

UNIVERSIDAD DE OVIEDO



Universidad de Oviedo

ESCUELA DE INGENIERÍA INFORMÁTICA

TRABAJO FIN DE GRADO

**“ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE
TEXTOS EDUCATIVOS”**

TUTOR: Jordán Pascual Espada

COTUTORA: Irene Cid Rico

AUTOR: Diego González Suárez

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

AGRADECIMIENTOS

Desde que empecé el grado allá por 2019, todo el esfuerzo que he empleado ha valido la pena, las palabras de agradecimiento tanto a mi familia, como amigos son infinitas.

En primer lugar me gustaría agradecer a mis padres, en especial a mi madre la cual me ha estado aguantando y apoyando incondicionalmente día a día, tanto en los pequeños éxitos como en los fracasos. También agradecer a mis abuelos, mis tíos y demás familia que siempre se ha interesado por mi trayectoria académica.

De este modo, jamás podré agradecer a mis amigos, todo lo que me han apoyado y aguantado durante este tiempo, cosa que, estoy seguro, de que no ha sido fácil. Muchos de ellos, amigos que conservo desde que tenía tres años y han estado a mi lado toda la vida. Sin ellos, estoy seguro de que todo hubiese sido mucho más cuesta arriba. Parte de los éxitos durante estos cuatro años han sido gracias a vosotros.

También agradecer a mis amigos de la universidad con los cuales he compartido grandes momentos, tanto a nivel académico, como personal. Momentos que espero compartir en el futuro.

Finalmente, dar las gracias a mis tutores de este trabajo, así como a todos los profesores que me han enseñado y he conocido durante estos cuatro años. Todo lo que he aprendido estos años es gracias a vosotros y estaré eternamente agradecido.

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

CONTROL DE VERSIONES

Versión	Fecha	Descripción
1.0.0	07/06/2022	Versión inicial del trabajo de fin de grado

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

ÍNDICE DE CONTENIDO

1	¿QUÉ ES ESTE TRABAJO?	16
1.1	RESUMEN	16
1.2	PALABRAS CLAVE	16
1.3	ABSTRACT	17
1.4	KEYWORDS.....	17
2	ESTUDIO DEL ALGORITMO	18
2.1	CONTEXTO EN QUE SE ENMARCA LA EXTRACCIÓN DE KEYWORDS	18
2.1.1	<i>Natural Language Processing (NLP)</i>	18
2.1.2	<i>Búsqueda de keywords</i>	18
2.1.3	<i>Alternativas para la extracción de keywords</i>	19
2.2	ALGORITMO DESARROLLADO BASADO EN TEXTRank.....	22
2.2.1	<i>¿Qué es TextRank y de dónde viene?</i>	22
2.2.2	<i>Procesamiento del texto</i>	23
2.2.3	<i>Creación del grafo</i>	24
2.2.4	<i>Cálculo de la puntuación</i>	25
2.2.5	<i>Diagrama de componentes del algoritmo</i>	27
2.3	ANÁLISIS DE RESULTADOS.....	28
2.3.1	<i>Precisión</i>	28
2.3.2	<i>Recall o Sensitivity</i>	28
2.3.3	<i>F1 Score</i>	29
2.3.4	<i>Análisis de resultados</i>	29
2.3.5	<i>Conclusiones obtenidas del estudio</i>	33
3	ESTUDIO DE VIABILIDAD DEL SISTEMA.....	36
3.1	ANÁLISIS DE SISTEMAS SIMILARES	36
3.1.1	<i>QuillionZ (https://www.quillionz.com/)</i>	36
3.1.2	<i>Quizgecko (https://quizgecko.com/)</i>	37
3.1.3	<i>OpExams (https://opexams.com/free-questions-generator/)</i>	37
3.2	VALORACIÓN DE ALTERNATIVAS DE SOLUCIÓN	38
3.2.1	<i>Alternativas de enfoque del algoritmo</i>	38
3.2.2	<i>Desarrollo del backend de la aplicación</i>	38

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es
--	---	--	---

3.2.3 <i>Desarrollo del frontend de la aplicación web</i>	40
4 PLANIFICACIÓN Y GESTIÓN DEL PROYECTO	43
4.1 PLANIFICACIÓN INICIAL DEL PROYECTO	43
4.1.1 <i>Planificación inicial. WBS</i>	43
4.1.2 <i>Riesgos</i>	48
4.1.3 <i>Resumen presupuesto inicial</i>	53
4.2 PLANIFICACIÓN FINAL DEL PROYECTO	54
5 ANÁLISIS DEL SISTEMA DE INFORMACIÓN	55
5.1 DETERMINACIÓN DEL ALCANCE DEL SISTEMA	55
5.2 IDENTIFICACIÓN DE ACTORES DEL SISTEMA	55
5.3 ESTABLECIMIENTO DE REQUISITOS	56
5.3.1 <i>Obtención de los Requisitos del Sistema</i>	56
5.4 ESPECIFICACIÓN DE CASOS DE USO Y ESCENARIOS	66
5.4.1 <i>Diagrama de contexto</i>	66
5.4.2 <i>Especificación de casos de uso</i>	66
5.4.3 <i>Análisis de escenarios</i>	78
5.5 IDENTIFICACIÓN DE SUBSISTEMAS A NIVEL DE ARQUITECTURA DE LA APLICACIÓN	94
5.5.1 <i>Descripción de los Subsistemas a Nivel de Arquitectura</i>	95
5.6 DEFINICIÓN DE INTERFACES DE USUARIO.....	96
5.6.1 <i>Descripción de la Interfaz</i>	96
5.6.2 <i>Diagrama de Navegabilidad</i>	103
5.7 ESPECIFICACIÓN DEL PLAN DE PRUEBAS	104
6 DISEÑO DEL SISTEMA DE INFORMACIÓN	106
6.1 DISEÑO DE FÍSICO DE DATOS	106
6.1.1 <i>Descripción del SGBD utilizado</i>	106
6.1.2 <i>Integración del SGDB en el sistema</i>	106
6.1.3 <i>Diagrama E-R</i>	107
6.2 ARQUITECTURA DE MÓDULOS DEL SISTEMA	109
6.2.1 <i>Diagrama de componentes Webapp</i>	109
6.2.2 <i>Diagrama de componentes Restapi</i>	111
6.3 DISEÑO DE LA API REST	113

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

6.3.1	<i>Gestión de palabras clave</i>	113
6.3.2	<i>Gestión de preguntas</i>	113
6.3.3	<i>Gestión de usuarios</i>	114
6.3.4	<i>Gestión de tests</i>	115
6.4	DIAGRAMA DE DESPLIEGUE	117
6.5	DISEÑO DEL PLAN DE PRUEBAS	118
6.5.1	<i>Pruebas Unitarias</i>	118
6.5.2	<i>Pruebas de Sistema</i>	118
7	IMPLEMENTACIÓN DEL SISTEMA DE INFORMACIÓN	129
7.1	ESTÁNDARES Y NORMAS SEGUIDOS	129
7.1.1	<i>Guías PEP de Python</i>	129
7.2	LENGUAJES DE PROGRAMACIÓN	129
7.2.1	<i>Python</i>	129
7.2.2	<i>JavaScript</i>	129
7.3	HERRAMIENTAS USADAS PARA EL DESARROLLO	130
7.3.1	<i>Visual Studio Code</i>	130
7.3.2	<i>Mozilla Firefox</i>	130
7.3.3	<i>PyCharm Professional</i>	130
7.3.4	<i>Global Protect</i>	130
7.3.5	<i>Escritorio Remoto Windows</i>	131
7.3.6	<i>Git</i>	131
7.4	CONSIDERACIONES DE SEGURIDAD	131
7.4.1	<i>Token de acceso</i>	131
7.4.2	<i>Almacenamiento de contraseñas</i>	132
8	EJECUCIÓN DE LAS PRUEBAS	133
8.1	PRUEBAS ESTÁTICAS	133
8.1.1	<i>Análisis estático</i>	133
8.2	PRUEBAS UNITARIAS	134
8.2.1	<i>Restapi</i>	134
8.3	PRUEBAS DEL SISTEMA	134
8.3.1	<i>Pruebas Funcionales (end-to-end)</i>	134
8.3.2	<i>Pruebas con Usuarios</i>	135

