



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior (Jaén)

Trabajo Fin de Grado

SIGADO: SISTEMA DE GESTIÓN Y ANÁLISIS DE DOCUMENTOS

Alumno: Pablo Álvarez Ojeda

Tutores: Prof. D. Manuel Carlos Díaz Galiano
Prof. D. Pilar López Úbeda

Depto: Departamento de Informática

Septiembre, 2022



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Jaén
Departamento de Informática

Don MANUEL CARLOS DÍAZ GALIANO y Doña PILAR LÓPEZ ÚBEDA, tutores del trabajo de fin de grado titulado: SIGADO: SISTEMA DE GESTIÓN Y ANÁLISIS DE DOCUMENTOS, que presenta PABLO ÁLVAREZ OJEDA, autorizan su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, NOVIEMBRE de 2022

FICHA DE TRABAJO DE FIN DE TÍTULO	
Titulación	Grado en Ingeniería Informática
Modalidad	Proyecto de Ingeniería
Especialidad	Tecnologías de la Información
Mención	General
Idioma	Español
TFT en equipo	No
Autor	Pablo Álvarez Ojeda
Fecha de asignación	Octubre de 2021
Descripción corta	El objetivo de este proyecto consiste en aplicar técnicas de Procesamiento del Lenguaje Natural (PLN) para el análisis de corpus documentales, que permita obtener las distintas características de estos corpus y buscar la información existente en ellos de forma precisa y automática. Con este TFG se pretende crear una plataforma multiusuario que permita la gestión de usuarios y corpus, además de realizar analítica sobre el contenido de dichos corpus.

Tabla 1: Ficha del Trabajo Fin de Grado

NORMAS APLICADAS EN ESTE DOCUMENTO	
LOCALES	
TFT-UJA:2017	Normativa de Trabajos Fin de Grado, Fin de Máster y otros Trabajos Fin de Título de la Universidad de Jaén (Normativa marco UJA aprobada en Consejo de Gobierno).
TFT-EPSJ:2017	Normativa sobre Trabajos Fin de Grado y Fin de Máster en la Escuela Politécnica Superior de Jaén (Normativa EPSJ aprobada en Junta de Escuela).
TFT-EPSJ	Criterios de evaluación y normas de estilo para TFG y TFM de la Escuela Politécnica Superior de Jaén.
NACIONALES E INTERNACIONALES	
ISO 2145:1978	Documentación - Numeración de divisiones y subdivisiones en documentos escritos.
UNE 50132:1994	Traducción de la ISO 2145
IEEE	Estilo de referencias y citas de IEEE (Institute of Electrical and Electronics Engineers).

Tabla 2: Normas aplicadas en este documento

NORMAS UTILIZADAS COMO BASE O REFERENCIA	
NACIONALES	
UNE 157001:2014	Criterios generales para la elaboración formal de los documentos que constituyen un proyecto técnico.
UNE 157801:2007	Criterios generales para la elaboración de proyectos de sistemas de información.
<i>Estas normas se han utilizado como base o referencia para la inclusión de algunos contenidos y definiciones sobre elaboración de proyectos, entendiendo como proyecto la documentación consensuada entre una empresa y un cliente, que da lugar al perfeccionamiento de un contrato para la elaboración de una obra o la prestación de un servicio. Por consiguiente, no debe esperarse la aplicación de estas normas en cuanto a la completitud de los contenidos ni a la organización de los mismos.</i>	

Tabla 3: Normas utilizadas como base o referencia

AGRADECIMIENTOS

A mis tutores Manuel Carlos y Pilar, por darme la oportunidad en su día de crear este proyecto, por su paciencia, cariño, calidad humana y docente e implicación y supervisión a lo largo del trabajo, sin ellos nada de esto hubiera sido posible.

A mi familia y padres por darme la oportunidad de estudiar un grado universitario, apoyarme en todo momento y ser mi inspiración de trabajo duro y esfuerzo.

A mi grupo de amigos, especialmente a los más cercanos, Alberto, Juan Carlos y Juan Antonio por nunca dejar de creer en mí, estar ahí para echarme una mano, levantarme en los momentos difíciles y darme esos momentos de respiro tan necesarios a lo largo de este trayecto.

A los compañeros que conocí durante la carrera y a todos en los que en algún momento me ayudaron a llegar a esta meta, en especial a los de Send Fútbol a los que considero amigos y que son de lo mejor que me llevo.

A todos y cada uno de los profesores de la universidad por los que he tenido el placer de recibir clase, por su profesionalidad y enseñanzas a lo largo de estos años.

Y por último a mi abuelo, por ser mi inspiración para dedicarme a este ámbito.

Índice

Índice de tablas	11
Índice de figuras	12
1. Introducción	15
1.1. Motivación	16
1.2. Objetivos	16
2. Planificación del proyecto	18
2.1. Metodología de desarrollo	18
2.2. Diagrama de planificación	19
3. Presupuesto del proyecto	21
3.1. Presupuesto detallado	21
3.1.1. Recursos humanos	21
3.1.2. Licencias software	22
3.1.3. Hardware	22
3.1.4. Costes indirectos	22
3.1.5. Coste total	23
3.1.6. Alojamiento del sitio en servicio de computación en la nube	23
4. Software empleado	24
4.1. Herramienta de prototipado: Diagrams.net	24
4.2. Análisis de bibliotecas de PLN	25
4.2.1. Python NLTK	25
4.2.2. Python spaCy	27
4.2.3. Java Apache OpenNLP	27
4.2.4. C++ MITIE	28
4.3. Análisis de bases de datos	29
4.3.1. SQLite	29

4.3.2.	MySQL	30
4.3.3.	MariaDB	32
4.4.	Análisis de frameworks o plataformas	33
4.4.1.	Django	33
4.4.2.	Flask	35
4.4.3.	CodeIgniter	37
4.4.4.	WordPress	38
4.4.5.	Comparativa de frameworks web	39
4.5.	Selección de alternativas	41
4.5.1.	Biblioteca PLN	41
4.5.2.	Base de datos	41
4.5.3.	Framework	42
5.	Desarrollo inicial	44
5.1.	Iteración 1: Inicialización del proyecto	44
5.1.1.	Análisis de software para desarrollo de la plataforma	44
5.1.2.	Realización de storyboard, sección de gestión de usuarios y colecciones	45
6.	Desarrollo del proyecto	57
6.1.	Iteración 2: desarrollo del login y registro en plataforma	57
6.1.1.	Login	57
6.1.2.	Registro	59
6.2.	Iteración 3: implementación de funcionalidades de seguridad	59
6.3.	Iteración 4: implementación de sección para gestión de usuarios	60
6.3.1.	Tabla de registros de usuario	60
6.3.2.	Pestaña de edición de usuario	61
6.3.3.	Pestaña de eliminación de usuario	62
6.4.	Iteración 5: implementación de sección para gestión de colecciones	62
6.4.1.	Pestaña de creación de colección y tabla de registros	62
6.4.2.	Pestaña de edición de colección	63
6.4.3.	Pestaña de eliminación de colección	64
6.4.4.	Implementación de subida de documentos a la plataforma y asignación a colecciones	64
6.4.5.	Reestructuración del apartado para mejora de experiencia de usuario . .	69
6.4.6.	Mejoras en el apartado	69
6.5.	Iteración 6: implementación de vista de perfil de usuario	71
6.5.1.	Controlador para vista de perfil de usuario	71

6.5.2. Incorporación de funcionalidad para cambio de contraseña (usuario y administrador)	71
6.6. Iteración 7: implementación de herramientas de PLN para corpus textuales . .	72
6.6.1. Realización de storyboard, sección de análisis de texto	72
6.6.2. Realización de funcionalidad 'Wordlist'	74
6.6.3. Realización de funcionalidad 'N-Grams'	76
6.6.4. Reconstrucción de sistema de análisis para comodines	77
6.6.5. Realización de funcionalidad 'Concordance'	78
6.6.6. Realización de funcionalidad 'Concordance Plot'	79
6.6.7. Mejoras y corrección de errores	80
7. Librería de analítica de corpus textuales	81
7.1. Consideraciones previas	81
7.2. Técnicas PLN aplicables al proyecto	82
7.2.1. WordList	83
7.2.2. N-Grams	83
7.2.3. Concordance	84
7.2.4. Concordance Plot	84
8. Conclusiones	86
8.1. Conclusión final	86
8.2. Mejoras futuras	88
9. Bibliografía	89
A. Manual de instalación	94
B. Manual de uso	96
B.1. Consideraciones previas	96
B.2. Creación de un usuario en la plataforma	97
B.3. Iniciar sesión en la plataforma	100
B.4. Cerrar sesión en la plataforma	101
B.5. Gestión de usuarios	101
B.5.1. Mostrar registros	102
B.5.2. Ordenar registros	103
B.5.3. Buscar registros	103
B.5.4. Cambiar contraseña de usuario	103
B.5.5. Editar usuarios	104
B.5.6. Eliminar usuarios	105

B.6. Mi perfil	105
B.6.1. Cambiar contraseña propia	106
B.7. Gestión de colecciones	107
B.7.1. Crear colecciones	108
B.7.2. Editar colecciones	109
B.7.3. Eliminar colecciones	110
B.8. Gestión de documentos	111
B.8.1. Subir un documento	112
B.8.2. Subir un fichero comprimido ZIP	114
B.8.3. Editar documentos	116
B.8.4. Eliminar un documento	117
B.8.5. Eliminar varios documentos a la vez	118
B.8.6. Asignar documentos a colecciones	120
B.9. Uso de herramientas	121
B.9.1. Wordlist	121
B.9.2. N-Grams	122
B.9.3. Concordance	123
B.9.4. Concordance Plot	124

Índice de tablas

1.	Ficha del Trabajo Fin de Grado	4
2.	Normas aplicadas en este documento	4
3.	Normas utilizadas como base o referencia	5
3.1.	Costes en recursos humanos	21
3.2.	Costes en licencias software	22
3.3.	Costes en hardware	22
3.4.	Coste total del proyecto	23
4.1.	Tabla comparativa de frameworks	40

Índice de figuras

5.1. Escena 1: inicio de sesión	46
5.2. Escena 2: registro	46
5.3. Escena 3: colecciones	47
5.4. Escena 4: crear colección	47
5.5. Escena 5: editar colección	48
5.6. Escena 6: herramientas	48
5.7. Escena 7: eliminar colección	49
5.8. Escena 8: vista detallada de colección	49
5.9. Escena 9: crear documento	50
5.10. Escena 10: visualizar documento	50
5.11. Escena 11: editar documento	51
5.12. Escena 12: eliminar documento	51
5.13. Escena 13: etiquetas de los documentos	52
5.14. Escena 14: vista de usuarios	52
5.15. Escena 15: crear usuario	53
5.16. Escena 16: editar usuario	53
5.17. Escena 17: eliminar usuario	54
5.18. Escena 18: visualizar equipos	54
5.19. Escena 19: menú de perfil	55
5.20. Escena 20: vista de perfil de usuario	55
5.21. Escena 21: preferencias	56
5.22. Escena 22: cerrar sesión	56
6.1. Escena 23: herramienta 'Wordlist'	73
6.2. Escena 24: herramienta 'N-Grams'	73
6.3. Escena 25: herramienta 'Concordance'	74
6.4. Escena 26: herramienta 'Concordance Plot'	74
B.1. Símbolo de ayuda	96
B.2. Crear cuenta en CEALMTools	97

B.3. Registro en CEALMTools	98
B.4. Vuelta a inicio de sesión en CEALMTools	99
B.5. Inicio de sesión en CEALMTools	100
B.6. Espacio de trabajo CEALMTools	100
B.7. Enlace a cierre de sesión	101
B.8. Vista de cierre de sesión	101
B.9. Enlace a sección de gestión de usuarios	101
B.10. Sección de gestión de usuarios	102
B.11. Mostrar registros	102
B.12. Pestañas de registros	102
B.13. Ordenar registros	103
B.14. Buscar registros	103
B.15. Enlace a cambio de contraseña de usuario	104
B.16. Cambio de contraseña a usuario	104
B.17. Enlace a edición de usuario	104
B.18. Vista de edición de usuario	105
B.19. Confirmación de eliminación de usuarios	105
B.20. Enlace a mi perfil	106
B.21. Mi perfil	106
B.22. Enlace a cambio de contraseña propia	107
B.23. Vista de cambio de contraseña propia	107
B.24. Enlace a colecciones	108
B.25. Vista de colecciones	108
B.26. Enlace a añadir colección	108
B.27. Vista de creación de colección	109
B.28. Enlace a edición de colección	109
B.29. Vista de edición de colección	110
B.30. Enlace a eliminación de colección	110
B.31. Vista de eliminación de colección	110
B.32. Denegada eliminación de colección	111
B.33. Enlace a vista de documentos de la colección	111
B.34. Vista de documentos de la colección	112
B.35. Enlace a subida de documentos a la colección	112
B.36. Vista de error al seleccionar codificación del documento	113
B.37. Vista de subida de documentos	114
B.38. Vista de subida de fichero ZIP	115
B.39. Vista de resumen de importación de documentos de un ZIP	115
B.40. Enlace a edición de documento	116

B.41. Vista de edición de documento	117
B.42. Enlace a eliminación individual de documento	118
B.43. Vista de desvinculación de documento	118
B.44. Vista de eliminación definitiva de documento	118
B.45. Selección de documentos para eliminación múltiple y enlace a vista	119
B.46. Vista de confirmación de selecciones para eliminación	119
B.47. Enlace a asignación de documentos a colección	120
B.48. Vista de selección y confirmación de vinculación de documentos	120
B.49. Enlace a análisis de documentos	121
B.50. Vista de herramienta Wordlist	122
B.51. Vista de herramienta N-Grams	123
B.52. Vista de herramienta Concordance	123
B.53. Vista de herramienta Concordance Plot 1	124
B.54. Vista de herramienta Concordance Plot 2	125

Capítulo 1

Introducción

En este capítulo, introduciremos el proyecto y hablaremos sobre la motivación para la realización del mismo, así como los objetivos a alcanzar.

Hoy en día el procesamiento del lenguaje natural está muy presente en la sociedad, desde cualquier asistente en un dispositivo móvil (Siri, Google assistant...), pasando por uno doméstico como Alexia hasta Cortana en un computador personal o cualquier otro agente que capta nuestra voz o nuestra escritura, procesa la información contenida y actúa en función a la misma. El PLN busca construir máquinas o programas que tengan la capacidad de entender y extraer el significado de la comunicación escrita y verbal y es aplicable a diversos ámbitos como el Big Data, en el que se analizan ingentes volúmenes de datos, como por ejemplo correos electrónicos, documentos, ensayos clínicos, notas de médicos... Como podemos ver, desestructurados en numerosos casos, el procesamiento del lenguaje natural presenta una oportunidad para aprovechar lo que contienen los almacenes de datos para revelar patrones, conexiones y tendencias en los datos. También es empleado en el ámbito sanitario y biomédico, en el que debido al alto contenido de expresiones específicas como unidades de medida, nombres de sustancias o códigos alfanuméricos, plantea un número no desdeñable de dificultades a la hora de procesarlo automáticamente. Los análisis de Big Data arrojan que el 80 % de la información médica está desestructurada y, por lo tanto, se desaprovecha en gran medida, ya que la minería y la extracción de estos datos suponen un reto y una gran cantidad de recursos. Sin la tecnología de PLN, esos datos no están en un formato utilizable para que los algoritmos informáticos modernos puedan extraerlos.

En el proyecto que nos ocupa, emplearemos el procesamiento del lenguaje natural para analizar texto proveniente del Centro de Estudios Avanzados en Lenguas Modernas de la Universidad de Jaén de tal manera que los profesores e investigadores del CEALM puedan sacar conclusiones sobre la información arrojada por el examen del texto. Para hacer posible esta idea de forma práctica e intuitiva, nos disponemos a presentar el desarrollo de una plataforma para la gestión de corpus textuales y usuarios en la misma, que incorpora una librería de analítica de

corpus textuales con las distintas funcionalidades necesarias para el estudio de textos escritos. A lo largo de la documentación del proyecto trataremos diversos aspectos sobre el desarrollo del mismo como los lenguajes de programación empleados, librerías de procesamiento de lenguaje natural utilizadas o entornos de desarrollo escogidos.

1.1. Motivación

Desde el Centro de Estudios Avanzados en Lenguas Modernas de la Universidad de Jaén, más concretamente a través de María Belén Díez Bedmar, profesora del departamento de filología inglesa de la UJA, se planteó la posibilidad de realizar una plataforma de gestión de corpus de texto y usuarios con distintos roles dentro del sitio en la que se pudieran almacenar documentos agrupados en colecciones para su posterior análisis, sometiendo a examen las redacciones realizadas por los alumnos y extrayendo información valiosa de los corpus de tal manera que se pudiera agilizar el trabajo correspondiente a la enseñanza del alumnado. La idea consistía en realizar unas funcionalidades similares a las del software AntConc, freeware de análisis de corpus textuales, integrado en un sistema que incorporase herramientas útiles para el cometido de los investigadores de las que hablaremos más adelante de forma detallada en la documentación.

Se realizó un estudio sobre entornos de desarrollo útiles para la realización de la plataforma, lenguajes de programación y bibliotecas empleables para la creación de funcionalidades de análisis de texto, así como la opción de sistema de gestión de base de datos relacional que mejor se ajustase a la idea que se quería acometer.

1.2. Objetivos

El **objetivo principal** del proyecto es la aplicación de técnicas de Procesamiento del Lenguaje Natural para el análisis de corpus documentales, que permita obtener las distintas características de estos corpus y buscar la información existente en ellos de forma precisa y automática, además de la creación de una plataforma multiusuario que permita la gestión de usuarios y corpus, así como la realización de analítica sobre el contenido de los mencionados corpus.

Los **objetivos específicos** del trabajo son los siguientes:

- Realizar un estudio de las diferentes técnicas de PLN aplicables a este problema.
- Diseñar y desarrollar una librería de analítica de corpus textuales con las distintas funcionalidades.

- Diseñar y crear una plataforma web que permita realizar la gestión de usuarios y corpus.
- Incluir en la plataforma web acceso a la visualización y búsqueda de información en los corpus incorporados.
- Redactar la presente memoria que recoge todo el trabajo realizado así como los manuales de instalación y usuario.

Capítulo 2

Planificación del proyecto

En este capítulo se tratará la organización temporal del proyecto, así como las iteraciones del mismo y el cronograma en forma de diagrama de Gantt correspondiente.

2.1. Metodología de desarrollo

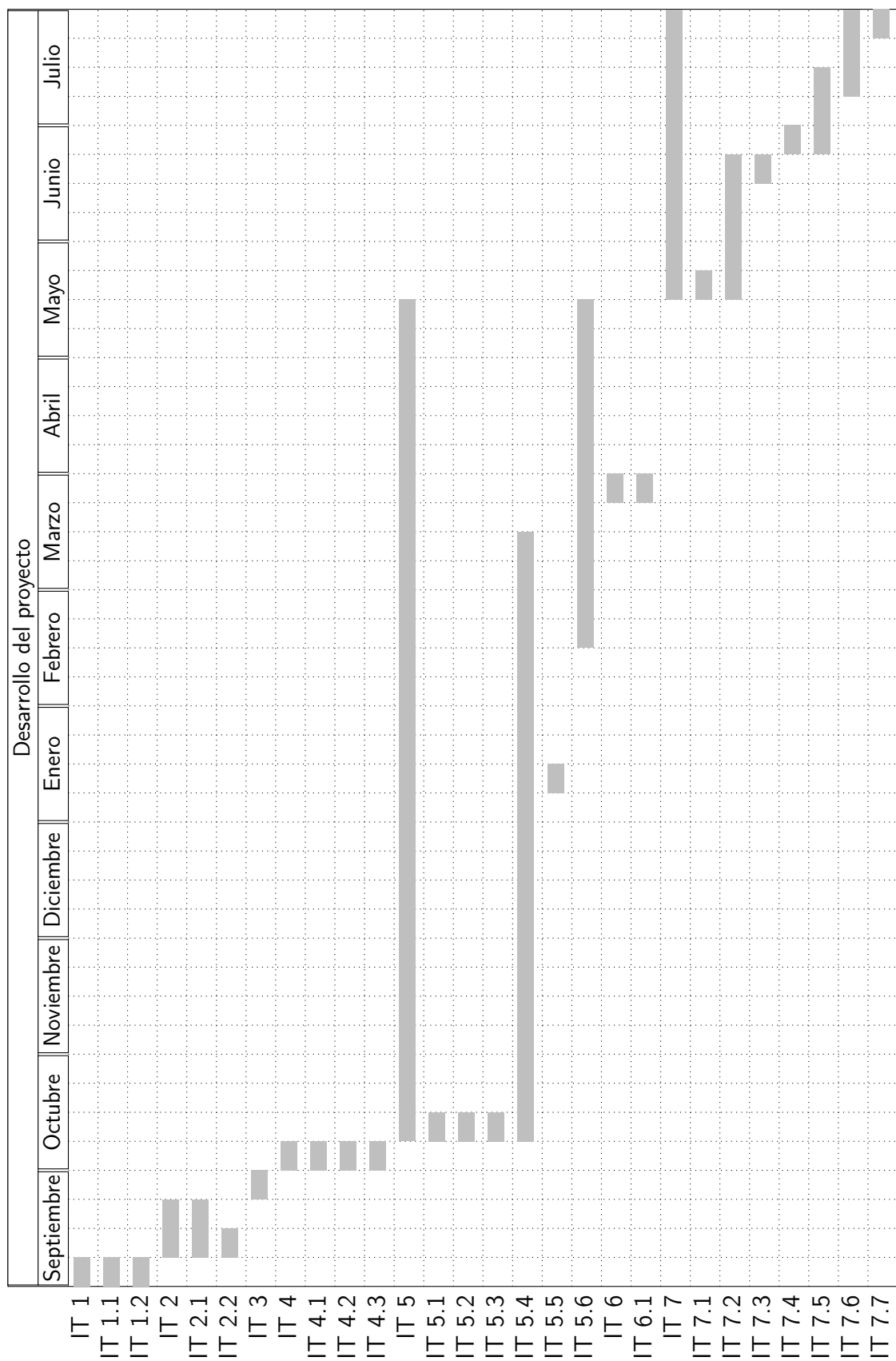
Se escogió para el desarrollo del proyecto una metodología de tipo incremental, incluyendo en cada iteración del mismo una funcionalidad nueva de la plataforma. Si existían dificultades para implementar una funcionalidad, el desarrollo se centraba en otra parte y posteriormente se volvía a la pendiente. La elaboración del trabajo estuvo comprendida entre septiembre de 2021 y julio de 2022. Las iteraciones del proyecto son las siguientes:

1. Inicialización del proyecto (1 semana)
 - 1.1 Análisis de software para desarrollo de la plataforma (1 semana)
 - 1.2 Realización de storyboard (sección de gestión de usuarios y colecciones) (1 semana)
2. Desarrollo del login y registro de la plataforma (2 semanas)
 - 2.1 Login (2 semanas)
 - 2.2 Registro (1 semana)
3. Implementación de funcionalidades de seguridad (1 semana)
4. Implementación de sección para gestión de usuarios (1 semana)
 - 4.1 Tabla de registros de usuario (1 semana)
 - 4.2 Pestaña de edición de usuario (1 semana)
 - 4.3 Pestaña de eliminación de usuario (1 semana)

5. Implementación de sección para gestión de colecciones (29 semanas)
 - 5.1 Pestaña de creación de colección y tabla de registros (1 semana)
 - 5.2 Pestaña de edición de colección (1 semana)
 - 5.3 Pestaña de eliminación de colección (1 semana)
 - 5.4 Implementación de subida de documentos a la plataforma y asignación a colecciones (21 semanas)
 - 5.5 Reestructuración del apartado para mejora de experiencia de usuario (1 semana)
 - 5.6 Mejoras en el apartado (12 semanas)
6. Implementación de vista de perfil de usuario (1 semana)
 - 6.1 Incorporación de funcionalidad para cambio de contraseña (usuario y administrador) (1 semana)
7. Implementación de herramientas de PLN para corpus textuales (10 semanas)
 - 7.1 Realización de storyboard (secciones pertenecientes al análisis de texto) (1 semana)
 - 7.2 Realización de funcionalidad 'WordList' (5 semanas)
 - 7.3 Realización de funcionalidad 'N-Grams' (1 semana)
 - 7.4 Reconstrucción de sistema de análisis para comodines (1 semana)
 - 7.5 Realización de funcionalidad 'Concordance' (3 semanas)
 - 7.6 Realización de funcionalidad 'Concordance Plot' (3 semanas)
 - 7.7 Mejoras y corrección de errores (1 semana)

2.2. Diagrama de planificación

Se presenta el diagrama de Gantt de planificación del proyecto, estructurado por iteraciones y dividido en semanas de trabajo.



Capítulo 3

Presupuesto del proyecto

En esta sección de la memoria se expondrá el presupuesto destinado al proyecto, tanto para los recursos humanos empleados, como para las licencias software y hardware informático necesario.

3.1. Presupuesto detallado

A continuación se presenta una descripción detallada del presupuesto realizado al proyecto de ingeniería realizado:

3.1.1. Recursos humanos

Este proyecto consta de dos integrantes, el desarrollador y el supervisor o jefe de proyecto del mismo. Los empleados trabajan 5 días a la semana (de lunes a viernes), 5 horas al día, desde las 9 de la mañana a las 2 de la tarde. Se especifican las siguientes cifras salariales en la tabla 3.1 basadas en los datos arrojados por el Boletín Oficial del Estado del XVII Convenio colectivo estatal de empresas de consultoría y estudios de mercado y de la opinión pública, resolución publicada el 22 de febrero de 2018¹:

Rol	Salario anual	Salario mensual	Meses de desarrollo	Total
Jefe de proyecto	15.442,56 €	1.286,88 €	10	12.868,8 €
Desarrollador junior	13.827,66 €	1.152,305 €	10	11.523,05 €

Tabla 3.1: Costes en recursos humanos

TOTAL: 24.391,85 €

¹Disponible en: <https://www.boe.es/boe/dias/2018/03/06/pdfs/B0E-A-2018-3156.pdf>

3.1.2. Licencias software

Sólo se requieren licencias para los computadores implicados en el proyecto, en este caso dos. Se procede al desglose de costes (por unidad) en la tabla 3.2 prorrateado por la vida útil del sistema operativo, considerada con una duración de 10 años:

Software	Coste
Windows 10 Professional (x2)	10,00 €
Pycharm Community Edition (x2)	0,00 €
MySQL Workbench (x2)	0,00 €

Tabla 3.2: Costes en licencias software

TOTAL: 20,00 €

3.1.3. Hardware

Se requieren dos equipos informáticos para el desarrollo de la plataforma. Cada uno de ellos con sus periféricos correspondientes (monitores, teclado y ratón). A continuación en la tabla 3.3 se presenta una tabla con los recursos empleados y sus costes por unidad asociados, como ya se indicó en el apartado de licencias software, prorrateados por la vida útil del equipo, estimada de 5 a 6 años aproximadamente.

Artículo	Coste
Caja ordenadores: Owlotech Start Case USB 3.0 500W Torre ATX (x2)	30,00 €
Procesador: Intel Core i5-10400 2,9GHz (x2)	180,99 €
Placa base: Gigabyte H510M S2H V2 (x2)	72,99 €
Disco duro: 500GB SSD NVMe M.2 2280 (x2)	92,99 €
Memoria RAM: 16GB 2 pastillas x 8GB DDR4 2666MHz CL16 (x2)	113,98 €
Fuente de alimentación: Ewent EW3909 500W (x2)	17,85 €
Monitor: LG 24MP400-B, 75 Hz, Full HD IPS, AMD Freesync (x4)	119,00 €
Teclado y ratón Logitech MK120 (x2)	17,99 €

Tabla 3.3: Costes en hardware

TOTAL: 1529,58 €

3.1.4. Costes indirectos

Son necesarios fondos para posibles imprevistos y gastos indirectos durante el desarrollo que no afecten al presupuesto pactado y para el mantenimiento de los propios equipos,

instalaciones de trabajo, desplazamientos, servidores... El capital destinado a mantenimiento será el 10 % de los gastos anteriores.

TOTAL: 2.594.143 €

3.1.5. Coste total

Sección	Coste
Recursos humanos	24.391,85 €
Licencias software	20,00 €
Hardware	1529,58 €
Mantenimiento	2.594,143 €

Tabla 3.4: Coste total del proyecto

TOTAL: 28.535,573 €

3.1.6. Alojamiento del sitio en servicio de computación en la nube

En caso de que el sitio quiera ser alojado en una plataforma de computación en la nube, se han valorado dos posibles opciones:

- **AWS:** se realizó un presupuesto² de Amazon Web Service para alojar la plataforma, el precio resultante tras los cálculos es de 109,76 USD mensuales, que convertidos a euros dan una tarifa de 106,01€. En esta estimación se incluyen una máquina Amazon Simple Storage Service (S3) y una base de datos Amazon Aurora compatible con MySQL.
- **Google Cloud:** otra opción interesante para alojamiento en la nube proviene de Google Cloud. En este caso el presupuesto³ está formado por una máquina Compute Engine y una base de datos Cloud SQL para MySQL además de un disco duro persistente, cuyo coste conjunto es de 184.84€ mensuales.

