componentes son estructuras de código más pequeñas, que, en conjunto, llegan a formar las páginas mencionadas anteriormente.

Las distintas partes de las páginas, así como las propias páginas, hacen uso de los componentes de *data*, es decir, los mostrados en el cuadro amarillo. Estos son *componentess* que permiten automatizar el listado de selección de campos de cabecera, como puede ser el organismo de una ley, la sección a la que pertenece, el colectivo que lo publica, etc.

En cuanto a las páginas, estas hacen uso del componente **hooks** para poder llegar a establecer conexión con los datos del servidor. Las páginas hacen un llamamiento a *hooks*, que a su vez realiza un llamamiento al componente **api**, el cual establece una interfaz con todas las solicitudes HTTP realizadas al servidor. Este componente realiza las solicitudes al servidor en primera instancia, y recibe la respuesta correspondiente del servidor, la cual se envía de vuelta hasta la página correspondiente, siguiendo el camino inverso.

Existe un caso especial con respecto al párrafo anterior. Cuando se realiza la operación de inicio de sesión, esta se realiza en la página de Login. En este caso, se hace un llamamiento al componente **auth** del *context*. Este componente se encarga de gestionar el almacenamiento local en la parte del cliente. Aquí se almacena, por ejemplo, el token JWT que fue enviado desde el servidor al realizar el inicio de sesión. De esta forma, en todo momento el usuario dispone del token necesario para realizar las operaciones que desee al estar autenticado.

3.3. Patrones utilizados

Se explican a continuación los patrones empleados durante el desarrollo de la aplicación. Se dividen en patrones de arquitectura y patrones de diseño.

3.3.1. Patrones de arquitectura

Arquitectura orientada a servicios

La arquitectura orientada a servicios [29] es un estilo de arquitectura caracterizado por la orientación de servicios en servidores web. Un servicio es una unidad

autónoma de una o más funciones del software diseñado para realizar una tarea específica, como recuperar información o la ejecución de operaciones.

Además, la arquitectura orientada a servicios tiene como ventaja el establecimiento de una alta interoperabilidad. Esto se debe a que los entornos SOA (*Service Oriented Architecture*) carecen de sentido individual, pues normalmente se trata de un conjunto de servicios, creando sistemas heterogéneos. Como consecuencia de esto, existe una fácil escalabilidad y mantenimiento, ya se cada servicio se implementa por separado.

Otra de las ventajas que permite es que se pueden utilizar lenguajes de programación completamente diferentes en el lado del cliente y en el servidor al realizar llamadas HTTP. Esto no ocurre, por ejemplo, en el caso de los *Servlets* [30] o *Django* [31], donde la página web se genera en JAVA o Python respectivamente, limitándolos a utilizar un único lenguaje de programación.

Arquitectura SPA

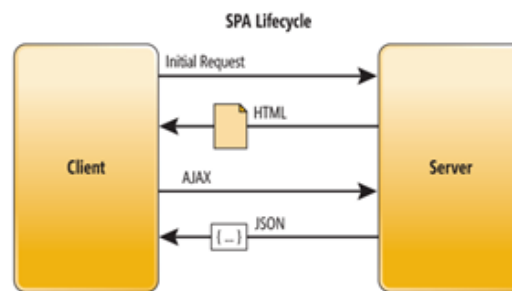


Figura 3.5: Patrón de arquitectura SPA.

La arquitectura SPA (*Single Page Application*) [32], reflejada en la Figura 3.5, es conocida por realizar la navegación desde el lado del cliente, mientras que el navegador no se recarga durante su uso.

Una de las principales ventajas que tiene con respecto a las aplicaciones tradicionales es que permite realizar cualquier aplicación de escritorio vía web, siendo el tiempo de respuesta mucho más rápido. Esto es así porque no tiene que ir a buscar continuamente la información al servidor. Además, se reduce la carga del servidor al realizar todas las operaciones sobre el cliente. Al generarse la aplicación en el navegador web, solo es necesario ir llamando a la API cuando se necesita hacer uso del servidor.

3.3.2. Patrones de diseño

Composite

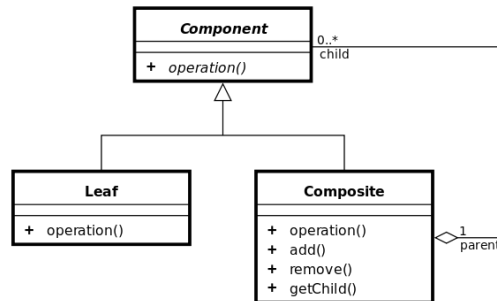


Figura 3.6: Patrón de diseño Composite.

En esta aplicación se ha utilizado el patrón Composite [33], el cual se refleja en la Figura 3.6, que se trata de un patrón estructural. Este permite la composición de objetos complejos a partir de otros más pequeños.

El ejemplo de este patrón de diseño es muy sencillo. Todas las páginas están formadas por distintos componentes: una cabecera, barra lateral, el propio cuerpo de la página, etc. En su conjunto, todos los componentes forman la página.

Front Controller

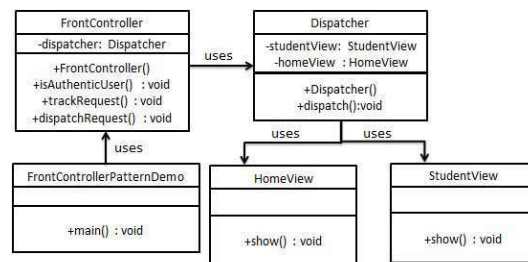


Figura 3.7: Patrón de diseño Front Controller.

Otro patrón estructural utilizado es el Front Controller [34], ilustrado en la Figura 3.7. Este patrón es muy utilizado comunmente en las aplicaciones web, en las que una clase se encarga de recibir las entradas, llamando a los métodos necesarios para generar las salidas.

En el caso de esta aplicación, la detección de este patrón es sencilla de ver. Se utilizan hasta cuatro controladores que llaman a sus respectivos servicios, donde

se implementa la lógica de negocio. Estos controladores reciben las solicitudes HTTP desde el cliente, centralizando las llamadas, y solicitan las salidas a los diferentes servicios. Los controladores son: **LocalController**, **XuntaController**, **AuthController** y **UserController**.

State

En esta aplicación también se ha utilizado el patrón State [35], que se trata de un patrón de comportamiento. El objetivo es que el comportamiento de un objeto depende de su estado y cambia en tiempo de ejecución.

En React, la información se almacena localmente en estados (*state*). Cada vez que se actualiza un estado, varía el objeto que está referenciando dicho estado. En el caso de estar utilizando varios componentes para formar una página, cada vez que un componente se actualice, también lo hace la página que lo contiene.

Hook

El patrón Hook [36] es también un patrón de comportamiento. Este es empleado para establecer un estado a los objetos de React sin ser necesario escribir una clase nueva.

En toda la parte del cliente se han utilizado los hooks. Cada vez que se actualiza un estado, se está utilizando un hook nuevo. Además, en todas las llamadas a la API, se hace uso de la carpeta **hooks**, la cual contiene hooks que actualizan el estado de elementos de la página, como puede ser una ley, los datos de un usuario, etc.

3.4. Modelo de datos

En este apartado se muestra la estructura de los datos almacenados en la base de datos. Estos son denominados documentos, y todos sus datos permitirán representar el diseño de las clases con sus respectivos atributos.

3.4.1. User

User
email: String nome: String apelidos: String password: String roles: List<String>

Figura 3.8: Documento User.

La Figura 3.8 ilustra las características del documento *User* de la aplicación. Está formado por los siguientes atributos:

- **email:** Correo electrónico del usuario.
- **nome:** Nombre del usuario.
- **apellidos:** Apellidos del usuario.
- **password:** Contraseña del usuario. Se almacena encriptada.
- **roles:** Roles del usuario. Solo existen dos posibles valores: “ROLE_USER” y “ROLE_ADMIN”.

3.4.2. FinalDocument

Se corresponde con todos los datos almacenados acerca de una ley. Es un objeto complejo constituido por otros más pequeños.

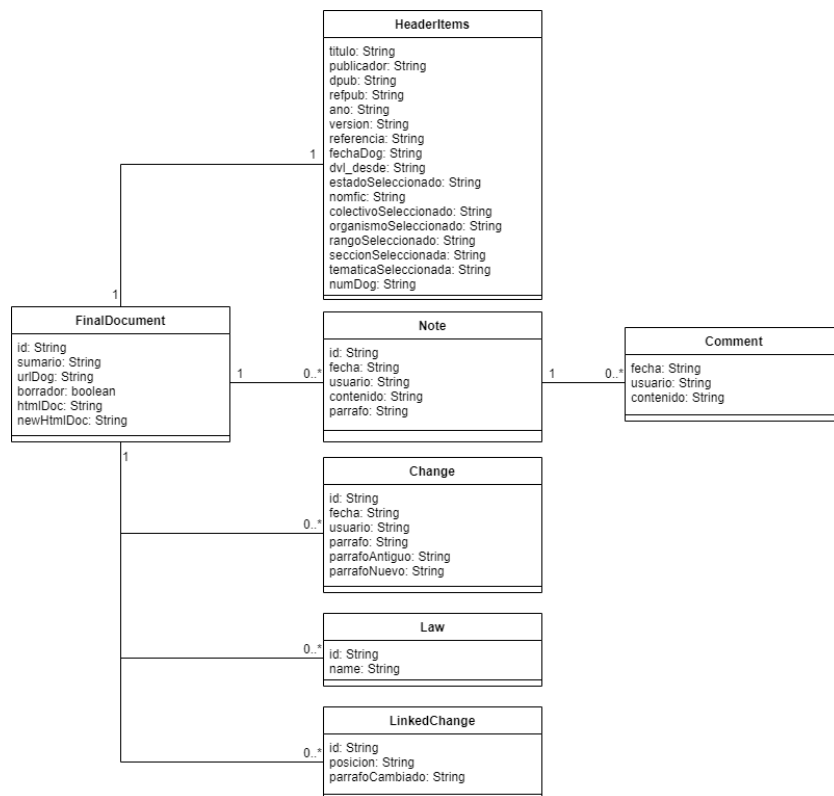


Figura 3.9: Documento FinalDocument.

La Figura 3.9 representa como se constituye el documento *FinalDocument* de la aplicación. Una ley está formada por un id, sumario, enlace de la ley en el

DOG, indicador de si la ley es un borrador, documento HTML, y el documento HTML modificado. A su vez, esta contiene un conjunto de datos de cabecera (HeaderItems), que son metadatos que conforman la ley. Entre ellos encontramos, por ejemplo, el título, publicador, fecha de publicación en el DOG, etc.

Las leyes pueden contener o no por una lista de notas (Note). Sus atributos son el id, fecha, usuario, contenido, y el párrafo sobre el que se realizan. Estas a su vez pueden contener una lista de comentarios (Comment), que contienen la fecha, usuario y contenido del propio comentario. También pueden tener o no cambios (Change), donde se indica su id, fecha, usuario, párrafo al que afectan, párrafo antiguo y el párrafo nuevo.

Con respecto a las leyes vinculadas (Law), se almacena su id y el nombre. Sobre estas se pueden realizar cambios (LinkedChange), donde se almacena el id, párrafo al que afectan, y el contenido del nuevo párrafo.

3.5. Diagramas de secuencia

Los diagramas de secuencia se utilizan para modelar la interacción entre las distintas partes de un sistema según UML. De esta forma, se podrá observar de una forma más ilustrativa el funcionamiento de la aplicación, y las interacciones del usuario con esta última.

Se han realizado diagramas de secuencia para todas las operaciones realizadas en el sistema, y se pueden observar detenidamente en el Apéndice D. Diagramas de secuencia. Para explicar en mayor detalle estos, se explican a continuación los diagramas más importantes y/o representativos de la aplicación.

3.5.1. Inicio de sesión

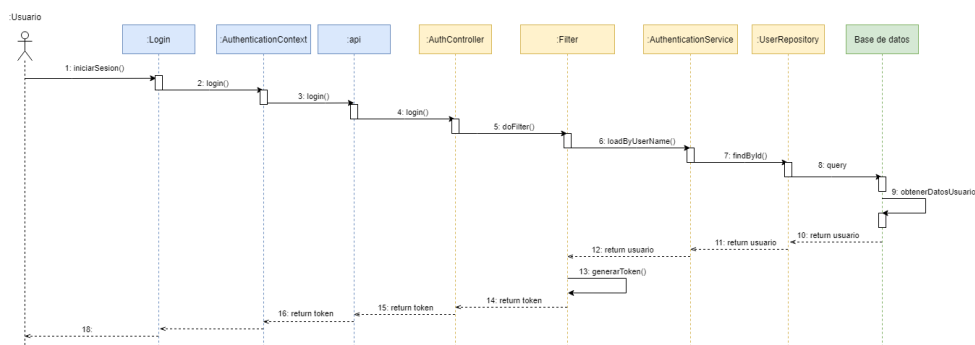


Figura 3.10: Diagrama de secuencia del inicio de sesión.

El inicio de sesión es una de las funcionalidades más importantes del sistema. Si un usuario no tiene iniciada la sesión, no se permitirá en ningún caso el uso de la aplicación. El inicio de sesión se realizará mediante el correo electrónico y la contraseña de un usuario, y el diagrama de secuencia correspondiente se puede observar en la Figura 3.10. Se describe a continuación el proceso:

1. El usuario, tras introducir su correo y contraseña, pincha en iniciar sesión en la página de **Login**.
2. Se llama a la función *login()* de **AuthenticationContext**, que es donde se gestionan todos los datos de inicio de sesión de un usuario en el cliente.
3. Se hace un llamamiento al método *login()* de la carpeta **api**, para que se localice el método en el servidor.
4. **AuthController** detecta que se desea iniciar sesión porque recibe una solicitud HTTP de tipo POST a su método *login()*.
5. Se hacen pasar los datos del usuario por los filtros de la carpeta **Filter** para comprobar que los datos son correctos.
6. Para que los filtros puedan comprobar que los datos son correctos, deben obtener los datos del usuario, por ello, hacen un llamamiento al método *loadByUsername()* de **AuthenticationService**.
7. Se busca el usuario en el repositorio **UserRepository** mediante el método *findById()*.
8. El repositorio hace una *query* a la base de datos de MongoDB invocando al método de obtención de usuarios.
9. La base de datos busca el usuario correspondiente en la base de datos.
10. En caso de encontrar el usuario, este se devuelve a los filtros.
11. Si los datos de inicio de sesión son correctos, se genera un token JWT de inicio de sesión.
12. Este token se devuelve hasta **AuthenticationContext**, para que se encargue de almacenar los datos en el almacenamiento local del cliente.
13. Por último, se indica al usuario que el inicio de sesión fue realizado.

3.5.2. Buscar una ley en el portal de edición

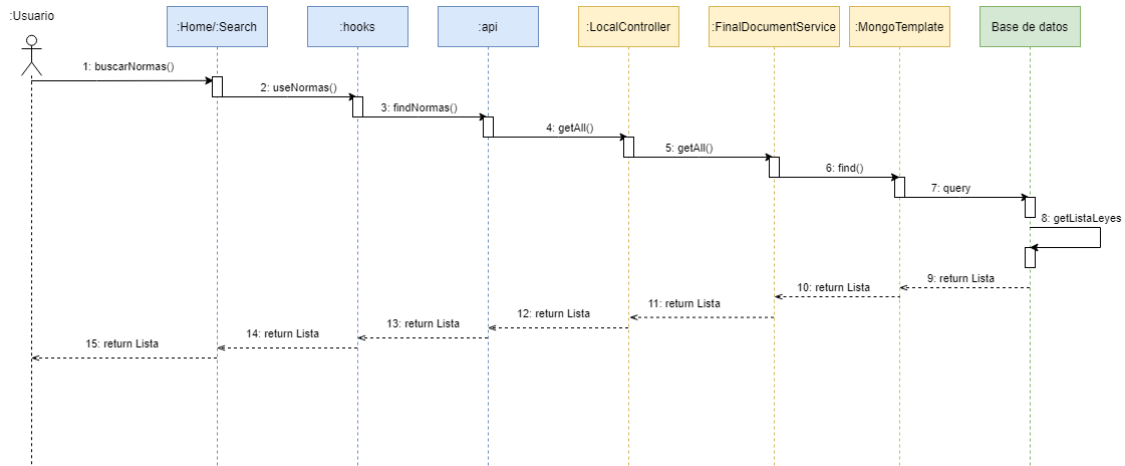


Figura 3.11: Diagrama de secuencia de la búsqueda de una ley en la aplicación.

En la Figura 3.11 se representa la operación de búsqueda de leyes en la base de datos de la aplicación. En este caso, puede ser que se encuentren o no leyes, devolviendo las encontradas en el primero de los casos. Esta operación se puede realizar desde las páginas **Home** y **Search** del cliente. Se puede resumir en los siguientes pasos:

1. El usuario pincha en buscar leyes en la página de búsqueda, con el texto de sumario si este ha escrito algo, o sin él.
2. Este método invoca al hook *useNormas()* del archivo contenido en **hooks**.
3. Se invoca al método *findNormas()* de la carpeta **api**.
4. Se hace un llamamiento mediante una solicitud HTTP al servidor. En concreto, a la función *getAll()* de **LocalController**, pues se hace un GET. Se comprueba en los filtros que el usuario está autenticado.
5. Se llama al servicio *getAll()* de **FinalDocumentService**.
6. Se invoca la operación *find()* de **MongoTemplate**, pues se busca según un texto, páginas y también tamaño de búsqueda.
7. Se invoca la operación de recuperar en la base de datos de MongoDB.
8. Se recuperan las leyes encontradas en la base de datos.
9. Se procede a devolver la respuesta al usuario desde este paso, realizando el proceso inverso. El usuario recibirá las leyes encontradas de forma paginada, según el texto de sumario que había escrito.