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

8.3.3 <i>Pruebas de Adaptabilidad</i>	141
9 MANUALES DE USUARIO	146
9.1 MANUAL PARA EL DESARROLLADOR	146
9.1.1 <i>Descarga del código fuente</i>	146
9.1.2 <i>Creación de ficheros adicionales</i>	147
9.1.3 <i>Instalación de la REST API</i>	147
9.1.4 <i>Instalación de la Webapp</i>	149
9.1.5 <i>Añadir nuevos tipos de preguntas</i>	150
9.1.6 <i>Añadir nuevos formatos de exportación de preguntas</i>	150
9.1.7 <i>Añadir nuevos idiomas</i>	151
9.2 MANUAL DE USUARIO.....	152
9.2.1 <i>Acceso al sistema</i>	152
9.2.2 <i>Cambio de idioma</i>	152
9.2.3 <i>Autenticación de usuarios</i>	152
9.2.4 <i>Procesamiento de textos</i>	153
9.2.5 <i>Gestión de palabras clave</i>	154
9.2.6 <i>Gestión de preguntas</i>	155
9.2.7 <i>Gestión de cuestionarios</i>	156
10 CONCLUSIONES Y AMPLIACIONES.....	160
10.1 CONCLUSIONES	160
10.2 AMPLIACIONES	160
11 REFERENCIAS BIBLIOGRÁFICAS	162
12 GLOSARIO.....	164
13 ÍNDICE ALFABÉTICO	166
14 ANEXOS	168
14.1 PLAN DE GESTIÓN DE RIESGOS	168
14.1.1 <i>Valoración de riesgos</i>	169
14.1.2 <i>Gestión de riesgos</i>	169
14.2 MODELO DE EMPRESA SUPUESTO.....	170
14.2.1 <i>Costes de personal</i>	170
14.2.2 <i>Productividad del personal</i>	171
14.2.3 <i>Horas productivas y precio hora</i>	172

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

14.2.4	<i>Resumen modelo de empresa</i>	172
14.3	DETALLE DEL PRESUPUESTO.....	173
14.3.1	<i>Presupuesto de costes</i>	173
14.3.2	<i>Presupuesto de cliente</i>	185
14.4	DISEÑO DE LAS PRUEBAS	187
14.5	ESPECIFICACIÓN OPENAPI (OAS).....	187
14.6	LIBRERÍAS DE COMPONENTES EN REACT	188
14.7	CHATGPT EN LA BÚSQUEDA DE KEYWORDS.....	188
14.7.1	<i>¿Qué es ChatGPT?</i>	188
14.7.2	<i>¿Cómo probar ChatGPT en la búsqueda de keywords?</i>	189
14.8	GESTORES DE REFERENCIAS: MENDELEY	190
14.9	CONTENIDO ENTREGADO EN LOS ANEXOS.....	191

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

ÍNDICE DE FIGURAS

Figura 2.1. Ejemplo de grafo formado para un texto. Fuente: [13]	23
Figura 2.2. Esquema del grafo creado.	24
Figura 2.3. Ejemplo de los trigramas que va tomando el algoritmo.....	25
Figura 2.4. Generación de nuevos nodos a partir de varias palabras.....	26
Figura 2.5. Diagrama de componentes del algoritmo.....	27
Figura 2.6. <i>Precision</i> y <i>Recall</i> visto gráficamente [18].....	29
Figura 2.7. Ejecución de resultados para el capítulo 1 del libro <i>US History</i> . [17]	30
Figura 4.1. WBS con tareas resumen y diagrama de Gantt.....	44
Figura 4.2. Porcentaje de dedicación a cada tarea en base a la planificación inicial.	45
Figura 4.3. Desglose de tareas de la Investigación del algoritmo.	45
Figura 4.4. Desglose de tareas Elaborar la planificación y presupuestos.....	46
Figura 4.5. Desglose de tareas del Estudio de viabilidad del sistema.	46
Figura 4.6. Desglose de tareas del Análisis del sistema.	47
Figura 4.7. Desglose de tareas del Diseño del sistema.	47
Figura 4.8. Desglose de tareas de la Implementación y pruebas del sistema.	48
Figura 4.9. Desglose de tareas de la revisión de la documentación.	48
Figura 4.10. Matriz de probabilidad e impacto.	51
Figura 5.1. Diagrama de contexto de la aplicación	66
Figura 5.2. Diagrama de casos de uso de procesamiento del texto.	69
Figura 5.3. Diagrama de casos de uso para la gestión de palabras clave.	70
Figura 5.4. Diagrama de casos de uso para la gestión de preguntas de examen.	73
Figura 5.5. Diagrama de casos de uso para la gestión de cuestionarios.	76
Figura 5.6. Subsistemas de arquitectura con interfaces entre estos.....	95
Figura 5.7. Boceto de la página de inicio de sesión.	96
Figura 5.8. Boceto de la página de registro.	97
Figura 5.9. Boceto de la página principal.	98
Figura 5.10. Boceto de la vista de procesamiento del texto.	98
Figura 5.11. Boceto de vista de gestión de palabras clave.	99
Figura 5.12. Boceto de la página de gestión de preguntas.	100
Figura 5.13. Boceto de página de documentación.....	101

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

Figura 5.14. Boceto de vista de ver resultados de cuestionarios	101
Figura 5.15. Boceto de vista de entrar a cuestionario.....	102
Figura 5.16. Boceto de la vista de contestar pregunta en el cuestionario.....	103
Figura 5.17. Diagrama de navegabilidad del sistema	104
Figura 6.1. Funcionamiento ORM.....	106
Figura 6.2. Diagrama Entidad-Relación	108
Figura 6.3. Diagrama de componentes de la webapp.....	109
Figura 6.4. Diagrama de componentes de la restapi.....	111
Figura 6.5. Diagrama de despliegue del sistema	117
Figura 7.1. Ejemplo de envío de petición con token de acceso.....	132
Figura 8.1. Vista del panel de administración de SonarCloud.	133
Figura 8.2. <i>Quality Gate</i> en el repositorio (https://github.com/uo276406/tfg_app)....	133
Figura 8.3. Resultado de ejecución de pruebas unitarias de la restapi.	134
Figura 8.4. Panel de opciones de ejecución de tests.....	134
Figura 8.5. Resultado ejecución pruebas funcionales de la webapp.	135
Figura 8.6. Gráfico con encuestas sobre colores empleados y tamaño y tipo de letra.	140
Figura 8.7. Formulario de inicio de sesión adaptable a dispositivo móvil.....	141
Figura 8.8. Formulario de registro adaptable a dispositivo móvil.....	141
Figura 8.9. Formulario de unirse a test adaptable a dispositivo móvil.....	142
Figura 8.10. Vista de realización de cuestionarios adaptable a dispositivo móvil.	142
Figura 8.11. Formulario inicio de sesión adaptable tableta.....	142
Figura 8.12. Formulario de registro adaptable a tableta.	143
Figura 8.13. Formulario de unirse a test adaptable a tableta.	143
Figura 8.14. Vista de realización de cuestionarios adaptable a tableta.	144
Figura 8.15. Formulario de inicio de sesión adaptable a monitor.	144
Figura 8.16. Formulario de registro de usuarios adaptable a monitor.....	145
Figura 8.17. Formulario de unirse a test adaptable a monitor.	145
Figura 8.18. Vista de realización de cuestionarios adaptable a monitor.	145
Figura 9.1. Descarga proyecto comprimido de GitHub.....	146
Figura 9.2. Descarga de la última <i>release</i> del repositorio.	146
Figura 9.3. Descarga de versión concreta de Python.....	148
Figura 9.4. Versión de ejecutable de Python a descargar.....	148

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

Figura 9.5. Página de descarga de Node.js junto con npm.....	149
Figura 9.6. Ubicación generadores de preguntas.....	150
Figura 9.7. Cabecera del método de generar preguntas.....	150
Figura 9.8. Ubicación de exportadores de preguntas.....	151
Figura 9.9. Cabecera clase y método de exportación de preguntas.....	151
Figura 9.10. Esquema botones de exportación.....	151
Figura 9.11. Directorio donde se ubican los ficheros de internacionalización.....	151
Figura 9.12. Cambio de idioma.....	152
Figura 9.13. Acceso a registro e inicio de sesión.....	153
Figura 9.14. Formulario de inicio de sesión.....	153
Figura 9.15. Formulario de registro de usuarios.....	153
Figura 9.16. Vista de procesamiento de texto.....	154
Figura 9.17. Selección de palabras clave.....	154
Figura 9.18. Edición de palabras y modificación del número de preguntas a generar.	155
Figura 9.19. Vista de preguntas propuestas.....	155
Figura 9.20. Añadir preguntas.....	156
Figura 9.21. Sistema de <i>drag and drop</i> para ordenar preguntas.....	156
Figura 9.22. Botón de generación de cuestionarios.....	157
Figura 9.23. Cuestionario o test creado correctamente.....	157
Figura 9.24. Vista de resultados de cuestionarios.....	158
Figura 9.25. Formulario para unirse a test.....	158
Figura 9.26. Vista de realización del cuestionario.....	158
Figura 9.27. Vista de retroalimentación del cuestionario.....	159
Figura 14.1. Metodología para el plan de gestión de riesgos.....	168
Figura 14.2. Pregunta a GPT del libro a partir de referencia bibliográfica.....	189
Figura 14.3. Listado de palabras clave encuentra.....	189
Figura 14.4. Vista de la aplicación de escritorio de Mendeley.....	190
Figura 14.5. Extensión de Mendeley para Word.....	191