Si se desean ver las especificaciones detalladas de las máquinas, es posible revisarlas en los enlaces en el pie de la actual página.

²AWS: <https://calculator.aws/#/estimate?id=cf538992c1c4cd41d15228c2abab4f2a53b29fdb>

³Google Cloud: <https://cloud.google.com/products/calculator/#id=fd9ff74e-8018-4440-bdcd-ded1ea127afb>

Capítulo 4

Software empleado

Como ya adelantamos en el capítulo 2, para el software utilizado en el desarrollo del sistema se realizó en primera instancia un estudio sobre los posibles entornos de desarrollo y lenguajes de programación utilizables para la construcción del sitio web, además de los disponibles para la elaboración de la librería de analítica de corpus textuales y las bibliotecas que ofrecen cada uno de ellos en este ámbito, así como las bases de datos empleables para el almacenamiento de modelos del sistema. A lo largo de este capítulo plasmaremos dicho examen de recursos para poder entender el porqué de nuestras elecciones a la hora de la confección del sistema.

4.1. Herramienta de prototipado: Diagrams.net

En este caso no se realizó un análisis sobre la herramienta de prototipado debido al uso que se le había dado a Diagrams.net¹ durante el grado por lo que se poseía gran soltura y experiencia con la misma y se creyó que era lo más conveniente para realizar el esquema de una forma rápida y eficiente.

Diagrams.net es un software para realización de diagramas de flujo, prototipos y esquemas en línea. Permite crear y distribuir diagramas a través de operaciones básicas como agarrar y soltar. Los profesionales emplean esta herramienta para crear capas de conversión con URLs personalizadas y alinear la información de diferentes formas. Además el programa permite generar y exportar diagramas a ficheros PDF de manera automática. Algunas de sus **características técnicas** son:

- Colaboración en tiempo real.
- Diferentes formas para puntos de conexión entre elementos.
- Diferentes estilos de objetos.

¹Disponible en: <http://diagrams.net>

- Múltiples paquetes de elementos gráficos para prototipado.

Ventajas e inconvenientes

Las **ventajas** que ofrece Diagrams.net son:

- Posee una base de datos rica en casos de uso, modelaje, diagramas de red, *wireframes*, etc.
- Interfaz de usuario simple.
- Útil para trabajo en equipo.
- Plantillas de diagrama personalizables.
- Alineamiento simple: gracias al sistema de plantillas que incorpora.
- Accesible también de forma offline.
- Gratuidad.
- Integración con Google Drive.

Las **desventajas** que arroja Diagrams.net son:

- Limitaciones de uso: a la hora de trabajar con texto y formas.
- No es apropiado para diagramas extremadamente complejos.

4.2. Análisis de bibliotecas de PLN

Se consideraron inicialmente 3 lenguajes de programación que poseen bibliotecas de Procesamiento de Lenguaje Natural: Python, Java y C++. Dado que en el grado se imparten las bibliotecas de Python para este cometido y la gran mayoría de proyectos PLN se realizan en el mencionado lenguaje, existía una mayor inclinación hacia ellas, sin embargo quisimos explorar las demás opciones en cualquier caso.

4.2.1. Python NLTK

NLTK es una de las bibliotecas de PLN más completas, conocidas y usadas en educación e investigación, permitiendo numerosos avances en análisis de texto. Asimismo es una herramienta sumamente potente con una extensa variedad de corpus textuales y modelos previamente entrenados, además de librerías de procesamiento textual y otras propiedades que

expondremos a continuación, permitiendo el análisis textual de forma rápida y sencilla. Se trata de una biblioteca excelente que proporciona las funcionalidades requeridas para cualquier tipo de tarea PLN con Python.

Algunas de sus **características técnicas** son:

- Part of speech (POS).
- Reconocimiento de entidades (NER).
- Clasificación.
- Análisis de sentimientos.
- Detección de sentencias.
- *Tokenización*.
- Lematización.
- Reducción de palabras a raíces.
- Análisis sintáctico.
- Fragmentación.
- Etiquetado POS.

Ventajas e inconvenientes

Las **ventajas** que ofrece NLTK son:

- Es la biblioteca más famosa de PLN por lo que es sencillo encontrar información y soporte en la red sobre la misma.
- Posee numerosas extensiones.
- Versatilidad.
- Permite la realización de la mayoría de herramientas que quieren incorporarse a la librería de análisis textual.

Las **desventajas** que arroja NLTK son:

- Curva de aprendizaje elevada: sin embargo, al ser una biblioteca que se estudia durante el grado, se posee familiarización y agilidad con la misma.
- No analiza la estructura semántica del texto.
- No incorpora modelos de redes neuronales.

4.2.2. Python spaCy

spaCy es una biblioteca avanzada de código abierto de PLN diseñada desde la eficiencia y la rapidez, que opera con entornos de trabajo *deep learning* como TensorFlow o PyTorch y permite el desarrollo de sistemas entendedores de lenguaje natural y aplicaciones que necesitan procesar y entender enormes volúmenes de texto. Posee modelos estadísticos entrenados previamente y vectores de palabras. Está especialmente dirigido hacia análisis morfosintáctico. Sus **características técnicas** son:

- Part of speech (POS).
- Reconocimiento de entidades (NER).
- Clasificación.
- Análisis de sentimientos.
- *Tokenización (para más de 50 idiomas).*
- Análisis de dependencias.

Ventajas e inconvenientes

Las **ventajas** que ofrece spaCy son:

- Rapidez.
- Facilidad de aprendizaje y uso.
- Usa redes neuronales para el entrenamiento de modelos.

Las **desventajas** que arroja spaCy son:

- Poca flexibilidad.
- Tan solo soporta 7 lenguajes de programación.

4.2.3. Java Apache OpenNLP

Apache OpenNLP es una biblioteca Java basada en *machine learning* de código abierto empleada para PLN.

Sus **características técnicas** más destacadas son:

- Reconocimiento de entidades (NER).

- Etiquetado POS.
- Reconocimiento del habla.
- *Tokenización*.
- Fragmentación.
- Análisis sintáctico.
- Traducción.
- Búsqueda.
- Agrupación de información.

Ventajas e inconvenientes

Las **ventajas** que ofrece Apache OpenNLP son:

- Servicios avanzados de procesamiento de texto.
- Soporta las tareas más comunes de PLN.

Las **desventajas** que arroja Apache OpenNLP son:

- Poca información y soporte en la red sobre OpenNLP.
- No es una biblioteca Python, lo que podría dificultar su desarrollo, aprendizaje e integración en el proyecto.

4.2.4. C++ MITIE

MITIE (MIT Information Extraction) es un conjunto de funcionalidades escrito en C++ que contiene algoritmos *machine learning* y ofrece lo último en herramientas de extracción de información como las de entidades con nombre, las de detección de relaciones y relaciones binarias o las de entrenamiento de extractores personalizados para la elaboración de software C++ complejo que resuelva problemas del mundo real. Sus **características técnicas** más interesantes son:

- Uso de incrustaciones distribuidas de palabras y máquinas de soporte vectorial.
- Uso de modelos previamente entrenados empleando una gran variedad de recursos lingüísticos que ofrecen niveles de soporte variables para alemán, inglés y español.
- Reconocimiento de entidades (NER).
- Se compila como una librería compartida.

Ventajas e inconvenientes

Las **ventajas** que ofrece MITIE son:

- Es posible usar MITIE desde un programa de Python 2.7, uno de R, C, C++ o Java.

Las **desventajas** que arroja MITIE son:

- Compatibilidad con una cantidad pequeña de lenguajes de programación.
- Poca información y soporte sobre MITIE en la red.

4.3. Análisis de bases de datos

Nos decantamos por 3 opciones: **SQLite**, **MariaBD** y **MySQL**. A continuación realizaremos un estudio de los 3 sistemas y en la sección de elecciones expondremos la alternativa escogida con su respectiva justificación.

4.3.1. SQLite

SQLite es una biblioteca software que ofrece una base de datos SQL incrustada. Es un motor de bases de datos relacionales que está diseñado para proporcionar una manera de que las aplicaciones software manejen datos por si solas. Es también el motor de bases de datos más desplegado en el mundo a día de hoy. Las **características técnicas** más importantes de SQLite son:

- Las transacciones son atómicas, consistentes, aisladas y duraderas.
- Sin configuración.
- Soporta bases de datos de terabytes de datos de tamaño.
- Escrito en ANSI-C.
- Independiente puesto que no posee dependencias externas.
- Multiplataforma.
- Línea de comandos propia.

Ventajas e inconvenientes

Las **ventajas** que ofrece SQLite son:

- Totalmente gratuito: es código abierto por lo que no se requiere licencia.
- Sin servidor: no se requiere un proceso servidor o sistema separados para operar.
- Flexibilidad: facilita el trabajo en múltiples bases de datos en la misma sesión al mismo tiempo.
- Ligereza: simple, no está sobrecargado por funcionalidades irrelevantes, no requiere configuración ni mucha gestión de la base de datos y demanda los recursos justos y necesarios.
- Facilidad y eficiencia para almacenar información
- Longitud variable de columnas.
- Disponibilidad en UNIX y Windows.

Las **desventajas** que arroja SQLite son:

- No ofrece acceso desde la red: el que SQLite tenga el beneficio de poseer un modelo sin servidor, también arroja una clara desventaja y es que la base de datos queda confinada exclusivamente en la máquina en la que se almacenó, es decir, no permite trabajo remoto desde otra máquina.
- No es adecuado para aplicaciones a gran escala.
- No posee gestión de usuarios.

4.3.2. MySQL

MySQL es un sistema multiplataforma de código abierto de gestión de bases de datos basado en consultas SQL. Es posiblemente el sistema de gestión de bases de datos relacionales más popular del mundo y es usado por importantes compañías como Facebook, YouTube, Uber o Google. Aunque suele ser usado con PHP, puede ser empleado con cualquier lenguaje de programación y puede ejecutarse en plataformas como Linux, Windows, Unix y un esquema de información para definir y gestionar los metadatos. Para más inri, es posible instalarlo en local o en un servidor. Las **características técnicas** más importantes de MySQL son:

- Arquitectura cliente-servidor: el sistema engloba la base de datos primaria MySQL Server y múltiples clientes (programas de aplicación). MySQL Server crea la base de datos y define las relaciones entre tablas. Los clientes se comunican con el servidor para solicitar y manipular datos con las consultas SQL específicas. Las tareas pueden ser la consulta de datos, adición y salvado de datos, etc. Finalmente la aplicación servidor envía la información requerida al cliente. Además los clientes y los servidores pueden estar en el mismo ordenador o en diferentes máquinas.
- Comandos y consultas SQL: Es posible usar vistas, disparadores y procedimientos almacenados; elementos SQL vitales para administrar una base de datos de gran escala.
- Replicación: es posible copiar los contenidos de una base de datos MySQL en otros ordenadores. De esta manera, es factible proteger los datos contra fallos del sistema y hacer que mejore la velocidad de consultas a la base de datos.
- Transacciones: esta característica simplifica todos los procesos. Las transacciones consisten en la ejecución de varias operaciones como un bloque único. Una condición crucial es que todas las operaciones puedan ser ejecutadas satisfactoriamente en la transacción, en caso contrario ninguna será realizada. Por lo tanto, las transacciones permiten a los programadores, por ejemplo, interrumpir o revocar comandos que ya están en ejecución.
- Restricciones de claves foráneas: en las bases de datos MySQL, las restricciones de claves foráneas son soportadas por completo por tablas InnoDB, asegurando que todas las referencias entre tablas son válidas y precisas.
- Diccionario de datos: MySQL incluye incontables objetos e ingentes volúmenes de información sobre los mismos, motivo por el cual se introdujo el diccionario de datos dedicado, almacenando toda la información sobre dichos objetos. Es un formato seguro contra bloqueos, centralizado y más optimizado que el almacenamiento de metadatos basado en ficheros. También es mucho más conveniente para acceder y trabajar.

Ventajas e inconvenientes

MySQL presenta numerosas **ventajas** y son las siguientes:

- Facilidad de uso.
- Gran rendimiento y potencia para sitios con tráfico elevado.
- Escalabilidad.
- Compatibilidad y flexibilidad.

- Seguridad total de los datos.
- Eficiencia en memoria.
- Capacidades para gestión de usuarios.

Las **desventajas** que presenta MySQL son:

- Peor portabilidad que SQLite.
- El desarrollo se ha visto ralentizado desde la compra de MySQL por parte de Oracle.
- No posee una herramienta de desarrollo y depuración tan buena como las de otras bases de datos.

4.3.3. MariaDB

MariaDB es una bifurcación del sistema de gestión de bases de datos de MySQL creada en 2009 para preservar su estructura y funcionalidades. Ofrece capacidades para procesamiento de datos para tareas de menor y mayor envergadura. Posee diversas características potentes y abundantes utilidades, además de mejoras en cuanto a seguridad y rendimiento. Cada nueva versión de MariaDB es compatible con MySQL y tiene total afinidad con los datos y definiciones de tablas que emplea MySQL. Asimismo emplea protocolos cliente, APIs cliente, puertos y sockets idénticos a los de MySQL. Las **características técnicas** más destacadas de MariaDB son:

- Compatibilidad con InnoDB: es un motor de almacenamiento de propósito general conocido por ofrecer alto rendimiento y fiabilidad. Es el motor de almacenamiento por defecto en MySQL y una opción popular para usar con MariaDB. Proporciona funcionalidades estándar de transacciones que cumplen las características ACID (atomicidad, consistencia, aislamiento y durabilidad) y tiene en su haber soporte para claves foráneas.
- Código abierto.
- Incorpora 12 motores de almacenamiento.
- Proporciona múltiples métodos de autenticación por cuenta.
- Soporte para PHP.
- Proporciona retrocompatibilidad.

Ventajas e inconvenientes

Las **ventajas** que proporciona MariaDB son:

- Puede ser fácilmente integrado con otros sistemas de gestión de bases de datos relacionales.
- Buena ejecución de consultas.
- Adecuado para conjuntos de datos de gran envergadura.
- Distribuido bajo licencia GPL, por lo que da acceso completo a sus funcionalidades y software.
- Gran velocidad, mejoras en replicación y actualizaciones rápidas.

Las **desventajas** que arroja MariaDB son:

- Sistema de soporte costoso.
- Sólo soporta tipos de datos JSON.

4.4. Análisis de frameworks o plataformas

4.4.1. Django

Django es un framework web de alto nivel construido en Python por desarrolladores experimentados, que da soluciones a muchas de las complicaciones de la programación web y que fomenta el desarrollo limpio, rápido y pragmático. Además es gratuito y de código abierto con miles de usuarios y contribuidores y proporciona:

- **Completitud:** Django incorpora gran cantidad de características para el desarrollo 'de fábrica' que funcionan a la perfección, sigue principios de diseño consistentes y tiene una amplia y actualizada documentación.
- **Versatilidad:** Django ha sido empleado para la construcción de todo tipo de sitios web, desde sistemas de gestión de contenido y wikis, hasta redes sociales y portales de noticias. Puede funcionar con cualquier framework en el lado del cliente y puede devolver contenido en formato HTML, RSS feeds, JSON, XML, etc. Internamente, mientras ofrece opciones para diversas funcionalidades como distintas opciones para motores de bases de datos o de plantillas, también puede ser extendido para usar otros componentes si es menester.

- **Seguridad:** proporciona soporte para la evasión de errores comunes en seguridad, al proveer un framework diseñado para proteger un sitio automáticamente. Algunos ejemplos son desechar la colocación de información de sesión en cookies donde es vulnerable, (en Django las cookies solo contienen una clave y los datos se almacenan en la base de datos) o almacenar directamente las contraseñas en un hash de contraseñas.
- **Escalabilidad:** Django usa un componente basado en la arquitectura *shared-nothing*, donde cada parte de la arquitectura es independiente de las otras y por lo tanto puede ser reemplazada si es necesario. Teniendo en cuenta una clara separación entre las partes significa que es posible escalar para aumentar el tráfico al agregar hardware en cualquier nivel: servidores caché, servidores de bases de datos... Algunos de los sitios más concurridos han escalado a Django para satisfacer sus demandas, como por ejemplo, Instagram, Mozilla o Pinterest.
- **Mantenibilidad:** el código de este framework está escrito usando principios y patrones de diseño para fomentar la creación de código mantenible y reutilizable, como *Don't repeat yourself* (DRY) para que no exista una duplicación innecesaria, reduciendo la cantidad de código. Asimismo, se promueve la agrupación de la funcionalidad relacionada en 'aplicaciones' reutilizables y en un nivel más bajo, agrupa código relacionado en módulos siguiendo el MVC (Modelo-Vista-Controlador).
- **Portabilidad:** al estar construido en Python y ejecutarse éste en multitud de plataformas, significa que no está sujeto a ninguna en particular y puede ejecutar sus aplicaciones en muchas distribuciones de Linux, Windows o Mac OS X. Igualmente Django cuenta con el respaldo de muchos proveedores de alojamiento web y que a menudo proporcionan una infraestructura específica y documentación para el *hosting* de sitios de Django.

Las **características técnicas** más destacadas de Django son:

- Posee un motor de plantillas.
- Control de versiones de base de datos.
- Soporte para autenticación.
- Soporte para servidores web.
- Posee *mapeador* objeto-relacional.
- Gran variedad de librerías.
- Soporte para internacionalización.

Ventajas e inconvenientes

Las **ventajas** que ofrece Django son:

- Permite ahorrar tiempo: la funcionalidad básica existe por defecto, como la autenticación o la gestión de sesiones.
- Comunidad poblada: posee una rica comunidad de desarrolladores que trabajan en la mejora y mantenimiento del framework y resolución de dudas en la red, además de creadores de contenido educacional.
- Documentación: Django posee unos manuales de uso y características comprensivos, extensos, correctamente organizados y con multitud de ejemplos.
- Escalabilidad: como se comentó anteriormente, permite aplicaciones que vayan desde lo general y minimalista hasta lo particular y extenso.
- Velocidad de consultas de datos.
- Seguridad: da soluciones óptimas a ataques web como *cross-site scripting*, inyección de SQL o ataques CSRF.

Las **desventajas** que arroja Django son:

- No es un framework ideal para desarrollo de aplicaciones de tiempo real.
- Puede parecer monolítico al venir pre-empaquetado con características y librerías listas para ser usadas.
- Curva de aprendizaje elevada.

4.4.2. Flask

Flask es un *micro* framework web (al igual que Django, escrito en Python) concebido para facilitar el desarrollo de aplicaciones web bajo el patrón MVC. La designación *micro* hace referencia a que con una instalación base de Flask tenemos las herramientas necesarias para la creación de una aplicación web funcional. Sin embargo, si es necesaria alguna nueva funcionalidad, existe un conjunto amplio de extensiones que pueden ser instaladas en Flask para dotar al entorno de más funcionalidades. Algunas de estas permiten tener líneas de comando para manejar la aplicación, hojas de estilo para la web, generación de formularios HTML con clases y objetos, generar modelos de datos o autenticación de usuario y contraseña.

Sus principales **características técnicas** son:

- Flask es un *micro* framework: útil para el desarrollo ágil y rápido de una aplicación simple.
- Incluye un servidor de desarrollo: no es necesario una infraestructura con un servidor web para ir testeando la aplicación que está siendo desarrollada.
- Depurador y soporte integrado para pruebas unitarias: es posible depurar errores en el código que está siendo escrito y comprobar el valor de las variables.
- Compatibilidad con Python3.
- Compatibilidad con WSGI.
- Buen manejo de rutas.
- Plantillas con Jinja2
- Soporte nativo de cookies seguras.
- Soporte para uso de sesiones.
- Open Source y amparo bajo licencia BSD.

Ventajas e inconvenientes

Las **ventajas** que posee Flask son las siguientes

- Flexibilidad.
- Facilidad para la puesta en producción.
- Documentación: comprensiva, llena de ejemplos y bien estructurada.
- Directo y minimalista.
- Simple para creación de prototipos rápidos.

Las **desventajas** que posee Flask son las siguientes:

- Flask no tiene ORMs.
- Se requieren habilidades avanzadas para programación en Flask: además presenta dificultades para familiarizarse con una app de Flask de envergadura considerable debido a su naturaleza modular.
- Falta de herramientas: a diferencia de Django, Flask echa en falta una mayor variedad de herramientas.

4.4.3. Codelgniter

Framework PHP de código abierto (considerado como uno de los mejores para expertos) para la creación de páginas web dinámicas. Es el framework más popular de PHP para creación de sitios web rápidos y adaptables. Además, emplea el patrón MVC del que ya hemos hablado con anterioridad, posee librerías avanzadas, *plug-ins* y otros recursos que simplifican distintas funciones y procesos.

Algunas de las principales **características técnicas** de Codelgniter son:

- Distintas clases de bases de datos con soporte para varias plataformas.
- Validación de datos y formularios.
- Seguridad y filtrado XSS.
- Gestión de sesiones.
- Librerías y clases de manipulación de imágenes, envío de correo electrónico, FTP o subida de ficheros.
- Cifrado de datos.
- Internacionalización.

Ventajas e inconvenientes

Las **ventajas** que posee Codelgniter son las siguientes

- Versatilidad: Codelgniter puede trabajar con una gran cantidad de entornos o servidores, incluyendo los de alojamiento compartido.
- Compatibilidad con PHP4.
- Núcleo ligero: impide sobrecargas en el servidor y carga rápida de páginas.
- Sencilla comprensión y uso.
- Posibilidad de escritura de nuevas bibliotecas y modificación de las existentes.
- Seguridad: posesión de procesos avanzados de cifrado y descifrado y características mejoradas para asegurar la fiabilidad de aplicaciones.

Las **desventajas** que posee Codelgniter son las siguientes:

- No pone énfasis en la mantenibilidad del código puesto que no posee separación modular del mismo por defecto.

- Carece de librerías exhaustivas: las librerías que ofrece CodeIgniter no son tan exhaustivas como las que ofrecen otros frameworks PHP.
- Desarrollo de ficheros del kernel: hay ocasiones que es necesario el desarrollo de ficheros de kernel para el correcto funcionamiento de la aplicación. CodeIgniter enfrenta complejos desafíos en el momento en el que el núcleo es modificado por lo que se opta por programar código de tal manera que el núcleo permanece sin modificar, algo dificultoso.

4.4.4. WordPress

Pese a que WordPress no es una plataforma de desarrollo sino un sistema de gestión de contenidos, para crear y gestionar sitios web, decidimos tenerlo en cuenta en el análisis preliminar como un candidato para el desarrollo de la plataforma que podía otorgar *plugins* útiles para las funcionalidades que se buscaba crear. Es un proyecto de software libre y de fuentes abiertas, siendo su código creado y mantenido por una comunidad de miles de desarrolladores voluntarios, siendo por lo tanto WordPress una aplicación totalmente gratuita. WordPress permite la creación de páginas web y sus contenidos por medio de una interfaz visual, sin necesidad de realizar programación, lo que hace la labor mucho más sencilla y amigable a personas que no posean conocimientos de escritura de código. Goza de una arquitectura tremendamente flexible gracias a los temas y *plugins*, permitiéndose ajustar a casi cualquier necesidad del usuario. Los mencionados *plugins* son componentes de software con funcionalidades muy concretas que se añaden a las aplicaciones para ampliarlas, existiendo más de 50.000 *plugins* gratuitos en WordPress y otra gran cantidad de pago.

Por otro lado, los temas o plantillas son un concepto muy similar al de *plugin*, sin embargo están pensados específicamente para el diseño web. Funcionan a modo de diseños 'prefabricados' y son totalmente independientes del contenido. Algunas de las principales **características técnicas** de WordPress son:

- Panel de administración y configuración: permite a los administradores y colaboradores autorizados administrar y mantener el sitio web. Contiene todas las funciones de gestión de la página, como el editor de contenidos, personalizado de temas, *plugins* y *widgets*, además del menú de ajustes.
- Editor de contenidos: herramienta para edición, adición y publicación de entradas y páginas.
- *Plugins*
- *Widgets*
- Editor CSS: permite personalizar las plantillas de WordPress.

Ventajas e inconvenientes

Las **ventajas** que posee WordPress son las siguientes

- Gratuidad.
- Flexibilidad.
- Escalabilidad.
- Facilidad de uso.
- Comunidad: uso mundial y foros de ayuda de gran utilidad.
- Código abierto: se mejora el código principal y se crean nuevos *plugins* y temas continuamente.

Las **desventajas** que posee WordPress son las siguientes:

- La mayoría de plantillas, especialmente las gratuitas, suelen ser bastante limitadas en cuanto a posibilidades de personalización.
- Seguridad: al ser de código abierto, WordPress es atractivo para potenciales atacantes. Sin embargo aplicando *plugins* de seguridad y las mejores prácticas, el riesgo se reduce en gran medida.
- Contenido de terceros: la mayoría de *plugins* y temas son creados por desarrolladores externos, por lo que pueden contener errores.
- Tiempos de carga: si se tiene un gran número de *plugins* y código innecesario, es posible que la velocidad del sitio descienda, causando tiempos de carga lentos.

4.4.5. Comparativa de frameworks web

Antes de empezar con la comparativa en forma de tabla, anunciaremos que WordPress no entrará en la misma puesto que fue descartado en fases iniciales del análisis, los motivos los comentaremos en el siguiente punto, donde trataremos la elección definitiva y los motivos que nos han llevado a la misma, entre otros aspectos.

En la tabla 4.1 se presenta una comparativa entre los 3 frameworks restantes, analizando distintos aspectos técnicos que poseen cada uno:

	Django	Flask	CodeIgniter
Lenguaje	Python	Python	PHP ≥ 7.2 ó 5.6.0
Ajax	Sí	Sí	Cualquiera
MVC	Sí	-	Sí
MVC push-pull	Push	-	Push
i18n y L10n	Sí	Sí	Generalmente
ORM	Sí	Agnóstico	3 ^{as} partes
Testing	Sí	Unittest	Unit tests
Migraciones	Sí	Depende ORM	Sí
Seguridad	Sí	Sí	Sí
Plantillas	Incorporadas, Jinja2, Cheetah, Mako	Jinja2	Sí
Caching	Sí	Sí	Sí
Validación de formularios	Sí	Sí	Sí
Scaffolding	-	-	No
Rad	-	-	Sí
Movilidad	-	-	Plantillas
Python 3	Sí	Sí	-

Tabla 4.1: Tabla comparativa de frameworks

4.5. Selección de alternativas

4.5.1. Biblioteca PLN

Analizando las opciones descritas en el apartado correspondiente de comparación de bibliotecas PLN para el desarrollo de la librería de análisis textual, en primer lugar se desecharon las opciones que ofrecía Java (Java Apache OpenNLP) y C++ (MITIE) por desconocimiento de las mismas, el poco soporte y documentación que existe respecto a ellas y la experiencia que ya se poseía con las herramientas Python. Por lo que la opción escogida finalmente fue **NLTK** sobre spaCy por los motivos que se exponen:

- spaCy está más orientada a análisis morfosintáctico.
- NLTK presenta funcionalidades mucho más útiles para la construcción de la librería.
- La cantidad de información y documentación que se encuentra sobre NLTK.
- La experiencia y soltura que se posee por haber trabajado previamente con NLTK durante el grado.
- La versatilidad que ofrece.
- La compatibilidad y facilidad de incorporación a un framework Python de los propuestos.

4.5.2. Base de datos

Tras la revisión de sistemas de gestión de bases de datos relacionales, se decidió que la plataforma escogida para el sitio web sería **MySQL** y las razones que nos llevaron a dicha conclusión sobre las demás opciones son las siguientes:

- Se eligió MySQL sobre SQLite por tener la certeza de que no habría problemas de rendimiento en la base de datos, dado que SQLite no es la mejor de las opciones para proyectos de una envergadura superior.
- Se eligió MySQL sobre MariaDB por la facilidad de uso que ofrece MySQL y por la experiencia que se poseía con el sistema al haber trabajado con él durante el grado.
- Por la escalabilidad que posee MySQL.
- Por la seguridad total de los datos que proporciona MySQL.
- Por la facilidad de uso y sencillez de programas de administración de bases de datos compatibles con MySQL que querían emplearse para la gestión de la misma durante el desarrollo como MySQL Workbench.

4.5.3. Framework

Tras hacer una revisión de las opciones que valoramos para la realización del proyecto, como comentamos anteriormente, la primera alternativa que se descartó fue WordPress debido a que se buscaba una plataforma lo más similar a la del prototipo realizado en el storyboard y esa libertad de personalización no la concedía WordPress por lo que se decidió que la mejor alternativa para realizar un sitio lo más similar y fiel a la idea inicial de las funcionalidades que se querían incorporar al mismo, era una de las otras 3, Django, Flask o CodeIgniter. Dicho esto, pasaremos a comentar cuál fue la elección para el desarrollo y la justificación del mismo.