3.5.3. Validar y publicar una ley en el portal de edición

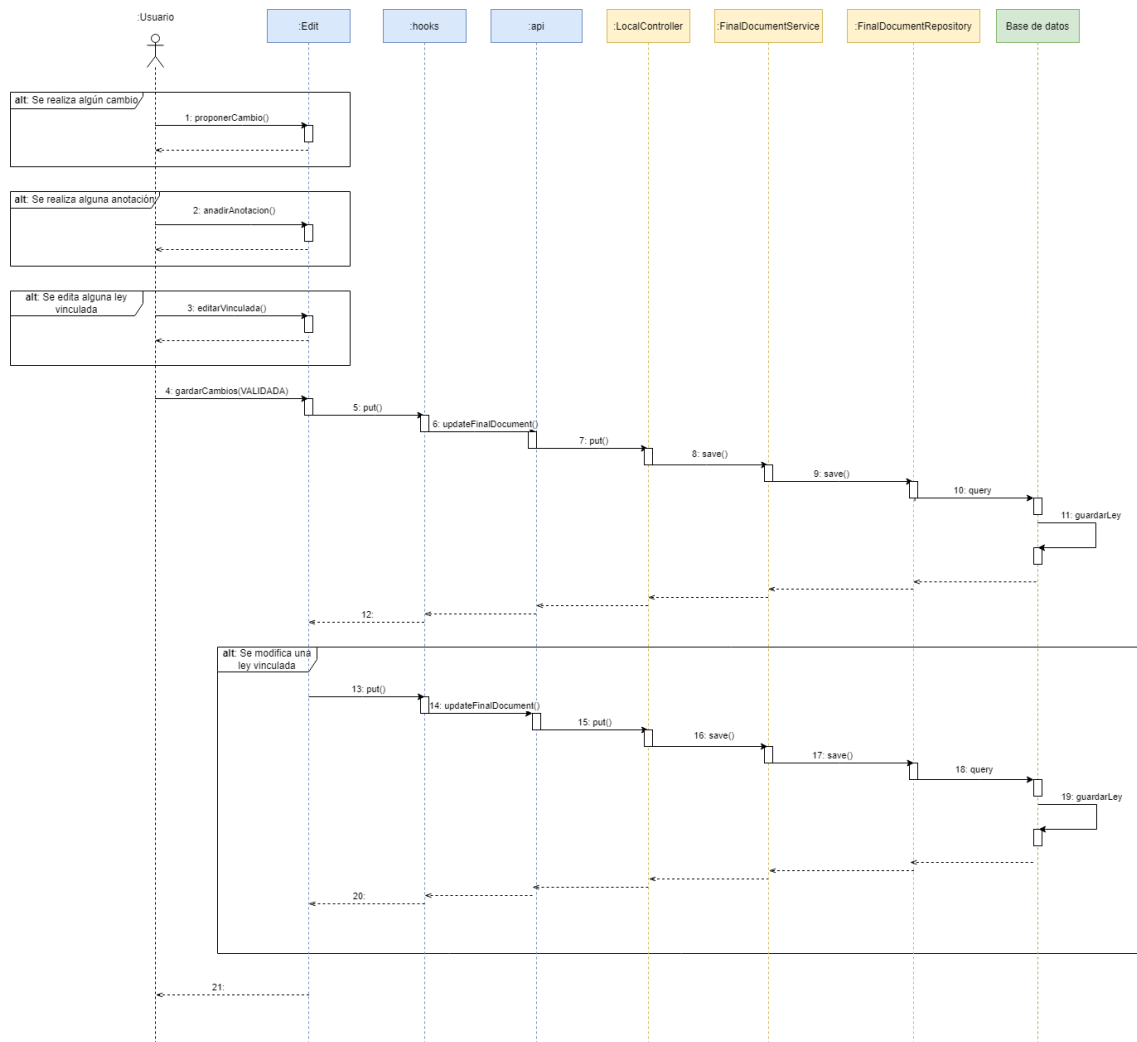


Figura 3.12: Diagrama de secuencia de la validación y publicación de una ley en el portal.

La principal funcionalidad de esta aplicación es la de editar una ley de lex.gal. En esta, un usuario estará visualizando una ley de lex.gal, y procederá a editar sus datos, así como los de las leyes vinculadas. En la Figura 3.12 se ilustra el diagrama de secuencia de la operación de validar y publicar una ley de lex.gal, que se puede resumir en los siguientes pasos:

1. El usuario puede proponer o no cambios en la página **Edit**.
2. El usuario puede añadir o no anotaciones en la página **Edit**.
3. El usuario propone cambios o no en leyes vinculadas en la página **Edit**.

4. El usuario guarda los cambios. Se utiliza el estado de **VALIDADA** para la ley principal y, en caso de modificar alguna, a sus leyes vinculadas.
5. Este método invoca al método *put()* del hook *useFinalDocument()*.
6. Se invoca al método *updateFinalDocument()* de la carpeta **api**.
7. Se hace un llamamiento mediante una solicitud HTTP al servidor. En concreto, la función *put()* de **LocalController**, pues se hace un PUT. Se comprueba en los filtros que el usuario está autenticado.
8. Se invoca al servicio *save()* de **FinalDocumentService**.
9. Se invoca la operación *save()* del repositorio **FinalDocumentRepository**, para poder modificar una ley.
10. Se invoca la operación de modificación en la base de datos de MongoDB.
11. Se modifica la ley correspondiente en la base de datos.
12. Se procede a devolver la respuesta al cliente desde este paso, realizando el proceso inverso.
13. En caso de que todo fue correctamente en los pasos anteriores, y que la ley principal modifica alguna ley vinculada, se procede a editar esta correspondientemente. Se repiten todos los pasos descritos desde el punto 5 al punto 12 para la ley vinculada.
14. Por último, se indica al usuario que se validó y publicó la ley principal, así como las leyes vinculadas si corresponde.

3.6. Diseño de la interfaz

En esta sección se describe como se ha diseñado la interfaz de usuario de la aplicación. En primer lugar, se realizó un *mockup* [37] de un par de páginas de la aplicación para comprobar que la apariencia era la deseada por parte de los expertos jurídicos.

La estructura y guía de colores están basados en los utilizados en la parte pública del portal. En esta, se utilizan como colores principales el azul, blanco, gris y negro. En cuanto a la estructura, se coloca a la izquierda de todas las páginas una barra lateral donde se encontrará el nombre de la aplicación, así como distintas opciones a realizar dentro de la página. Por su parte, el contenido en todas ellas abarcará todo el resto de la página.

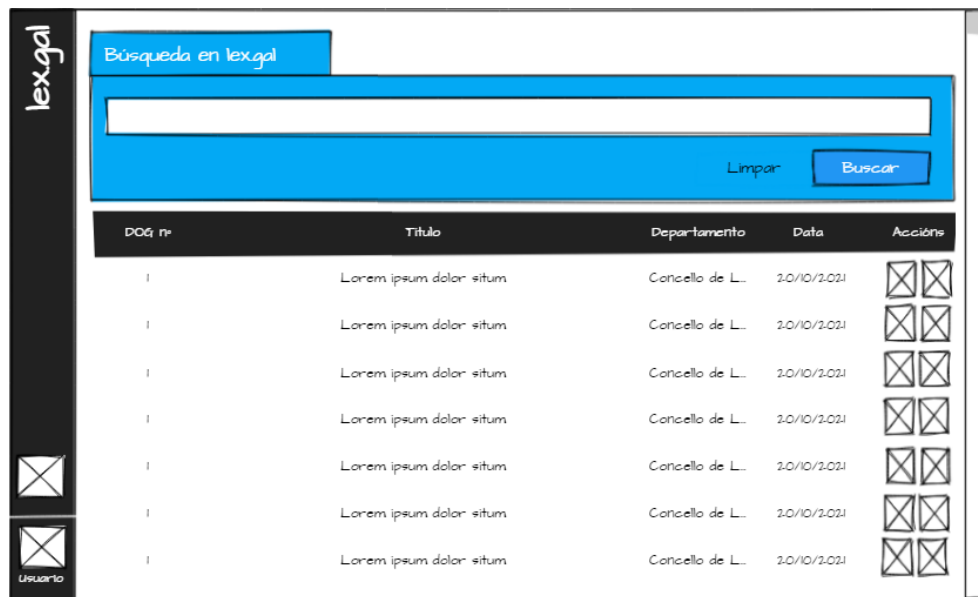


Figura 3.13: Diseño inicial de la página de búsqueda de leyes en lex.gal.

Se puede observar en la Figura 3.13 un primer diseño de la página de búsquedas. En esta destacan un buscador en la parte superior de la aplicación, y, debajo de este buscador, se muestran las leyes encontradas en una tabla.

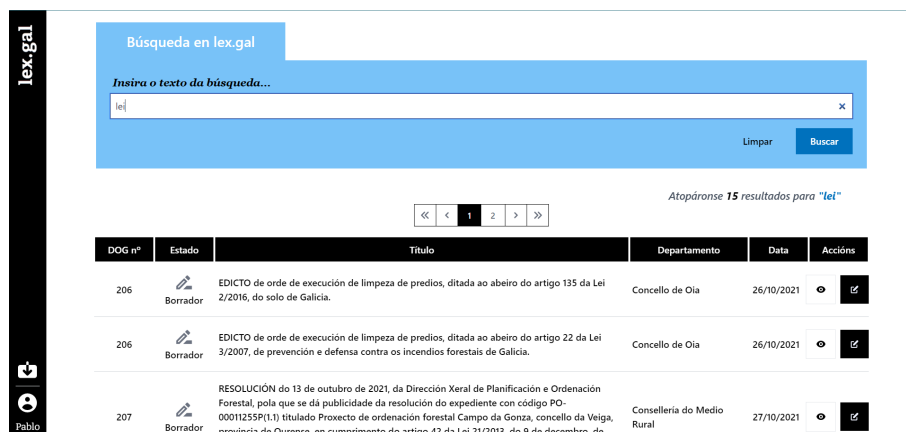


Figura 3.14: Diseño final de la página de búsqueda de leyes en lex.gal.

El resultado final de esta página varió ligeramente con respecto al diseño inicial, añadiendo un mayor número de elementos, como se puede observar en la figura Figura 3.14. En ella, se muestra, al igual que en el primer diseño, el buscador y la tabla de leyes. No obstante, se ha añadido que se muestre el número de resultados encontrados para el texto buscado, se permite navegar entre páginas y se ha añadido la columna de estado de la ley.

Figura 3.15: Diseño inicial de la página de edición de una ley de lex.gal.

Otra página de la que se realizó un primer diseño fue la página de edición. El *mockup* realizado se puede observar en la Figura 3.15. En este caso, se podían localizar los datos de cabecera en la parte superior de la página, el texto de la ley principal a la izquierda, donde se podía cambiar a la pestaña de leyes vinculadas, y, a la derecha, navegación entre notas y cambios.

Figura 3.16: Diseño final de la página de edición de una ley de lex.gal - I.



Figura 3.17: Diseño final de la página de edición de una ley de lex.gal - II.

En la Figura 3.16 y en la Figura 3.17 se puede ver el resultado final de la página de edición. Se observa a simple vista que esta cambió bastante con respecto a su diseño inicial.

En primer lugar, el número de datos de cabecera era muy amplio, con lo que se consideró hacer un desplegable según se quisieran ver o no. El siguiente cambio aparente fue mostrar un fragmento del sumario encima del texto, para poder saber qué ley se está editando. Sobre el texto de esta ley, se pueden añadir cambios (párrafos con fondo de color verde), y añadir anotaciones sobre párrafos (párrafos con color naranja de fondo).

En cuanto a la parte de la derecha, se han movido a esta parte las leyes vinculadas. Esto se debe a que en todo momento el experto en textos jurídicos desea ver el texto sobre el que trabaja, y en el caso del *mockup*, si quería ver las leyes vinculadas, este dejaba de ver el texto de la ley.

Por su parte, las notas y cambios ahora aparecen comprimidos para que el usuario trate de reconocerlos a simple vista, y en caso de querer más información, desplegarlos. También se permite acceder al párrafo donde se realizan, lo cual antes no se contemplaba. Esto se realiza guardando el párrafo sobre el que se realiza la nota/cambio, y la página realiza un scroll automático al párrafo correspondiente. Además, ahora sobre las notas también se permiten añadir comentarios.

Por último, se han añadido en la parte inferior de la página cuatro botones que se mantendrán siempre en dicha posición. Estos botones permitirán realizar diferentes operaciones sobre las leyes que se editan, ya sea validarlas, guardar como borrador, o eliminar.

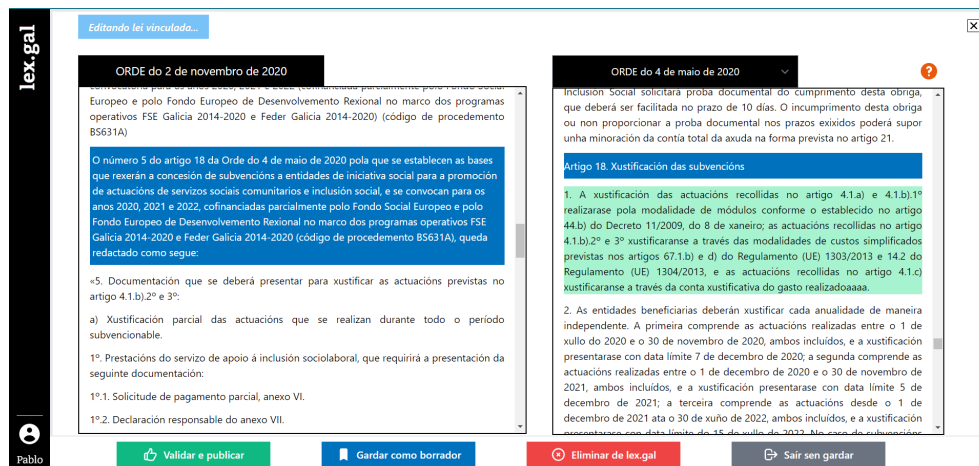


Figura 3.18: Diseño de la pestaña de edición de leyes vinculadas.

Por último, también cabe destacar el diseño de la pestaña de edición de una ley vinculada, si bien de esta no se hizo un diseño en su versión inicial al ser una funcionalidad que no se planteó al inicio del proyecto. Este diseño se puede ver en la Figura 3.18.

En ella, se puede observar la ley principal (izquierda), donde se localizan cambios que se realizan sobre la ley vinculada (derecha). Estos cambios son localizados mediante expresiones regulares que detectan frases sobre el texto de la ley principal (por ejemplo, “queda redactado como segue”). Si se pincha en el párrafo con fondo azul, se redirigirá al usuario a la parte de la ley vinculada donde se realiza el cambio, también detectado mediante el uso de expresiones regulares (por ejemplo, detección de “Artigo 18”). Si se realiza un cambio sobre un párrafo, este quedará marcado con un fondo de color verde.

Capítulo 4

Pruebas

El objetivo de las pruebas es encontrar posibles errores, defectos o problemas de conformidad en el producto. De esta forma, se pueden encontrar posibles errores de forma temprana, evitando cualquier tipo de error a la hora de entregar la aplicación.

Debido a que este TFG se basa en la realización de una aplicación web, se ha dividido el plan de pruebas en la realización de pruebas unitarias, pruebas de integración, y la validación de la interfaz del usuario.

En los siguientes apartados de la presente memoria se podrán observar los distintos resultados obtenidos a la hora de realizar los casos de prueba comentados.

4.1. Pruebas unitarias

Las pruebas unitarias [38] tienen como objetivo comprobar que las funcionalidades de la aplicación cumplen los objetivos esperados a la hora de introducir parámetros válidos e inválidos en la aplicación. Es por ello que estas pruebas han de ser siempre lo más completas posible, y deben abarcar un gran número de casos de entrada.

Para especificar cada caso de prueba unitaria se ha empleado la plantilla E.5. Plantilla de pruebas unitarias. Cabe mencionar también que en las dependencias se incluyen los requisitos funcionales especificados en el Apéndice C.1. Requisitos funcionales. Los distintos casos se muestran a continuación.

4.1.1. Casos de prueba

PU-01	Inicio de sesión con parámetros correctos
Versión	1.0 (08/06/2022)
Dependencias	FRQ-01.
Descripción	El usuario introduce su email y contraseña e inicia sesión.
Criterio de aceptación	La sesión se inicia correctamente.
Estado	Superada

Cuadro 4.1: Prueba unitaria 01. Inicio de sesión con parámetros correctos.

PU-02	Inicio de sesión con parámetros incorrectos
Versión	1.0 (08/06/2022)
Dependencias	FRQ-01.
Descripción	El usuario introduce un email y/o contraseña incorrectos e inicia sesión.
Criterio de aceptación	Se muestra un mensaje de error al usuario y no se inicia la sesión.
Estado	Superada

Cuadro 4.2: Prueba unitaria 02. Inicio de sesión con parámetros incorrectos.

PU-03	Cierre de sesión
Versión	1.0 (08/06/2022)
Dependencias	FRQ-02.
Descripción	El usuario pincha en el botón de cerrar sesión.
Criterio de aceptación	Se cierra la sesión del usuario en la aplicación.
Estado	Superada

Cuadro 4.3: Prueba unitaria 03. Cierre de sesión.

PU-04	Registro de usuario con parámetros válidos
Versión	1.0 (08/06/2022)
Dependencias	FRQ-03.
Descripción	El usuario administrador registra un usuario con todos los parámetros de registro válidos.
Criterio de aceptación	Se registra al usuario en la aplicación.
Estado	Superada

Cuadro 4.4: Prueba unitaria 04. Registro de usuario con parámetros válidos.

PU-05	Registro de usuario con parámetros inválidos
Versión	1.0 (08/06/2022)
Dependencias	FRQ-03.
Descripción	El usuario administrador intenta registrar un usuario con uno de sus parámetros con condiciones no válidas para el registro.
Criterio de aceptación	Se muestra un mensaje de error al usuario administrador indicando cuál es el campo inválido.
Estado	Superada

Cuadro 4.5: Prueba unitaria 05. Registro de usuario con parámetros inválidos.

PU-06	Búsqueda de leyes en el DOG con filtros válidos
Versión	1.0 (08/06/2022)
Dependencias	FRQ-04.
Descripción	El usuario busca cualquier ley en el DOG con filtros válidos.
Criterio de aceptación	Se muestran al usuario las leyes encontradas en el DOG.
Estado	Superada

Cuadro 4.6: Prueba unitaria 06. Búsqueda de leyes en el DOG con filtros válidos.

PU-07	Búsqueda de leyes en el DOG sin introducir texto
Versión	1.0 (08/06/2022)
Dependencias	FRQ-04.
Descripción	El usuario busca cualquier ley en el DOG sin introducir nada en el campo de texto de búsqueda.
Criterio de aceptación	Se muestra al usuario un mensaje indicando que es necesario introducir un texto para buscar.
Estado	Superada

Cuadro 4.7: Prueba unitaria 07. Búsqueda de leyes en el DOG sin introducir texto.

PU-08	Previsualizar una ley en el DOG
Versión	1.0 (08/06/2022)
Dependencias	FRQ-05.
Descripción	El usuario intenta previsualizar una ley en el DOG pinchando en el icono del ojo de las acciones de una ley del DOG.
Criterio de aceptación	Se muestra al usuario la ley en la página del DOG.
Estado	Superada

Cuadro 4.8: Prueba unitaria 08. Previsualizar una ley en el DOG.

PU-09	Importar ley del DOG en lex.gal
Versión	1.0 (08/06/2022)
Dependencias	FRQ-06.
Descripción	El usuario intenta importar una ley del DOG en lex.gal haciendo click en el botón de importar.
Criterio de aceptación	La ley se almacena en lex.gal y se muestra al usuario un mensaje de que se ha almacenado correctamente.
Estado	Superada

Cuadro 4.9: Prueba unitaria 09. Importar ley del DOG en lex.gal.

PU-10	El usuario busca una ley en lex.gal
Versión	1.0 (08/06/2022)
Dependencias	FRQ-07.
Descripción	El usuario busca cualquier ley en lex.gal, ya sea con texto de búsqueda o sin él.
Criterio de aceptación	Se muestran al usuario las leyes encontradas.
Estado	Superada

Cuadro 4.10: Prueba unitaria 10. El usuario busca una ley en lex.gal.

PU-11	Previsualizar una ley en lex.gal
Versión	1.0 (08/06/2022)
Dependencias	FRQ-08.
Descripción	El usuario intenta previsualizar una ley en lex.gal pinchando en el icono del ojo de las acciones de una ley de lex.gal.
Criterio de aceptación	Se muestra al usuario la ley correspondiente en una pestaña nueva.
Estado	Superada

Cuadro 4.11: Prueba unitaria 11. Previsualizar una ley en lex.gal.

PU-12	Cargar los datos de una ley de lex.gal
Versión	1.0 (08/06/2022)
Dependencias	FRQ-09.
Descripción	Al pinchar el botón de editar, se cargan los datos de la ley a editar en la página de edición.
Criterio de aceptación	Se muestra al usuario la ley correspondiente con sus datos en la página de edición.
Estado	Superada

Cuadro 4.12: Prueba unitaria 12. Cargar los datos de una ley de lex.gal.

PU-13	Proponer un cambio en la ley principal de lex.gal
Versión	1.0 (08/06/2022)
Dependencias	FRQ-10.
Descripción	El usuario propone un cambio sobre cualquier párrafo de la ley principal.
Criterio de aceptación	El cambio propuesto se añade a la lista de cambios, y se cambia el color de fondo del párrafo donde se introduce el cambio.
Estado	Superada

Cuadro 4.13: Prueba unitaria 13. Proponer un cambio en la ley principal de lex.gal.