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

ÍNDICE DE TABLAS

Tabla 2.1. Resultados medios obtenidos para cada libro con el algoritmo propuesto....	30
Tabla 2.2. Resultados obtenidos para el capítulo 1 del libro US History. [17]	32
Tabla 2.3. Resultados obtenidos para el capítulo 1 libro <i>Modern World History</i> . [16]	32
Tabla 2.4. Resultados para el capítulo 1 del libro <i>From yesterday to tomorrow</i> . [14]...	33
Tabla 2.5. Resultados para el capítulo 15 del libro <i>The American Peageant</i> . [15]	33
Tabla 2.6. Tabla con el valor de F1 Score medio para cada libro con algoritmos.	34
Tabla 3.1. Comparación tecnologías para implementación del backend.	40
Tabla 3.2. Comparación tecnologías para implementación del frontend.	42
Tabla 4.1. Riesgos identificados durante el análisis.....	50
Tabla 4.2. Registro de riesgos con probabilidades de impacto de cada riesgo.....	51
Tabla 4.3. Estrategias de respuesta al riesgo.	52
Tabla 4.4. Presupuesto de costes resumido.	53
Tabla 4.5. Presupuesto de cliente resumido.	53
Tabla 5.1. Caso de uso de ver la página principal.	67
Tabla 5.2. Caso de uso de ver la página de documentación.	67
Tabla 5.3. Caso de uso de la página de contacto.	68
Tabla 5.4. Caso de uso de inicio de sesión de usuarios.....	68
Tabla 5.5. Caso de uso de registro de usuarios.....	68
Tabla 5.6. Caso de uso de carga de ficheros.....	69
Tabla 5.7. Caso de uso de carga de texto.....	69
Tabla 5.8. Caso de uso de eliminar todo el texto.....	70
Tabla 5.9. Caso de uso de mostrar palabras clave.	70
Tabla 5.10. Caso de uso de añadir palabras clave.	71
Tabla 5.11. Caso de uso de editar palabras clave.	71
Tabla 5.12. Caso de uso de eliminar palabras clave.	71
Tabla 5.13. Caso de uso de Seleccionar palabras clave.....	72
Tabla 5.14. Caso de uso de filtrar palabras clave.	72
Tabla 5.15. Caso de uso de modificar número de preguntas a generar.	72
Tabla 5.16. Caso de uso de Editar respuestas.....	73
Tabla 5.17. Caso de uso de Añadir respuestas.	73

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

Tabla 5.18. Caso de uso de Eliminar respuestas.....	74
Tabla 5.19. Caso de uso de Editar enunciados.	74
Tabla 5.20. Caso de uso de Exportar preguntas de examen.	74
Tabla 5.21. Caso de uso de mostrar preguntas.	75
Tabla 5.22. Caso de uso de Añadir preguntas.	75
Tabla 5.23. Caso de uso de Ordenar preguntas.	75
Tabla 5.24. Caso de uso de eliminar preguntas.	76
Tabla 5.25. Caso de uso de generar cuestionarios.	76
Tabla 5.26. Caso de uso de ver estado de cuestionarios.	77
Tabla 5.27. Caso de uso de cerrar cuestionarios.....	77
Tabla 5.28. Caso de uso de unirse a cuestionarios.	78
Tabla 5.29. Caso de uso de Contestar pregunta.....	78
Tabla 5.30. Caso de uso de enviar cuestionarios.	78
Tabla 5.31. Escenario de ver página principal.....	79
Tabla 5.32. Escenario de ver página de documentación.....	79
Tabla 5.33. Escenario de ver página de contacto.	80
Tabla 5.34. Escenario de registro.	81
Tabla 5.35. Escenario de inicio de sesión.....	82
Tabla 5.36. Escenario de carga de texto/fichero.....	82
Tabla 5.37. Escenario de añadir de palabras clave.	83
Tabla 5.38. Escenario de eliminar palabras clave.	84
Tabla 5.39. Escenario de editar palabras clave.....	84
Tabla 5.40. Escenario de seleccionar palabras clave.	85
Tabla 5.41. Escenario de filtro de búsqueda de palabras clave.	85
Tabla 5.42. Escenario de modificar número de preguntas para cada palabra.	86
Tabla 5.43. Escenario de mostrar preguntas.	87
Tabla 5.44. Escenario de modificar preguntas.	87
Tabla 5.45. Escenario de añadir respuestas.	88
Tabla 5.46. Escenario de eliminar respuesta.	88
Tabla 5.47. Escenario de exportar preguntas.....	89
Tabla 5.48. Escenario de ordenar preguntas.	89
Tabla 5.49. Escenario de añadir nuevas preguntas.....	90

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

Tabla 5.50. Escenario de eliminar preguntas.....	90
Tabla 5.51. Escenario de generar cuestionario.....	91
Tabla 5.52. Escenario de ver estado de cuestionario.....	92
Tabla 5.53. Escenario de cerrar cuestionario.....	92
Tabla 5.54. Escenario de unirse a cuestionario.	93
Tabla 5.55. Escenario de contestar a pregunta.	93
Tabla 5.56. Enviar cuestionario.....	94
Tabla 6.1. Situaciones de prueba para la gestión de usuarios.....	119
Tabla 6.2. Situaciones de prueba para el procesamiento del texto.	120
Tabla 6.3. Situaciones de prueba para el procesamiento de palabras clave.	121
Tabla 6.4. Situaciones de prueba para la gestión de preguntas.	123
Tabla 6.5. Situaciones de prueba para la gestión de cuestionarios.	125
Tabla 6.6. Situaciones de prueba para las pruebas con usuarios.	126
Tabla 6.7. Diseño de cuestionario previo.	127
Tabla 6.8. Cuestionario posterior.	127
Tabla 6.9. Situaciones de prueba para pruebas de adaptabilidad.	128
Tabla 8.1. Cuestionario previo usuario 1.....	135
Tabla 8.2. Cuestionario posterior del usuario 1.....	136
Tabla 8.3. Cuestionario previo del usuario 2.....	137
Tabla 8.4. Cuestionario posterior del usuario 2.....	138
Tabla 8.5. Cuestionario previo del usuario 3.....	139
Tabla 8.6. Cuestionario posterior del usuario 3.....	139
Tabla 14.1. Información de costes del personal supuesto en la empresa.	171
Tabla 14.2. Productividad y relación de costes directos e indirectos en salarios.	171
Tabla 14.3. Costes indirectos para el funcionamiento de la empresa supuesta.	171
Tabla 14.4. Costes indirectos relativos a materiales para el desarrollo del proyecto... ..	172
Tabla 14.5. Horas productivas y precio por hora con y sin beneficios.....	172
Tabla 14.6. Resumen de modelo de la empresa.	173
Tabla 14.7. Tareas necesarias para la Partida 1.....	173
Tabla 14.8. Presupuesto de costes de la partida de 1 de investigación del algoritmo..	174
Tabla 14.9. Tareas correspondientes a la Partida 2.	175
Tabla 14.10. Presupuesto de costes para partida 2 de estudios y planificación.	176

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

Tabla 14.11. Tareas relativas a la Partida 3	177
Tabla 14.12. Presupuesto de costes para la partida 3 de análisis del sistema.....	178
Tabla 14.13. Tareas relativas a la partida 4 de diseño del sistema.....	179
Tabla 14.14. Presupuesto de costes para la partida 4 de diseño del sistema	180
Tabla 14.15. Tareas relativas a la partida 5	181
Tabla 14.16. Presupuesto de costes para partida 5 de implementación y pruebas.	182
Tabla 14.17. Tareas relativas a la partida 6	183
Tabla 14.18. Presupuesto de costes de la partida 6 de documentación restante.....	184
Tabla 14.19. Presupuesto de costes de la partida 7 de otros costes.....	185
Tabla 14.20. Presupuesto de costes agregado.....	185
Tabla 14.21. Partidas e ítems de cliente.	186
Tabla 14.22. Coste para repartir entre las demás partidas.	186
Tabla 14.23. Presupuesto de coste con ponderaciones y presupuesto cliente.	186
Tabla 14.24. Presupuesto de cliente resumido.	187

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

1 ¿Qué es este trabajo?

1.1 Resumen

El presente proyecto indagará en un problema dentro del procesamiento de lenguaje natural (NLP) se refiere. Concretamente en la búsqueda de palabras clave, una tarea ardua y compleja que, aunque *a priori* pueda parecer sencilla no lo es. El trabajo mostrado a continuación es el resultado de varios meses de investigación y desarrollo en esta temática.

Primeramente, el trabajo aborda en el capítulo 2 el detalle del algoritmo desarrollado, más a modo de investigación para dar contexto de la temática. Se comienza por dar un análisis de las diferentes alternativas ofreciendo una perspectiva del estado del arte de la temática. Además de eso, también se analizan los resultados de efectividad obtenidos con el algoritmo propuesto comparándolo con otras alternativas.

Posteriormente, el resto de los capítulos del trabajo y que conforman la mayor parte del trabajo, se centrarán en el desarrollo e implementación de una herramienta web que utilice este algoritmo. Principalmente, esta herramienta tendrá el objetivo de ofrecer una alternativa rápida para profesores que deseen crear preguntas de tipo test a partir de un material textual.