El framework elegido para el sistema fue **Django**, y las razones las exponemos a continuación:

- La cantidad de funcionalidades que Django trae 'de serie' como el sistema de autenticación, el sistema de base de datos y modelos o el sitio de administración que harían el desarrollo más ágil, rápido y sencillo.
- El robusto sistema de seguridad que posee y las buenas prácticas que emplea, haciendo sencilla la evasión de los ataques web más típicos.
- La escalabilidad de Django puesto que en caso de que el sistema tenga un gran número de usuarios y contenido en el mismo, sería sencillo escalar cuando se requiera una envergadura mayor, aplicando la solución que mejor se ajuste a las necesidades de la aplicación (previamente comentadas en el análisis de frameworks).
- La portabilidad previamente comentada. El desarrollo se realizó en un sistema Windows, mientras que será alojado en un servidor Debian, Django permite la ejecución de sus aplicaciones en cualquier sistema operativo indistintamente.
- Django es un framework Python, lo cual es una ventaja por la sencillez y facilidad de programación del mismo. Además, al ser la librería de análisis de corpus textuales escrita en Python, se podía asegurar que no habría problema con la codificación de los algoritmos de análisis textual ni con la inclusión de librerías de Procesamiento de Lenguaje Natural para poder realizar dicho análisis, las cuales son especialmente potentes en Python y ya existía familiarización con ellas debido a que son impartidas durante el grado.
- La cantidad de información y referencias que existen en la red sobre Django y proyectos que se han realizado en el mismo, lo cual agiliza el desarrollo en caso de que alguna funcionalidad no termine de solventarse. La comunidad es atenta y servicial para cualquier tipo de duda que se plantee, además de la documentación oficial del framework, que es

extensa, muy útil y está repleta de ejemplos que ayudan a comprender mejor el aspecto que se está tratando en una sección en concreto.

Todo esto convierte a Django en la mejor opción para el proyecto que se planteó. Por otro lado, la elección finalmente estuvo entre Flask y Django, Codelgniter es un framework potente, sin embargo emplea PHP y preferimos decantarnos por un entorno de trabajo que emplease Python para casar así con la librería de análisis textual, además de explorar otras opciones de entornos de desarrollo web. Flask, aunque es un framework Python como Django, no presenta una escalabilidad tan marcada como la de Django y podría entorpecer en mayor medida el desarrollo debido a las habilidades que se requieren para la construcción de un sitio web con Flask.

Una vez hemos definido el framework, el sistema de gestión de bases de datos y la biblioteca PLN con los que realizar el trabajo, pasaremos a hacer un repaso del desarrollo del proyecto, revisando lo realizado en cada iteración, dificultades encontradas y funcionalidades creadas.

Capítulo 5

Desarrollo inicial

En este apartado hablaremos más detenidamente sobre el desarrollo inicial del proyecto, previo a la realización de código.

5.1. Iteración 1: Inicialización del proyecto

Esta iteración corresponde a la realización de los pasos previos al desarrollo del software de la plataforma, construyendo un diseño preliminar a modo de prototipo a papel de la misma y realizando un análisis de las herramientas a emplear para la creación del sistema.

5.1.1. Análisis de software para desarrollo de la plataforma

En esta sección no nos extenderemos puesto que el análisis ya ha sido detallado en puntos anteriores por lo que realizaremos un breve resumen en cada subapartado que incumba al análisis de las tecnologías empleadas.

Análisis de software para desarrollo de tecnologías web

Se realizó un estudio detallado de las herramientas empleables en la construcción del sitio web que alojase el sistema, consultando en la red artículos de expertos en el campo para llegar a la conclusión de que la mejor opción en nuestro caso era **Django**.

Análisis de software para bases de datos

Se procedió a la ejecución de un estudio detallado de los sistemas de bases de datos empleables para el alojamiento de la información del sistema, consultando en la red artículos de expertos en el campo e información respectiva a cada uno de los candidatos para llegar a la conclusión de que la mejor opción en nuestro caso era **MySQL**.

Análisis de software para análisis textual

Confección de una investigación detallada de las herramientas de análisis textual y Procesamiento de Lenguaje Natural aprovechables para la construcción de la librería de análisis de corpus, consultando en la red artículos de expertos en el campo, contrastando las diferentes opciones e información respectiva a cada uno de los candidatos para llegar a la conclusión de que la mejor opción en nuestro caso era **Python-NLTK**.

5.1.2. Realización de storyboard, sección de gestión de usuarios y colecciones

Se procedió a la construcción de un guión gráfico que sirviera como prototipo de la idea inicial sin funcionalidad para el sistema. Se empleó la herramienta *Diagrams.net* como ya mencionamos en la parte de software, que proporciona características para creación de esquemas. Se construyeron las vistas de la aplicación correspondientes a la gestión de usuarios y documentos (más adelante se realizarían las de análisis textual) empleando formas y plantillas orientadas al desarrollo de *mockups* y guiones gráficos, uniendo las vistas por medio de líneas con orientación, partiendo estas del botón o elemento interactuable del sistema y acabando en la pantalla correspondiente que se genera al realizar la acción de tal manera que el usuario al que se le muestra la idea inicial puede seguir los movimientos que se producen mediante sus interacciones con el sitio web como si se tratase de un prototipo a papel clásico. A continuación se plasman las escenas realizadas para el prototipo. Para la navegación indicaremos junto a los botones que permiten la misma, el número de escena a la que lleva la acción realizada.

CEALMTools

https://www.cealmtools.es

Iniciar sesión

Correo electrónico

Contraseña

☒ Recuérdame

Iniciar sesión (3)

[¿Olvidó su contraseña?](#)

¿No dispone de cuenta?

Crear cuenta (2)

Figura 5.1: Escena 1: inicio de sesión

CEALMTools

https://www.cealmtools.es

Registro

Correo electrónico

Nombre y apellidos

Institución

Contraseña

Confirmar contraseña

Crear cuenta (3)

Figura 5.2: Escena 2: registro

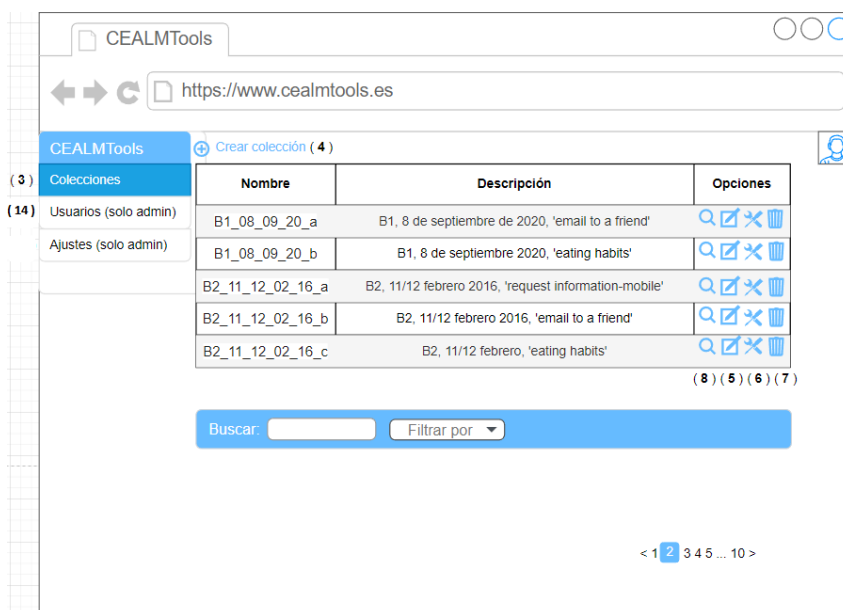


Figura 5.3: Escena 3: colecciones



Figura 5.4: Escena 4: crear colección



Figura 5.5: Escena 5: editar colección

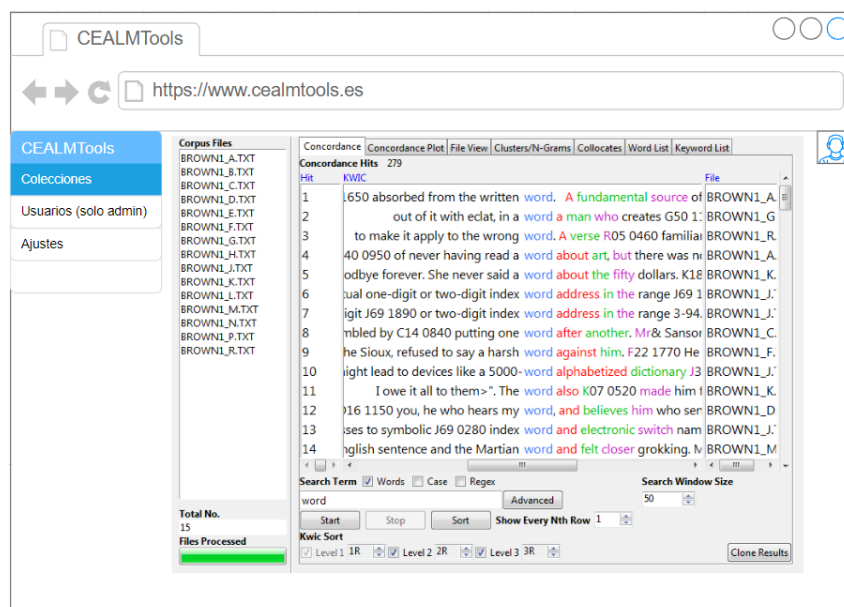


Figura 5.6: Escena 6: herramientas

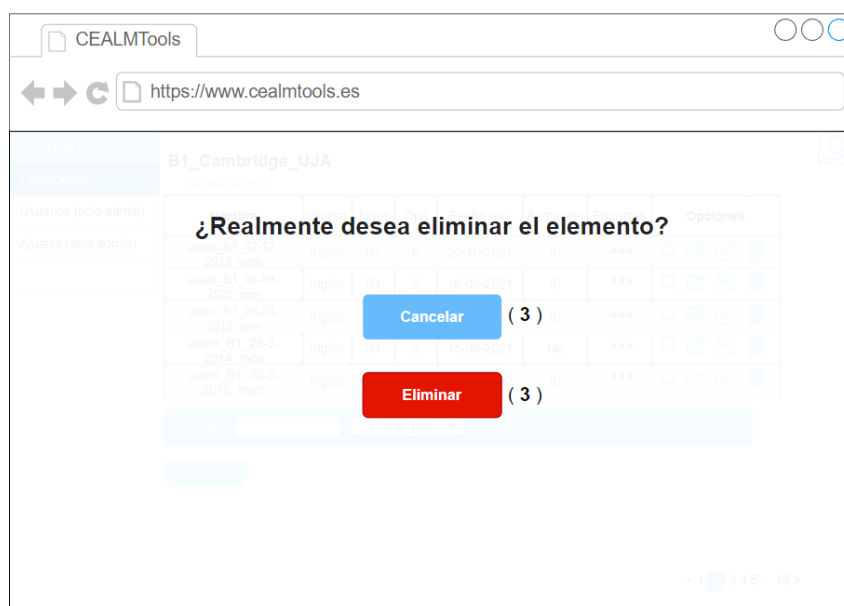


Figura 5.7: Escena 7: eliminar colección

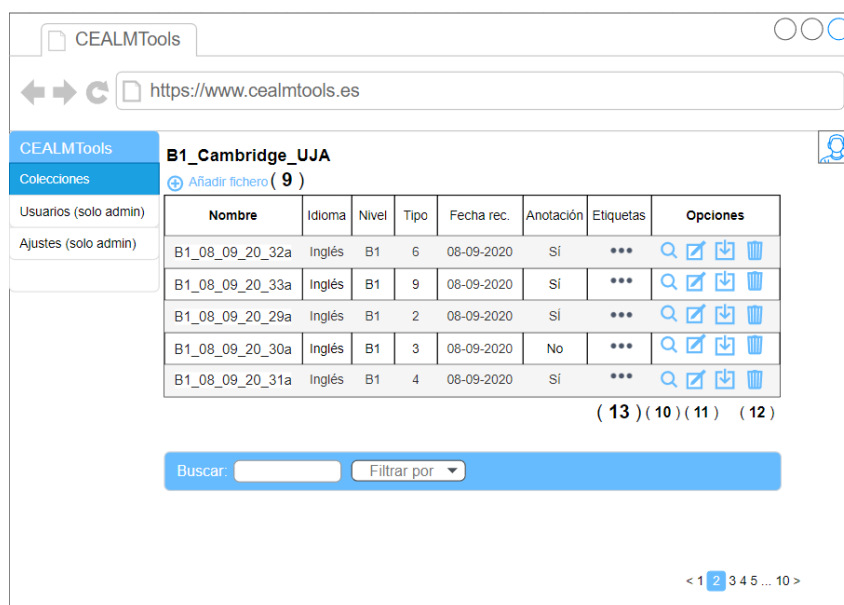


Figura 5.8: Escena 8: vista detallada de colección

The screenshot shows the CEALMTools web interface at <https://www.cealmttools.es>. The left sidebar contains navigation links: CEALMTools, Colecciones, Usuarios (solo admin), and Ajustes (solo admin). The main area is a form for creating a document. It includes fields for:

- Nombre fichero: B1_08_09_20_32a
- Idioma del texto: Selecccionar idioma (dropdown)
- Nivel del texto: Selecccionar nivel (dropdown)
- Tipo de texto: Selecccionar tipo (dropdown)
- Fecha recuperación: 08-09-2020
- Anotación: Selecccionar anotación (dropdown)
- Etiquetas: UJA x, B1 x, CEALM x
- Selecccionar fichero: Selecccionar archivo
- Selecccionar archivo de otra colección: B1_08_09_20_32a.txt

 At the bottom, there is a blue 'Subir' button and a '(8)' indicator.

Figura 5.9: Escena 9: crear documento

The screenshot shows the CEALMTools web interface displaying a document. The document title is 'B1_Cambridge_UJA' and the file name is 'B1_08_09_20_32a'. The document content is a text message from 'XXX' to 'Lawrence!'. The text includes a greeting, a statement about the user's family and the risk of coronavirus, and suggestions for activities in Jaén. The interface includes a sidebar with navigation links, a table of document metadata, and a list of document options. The document content is displayed in a large text area with a blue highlight. The bottom right corner shows a page number '345' and a total of '10' pages.

Nombre	Idioma	Nivel	Tipo	Fecha rec.	Anotación	Etiquetas	Opciones
From: XXX				20-10-2021	SI	***	
Subject: Reply, I need holiday advice!							

Figura 5.10: Escena 10: visualizar documento

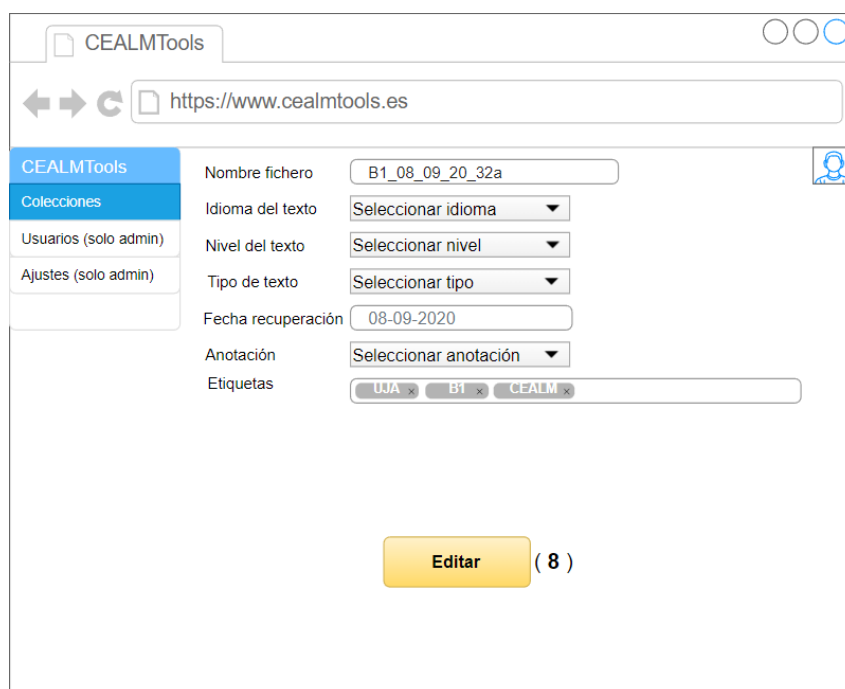


Figura 5.11: Escena 11: editar documento

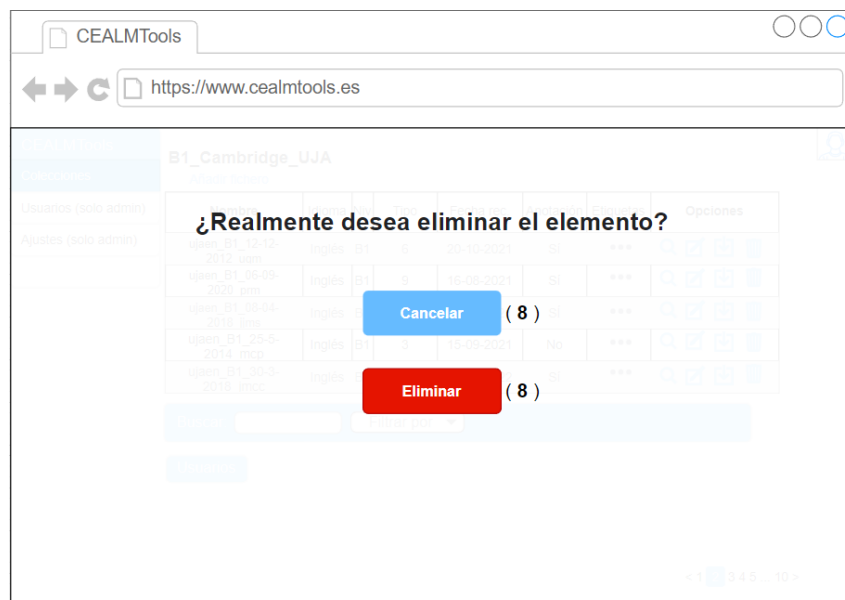


Figura 5.12: Escena 12: eliminar documento

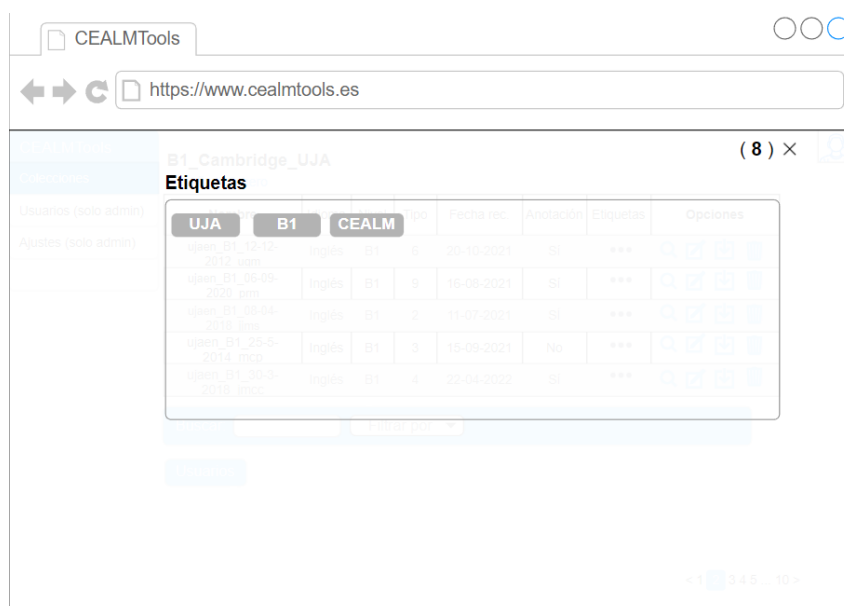


Figura 5.13: Escena 13: etiquetas de los documentos



Figura 5.14: Escena 14: vista de usuarios

CEALMTools

https://www.cealmttools.es

CEALMTools

Colecciones

Usuarios (solo admin)

Ajustes (solo admin)

Nombre usuario: javierpaloca

Nombre completo: Javier Palomino Carreño

Correo electrónico: javierpaloca@red.ujaen.es

Institución: Universidad de Jaén

Fecha de alta: 16-11-2021

Fecha de baja: 14-07-2022

Tipo usuario: Seleccionar tipo

Equipos: UJA_ENG x UJA_BI x CEALM_TEAM x

Añadir (14)

Figura 5.15: Escena 15: crear usuario

CEALMTools

https://www.cealmttools.es

CEALMTools

Colecciones

Usuarios (solo admin)

Ajustes (solo admin)

Nombre usuario: belendb

Nombre completo: María Belén Díez Bedmar

Correo electrónico: belendb@cealm.es

Institución: Universidad de Jaén

Fecha de alta: 16-11-2021

Fecha de baja: 14-07-2022

Tipo usuario: Seleccionar tipo

Equipos: UJA_ENG x UJA_BI x CEALM_TEAM x

Editar (14)

Figura 5.16: Escena 16: editar usuario

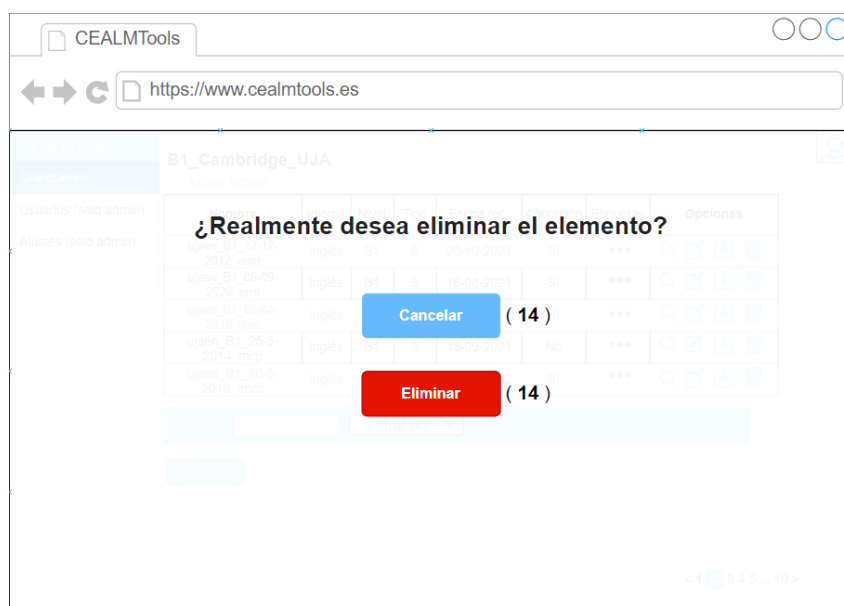


Figura 5.17: Escena 17: eliminar usuario

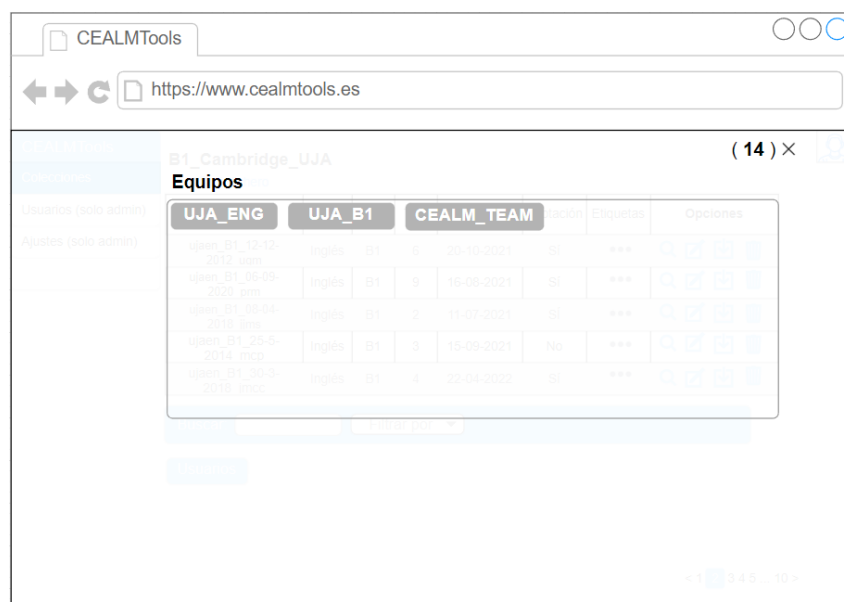


Figura 5.18: Escena 18: visualizar equipos

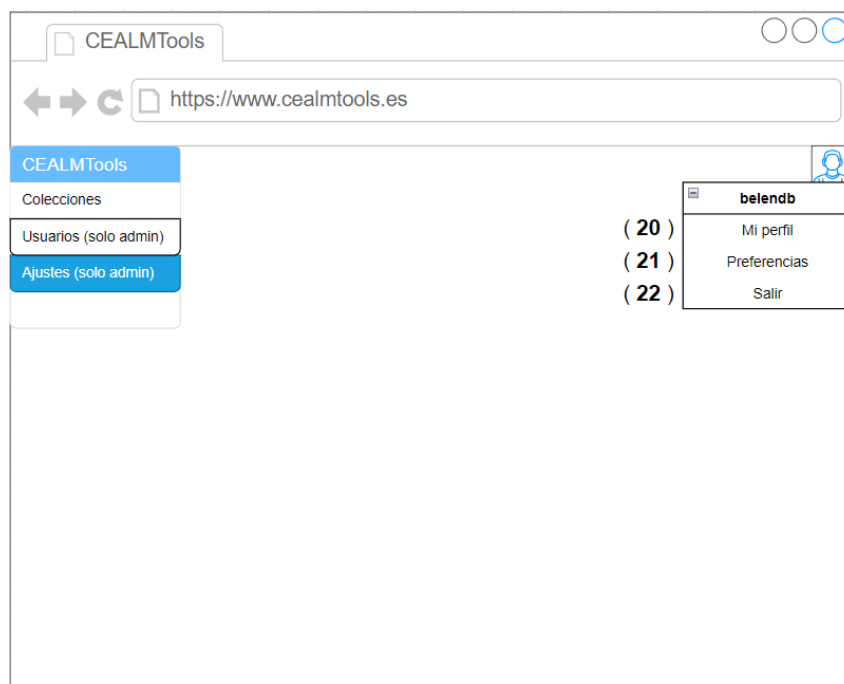


Figura 5.19: Escena 19: menú de perfil

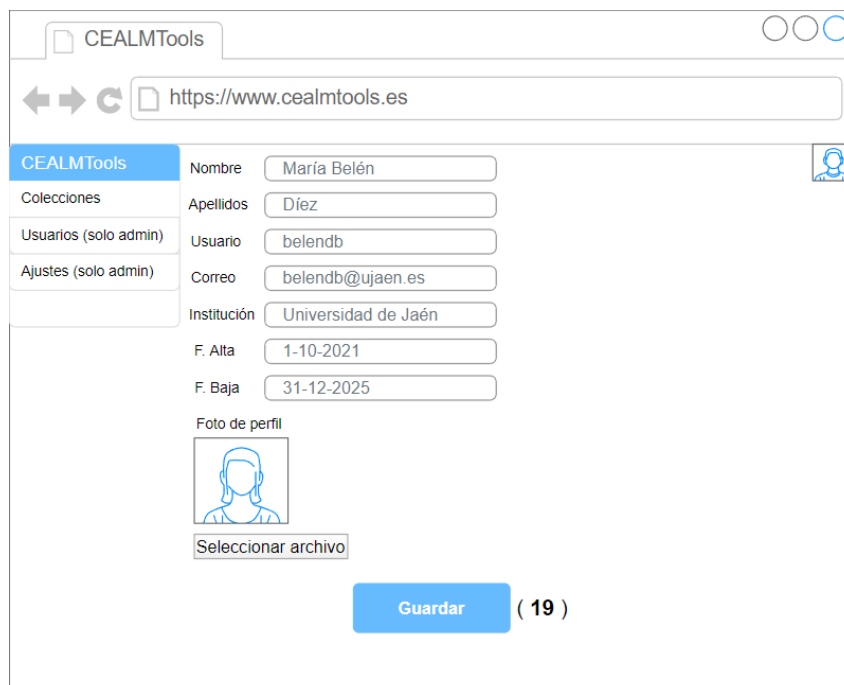


Figura 5.20: Escena 20: vista de perfil de usuario

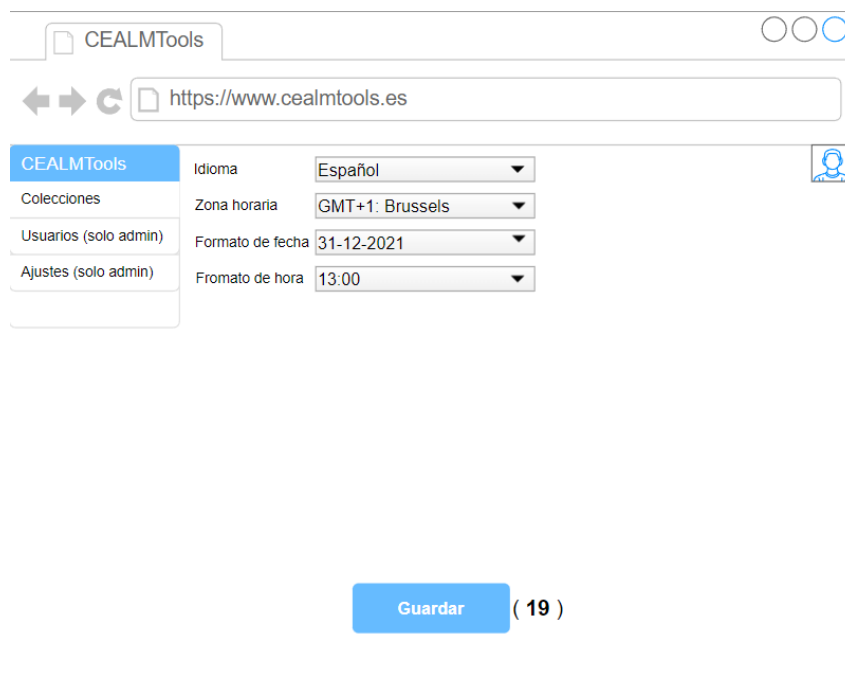


Figura 5.21: Escena 21: preferencias

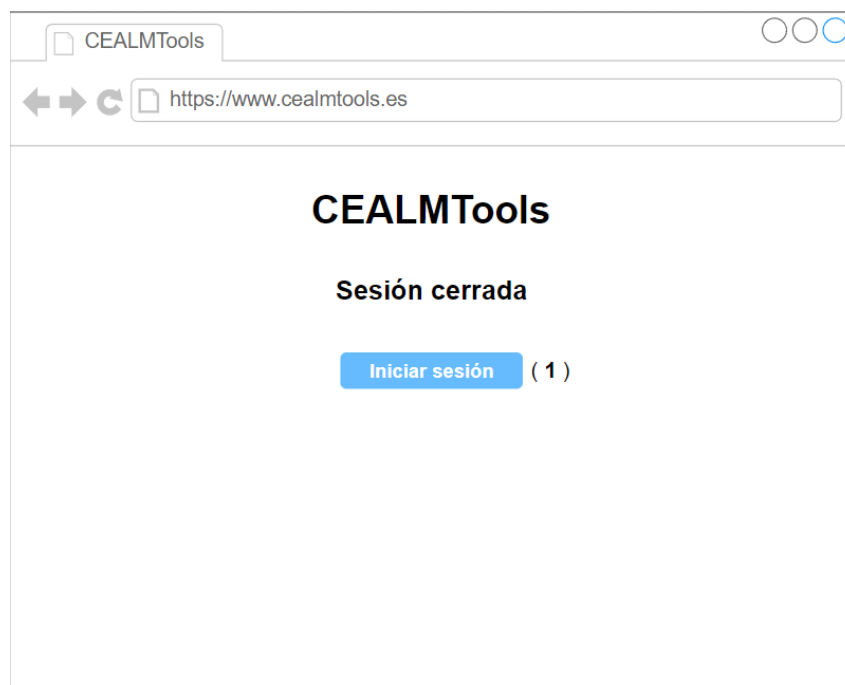


Figura 5.22: Escena 22: cerrar sesión

Capítulo 6

Desarrollo del proyecto

En este apartado se tratará la explicación detallada del software desarrollado para la plataforma creada, explicando ciertos conceptos como la construcción de vistas, controladores o modelos, así como el funcionamiento de los algoritmos de Procesamiento de Lenguaje Natural para análisis textual.