PU-14	Descartar una selección de cambios de lex.gal
Versión	1.0 (08/06/2022)
Dependencias	FRQ-10.
Descripción	El usuario descarta una selección de cambios en la lista de cambios propuestos.
Criterio de aceptación	Se descarta la selección de cambios en la lista de cambios propuestos.
Estado	Superada

Cuadro 4.14: Prueba unitaria 14. Descartar una selección de cambios de lex.gal.

PU-15	Descartar todos los cambios propuestos a una ley de lex.gal
Versión	1.0 (08/06/2022)
Dependencias	FRQ-10.
Descripción	El usuario descarta todos los cambios en la lista de cambios propuestos.
Criterio de aceptación	Se descartan todos los cambios en la lista de cambios propuestos.
Estado	Superada

Cuadro 4.15: Prueba unitaria 15. Descartar todos los cambios propuestos a una ley de lex.gal.

PU-16	Añadir una anotación sobre un párrafo de la ley principal de lex.gal
Versión	1.0 (08/06/2022)
Dependencias	FRQ-11.
Descripción	El usuario añade una anotación sobre cualquier párrafo de la ley principal.
Criterio de aceptación	La anotación se añade a la lista de notas, y se cambia el color de fondo del párrafo donde se introduce el cambio.
Estado	Superada

Cuadro 4.16: Prueba unitaria 16. Añadir una anotación sobre un párrafo de la ley principal de lex.gal.

PU-17	Descartar una selección de anotaciones de una ley de lex.gal
Versión	1.0 (08/06/2022)
Dependencias	FRQ-11.
Descripción	El usuario descarta una selección de anotaciones en la lista de notas.
Criterio de aceptación	Se descarta la selección de anotaciones en la lista de notas.
Estado	Superada

Cuadro 4.17: Prueba unitaria 17. Descartar una selección de anotaciones de una ley de lex.gal.

PU-18	Descartar todas las anotaciones de una ley de lex.gal
Versión	1.0 (08/06/2022)
Dependencias	FRQ-11.
Descripción	El usuario descarta todas las anotaciones en la lista de notas.
Criterio de aceptación	Se descartan todas las anotaciones en la lista de notas.
Estado	Superada

Cuadro 4.18: Prueba unitaria 18. Descartar todas las anotaciones de una ley de lex.gal.

PU-19	Añadir un comentario sobre cualquier anotación
Versión	1.0 (08/06/2022)
Dependencias	FRQ-11.
Descripción	El usuario añade un comentario sobre una anotación.
Criterio de aceptación	El comentario es añadido a la anotación.
Estado	Superada

Cuadro 4.19: Prueba unitaria 19. Añadir un comentario sobre cualquier anotación.

PU-20	Cargar las leyes vinculadas a una ley de lex.gal
Versión	1.0 (08/06/2022)
Dependencias	FRQ-12.
Descripción	El usuario pincha en el apartado “Leis vinc.” en la pantalla de edición.
Criterio de aceptación	Se muestra la lista de leyes vinculadas a la ley.
Estado	Superada

Cuadro 4.20: Prueba unitaria 20. Cargar las leyes vinculadas a una ley de lex.gal.

PU-21	Añadir una ley vinculada a la lista de leyes vinculadas a una ley
Versión	1.0 (08/06/2022)
Dependencias	FRQ-13.
Descripción	El usuario añade una ley vinculada a la lista de leyes vinculadas de una ley principal.
Criterio de aceptación	Se añade la ley vinculada a la lista de leyes vinculadas.
Estado	Superada

Cuadro 4.21: Prueba unitaria 21. Añadir una ley vinculada a la lista de leyes vinculadas a una ley.

PU-22	Ir al párrafo donde se modifica una sección de la ley vinculada
Versión	1.0 (08/06/2022)
Dependencias	FRQ-14.
Descripción	El usuario pincha en la ley principal en un párrafo con fondo de color azul.
Criterio de aceptación	Se redirige al usuario a la parte de la ley vinculada donde se ha de realizar el cambio.
Estado	Superada

Cuadro 4.22: Prueba unitaria 22. Ir al párrafo donde se modifica una sección de la ley vinculada.

PU-23	Proponer un cambio sobre la ley vinculada
Versión	1.0 (08/06/2022)
Dependencias	FRQ-14.
Descripción	El usuario propone un cambio a una ley vinculada.
Criterio de aceptación	Se guarda el cambio en la ley vinculada y se muestra con un fondo de color verde.
Estado	Superada

Cuadro 4.23: Prueba unitaria 23. Proponer un cambio sobre la ley vinculada.

PU-24	Descartar un cambio sobre la ley vinculada
Versión	1.0 (08/06/2022)
Dependencias	FRQ-14.
Descripción	El usuario descarta un cambio sobre una ley vinculada.
Criterio de aceptación	Se descarta el cambio en la ley vinculada y se quita el color de fondo.
Estado	Superada

Cuadro 4.24: Prueba unitaria 24. Descartar un cambio sobre la ley vinculada.

PU-25	Eliminar una ley de lex.gal
Versión	1.0 (08/06/2022)
Dependencias	FRQ-15.
Descripción	El usuario pincha en “Eliminar de lex.gal”.
Criterio de aceptación	Se elimina la ley de lex.gal.
Estado	Superada

Cuadro 4.25: Prueba unitaria 25. Eliminar una ley de lex.gal.

PU-26	Validar y publicar una ley de lex.gal
Versión	1.0 (08/06/2022)
Dependencias	FRQ-10, FRQ-11, FRQ-12, FRQ-13, FRQ-14.
Descripción	El usuario pincha en “Validar e publicar”.
Criterio de aceptación	Se almacena la ley con todos los cambios, anotaciones, cambios sobre leyes vinculadas, leyes vinculadas a la ley y datos de cabecera editados en lex.gal. El estado de la ley, y el estado de sus leyes vinculadas, pasa a ser de Validada.
Estado	Superada

Cuadro 4.26: Prueba unitaria 26. Validar y publicar una ley de lex.gal.

PU-27	Guardar como borrador una ley de lex.gal
Versión	1.0 (08/06/2022)
Dependencias	FRQ-10, FRQ-11, FRQ-12, FRQ-13, FRQ-14.
Descripción	El usuario pincha en “Gardar como borrador”.
Criterio de aceptación	Se almacena la ley con todos los cambios, anotaciones, cambios sobre leyes vinculadas, leyes vinculadas a la ley y datos de cabecera editados en lex.gal. El estado de la ley, y el estado de sus leyes vinculadas, pasa a ser de Borrador.
Estado	Superada

Cuadro 4.27: Prueba unitaria 27. Guardar como borrador una ley de lex.gal.

4.1.2. Matriz de trazabilidad

Para hacer un resumen de los resultados de las pruebas unitarias, se ha realizado una matriz de trazabilidad donde se han relacionado los casos de prueba

con los requisitos funcionales. De esta forma, comprobamos que se han cumplido todos los requisitos funcionales en las pruebas.

	FRQ-01	FRQ-02	FRQ-03	FRQ-04	FRQ-05	FRQ-06	FRQ-07	FRQ-08	FRQ-09	FRQ-10	FRQ-11	FRQ-12	FRQ-13	FRQ-14	FRQ-15
PU-01	X														
PU-02	X														
PU-03		X													
PU-04			X												
PU-05			X												
PU-06				X											
PU-07				X											
PU-08					X										
PU-09						X									
PU-10							X								
PU-11								X							
PU-12									X						
PU-13										X					
PU-14										X					
PU-15										X					
PU-16											X				
PU-17											X				
PU-18											X				
PU-19											X				
PU-20												X			
PU-21													X		
PU-22														X	
PU-23														X	
PU-24														X	
PU-25															X
PU-26										X	X	X	X	X	
PU-27										X	X	X	X	X	

Cuadro 4.28: Matriz de trazabilidad de Pruebas unitarias-Requisitos funcionales.

4.2. Pruebas de integración

Una vez superadas las pruebas unitarias, se puede proceder a realizar las pruebas de integración [39]. Estas pruebas tratan de comprobar que la interac-

ción entre los diferentes módulos de la herramienta funcionan correctamente y sin fallos.

Para especificar cada caso de prueba de integración se ha empleado la plantilla E.6. Plantilla de pruebas de integración, y estos casos se muestran a continuación.

PI-01	Inicio de sesión
Versión	1.0 (08/06/2022)
Descripción	Cuando un usuario inicia sesión en la aplicación, este debe ser redirigido a la página principal.
Criterio de aceptación	Al iniciar sesión se recibe una respuesta HTTP con un código de estado 200, indicando que el inicio de sesión ha sido correcto.
Estado	Superada

Cuadro 4.29: Prueba de integración 01. Inicio de sesión.

PI-02	Registro de usuario
Versión	1.0 (08/06/2022)
Descripción	Cuando un usuario administrador trate de registrar a otro usuario, debe recibir un mensaje indicativo de que la operación se realizó correctamente.
Criterio de aceptación	Si se recibe un código HTTP 201, significará que se ha registrado, y, en ese caso, se muestra un mensaje en la pantalla indicando que esto ha sido así.
Estado	Superada

Cuadro 4.30: Prueba de integración 02. Registro de usuario.

PI-03	Búsqueda de una ley en el DOG
Versión	1.0 (08/06/2022)
Descripción	Cuando un usuario busca leyes en el DOG, estas se le muestran en la pantalla según los filtros de búsqueda.
Criterio de aceptación	Si se recibe un código HTTP 200, significará que se han obtenido los resultados de búsqueda correctamente, y, en ese caso, se muestran en la pestaña.
Estado	Superada

Cuadro 4.31: Prueba de integración 03. Búsqueda de una ley en el DOG.

PI-04	Importar una ley del DOG a lex.gal
Versión	1.0 (08/06/2022)
Descripción	Cuando un usuario pincha en el botón de importar, se debe almacenar la ley del DOG en la base de datos de lex.gal.
Criterio de aceptación	Si se recibe un código HTTP 201, significará que se ha almacenado la ley en la base de datos, y, en ese caso, se muestra un mensaje al usuario indicando que la operación se realizó correctamente.
Estado	Superada

Cuadro 4.32: Prueba de integración 04. Importar una ley del DOG a lex.gal.

PI-05	Búsqueda de una ley en lex.gal
Versión	1.0 (08/06/2022)
Descripción	Cuando un usuario busca leyes en lex.gal, estas se le muestran en la pantalla según los filtros de búsqueda.
Criterio de aceptación	Si se recibe un código HTTP 200 o un código de HTTP 404, significará que se han obtenido los resultados de búsqueda correctamente, o bien que no se han encontrado resultados. En dicha situación, se muestran las leyes encontradas, o un mensaje de aviso de que no se encontraron leyes.
Estado	Superada

Cuadro 4.33: Prueba de integración 05. Búsqueda de una ley en lex.gal.

PI-06	Previsualizar una ley de lex.gal
Versión	1.0 (08/06/2022)
Descripción	Cuando un usuario trate de previsualizar una ley en lex.gal, esta se le ha de mostrar en una nueva pantalla con todo el contenido de la ley.
Criterio de aceptación	Si se recibe un código HTTP 200, significará que se han obtenido los datos de la ley, y, en ese caso, se muestra en la nueva pestaña una previsualización de la correspondiente ley.
Estado	Superada

Cuadro 4.34: Prueba de integración 06. Previsualizar una ley de lex.gal.

PI-07	Comenzar a editar una ley de lex.gal
Versión	1.0 (08/06/2022)
Descripción	Cuando un usuario trate de editar una ley, este debe recibir todos sus datos en la pantalla de edición, ya sean los cambios, datos de cabecera, contenido, anotaciones, etc.
Criterio de aceptación	Si se recibe un código HTTP 200, significará que se han obtenido los datos de la ley, y, en ese caso, se muestran en la pantalla todos los datos de la correspondiente ley.
Estado	Superada

Cuadro 4.35: Prueba de integración 07. Comenzar a editar una ley de lex.gal.

PI-08	Comenzar a editar una ley vinculada a otra ley de lex.gal
Versión	1.0 (08/06/2022)
Descripción	Cuando un usuario trate de editar una ley vinculada, este debe recibir todo el contenido de la ley vinculada que desea modificar.
Criterio de aceptación	Si se recibe un código HTTP 200, al abrir la pestaña de modificar ley vinculada se recibe el contenido de la ley vinculada, con los cambios realizados sobre esta.
Estado	Superada

Cuadro 4.36: Prueba de integración 08. Comenzar a editar una ley vinculada a otra ley de lex.gal.

PI-09	Validar y publicar una ley de lex.gal
Versión	1.0 (08/06/2022)
Descripción	Cuando un usuario trate de validar y publicar una ley, deberá recibir un mensaje indicándole que todo ha ido correctamente, y ser redirigido a la página principal.
Criterio de aceptación	Si se recibe un código HTTP 200, esto indicará que tanto la ley principal como la vinculada han sido validadas y publicadas. Se le muestra un mensaje indicando que las leyes fueron validadas, y el usuario es redirigido a la página principal.
Estado	Superada

Cuadro 4.37: Prueba de integración 09. Validar y publicar una ley de lex.gal.

PI-10	Guardar como borrador una ley de lex.gal
Versión	1.0 (08/06/2022)
Descripción	Cuando un usuario trate de guardar como borrador una ley, deberá recibir un mensaje indicándole que todo ha ido correctamente, y ser redirigido a la página principal.
Criterio de aceptación	Si se recibe un código HTTP 200, esto indicará que tanto la ley principal como la vinculada han sido guardadas como borrador. Se le muestra un mensaje que las leyes fueron guardadas como borrador, y el usuario es redirigido a la página principal.
Estado	Superada

Cuadro 4.38: Prueba de integración 10. Guardar como borrador una ley de lex.gal.

PI-11	Eliminar una ley de lex.gal
Versión	1.0 (08/06/2022)
Descripción	Se eliminará una ley de la página de lex.gal al pinchar en el botón “Eliminar de lex.gal”, y el usuario será redirigido a la página principal.
Criterio de aceptación	Si se recibe un código de HTTP 204, esto indicará que la ley se ha borrado correctamente de la base de datos de lex.gal. Se redirige al usuario a la página principal, mostrando un mensaje de que la ley fue borrada correctamente.
Estado	Superada

Cuadro 4.39: Prueba de integración 11. Eliminar una ley de lex.gal.

4.3. Validación de la interfaz de usuario

A la hora de realizar una interfaz con el usuario, se ha de tener suma precaución, ya que es un aspecto clave en toda aplicación. Una aplicación siempre ha de resultar sencilla de utilizar por el usuario, mantener la usabilidad de esta, mantener un código de colores estable, etc.

Para evaluar que la interfaz de la aplicación cumple estos requisitos, se han barajado distintas opciones. Por una parte, se ha realizado un cuestionario SUS [40], donde se ha preguntado a expertos en textos jurídicos acerca de la usabilidad de la aplicación. Por otro lado, también se han utilizado los principios heurísticos de Nielsen [41] para evaluar la usabilidad de forma autónoma.

4.3.1. Cuestionario SUS

Un cuestionario SUS (System Usability Scale) se trata de un método para evaluar la usabilidad de cualquier sistema. Gracias a este método se permite medir la eficacia, eficiencia y satisfacción de un usuario al usar la aplicación.

Este cuestionario cuenta con un total de 10 preguntas y una escala del 1-5, siendo 1: *Totalmente en desacuerdo*, y 5: *Totalmente de acuerdo*. Las cuestiones han de ser respondidas obligatoriamente, y son las siguientes:

1. Me gustaría usar este sistema frecuentemente.
2. Encontré el sistema innecesariamente complejo.
3. Pensé que el sistema era fácil de usar.
4. Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema.
5. Encontré que las diversas funciones de este sistema estaban bien integradas.
6. Pensé que había demasiada inconsistencia en este sistema.
7. Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente.
8. El sistema no informa en ningún momento de errores.
9. El sistema informa de forma clara y concisa del estado de las operaciones que se realizan en él.
10. Necesitaba aprender muchas cosas antes de empezar con este sistema.

Si se presta atención a las cuestiones, se puede ver que las impares tienen en cuenta aspectos positivos de la aplicación, mientras que las cuestiones pares tratan debilidades que esta puede poseer. Teniendo en cuenta esto, se puede proceder a calcular los resultados, donde se han tenido en cuenta las dos reglas siguientes:

- Se suman las respuestas de los enunciados impares y después se resta 5.
- Se suman las respuestas de los enunciados pares y después se resta 25.
- Se suman los dos resultados y se multiplica por 2,5.

Teniendo en cuenta las reglas indicadas anteriormente, se han obtenido los siguientes resultados:

	Pregunta 1	Pregunta 2	Pregunta 3	Pregunta 4	Pregunta 5	Pregunta 6	Pregunta 7	Pregunta 8	Pregunta 9	Pregunta 10	Valoración (sobre 100)
Experto 1	4	2	4	2	5	1	4	1	4	3	80
Experto 2	4	2	5	3	4	2	5	2	4	3	75
Experto 3	5	2	4	1	4	2	4	2	4	2	80
Experto 4	4	2	5	1	4	1	4	1	4	2	85

Cuadro 4.40: Resultados del cuestionario SUS.

La escala de usabilidad va de 0-100, siendo 68 un estado aceptable de usabilidad. Si realizamos una media de los resultados obtenidos, la puntuación final será de **80/100**, cumpliendo con un nivel aceptable de usabilidad para el usuario.

4.3.2. Principios heurísticos de Nielsen

También se han utilizado como método de evaluación de la usabilidad de la aplicación los **principios heurísticos de Nielsen**. Son un total de diez principios que se utilizan para evaluar la usabilidad de una interfaz web, y, en este caso, se han utilizado para evaluar en qué grado es usable la aplicación.

1. **Visibilidad del estado del sistema:** El sistema debe mantener siempre informado al usuario acerca de lo que está ocurriendo.

Se considera que la aplicación cumple con este principio, puesto que en toda pantalla se muestra al usuario la operación que está realizando o va a realizar. Además, se muestran mensajes cuando un usuario realiza una operación como puede ser la edición de un documento, indicando que esta se realizó correctamente.

2. **Relación entre el sistema y el mundo real:** El sitio web utiliza el lenguaje del usuario con expresiones y palabras que le resulten familiares. Además la información debe aparecer en un orden lógico y natural.

La iconografía se considera adecuada en aquellos botones donde se ha utilizado, el código de colores se adecúa correctamente con las operaciones (verde para aceptar, rojo para eliminar, naranja/amarillo para pedir ayuda, ...). Además, el lenguaje empleado es el propio de los textos jurídicos. Estos aspectos permiten que se cumpla el segundo principio heurístico en la aplicación.

3. **Control y libertad del usuario:** En caso de seleccionar una función por error, el usuario tiene una salida fácil a la página anterior u otra página.

En toda página que el usuario entre sin querer se muestra información para volver, o bien para cerrar la pestaña y volver al estado anterior con iconografía. Además, en este aspecto los códigos de colores empleados se consideran ilustrativos para volver atrás en todo momento.

4. **Consistencia y estándares:** Se establecen convenciones lógicas, pues el usuario no tiene por qué saber que diferentes palabras, situaciones o acciones significan lo mismo.

En la aplicación todo título de las ventanas resume su funcionalidad, las abreviaturas empleadas son consistentes en el sistema, al igual que ocurre con la disposición de los elementos en la página y los colores empleados para las acciones son siempre los mismos.

5. **Prevención de errores:** Ayudar al usuario a que no caiga en un error.

Se previene en gran medida que el usuario no caiga en un error, ya sea indicando que busque con texto adecuado a la hora de búsqueda de leyes, mostrando mensajes de apoyo, o posibles ejemplos de texto a introducir en los campos de texto.