La herramienta permitirá subir material docente, a partir del cual extraerá aquellas palabras clave. Del mismo modo, la herramienta permitirá refinar esta lista de palabras clave, eliminando existentes, añadiendo nuevas o modificando las existentes. El usuario seleccionará en la herramienta aquellas que considere más importantes, a partir de estas seleccionadas, posteriormente se propondrán preguntas al usuario así como las opciones de respuesta a dichas preguntas. Estas preguntas podrán exportarse a diferentes formatos, modificarse y al mismo tiempo podrán ser parte de cuestionarios creados por el usuario. Dichos cuestionarios también podrán ser configurados para dar o no retroalimentación al terminar o permitir el salto de preguntas, a estos cuestionarios también se podrá acceder desde la herramienta utilizando un código QR. Del mismo modo, la funcionalidad para que los alumnos realicen los formularios permitirá ir almacenando las respuestas en tiempo real, guardando en todo momento las respuestas que realiza el alumno. Esta herramienta también permitirá a los profesores gestionar aquellos cuestionarios creados, ver las notas obtenidas por los estudiantes, abrir o cerrar cuestionarios, visualizar los códigos de acceso, etc. Registrando así la mayor parte de información en cuanto a la realización de los cuestionarios de refiere. A lo largo del trabajo, se abordarán todas las fases típicas en el desarrollo de un proyecto informático: estudio de alternativas y viabilidad, planificación, análisis, diseño e implementación.

1.2 Palabras clave

Procesamiento de lenguaje natural, Búsqueda de *keywords*, TextRank, Python, API REST, Desarrollo web, JavaScript, React

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

1.3 Abstract

This project will be focused on a recurring problem in natural language processing (NLP). Specifically in the extraction of keywords, an arduous and complex task that, although it may seem simple at first glance, is not. The work shown below is the result of several months of research and development in this area.

Firstly, in chapter 2, the work deals in detail with the algorithm developed, more in the way of research to give context to the subject. It begins with an analysis of the different alternatives that offer a state-of-the-art perspective on the subject. In addition to that, the results obtained for the proposed algorithm are also analysed to give a perspective on the effectiveness of the algorithm.

Subsequently, the rest of the chapters of the project will focus on the development and implementation of a web tool that uses this algorithm. Mainly, this tool will aim to offer a quick alternative for teachers who wish to create test-type questions from textual material.

The tool will allow the uploading of teaching material, from which it will extract those keywords. In the same way, the tool will allow refining this list of keywords, eliminating existing ones, adding new ones, or modifying the existing ones. The user will select in the tool those that he/she considers most important, from these selected ones, questions will be proposed to the user as well as the answer options to these questions. These questions can be exported to different formats, modified and at the same time they can be part of questionnaires created by the user. These questionnaires can also be configured to give or not to give feedback at the end or to allow skipping of questions, these questionnaires can also be accessed from the tool using a QR code. In the same way, the functionality for students to fill in the forms will allow the answers to be stored in real time, saving the answers every time the student makes an answer. This tool will also allow teachers to manage the questionnaires created, see the marks obtained by the students, open or close questionnaires, visualise access codes, etc. Throughout the work, all the typical phases in the development of an IT project will be fulfilled: study of alternatives and feasibility, planning, analysis, design, and implementation.

1.4 Keywords

Natural Language Processing (NLP), keyword extraction, TextRank, Python, API REST, Web development, JavaScript, React

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

2 Estudio del algoritmo

2.1 Contexto en que se enmarca la extracción de keywords

A continuación, se explicará el ámbito de estudio del trabajo donde se encuadra la problemática de la búsqueda de palabras clave, así como una breve descripción de la extracción de *keywords*.

2.1.1 Natural Language Processing (NLP)

El procesamiento de lenguaje natural es una tarea compleja, principalmente por la ambigüedad de los lenguajes que utilizamos diariamente. La búsqueda de *keywords* se encuadra en este ámbito ya que trata principalmente con textos y datos que no dejan de estar escritos en un lenguaje natural. El español, el inglés o en general, cualquier idioma que se nos ocurra es ambiguo ya que tiene expresiones con doble sentido, ironías, sarcasmo, sinónimos, emociones, etc. Parámetros que son muy difíciles de controlar programáticamente. A pesar de su dificultad, existen múltiples herramientas y empresas que centran sus esfuerzos en este ámbito: Google con sus traductores y APIs, MonkeyLearn o Amazon con su Amazon Comprehend.

2.1.2 Búsqueda de keywords

La búsqueda de *keywords* se utiliza en multitud de campos como detección de spam, clasificación de información, análisis de sentimientos, entre muchos otros. A poco que se busque por internet, nos podemos encontrar miles de propuestas que utilizan diferentes técnicas para llevar a cabo esta labor, muchas de estas propuestas suelen estar destinadas a un objetivo concreto. Es común, la modificación de algoritmos populares, para adecuarlos la extracción de *keywords* en textos de una temática concreta. Algunos ejemplos podrían ser los algoritmos de LinkedIn que especializados en la extracción de aptitudes profesionales o algoritmos de detección de spam en Gmail, especializados en la búsqueda de palabras relacionadas con temáticas comerciales.

Esto se debe principalmente a que una palabra dependiendo del contexto, puede ser o no palabra clave. Una palabra clave, se define por aquella palabra que es capaz de definir precisamente un texto [1]. Por ejemplo, en un libro sobre los antiguos griegos que traten el origen de las primeras teorías heliocéntricas, la palabra “Sol” tendrá más relevancia ahí, que en el contexto de un artículo científico sobre la fusión nuclear en la superficie del Sol donde habrá otro tipo de palabras clave más específicas.

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

2.1.3 Alternativas para la extracción de *keywords*

A continuación, se busca revisar diferentes métodos y enfoques que pueden resolver el problema de la búsqueda de palabras clave. Los enfoques se han agrupado en grandes grupos según exponen los autores en [2]:

2.1.3.1 Modelos basados en estadísticas

Este tipo de enfoques se basan normalmente en las estadísticas del texto, suelen ser técnicas de TF-IDF (*Term Frequency – Inverse Document Frecuency*), coocurrencias de palabras, algoritmos tipo RAKE (*Rapid Automatic Keyword Extraction algorithm*) [3], o tipo YAKE (*Yet Another Keyword Extractor*) [4], PAT Tree (*Patricia Tree; a suffix tree or position tree*) [2].

Por explicar uno de ellos y entender el enfoque, TF-IDF introducido por Karen Spärk-Jones, es un enfoque basado en estadísticas. Este enfoque determina la relevancia de un término de forma directamente proporcional a la frecuencia de aparición en un documento y de forma inversamente proporcional al número de documentos en que aparece [5].

Para entenderlo con un ejemplo. Véase que tenemos un conjunto de documentos sobre fútbol de segunda división, si en todos estos documentos tenemos palabras que se repiten como fútbol, árbitro, estadio, etc. Estas pierden relevancia ya que no son representativas y no son clave. Sin embargo si tenemos, que una palabra como “Real Oviedo” que se repite mucho en un mismo documento y no es común en el resto de los documentos, esta palabra sería muy relevante en dicho documento y por tanto sería clave.

A continuación, se exponen diferentes ventajas e inconvenientes de dichos modelos:

2.1.3.1.1 Ventajas

- No dependen del idioma ni del dominio.
- Suelen ser algo más rápidos.
- No guardan muchos datos en memoria.

2.1.3.1.2 Desventajas

- Por lo general, suelen ser poco efectivos [2].
- Tiende a fijarse más en número de veces que aparece una palabra en relación con el resto y no tanto en otras características de la palabra.

2.1.3.2 Modelos basados en grafos

Este enfoque es una alternativa que basa su potencia en las relaciones entre palabras y cómo estas están distribuidas a lo largo del texto. Un enfoque popular es representar los textos como grafos no dirigidos donde cada nodo es una palabra. Cada una de estas se

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

asignará con una puntuación de forma que pueda tener una calificación numérica en relación con el resto. Se explicará en detalle posteriormente.

2.1.3.2.1 Ventajas

- Permite almacenar el texto como una estructura de grafo, esto permitiría el uso de algoritmos de procesamiento orientados a grafos.
- Permite conocer relaciones de una palabra con todas las demás del texto y no solo con algunas.
- Podrían incluir dependencias con el idioma y sus estructuras lingüísticas específicas.
- Pueden tener la capacidad de aprender y adaptarse a partir de los datos.

2.1.3.2.2 Desventajas

- En textos grandes la creación de las estructuras de datos puede llegar a tomar mucho tiempo.
- Pueden tender a generar demasiadas palabras clave, aunque no siempre es así.
- En textos cortos, con pocas relaciones entre palabras, los enfoques basados en grafos es posible que experimenten problemas para extraer aquellas palabras de mayor relevancia.
- Suelen ser dependientes de diferentes parámetros como puede ser la ventana elegida, la cual está explicada en el apartado 2.2.3.