6.1. Iteración 2: desarrollo del login y registro en plataforma

Se procedió al desarrollo de las vistas para registro e inicio de sesión así como el software que requiere la autenticación y creación de usuarios.

6.1.1. Login

En primer lugar lo más importante es la creación de un modelo de usuario. Django otorga uno por defecto con el que se puede trabajar cómodamente y realizar el login empleando el nombre de usuario y una contraseña. Sin embargo, uno de los primeros escollos del desarrollo fue el hecho de desear realizar el inicio de sesión en la plataforma con el correo electrónico y una contraseña. Django no permite configurar el uso de credenciales para el inicio de sesión si se emplea el sistema por defecto, por lo que se optó por la realización de un modelo de usuario propio del que hablaremos a continuación.

Modelo de usuario personalizado

Esta clase para el registro de usuarios se creó desde cero y permite mucha más libertad que el sistema por defecto. Se le llamó **UserCustom** y se incluyeron los siguientes campos:

- **Email:** correo electrónico del usuario.

- **Name:** nombre del usuario.
- **Surname:** apellidos del usuario.
- **Institution:** institución a la que pertenece el usuario.
- **Membership_Date:** fecha en la que el usuario se registró en la plataforma.
- **Leave_Date:** fecha de baja de la cuenta del usuario, establecida por defecto dos años después de la creación de la misma.
- **Manager:** indica si el usuario es gestor del sistema.
- **Admin:** indica si el usuario es administrador del sistema
- **Is_active:** indica si el usuario sigue activo en la plataforma.

Se crearon los procedimientos de apoyo del modelo para acceder a los atributos de la misma y una clase de apoyo denominada **UserManager** para gestionar la creación de usuarios (tanto bases, como gestores, como administradores).

Formulario de inicio de sesión

En este caso y en muchos otros, emplearemos el sistema de formularios que ofrece Django, mediante el cual seleccionamos los campos del modelo que queremos que aparezcan en el mismo y se construye uno con los parámetros que se le indiquen, como clase de bootstrap, *placeholder* y tipo de *input* de HTML que deseemos, además de preparar el CSS y Javascript necesario (que los datos que se introduzcan en un campo del formulario, coincidan con el tipo del mismo, que se rellenen todos ellos antes de enviar el formulario, etc) en cada caso, lo cual nos ahorrará mucho código y tiempo al tener tan solo que llamar al formulario en el código HTML de la plantilla de la vista. En este caso serán los campos correo electrónico y contraseña de tipos *email* y *password* respectivamente.

Controlador para inicio de sesión

Se hacen las comprobaciones pertinentes en un inicio de sesión, si los parámetros son correctos, el usuario será redirigido a la página de inicio, en caso contrario deberá volver a introducir los credenciales.

Vista de inicio de sesión

Llamamos al formulario y habilitamos los botones necesarios para el envío del formulario.

6.1.2. Registro

Al mismo tiempo se desarrolló el sistema de registro de un usuario en la plataforma, empleando para ello las siguientes directrices:

Formulario de registro

Se construyó un formulario para el registro, proveniente del modelo de usuario personalizado que creamos con anterioridad, que contiene los campos nombre, apellidos, correo electrónico e institución además de una contraseña junto con su confirmación y comprobación correspondiente para asegurar fiabilidad de la misma.

Controlador de registro

Se trata de un controlador simple en el que se llama al formulario, la plantilla a renderizar y la URL que se carga en caso de que los datos del formulario sean correctos.

Vista de registro

Plasmamos los campos del formulario en el código HTML, el botón para confirmación de envío de formulario y el enlace de vuelta a inicio de sesión.

6.2. Iteración 3: implementación de funcionalidades de seguridad

Con motivo de reforzar la seguridad en cuanto a acceso a ciertas vistas en función del rol del usuario, se crearon una serie de decoradores útiles para este cometido. Un **decorador** en Python, de manera resumida, es un patrón de software utilizado para alterar el funcionamiento de una pieza de código, ya sea una función, o una clase, sin la necesidad de emplear otros mecanismos como la herencia. A continuación presentamos los decoradores desarrollados con vistas a mejorar la seguridad del sistema:

- ***allowed_users(allowed_roles=[])***: este decorador recibe por su cabecera los roles de usuario permitidos para una vista en concreto y en función del usuario que acceda a ella, se le concederá paso o no.
- ***authenticationDecorator()***: comprueba si el usuario que está intentando acceder a una URL en concreto posee una sesión iniciada en la plataforma, en caso contrario lo redirige a la vista de inicio de sesión.

Finalmente en el caso del decorador de autenticación se optó por emplear el que posee Django por defecto y otorga un buen rendimiento denominado *login_required* al que se le pasa la URL que debe mostrar en caso de que un usuario intente acceder a una vista de la plataforma y no posea una sesión iniciada.

6.3. Iteración 4: implementación de sección para gestión de usuarios

La cuarta iteración consistió en el desarrollo de la sección para gran parte de la gestión de usuarios, el resto se haría más adelante en iteraciones posteriores. Es necesario mencionar que para la gestión de usuarios se crearon 3 roles dentro de la plataforma que nos disponemos a enunciar a continuación:

- **Administrador:** se trata del administrador único del sitio y tiene potestad plena para la gestión y administración de usuarios, colecciones y documentos, así como acceso a las herramientas de las colecciones.
- **Gestor:** pueden existir varios en la plataforma y pueden realizar gestión de colecciones y documentos, siendo posible para ellos, crear, editar, leer y eliminar los mismos. Asimismo, los gestores tienen acceso a las herramientas de las colecciones, sin embargo no lo poseen para la sección de gestión de usuarios.
- **Usuario base:** pueden existir múltiples dentro del sistema y tendrán acceso exclusivamente a las herramientas asociadas a las colecciones, no siendo posible para ellos, realizar ninguna modificación, creación o eliminación de registros, tanto de usuarios como de colecciones y documentos.

6.3.1. Tabla de registros de usuario

Esta es la primera vez que empleamos en la construcción del sitio las tablas de datos de CDN de Bootstrap, de las cuales hablaremos a continuación. Éstas serán de gran ayuda para representar información de forma gráfica cuando se requiera y el proceso seguido para ello consiste en la creación de un controlador que realice una consulta a la base de datos sobre la información de los usuarios y una vez obtenidos los datos, en la vista rellenamos la tabla con las propiedades que posee cada entrada de la tabla de usuarios de la base de datos y las respectivas acciones que se pueden realizar con cada entrada (las cuales comentaremos en los siguientes puntos).

Tablas de datos de CDN de Bootstrap

Las tablas de datos de CDN con estilo Bootstrap de permiten mostrar una serie de registros con varios atributos de una tabla de una base de datos. Son fácilmente importables y de manejar por código Javascript. Poseen varias funcionalidades por defecto de gran utilidad para nuestros cometidos como son:

- **Muestra limitada de registros:** los registros de la tabla se mostrarán en series de un número determinado de registros. Por ejemplo, en el caso de nuestra plataforma, los registros es posible mostrarlos en pestañas de 10, 20 ó 30 elementos o incluso mostrarlos todos en una sola página. Esto es útil si no queremos saturar la vista en la que estemos navegando en ese momento. Para navegar entre secciones podemos hacer uso de las páginas y botones de la parte inferior derecha de la pantalla.
- **Ordenación de columnas:** al hacer click en las flechas que aparecen al lado de cada nombre de cada columna de la tabla podemos alterar el orden de los registros que aparecen en la tabla y mostrarlos en base a ese atributo en concreto (por ejemplo en un atributo numérico podemos ordenar registros de menor a mayor o de mayor a menor). Útil para ordenaciones concretas personalizadas.
- **Búsqueda de datos:** poseemos una caja de texto en la esquina superior derecha que nos permite buscar registros en la tabla. Podemos introducir el término que queramos y si coincide con cualquier valor de atributo de cualquier registro, se filtrarán y aparecerán solamente los ocurientes con la búsqueda.

6.3.2. Pestaña de edición de usuario

Es posible modificar los datos de un usuario tras su creación en la plataforma. Para este cometido, volveremos a seguir los pasos tomados para otros formularios, desarrollando una vista un formulario derivado de un modelo y un controlador para la sección.

Formulario de edición de usuario

Para este formulario permitiremos la edición de los atributos nombre, apellidos, correo electrónico, institución y fecha de baja del modelo de usuario de la plataforma. Todos los campos serán de texto y la fecha del tipo homónimo.

Controlador de edición de usuario

En este caso el controlador estará dividido en dos partes: la primera consistirá en acceder al registro de la base de datos y devolver los datos del usuario al lado del cliente. Una vez hecho

esto, en el momento que el administrador desee actualizar la información del usuario, enviará el formulario al lado del servidor y este actualizará la información del registro, redirigiendo al administrador de nuevo a la vista de la tabla de usuarios.

Vista de edición de usuario

En la vista construimos el formulario como en otras ocasiones y creamos los botones correspondientes para confirmación o cancelación de la operación.

6.3.3. Pestaña de eliminación de usuario

Para eliminar registros de usuario del sistema, la funcionalidad necesaria se desarrolló de la siguiente manera:

Controlador para eliminación de usuarios

Controlador simple mediante el cual, en base al usuario seleccionado y confirmación de borrado, se realiza la búsqueda de la entrada en la tabla de la base de datos y se realiza un borrado de la misma, anulando al usuario del sitio web.

Vista de eliminación de usuario

Plantilla HTML simple en la que podemos denegar o confirmar la operación.

6.4. Iteración 5: implementación de sección para gestión de colecciones

En esta iteración del desarrollo se implementó toda la parte correspondiente a la gestión de documentos y colecciones de la plataforma.

6.4.1. Pestaña de creación de colección y tabla de registros

Tabla de registros en vista general

En cuanto a la tabla de registros de colecciones, el proceso será el mismo que el empleado para la de usuarios, usando el mismo tipo de tabla de datos. En este caso el modelo desde el que se recuperará la información será el de colecciones denominado *Collection*. Crearemos la tabla con los campos del modelo a mostrar y las distintas opciones de operativa sobre los documentos de la colección y los conjuntos de documentos en sí, como por ejemplo las herramientas de análisis textual.

Creación de colecciones

Para crear colecciones emplearemos un formulario simple a rellenar con los datos básicos sobre una cualquiera. Tanto identificador del corpus como su propietario, son atributos que se rellenarán automáticamente en el lado del servidor, de tal forma que el ID sea único en formato universal y el propietario sea el usuario que creó la colección. Presentamos las partes que componen esta utilidad:

Formulario de creación de colección

El formulario empleado nos será útil tanto para crear registros de colecciones como para editarlos, funcionalidad que posteriormente explicaremos. Los campos que deberá rellenar el usuario serán todos salvo el identificador de la colección y el propietario como hemos explicado anteriormente por lo que los atributos serán: el nombre, el idioma, el nivel de idioma que posee, la fecha de recuperación y la anotación. Tanto el idioma como el nivel como la anotación de la colección serán internamente 3 modelos que poseerán cada uno las opciones para cada campo, es decir, el modelo de idiomas, denominado **Languages** poseerá los valores español e inglés, el modelo para los niveles de los textos, llamado **Levels** tendrá registradas las entradas A1, A2, B1, B2, C1 y C2 y para el caso de la anotación, ésta estará alojada en un modelo con el nombre **Notes** tan sólo tendrá en su haber los registros y por lo tanto opciones sí o no. Se emplea esta forma de proceder puesto que en el caso de los 3 campos mencionados emplearemos un desplegable para cada uno para brindarle las opciones disponibles al usuario. Los campos restantes, nombre y fecha de recuperación, serán de tipo texto y fecha respectivamente.

Controlador para creación de colecciones

Se renderiza el formulario y una vez el usuario lo envíe, se procesa en el lado del servidor en caso de que sea válido, estableciendo el propietario como el que crea el corpus y el identificador internamente. Una creación exitosa de colección lleva al usuario a la vista general de colecciones, donde se encuentra la tabla de registros.

Vista de creación de colecciones

Utilización de código HTML simple. Se llama al formulario para que se renderice por completo y se crean los botones de confirmación o cancelación para navegación.

6.4.2. Pestaña de edición de colección

En el caso de la edición de colecciones nos encontramos con una operativa y vistas prácticamente idénticas a las de creación de colecciones. La diferencia radicará en que el

formulario en primera instancia se presentará cargado con la información perteneciente al registro de colección que deseamos modificar. Una vez el usuario confirme la acción, si se ha modificado algún campo, este se actualizará con el cambio acometido.

6.4.3. Pestaña de eliminación de colección

Para la eliminación de colecciones, empleamos una vista y funcionalidad muy similar a la de usuarios. La diferencia radicará en que una colección no podrá ser eliminada de la plataforma hasta que no esté vacía, o lo que es lo mismo, no es posible eliminar una colección que posea en el momento de la eliminación documentos vinculados a ella. Si este es el caso, obtendremos una pantalla en la que se nos indicará que no es posible eliminar el elemento y que deberemos primero 'limpiar' el corpus de documentos de texto para poder realizar el desecho del registro. En el momento que una colección se halle vacía, sí podremos eliminar la colección del sistema.

6.4.4. Implementación de subida de documentos a la plataforma y asignación a colecciones

En esta etapa del desarrollo, los esfuerzos estuvieron centrados en toda la funcionalidad de subida de documentos individuales y comprimidos en ficheros ZIP, su edición y eliminación, codificación de los mismos, etiquetado, eliminación múltiple, etc. La sección de documentos estará presente en cada colección de ficheros y en ella aparecerán sólo los vinculados al presente corpus. La estructura de la vista es la misma que la de las de registros de usuarios y colecciones, estando compuesta por una tabla de elementos del modelo de documentos, en este caso llamado *Document* con sus respectivas columnas aunque en este caso sólo se mostrarán el nombre, el idioma, el nivel, la fecha de recuperación, la anotación, el propietario y las etiquetas asociadas junto con las opciones de edición y eliminación de registros para cada elemento. Así mismo, poseemos ciertas utilidades de las que hablaremos en breves instantes.

Subida de documentos a la plataforma

Dentro de la sección de documentos de cada colección podremos añadir nuevos ficheros de texto a la misma.

Formulario de subida de documentos

Para este cometido, se elaboró un nuevo formulario de Django denominado *CreateDocumentForm* que requiere que se rellenen los campos: nombre (tipo texto), idioma (desplegable), nivel (desplegable), fecha de recuperación (tipo fecha), anotación (desplegable), etiquetas del documento (tipo especial derivado de librería Python) y subida del fichero de texto

al sistema (tipo archivo). Independientemente del modelo, también crearemos en la plantilla HTML un desplegable con las codificaciones disponibles para que el usuario seleccione la del documento a subir y así poder realizar la codificación correctamente a UTF-8 en el lado del servidor.

Controlador de subida de ficheros

Este controlador posee un nivel más elevado de complejidad que el resto de los revisados hasta ahora puesto que tenemos el componente de codificación, gestión de etiquetas y subida de ficheros de texto al sitio. Explicaremos el funcionamiento básico del mismo y nos centraremos en las funcionalidades concretas en los puntos correspondientes. Aclarar en primer lugar que los ficheros se guardarán en la ruta establecida en el fichero *settings.py*. El funcionamiento básico es el mismo que en casos anteriores, generando en caso de que el formulario sea correcto una instancia de documento en la base de datos y guardando el fichero asociado en el servidor. Para evitar nombres de instancias y **slugs** (versiones únicas de nombres de etiquetas para URLs) iguales, si se suben documentos con el mismo nombre, el sistema tratará de alterar el del documento a cargar añadiéndole una serie de números aleatorios hasta que el nombre generado sea único. En el momento de la subida, se guardan asimismo el mencionado **slug**, el propietario del documento (que será nuevamente el usuario que creó el objeto) y el fichero de texto asociado, además de, por defecto, vincular la presente colección al nuevo documento. Una vez confirmada la carga, volveremos a la tabla de registros.

Vista de subida de ficheros

En la plantilla HTML para esta vista llamaremos, como es habitual, al formulario. Para el campo de etiquetas emplearemos un tipo especial de cuadro de texto, que genera las mismas a partir de las palabras separadas por comas o espacios que se encuentran en su interior. En el caso del desplegable de codificaciones, crearemos uno simple con HTML para el posterior uso del valor seleccionado en el lado del servidor. Se añaden los botones para envío del formulario y cancelación de operación.

Funcionalidad para etiquetas en documentos

Se usó la librería de software de etiquetas *Taggit* la cual proporciona la funcionalidad necesaria para la gestión y uso de etiquetas en desarrollos web Python. Sin embargo, al incorporar la importación de etiquetas desde el nombre de documentos, fue necesario el desarrollo de funcionalidad específica para el cometido. Los métodos que se ocupan de ello son *addTagsView*, que separa las etiquetas del nombre del documento y las añade a la tabla de etiquetas si no existen en la misma (este procedimiento fue desarrollado posteriormente y será explicado en puntos siguientes) y *assignTagsView* que emplea la misma operativa que

el anterior método pero en este caso se realiza la asociación etiqueta-documento en la base de datos (al igual que el anterior, esta función será detallada en puntos posteriores).

Edición de documentos

Para la edición de objetos *Document* seguiremos el procedimiento habitual para este tipo de secciones.

Formulario para edición de documentos

En este caso emplearemos un formulario distinto al de creación de documentos puesto que no permitiremos la modificación del fichero de texto asociado a la instancia. En este caso el formulario de Django será denominado ***EditDocumentForm*** y constará de los mismos atributos que los mencionados con anterioridad salvo el fichero de texto.

Controlador para edición de documentos

Como es habitual en este tipo de funcionalidades, se devuelven los datos del registro al cliente y se actualizan en caso de que el usuario haya modificado datos y hecho la confirmación de envío.

Vista de edición de documentos

Se carga el formulario en la plantilla y se incorporan los botones para confirmación de envío y cancelación de procedimiento.

Eliminación de documentos

Una vez más, el software asociado a esta funcionalidad es en gran medida similar al de anteriores secciones. En este caso, pueden aparecer dos vistas a la hora de eliminar un documento: una que nos advierte que el documento va a desvincularse (lo cual quiere decir que permanece aún ligado a al menos una colección más) y otro que nos avisará de que el documento será borrado definitivamente de la plataforma al sólo estar asociado al presente corpus.

Controlador para eliminación de documentos

Se comprueba el número de asociaciones del presente objeto con las colecciones presentes en la plataforma. En caso de ser más de una, se carga la plantilla correspondiente y se procede a la desvinculación del documento cuando se confirme la acción, en caso contrario, se carga una plantilla distinta y se realiza el desecho total del objeto y documento de texto del sistema.

Vistas de eliminación de documentos

En ambos casos se establecen los botones que vienen siendo habituales de confirmación y cancelación de operación. Lo único que cambiará entre una opción u otra será el texto de aviso en cada caso.

Funcionalidad para normalización de ficheros a UTF-8

Para poder trabajar con los ficheros de texto, es necesario normalizarlos a una codificación común. En nuestro caso emplearemos UTF-8. Antes que nada, es necesario saber en qué codificación está escrito el documento a subir y para ello empleamos el ya mencionado campo de codificación del fichero en el que el usuario deberá indicar la codificación que posee el archivo o conjunto de archivos que se desean incluir en la plataforma. En caso de que la codificación seleccionada no sea la correcta o que el sistema no pueda decodificar y codificar correctamente el documento, así se hará saber al usuario, indicándole que reintente la carga del fichero. La función que realiza este cometido, consiste en la decodificación desde la codificación seleccionada en el formulario y la codificación en UTF-8.

Asignación de ficheros a colecciones

Se pretende que un documento pueda estar vinculado a varias colecciones a la vez por lo que se realizó una utilidad para ello. En el caso de Django, las tablas intermedias que conectan dos independientes como en este caso serían las tablas de colecciones y documentos, son creadas como campos de modelos, que serán conectados a la clase que se indique en la declaración, denominados *ManyToManyField*. Automáticamente, Django crea en la base de datos una tabla intermedia compuesta por identificador único de registro y dos campos más, que corresponde a los IDs de los elementos conectados.

Controlador de asignación de ficheros a colecciones

En primer lugar se hace una consulta a la base de datos para obtener los documentos que no se hallen asignados a la presente colección y se renderiza una pequeña tabla con ellos. Una vez el usuario haya marcado los elementos que desee incorporar al corpus en cuestión y confirme la operación pulsando el botón que se encargue de ello, el servidor añadirá una relación a la tabla intermedia entre colección y documentos por cada documento marcado en el formulario de asignación. Una vez finalice la operación, volverá a aparecer la misma vista, sin embargo, esta vez sin los elementos que ya se vincularon a la colección.

Vista de asignación de ficheros a colecciones

Crearemos un formulario personalizado de HTML que consistirá en enviar al servidor los identificadores de los documentos marcados como 'a vincular' a la colección con la que nos encontremos trabajando. Mostraremos en la tabla las casillas para marcaje junto con los documentos que pueden ser vinculados provenientes de la lista de los mismos, recorrida mediante un ciclo *for* de Django, generada por la consulta a la base de datos realizada anteriormente. Para finalizar, como siempre, añadimos los botones para confirmación o denegación de operación

Funcionalidad de subida de ficheros ZIP con diversos documentos contenidos en ellos

Se desarrolló una utilidad para que ficheros comprimidos ZIP de igual modo pudieran ser subidos a la plataforma. Para ello, el software que lo hace posible se dividió en diversas funciones en el fichero **views.py** y se realizó una vista de resumen de importación de los mismos a la plataforma. Descripción detallada del trabajo, a continuación.

Controlador para subida de ficheros ZIP

La operativa general es muy similar a la subida de un fichero de texto individual, tanto para darles nombre a los ficheros del archivo, como para crear sus etiquetas o codificarlos. Sin embargo hay ciertas consideraciones y matices a tener en cuenta:

- Se emplea la librería de Python **zipfile** para trabajo con ZIPs y descompresión.
- Una vez descomprimido el fichero ZIP, recorreremos uno a uno los textos contenidos en el mismo y realizando las operaciones pertinentes sobre cada uno de ellos:
 - Decodificación y codificación: mismo procedimiento que para ficheros individuales sólo que esta vez se crean dos listas de Python en la función, donde se introducirán en cada una de ellas los elementos que pudieron ser importados y los que no para la vista resumen posterior.
 - Etiquetas: misma operativa que para ficheros únicos (también se incluyen las etiquetas del nombre del fichero ZIP si así lo indica el usuario, además de las que pueda poseer por separado cada fichero).
 - En caso de ser importado el documento, la instancia se guarda de la misma manera que un documento individual.
- Una vez finalizado el procesamiento de los documentos, obtendremos una vista resumen en la que se nos indicará qué ficheros pudieron ser subidos a la plataforma y los que

no, pidiendo al usuario que reintente cargar los que obtuvieron un resultado negativo, debido a que poseían una codificación distinta a la que se indicó en un principio.

Vista de resumen de importación del fichero ZIP

Creamos dos pequeñas tablas independientes para mostrar los documentos que pudieron ser importados y los que no. Los documentos que van a una u otra son los que se hayan en las listas mencionadas anteriormente que serán recorridas en la plantilla para plasmar los datos. Creamos para finalizar un botón para continuar navegando por la plataforma.

6.4.5. Reestructuración del apartado para mejora de experiencia de usuario

Se optó por realizar una reestructuración del apartado de colecciones y documentos. En primera instancia, las colecciones de la vista de corpus textuales no poseían vista personalizada de documentos vinculados a las mismas, sino que todos los documentos estaban agrupados en una vista única en la que se indicaba en la tabla de registros a qué colección o colecciones pertenecían cada uno de ellos. Si se quería crear un nuevo documento, para realizar la conexión con una colección, se debía seleccionar en el formulario a la que se deseaba que perteneciera. Se optó por una opción más intuitiva en la que cada colección tenía en su haber una vista de documentos en la que sólo aparecerían los ligados a la misma y si se quería crear uno que estuviera por defecto vinculado a la colección en la que se hallase el usuario.

6.4.6. Mejoras en el apartado

Al mismo tiempo se optó por realizar una serie de mejoras de experiencia de usuario, de tal forma que se le ahorrara tiempo en ciertas operaciones como la eliminación de varios documentos de texto a la vez, al ser demasiado tediosa la eliminación uno a uno si se desean desechar numerosos de una sola vez.

Eliminación múltiple de documentos

Se introdujo esta utilidad de tal manera que pudieran eliminarse o desvincularse diversos ficheros de texto al mismo tiempo. Para ello se introdujeron cajas seleccionables en la tabla de documentos de cada colección y un botón para confirmar las mismas que llevase a una vista de resumen previa a la eliminación definitiva de los registros de la plataforma. A continuación se detalla el código empleado para la construcción de la funcionalidad.

Controlador para eliminación múltiple de documentos

En primer lugar se sigue un procedimiento parecido al utilizado en la vista resumen de importación para archivos ZIP. Se separarán las selecciones en dos tablas, una para documentos relacionados con más de una colección y otra para ficheros que sólo pertenezcan al presente corpus. En caso de confirmar las selecciones, para el primer caso tan sólo se realizará una desvinculación del documento de la colección, sin llegar a eliminar el registro definitivamente de la base de datos. Para el segundo caso sí se ejecutará un desecho definitivo del objeto. El rellenado de las tablas se hará mediante listas. Para separar en las estructuras de datos los elementos con una o múltiples relaciones, realizaremos consultas a la base de datos y contaremos cuántas ocurrencias en la tabla intermedia hay de cada documento seleccionado para así en función del resultado introducir el elemento en una lista u otra. Una vez se confirme la eliminación o desvinculación de los documentos, el sistema desvinculará o eliminará completamente los objetos en función de la lista en la que estuvieran.