6. **Reconocimiento antes que recuerdo:** Se deben hacer visibles las acciones y opciones antes de que el usuario deba recordar información entre distintas secciones o partes de la aplicación.

La paleta de colores empleada se mantiene en el rango de 7 ± 2 colores, al igual que existen 7 ± 2 elementos en toda pantalla. La información de la ley principal se mantiene en todo momento a la hora de editar una ley vinculada a esta. El texto introducido en el campo de búsqueda de leyes se mantiene en toda búsqueda, al igual que se indica al usuario en todo momento la página en la que se encuentra.

7. **Flexibilidad y eficiencia de uso:** El sistema incluye atajos para que el usuario experto pueda realizar las operaciones de forma más rápida, sin que esto cause un problema a un usuario con menor experiencia.

La aplicación contiene algún atajo como el de buscar leyes al presionar la tecla ENTER, campos de entrada con valores explicativos o comandos como CTRL+C y CTRL+V a la hora de copiar y pegar el contenido en la edición de una ley.

8. **Estética y diseño minimalista:** Las páginas no contienen información innecesaria, siendo toda esta relevante en todo momento.

La cantidad de elementos en pantalla no resulta excesiva a la vista del usuario, y toda información presente, sobre todo a la hora de buscar una

ley, resulta ilustrativa para el usuario a la hora de saber si es la ley que deseará editar o importar.

9. **Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de errores:** Los mensajes de error ayudan a los usuarios a saber qué está ocurriendo en el sistema y cómo recuperarse.

Si bien se pueden producir pocos errores en la aplicación, estos son mostrados al usuario. Ejemplos para este principio son el caso de que se trate de buscar una ley en el DOG sin introducir texto, o de que no se encuentren leyes a la hora de la búsqueda. En todo caso se muestra el mensaje de error correspondiente al usuario para que entienda que está ocurriendo.

10. **Ayuda y documentación:** Siempre que sea posible se ha de poder utilizar la aplicación sin la necesidad de documentación adicional. No obstante, nunca está de más añadir posible ayuda y/o documentación en caso de ser esta necesaria.

En este caso la aplicación contiene una serie de *tooltips*, que se puede traducir como “información sobre herramientas” [42], los cuales indican al usuario como realizar diferentes acciones en la página. Además, si esta información no fuese suficiente, siempre estará disponible el Apéndice B. Manuales de usuario para guiar al usuario en la aplicación.

4.4. Validación de los resultados de las pruebas

Tras realizar todas las pruebas se ha llegado a la conclusión de que la aplicación cumple correctamente con los requisitos propuestos en la Especificación de requisitos.

Todas las pruebas unitarias han cumplido con los requisitos establecidos en ellas para todos los casos establecidos y sus diferentes valores de entrada. En cuanto a las pruebas de integración, todos los módulos interaccionan entre ellos sin mostrar fallos. Esto permite establecer el estado de todas las pruebas como superado.

En cuanto al cuestionario SUS, los usuarios han quedado satisfechos con la usabilidad de la aplicación, pues se ha obtenido una valoración alta. Además, con respecto a los principios de Nielsen, estos se cumplen en gran medida. Todo esto permite concluir que la aplicación es realmente usable para las necesidades de los profesionales en textos jurídicos.

De esta forma, se ha validado que la aplicación cumple con todo requisito y objetivo establecido para la aplicación con un buen grado de conformidad.

Capítulo 5

Conclusiones y posibles ampliaciones

Para finalizar, en este último capítulo se detallan las conclusiones obtenidas de este proyecto. Además, se han extraído una serie de posibles ampliaciones de cara a una mejora de la aplicación de lex.gal en un futuro.

5.1. Conclusiones

La redacción de textos jurídicos requiere un trabajo muy laborioso, donde en todo momento se ha de ser muy preciso con las palabras empleadas. Cualquier afirmación hecha en una ley no puede dejar lugar a dudas, pues podrían quedar ambigüedades de las que cualquier persona podría aprovecharse de la redacción de una ley para determinadas acciones.

Gracias a este TFG se han adquirido una gran cantidad de conocimientos acerca de la redacción de textos jurídicos, además de como deben ser redactados, modificados, y como han de ser filtrados en las búsquedas.

En cuanto a las tecnologías empleadas, se ha adquirido un alto grado de conocimiento en tecnologías relacionadas con el desarrollo web. Si bien se habían empleado anteriormente algunos lenguajes como Java, HTML o JavaScript, en ningún momento se había procedido a la realización de una aplicación de forma completa, tanto la parte del servidor, como el cliente, y la propia base de datos.

Se han adquirido conocimientos en React.js, uno de los frameworks de JavaScript más utilizados hoy en día por su gran rendimiento y calidad a la hora de realizar *frontend*. En cuanto al estilizado de las páginas, se ha aprendido a utilizar Tailwind, un framework de CSS muy popular en los últimos años.

En cuanto a la parte del servidor y la base de datos, se han ampliado múltiples conocimientos en Spring Boot y MongoDB. También se ha comprendido como documentar correctamente la API de esta aplicación mediante el uso de Swagger.

Para finalizar, cabe destacar que si bien la aplicación está diseñada para utilizar por parte de los usuarios de lex.gal, realizando una serie de pequeñas modificaciones podría llegarse a destinar el proyecto a cualquier aplicación donde fuese necesaria la edición de documentos de forma colaborativa entre usuarios.

5.2. Posibles ampliaciones

Si bien el proyecto ha quedado completo para una versión inicial de la aplicación, esta puede extenderse con nuevas funcionalidades, o funcionalidades existentes pero con versiones mejoradas de las mismas.

Un ejemplo de esto es la búsqueda de modificaciones sobre una ley vinculada. Estas son localizadas mediante expresiones regulares. Si bien se han encontrado expresiones regulares para localizar dichas modificaciones, debería realizarse una búsqueda exhaustiva de todas las expresiones posibles para localizar modificaciones de una ley sobre otra.

Otra posible ampliación relacionada con este aspecto, sería la automatización del proceso de aplicar el cambio sobre la ley vinculada. Actualmente la solución propuesta en este proyecto es que se localice el cambio, y se pueda ir a este con un simple click, donde el usuario procederá a aplicar el cambio. No obstante, en una futura versión podría aplicarse que el cambio se genere automáticamente a partir del contenido de la ley principal, sin que el usuario tenga que realizar ninguna interacción.

Adicionalmente, se podría ampliar el concepto de añadir anotaciones a la pestaña de leyes vinculadas. De esta forma, desde la ley principal donde se modifica una vinculada, podrían añadirse anotaciones también sobre esta última.

Con respecto a la página de búsquedas, actualmente se pueden filtrar las leyes únicamente por el texto contenido en el sumario, así como navegar entre todas las páginas encontradas. Esto sería ampliable mediante nuevos filtros de búsqueda, pero una gran ampliación podría ser buscar también palabras en todo el contenido de cada una de las leyes, y no tan solo en el sumario.

Como ampliación de cara a la previsualización de leyes, podría ser una buena idea implementar un control de versiones sobre la ley. De esta forma, cada vez que un cambio fuese validado y publicado, se podría acceder a una nueva versión de

la ley. Asimismo, se podría acceder a una versión anterior de la ley, si deseásemos ver en qué momento fue aplicado un cambio.

En cuanto al tratamiento de documentos, actualmente solo se permite trabajar con documentos en lenguajes de etiquetas. En un futuro podría implementarse una versión donde trabajar con otro tipo de extensiones, como puede ser el caso de un archivo en formato JSON.

Como resulta obvio en cualquier tipo de aplicación, se deberían realizar estudios de usabilidad a medida que se van realizando nuevas versiones de una aplicación. Es por ello que si bien en esta aplicación se realizan muchas mejoras con respecto a la versión actual de edición de documentos de *lex.gal*, se puede decir con certeza que hay ciertos aspectos donde la usabilidad se puede mejorar, ya que una aplicación nunca llega a ser perfecta para todos los públicos, pero sí serlo para la gran mayoría de ellos.

Finalizando ya con posibles ampliaciones, podría tratar de implementarse también que la edición colaborativa fuese en tiempo real. Si bien la complejidad de la aplicación escalaría enormemente, siendo necesaria la implementación de *web-sockets* [43], daría un gran margen de mejora a la aplicación al poder ser empleada por varios usuarios simultáneamente.

Por último, teniendo ya acceso a la base de datos de la aplicación de *lex.gal* publicada por la Xunta, podría sincronizarse el acceso a documentos oficiales y a usuarios existentes. De este modo, se obtendría control sobre todos los documentos existentes, y el control de cambios entre usuarios sería completamente real.

Apéndice A

Manuales técnicos

En este apéndice se puede consultar como desplegar la aplicación, tanto el cliente, como el servidor, y la propia base de datos. Cabe recordar que se debe tener acceso a Internet en todo momento para un correcto funcionamiento de la herramienta, puesto que la página de búsquedas localizará los documentos contra la API del DOG. Otro detalle a tener en cuenta es que la aplicación se ha probado mayoritariamente en un entorno con un sistema Windows 10.

El enlace al repositorio del código de la aplicación es el siguiente https://nubeusc-my.sharepoint.com/:f:/g/personal/pablo_tarrio_otero_rai_usc_es/Eikm9g7gY45AsFDhafD5fssBJAQF4qyJSGU4deA6H96VZw?e=xxKGvs.

A.1. Manual de despliegue

Para poder ejecutar la aplicación en cualquier ordenador, se ha de contar con las siguientes herramientas instaladas:

- **Node.js**: entorno donde se desplegará el cliente. Permite la instalación de **npm**, que se encarga de gestionar los paquetes de Node.js.
- **IntelliJ Idea** [44] o cualquier **Java SE Development Kit (JDK)** [45] **versión 15.0 o superior**: necesario para poder ejecutar los archivos que contienen el servidor de la aplicación.
- **MongoDB**: sistema de base de datos NoSQL necesario para poder emplear la base de datos de la aplicación.

Una vez hemos instalado estos tres programas, procedemos al despliegue. Para poder proceder al despliegue de la **capa de servicios** (elegir una de las dos opciones):

- En caso de contar con un JDK instalado en nuestro ordenador:

1. Acceder al directorio *Servidor* en la carpeta del proyecto.
2. Ejecutar el archivo *TFG-Servidor.jar*.
 - En caso de contar con IntelliJ Idea en nuestro ordenador:
1. Ejecutar la aplicación de IntelliJ Idea en nuestra máquina.
2. Pinchar en *Open Project* y seleccionamos la carpeta *Servidor* que se encuentra dentro del directorio proyecto.
3. Ejecutar el archivo *Application.java*.
4. Una vez se ha ejecutado, esperamos unos segundos para que termine de compilar. El servicio será desplegado en el puerto 8000 (acceso mediante *http://localhost:8000/*).

En cuanto al despliegue de la parte del **cliente**:

1. Abrir cualquier terminal disponible en el ordenador (por ejemplo, *Símbolos del Sistema de Windows (CMD)* [46]).
2. Acceder a la carpeta *Cliente* dentro del directorio del proyecto.
3. Ejecutar el comando *npm install*.
4. Ejecutar el comando *npm start*.
5. Se abrirá una pestaña en el navegador predeterminado. Este proceso puede demorarse unos minutos. En caso de no abrirse automáticamente, se puede acceder manualmente accediendo al puerto 3000, *http://localhost:3000/*.

Por último, cabe indicar que por motivos de seguridad se ha de introducir un usuario a mano en la base de datos. Esto se debe a que en la aplicación solo se permite registrar usuarios cuando existe un administrador. Es por ello que se debe acceder a MongoDB, crear una nueva base de datos denominada “tfg”, y la colección ha de ser “users”. Importaremos el archivo *data.json* (situado en el directorio raíz del proyecto), que contiene el usuario inicial.

Cuando sea el caso de iniciar sesión por primera vez, habrá que acceder con el email *usuario@usuario.com* y la contraseña *Abc123...*

A.2. Versiones de las herramientas recomendadas

Para un correcto funcionamiento de la aplicación, se recomienda utilizar las siguientes herramientas:

- **Node.js** (16.15.0 (64 bits)).
- **IntelliJ Idea** (2022.1.1 (64 bits)).
- **Java SE Development Kit (JDK)** (18.0.1.1 (64 bits)).
- **MongoDB Community** (5.0.8 (64 bits)).
- **Navegador Google Chrome** (Versión 101.0.4951.67 (Build oficial) (64 bits)).
- **Sistema Operativo Windows 10** (Versión 21H2 (64 bits)).
- **Resolución de pantalla** de 15.6" (1920x1080).

Apéndice B

Manuales de usuario

A continuación, se presenta el manual de usuario, el cual permitirá entender el funcionamiento de la aplicación desarrollada. De esta forma, cualquier persona será capaz de comprender como realizar cualquier operación disponible en lex.gal.

Cabe destacar que en todo momento se puede volver a la página principal presionando en la palabra lex.gal, situada en la parte superior izquierda de la aplicación.

B.1. Página de inicio de sesión

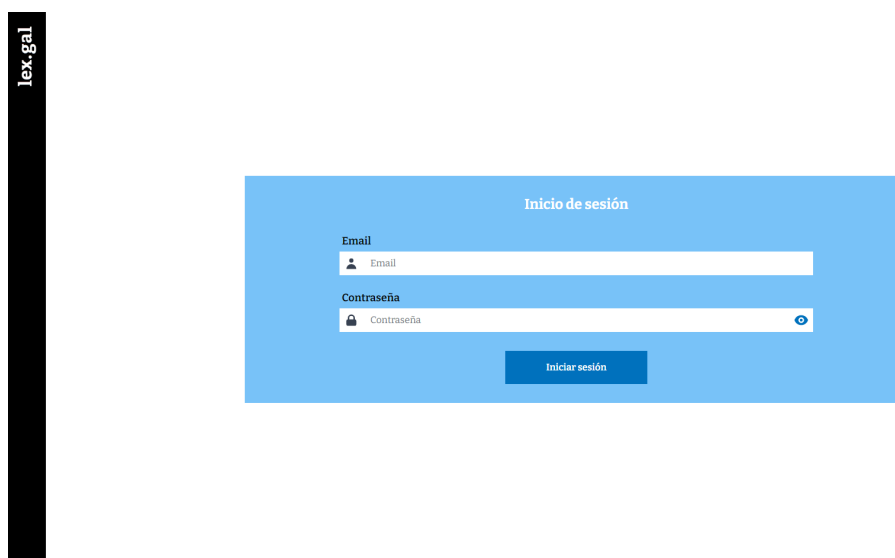


Figura B.1: Página de inicio de sesión.

En un primer momento el usuario ha de iniciar sesión para poder utilizar la aplicación. En este caso ha de introducir el correo electrónico y su contraseña. Si los datos de inicio de sesión son correctos, será redirigido automáticamente a la página principal de la aplicación. Se puede ver como se ha de hacer en la Figura B.1.

B.2. Página principal



Búsqueda en lex.gal

Insira o texto da búsqueda...

Búsqueda de normas...

Limpar Buscar

Atopáronse 9 resultados




DOG nº	Estado	Título	Departamento	Data	Accións
90	Borrador	ORDE do 4 de maio de 2020 pola que se establecen as bases que rexerán a concesión de subvencións a entidades de iniciativa social para a promoción de actuacións de servizos sociais comunitarios e inclusión social, e se convocan para os anos 2020, 2021 e 2022, cofinanciadas parcialmente polo Fondo Social Europeo e polo Fondo Europeo de Desenvolvemento Rexional no marco dos programas operativos FSE Galicia 2014-2020 e Feder Galicia 2014-2020 (código de procedemento B5631A).	Consellería de Política Social	11/05/2020	 
226	Validada	ORDE do 2 de novembro de 2020 pola que se modifica a Orde do 4 de maio de 2020 pola que se establecen as bases que rexerán a concesión de subvencións a entidades de iniciativa social para a promoción de actuacións de servizos sociais comunitarios e inclusión social, e se convocan para os anos 2020, 2021 e 2022, cofinanciadas parcialmente polo Fondo Social Europeo e polo Fondo Europeo de Desenvolvemento Rexional no marco dos programas operativos FSE Galicia 2014-2020 e Feder Galicia 2014-2020 (código de procedemento B5631A).	Consellería de Política Social	09/11/2020	 
161	Borrador	ORDE do 22 de xullo de 2020 pola que se desenvolve a normativa para o exercicio da docencia nos centros privados nas ensinanza de educación infantil, primaria, secundaria obrigatoria, bacharelato e formación profesional, e se establece o procedemento de habilitación (código de procedemento ED302B).	Consellería de Educación, Universidade e Formación Profesional	12/08/2020	 
61	Validada	ORDE do 22 de marzo de 2021 pola que se regulan as bases que rexerán o procedemento de concesión de axudas para a posta en marcha de casas do maior e se procede á súa convocatoria en réxime de concorrencia competitiva para os anos 2021, 2022 e 2023 (código de procedemento B5212A).	Consellería de Política Social	31/03/2021	 

Figura B.2: Página principal/búsquedas.

Una vez ha sido redirigido a la página principal de la aplicación, se puede observar que se muestran todas las leyes almacenadas en la aplicación, como se ve en la Figura B.2. En estas se pueden diferenciar fácilmente cinco columnas: el número del DOG, el estado (validada o borrador), el título/sumario de la ley, el departamento que lo publica, la fecha de publicación y posibles acciones a realizar. Estas acciones son: el ojo para previsualizar la ley en lex.gal (explicada en la Sección B.5. Página de previsualización de una ley de lex.gal), y el lápiz para editar la ley (explicada en la Sección B.5. Página de edición de una ley de lex.gal).

Para introducir cualquier texto que se desee buscar, se ha de insertar en el cuadro de búsqueda situado en el interior del cuadro azul. En caso de querer limpiar este texto, se puede pinchar en “Limpar”. Además, se permite navegar entre diferentes páginas pinchando en los números situados encima y debajo de la tabla. El botón con doble flecha izquierda redirige a la primera página, y el de doble flecha derecha a la última.

Si se pincha en el botón con el icono de una persona situado en la parte inferior izquierda, se abre el siguiente menú:

The screenshot shows the lex.gal website interface. At the top, there is a search bar with the text "Búsqueda en lex.gal" and a placeholder "Insira o texto da búsqueda...". Below the search bar, there is a table with 5 columns: DOG nº, Estado, Título, Departamento, and Data. The table contains 4 rows of results. On the left side, there is a user menu for "Pablo Tarrío Otero" with options "Registrar usuarios" and "Cerrar sesión". At the bottom left, there is a user icon and the name "Pablo".