2.1.3.3 Modelos basados en aprendizaje automático

Esta alternativa ofrece la potencia de las técnicas de aprendizaje automático y de los clasificadores. El uso de máquinas de vector soporte por ejemplo como se comenta en [6]. En este método se interpreta cada una de las instancias (en este caso las palabras) como puntos de un espacio n-dimensional. Cada instancia se etiqueta de una forma de modo que exista una frontera que separe ambos tipos. Una forma de formalizar esta división es la que ofrecen en [6] donde se etiquetan las palabras clave en “*Good keyword*”, “*Indifferent keyword*” y “*Bad keyword*”. También enfoques basados en la teoría probabilista de Bayes para la solución del problema de la extracción de *keywords* [7]. También existen métodos basados en aprendizaje profundo de redes neuronales [8]–[10]. Un algoritmo que se probará más adelante será el denominado como KeyBERT, cuyo código se encuentra en la referencia [11].

2.1.3.3.1 Ventajas

- Una vez entrenados, este tipo de sistemas suelen ser más eficientes que otras alternativas.
- Ofrece resultados de cercanos al 22% de efectividad en cuanto a F1 score se refiere, tal y como comentan los autores en [6] en algunos tipos de textos. El F1

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

es una medida que permite evaluar la efectividad de diferentes sistemas, este se explica en el apartado 2.3.3.

- Permite identificar instancias mejor clasificadas y peor clasificadas ya que se pueden ubicar en el plano n-dimensional.

2.1.3.3.2 *Desventajas*

- Dependen principalmente de los datos de entrenamiento que se utilicen. Unos datos de entrenamiento que no sean buenos pueden derivar en malos resultados. Es por ello por lo que una gran desventaja es el tiempo que se necesita invertir en el análisis y refinamiento del conjunto de datos. Además, se debe tener en cuenta que este conjunto de datos jamás representará plenamente el mundo real.
- Puede ser difícil de interpretar.
- Impide manejar el texto como una estructura organizada.
- Requieren grandes conjuntos de datos etiquetados para entrenar el modelo que no siempre son fáciles de conseguir, puede requerir un gran esfuerzo.
- Normalmente, pueden depender del idioma y del dominio del texto.

2.1.3.4 Enfoques basados en análisis lingüísticos

Este tipo de enfoques incluyen otro tipo de técnicas como son el análisis léxico, sintáctico y se centran en la estructura del texto [2]. Suelen basarse en el entendimiento del lenguaje, así como sus estructuras gramaticales. Estas características les emplean para identificar aquellos aspectos del texto de mayor relevancia.

2.1.3.4.1 *Ventajas*

- Dependen únicamente del texto sobre el que trabajan, con lo cual, evitan la dependencia de conjuntos de datos etiquetados.
- Hacen uso de categorías semánticas, suelen tener capacidad para detectar las diferentes categorías gramaticales, sinónimos, etc.
- Pueden ser capaces de identificar diferentes estructuras lingüísticas.
- Al utilizar reglas y patrones lingüísticos, estos enfoques suelen ser más comprensibles e interpretables.

2.1.3.4.2 *Desventajas*

- Suelen ser difíciles de mantener, en el sentido de que requieren adaptaciones manuales para cada tipo de texto.
- Requieren de técnicas complejas de NLP con lo cual son difíciles de implementar [2]. Normalmente, conocimientos lingüísticos expertos para el desarrollo de estructuras y reglas lingüísticas efectivas.
- Dependen, en gran medida, del idioma en el que esté escrito el texto. Esto es debido a que cada idioma tiene sus propias estructuras particulares y características únicas.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

- Al depender solo del texto, pueden ser sensibles al contexto y al estilo de escritura del texto. Esto puede hacer que los resultados obtenidos por este tipo de enfoques varíen y sean enfoques difíciles de adaptar a diferentes dominios.

2.2 Algoritmo desarrollado basado en TextRank

A pesar de las múltiples técnicas para abordar y encontrar una solución a esta problemática, en este trabajo se ha optado por el estudio e implementación de un algoritmo basado en ranking de palabras. Concretamente, basado en grafos (véase 2.1.3.2). Cada palabra del texto se calificará con una puntuación, la cual determinará aquellas palabras que son más centrales y por tanto son las que son más susceptibles de ser palabras clave. En los subapartados venideros se indagará más en detalles técnicos de la implementación, así como un estudio con los resultados obtenidos.

La estructura del algoritmo tiene tres etapas bien diferenciadas, las cuales serían las que siguen a continuación:

- **Procesamiento del texto:** En esta etapa, el algoritmo adecuará el texto a lo que vendrá después, eliminará palabras vacías, asignará categorías gramaticales, etc.
- **Creación del grafo:** en esta etapa se creará una estructura de datos que representará las palabras y las relaciones entre estas.
- **Cálculo de la puntuación:** esta será la última etapa que asignará un valor numérico a palabras y conjuntos de palabras de forma que podamos determinar las más relevantes.

2.2.1 ¿Qué es TextRank y de dónde viene?

TextRank no deja de ser una aplicación a textos de lo que el algoritmo PageRank [12] hace con los hipertextos de una página web. PageRank es capaz de dar una puntuación a cada página web en base a sus hiperenlaces, y no al contenido textual de las mismas. Es un algoritmo que entiende la web como un grafo donde los sitios web son los nodos, y los hiperenlaces son las aristas que unen nodos entre sí. Es intuitivo pensar que, si una página web es referenciada por muchas otras, esta será por ende de mayor relevancia. Véase, un ejemplo con la web del Ministerio de Asuntos Exteriores y la de un blog sobre tratados de España en el exterior. La primera será de mayor relevancia para el algoritmo ya que estará referenciada por páginas web de muchos otros países que enlacen a la información oficial publicada por el Ministerio. De esta forma, el blog tendrá muchas menos referencias y será de menor relevancia.

La misma filosofía puede aplicarse análogamente a las páginas web y los hiperenlaces tendríamos que las palabras del texto serían los nodos y las coocurrencias de palabras las aristas del grafo. Estas aristas tendrán un peso el cual indicará la fuerza o intensidad de la conexión, aspecto que no tiene PageRank.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

Con el fin de entenderlo gráficamente se puede observar en la imagen de la figura a continuación, las palabras que conforman los nodos como *systems*, *compatibility*, *criteria*, etc. así como las aristas que aparecen como consecuencia de las coocurrencias de palabras en el texto. Como, por ejemplo, la arista de *systems* a *compatibility* dado que aparecen casi seguidas al inicio del texto.

Compatibility of systems of linear constraints over the set of natural numbers.
 Criteria of compatibility of a system of linear Diophantine equations, strict inequations, and nonstrict inequations are considered. Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given.
 These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types systems and systems of mixed types.

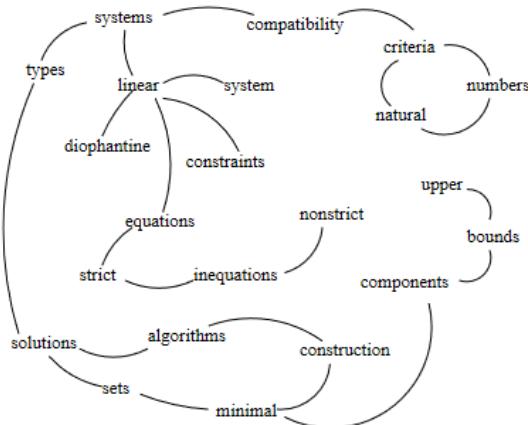


Figura 2.1. Ejemplo de grafo formado para un texto. Fuente: [13]

2.2.2 Procesamiento del texto

Tal y como nos podemos imaginar las posibilidades de texto¹ que un usuario puede adjuntar son infinitas, asimismo, cada texto puede contener muchos datos inútiles. Es por ello por lo que no podemos construir el grafo del texto sin antes hacer un procesado de este. El esquema seguido se muestra con el pseudocódigo de la figura siguiente.

Algorithm 1 Process Text

```

clean(rawText)                                ▷ Removes non-printable chars and turns to lowercase
tokenize(cleanedText)                         ▷ Assigns grammatical categories
lemmatizer(tokenizedText)                     ▷ Reduce words to their lemma
tokenize(cleanedText)
setStopwords()
filterWords(lemmatizedText)                  ▷ Removes stopwords
createVocabulary()                           ▷ Deletes duplicates from filtered text
  
```

Algoritmo 1. Procesado del texto.

¹ Inicialmente el algoritmo se desarrollará para ofrecer buenos resultados en textos en Inglés.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

En el pseudocódigo del Algoritmo 1. Procesado del texto. aparecen numerosas tareas típicas en el procesamiento de textos. Por destacar algunas de ellas la lematización se encargará de hallar los lemas de las palabras. El lema es la forma que por convenio se acepta como representante de todas las formas de una misma palabra. Por ejemplo, lematizar la palabra “amiguitas”, significaría obtener “amigo” como lema. Otra técnica importante es la obtención de *stopwords* o palabras vacías, palabras que carecen de significado como artículos, pronombres, preposiciones, etc. Esta lista de palabras vacías se ha obtenido principalmente del paquete *spacy* de Python y de palabras del propio texto de entrada. el algoritmo propuesto incorpora nuevas palabras vacías en base a la categoría gramatical. Es decir, si el algoritmo solo busca palabras clave como sustantivos, si encuentra un adjetivo también lo añadiría como palabra vacía, esto permite identificar palabras más importantes que otras en base al tipo de texto.