Vista para eliminación múltiple de documentos

Al igual que en la vista resumen de importación, creamos dos pequeñas tablas en las que se agruparán los documentos, cada una con un mensaje de aviso sobre la situación de los elementos de cada estructura. Para cargar las tablas importamos mediante contexto las mencionadas listas y las recorremos para rellenar la información. Por último creamos los botones para la confirmación o supresión de la operación.

Corrección de errores y mejoras

Por último, se solucionaron problemas relativos a las etiquetas, software de tablas de datos, software de codificación, asignación de documentos, comprensión de los campos de tipo *ManyToMany* y arreglo de otros errores en el software.

Nueva funcionalidad para etiquetas

En esta sección también se realizó la modificación en el software de etiquetas para que pudieran ser importadas desde el nombre de los ficheros. Como se explicó en secciones anteriores, dados los problemas que generaba la transformación de una lista de cadenas de caracteres a una serie de etiquetas, se tuvo que optar por realizar consultas en código SQL en el modelo correspondiente a los documentos, a la base de datos empleando la funcionalidad *cursor* de Django. Esta operación era llamada desde el controlador, enviando una a una las etiquetas extraídas a ser procesadas.

6.5. Iteración 6: implementación de vista de perfil de usuario

Se decidió que los usuarios pudieran alterar alguna de la información sobre su usuario en la plataforma sin necesidad de recurrir a un administrador para ello. Un usuario puede modificar su nombre, apellidos, correo electrónico, institución y contraseña (de la que hablaremos a continuación). Esta vista funcionará como un formulario de edición de registro (formulario de HTML, no se empleará uno de Django) y la funcionalidad asociada a ella será detallada a continuación.

6.5.1. Controlador para vista de perfil de usuario

Se realiza la consulta a la base de datos para conseguir la información del usuario y así pasarla al lado del cliente para poder renderizar el formulario relleno con los datos actuales de la persona. Se toman precauciones en cuanto a seguridad para que los usuarios sólo puedan acceder a su información y en ningún caso a la del resto de personas.

Vista de perfil de usuario

Se carga el formulario con la información recibida del servidor, se crea el enlace para cambio de contraseña (del que hablaremos en el siguiente punto) y los botones para confirmación o declinación de operación.

6.5.2. Incorporación de funcionalidad para cambio de contraseña (usuario y administrador)

Se desarrolló una utilidad para que el propio usuario pudiera cambiar su contraseña cuando deseara. En caso de que no recordase su clave, un administrador también podría hacerlo desde el panel de gestión de usuarios. Se precisará sobre ambas funcionalidades a continuación.

Cambio de contraseña propio

Para este cometido se emplearán formularios de Django, más concretamente un formulario que viene incluido en el software de Django y que recoge toda la funcionalidad y medidas de seguridad necesarias para una acción tan delicada.

Controlador de cambio de contraseña propio

Se llama al mencionado formulario y una vez el usuario ha rellenado la información correctamente, se actualizan tanto el registro como el *hash* de la contraseña para que no

se cierre la sesión del usuario en la plataforma y se redirige al usuario a su vista de perfil.

Vista de cambio de contraseña propio

Se llama al formulario en la plantilla y se declaran los botones necesarios para confirmación y cancelación de operación.

Cambio de contraseña ajeno

Como se ha comentado con anterioridad, se consideró que un usuario administrador del sistema también tuviera la potestad de cambiar contraseñas de otros usuarios si así lo requieren estos. En este caso se empleó una operativa un punto diferente a la anterior.

Controlador de cambio de contraseña ajena

El funcionamiento es sencillo, el administrador introduce la contraseña en dos ocasiones para asegurar que es correcta y el servidor procesa ambas. Si se confirma la corrección de ambas se actualiza en la base de datos.

Vista de cambio de contraseña ajena

Se crea un formulario HTML simple en el que se introduce la nueva contraseña en dos ocasiones y se crean los botones para navegación, tanto el de cancelación de operación como el de confirmación de operación.

6.6. Iteración 7: implementación de herramientas de PLN para corpus textuales

La séptima y última iteración consistió en el desarrollo de la librería de análisis textual para la plataforma, consistiendo esta en cuatro herramientas principales: Wordlist, N-Grams, Concordance y Concordance Plot, las cuales serán detalladas en los puntos siguientes.

6.6.1. Realización de storyboard, sección de análisis de texto

Se realizó la parte restante del storyboard correspondiente a la sección de análisis textual, utilizando la misma herramienta y prototipo que el que ya se construyó con anterioridad y a continuación se presenta el resultado final del prototipo.

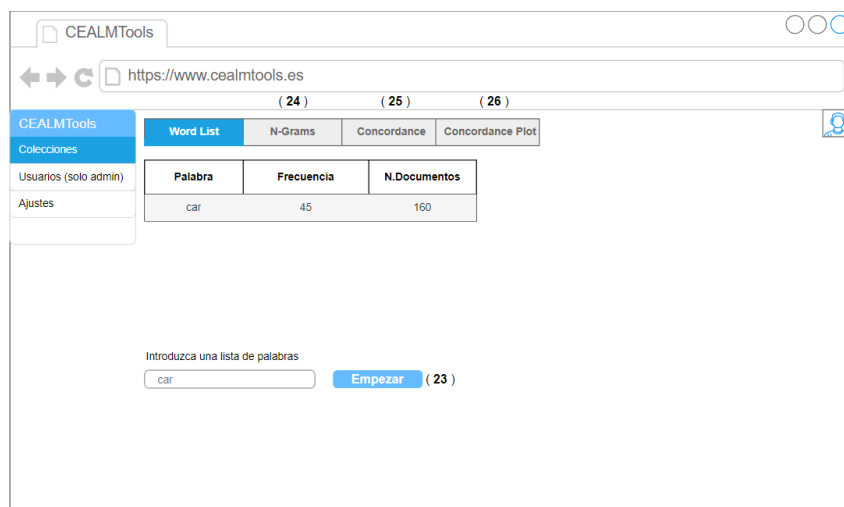


Figura 6.1: Escena 23: herramienta 'Wordlist'

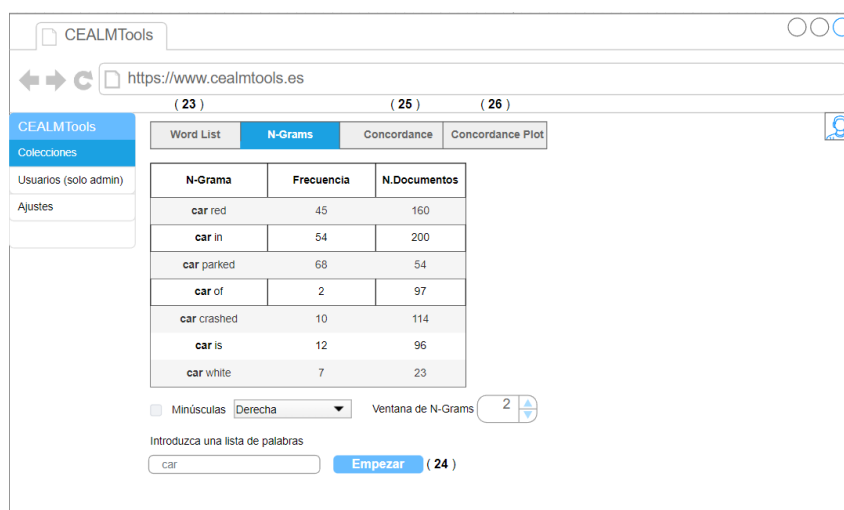


Figura 6.2: Escena 24: herramienta 'N-Grams'



Figura 6.3: Escena 25: herramienta 'Concordance'

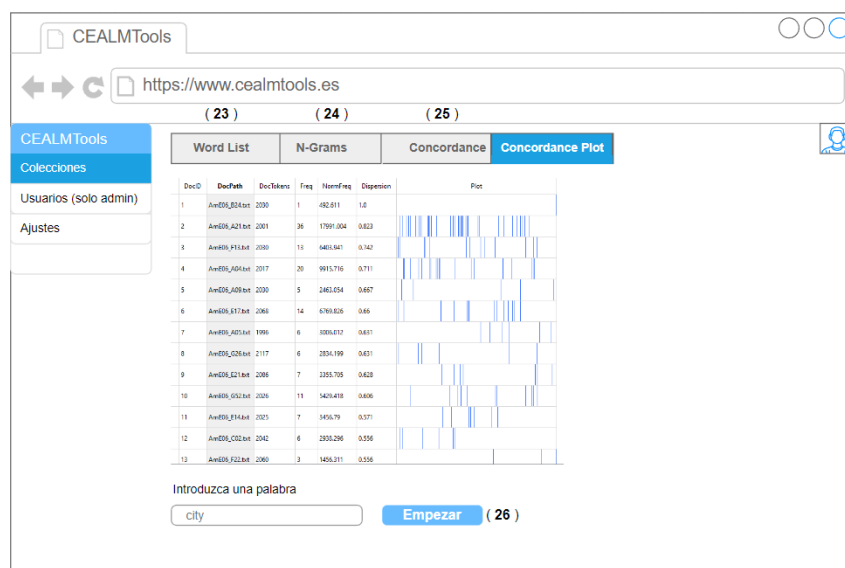


Figura 6.4: Escena 26: herramienta 'Concordance Plot'

6.6.2. Realización de funcionalidad 'Wordlist'

La primera de las herramientas desarrolladas fue la más simple, 'Wordlist' que consiste en realizar un conteo de palabras coincidentes con la solicitada a lo largo de una serie de textos pertenecientes a un corpus. Es posible incorporar una serie de restricciones, como que se

transformen todas las palabras de los textos a minúsculas o que se eliminen las palabras vacías de los mismos. Se precisa sobre el algoritmo y vista de la plataforma para esta herramienta seguidamente.

Controlador de 'Wordlist'

En resumidas cuentas, llamaremos desde el controlador al algoritmo que realiza el análisis de los textos con los parámetros seleccionados por el usuario en la vista. Crearemos una lista para guardar los resultados como objetos 'Wordlist', clase de la que hablaremos en el siguiente punto y una vez el algoritmo nos devuelva el procesamiento realizado, enviaremos la lista al lado del cliente para mostrar los datos recopilados.

Implementación de funcionalidad 'Wordlist'

En primer lugar, como estipulamos en el punto anterior, se creó una clase de objetos denominada con el mismo nombre de la herramienta para la creación de instancias de tal forma que fuera más sencillo mostrar datos en la plantilla de la funcionalidad. El algoritmo filtra los documentos pertenecientes a la colección con la que se esté trabajando y procede a la creación del *tokenizador*. En las opciones de texto, para el caso de la transformación de los términos a minúsculas, se emplea el método `.lower()` de Python y para la eliminación de palabras vacías se filtran las coincidencias del texto con los términos existentes en la lista de palabras vacías de NLTK. Se recorren los ficheros, se *tokenizan* y en caso de existir coincidencia, se anota para el conteo de ocurrencias por fichero. Una vez terminado el recorrido, se crea un diccionario con frecuencias de palabras (diccionario *Counter*) para facilitar el conteo de las mismas. Si se empleó un comodín se realiza el procesamiento para comodines, que se explicará en las posteriores secciones, y se filtran las ocurrencias repetidas. Volvemos a anotar las ocurrencias por fichero y creamos un objeto *Wordlist* para cada coincidencia encontrada, siendo éste formado por la palabra, el número de veces que aparece en los textos (extrayendo el dato del diccionario *Counter*) y el número de ocurrencias por fichero anotado a lo largo del procesamiento, introduciendo estos en la lista que enviaremos al cliente. Si lo que se buscó fue una palabra sin comodines tan sólo creamos el objeto y lo insertamos en la lista. En caso de que no se haya introducido nada en la caja de texto, se realizará una búsqueda con el comodín '*' siendo coincidencia por tanto, todas las palabras de los textos. Devolveremos la lista al cliente.

Implementación de vista 'Wordlist'

En general, las vistas pertenecientes a las herramientas de análisis textual en la plataforma constarán de un esqueleto básico: un menú en la parte superior de la ventana para navegar a través de las distintas funcionalidades desarrolladas, una tabla para mostrar los datos recogidos

sobre una palabra en concreto, las distintas opciones o restricciones que podemos incorporar a una búsqueda y la caja de texto para escribir la consulta que deseamos realizar. Las consultas se enviarán al lado del servidor como formularios para el procesamiento de los algoritmos de las herramientas. En este caso, la tabla mostrará para cada registro de la misma la palabra coincidente con la búsqueda (es posible realizar una consulta con comodines), la frecuencia de la misma a lo largo de los textos y el número de archivos en los que aparece. Las opciones que es posible seleccionar ya las comentamos con anterioridad y serán representadas como cajas seleccionables de HTML. La consulta también mencionamos que sería en un cuadro de texto y finalmente se establece el botón de envío de formulario para el procesamiento en el servidor.

6.6.3. Realización de funcionalidad 'N-Grams'

La segunda de las herramientas consiste en la recopilación de términos coincidentes con la búsqueda y la muestra de las ocurrencias junto con la palabra o palabras que la acompañen, ya sea por la derecha o por la izquierda. Las opciones presentes para esta funcionalidad consisten en la transformación de las palabras del texto a minúsculas y eliminación de palabras vacías de nuevo, así como dirección de la búsqueda (si las palabras acompañantes se desea que aparezcan por la derecha o por la izquierda) y tamaño de ventana (número de palabras que acompañarán a la requerida).

Controlador de 'N-Grams'

El controlador para la vista de la funcionalidad N-Grams sigue la misma filosofía que la que se utilizó en 'Wordlist', llamando al algoritmo con los parámetros de las opciones que se mencionaron anteriormente, sólo con la restricción de que la ventana de palabras no puede ser mayor a 5. Obtendremos una lista de objetos que plasmaremos en la tabla de la plantilla posteriormente.

Implementación de funcionalidad 'N-Grams'

En primer lugar creamos la clase para objetos 'N-Grams' que posee el *n-grama* en cuestión, la frecuencia del mismo a lo largo de los textos y la frecuencia por texto de cada uno (número de documentos en los que aparece un *n-grama* en concreto). La parte inicial del algoritmo es idéntica a la del utilizado para 'Wordlist', se filtran los textos del corpus y se crean las estructuras y *tokenizador*. En caso de haberse realizado una búsqueda con comodines, se realiza el filtrado de ocurrencias, se eliminan las repetidas de la lista resultante y se recorren con un bucle. Para cada una, en caso de ser encontrada en el texto, dependiendo de la dirección seleccionada, así escogemos las palabras que acompañan a la ocurrencia. En Python, esta operación es sencilla de realizar mediante selección de índices en una lista. Para el caso de la

ventana por la derecha, la sentencia a utilizar sería: `lista[índice + 1][:tamañoVentana - 1]`, siendo el primer corchete 'desde' y el segundo 'hasta'. Para el caso de una ventana por la izquierda el código empleado consistió en `lista[índice - tamañoVentana + 1 : índice]` siendo la parte izquierda de los dos puntos el 'desde' y la parte derecha el 'hasta'. Iremos guardando los *n-gramas* generados para cada fichero en un diccionario para poder acceder posteriormente y así obtener el número de ficheros en los que aparecieron cada uno de los obtenidos. En caso de que no hayamos realizado la búsqueda con comodines, simplemente buscaremos ocurrencias en los textos con la palabra solicitada. De la misma manera que en 'Wordlist', creamos un diccionario *Counter* para la frecuencia de *n-gramas* y recorremos la lista de resultados para elaborar los objetos con los datos recopilados y enviarlos al controlador para su posterior exposición en la vista.

Implementación de vista 'N-Grams'

La vista en términos generales es idéntica a la de 'Wordlist', incorpora el menú de herramientas, la tabla de resultados y las opciones respectivas a N-Grams, además del cuadro de texto para la consulta y el botón para envío del formulario.

6.6.4. Reconstrucción de sistema de análisis para comodines

Se optó por una reconstrucción del algoritmo para tratado de términos con comodines debido a un funcionamiento errático de la idea anterior que consistía en el uso del módulo Python *fnmatch*. Esta vez se optó por realizar una función que emplease expresiones regulares para tratar los caracteres conocidos como comodines, como son el símbolo de cierre de interrogación o el asterisco. En primer lugar se escapan los caracteres especiales empleando la operación para expresiones regulares de Python `re.escape()`. En función de los patrones obtenidos en la expresión resultante tras ser escapada, sustituiremos estos patrones por expresiones regulares. Si se encuentra un patrón `\*` querrá decir que el usuario escribió un asterisco, por lo que lo sustituiremos por `.+`, mientras que si se encuentra `\?` querrá decir que se escribió un símbolo de cierre de interrogación por lo que sustituiremos el patrón por `.`. Compilamos la expresión regular resultante y la guardamos en una variable para finalmente filtrar las coincidencias que existan con el resultado, ayudándonos de la operación `.fullmatch()` la cual compara cadenas de caracteres en la lista de *tokens* que obtuvimos tras la *tokenización* del texto con el que se estaba trabajando, con expresiones regulares y devuelve el objeto coincidente si existe o *None* en caso contrario, generando como resultado final, una lista de coincidencias de palabras con la expresión regular que se demandó.

6.6.5. Realización de funcionalidad 'Concordance'

La tercera herramienta desarrollada consiste en mostrar el contexto en el que se haya una palabra que busquemos. Las opciones incorporadas a esta funcionalidad consisten en, como viene siendo habitual, la transformación del texto a minúsculas, eliminación de palabras vacías y dos novedades, la dirección de la ordenación, lo cual coloreará las 3 primeras palabras por la izquierda o derecha (en función de la opción seleccionada) del contexto de la búsqueda y el tamaño de la ventana, que indica el número de palabras que se mostrarán en el contexto a derechas y a izquierdas de la palabra.

Controlador de 'Concordance'

Siguiendo una metodología en gran medida similar a la empleada para las 2 herramientas ya tratadas, en el controlador obtenemos los parámetros enviados por el usuario en el formulario de la utilidad y para el tamaño de la ventana establecemos los límites superiores e inferiores para la envergadura de la misma. Los parámetros obtenidos son enviados al algoritmo desarrollado para 'Concordance' y se trabaja con ellos. Una vez se tenga un resultado, se vuelve a remitir al lado del cliente para que se plasmen los datos en la tabla de la utilidad.

Implementación de funcionalidad 'Concordance'

La clase de objetos para la funcionalidad 'Concordance' poseerá los atributos 'doc' que hace referencia al nombre del fichero donde tuvo lugar la ocurrencia, contexto por la derecha e izquierda, que serán dos diccionarios de palabras (clave) y color a utilizar (valor) para su posterior coloreado a la hora de la representación en la tabla para el contexto a cada lado del término y finalmente, el término en sí. Para el algoritmo en primer lugar filtramos los documentos en la base de datos que correspondan a la colección en la que se trabaja y se crea el *tokenizador*. Para obtener el contexto de la palabra nos apoyamos en la utilidad `concordance_list()` de NLTK que devuelve una lista de listas de concordancias de la palabra indicada que se hayan encontrado a lo largo del texto, donde la primera posición de cada sublista será el contexto por la izquierda, la segunda la palabra y la tercera el contexto por la derecha. Para realizar el procesamiento, recorreremos la lista de listas y vamos obteniendo los contextos y quedándonos sólo con el número de palabras de cada uno que se indicasen en el formulario enviado por el usuario. Dependiendo de si la dirección escogida fue la izquierda o la derecha procederemos a colorear las 3 primeras palabras (en caso de que existan) de la ventana correspondiente. Para ello emplearemos un diccionario para cada dirección generados a partir de las listas de palabras de los contextos a ambos lados del término. El valor de las claves será un número, si es 0, la palabra no se colorea, si es 1, 2 ó 3, sí será coloreada a la hora de la creación de la vista. Finalmente creamos el objeto 'Concordance' con los parámetros

mencionados al principio del punto. En caso de que el usuario haya empleado comodines en la consulta, el proceso será el mismo, sólo que se iterará por una lista de coincidencias con la expresión solicitada realizando el proceso mencionado anteriormente.

Implementación de vista 'Concordance'

Para la vista de la herramienta seguiremos las mismas pautas hasta ahora. Sin embargo, para el coloreado de las palabras deberemos recorrer los diccionarios y aplicar color a los términos que vengan marcados del servidor. Por lo demás, seguiremos representando el menú de herramientas, la tabla de resultados, las opciones de la funcionalidad, el cuadro de texto para la consulta y el botón de envío de la misma.

6.6.6. Realización de funcionalidad 'Concordance Plot'

Por último, la última herramienta consistirá en la creación de un diagrama de dispersión por fichero (por palabra y fichero en caso de que se empleen comodines) que escenifique el número de ocurrencias y la posición de las mismas a lo largo de cada texto, de tal manera que si existen numerosas líneas en un diagrama, querrá decir que la palabra aparece en cuantiosas ocasiones y por el contrario si el número de líneas es pobre, significará que no es una palabra que aparezca frecuentemente en el texto. La izquierda del diagrama significa el principio del texto y la derecha el final. Las opciones para esta funcionalidad vuelven a ser las mismas que las empleadas en 'Wordlist' pudiendo transformar el texto a minúsculas y eliminar las palabras vacías.

Controlador de 'Concordance Plot'

Se desarrolló un controlador simple para 'Concordance Plot', siendo extremadamente similar de nuevo al de 'Wordlist', obteniendo los parámetros enviados en el formulario por el usuario y pasándolos al algoritmo de la herramienta.

Implementación de funcionalidad 'Concordance Plot'

En primer lugar, los objetos 'Concordance Plot' poseerán los atributos correspondientes al nombre del documento al que pertenece la palabra, la palabra en sí, el número de *tokens* del texto, la frecuencia de aparición del vocablo en cuestión, la frecuencia normalizada del mismo (número de apariciones de la palabra entre el número total de *tokens* del documento, siendo el resultado multiplicado por 100 para obtener un porcentaje), y los parámetros para la construcción del diagrama mediante Javascript, que consistirá en una lista binaria de ceros y unos y explicaremos seguidamente. El algoritmo comenzará como el resto, filtrando los documentos pertenecientes al corpus y creando el *tokenizador*. Una vez hecho esto, el

funcionamiento consistirá en recorrer los *tokens* obtenidos del texto y comprobar si coinciden con la palabra solicitada. En caso afirmativo, guardaremos en la lista binaria un 1, en caso negativo un 0. Crearemos el objeto 'Concordance Plot' y para la frecuencia generaremos un diccionario de tipo *Counter* al que le pasaremos la lista de *tokens*, para la frecuencia normalizada, realizamos la operación explicada con anterioridad, para el número de *tokens* emplearemos el número de elementos de la lista de los mismos, la palabra será la palabra y el nombre del fichero podemos obtenerlo directamente utilizando la variable del bucle que recorre la lista de textos. Una vez generados todos los objetos, enviamos la lista de los mismos al cliente y una vez allí, realizaremos el volcado de los datos a la tabla y el dibujo de los diagramas.

Implementación de vista de 'Concordance Plot'

La distribución de la vista sigue las pautas marcadas por las anteriores, estando presente el menú de herramientas, la tabla de resultados, las opciones correspondientes de la herramienta y el cuadro de texto junto con su botón de confirmación para la consulta. El trazado de los diagramas de dispersión se realiza mediante Javascript como se indicó anteriormente. El algoritmo sólo trazará una línea en caso de que el elemento de la lista sea un 1. Se crea el gráfico y se calcula la posición de la línea en el diagrama. Para ello, multiplicaremos el índice del bucle *for* por 400, que es el ancho del diagrama y dividiremos el resultado entre el tamaño del vector, a modo de regla de tres. Se crea la línea en formato SVG y se le da la posición 'X' calculada, la posición 'Y' inicial será 0 y la final siempre 80, de tal manera que se trace una línea desde abajo hasta el alto del gráfico. El color escogido será el azul y el ancho 1. Se incorpora el nuevo hijo al diagrama y se continúa con los demás elementos del *array* hasta el final. De esta manera obtenemos un diagrama para cada elemento del vector de objetos 'Concordance Plot'.

6.6.7. Mejoras y corrección de errores

Por último, se hizo una revisión de la plataforma, corrigiendo errores menores, modificando colores de botones e indicaciones en ellos y realizando mejoras mediante Javascript a los formularios personalizados de HTML, entre otros ajustes.

Capítulo 7

Librería de analítica de corpus textuales

En este capítulo se explicará en qué consisten cada una de las herramientas que componen la librería de analítica textual. Puesto que la definición técnica ya se realizó en el apartado anterior, en esta ocasión nos centraremos en una descripción más general sobre las funcionalidades incorporadas.

7.1. Consideraciones previas

Antes de introducir las técnicas incorporadas a la plataforma, es necesario exponer ciertas decisiones tomadas a la hora del análisis del texto presente en los documentos ubicados en la plataforma:

- **Tokenizador**: el tokenizador que se emplea para todas las herramientas construidas fue creado a partir de la clase *RegexTokenizer* de NLTK utilizando expresiones regulares y extrae tan sólo las palabras que estén presentes en el texto, es decir, ignora signos de puntuación, tabuladores, saltos de línea etc. Asimismo, el *tokenizador* tampoco separa las contracciones del texto, por lo que palabras contraídas en inglés como *can't* serán tomadas como un término completo y no se realizará una separación en la que la parte izquierda de la contracción sea un vocablo y la derecha otro o incluso la parte derecha se tome incluyendo el apóstrofe de la contracción. De esta manera es inequívoco el proceso de *tokenizado* en base a nuestras necesidades.
- El **procedimiento general** consiste en:
 - Localizar en la base de datos los ficheros que formen parte de la colección que se quiere analizar.

- Extraer, uno por uno, el texto de cada documento.
 - *Tokenizar* el texto empleando el *tokenizador* del que se habla en el punto anterior.
 - Realizar el análisis pertinente a la herramienta que se esté empleando.
- Existen ciertos **parámetros** que el usuario puede seleccionar:
- Eliminación de **palabras vacías**: las palabras vacías no tienen un significado por sí solas, sino que modifican o acompañan a otras y suelen ser artículos, pronombres, preposiciones, etc. Estos términos pueden ser suprimidos si el usuario así lo desea. El procedimiento consiste en acometer un filtrado sobre la lista de palabras, eliminando de la misma las que se encuentran en la de palabras vacías de NLTK.
 - Transformación del texto a **minúsculas**: las letras mayúsculas se suprimen del documento, siendo sustituidas por letras minúsculas gracias a la función `.lower()` de Python.
 - **Dirección de ventana de texto** (N-Grams y Concordance): es posible seleccionar si la palabra o palabras que acompañen a la que se buscó aparecerán por la izquierda o la derecha de la misma. En el caso de Concordance, la dirección de la ventana de texto (contexto en ese caso) coloreará las 3 primeras palabras que acompañen a la búsqueda en función de la dirección elegida.
 - **Tamaño de ventana**:
 - N-Grams: el selector de tamaño de la ventana de N-Grams permite seleccionar el número de palabras que se mostrarán de las que acompañan a la búsqueda.
 - Concordance: cantidad de palabras que aparecen a un lado y a otro de la palabra solicitada.
- **Comodines**: es posible emplear comodines en las palabras que se desean buscar, siendo en este caso `?` o `*`. En caso de que se realice una búsqueda con esta modalidad, en primer lugar se escapan los comodines que existan en la palabra y se sustituyen por expresiones regulares para asegurar una buena traducción. De los *tokens* extraídos del documento, se busca los que coincidan con la palabra y comodines empleados y se trabaja en base a ellos.

7.2. Técnicas PLN aplicables al proyecto

Las funcionalidades que se incorporaron al proyecto consistieron en 4 técnicas PLN diferenciadas: **WordList**, **N-Grams**, **Concordance** y **Concordance Plot** que introduciremos a continuación:

7.2.1. WordList

Esta herramienta tiene un funcionamiento simple y consiste en la búsqueda en el corpus con el que se trabaje en ese instante de la palabra que se desee. Como ya se ha comentado, existen ciertas opciones a la hora del análisis (paso a minúsculas y eliminación de palabras vacías) y dependiendo si la búsqueda incluye un comodín o no, la salida que proporciona la plataforma será distinta (una o varias palabras si cumplen con el patrón del comodín). Los datos que se arrojan sobre el análisis realizado son plasmados en una tabla que incluye los siguientes campos:

- Palabra: término que se buscó en el cuadro de texto o una de las coincidencias que existen si se incluyeron comodines (cada fila de la tabla será una correspondencia de las que se hayan encontrado en los textos y coincidan con el patrón buscado).
- Frecuencia: número de veces que se encontró el término a lo largo de todos los textos de la colección con la que se trabaja.
- Número de documentos: cantidad de documentos en los que se encontró el vocablo en cuestión.

7.2.2. N-Grams

Los N-Grams son secuencias continuas de palabras, símbolos o *tokens* en un documento. En términos técnicos, pueden ser definidos como las secuencias vecinas de ítems en un texto, básicamente un conjunto de palabras contiguas dentro de una ventana. Son empleados en minería de texto y tareas de PLN como la que nos incumbe.