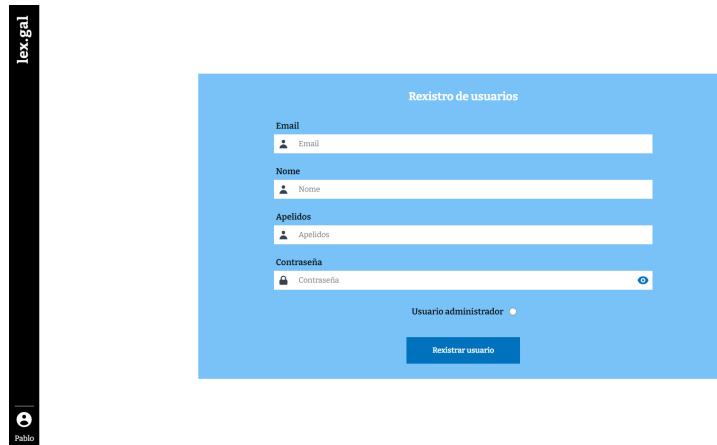
DOG nº	Estado	Título	Departamento	Data	Accións
90	Borrador	ORDE do 4 de maio de 2020 pola que se establecen as bases que rexerán a concesión de subvencións a entidades de iniciativa social para a promoción de actuacións de servizos sociais comunitarios e inclusión social, e se convocan para os anos 2020, 2021 e 2022, cofinanciadas parcialmente polo Fondo Social Europeo e polo Fondo Europeo de Desenvolvemento Rexional no marco dos programas operativos FSE Galicia 2014-2020 e Feder Galicia 2014-2020 (código de procedemento B5631A).	Consellería de Política Social	11/05/2020	
226	Validada	ORDE do 2 de novembro de 2020 pola que se modifica a Orde do 4 de maio de 2020 pola que se establecen as bases que rexerán a concesión de subvencións a entidades de iniciativa social para a promoción de actuacións de servizos sociais comunitarios e inclusión social, e se convocan para os anos 2020, 2021 e 2022, cofinanciadas parcialmente polo Fondo Social Europeo e polo Fondo Europeo de Desenvolvemento Rexional no marco dos programas operativos FSE Galicia 2014-2020 e Feder Galicia 2014-2020 (código de procedemento B5631A).	Consellería de Política Social	09/11/2020	
	Borrador	ORDE do 22 de xullo de 2020 pola que se desenvolve a normativa para o exercicio da docencia nos centros privados nas ensinanzas de educación infantil, primaria, secundaria obrigatoria, bacharelato e formación profesional, e se establece o procedemento de habilitación (código de procedemento ED302B).	Consellería de Educación, Universidade e Formación Profesional	12/08/2020	
61	Validada	ORDE do 22 de marzo de 2021 pola que se regulan as bases que rexerán o procedemento de concesión de axudas para a posta en marcha de casas do maior e se procede á súa convocatoria en réxime de concorrencia competitiva para os anos 2021, 2022 e 2023 (código de procedemento B5212A).	Consellería de Política Social	31/03/2021	

Figura B.3: Menú de usuarios.

En este menú, que se puede observar en la Figura B.3, se muestra el nombre de usuario y posibles acciones a realizar por este. En primer lugar, se muestra la acción de Registrar usuarios (la cual solo es visible para usuarios administradores). Si se pincha en este botón, se redirige a la página de Registro de usuarios, la cual se explica en la Sección B.3. Página de registro de usuarios. Si se pincha en el botón de Cerrar sesión, se cierra la sesión en la página y se redirige al usuario a la página de Inicio de sesión, explicada en la Sección B.1. Página de inicio de sesión.

Por último, existe un último botón el cual tiene una flecha hacia abajo, y se sitúa encima del icono de la persona. Si se pincha en dicho botón, se abre una pequeña pestaña encima de la actual, la cual permitirá importar leyes del DOG, y se explica en la Sección. B.4 Pestaña para importar del DOG.

B.3. Página de registro de usuarios



lex.gal

Registro de usuarios

Email
Email

Nome
Nome

Apellidos
Apellidos

Contraseña
Contraseña

Usuario administrador ☐

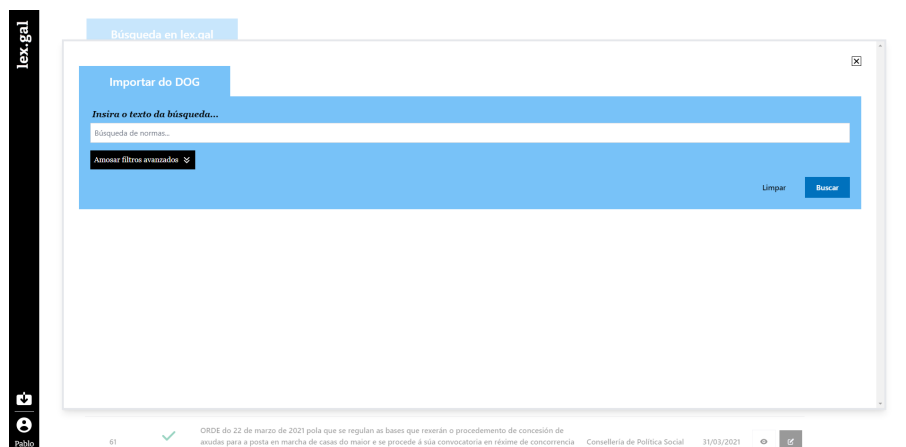
Registrar usuario

Pablo

Figura B.4: Página de registro de usuarios.

En esta página, que se ilustra en la Figura B.4, se permite el registro de usuarios, aunque esta solo puede ser accedida por usuarios administradores. En ella se puede introducir el correo electrónico, nombre, apellidos y contraseña. Por último, también se permite indicar si el usuario a registrar será administrador o no. Una vez se pincha en el botón de registrar usuario, se indica que el usuario ha sido registrado en la aplicación, y permite seguir registrando más usuarios.

B.4. Pestaña para importar del DOG



lex.gal

Búsqueda en lex.gal

Importar del DOG

Intira o texto da búsqueda...

Búsqueda de normas...

Mostrar filtro avanzado

Limpiar

Buscar

Pablo

61

Validada

ORDE do 22 de marzo de 2021 pola que se regulan as bases que rexerán o procedemento de concesión de axudas para a posta en marcha de casas do maior e se procede á súa convocatoria en réxime de concorrencia

Consellería de Política Social

31/03/2021

Figura B.5: Pestaña para importar del DOG - I.

En esta pestaña, Figura B.5, se permite buscar leyes en el DOG. Al igual que en la búsqueda de la página principal, tenemos un campo de texto, un botón para limpiar el campo, y el propio botón de buscar. Si pinchamos en “Amosar filtros avanzados”, se mostrarán una serie de filtros a la hora de realizar la búsqueda en el DOG.

Una vez realizada la búsqueda, se deberá ver algo semejante a lo siguiente:

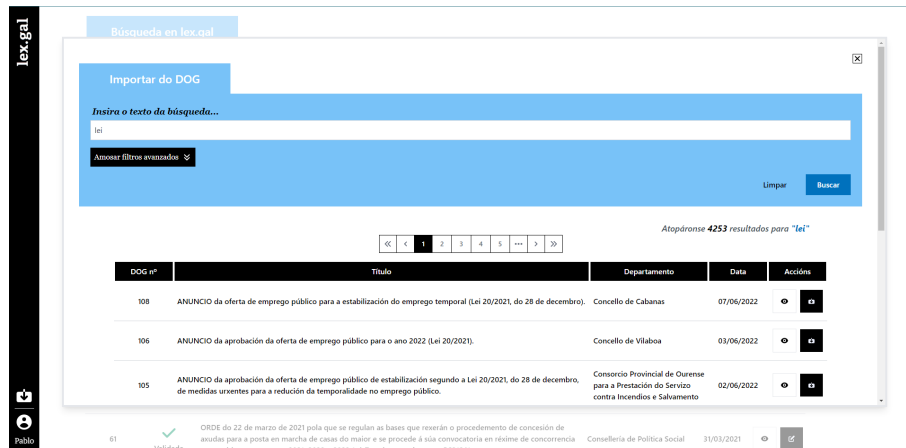


Figura B.6: Pestaña para importar del DOG - II.

En la Figura B.6 se muestra una tabla con las leyes del DOG encontradas, donde encontramos una serie de acciones por cada ley. Si se pincha en el botón del ojo, se podrá previsualizar la ley en el propio DOG, como se puede observar:



Figura B.7: Previsualización de ley del DOG.

En el caso de pinchar en el botón con una flecha hacia abajo de la Figura B.7, se importará una ley a la aplicación de lex.gal, pasando a estar disponible en la página principal del programa.

B.5. Página de previsualización de una ley de lex.gal

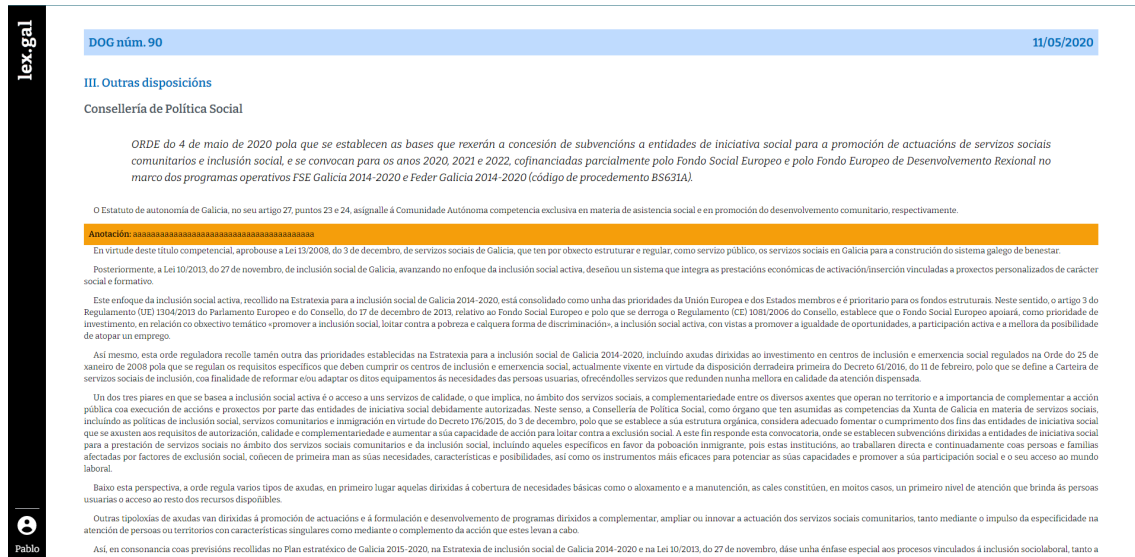


Figura B.8: Página de previsualización de una ley de lex.gal.

En esta página, mostrada en la Figura B.8, se puede previsualizar el contenido de una ley de lex.gal. Se mostrará en ella:

- En el recuadro azul el número del DOG y la fecha de publicación en el DOG.
- Con color de letra azul el título de la sección a la que pertenece.
- Con color gris el publicador del documento.
- Con letra cursiva el título/sumario de la ley.
- En texto plano, el contenido de la propia ley.
- Con fondo naranja, las anotaciones realizadas sobre un párrafo de la ley.

B.6. Página de edición de una ley de lex.gal

lex.gal

Datos de cabeceira ▾

ORDE do 4 de maio de 2020 pola que se establecen a ... Previsualizar

O Estatuto de autonomía de Galicia, no seu artigo 27, puntos 23 e 24, asigne á Comunidade Autónoma competencia exclusiva en materia de asistencia social e en promoción do desenvolvemento comunitario, respectivamente.

En virtude deste título competencial, aprobouse a Lei 13/2008, do 3 de decembro, de servizos sociais de Galicia, que ten por obxecto estruturar e regular, como servizo público, os servizos sociais en Galicia para a construción do sistema galego de benestar.

Posteriormente, a Lei 10/2013, do 27 de novembro, de inclusión social de Galicia, avanzando no enfoque da inclusión social activa, deseñou un sistema que integra as prestacións económicas de activación/inserción vinculadas a proxectos personalizados de carácter social e formativo.

Este enfoque da inclusión social activa, recollido na Estratexia para a inclusión social de Galicia 2014-2020, está consolidado como unha das prioridades da Unión Europea e dos Estados membros e é prioritario para os fondos estruturais. Neste sentido, o artigo 3 do Regulamento (UE) 1304/2013 do Parlamento Europeo e do Consello, do 17 de decembro de 2013, relativo ao Fondo Social Europeo e polo que se derroga o Regulamento (CE) 1081/2006 do Consello, establece que o Fondo Social Europeo apoiará, como prioridade de investimento, en relación co obxectivo temático «promover a inclusión social, loitar contra a pobreza e calquera forma de discriminación», a inclusión social activa, con vistas a promover a igualdade de oportunidades, a participación activa e a mellora da posibilidade de atopar un emprego.

Cambios Leis vinc. Notas

Validar e publicar Gardar como borrador Eliminar de lex.gal Sair sen gardar

Pablo

Figura B.9: Página de edición de una ley de lex.gal.

La página de edición, que se puede ver en la Figura B.9, es donde se centran la mayor parte de las funcionalidades de la aplicación. En ella, observamos en la parte superior un cuadro azul denominado “Datos de cabeceira”, que, al desplegarse/replegarse, muestra/oculta los datos de cabecera de la ley. Los campos de texto de color blanco permiten la edición, mientras que el resto son invariables.

lex.gal

Datos de cabeceira ▴

Título
ORDE do 4 de maio de 2020

Sumario
ORDE do 4 de maio de 2020 pola que se establecen as bases que rexerán a concesión de subvencións a entidades de iniciativa social para a promoción de actuacións de servizos sociais comunitarios e inclusión social, e se convocan para os anos 2020, 2021 e 2022, cofinanciadas parcialmente polo Fondo Social Europeo e polo Fondo Europeo de Desenvolvemento Rexional no marco dos programas operativos FSE Galicia 2014-2020 e Feder Galicia 2014-2020 (código de

Data de publicación
dd/mm/aaaa

Referencia da publicación
Referencia de publicación...

Ano
2022

Versión
20220607

Referencia
Referencia da norma...

Data do DOG

Data de entrada en vigor
dd/mm/aaaa

Estado
1 - En vigor

Nome ficheiro
fj: ficheiro.pdf

Colectivo
21005 - Cidadáns

Organización
20008 - Traballo e Benestar

Rango
8 - Orde

Sección
6 - III. Outras disposicións

Área temática
-- Selecciona unha opción --

Número do DOG

Validar e publicar Gardar como borrador Eliminar de lex.gal Sair sen gardar

Pablo

Figura B.10: Datos de cabecera de una ley de lex.gal.

Si se pincha en el botón de “Previsualizar” que se ve en la Figura B.10, se

redirige a la página de previsualización de la ley que se está editando en lex.gal. Esta sección aparece explicada en la Sección B.5. Página de previsualización de una ley de lex.gal.

Si se realiza click derecho sobre cualquier párrafo del texto, se mostrará un menú donde podremos copiar o pegar contenido, proponer un cambio o añadir una anotación al párrafo.



Figura B.11: Menú de edición de un párrafo.

Si pinchamos en el botón “Proponer cambio” de la Figura B.11, se abrirá una pestaña como la siguiente, donde podremos proponer un cambio. El cambio se almacenará en la pestaña de cambios de la derecha de la página, el cual puede ser desplegado para ver el cambio. Se podrá ver en el texto principal con un fondo de color verde.

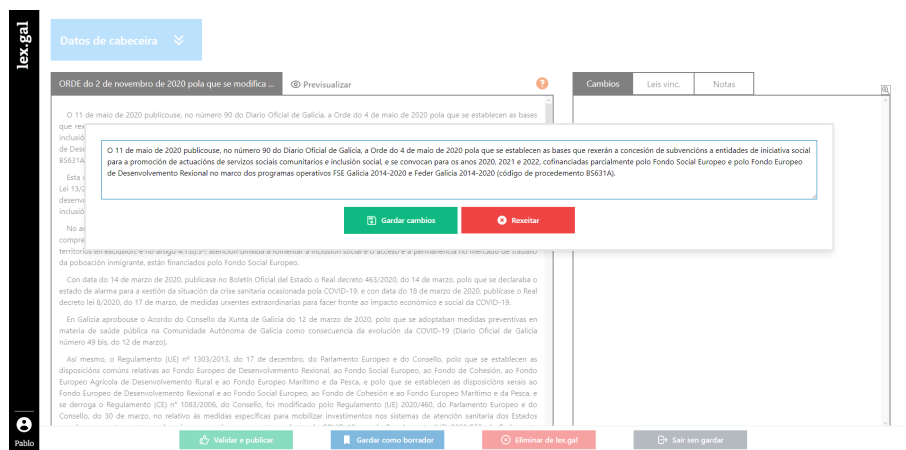


Figura B.12: Pestaña de edición de un párrafo.



Figura B.13: Pestaña con un cambio introducido.

Se muestra en las figuras Figura B.12 y Figura B.13 el resultado de realizar un cambio.

Si pinchamos en el botón “Engadir anotación ao parágrafo” de la Figura B.11, se abrirá una pestaña como la siguiente, donde podremos añadir una anotación al párrafo. La anotación se almacenará en la pestaña de notas de la derecha de la página, el cual puede ser desplegado para ver la nota completa y añadir comentarios sobre ella. Se podrá ver en el texto principal con un fondo de color naranja.

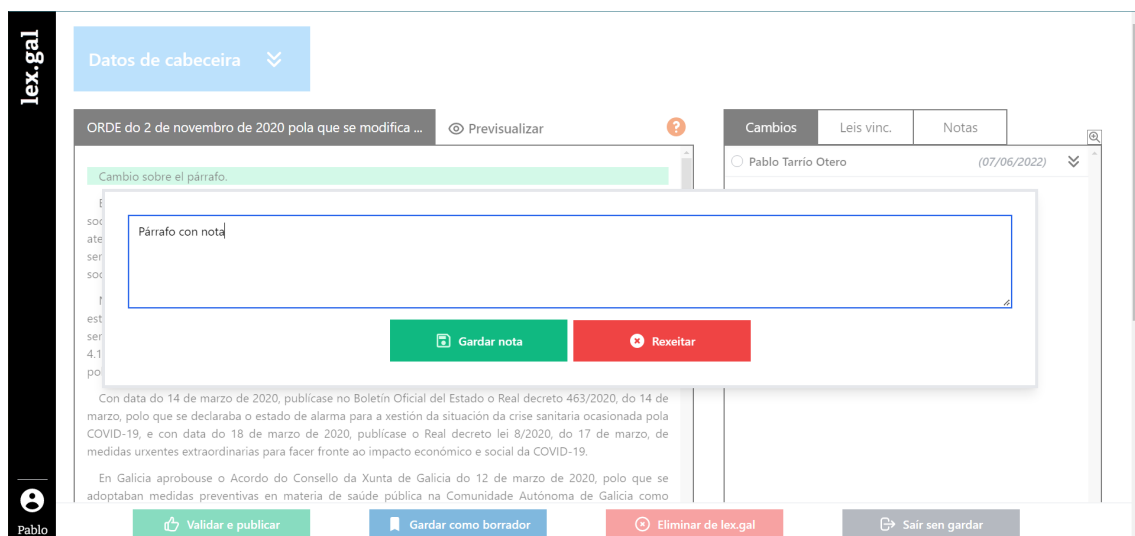


Figura B.14: Pestaña para añadir una nota a un párrafo.



Figura B.15: Pestaña con una nota introducida.

Se muestra en las figuras Figura B.14 y Figura B.15 el resultado de añadir una anotación.

En ambos casos (cambios y notas), se podrá ir al párrafo donde se ha añadido, resolver/descartar una selección de cambios/notas y resolver/descartar todos los cambios/notas.

En caso de presionar sobre la pestaña de “Leis vinc.”, se podrán observar las leyes vinculadas a una ley. En caso de poder modificar una ley vinculada, se podrá pinchar sobre esta, accediendo a la pestaña de leyes vinculadas, explicada en la Sección B.7. Pestaña de edición de leyes vinculadas. En caso de querer añadir una nueva ley vinculada, se abrirá una pequeña sección donde añadir la ley.



Figura B.16: Apartado de leyes vinculadas.

Para finalizar con esta página, se puede validar y publicar una ley, o también guardar como borrador (realizan la misma operación pero guardan con distintos estados la ley principal y sus leyes vinculadas), se puede eliminar una ley o se puede salir sin guardar, siendo redirigido siempre a la página principal. Y además, se muestra un mensaje de aviso antes de realizar cada operación. Todo ello se muestra en la Figura B.16.