Tal y como nos podemos imaginar la mayoría de las palabras clave son nombres propios referidos a personas o a lugares, acontecimientos, etc. Básicamente sustantivos o nombres. De este modo, tras procesar el texto y obtener el respectivo vocabulario, este contendrá únicamente sustantivos. De este modo, se consigue acotar mucho el conjunto de palabras y se puede ser más preciso a la hora de identificar las palabras clave en cuestión.

2.2.3 Creación del grafo

El siguiente paso es el de creación del grafo, este paso se puede representar de múltiples formas, en este caso se ha optado por representarlo a través de una matriz cuadrada como la que sigue a continuación. La dimensión de la matriz vendría dada por el número de palabras del vocabulario obtenido en el paso anterior. Asimismo, una posición ij de la matriz representaría la arista entre dos nodos o palabras. Hay enfoques que optan por la creación de una clase grafo con una lista de objetos nodo, sin embargo, esta alternativa es más lenta. Utilizar esta estructura con matrices a la hora de crear el grafo ofrece una mayor velocidad de acceso a los datos, lo cual redundaría en mayor eficiencia sobre todo en textos largos.

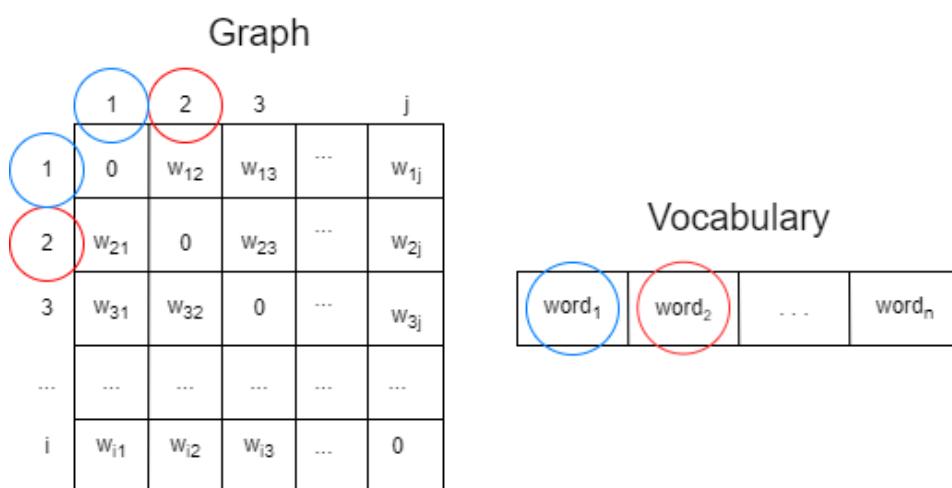


Figura 2.2. Esquema del grafo creado.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

El valor w_{ij} de cada casilla será el peso de cada arista. Un mayor peso significaría mayor número de coocurrencias entre la palabra i y la j. Del mismo modo, el grafo es no dirigido con lo cual la matriz sería simétrica, con lo cual se cumple que $w_{ij} = w_{ji}$. Ni que decir tiene que las aristas de una palabra a sí misma tampoco se contemplan. El cálculo del peso se hace de la siguiente forma:

Algorithm 2 Calculate weights

```

Require: windowValue > 0                                ▷ This number represents n-grams, usually 3
    for word1 in vocabulary do
        word1.score  $\leftarrow$  1                               ▷ Initialize the score of the word
        for word2 in vocabulary do
            if word1 is not equal word2 then
                for Every n-gram in text processed do
                    if word1 and word2 in n-gram then
                        weight[word1][word2]  $\leftarrow$  increment
                    end if
                end for
            end if
        end for
    end for

```

Algoritmo 2. Cálculo de pesos.

Un n-grama es una subsecuencia de n elementos de una secuencia dada. Esto nos permite encontrar coocurrencias en todo el texto y por tanto calcular los pesos. El proceso es similar al siguiente, ejemplificaré con el texto de la Figura 2.1. Ejemplo de grafo formado para un texto. Fuente: ya habiendo realizado el procesado previo. Para cada par de palabras distintas en el vocabulario comprueba si aparecen en cada n-grama, en este caso trigramas.

The diagram shows a text block with several words highlighted by red boxes. The highlighted words are: compatibility, system, linear, constraint, set, natural, number, criterion, compatibility, system, linear, diophantine, equation, strict, inequations, nonstrict, inequations, upper, bound, component, minimal, set, solution, algorithm, construction, minimal, generating, set, solution, type, system, criterion, corresponding, algorithm, constructing, minimal, supporting, set, solution, solving, type, system, system, mixed, type.

Figura 2.3. Ejemplo de los trigramas que va tomando el algoritmo.

2.2.4 Cálculo de la puntuación

La puntuación es lo que permitirá identificar aquellas palabras de mayor relevancia en el texto. La puntuación se calcula estableciendo un número de iteraciones. Estas iteraciones pueden llegar o no a ejecutarse ya que llegará un momento en el que las puntuaciones se stabilicen y converjan.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

La puntuación se calcula tal y como se ve en el pseudocódigo siguiente, donde depende de un valor (*summation*). Este valor depende de todas las palabras que tienen conexión con la palabra para la que se calcula la puntuación, así como un *dampingFactor* (0.85 normalmente). Este valor indica que hay un 85% de probabilidad de saltar a otro nodo aleatorio, evitando que el algoritmo quede de forma indefinida en algunos ciclos del grafo (ver *Random Surfer Model*). [12]

Algorithm 3 Calculate scores

```

Require: iterations > 0
Require: dampingFactor ∈ [0, 1]
Require: threshold > 0
for i in iterations do
    for word1 in vocabulary do
        for word2 in vocabulary do
            if weight[word1][word2] is not 0 then
                summation ← (weight[word1][word2]/weight[word2][*]) * word2.score
            end if
        end for
        word1.score ← (1 - dampingFactor) + dampingFactor * summation
    end for
    if abs(lastScores - newScores) ≤ threshold then
        break
    end if
end for

```

Algoritmo 3. Cálculo de puntuaciones.

A pesar de que este primer paso de calcular la puntuación para cada una de las palabras es útil y permite identificar varias *keywords*. Lo cierto es que hay muchas otras palabras clave que no están formadas por una sola, sino por conjuntos de estas. Por ejemplo, en un texto sobre historia, lo suyo sería identificar como palabra clave “Marco Polo” en lugar de “Marco” por un lado y “Polo” por otro, y así con muchas otras palabras como “Reyes Católicos”, “Bartolomé de las Casas”, etc.

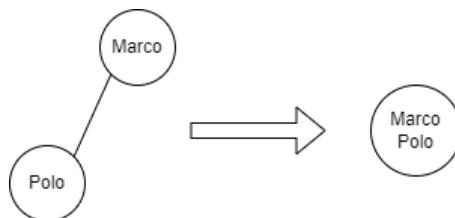


Figura 2.4. Generación de nuevos nodos a partir de varias palabras.

Este proceso puede llevarse a cabo de muchas formas, por ejemplo, podemos hacerlo a través de umbrales que detecten cuándo el peso de una arista es suficientemente grande, de forma que se reorganice el grafo. También otra opción es utilizar las palabras vacías como delimitadores de forma estas puedan definir hasta dónde llega cada una de las palabras compuestas. De esta forma podemos iterar sobre el texto de forma que se vayan generando palabras compuestas.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

Algorithm 4 Create keyphrases

```

for word in lemmatizedText do
    if word is in stopwords then
        if newKeyPhrase is not empty and newKeyPhrase in Text then
            candidateKeyPhrases.append(newKeyPhrase)
        end if
        newKeyPhrase ← ∅
    else if word is not in stopwords then
        newKeyPhrase ← newKeyPhrase + word
    end if
end for

```

Algoritmo 4. Creación de *Keyphrases*.

Una vez obtenidas estas nuevas combinaciones de palabras será necesario recalcular el *score* de estas. Para ello, en este caso se sumará el *score* de cada uno de los términos que conforman la nueva palabra.

2.2.5 Diagrama de componentes del algoritmo

En este apartado, se expondrá un diagrama de los componentes que componen la estructura del algoritmo de una forma más visual.