En esta funcionalidad, al introducir una palabra en el cuadro de texto, obtendremos como salida otra tabla con diferentes campos. En función de los parámetros escogidos y si se incluyó o no un comodín en el término, obtendremos una salida u otra. Las columnas que componen la mencionada tabla consisten en:

- N-Grama: se trata de la palabra buscada y las N palabras que la acompañan. Si se eligió una ventana a derechas aparecerán las que se encuentran en esa dirección, en la contraria si se escogió izquierdas. Dependiendo del tamaño de la ventana, así será el número de vocablos que aparezcan junto con el introducido. Cada fila de la tabla es un N-Grama distinto y dependiendo si se introdujo un término con comodines o no, si existen coincidencias, aparecerán los vocablos que cumplan el patrón.
- Frecuencia: número de veces que aparece un N-Grama en concreto a lo largo de todos los textos del corpus que se esté analizando.

- Número de documentos: cantidad de ficheros en los que se encontró un N-Grama concreto.

7.2.3. Concordance

Concordance es una funcionalidad cuyo objetivo consiste en la búsqueda de ocurrencias de una palabra y el contexto en el que se encuentran. Es una herramienta útil si por ejemplo, queremos conocer en qué contexto aparece el nombre de una persona a lo largo de los ficheros de texto de una colección o realizar un análisis sobre las situaciones en las que se utiliza un verbo en concreto. Como es habitual, los resultados aparecen plasmados en una tabla que está compuesta por 3 columnas principales:

- Fichero: fichero en el que apareció la coincidencia.
- Contexto a izquierdas: secuencia de palabras que se encuentran a la izquierda del término buscado.
- Palabra: se trata del vocablo que se buscó a través del cuadro de texto de la herramienta. En caso de realizar una búsqueda con comodines, obtendremos, si existen, entradas en la tabla con palabras coincidentes al patrón.
- Contexto a derechas: secuencia de palabras que se encuentran a la derecha del término buscado.

7.2.4. Concordance Plot

Esta herramienta consiste en la creación de un diagrama de dispersión que represente al texto que se está analizando, donde cada línea coloreada del mismo simboliza una coincidencia con la palabra buscada. La posición de la línea a lo largo del diagrama también manifiesta la ubicación de la misma en el texto (la izquierda representa el principio del texto y la derecha el final). Los resultados del análisis al igual que el resto de funcionalidades, aparecen reflejados en una tabla compuesta por las siguientes columnas:

- Fichero: se trata del fichero en el que se ha realizado el análisis. Cada fila de la tabla representa un documento distinto.
- Palabra: término buscado en el cuadro de texto de la herramienta. Si se emplean comodines obtendremos, si existen, entradas en la tabla con palabras coincidentes al patrón.
- Tokens: representa al número de tokens que contiene el texto en cuestión.

- Frecuencia: frecuencia con la que aparece un vocablo concreto a lo largo del fichero que esté analizándose.
- Frecuencia normalizada: esta estadística representa el porcentaje de aparición de una palabra respecto al número de *tokens* totales del documento.
- Diagrama: gráfico de dispersión que representa las apariciones de un término a lo largo del texto.

Capítulo 8

Conclusiones

En el último apartado de la memoria realizaremos una reflexión sobre los conocimientos adquiridos a través del desarrollo del proyecto, mejoras a realizar en un futuro sobre el mismo y un cierre a la documentación de la plataforma.

8.1. Conclusión final

Al comienzo del proyecto se plantearon una serie de objetivos que procedemos a analizar y reflexionar sobre si han sido cumplidos o no y de qué manera:

- Realizar un estudio de las diferentes técnicas de PLN aplicables a este problema: se considera esta meta como cumplida gracias al análisis realizado de la biblioteca NLTK, entre otras, así como la investigación y desarrollo de las necesidades para análisis textual del proyecto, aplicando y creando el código perteneciente a las técnicas pertinentes.
- Diseñar y desarrollar una librería de analítica de corpus textuales con las distintas funcionalidades: se da como satisfecho el requisito al dar por finalizado el desarrollo de las herramientas que componen la librería que en este caso son Wordlist, N-Grams, Concordance y Concordance Plot, además de la utilización de métodos de tratamiento de información textual, realización de la *tokenización* de los documentos pertenecientes a las colecciones y procesado de expresiones regulares, entre otras técnicas y operativas.
- Diseñar y crear una plataforma web que permita realizar la gestión de usuarios y corpus: de nuevo consideramos este fin como alcanzado dado que la creación y puesta en marcha de la plataforma web ha sido realizada. El sitio posee las funcionalidades previstas de gestión tanto de usuarios como de corpus textuales y ficheros que los componen así como las funcionalidades de Procesamiento de Lenguaje Natural necesarias. Asimismo, se han adquirido competencias relacionadas con el desarrollo web, tanto en el lado del

cliente como del servidor, gestión y administración de una base de datos, así como de un servidor web donde alojar el sistema, desarrollo de software para análisis de corpus textuales, desarrollo de herramientas PLN y lenguajes de programación como Python, HTML, Javascript o SQL.

- Incluir en la plataforma web acceso a la visualización y búsqueda de información en los corpus incorporados: de nuevo se da el objetivo como conseguido gracias a las herramientas de visualización, procesado y búsqueda de información textual desarrolladas para la plataforma, presentes en el apartado correspondiente dentro del sitio web. Éstas nos permiten realizar una consulta al sistema sobre un término o conjunto de términos que el usuario desee y se recibe un resultado o conjunto de resultados, los cuales se presentan a la persona de forma clara y concisa.
- Redactar la presente memoria que recoge todo el trabajo realizado así como los manuales de instalación y usuario: se da la finalidad como cumplida tras la redacción del presente documento, así como de sus anexos, los mencionados manuales de instalación y usuario.

La construcción del sistema SiGADO otorga una potente herramienta en línea a los investigadores en el ámbito del análisis textual, permitiéndoles realizar una gestión intuitiva y cómoda a los mismos de usuarios y colecciones de documentos con los que trabajar. Se ha desarrollado una librería de análisis textual con múltiples utilidades para estudio de corpus de texto que incorpora diferentes opciones como transformación de palabras a minúsculas, entre otras, además del soporte para comodines. Mediante la creación de la plataforma, se han adquirido valiosos conocimientos respectivos al desarrollo de sitios web, más concretamente respectivos al *framework* Django de Python, así como aprendizaje para gestión y administración de una base de datos para un sistema de estas características, diseño de vistas, creación de controladores y modelos o tratamiento de URLs, entre otros. Así mismo, se han adquirido competencias y profundizado en el desarrollo de algoritmos PLN con distintos parámetros para la creación de la librería, realizando estudios sobre las distintas bibliotecas Python para este cometido y proponiendo nuevas funcionalidades para la plataforma. Finalmente, a modo de valoración personal, el Procesamiento de Lenguaje Natural previamente al desarrollo del sistema ya era un ámbito de estudio que consideraba de gran importancia por la cantidad de aplicaciones que posee, por la rapidez de crecimiento de la que goza gracias al auge de la inteligencia artificial y las posibilidades que ofrece a la hora de mejorar la calidad de vida de la ciudadanía. Espero que el proyecto continúe dando visibilidad al estudio y desarrollo de esta rama de la computación puesto que la construcción de este sistema no hace más que reafirmar mis creencias entorno al PLN y puedo dar por seguro que será un área de estudio en la que seguiré trabajando en mi futuro profesional.

8.2. Mejoras futuras

A continuación trataremos las posibles mejoras y nuevas funcionalidades a incorporar al sitio web:

- Recuperación de contraseña por correo electrónico: realización de funcionalidad para recuperación de contraseña vía email, mediante la cual, en caso de que el usuario no recuerde su clave de acceso a la plataforma, esta pueda ser restaurada mediante un enlace que se envíe al correo electrónico del afectado, de la misma manera que se sigue en otras plataformas de la red.
- Incorporación de múltiples herramientas a la librería de análisis textual:
 - Funcionalidades para análisis morfosintáctico: como ya comentamos en el apartado de análisis de software, la biblioteca *spaCy* ofrece utilidades más orientadas al análisis morfosintáctico del texto, por lo que podría ser interesante la incorporación de funcionalidades como conteo de sustantivos, adjetivos, etc, realización de proporciones de unos tipos de palabras respecto de otros o incluso reconocimiento de entidades.
 - Uso de aprendizaje supervisado y no supervisado: es posible el empleo de técnicas de aprendizaje automático para análisis textual, tanto supervisado utilizando máquinas de vectores de soporte para predicción como no supervisado entrenando modelos K-Means mediante vectores de características TF-IDF para asociación de términos en agrupaciones. Empleando estas técnicas, las posibilidades aumentan en gran medida. Para el sitio podemos proponer un clasificador de opiniones para analizar la polaridad de los documentos.
- Desarrollo más profundo del espacio de trabajo: donde se muestren accesos directos a las colecciones más empleadas, herramientas más utilizadas, etc.

Capítulo 9

Bibliografía

- [1] Expert.ai team. NLP for Big Data: What Should Everyone Know? <https://www.expert.ai/blog/nlp-big-data-everyone-know/>, 2016. [Online; accedido en agosto de 2022].
- [2] Gil Allouche. Natural Language Processing and Big Data: A Powerful Combination. <https://www.dataversity.net/natural-language-processing-big-data-powerful-combination/#>, 2014. [Online; accedido en agosto de 2022].
- [3] Carmen Torrijos. Procesamiento del lenguaje natural en el ámbito biomédico. <https://www.iic.uam.es/lasalud/procesamiento-del-lenguaje-natural-ambito-biomedico/>, 2022. [Online; accedido en agosto de 2022].
- [4] ForeSee Medical. Natural language processing in healthcare. <https://www.foreseemed.com/natural-language-processing-in-healthcare>, 2022. [Online; accedido en agosto de 2022].
- [5] Django Software Foundation. Meet Django. <https://www.djangoproject.com/>, Creado en 2005, actualizado en 2022. [Online; accedido en agosto de 2022].
- [6] Contribuidores de Mozilla Developer Network. Introducción a Django. <https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introduction>, Actualizado en 2022. [Online; accedido en agosto de 2022].
- [7] José Domingo Muñoz. Qué es Flask. <https://openwebinars.net/blog/que-es-flask/>, 2017. [Online; accedido en agosto de 2022].

- [8] Rahul Panchal. What are the top 7 frameworks in Python? <https://www.rlogical.com/blog/what-are-the-top-7-frameworks-in-python/>, 2021. [Online; accedido en agosto de 2022].
- [9] Quintagroup. Flask Python framework. <https://quintagroup.com/cms/python/flask>, Creado en 2012, actualizado en 2018. [Online; accedido en agosto de 2022].
- [10] Matthew Deery. What is Flask and how do developers use it? A quick guide. <https://careerfoundry.com/en/blog/web-development/what-is-flask/>, 2022. [Online; accedido en agosto de 2022].
- [11] Uschie Maun. 10 Benefits of Using CodeIgniter for Web Development. <https://www.seasiainfotech.com/blog/10-benefits-of-using-codeigniter-for-web-development/>, Creado en 2019, actualizado en 2022. [Online; accedido en agosto de 2022].
- [12] Chetan Patel. What is CodeIgniter Framework, and How Does It Works? Top Features and Benefits Explained! <https://www.metizsoft.com/blog/codeigniter-framework-features-and-benefits>, 2022. [Online; accedido en agosto de 2022].
- [13] Appyweb. CodeIgniter. <https://www.appyweb.es/diccionario/codeigniter/>, 2020. [Online; accedido en agosto de 2022].
- [14] CodeIgniter Foundation. CodeIgniter Features. <https://www.codeigniter.com/userguide3/overview/features.html>, 2022. [Online; accedido en agosto de 2022].
- [15] Mindfire Solutions. Pros and Cons of CodeIgniter Framework. <https://www.mindfiresolutions.com/blog/2018/03/pros-cons-codeigniter-framework/>, 2018. [Online; accedido en agosto de 2022].
- [16] Kanhasoft. Pros and Cons of most Favoured PHP Framework: CodeIgniter. <https://www.kanhasoft.com/blog/pros-and-cons-of-most-favoured-php-framework-codeigniter-infographic/>, 2018. [Online; accedido en agosto de 2022].
- [17] Berto López. ¿Qué es WordPress, para qué sirve y cómo funciona? <https://www.ciudadano2cero.com/que-es-wordpress/>, 2022. [Online; accedido en agosto de 2022].
- [18] Gustavo B. ¿Qué es WordPress? Revisión del CMS más popular. https://www.hostinger.es/tutoriales/que-es-wordpress#Caracteristicas_de_WordPress, 2022. [Online; accedido en agosto de 2022].

- [19] ArcStone. The Pros and Cons of Choosing WordPress for your Website CMS. <https://www.arcstone.com/blog/pros-and-cons-of-wordpress-cms>, 2020. [Online; accedido en agosto de 2022].
- [20] Roger James. Top Web Development Frameworks for 2022 Comparison: All You Need To Know. https://www.pixelcrayons.com/blog/best-web-development-frameworks-comparison/#4_Django, 2022. [Online; accedido en agosto de 2022].
- [21] Linas Kiguolis. The 5 Absolute Best Backend Web Frameworks in 2022. <https://codeornocode.com/software-development/best-backend-web-frameworks-2022/>, 2022. [Online; accedido en agosto de 2022].
- [22] Tomasz Bąk. Top Python NLP Libraries: Features, Use Cases, Pros and Cons. <https://www.softkraft.co/python-nlp-libraries-features-us-cases-pros-and-cons/>, 2022. [Online; accedido en agosto de 2022].
- [23] Kechit Goyal. Top 7 Python NLP Libraries [And Their Applications in 2022]. <https://www.upgrad.com/blog/python-nlp-libraries-and-applications/>, 2021. [Online; accedido en agosto de 2022].
- [24] Dominik Kozaczko. 8 best Python Natural Language Processing (NLP) libraries. <https://sunscrapers.com/blog/8-best-python-natural-language-processing-nlp/>, 2018. [Online; accedido en agosto de 2022].
- [25] Tutorialspoint. OpenNLP - Overview. https://www.tutorialspoint.com/opennlp/opennlp_overview.htm, 2022. [Online; accedido en agosto de 2022].
- [26] Predictive Analytics Today. Apache OpenNLP. <https://www.predictiveanalyticstoday.com/apache-opennlp/>, 2022. [Online; accedido en agosto de 2022].
- [27] SQLite. Features Of SQLite. <https://www.sqlite.org/draft/features.html>, 2022. [Online; accedido en agosto de 2022].
- [28] Davis E. King y contribuidores. MITIE: MIT Information Extraction. <https://www.linuxlinks.com/mitie-mit-information-extraction/>, Publicado en 2019, actualizado en 2022. [Online; accedido en agosto de 2022].

- [29] JavaTpoint. SQLite Features/ Why to use SQLite. <https://www.javatpoint.com/sqlite-features>, 2021. [Online; accedido en agosto de 2022].
- [30] Niamh Reed. SQLite pros and cons: a short overview. <https://www.thinkautomation.com/our-two-cents/sqlite-pros-and-cons-a/>, 2022. [Online; accedido en agosto de 2022].
- [31] keycdn Proinity LLC. SQLite vs MySQL - Comparing 2 Popular Databases. <https://www.keycdn.com/support/sqlite-vs-mysql/>, 2018. [Online; accedido en agosto de 2022].
- [32] DbForgeTeam. What is MySQL: The Essence and the Features. <https://blog.devart.com/what-is-mysql-the-essence-and-the-features.html>, 2021. [Online; accedido en agosto de 2022].
- [33] Param Chahal. What is MySQL? Definition, Features, Explanation. <https://blog.templatetoaster.com/what-is-mysql/>, 2019. [Online; accedido en agosto de 2022].
- [34] TechStrikers Taygete Software Solutions Inc. MySQL Advantages and Disadvantages. <https://www.techstrikers.com/MySQL/advantages-and-disadvantages-of-mysql.php>, 2022. [Online; accedido en agosto de 2022].
- [35] Luis Olave Alarcón. Stopwords. <https://luisolavea.xyz/stopwords/>, Publicado en 2017, actualizado en 2021. [Online; accedido en agosto de 2022].
- [36] Andrew Pomponio. MariaDB Overview: Key Features, Benefits, and FAQ. <https://www.openlogic.com/blog/mariadb-overview>, 2021. [Online; accedido en agosto de 2022].
- [37] Richard Peterson. MariaDB vs MySQL: What is the Difference Between MariaDB and MySQL. <https://www.guru99.com/mariadb-vs-mysql.html>, 2022. [Online; accedido en agosto de 2022].
- [38] Hitesh Jethva. MariaDB vs MySQL – What's the Difference? (Pros and Cons). <https://cloudinfrastructureservices.co.uk/mariadb-vs-mysql/>, 2021. [Online; accedido en agosto de 2022].
- [39] Jordana Alexandra. MariaDB vs MySQL – Key Differences, Pros and Cons, and More. <https://www.hostinger.com/tutorials/mariadb-vs-mysql>, 2022. [Online; accedido en agosto de 2022].

- [40] Nithyashree V. What Are n-grams and How to Implement Them in Python? <https://www.analyticsvidhya.com/blog/2021/09/what-are-n-grams-and-how-to-implement-them-in-python/>, 2021. [Online; accedido en agosto de 2022].
- [41] Kavita Ganesan. What are N-Grams? <https://kavita-ganesan.com/what-are-n-grams/#.Yw5b43ZByUn>, Publicado en 2018, actualizado en 2022. [Online; accedido en agosto de 2022].
- [42] Afham Fardeen. 9 Cool NLTK Functions You Did Not Know Exist. <https://machinelearningknowledge.ai/cool-nltk-functions-you-did-not-know-exist/>, 2021. [Online; accedido en agosto de 2022].
- [43] Zen Flowchart. Diagrams.net Flowchart Maker - Product Review. <https://www.zenflowchart.com/blog/diagrams-net-flowchart-maker-product-review>, 2022. [Online; accedido en agosto de 2022].
- [44] Diego Guzmán. Python: ¿qué es un decorador? <https://codingornot.com/python-que-es-un-decorador>, 2018. [Online; accedido en agosto de 2022].
- [45] Wikipedia. Comparison of server-side web frameworks. https://en.wikipedia.org/wiki/Comparison_of_server-side_web_frameworks, 2022 (última edición). [Online; accedido en septiembre de 2022].
- [46] Ministerio de Empleo y Seguridad Social. Resolución de 22 de febrero de 2018, de la Dirección General de Empleo, por la que se registra y publica el XVII Convenio colectivo estatal de empresas de consultoría y estudios de mercado y de la opinión pública. <https://www.boe.es/boe/dias/2018/03/06/pdfs/B0E-A-2018-3156.pdf>, 2018. [Online; accedido en septiembre de 2022].

Apéndice A

Manual de instalación

A continuación se presenta el manual de instalación en servidor de la plataforma, paso a paso:

- **Descarga del proyecto desde el repositorio**

- `git clone https://gitlab.ujaen.es/mcdiaz/CEALMTools`

- **Creación del entorno Python**

- `sudo apt install python3`
- `sudo apt install python3-pip`
- `pip3 install virtualenv`
- `python3.9 -m venv env`
- `source env/bin/activate`
- `pip install -upgrade pip`
- `cd cealmtools`
- `pip install -r requirements.txt`

- **Preparación de Django**

- `cd cealmtools`
- `cp settings.py.bak settings.py`
- Una vez nos encontremos en el fichero *settings* cambiaremos distintos parámetros:
 - En primer lugar el `SECRET_KEY`, borramos las 'x' y ponemos una clave.
 - En el parámetro `'ALLOWED_HOSTS'`, pondremos `'0.0.0.0'` y `'localhost'`, separados ambos por una coma.

- En la variable DATABASES modificaremos los campos:
 - ◇ 'NAME': el nombre de la base de datos.
 - ◇ 'USER': el nombre del usuario de la base de datos.
 - ◇ 'PASSWORD': la contraseña del usuario de la base de datos.
- **Preparar y crear la base de datos**
- **Creación de migraciones:**
 - `python manage.py makemigrations plataforma sessions auth admin`
 - `python manage.py migrate`
- **Incorporación de funcionalidades de texto**
 - En la consola escribir: `python`
 - Una vez abierto el terminal Python escribir:
 - `import nltk`
 - `nltk.download()`
 - En la interfaz gráfica, añadir los paquetes punkt y stopwords.
 - En caso de no aparecer, introducir los comandos `d stopwords` y `d punkt`

Apéndice B

Manual de uso

Se presenta una explicación detallada sobre las posibles acciones y el uso de las funcionalidades disponibles en la plataforma, desde el registro de un usuario, hasta su gestión, la de colecciones y documentos y el trabajo con los mismos.

B.1. Consideraciones previas

Si se necesita **ayuda**, es posible encontrar iconos con símbolos de cierre de interrogación a lo largo de la plataforma. Estos símbolos contienen mensajes de explicación sobre ciertas funcionalidades. Para ver las mencionadas notas basta con posicionar el ratón sobre los propios iconos y mantenerlo encima hasta que aparezca el texto. Los iconos poseen la siguiente apariencia:



Figura B.1: Símbolo de ayuda

B.2. Creación de un usuario en la plataforma

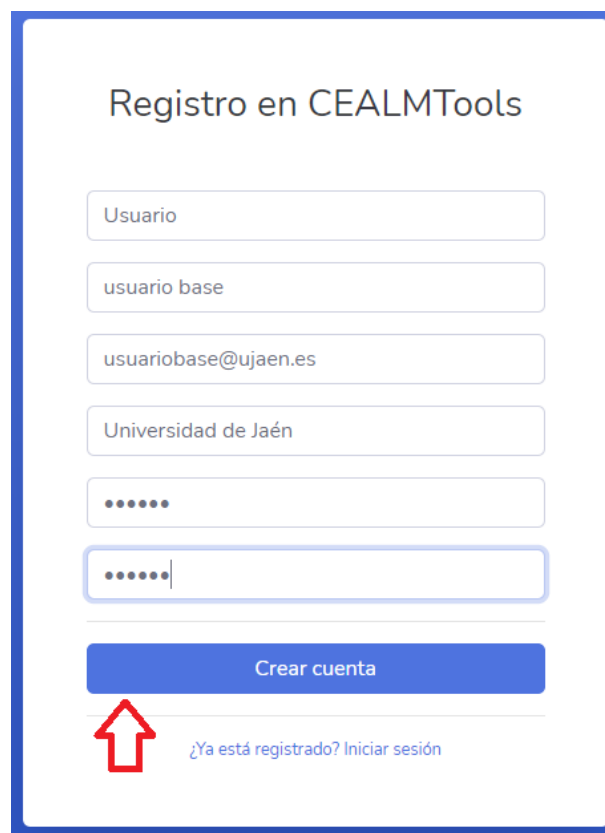
La primera vez que entramos a la plataforma necesitaremos una cuenta de usuario en la misma. Para ello, hacemos click en el enlace '**Crear cuenta**' de la página de inicio de sesión.



Figura B.2: Crear cuenta en CEALMTools

Seguidamente, rellenamos los campos del formulario que se nos presenta para crear una cuenta en la plataforma y hacemos click en '**Crear cuenta**'.

En caso de que ya tengamos una cuenta creada en el sistema, haremos click en el botón '**¿Ya está registrado? Iniciar sesión**' para volver a la página de inicio de sesión.



Registro en CEALMTools

Usuario

usuario base

uariobase@ujaen.es

Universidad de Jaén

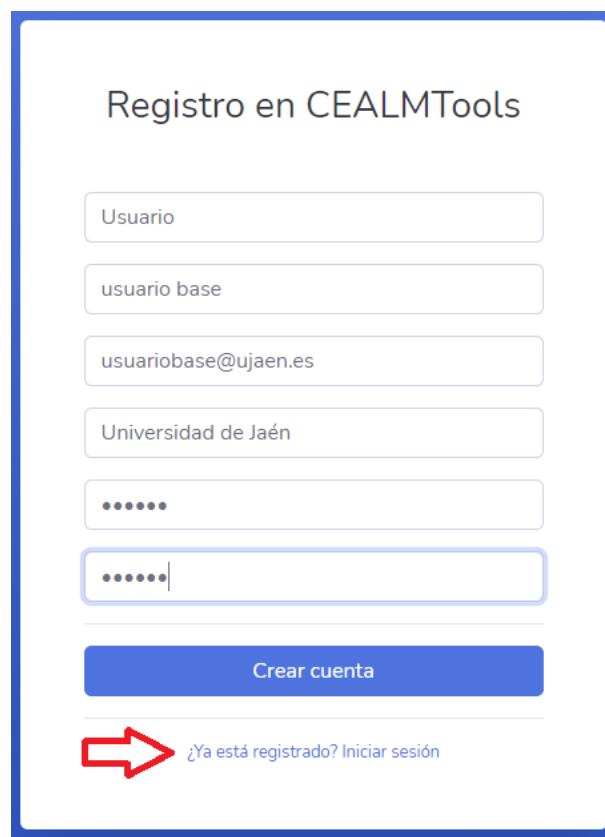
.....

.....

Crear cuenta

 ¿Ya está registrado? Iniciar sesión

Figura B.3: Registro en CEALMTools



Registro en CEALMTools

Usuario

usuario base

usuariobase@ujaen.es

Universidad de Jaén

.....

.....

Crear cuenta

 [¿Ya está registrado? Iniciar sesión](#)

Figura B.4: Vuelta a inicio de sesión en CEALMTools

B.3. Iniciar sesión en la plataforma

Para iniciar sesión en la plataforma, introduciremos el correo electrónico que indicamos cuando creamos la cuenta de usuario en la plataforma y la contraseña que escogimos. Una vez introducidos los datos, hacemos click en **'Iniciar sesión'**:



Figura B.5: Inicio de sesión en CEALMTools

Si nuestros datos son correctos, se nos concederá acceso a la plataforma, donde aparecerá nuestro espacio de trabajo y una serie de opciones en una barra lateral a la izquierda de la pantalla:

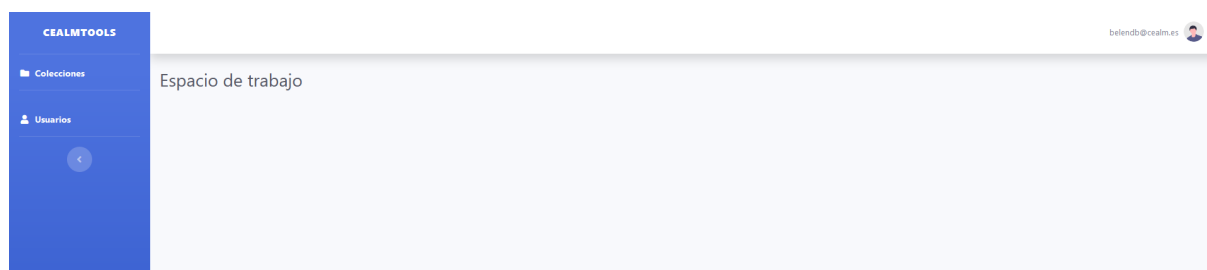


Figura B.6: Espacio de trabajo CEALMTools

B.4. Cerrar sesión en la plataforma

Para cerrar nuestra sesión en la plataforma, iremos a la esquina superior derecha, pulsaremos en nuestro correo, abriremos un pequeño menú desplegable y pulsaremos “**Cerrar sesión**”:

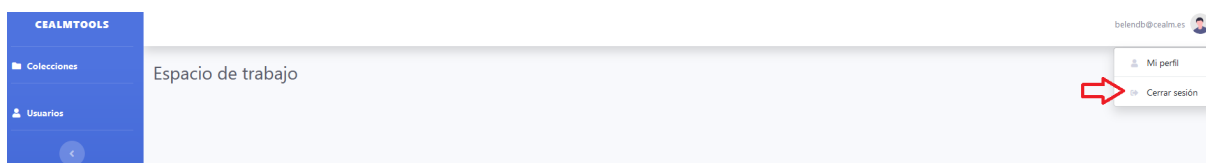


Figura B.7: Enlace a cierre de sesión



Figura B.8: Vista de cierre de sesión

B.5. Gestión de usuarios

Si hacemos click en la pestaña de la barra lateral “**Usuarios**” podremos navegar a la zona de gestión de usuarios:

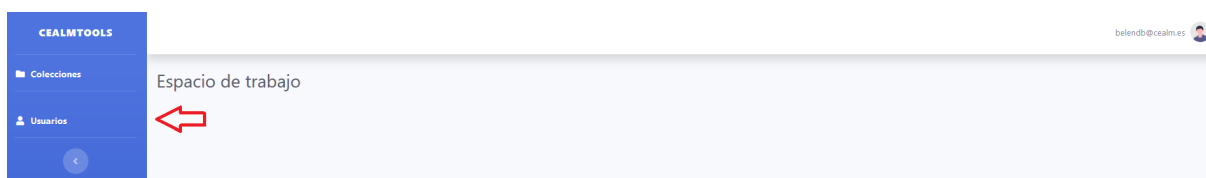
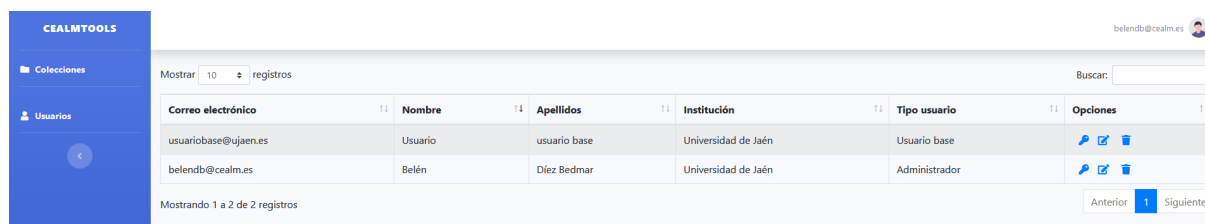


Figura B.9: Enlace a sección de gestión de usuarios



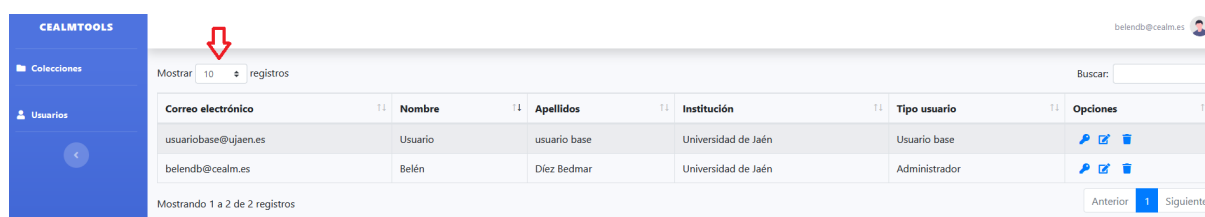
Correo electrónico	Nombre	Apellidos	Institución	Tipo usuario	Opciones
usuariobase@ujaen.es	Usuario	usuario base	Universidad de Jaén	Usuario base	
belendb@cealm.es	Belén	Díez Bedmar	Universidad de Jaén	Administrador	

Figura B.10: Sección de gestión de usuarios

En este espacio podremos, de forma gráfica, administrar los usuarios de la plataforma. Se nos presenta una tabla con los datos relevantes de los usuarios y podemos realizar diversas operaciones.