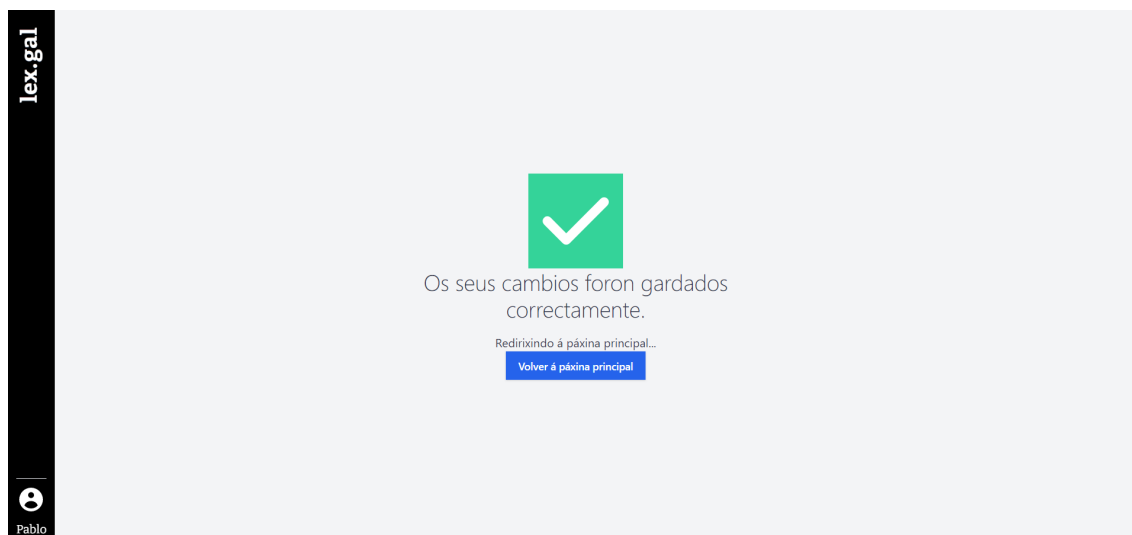


Figura B.17: Mensaje de cambios guardados.

En el caso de “Validar e publicar” o “Gardar como borrador”, se mostrará el siguiente mensaje como indica la Figura B.17:

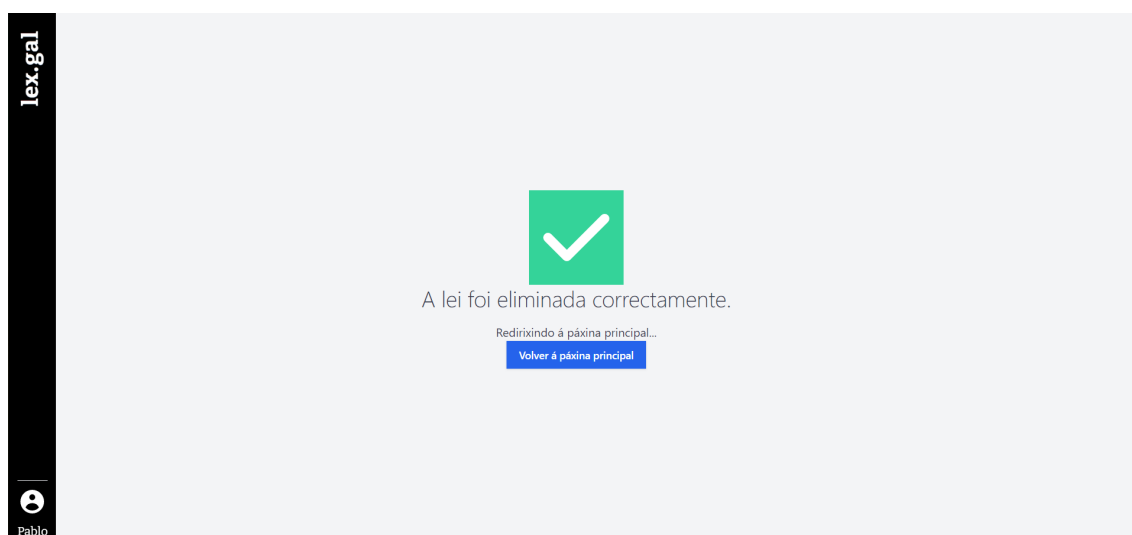


Figura B.18: Mensaje de ley eliminada.

En el caso de “Eliminar de lex.gal”, se mostrará el siguiente mensaje como indica la Figura B.18:

B.7. Pestaña de edición de leyes vinculadas

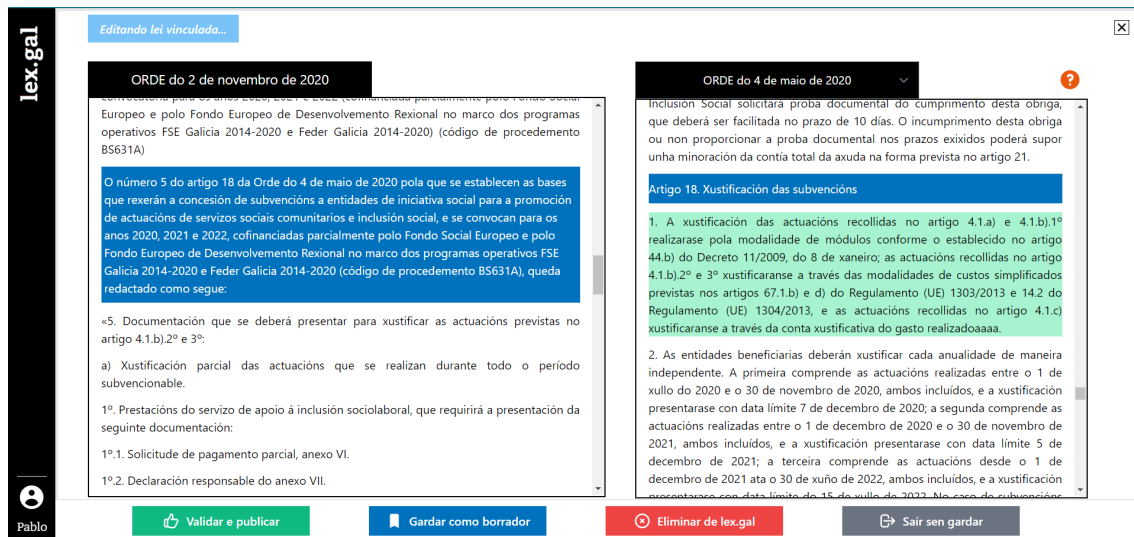


Figura B.19: Pestaña de edición de leyes vinculadas.

Como última pestaña relacionada con la edición (Figura B.19), encontramos la pestaña de edición de las leyes vinculadas. En esta, se muestra la ley principal a la izquierda de la pantalla, y la ley vinculada a modificar a la derecha. En el caso de existir más de una ley a modificar, esta se podría escoger en el seleccionable.

Los párrafos marcados con un fondo de color azul en la ley principal, indican la sección donde se ha de modificar la ley vinculada. Si se pincha sobre este párrafo, se redirige a la sección de la ley vinculada donde se ha de realizar el cambio. Por último, cualquier cambio realizado se marcará con fondo verde.

Para volver a la pantalla de edición de la ley principal, se ha de cerrar pinchando en la X situada en la parte superior derecha.

B.8. Página no encontrada

En el caso de no encontrar una página en la aplicación, se redirigirá a una página de aviso de que no se encuentra la página solicitada. Esta redirigirá a la página principal tras 10 segundos.

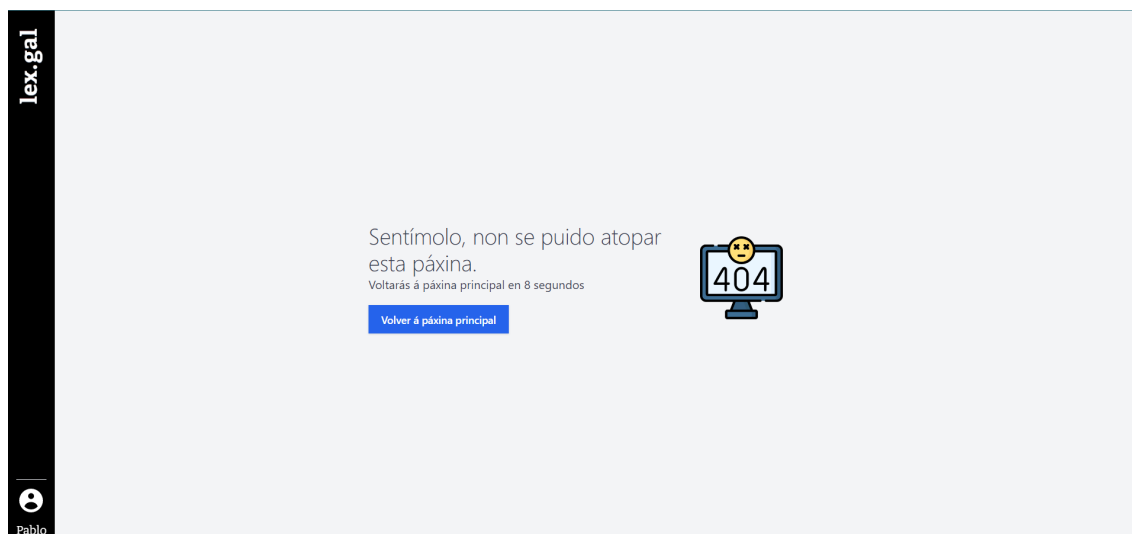


Figura B.20: Páxina no encontrada.

Se muestra en la Figura B.20 la página que indica que no se ha encontrado la página buscada por el usuario.

Apéndice C

Requisitos

C.1. Requisitos funcionales

FRQ-01	Iniciar sesión
Versión	1.0 (05/06/2022)
Descripción	El sistema deberá permitir al usuario iniciar sesión en la aplicación. Para ello ha de introducir su email y su contraseña, generando un token JWT que permite el uso de la aplicación de forma segura.
Importancia	Vital
Urgencia	Inmediatamente

Cuadro C.1: Requisito funcional 01. Iniciar sesión.

FRQ-02	Cerrar sesión
Versión	1.0 (05/06/2022)
Descripción	El sistema deberá permitir al usuario cerrar sesión en la aplicación. En este caso se elimina el token JWT del almacenamiento local, cerrando la sesión.
Importancia	Quedaría bien
Urgencia	Puede esperar

Cuadro C.2: Requisito funcional 02. Cerrar sesión.

FRQ-03	Registrar usuarios
Versión	1.0 (05/06/2022)
Descripción	El sistema deberá permitir al usuario con rol de administrador registrar usuarios. Para ello, el administrador introducirá el email, nombre, apellidos, contraseña y rol del usuario a registrar.
Importancia	Quedaría bien
Urgencia	Puede esperar

Cuadro C.3: Requisito funcional 03. Registrar usuarios.

FRQ-04	Búsqueda de leyes en el DOG
Versión	1.0 (05/06/2022)
Descripción	El sistema deberá permitir al usuario buscar las leyes deseadas en el DOG. Se podrán filtrar leyes según los siguientes parámetros de búsqueda: texto, buscar solo en el título, buscar por frase exacta, número del DOG desde/hasta, criterio de ordenación, colectivo, organización, rango, sección y área temática.
Importancia	Vital
Urgencia	Inmediatamente

Cuadro C.4: Requisito funcional 04. Búsqueda de leyes en el DOG.

FRQ-05	Previsualizar leyes en el DOG
Versión	1.0 (05/06/2022)
Descripción	El sistema permitirá al usuario previsualizar una ley del DOG en su propia página web. Se abrirá una pestaña nueva del DOG con la ley seleccionada.
Importancia	Quedaría bien
Urgencia	Puede esperar

Cuadro C.5: Requisito funcional 05. Previsualizar leyes en el DOG.

FRQ-06	Importar leyes del DOG
Versión	1.0 (05/06/2022)
Descripción	El sistema deberá permitir al usuario importar una ley del DOG a la base de datos de lex.gal. Importará todos los datos relevantes de una ley: sumario, publicador, fecha de publicación en el DOG, colectivo, organización, rango, sección, área temática, número del DOG y el texto del documento. Todas las leyes serán importadas como borrador.
Importancia	Vital
Urgencia	Inmediatamente

Cuadro C.6: Requisito funcional 06. Importar leyes del DOG.

FRQ-07	Búsqueda de leyes en lex.gal
Versión	1.0 (05/06/2022)
Descripción	El sistema deberá permitir al usuario buscar las leyes en la base de datos de lex.gal. En este caso solo se podrá filtrar por el texto contenido en el sumario.
Importancia	Vital
Urgencia	Inmediatamente

Cuadro C.7: Requisito funcional 07. Búsqueda de leyes en lex.gal.

FRQ-08	Previsualizar leyes en lex.gal
Versión	1.0 (05/06/2022)
Descripción	El sistema deberá permitir al usuario previsualizar el contenido del documento de una ley almacenada en lex.gal.
Importancia	Vital
Urgencia	Inmediatamente

Cuadro C.8: Requisito funcional 08. Previsualizar leyes en lex.gal.

FRQ-09	Cargar leyes de lex.gal
Versión	1.0 (05/06/2022)
Descripción	El sistema deberá permitir al usuario cargar los datos de cualquier ley de lex.gal. Se importarán los datos solicitados en todo momento, como el sumario, texto del documento, datos de cabecera, leyes vinculadas, cambios realizados y las anotaciones añadidas.
Importancia	Vital
Urgencia	Inmediatamente

Cuadro C.9: Requisito funcional 09. Cargar leyes de lex.gal.

FRQ-10	Proponer cambios en una ley de lex.gal
Versión	1.0 (05/06/2022)
Descripción	El sistema deberá permitir al usuario proponer cambios sobre cualquier párrafo de una ley de lex.gal. Los cambios serán introducidos en el documento para poder previsualizar la norma con los cambios propuestos.
Importancia	Quedaría bien
Urgencia	Puede esperar

Cuadro C.10: Requisito funcional 10. Proponer cambios en una ley de lex.gal.

FRQ-11	Añadir anotaciones sobre párrafos de una ley de lex.gal
Versión	1.0 (05/06/2022)
Descripción	El sistema deberá permitir al usuario añadir anotaciones sobre cualquier párrafo del documento de una ley de lex.gal. Se podrán añadir comentarios sobre la anotación realizada.
Importancia	Quedaría bien
Urgencia	Puede esperar

Cuadro C.11: Requisito funcional 11. Añadir anotaciones sobre párrafos de una ley de lex.gal.

FRQ-12	Cargar leyes vinculadas a una ley de lex.gal
Versión	1.0 (05/06/2022)
Descripción	El sistema deberá permitir al usuario cargar las leyes vinculadas a una ley de lex.gal. Se importará tanto el nombre de una ley vinculada, como el documento en el caso de que se realice alguna modificación sobre ella.
Importancia	Vital
Urgencia	Inmediatamente

Cuadro C.12: Requisito funcional 12. Cargar leyes vinculadas a una ley de lex.gal.

FRQ-13	Añadir leyes vinculadas a una ley de lex.gal
Versión	1.0 (05/06/2022)
Descripción	El sistema deberá permitir al usuario añadir manualmente una ley vinculada a cualquier ley de lex.gal. Se mostrará en la lista de leyes vinculadas a la propia ley.
Importancia	Vital
Urgencia	Inmediatamente

Cuadro C.13: Requisito funcional 13. Añadir leyes vinculadas a una ley de lex.gal.

FRQ-14	Proponer cambios en leyes vinculadas a una ley de lex.gal
Versión	1.0 (05/06/2022)
Descripción	El sistema deberá permitir al usuario proponer cambios sobre cualquier ley vinculada que modifique una ley de lex.gal. Se mostrará en la ley principal donde se han de realizar los cambios en la ley vinculada, permitiendo un método sencillo para el usuario a la hora de proponer el cambio.
Importancia	Vital
Urgencia	Inmediatamente

Cuadro C.14: Requisito funcional 14. Proponer cambios en leyes vinculadas a una ley de lex.gal.

FRQ-15	Eliminar una ley de lex.gal
Versión	1.0 (05/06/2022)
Descripción	El sistema deberá permitir al usuario eliminar cualquier ley de lex.gal. La ley será eliminada definitivamente de la base de datos.
Importancia	Quedaría bien
Urgencia	Puede esperar

Cuadro C.15: Requisito funcional 15. Eliminar una ley de lex.gal.

C.2. Requisitos no funcionales

NFR-01	Tiempo de respuesta < 2 segundos
Versión	1.0 (05/06/2022)
Descripción	El sistema deberá responder en menos de 2 segundos, sin tener en cuenta el tiempo que tarde la red en enviar la información de vuelta al usuario.
Importancia	Vital
Urgencia	Inmediatamente

Cuadro C.16: Requisito no funcional 01. Tiempo de respuesta < 2 segundos.

NFR-02	Los datos sensibles del usuario se almacenan encriptados
Versión	1.0 (05/06/2022)
Descripción	El sistema deberá encriptar los datos sensibles del usuario como las contraseñas en la base de datos de lex.gal. Se utilizará un algoritmo SHA-256 para encriptar.
Importancia	Vital
Urgencia	Inmediatamente

Cuadro C.17: Requisito no funcional 02. Los datos sensibles del usuario se almacenan encriptados.

NFR-03	No permitir que un rol realice operaciones de otro
Versión	1.0 (05/06/2022)
Descripción	El sistema deberá restringir aquellas operaciones exclusivas del administrador para el resto de usuarios.
Importancia	Vital
Urgencia	Inmediatamente

Cuadro C.18: Requisito no funcional 03. No permitir que un rol realice operaciones de otro.

NFR-04	Permanecer operativo durante el horario de trabajo
Versión	1.0 (05/06/2022)
Descripción	El sistema deberá permanecer operativo durante todo el horario de trabajo de los empleados de lex.gal.
Importancia	Vital
Urgencia	Inmediatamente

Cuadro C.19: Requisito no funcional 04. Permanecer operativo durante el horario de trabajo.

NFR-05	Tratamiento de documentos en lenguajes de etiquetas
Versión	1.0 (05/06/2022)
Descripción	El sistema deberá restringir el tratamiento de documentos a documentos escritos en lenguajes de etiquetas, como HTML o XML.
Importancia	Vital
Urgencia	Inmediatamente

Cuadro C.20: Requisito no funcional 05. Tratamiento de documentos en lenguajes de etiquetas.

NFR-06	Tiempo de aprendizaje por el usuario < 1h
Versión	1.0 (05/06/2022)
Descripción	El sistema deberá tener un proceso de aprendizaje rápido y sencillo para el usuario, que en todo caso deberá ser menor de 1 hora.
Importancia	Vital
Urgencia	Inmediatamente

Cuadro C.21: Requisito no funcional 06. Tiempo de aprendizaje por el usuario < 1h.

NFR-07	Proporcionar mensajes de apoyo al usuario
Versión	1.0 (05/06/2022)
Descripción	El sistema deberá proporcionar mensajes de apoyo al usuario en todas las operaciones, con el objetivo de facilitar la navegación en la aplicación.
Importancia	Quedaría bien
Urgencia	Puede esperar

Cuadro C.22: Requisito no funcional 07. Proporcionar mensajes de apoyo al usuario.

NFR-08	Uso de tecnologías web y servicios RESTful
Versión	1.0 (05/06/2022)
Descripción	El sistema deberá emplear tecnologías web para realizar la aplicación, incluyendo HTML5, Tailwind como framework de CSS y React.js como framework de JavaScript. Además, todos los servicios serán RESTful, y serán implementados con Spring Boot.
Importancia	Vital
Urgencia	Inmediatamente

Cuadro C.23: Requisito no funcional 08. Uso de tecnologías web y servicios RESTful.

NFR-09	Base de datos NoSQL
Versión	1.0 (05/06/2022)
Descripción	El sistema deberá almacenar todos los datos en una base de datos NoSQL de MongoDB.
Importancia	Vital
Urgencia	Inmediatamente

Cuadro C.24: Requisito no funcional 09. Base de datos NoSQL.