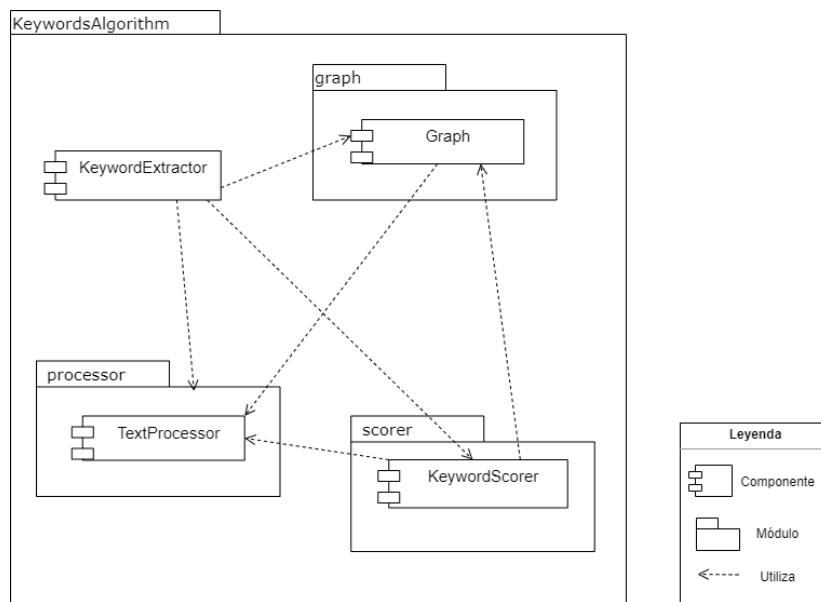


Figura 2.5. Diagrama de componentes del algoritmo.

Cada uno de los componentes encapsula las siguientes funcionalidades:

- **KeywordExtractor**: es el punto de inicio del algoritmo. Utilizado en la fase de análisis para calcular algunos resultados y obtener ficheros de datos.

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

- **TextProcessor:** componente encargado del procesamiento del texto de entrada. Se encarga de tareas redactadas en la primera fase mencionada en el apartado 2.2.2 sobre procesamiento del texto.
- **Graph:** este componente se encarga principalmente del almacenamiento del grafo que representa el texto. Para más información véase la sección 2.2.3 sobre la creación del grafo.
- **KeywordScorer:** este componente se apoya en el anterior para calcular las puntuaciones de las palabras más relevantes. Véase apartado 2.2.4 sobre cálculo de puntuaciones.

2.3 Análisis de resultados

El siguiente paso, una vez explicada la estructura del algoritmo, es el de analizar los resultados, es decir, cómo de efectivo es el algoritmo a la hora de hacer su labor. Para ello se usarán medidas que son de interés para medir la efectividad de sistemas en materia de recuperación de información, clasificadores... Asimismo, para medir esta efectividad se usarán unos textos de prueba, los cuales, se pueden ver en la bibliografía en las referencias [14]–[17]. Cada uno de estos libros están organizados en capítulos los cuales están en ficheros de texto. Para cada capítulo le acompaña un fichero de texto que almacena las palabras clave identificadas por la editorial. Con lo cual podemos evaluar la efectividad del algoritmo.

2.3.1 Precisión

La precisión mide el número de aquellos elementos que son relevantes respecto a todos los que se han encontrado. En otras palabras y aplicado a palabras clave, es el cociente de las palabras que encuentra y que realmente son palabras clave (*true positives*) entre el total de palabras encontradas (*true positives + false positives*).

2.3.2 Recall o Sensitivity

El *recall* mide de algún modo cuántos elementos se han encontrado entre el total de los que debería haber encontrado. Básicamente el cociente de las palabras que encuentra y que realmente son palabras clave (*true positives*) entre el total de palabras que debería haber encontrado (*true positives + false negatives*).

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

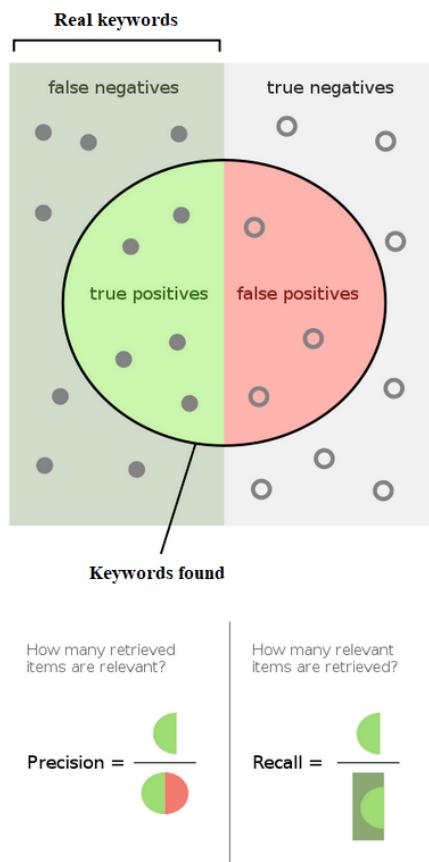


Figura 2.6. *Precision* y *Recall* visto gráficamente [18]

2.3.3 F1 Score

El valor de *F1 Score* no deja de ser la media armónica de la precisión y el *recall* vistos anteriormente. Se calcula mediante la siguiente fórmula:

$$F_1 = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} = 2 \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{2\text{tp}}{2\text{tp} + \text{fp} + \text{fn}}$$

Ecuación 1. Cálculo del F1 Score [18]

Esta medida, al igual que las anteriores nos da un valor entre 0 y 1 de forma que cuanto más cerca de 1, los resultados serán mejores y el algoritmo será capaz de encontrar mayor número palabras claves.

2.3.4 Análisis de resultados

El siguiente paso en la investigación y análisis del algoritmo es la evaluación de los resultados obtenidos. Para esta evaluación se han calculado los diferentes valores numéricos presentados en las secciones anteriores para los libros mostrados en las

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

referencias [14]–[17]. En las tablas a continuación se muestran los resultados obtenidos en cada experimento con el algunos de los algoritmos y métodos de búsqueda de *keywords* mostrados en el apartado 2.1.3 sobre alternativas para la búsqueda de palabras clave.

```
main() > for name, iteration
: main x
C:\Users\diego\OneDrive\Escritorio\keywordsAlgor
Keywords found for USHist_0
TP:11
FN:10
FP:532
Precision: 0.020257826887661142
Recall: 0.5238095238095238
F1Score: 0.03900709219858156
```

Figura 2.7. Ejecución de resultados para el capítulo 1 del libro *US History*. [17]

Para el algoritmo propuesto, en cada uno de los libros especificados en las referencias anteriores ([14]–[17]) obtenemos los siguientes resultados medios. Esta media se corresponde con la media de todos los capítulos de cada uno de los libros de historia:

	TP	FP	FN	Precisión	Recall	F1Score
US History	7,90625	612,4375	15,0625	0,012831	0,359805	0,024716
Modern World History de Cambridge	8,8	419,5	18,5	0,023965	0,315634	0,043994
From yesterday to tomorrow	8,833333	251	15,33333	0,036537	0,373839	0,066182
The American Peageant	4,472222	677,25	21,91667	0,006478	0,16348	0,012424

Tabla 2.1. Resultados medios obtenidos para cada libro con el algoritmo propuesto.

Con el fin de darle una mayor amplitud al trabajo también se ha decidido comparar los resultados con tecnologías emergentes hoy en día así como otro tipo de algoritmos conocidos para la extracción de palabras clave, estos son los siguientes:

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

- ChatGPT impulsado por OpenAI. En el anexo 14.7 se puede consultar más información.
- Se comparará con un sitio web de referencia que permite extraer palabras clave de textos educativos denominado QuillionZ y que sirve también para la generación de preguntas de examen, de este se puede extraer información en el apartado 3.1.1.
- El algoritmo TextRank sin modificaciones, el cual se ha utilizado siguiendo la implementación de la librería summa de Python, cuya referencia se puede encontrar en [19].
- Se probará también con un algoritmo basado en aprendizaje automático denominado KeyBERT, el cual está disponible en la referencia[11]. Este algoritmo permite tanto extraer palabras clave como frases clave. Esto permite comparar también con otro enfoque diferente a la propuesta. En este caso se probará con las características por defecto pero modificado el valor del parámetro “*top_n*” a -1 para que saque todas las palabras posibles.
- El algoritmo RAKE utilizado es el propio de Python que está implementado en la librería nltk y disponible su documentación en [3].
- El algoritmo YAKE es un algoritmo fácilmente configurable de la librería yake de Python. Este último se ha probado con diferentes configuraciones. La configuración por defecto y una personalizada en la que se amplía el número de *keywords* a obtener de 20 a 500, así como el tamaño de la ventana que se explica en la Figura 2.3, que pasa de 1 a 3. Esta configuración personalizada se indica en las tablas sucesivas por “*custom*”. Este está disponible en [20].