B.5.1. Mostrar registros

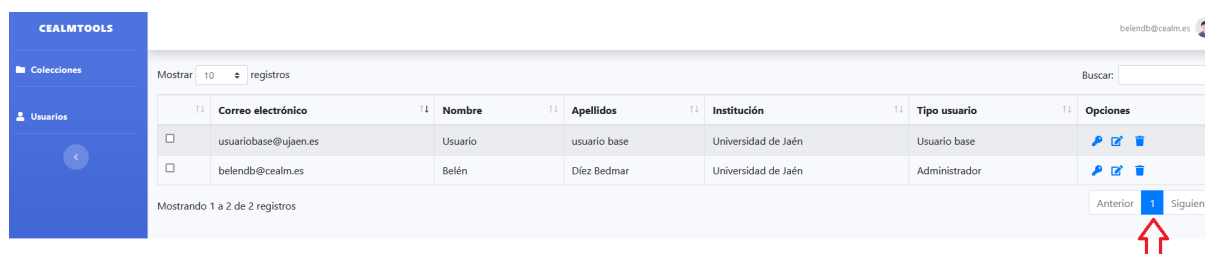
Si queremos mostrar más registros de los que aparecen actualmente haremos click en el botón selección “**Mostrar - registros**” siendo las opciones 10, 20, 30 o todos los registros:



Correo electrónico	Nombre	Apellidos	Institución	Tipo usuario	Opciones
usuariobase@ujaen.es	Usuario	usuario base	Universidad de Jaén	Usuario base	
belendb@cealm.es	Belén	Díez Bedmar	Universidad de Jaén	Administrador	

Figura B.11: Mostrar registros

En caso de que nos sea más cómodo mostrar de 10 en 10 registros, para mostrar la siguiente página de registros de usuarios, pulsaremos en la que deseemos en la sección “**Anterior - Siguiente**”:

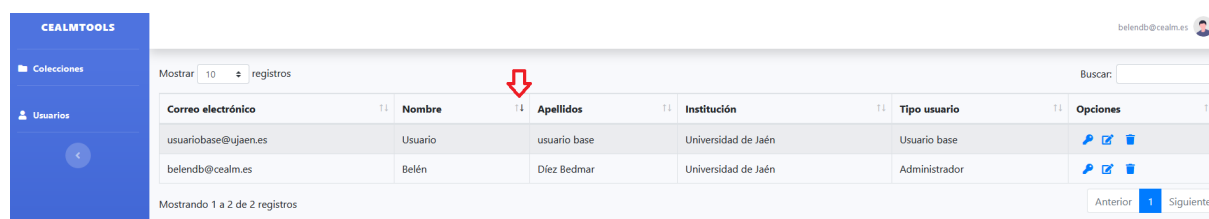


	Correo electrónico	Nombre	Apellidos	Institución	Tipo usuario	Opciones
☐	usuariobase@ujaen.es	Usuario	usuario base	Universidad de Jaén	Usuario base	
☐	belendb@cealm.es	Belén	Díez Bedmar	Universidad de Jaén	Administrador	

Figura B.12: Pestañas de registros

B.5.2. Ordenar registros

Si deseamos invertir el orden de los registros, haremos click en el icono de flecha situado al lado de cada uno de los nombres de los campos de la tabla:



Correo electrónico	Nombre	Apellidos	Institución	Tipo usuario	Opciones
usuariobase@ujaen.es	Usuario	usuario base	Universidad de Jaén	Usuario base	
belendb@cealm.es	Belén	Díez Bedmar	Universidad de Jaén	Administrador	

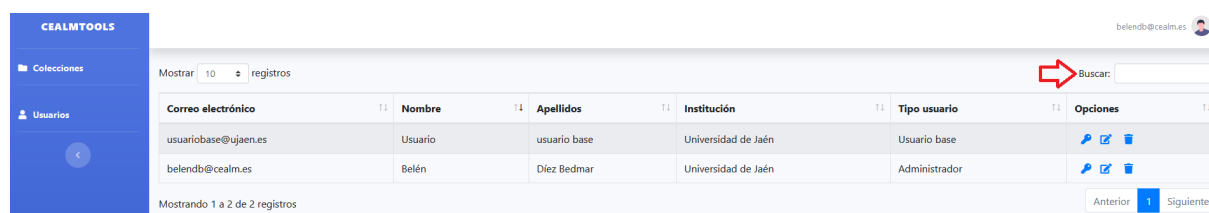
Mostrando 1 a 2 de 2 registros

Anterior 1 Siguiente

Figura B.13: Ordenar registros

B.5.3. Buscar registros

Si deseamos buscar un registro en concreto, escribiremos en el cuadro de texto con nombre “Buscar” el dato del campo que deseamos del registro que deseamos:



Correo electrónico	Nombre	Apellidos	Institución	Tipo usuario	Opciones
usuariobase@ujaen.es	Usuario	usuario base	Universidad de Jaén	Usuario base	
belendb@cealm.es	Belén	Díez Bedmar	Universidad de Jaén	Administrador	

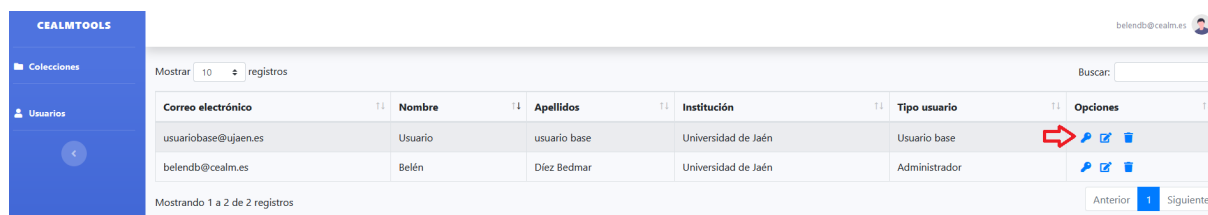
Mostrando 1 a 2 de 2 registros

Anterior 1 Siguiente

Figura B.14: Buscar registros

B.5.4. Cambiar contraseña de usuario

Para cambiar la contraseña de un usuario concreto, haremos click en el icono con forma de llave del registro que deseamos:



Correo electrónico	Nombre	Apellidos	Institución	Tipo usuario	Opciones
usuariobase@ujaen.es	Usuario	usuario base	Universidad de Jaén	Usuario base	 
belendb@cealm.es	Belén	Díez Bedmar	Universidad de Jaén	Administrador	 

Figura B.15: Enlace a cambio de contraseña de usuario

Una vez dentro podemos modificar la contraseña del usuario, escribiéndola dos veces para que así no haya fallos a la hora de la actualización y guardamos el resultado pulsando en el botón homónimo.

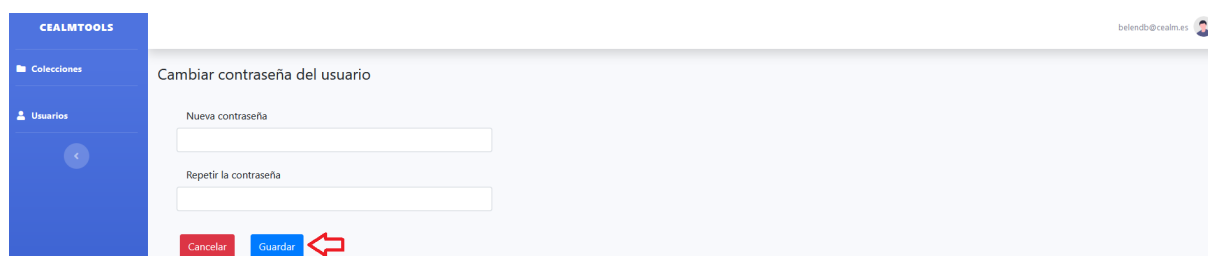
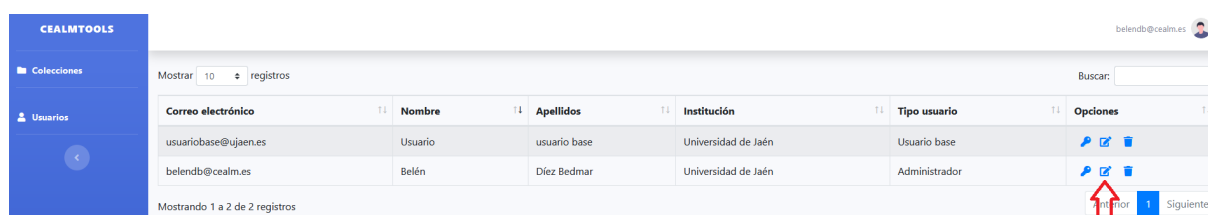


Figura B.16: Cambio de contraseña a usuario

B.5.5. Editar usuarios

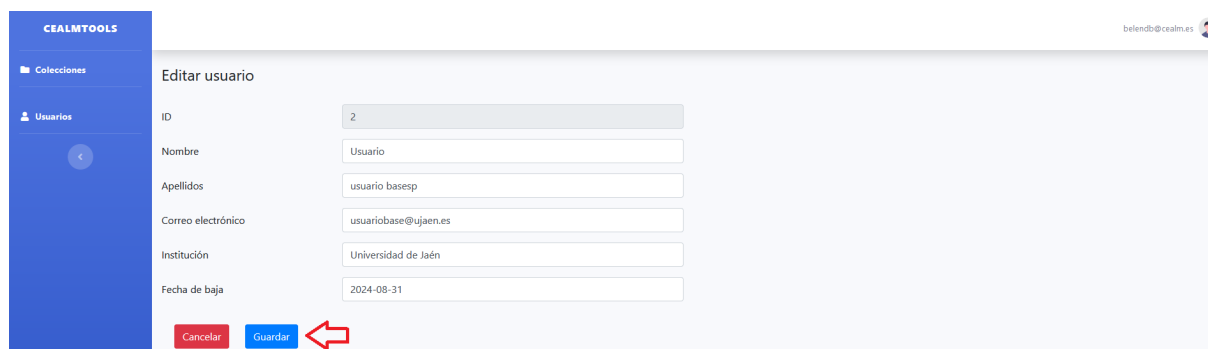
En caso de que deseemos editar un registro de usuario, en primer lugar haremos click en el icono de edición de la tabla:



Correo electrónico	Nombre	Apellidos	Institución	Tipo usuario	Opciones
usuariobase@ujaen.es	Usuario	usuario base	Universidad de Jaén	Usuario base	 
belendb@cealm.es	Belén	Díez Bedmar	Universidad de Jaén	Administrador	 

Figura B.17: Enlace a edición de usuario

Seguidamente nos encontramos con la página de edición de usuario, aquí podremos modificar el campo o campos que deseemos y una vez lo hayamos hecho presionamos el botón “**Guardar**”:



The screenshot shows the 'Editar usuario' (Edit user) form in the CEALMTOOLS system. The form is located on the right side of the interface, with a blue sidebar on the left containing 'Colecciones' and 'Usuarios' (selected). The form fields are as follows:

Field	Value
ID	2
Nombre	Usuario
Apellidos	usuario basesp
Correo electrónico	usuariobase@ujaen.es
Institución	Universidad de Jaén
Fecha de baja	2024-08-31

At the bottom of the form, there are two buttons: 'Cancelar' (red) and 'Guardar' (blue). A red arrow points to the 'Guardar' button.

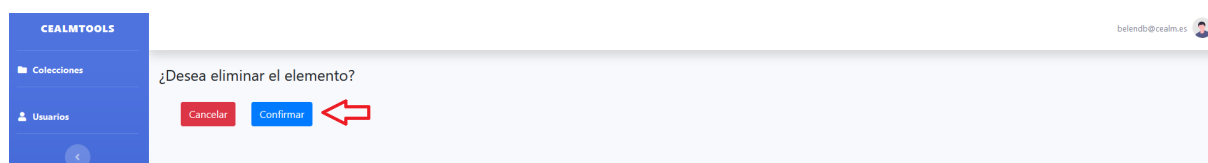
Figura B.18: Vista de edición de usuario

En caso de que no queramos modificar el registro finalmente, pulsaremos el botón “**Cancelar**” justo al lado del botón para editar el usuario.

Nota: al modificar la fecha, es importante mantener el formato AAAA-MM-DD que se muestra en esta vista.

B.5.6. Eliminar usuarios

Si deseamos eliminar un usuario antes de que caduque su cuenta en la plataforma, haremos click en el **icono de eliminar** en la tabla de registros:



The screenshot shows a confirmation dialog box titled '¿Desea eliminar el elemento?' (Do you want to delete the element?). The dialog is located on the right side of the interface, with a blue sidebar on the left containing 'Colecciones' and 'Usuarios' (selected). The dialog contains two buttons: 'Cancelar' (red) and 'Confirmar' (blue). A red arrow points to the 'Confirmar' button.

Figura B.19: Confirmación de eliminación de usuarios

B.6. Mi perfil

Es posible cambiar la información de nuestro propio usuario, pulsando en la esquina superior derecha, abriremos un desplegable y en el menú resultante, haremos click en el enlace “**Mi perfil**”:

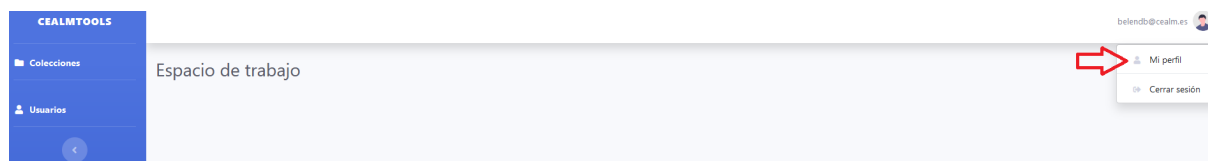


Figura B.20: Enlace a mi perfil

Si queremos modificar alguno de los parámetros de nuestra cuenta podremos hacerlo aquí (excepto la fecha de baja de la plataforma que es un valor inmutable). Una vez lo hayamos hecho, si queremos guardar nuestros cambios, pulsaremos “**Guardar**”, en caso contrario saldremos sin guardar mediante “**Cancelar**”.

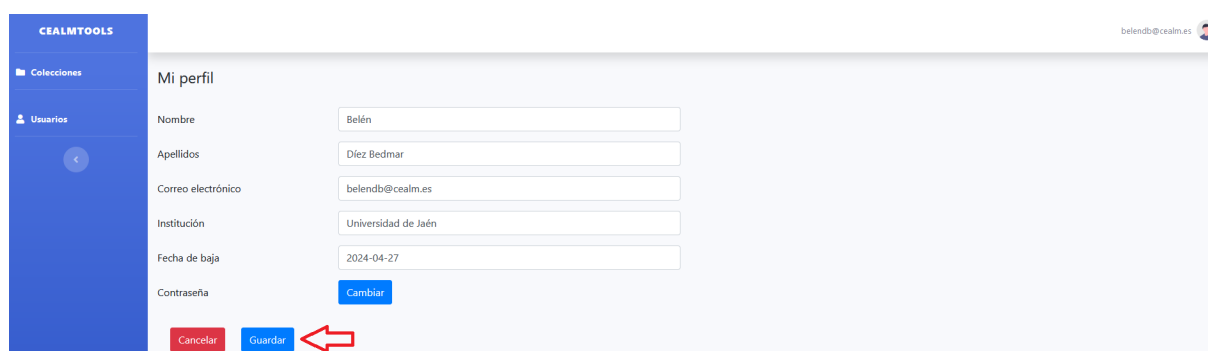


Figura B.21: Mi perfil

B.6.1. Cambiar contraseña propia

Podemos modificar nuestra propia contraseña desde esta vista pulsando en el botón “**Cambiar**”:

CEALMTOOLS

belendb@cealm.es

Mi perfil

Nombre: Belén

Apellidos: Díez Bedmar

Correo electrónico: belendb@cealm.es

Institución: Universidad de Jaén

Fecha de baja: 2024-04-27

Contraseña: [Cambiar](#)

[Cancelar](#) [Guardar](#)

Figura B.22: Enlace a cambio de contraseña propia

Esto nos llevará a un formulario donde deberemos escribir nuestra contraseña actual y la nueva contraseña dos veces, de esta manera aseguramos que la nueva contraseña es inequívoca. Es necesario seguir las normas que se estipulan en el formulario. Una vez hayamos terminado, si queremos confirmar la operación, pulsaremos “**Guardar**” en caso contrario “**Cancelar**”:

CEALMTOOLS

belendb@cealm.es

Cambiar contraseña

Contraseña antigua*

Contraseña nueva*

- Su contraseña no puede asemejarse tanto a su otra información personal.
- Su contraseña debe contener al menos 8 caracteres.
- Su contraseña no puede ser una clave utilizada comúnmente.
- Su contraseña no puede ser completamente numérica.

Contraseña nueva (confirmación)*

[Cancelar](#) [Guardar](#)

Figura B.23: Vista de cambio de contraseña propia

B.7. Gestión de colecciones

Si hacemos click en la pestaña de la barra lateral “**Colecciones**” podremos navegar a la zona de gestión de colecciones, donde aparecerá la tabla de registros de las mismas:

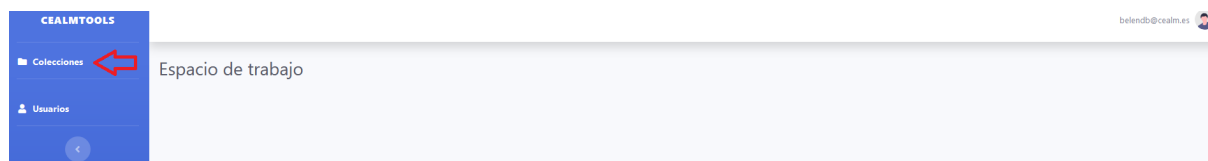


Figura B.24: Enlace a colecciones



Figura B.25: Vista de colecciones

Para mostrar, ordenar y buscar registros realizaremos las mismas acciones que en la sección de usuarios.

B.7.1. Crear colecciones

Para crear una colección, accederemos al apartado “**Añadir colección**”, justo encima de la tabla de registros en la pestaña colecciones:



Figura B.26: Enlace a añadir colección

Una vez dentro rellenamos los campos que se nos piden en el formulario y hacemos click en el botón “**Guardar**”. Si queremos cancelar la operación, pulsaremos el botón “**Cancelar**”:

CEALMTOOLS

belendb@cealm.es

Colecciones

Usuarios

Crear colección

Nombre de la colección*

Idioma de la colección*

Nivel de la colección*

Fecha de recuperación de la colección (DD/MM/AAAA)

Anotación de la colección*

Cancelar Guardar

Figura B.27: Vista de creación de colección

Nota: es importante que el formato de la fecha sea específicamente el estipulado en el título del campo de la fecha de recuperación, es decir, primero el día con dos dígitos (por ejemplo el día 7 del mes sería: 07), seguido del mes con dos dígitos (por ejemplo el mes de abril se escribiría: 04) y por último el año con sus cuatro dígitos, todos ellos separados por el símbolo “/” resultando en un formato como el que se muestra en la imagen.

B.7.2. Editar colecciones

Para editar una colección, procederemos de la misma manera que para editar un usuario, haremos click en el **icono de editar** correspondiente a la colección que deseamos modificar de la tabla de registros y una vez dentro editaremos los campos que deseemos:

CEALMTOOLS

belendb@ujaen.es

Colecciones

Usuarios

Añadir colección

Mostrar: 10 registros

Buscar:

Nombre	Idioma	Nivel	Fecha de recuperación	Anotación	Propietario	Opciones
B1	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	[Iconos de edición]
B2	Inglés	B2	23 de Febrero de 2023	No	belendb@ujaen.es	[Iconos de edición]

Mostrando 1 a 2 de 2 registros

Anterior 1 Siguiente

Figura B.28: Enlace a edición de colección

Para confirmar la operación, haremos click en el botón “**Guardar**”, para cancelarla, pulsaremos el botón “**Cancelar**”:

CEALMTOOLS

belendb@cealm.es

Editar colección

Nombre de la colección*

pruebas

Idioma de la colección*

Inglés

Nivel de la colección*

B1

Fecha de recuperación de la colección (DD/MM/AAAA)

21/02/2022

Anotación de la colección*

No

Cancelar Guardar

Figura B.29: Vista de edición de colección

B.7.3. Eliminar colecciones

Para eliminar una colección, procederemos de la misma manera que para eliminar un usuario, en primer lugar, haremos click en el **icono de eliminar** correspondiente a la colección que deseamos borrar en la tabla de registros.

CEALMTOOLS

belendb@ujaen.es

Colecciones

Añadir colección

Mostrar 10 registros

Buscar:

Nombre	Idioma	Nivel	Fecha de recuperación	Anotación	Propietario	Opciones
B1	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	[Iconos de acción]
B2	Inglés	B2	23 de Febrero de 2023	No	belendb@ujaen.es	[Iconos de acción]

Mostrando 1 a 2 de 2 registros

Anterior 1 Siguiente

Figura B.30: Enlace a eliminación de colección

Aparecerá una vista en la que decidiremos si eliminar el registro o cancelar la operación, haciendo click en el botón que represente la acción que queramos llevar a cabo.

CEALMTOOLS

belendb@cealm.es

¿Desea eliminar el elemento?

Cancelar Confirmar

Figura B.31: Vista de eliminación de colección

En caso de que la colección posea documentos en ella, es necesario desvincularlos de

Escuela Politécnica Superior de Jaén

la misma para poder eliminar una colección de la plataforma, a continuación la vista que aparecería en esta casuística:

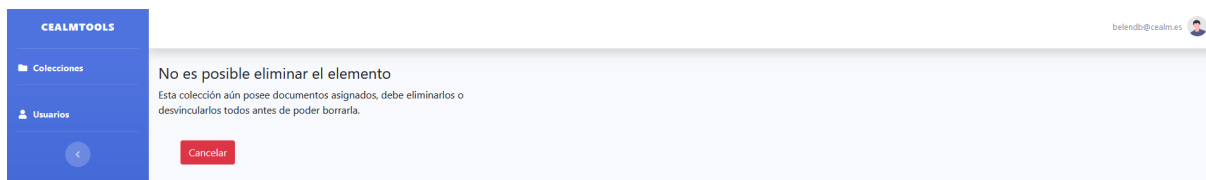


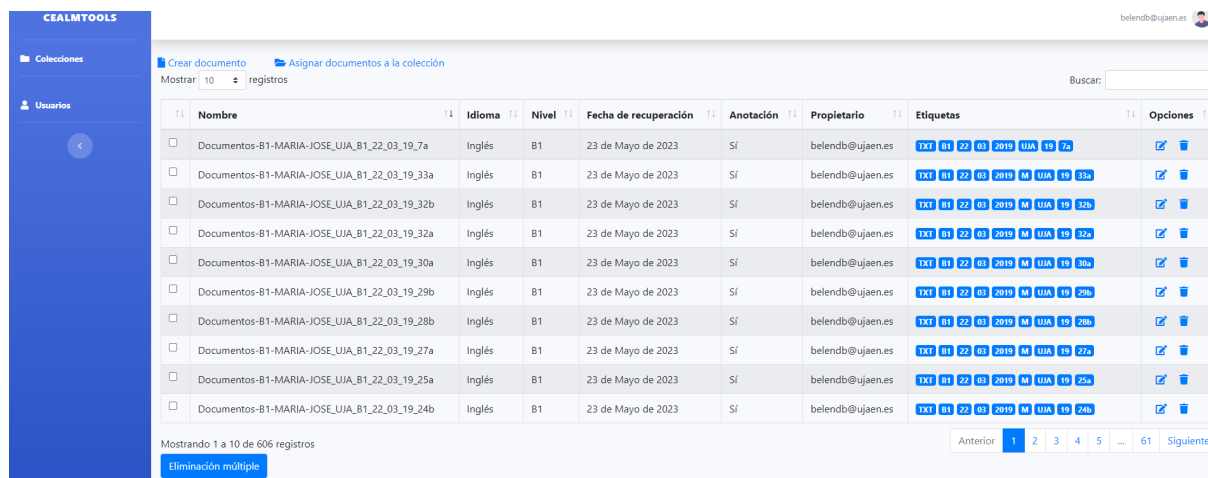
Figura B.32: Denegada eliminación de colección

B.8. Gestión de documentos

Para navegar a la sección de documentos, haremos click en el enlace llamado “**Ver archivos de la colección**” situado en la sección de acciones de la colección, siendo simbolizado por el icono de una lupa:



Figura B.33: Enlace a vista de documentos de la colección



CEALMTOOLS

belendb@ujaen.es

[Crear documento](#) [Asignar documentos a la colección](#)

Mostrar 10 registros

Buscar:

Ti	Nombre	Idioma	Nivel	Fecha de recuperación	Anotación	Propietario	Etiquetas	Opciones
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_7a	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXI B1 22 03 2019 UJA 19 7a	Editar Eliminar
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_33a	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXI B1 22 03 2019 M UJA 19 33a	Editar Eliminar
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_32b	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXI B1 22 03 2019 M UJA 19 32b	Editar Eliminar
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_32a	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXI B1 22 03 2019 M UJA 19 32a	Editar Eliminar
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_30a	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXI B1 22 03 2019 M UJA 19 30a	Editar Eliminar
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_29b	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXI B1 22 03 2019 M UJA 19 29b	Editar Eliminar
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_28b	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXI B1 22 03 2019 M UJA 19 28b	Editar Eliminar
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_27a	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXI B1 22 03 2019 M UJA 19 27a	Editar Eliminar
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_25a	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXI B1 22 03 2019 M UJA 19 25a	Editar Eliminar
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_24b	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXI B1 22 03 2019 M UJA 19 24b	Editar Eliminar

Mostrando 1 a 10 de 606 registros

[Eliminación múltiple](#)

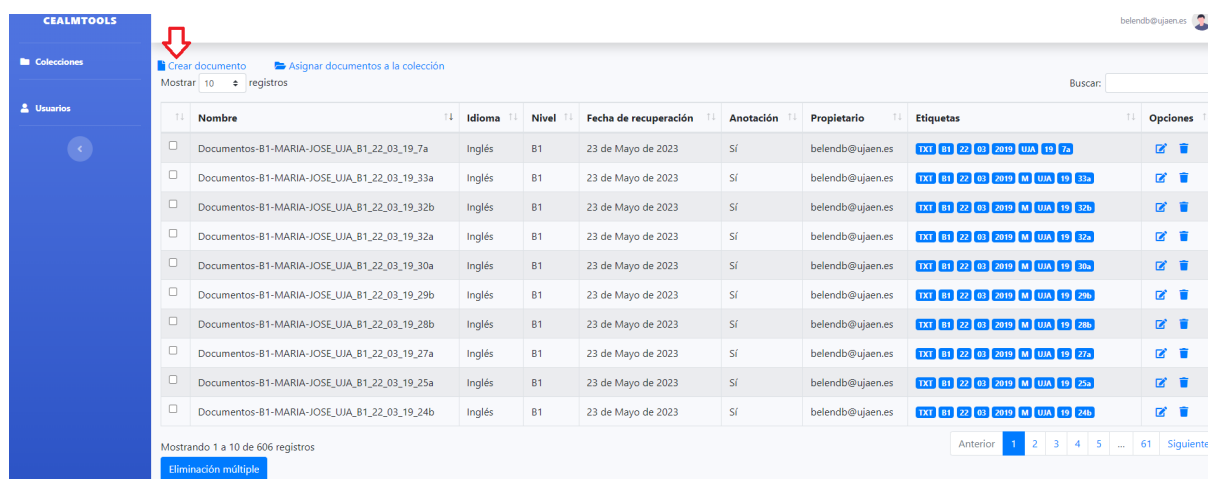
Anterior 1 2 3 4 5 ... 61 Siguiente

Figura B.34: Vista de documentos de la colección

Para mostrar, ordenar y buscar registros realizaremos las mismas acciones que en la sección de usuarios y colecciones.