NFR-10	Generación de tokens JWT
Versión	1.0 (05/06/2022)
Descripción	El sistema deberá generar tokens JWT cada vez que un usuario inicie sesión en la aplicación. De esta forma, los datos del usuario no quedan tan accesibles en la web, y será necesario utilizar un algoritmo <i>hash</i> para descryptarlos. En todo caso la contraseña no será almacenada en el token JWT.
Importancia	Vital
Urgencia	Inmediatamente

Cuadro C.25: Requisito no funcional 10. Generación de tokens JWT.

C.3. Requisitos de información

IRQ-01	Usuario
Versión	1.0 (05/06/2022)
Descripción	El sistema deberá almacenar la información correspondiente a los usuarios del sistema.
Datos específicos	<ul style="list-style-type: none"> - Correo electrónico. - Nombre. - Apellidos. - Contraseña. - Roles del usuario.
Importancia	Vital
Urgencia	Inmediatamente

Cuadro C.26: Requisito de información 01. Usuario.

IRQ-02	Documento
Versión	1.0 (05/06/2022)
Descripción	El sistema deberá almacenar la información correspondiente a las leyes de lex.gal.
Datos específicos	<ul style="list-style-type: none"> - Id. - Sumario. - Estado de la ley (validada/borrador). - Documento original. - Documento con los cambios realizados. - Datos de cabecera. - Cambios propuestos sobre la propia ley. - Anotaciones realizadas. - Leyes vinculadas. - Cambios propuestos sobre leyes vinculadas.
Importancia	Vital
Urgencia	Inmediatamente

Cuadro C.27: Requisito de información 02. Documento.

C.4. Casos de uso

UC-01	Iniciar sesión
Versión	1.0 (05/06/2022)
Dependencias	FRQ-01.
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario trate de iniciar sesión en la aplicación.
Precondición	El usuario debe de estar dado de alta en el sistema.
Secuencia normal	1. El usuario accede a la pantalla de inicio de sesión. 2. El usuario introduce sus credenciales en los campos de texto y pincha en Iniciar sesión. 3. El usuario accede a la aplicación con la sesión iniciada.
Postcondición	El usuario inicia sesión en lex.gal.
Importancia	Vital
Urgencia	Inmediatamente

Cuadro C.28: Caso de uso 01. Iniciar sesión.

UC-02	Cerrar sesión
Versión	1.0 (05/06/2022)
Dependencias	FRQ-02.
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario trate de cerrar sesión en la aplicación.
Precondición	El usuario debe de haber iniciado sesión en el sistema.
Secuencia normal	1. El usuario pincha en Cerrar sesión. 2. El usuario cierra la sesión en la aplicación.
Postcondición	El usuario cierra sesión en lex.gal.
Importancia	Vital
Urgencia	Inmediatamente

Cuadro C.29: Caso de uso 02. Cerrar sesión.

UC-03	Registrar usuario
Versión	1.0 (05/06/2022)
Dependencias	FRQ-03.
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario administrador trate de registrar un usuario en la aplicación.
Precondición	El usuario debe de haber iniciado sesión en el sistema y debe poseer el rol de administrador.
Secuencia normal	<ol style="list-style-type: none">1. El usuario administrador pincha en Registrar usuarios.2. El usuario administrador introduce los datos del usuario a registrar y pincha en Registrar usuario.3. El sistema almacena el usuario en la base de datos la aplicación.
Postcondición	Se registra el usuario en lex.gal.
Importancia	Quedaría bien
Urgencia	Puede esperar

Cuadro C.30: Caso de uso 03. Registrar usuario.

UC-04	Importar ley del DOG
Versión	1.0 (05/06/2022)
Dependencias	FRQ-04, FRQ-06.
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario importe una ley del DOG.
Precondición	El usuario debe de haber iniciado sesión en el sistema y debe tener acceso a Internet.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario pincha en el botón de Importar del DOG. 2. El usuario introduce el texto de búsqueda junto con sus filtros y pincha en Buscar. 3. El sistema muestra las leyes encontradas en el DOG. 4. El usuario pincha en el botón de Importar de la ley deseada. 5. El sistema almacena la ley en la base de datos de la aplicación.
Postcondición	Se almacena la ley seleccionada en lex.gal.
Excepciones	<ol style="list-style-type: none"> 1. El usuario no introduce ningún texto en el campo de búsqueda en el paso 2. A continuación, el caso de uso queda sin efecto. 2. No se encuentra ninguna ley a partir de los filtros en el paso 3. A continuación, el caso de uso queda sin efecto.
Importancia	Vital
Urgencia	Inmediatamente

Cuadro C.31: Caso de uso 04. Importar ley del DOG.

UC-05	Previsualizar una ley del DOG
Versión	1.0 (05/06/2022)
Dependencias	FRQ-04, FRQ-05.
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario previsualice una ley del DOG.
Precondición	El usuario debe de haber iniciado sesión en el sistema y debe tener acceso a Internet.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario pincha en el botón de Importar del DOG. 2. El usuario introduce el texto de búsqueda junto con sus filtros y pincha en Buscar. 3. El sistema muestra las leyes encontradas en el DOG. 4. El usuario pincha en el botón de Previsualizar de la ley deseada. 5. El sistema muestra al usuario el documento de la ley en la página web del DOG.
Postcondición	Se muestra al usuario la ley que desea previsualizar.
Excepciones	<ol style="list-style-type: none"> 1. El usuario no introduce ningún texto en el campo de búsqueda en el paso 2. A continuación, el caso de uso queda sin efecto. 2. No se encuentra ninguna ley a partir de los filtros en el paso 3. A continuación, el caso de uso queda sin efecto.
Importancia	Vital
Urgencia	Inmediatamente

Cuadro C.32: Caso de uso 05. Previsualizar una ley del DOG.

UC-06	Previsualizar una ley en lex.gal
Versión	1.0 (05/06/2022)
Dependencias	FRQ-07, FRQ-08, FRQ-09.
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario previsualice una ley de lex.gal.
Precondición	El usuario debe de haber iniciado sesión en el sistema.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario introduce el texto de búsqueda y pincha en Buscar. 2. El sistema muestra las leyes encontradas en lex.gal. 3. El usuario pincha en el botón de Previsualizar de la ley deseada. 4. El sistema muestra al usuario el documento de la ley de lex.gal.
Postcondición	Se muestra al usuario la ley que desea previsualizar.
Excepciones	1. No se encuentra ninguna ley a partir del texto en el paso 3. A continuación, el caso de uso queda sin efecto.
Importancia	Vital
Urgencia	Inmediatamente

Cuadro C.33: Caso de uso 06. Previsualizar una ley en lex.gal.

UC-07	Validar y publicar una ley y sus leyes vinculadas en lex.gal
Versión	1.0 (05/06/2022)
Dependencias	FRQ-07, FRQ-09, FRQ-10, FRQ-11, FRQ-12, FRQ-13, FRQ-14.
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario trate de validar y publicar una ley y sus leyes vinculadas de lex.gal.
Precondición	El usuario debe de haber iniciado sesión en el sistema.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario propone cambios, añade anotaciones, añade leyes vinculadas o edita leyes vinculadas. 2. El usuario pincha en “Validar e publicar”. 3. El sistema almacena los datos de la ley y su ley vinculada, siendo estas marcadas como validadas, en la base de datos. 4. El sistema muestra un mensaje de que la ley ha sido editada. 5. El sistema redirige al usuario a la página principal de lex.gal.
Postcondición	Se almacenan los datos editados en la respectiva ley.
Importancia	Vital
Urgencia	Inmediatamente

Cuadro C.34: Caso de uso 07. Validar y publicar una ley y sus leyes vinculadas en lex.gal.

UC-08	Guardar como borrador una ley y sus leyes vinculadas en lex.gal
Versión	1.0 (05/06/2022)
Dependencias	FRQ-07, FRQ-09, FRQ-10, FRQ-11, FRQ-12, FRQ-13, FRQ-14.
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario trate de guardar como borrador una ley y sus leyes vinculadas de lex.gal.
Precondición	El usuario debe de haber iniciado sesión en el sistema.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario propone cambios, añade anotaciones, añade leyes vinculadas o edita leyes vinculadas. 2. El usuario pincha en “Guardar como borrador”. 3. El sistema almacena los datos de la ley y su ley vinculada, siendo estas marcadas como borrador, en la base de datos. 4. El sistema muestra un mensaje de que la ley ha sido editada. 5. El sistema redirige al usuario a la página principal de lex.gal.
Postcondición	Se almacenan los datos editados en la respectiva ley.
Importancia	Vital
Urgencia	Inmediatamente

Cuadro C.35: Caso de uso 08. Guardar como borrador una ley y sus leyes vinculadas en lex.gal.

UC-09	Eliminar una ley en lex.gal
Versión	1.0 (05/06/2022)
Dependencias	FRQ-15.
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario elimine una ley de lex.gal.
Precondición	El usuario debe de haber iniciado sesión en el sistema.
Secuencia normal	<ol style="list-style-type: none">1. El usuario pincha en Eliminar lei de lex.gal.2. El sistema elimina la ley de lex.gal.3. El sistema muestra un mensaje de que la ley ha sido eliminada.4. El sistema redirige al usuario a la página principal de lex.gal.
Postcondición	La ley es eliminada de lex.gal.
Importancia	Quedaría bien
Urgencia	Puede esperar

Cuadro C.36: Caso de uso 09. Eliminar una ley en lex.gal.

Apéndice D

Diagramas de secuencia

D.1. Inicio de sesión

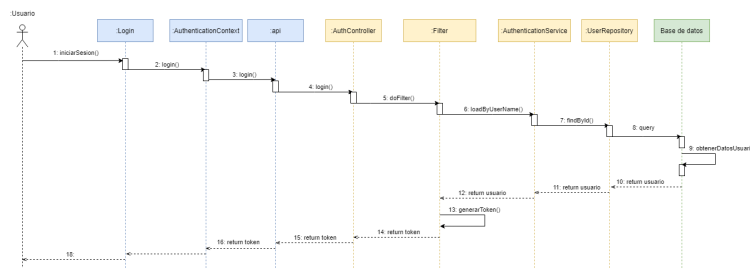


Figura D.1: Diagrama de secuencia 01. Inicio de sesión.

D.2. Cierre de sesión

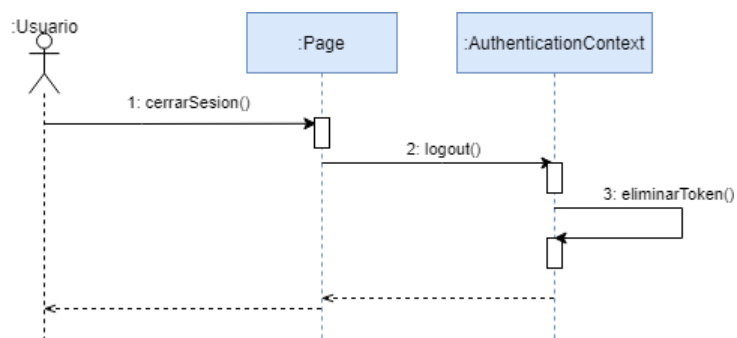


Figura D.2: Diagrama de secuencia 02. Cierre de sesión.

D.3. Registro de un usuario

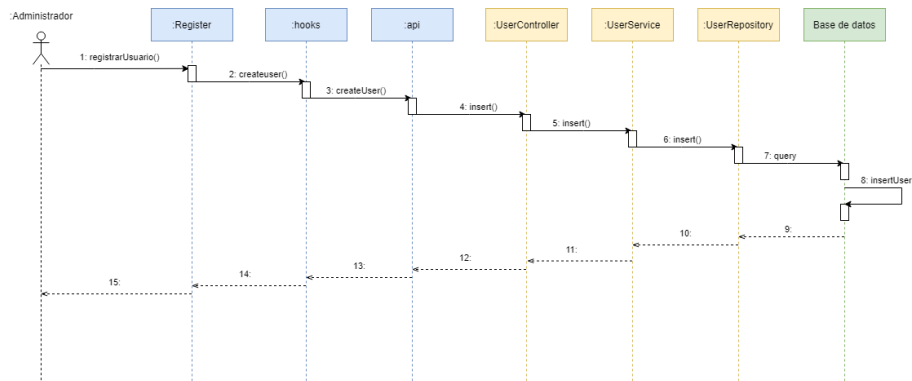


Figura D.3: Diagrama de secuencia 03. Registro de un usuario.

D.4. Previsualizar una ley del DOG

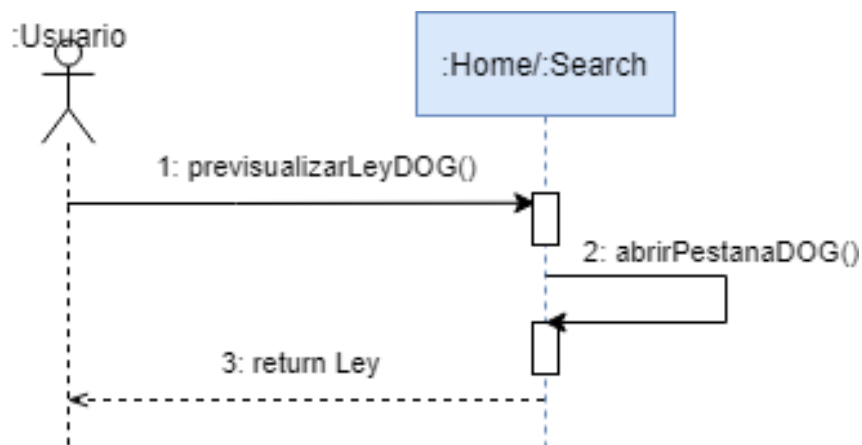


Figura D.4: Diagrama de secuencia 04. Previsualizar una ley del DOG.

D.5. Buscar una ley en el DOG

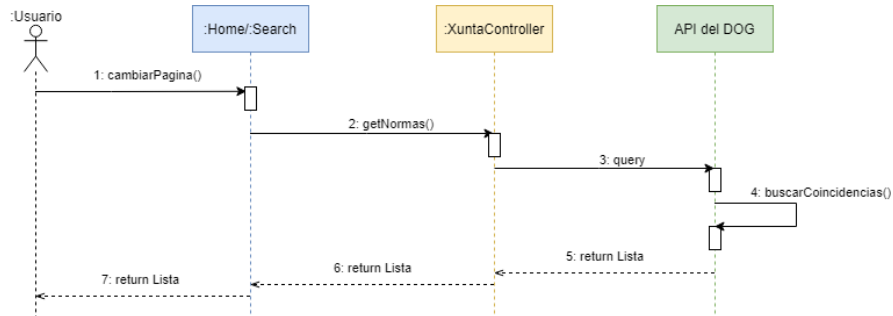


Figura D.5: Diagrama de secuencia 05. Buscar una ley en el DOG.

D.6. Importar una ley del DOG

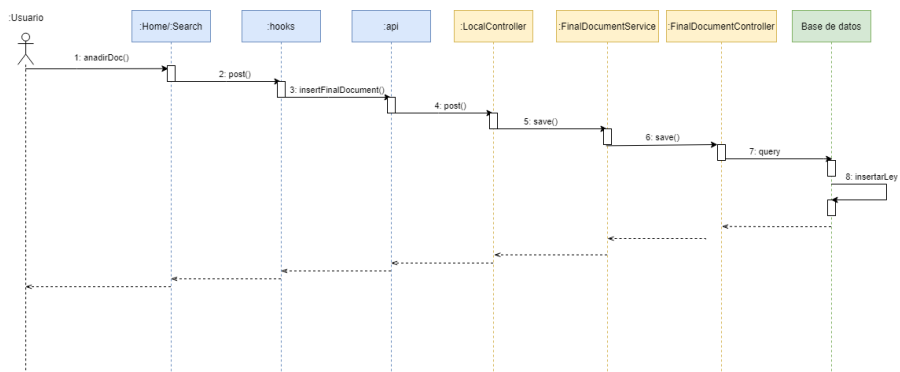


Figura D.6: Diagrama de secuencia 06. Importar una ley del DOG.

D.7. Buscar una ley en lex.gal

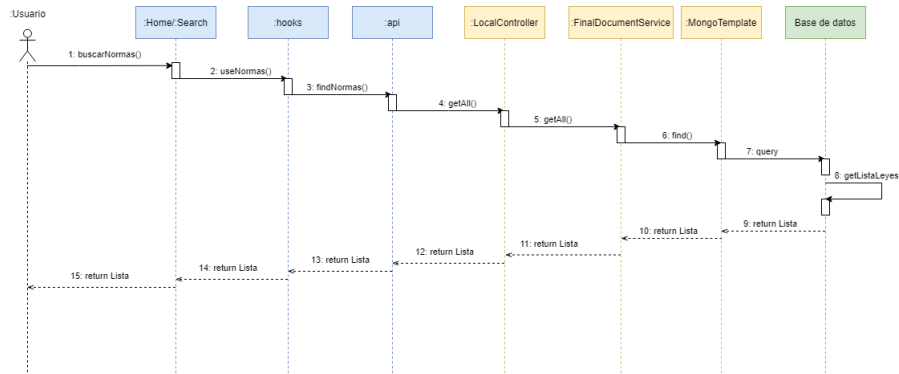


Figura D.7: Diagrama de secuencia 07. Buscar una ley en lex.gal.

D.8. Previsualizar una ley de lex.gal

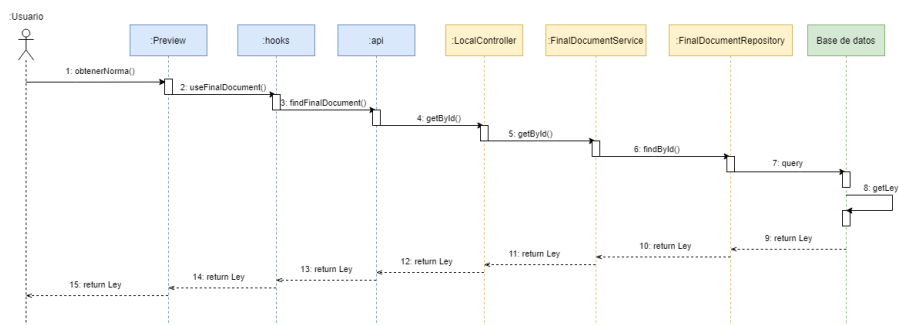


Figura D.8: Diagrama de secuencia 08. Previsualizar una ley de lex.gal.

D.9. Cargar una ley de lex.gal

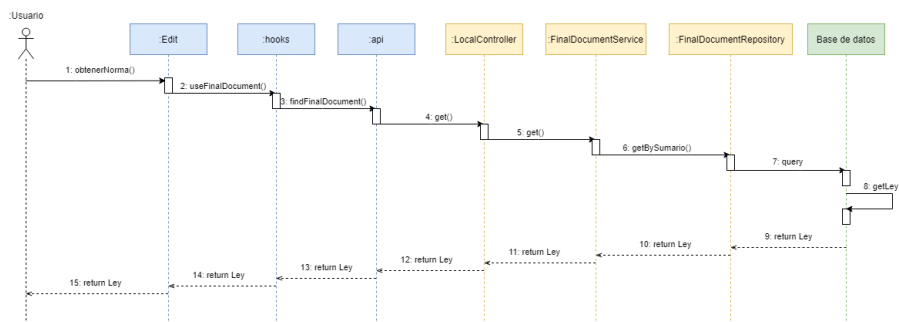


Figura D.9: Diagrama de secuencia 09. Cargar una ley de lex.gal.

D.10. Validar y publicar una ley de lex.gal

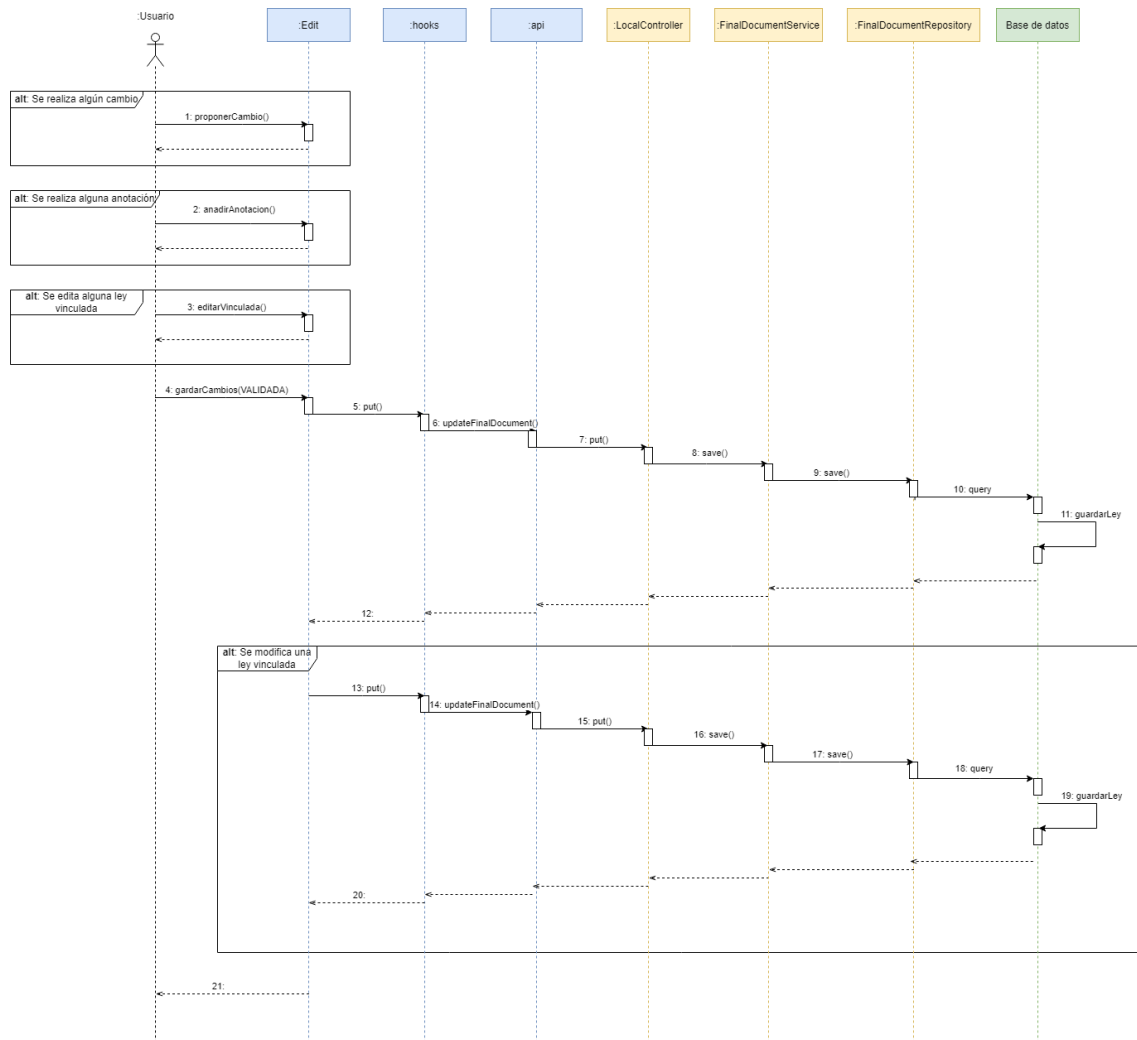


Figura D.10: Diagrama de secuencia 10. Validar y publicar una ley de lex.gal.

D.11. Guardar como borrador una ley de lex.gal

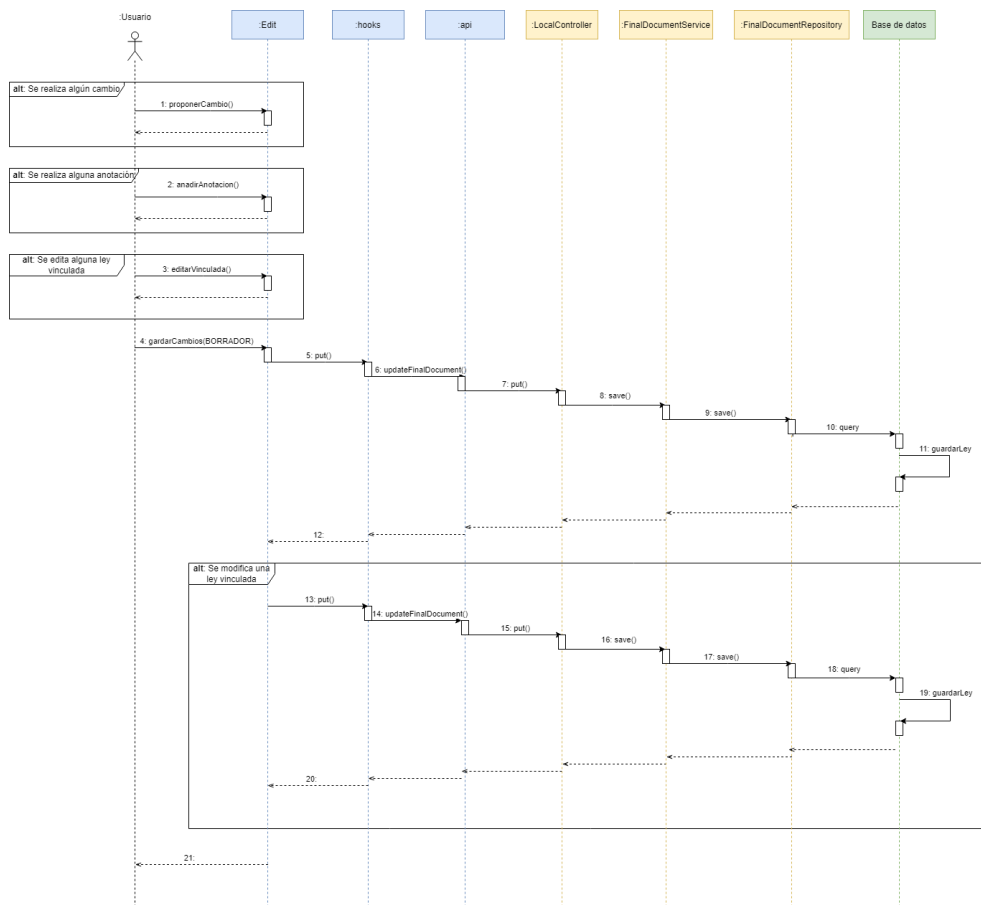


Figura D.11: Diagrama de secuencia 11. Guardar como borrador una ley de lex.gal.

D.12. Eliminar una ley de lex.gal

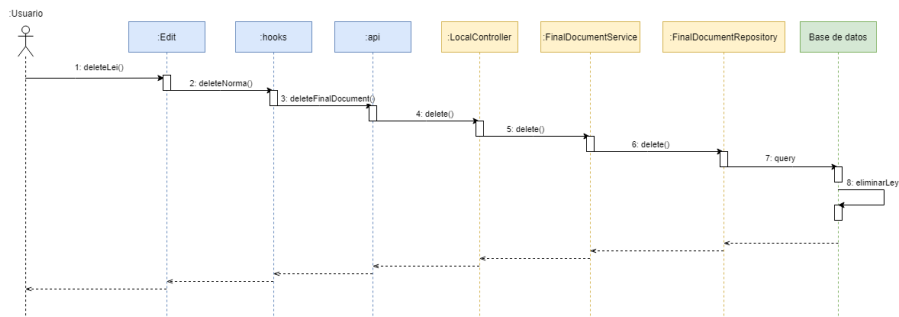


Figura D.12: Diagrama de secuencia 12. Eliminar una ley de lex.gal.

Apéndice E

Plantillas

E.1. Plantilla de requisitos funcionales

FRQ-XX	-
Versión	1.0 (XX/XX/XXXX)
Descripción	-.
Importancia	-
Urgencia	-

Cuadro E.1: Plantilla de requisitos funcionales.

E.2. Plantilla de requisitos no funcionales

NFR-XX	-
Versión	1.0 (XX/XX/XXXX)
Descripción	-.
Importancia	-
Urgencia	-

Cuadro E.2: Plantilla de requisitos no funcionales.

E.3. Plantilla de requisitos de información

IRQ-XX	-
Versión	1.0 (XX/XX/XXXX)
Descripción	-.
Datos específicos	-.
Importancia	-
Urgencia	-

Cuadro E.3: Plantilla de requisitos de información.

E.4. Plantilla de casos de uso

UC-XX	-
Versión	1.0 (XX/XX/XXXX)
Dependencias	-.
Descripción	-.
Precondición	-.
Secuencia normal	-.
Postcondición	-.
Excepciones	-.
Importancia	-
Urgencia	-

Cuadro E.4: Plantilla de casos de uso.

E.5. Plantilla de pruebas unitarias

PU-XX	-
Versión	1.0 (XX/XX/XXXX)
Dependencias	-.
Descripción	-.
Criterio de aceptación	-
Estado	-

Cuadro E.5: Plantilla de pruebas unitarias.

E.6. Plantilla de pruebas de integración

PI-XX	-
Versión	1.0 (XX/XX/XXXX)
Descripción	-.
Criterio de aceptación	-
Estado	-

Cuadro E.6: Plantilla de pruebas de integración.

Apéndice F

Licencia

Copyright © 2022 Pablo Tarrío Otero

El software y documentación asociada al mismo proporcionados en este trabajo de fin de grado está bajo licencia MIT [47], siendo esta la siguiente.

Por la presente se concede permiso, libre de cargos, a cualquier persona que obtenga una copia de este software y de los archivos de documentación asociados (el “Software”), a utilizar el *software* sin restricción, incluyendo sin limitación los derechos a usar, copiar, modificar, fusionar, publicar, distribuir, sublicenciar, y/o vender copias del *software*, y a permitir a las personas a las que se les proporcione el *software* a hacer lo mismo, sujeto a las siguientes condiciones:

El aviso de copyright anterior y este aviso de permiso se incluirán en todas las copias o partes sustanciales del *software*.

EL SOFTWARE SE PROPORCIONA “COMO ESTÁ”, SIN GARANTÍA DE NINGÚN TIPO, EXPRESA O IMPLÍCITA, INCLUYENDO PERO NO LIMITADO A GARANTÍAS DE COMERCIALIZACIÓN, IDONEIDAD PARA UN PROPÓSITO PARTICULAR E INCUMPLIMIENTO. EN NINGÚN CASO LOS AUTORES O PROPIETARIOS DE LOS DERECHOS DE AUTOR SERÁN RESPONSABLES DE NINGUNA RECLAMACIÓN, DAÑOS U OTRAS RESPONSABILIDADES, YA SEA EN UNA ACCIÓN DE CONTRATO, AGRAVIO O CUALQUIER OTRO MOTIVO, DERIVADAS DE, FUERA DE O EN CONEXIÓN CON EL SOFTWARE O SU USO U OTRO TIPO DE ACCIONES EN EL SOFTWARE.

Bibliografía

- [1] *Inicio - Diario Oficial de Galicia*. Disponible en <https://www.xunta.gal/diario-oficial-galicia/portalPublicoHome.do?lang=gl>. Consultado el 29 de abril de 2022.
- [2] *Legislación consolidada: Información y ayuda*. BOE. Disponible en https://www.boe.es/buscar/ayudas/legislacion_actualizada.php#faq001i. Consultado el 15 de junio de 2022.
- [3] *Inicio - Dereito consolidado do galego - lex.gal*. Disponible en <https://www.lex.gal/portada>. Consultado el 29 de abril de 2022.
- [4] Bani-Salameh, H., & Jeffery, C. (2014). Collaborative and social development environments: a literature review. *International Journal of Computer Applications in Technology*, 49(2), 89-103.
- [5] *Wikipedia, la enciclopedia libre*. Wikipedia. Disponible en <http://es.wikipedia.es>. Consultado el 2 de mayo de 2022.
- [6] Oeberst, A., Halatchliyski, I., Kimmerle, J., & Cress, U. (2014). Knowledge construction in Wikipedia: A systemic-constructivist analysis. *Journal of the Learning Sciences*, 23(2), 149-176.
- [7] Amina, W., & Warraich, N. F. (2022). Use and trustworthiness of Wikipedia information: students' perceptions and reflections. *Digital Library Perspectives*, 38(1), 16-32..
- [8] *Herramienta REM*. Departamento de lenguajes y sistemas informáticos. Universidad de Sevilla. . Disponible en http://www.lsi.us.es/descargas/descarga_programas.php?id=3. Consultado el 6 de junio de 2022.
- [9] *Requisitos (sistemas)*. Artículo de la Wikipedia ([https://es.wikipedia.org/wiki/Requisito_\(sistemas\)](https://es.wikipedia.org/wiki/Requisito_(sistemas))). Consultado el 6 de junio de 2022.
- [10] *Caso de uso*. Artículo de la Wikipedia (https://es.wikipedia.org/wiki/Caso_de_uso). Consultado el 6 de junio de 2022.

- [11] *Backend vs Frontend: Descubre qué es y sus diferencias*. Juan de Assembler Institute. Disponible en <https://assemblerinstitute.com/blog/backend-vs-frontend/>. Consultado el 12 de junio de 2022.
- [12] *React - Una biblioteca de JavaScript para construir interfaces de usuario*. Disponible en <https://es.reactjs.org/>. Consultado el 29 de abril de 2022.
- [13] *Tailwind CSS - Rapidly build modern websites without ever leaving your HTML*. Disponible en <https://tailwindcss.com/>. Consultado el 29 de abril de 2022.
- [14] *¿Qué es CSS? - CSS en español*. Lenguaje CSS (<https://lenguajecss.com/css/introduccion/que-es-css/>). Consultado el 12 de junio de 2022.
- [15] *Node.js*. Disponible en <https://nodejs.org/es/>. Consultado el 3 de junio de 2022.
- [16] *NPM, Node package manager*. Disponible en <https://www.npmjs.com/>. Consultado el 2 de mayo de 2022.
- [17] *React Diff Viewer*. Disponible en <https://www.npmjs.com/package/react-diff-viewer>. Consultado el 9 de junio de 2022.
- [18] *Heroicons*. Disponible en <https://heroicons.com/>. Consultado el 9 de junio de 2022.
- [19] *MUI: The React component library you always wanted*. Disponible en <https://mui.com/>. Consultado el 9 de junio de 2022.
- [20] *Spring Boot*. Disponible en <https://spring.io/projects/spring-boot>. Consultado el 29 de abril del 2022.
- [21] *Gradle Build Tool*. Disponible en <https://gradle.org/>. Consultado el 9 de junio del 2022.
- [22] *MongoDB: La Plataforma De Datos Para Aplicaciones - MongoDB*. Disponible en <https://www.mongodb.com/es>. Consultado el 2 de mayo de 2022.
- [23] *Qué es una base de datos NoSQL y por qué su empresa debería usarla para mejorar el rendimiento y los datos, reducir los costos y aumentar la velocidad de llegada al mercado*. Disponible en <https://www.rackspace.com/es/library/what-is-a-nosql-database#:~:text=NoSQL%20se%20refiere%20a%20una,formatos%20que%20no%20sean%20tablas..>. Consultado el 12 de junio de 2022.

- [24] *API documentation & Design Tools for Teams - Swagger*. Disponible en <https://swagger.io/>. Consultado el 2 de mayo de 2022.
- [25] *Arquitectura REST: Concepto y fundamentos*. GaussWebApp. Disponible en <https://gausswebapp.com/arquitectura-rest.html>. Consultado el 16 de junio de 2022.
- [26] *¿Qué es el navegador de hoja perenne?*. Techopedia. Disponible en <https://es.theastrologypage.com/evergreen-browser>. Consultado el 16 de junio de 2022.
- [27] *Lógica de negocio*. Conecta Software. Disponible en <https://conectasoftware.com/glosario/logica-de-negocio/>. Consultado el 16 de junio de 2022.
- [28] *JSON Web Tokens - jwt.io*. Disponible en <https://jwt.io/>. Consultado el 6 de junio de 2022.
- [29] *Arquitectura orientada a servicios*. Artículo de la Wikipedia (https://es.wikipedia.org/wiki/Arquitectura_orientada_a_servicios). Consultado el 9 de junio de 2022.
- [30] *Introducción a los servlets*. Manual Web. (<https://www.manualweb.net/javaee/introduccion-servlets/>). Consultado el 20 de junio de 2022.
- [31] *Django*. The web framework for perfectionists with deadlines. (<https://www.djangoproject.com/>). Consultado el 20 de junio de 2022.
- [32] *Arquitectura de un SPA*. Desarrollo de aplicaciones web. (https://juanda.gitbooks.io/webapps/content/spa/arquitectura_de_un_spa.html). Consultado el 20 de junio de 2022.
- [33] *Composite*. Refactoring Guru. (<https://refactoring.guru/es/design-patterns/composite>). Consultado el 12 de junio de 2022.
- [34] *Front Controller*. Tutorials Point. (https://www.tutorialspoint.com/design_pattern/front_controller_pattern.htm). Consultado el 20 de junio de 2022.
- [35] *State*. Refactoring Guru. (<https://refactoring.guru/es/design-patterns/state>). Consultado el 20 de junio de 2022.
- [36] *Hooks Pattern*. Patterns. (<https://www.patterns.dev/posts/hooks-pattern/>). Consultado el 20 de junio de 2022.
- [37] *¿Qué es un Mock Up?*. Claudia Bravo. Estudioka. (<https://estudioka.es/que-es-un-mock-up/>). Consultado el 19 de junio de 2022.

- [38] *Prueba unitaria*. Artículo de la Wikipedia (https://es.wikipedia.org/wiki/Prueba_unitaria). Consultado el 7 de junio de 2022.
- [39] *Prueba de integración*. Artículo de la Wikipedia (https://es.wikipedia.org/wiki/Prueba_de_integraci%C3%B3n). Consultado el 7 de junio de 2022.
- [40] *Cómo medir la usabilidad con un SUS*. Cris Busquets. Disponible en <https://www.uifrommars.com/como-medir-usabilidad-que-es-sus/>. Consultado el 7 de junio de 2022.
- [41] *Los 10 principios de usabilidad de Jakob Nielsen*. Beatriz Allas Miguelsanz. Disponible en <https://profile.es/blog/los-10-principios-de-usabilidad-web-de-jakob-nielsen/>. Consultado el 7 de junio de 2022.
- [42] *Información sobre herramientas*. Artículo de la Wikipedia (https://es.wikipedia.org/wiki/Informaci%C3%B3n_sobre_herramientas). Consultado el 7 de junio de 2022.
- [43] *WebSocket*. Artículo de la Wikipedia (<https://es.wikipedia.org/wiki/WebSocket>). Consultado el 8 de junio de 2022.
- [44] *IntelliJ IDEA: el IDE de Java eficaz y ergonómico de JetBrains*. Disponible en <https://www.jetbrains.com/es-es/idea/>. Consultado el 3 de junio de 2022.
- [45] *Java Downloads — Oracle*. Disponible en <https://www.oracle.com/java/technologies/downloads/#jdk18-windows>. Consultado el 5 de junio de 2022.
- [46] *Símbolo del sistema de Windows*. Artículo de la Wikipedia (https://es.wikipedia.org/wiki/S%C3%ADmbolo_del_sistema_de_Windows). Consultado el 24 de mayo de 2022.
- [47] *The MIT License - Open Source Initiative*. Disponible en <https://opensource.org/licenses/MIT>. Consultado el 29 de abril de 2022.