B.8.1. Subir un documento

Para subir un documento a la plataforma, haremos click en el apartado “Crear documento”:



CEALMTOOLS

belendb@ujaen.es

[Crear documento](#) [Asignar documentos a la colección](#)

Mostrar 10 registros

Buscar:

Ti	Nombre	Idioma	Nivel	Fecha de recuperación	Anotación	Propietario	Etiquetas	Opciones
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_7a	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXI B1 22 03 2019 UJA 19 7a	Editar Eliminar
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_33a	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXI B1 22 03 2019 M UJA 19 33a	Editar Eliminar
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_32b	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXI B1 22 03 2019 M UJA 19 32b	Editar Eliminar
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_32a	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXI B1 22 03 2019 M UJA 19 32a	Editar Eliminar
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_30a	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXI B1 22 03 2019 M UJA 19 30a	Editar Eliminar
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_29b	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXI B1 22 03 2019 M UJA 19 29b	Editar Eliminar
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_28b	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXI B1 22 03 2019 M UJA 19 28b	Editar Eliminar
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_27a	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXI B1 22 03 2019 M UJA 19 27a	Editar Eliminar
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_25a	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXI B1 22 03 2019 M UJA 19 25a	Editar Eliminar
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_24b	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXI B1 22 03 2019 M UJA 19 24b	Editar Eliminar

Mostrando 1 a 10 de 606 registros

[Eliminación múltiple](#)

Anterior 1 2 3 4 5 ... 61 Siguiente

Figura B.35: Enlace a subida de documentos a la colección

Una vez dentro, rellenaremos el formulario que se nos presenta. Es necesario tomar ciertas consideraciones en algunos apartados:

- **Fecha de recuperación:** como sucede con las fechas en otros apartados de la plataforma, es necesario escribir la misma en el formato que se indica en el formulario.
- **Etiquetas:** en cuanto a las etiquetas, si se quieren asignar varias a un documento, se escribirán separadas por comas, como se especifica en el formulario.
 - Si se quieren importar las etiquetas presentes en el nombre del fichero, se seleccionará la opción que así lo indica en ese campo (es importante que las etiquetas del nombre del fichero estén separadas por guiones bajos (_)).
- **Codificación:** es **muy importante** seleccionar correctamente la codificación que posee el fichero que vamos a subir a la plataforma. Por defecto estará marcada la de Windows, puesto que es la que poseen la gran mayoría de ejemplos con los que se nos ha indicado que se trabaja. En caso de no elegir correctamente la codificación del archivo el sistema nos indicará que es necesario realizar la subida de nuevo puesto que no es posible decodificar los datos:

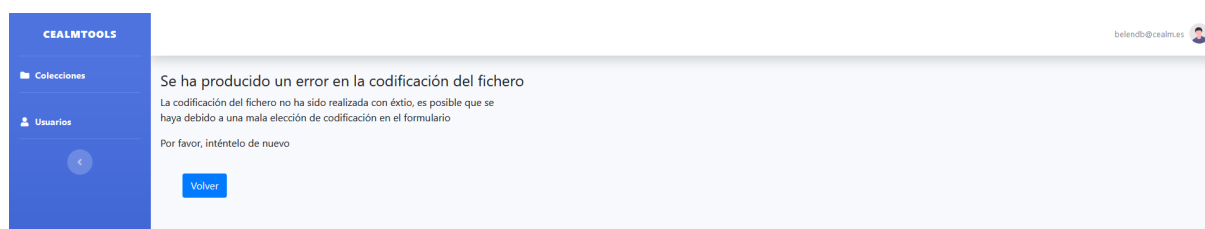


Figura B.36: Vista de error al seleccionar codificación del documento

- **Documento:** para subir un documento desde nuestro ordenador, pulsamos en “**Subir documento/s**” y escogeremos el que deseemos en nuestro sistema de archivos.

Finalmente para confirmar la subida del documento, haremos click en “**Guardar**”, si deseamos cancelar la operación, pulsaremos el botón “**Cancelar**”.

Figura B.37: Vista de subida de documentos

B.8.2. Subir un fichero comprimido ZIP

Para subir un fichero comprimido en **ZIP (sólo se admite este formato)** a la plataforma, el proceso será el mismo que para subir un único documento. Las consideraciones a tener en cuenta son las siguientes:

- **Fecha de recuperación:** como sucede con las fechas en otros apartados de la plataforma, es necesario escribir la misma en el formato que se indica en el formulario.
- **Etiquetas:** en cuanto a las etiquetas, si se quieren asignar varias a una serie de documentos, se escribirán separadas por comas, como se especifica en el formulario.
 - Si se quieren importar las etiquetas presentes en el nombre del fichero ZIP y las que existen en el nombre de cada documento, se seleccionará la opción que así lo indica en ese campo (es importante que las etiquetas de los nombres del fichero ZIP y documentos estén separadas por guiones bajos (_)).
- **Codificación:** es **muy importante** seleccionar correctamente la codificación que poseen los ficheros del archivo comprimido que vamos a subir a la plataforma. Por defecto estará marcada la de Windows, puesto que es la que poseen la gran mayoría de ejemplos con los que se nos ha indicado que se trabaja. Al subir el comprimido se nos presentará un resumen de los documentos que han podido ser subidos y los que no. En caso de que algún fichero no haya podido ser importado, inténtelo de nuevo seleccionando la codificación correcta.

- **Documentos:** para subir una serie de documentos comprimidos en **ZIP** desde nuestro ordenador, pulsamos en “**Subir documento/s**” y escogeremos el que deseemos en nuestro sistema de archivos.

Finalmente para confirmar la subida del documento, haremos click en “**Guardar**”, si deseamos cancelar la operación, pulsaremos el botón “**Cancelar**”.

Subir documento

Nombre del documento*

C1_enero

Idioma del documento*

Inglés

Nivel del documento*

C1

Fecha de recuperación del documento (DD/MM/AAAA)*

08/01/2021

Anotación del documento*

Sí

☐ Importar etiquetas del nombre del fichero ⓘ

Etiquetas del documento*

C1,enero

Una lista de etiquetas separadas por coma.

Codificación del documento

ANSI (Windows1252)

Subir documento/s*

Examinar... B2_B5_ingles_no.zip

Cancelar Guardar

Figura B.38: Vista de subida de fichero ZIP

CEALMTOOLS

Resumen de importación

Documentos que no pudieron ser importados (revise la codificación de los mismos)

Nombre
codif-iso_iso.txt
codif-iso_tagiso.txt

Documentos importados satisfactoriamente

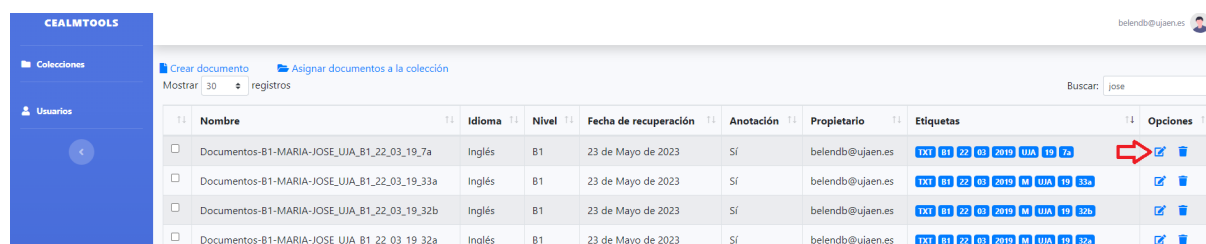
Nombre
codif-utf16_2635.txt

Continuar

Figura B.39: Vista de resumen de importación de documentos de un ZIP

B.8.3. Editar documentos

Para editar los datos de un documento de una colección, seleccionaremos el icono que representa dicha acción en la tabla de documentos de la colección:



CEALMTOOLS

belendb@ujaen.es

Crear documento Asignar documentos a la colección

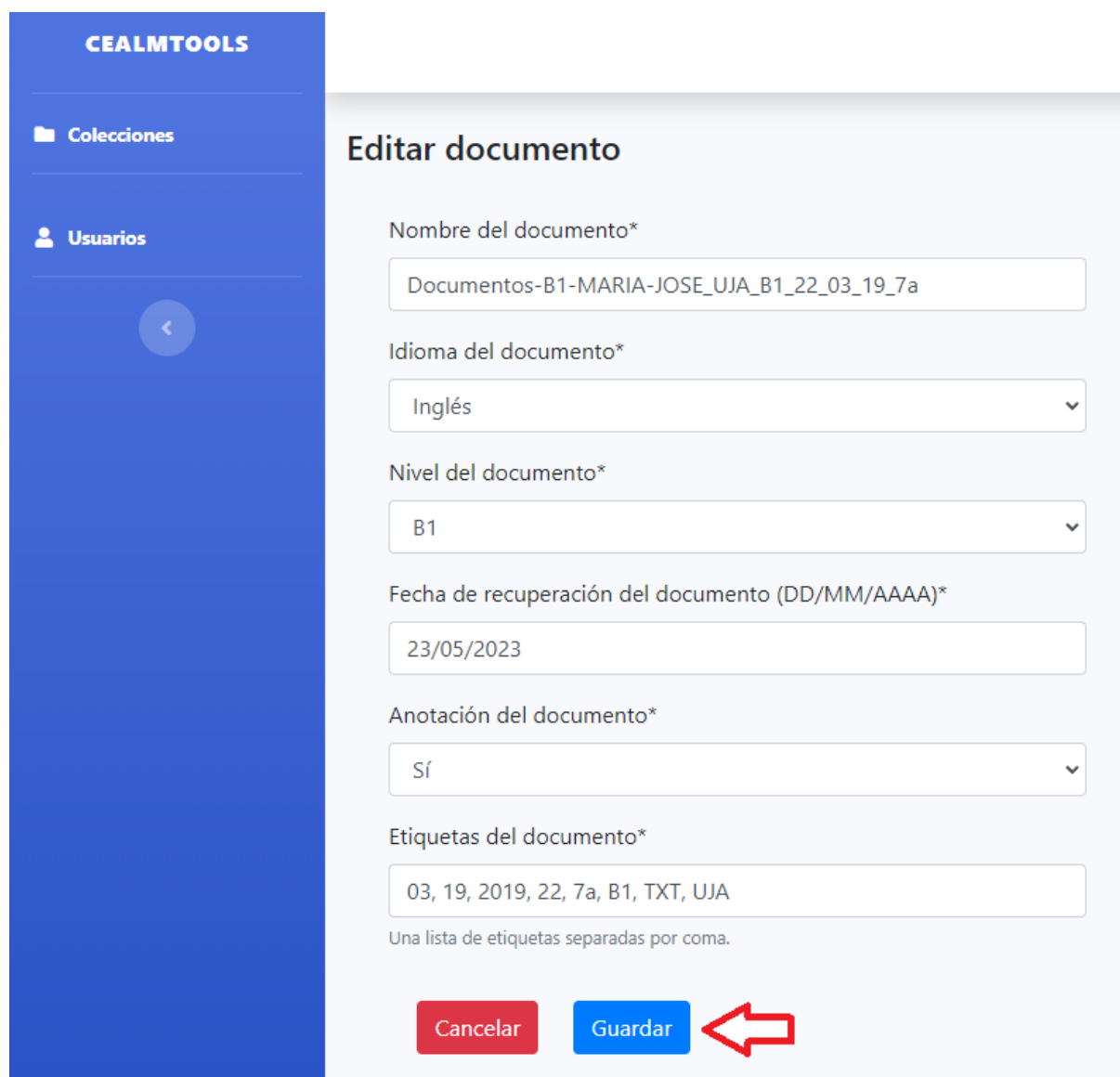
Mostrar 30 registros

Buscar: jose

	Nombre	Idioma	Nivel	Fecha de recuperación	Anotación	Propietario	Etiquetas	Opciones
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_7a	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXI B1 22 03 2019 UJA 19 7a	 
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_33a	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXI B1 22 03 2019 M UJA 19 33a	 
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_32b	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXI B1 22 03 2019 M UJA 19 32b	 
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_32a	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXI B1 22 03 2019 M UJA 19 32a	 

Figura B.40: Enlace a edición de documento

Pasaremos a la vista en la que es posible modificar la información perteneciente a un documento y una vez hayamos terminado de editar la información del archivo, pulsaremos en “**Guardar**” para salvar los cambios. En caso contrario, seleccionaremos “**Cancelar**” y volveremos a la vista anterior:



CEALMTOOLS

Colecciones

Usuarios

Editar documento

Nombre del documento*

Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_7a

Idioma del documento*

Inglés

Nivel del documento*

B1

Fecha de recuperación del documento (DD/MM/AAAA)*

23/05/2023

Anotación del documento*

Sí

Etiquetas del documento*

03, 19, 2019, 22, 7a, B1, TXT, UJA

Una lista de etiquetas separadas por coma.

Cancelar Guardar

Figura B.41: Vista de edición de documento

B.8.4. Eliminar un documento

Para eliminar un documento, procederemos de la misma manera que para eliminar un usuario o una colección, en primer lugar, haremos click en el **icono de eliminar** correspondiente al documento que deseamos borrar en la tabla de registros.



	Nombre	Idioma	Nivel	Fecha de recuperación	Anotación	Propietario	Etiquetas	Opciones
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_7a	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXT B1 22 03 2019 UJA 19 7a	
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_33a	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXT B1 22 03 2019 M UJA 19 33a	
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_32b	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXT B1 22 03 2019 M UJA 19 32b	
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_32a	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXT B1 22 03 2019 M UJA 19 32a	

Figura B.42: Enlace a eliminación individual de documento

En este caso pueden aparecer dos vistas, una que nos anunciará la desvinculación y no eliminación definitiva del documento de esa colección porque sigue vinculado a otra o la eliminación definitiva del documento de la plataforma puesto que solo tiene vínculo con el presente corpus, lo podemos ver a continuación:

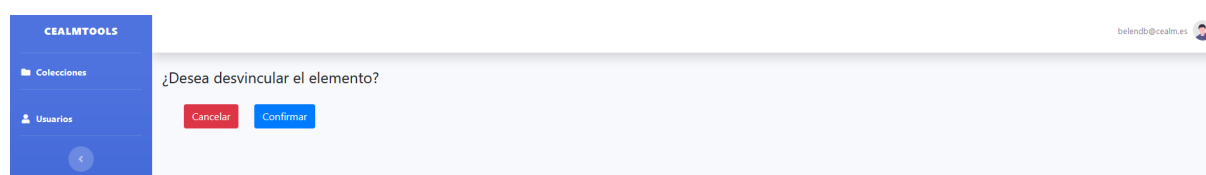


Figura B.43: Vista de desvinculación de documento

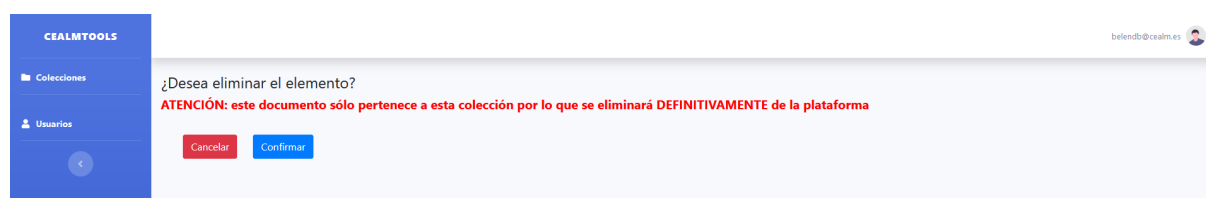


Figura B.44: Vista de eliminación definitiva de documento

B.8.5. Eliminar varios documentos a la vez

Es posible seleccionar varios documentos y eliminarlos de una sola vez. En la vista donde podemos ver los ficheros pertenecientes a un corpus, seleccionaremos las cajas a la izquierda del todo de la tabla correspondientes a cada uno de los textos que queramos desechar:

CEALMTOOLS

belendb@ujaen.es

Crear documento | Asignar documentos a la colección

Mostrar 10 registros

Buscar:

T1	Nombre	Idioma	Nivel	Fecha de recuperación	Anotación	Propietario	Etiquetas	Opciones
☒	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_7a	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXT B1 22 03 2019 UJA 19 7a	
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_33a	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXT B1 22 03 2019 M UJA 19 33a	
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_32b	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXT B1 22 03 2019 M UJA 19 32b	
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_32a	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXT B1 22 03 2019 M UJA 19 32a	
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_30a	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXT B1 22 03 2019 M UJA 19 30a	
☒	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_29b	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXT B1 22 03 2019 M UJA 19 29b	
☒	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_28b	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXT B1 22 03 2019 M UJA 19 28b	
☒	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_27a	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXT B1 22 03 2019 M UJA 19 27a	
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_25a	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXT B1 22 03 2019 M UJA 19 25a	
☐	Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_24b	Inglés	B1	23 de Mayo de 2023	Sí	belendb@ujaen.es	TXT B1 22 03 2019 M UJA 19 24b	

Mostrando 1 a 10 de 606 registros

Eliminar múltiple

Anterior 1 2 3 4 5 ... 61 Siguiente

Figura B.45: Selección de documentos para eliminación múltiple y enlace a vista

Una vez en la vista de eliminación, podremos ver un resumen en el que se separarán los documentos en dos categorías, los que pertenecen a otras colecciones y por lo tanto sólo se desvincularán de la presente, o los que únicamente están ligados a la actual y por lo tanto se eliminan definitivamente de la plataforma. Si queremos confirmar la operación, pulsaremos el botón homónimo, en caso contrario “**Cancelar**”:

CEALMTOOLS

belendb@ujaen.es

Confirmar selecciones

ATENCIÓN: los documentos en la siguiente tabla pertenecen sólo a esta colección por lo que si confirma la operación se eliminarán DEFINITIVAMENTE de la plataforma

Nombre
Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_29b
Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_28b
Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_27a

Documentos a desvincular de esta colección (pertenecientes a otras)

Nombre
Documentos-B1-MARIA-JOSE_UJA_B1_22_03_19_7a

Figura B.46: Vista de confirmación de selecciones para eliminación

B.8.6. Asignar documentos a colecciones

Para asignar documentos pertenecientes a otro corpus a la presente colección, haremos click en el botón “Asignar documentos a la colección”:

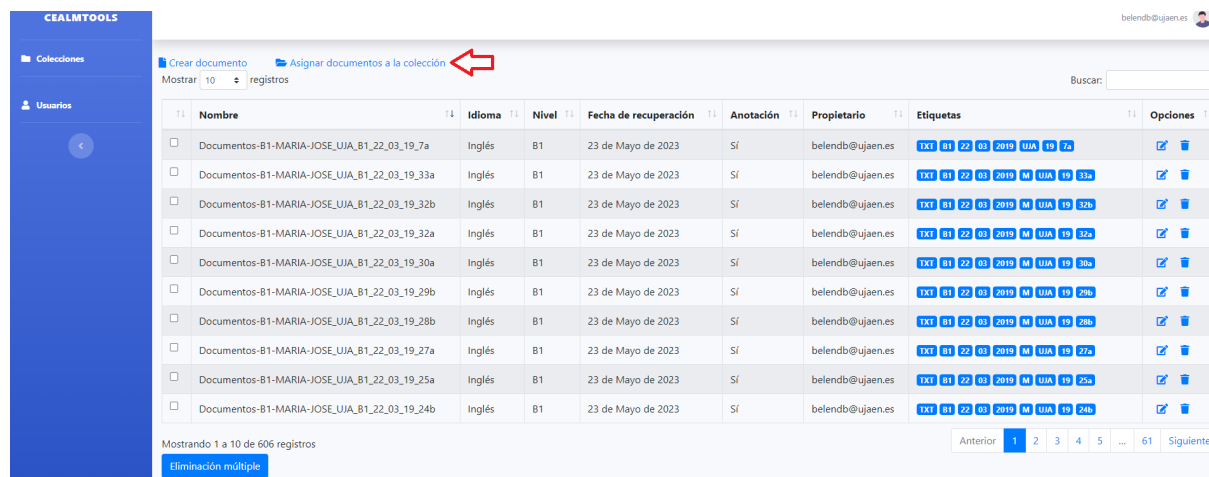


Figura B.47: Enlace a asignación de documentos a colección

Una vez dentro podremos asignar documentos a la actual colección seleccionando el recuadro de la izquierda de la tabla correspondiente a cada registro asignable. Una vez tengamos todos los que deseamos marcados, confirmaremos la operación pulsando “Guardar” y en caso contrario “Cancelar”:

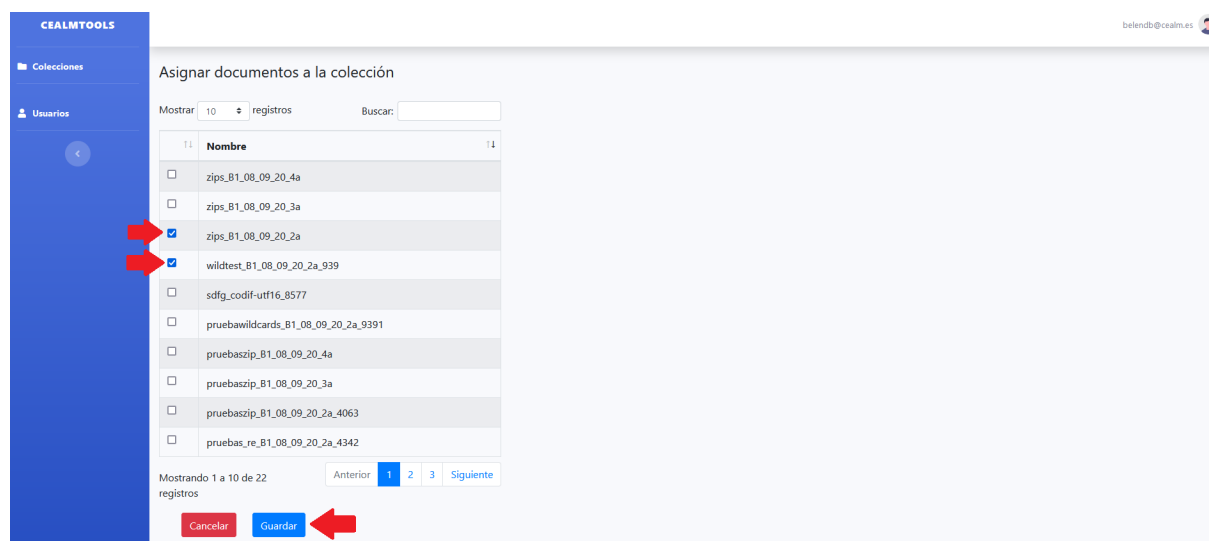


Figura B.48: Vista de selección y confirmación de vinculación de documentos

B.9. Uso de herramientas

Si queremos navegar hasta la sección de análisis textual nos situaremos en la vista de colecciones y haremos click en el icono que representa la acción como se muestra a continuación:



Figura B.49: Enlace a análisis de documentos

B.9.1. Wordlist

Esta herramienta realizará un conteo de palabras a lo largo de los textos de la colección en el instante en el que naveguemos a esa funcionalidad. En caso de que queramos consultar una palabra o palabras (**mediante comodines**) concreta/s, introduciremos el texto necesario en el recuadro, marcaremos si deseamos un procesado pasando todos los tokens a minúsculas y/o eliminando palabras vacías y pulsaremos **“Empezar”**:

Palabra	Frecuencia	N.Documentos
you	50	6
I	38	6
to	35	6
the	27	6
a	23	6
is	17	6
that	17	6
for	15	6
in	14	5
have	11	5

Mostrando 1 a 10 de 285 registros

☐ Pasarlo todo el texto a minúsculas ?
 ☐ Eliminar stopwords de los textos ?

Introduzca una palabra

Anterior 1 2 3 4 5 ... 29 Siguiente

Figura B.50: Vista de herramienta Wordlist

Para mostrar, ordenar y buscar registros realizaremos las mismas acciones que en los demás apartados. **NOTA:** se recomienda encarecidamente consultar los iconos con el símbolo de interrogación si necesitamos una breve descripción de la utilidad de cada campo.

B.9.2. N-Grams

La funcionalidad N-Grams permitirá visualizar la palabra que deseemos junto con las que la acompañan, en función de los parámetros escogidos. En caso de que queramos consultar una palabra o palabras (**mediante comodines**) concreta/s, introduciremos el texto necesario en el recuadro, marcaremos si deseamos un procesado pasando todos los tokens a minúsculas y/o eliminando palabras vacías, la dirección de la búsqueda (las palabras que acompañan a la consultada aparecen por la derecha o la izquierda) y el número de términos acompañantes que deseemos que aparezcan y pulsaremos **“Empezar”**:

Figura B.51: Vista de herramienta N-Grams

Para mostrar, ordenar y buscar registros realizaremos las mismas acciones que en los demás apartados.

B.9.3. Concordance

Gracias a esta característica podremos buscar las ocurrencias de la palabra que deseemos y el contexto en el que se encuentran. Para consultar una palabra o palabras (**mediante comodines**) concreta/s, introduciremos el texto necesario en el recuadro, marcaremos si deseamos un procesamiento pasando todos los tokens a minúsculas y/o eliminando palabras vacías, la dirección de la búsqueda (el contexto que acompaña a la consultada aparece resaltado por la derecha o la izquierda en función de la dirección escogida) y el número de términos acompañantes que deseemos que aparezcan y pulsaremos “**Empezar**”:

Figura B.52: Vista de herramienta Concordance

Para mostrar, ordenar y buscar registros realizaremos las mismas acciones que en los demás apartados.

B.9.4. Concordance Plot

Esta herramienta permite la creación de diagramas de dispersión que representan las ocurrencias de la palabra solicitada a lo largo de cada texto de un corpus concreto. Para consultar una palabra o palabras (**mediante comodines**) específica/s, introduciremos el texto necesario en el recuadro, marcaremos si deseamos un procesamiento pasando todos los tokens a minúsculas y/o eliminando palabras vacías y pulsaremos “**Empezar**”:

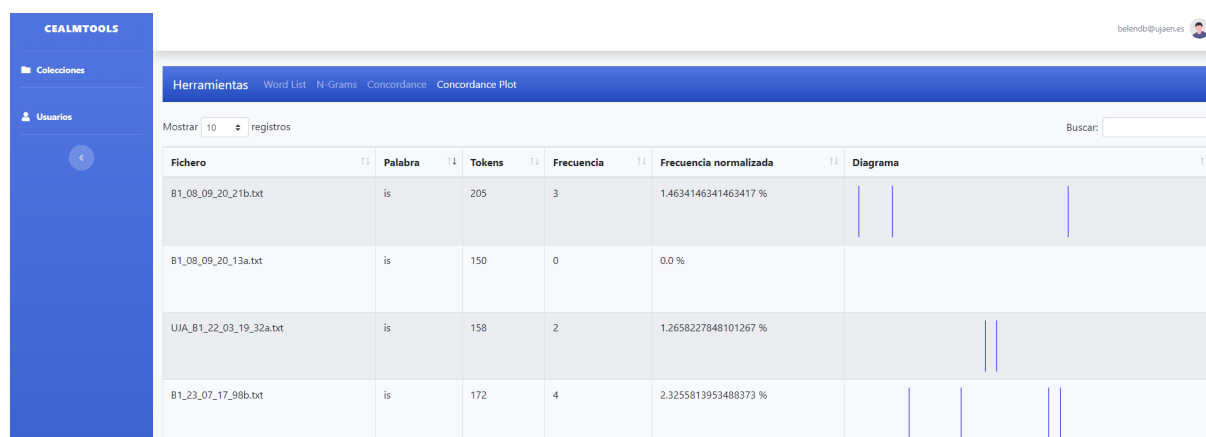


Figura B.53: Vista de herramienta Concordance Plot 1

B1_08_09_20_30a.txt	is	138	1	0.7246376811594203 %	
B1_23_07_17_66b.txt	is	123	2	1.6260162601626018 %	
23_05_17_14_a.txt	is	135	4	2.9629629629629632 %	
B1_F_L1Sp_EFL1_CN_1356a.txt	is	159	4	2.515723270440252 %	
B1_23_07_17_105a.txt	is	167	4	2.3952095808383236 %	
B1_08_09_20_5a.txt	is	156	2	1.282051282051282 %	

Mostrando 1 a 10 de 606 registros

☐ Pasar todo el texto a minúsculas 

☐ Eliminar stopwords de los textos 

Introduzca una palabra 

Anterior 1 2 3 4 5 ... 61 Siguiente

Figura B.54: Vista de herramienta Concordance Plot 2