Los datos mostrados en las tablas representan los resultados obtenidos de cada algoritmo y herramienta. En este caso se ha decidido optar por poner 4 resultados, concretamente un capítulo de cada libro:

	TP	FP	FN	Precisión	Recall	F1Score
Algoritmo propuesto	11	532	10	0,0203	0,5238	0,04
TextRank simple	3	3262	18	0,0009	0,1429	0,0018
KeyBERT	13	1924	8	0,0067	0,619	0,0133
RAKE	12	2607	9	0,0046	0,5714	0,0091
YAKE (default)	0	0	21	0	0	0
YAKE (custom)	1	499	20	0,002	0,0476	0,0038

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

ChatGPT (Feb 13)	0	12	21	0	0	0
QuillionZ	1	46	20	0,0217	0,0476	0,0299

Tabla 2.2. Resultados obtenidos para el capítulo 1 del libro *US History*. [17]

	TP	FP	FN	Precisión	Recall	F1Score
Algoritmo propuesto	10	204	20	0,0467	0,3333	0,082
TextRank simple	2	1048	29	0,0019	0,0645	0,0037
KeyBERT	10	822	21	0,012	0,3226	0,0232
RAKE	16	1295	14	0,0122	0,5333	0,0239
YAKE (default)	0	20	30	0	0	0
YAKE (custom)	5	495	26	0,01	0,1613	0,0188
ChatGPT (Feb 13)	1	20	29	0,0476	0,0333	0,0392
QuillionZ	2	44	28	0,0435	0,0667	0,0526

Tabla 2.3. Resultados obtenidos para el capítulo 1 libro *Modern World History*. [16]

	TP	FP	FN	Precisión	Recall	F1Score
Algoritmo propuesto	8	114	10	0,0656	0,4444	0,1143
TextRank simple	7	791	11	0,0088	0,3889	0,0172
KeyBERT	9	509	9	0,0173	0,5	0,0336
RAKE	10	656	8	0,015	0,5172	0,0292

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO		
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es					

YAKE (default)	0	20	18	0	0	0
YAKE (custom)	7	493	11	0,014	0,3889	0,027
ChatGPT (Feb 13)	1	37	17	0,0263	0,0555	0,0357
QuillionZ	1	45	17	0,0217	0,5556	0,0313

Tabla 2.4. Resultados para el capítulo 1 del libro *From yesterday to tomorrow*. [14]

	TP	FP	FN	Precisión	Recall	F1Score
Algoritmo propuesto	15	855	32	0,0172	0,3191	0,0327
TextRank simple	5	5735	42	0,0009	0,1064	0,0017
KeyBERT	7	3127	40	0,0022	0,1489	0,0044
RAKE	21	3556	26	0,0058	0,4468	0,0115
YAKE (default)	0	20	47	0	0	0
YAKE (custom)	1	499	46	0,002	0,0212	0,0037
ChatGPT (Feb 13)	1	28	46	0,0345	0,0213	0,0263
QuillionZ	2	43	45	0,4444	0,0426	0,0435

Tabla 2.5. Resultados para el capítulo 15 del libro *The American Peageant*. [15]

2.3.5 Conclusiones obtenidas del estudio

A pesar de que se han seleccionado una serie de capítulos, la tónica general es que se obtienen resultados pobres, principalmente debido a la gran longitud de los textos (aproximadamente 7000 palabras) lo cual hace difícil identificar las palabras clave. Muchas palabras clave son específicas, asimismo, combinar esto con el hecho de reducir

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

el número de palabras candidatas se hace difícil. Aun así, el algoritmo propuesto es el que mejor resultados ofrece en la mayoría de los casos.

Si nos fijamos en YAKE, por ejemplo, apenas es capaz de encontrar palabras clave. RAKE por el contrario funciona bastante bien. Obtiene más palabras clave respecto a la propuesta y además es muy rápido, sin embargo, genera una gran cantidad de falsos positivos. De hecho, el número de palabras clave que propone es casi la mitad de las palabras de las que tiene el texto. El algoritmo propuesto tiene un número de falsos positivos considerablemente menor.

Con respecto a KeyBERT, este obtiene buenos resultados, sin embargo, también peca de optimista ya que el número de falsos positivos es notablemente grande. A pesar de ello obtiene bastante bien las palabras clave de cada uno de los capítulos. El algoritmo TextRank sin modificaciones, no es excesivamente bueno, ya que en muchos casos el número de falsos positivos es excesivo en relación con la longitud del texto.

	USHist [17]	Modern World History [16]	From yesterday to tomorrow [14]	The American Peageant [15]
Algoritmo propuesto	0,02	0,04	0,07	0,008
TextRank simple	0,0009	0,0007	0,0106	0,0055
KeyBERT	0,005	0,015	0,03	0,002
RAKE	0,008	0,014	0,02	0,007
YAKE (default)	0,001	0	0,007	0
YAKE (custom)	0,004	0,028	0,021	0,0018

Tabla 2.6. Tabla con el valor de F1 Score medio para cada libro con algoritmos.

En esta última tabla figuran las medias aritméticas de los F1 Score obtenidos para cada uno de los libros con cada uno de los algoritmos y tecnologías que se han probado. Si nos fijamos la media del f1 score es prácticamente la mejor en nuestra propuesta.

Desde el punto de vista de los resultados obtenidos, el algoritmo propuesto es el que mejor balancea el número de palabras encontradas respecto al número de palabras que son clave. Es decir, verdaderos positivos (tp) respecto de falsos positivos (fp). Es cierto que hay algunos que encuentran mayor número de palabras clave como es el caso de RAKE, sin embargo, obtienen una cantidad ingente de palabras candidatas. Es por ello por lo que el algoritmo propuesto, a pesar de sacrificar en cierta medida, el número de palabras clave,

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

obtiene una menor cantidad de falsos positivos. Con respecto a los libros en los que se ha probado, el algoritmo propuesto es, al menos, un 43% más efectivo que la segunda mejor propuesta en cuanto a f1 score medio se refiere, concretamente en el caso del segundo libro entre el algoritmo propuesto y la opción de YAKE *custom*.

A pesar de no tener datos para todos los capítulos, en algunos casos, QuillionZ y ChatGPT obtiene resultados decentes en los capítulos que se ha probado, pero eso es principalmente debido a que obtienen pocas palabras clave.

A partir de ahora, se llevará a cabo un desarrollo de una herramienta para la generación de preguntas de examen. Esta herramienta, utilizará el algoritmo propuesto para buscar palabras clave y con estas, proponer preguntas. De esta forma se obtendrá una herramienta que ponga en valor lo visto hasta ahora en el estudio.

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

3 Estudio de Viabilidad del Sistema

3.1 Análisis de sistemas similares

En este apartado se describirán numerosos sistemas que existen hoy en día. Varios de ellos ofrecen una funcionalidad similar al sistema que se desarrollará y otros no. El fin de este análisis y valorar ventajas y desventajas del sistema a desarrollar con los existentes hoy en día.

3.1.1 QuillionZ (<https://www.quillionz.com/>)

QuillionZ es una página web que implementa la mayor parte de la funcionalidad a desarrollar. Ofrece un sistema, según hablan ellos, de inteligencia artificial para crear preguntas. También dispone de APIs de apoyo para desarrolladores. Al igual que la herramienta para desarrollar, QuillionZ funciona solo en inglés.

3.1.1.1 Ventajas de QuillionZ

- Una de las ventajas es que ofrece más funcionalidades aparte de solo la generación de preguntas, sino que también permite incluir notas a los usuarios
- QuillionZ permite la creación de más de un tipo de preguntas y no solo las de rellenar con la palabra que falte.
- Permite varios formatos de exportación para las preguntas.

3.1.1.2 Ventajas del nuevo sistema respecto QuillionZ

- Utiliza un algoritmo claramente definido, adaptable y con posibilidades de modificación. QuillionZ no comenta nada sobre cómo obtiene las palabras clave (solo pone “*AI-powered platform*”)
- Evita el uso de la selección de temáticas, QuillionZ es una herramienta que, mediante un *combobox*, permite seleccionar la temática del texto. Esto puede indicar que use listas de palabras clave predefinidas, lo que puede agregar sesgos a los resultados.
- Carece de sistemas de uso por suscripción. QuillionZ tiene funcionalidades limitadas por un sistema de *pricing*. Este sistema tiene diferentes tipos de suscripciones, por ejemplo dispone de una gratuita que está bastante limitada, ya que solo permite generar 5 bloques de preguntas al mes o solo permite un tipo de exportación de las preguntas. También tiene una suscripción de 29,99\$ cada tres meses la cual amplia las funcionalidades por ejemplo bloques ilimitados de preguntas. Además también la más potente de las suscripciones sería la de 14,99\$ al mes la cual elimina los anuncios por completo. Para el pago básicamente es necesario tener una cuenta y registrar una tarjeta de crédito.

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

- Permite la carga de ficheros directamente además de la escritura de texto plano.
- Permite la realización de cuestionarios integrados en la propia aplicación.

3.1.2 Quizgecko (<https://quizgecko.com/>)

Es una herramienta para la creación de preguntas a partir de texto. Utiliza IA para la creación de las preguntas. La forma de interactuar con esta aplicación es similar a la anterior y a la que se desarrollará en este trabajo.

3.1.2.1 Ventajas de Quizgecko

- Permite varios tipos de preguntas, no solo de rellenar con la palabra que falte.
- Permite elegir la dificultad de las preguntas.
- Permite cargar ficheros desde un URL.
- Dispone de opciones de autodetección del lenguaje, con lo cual le da bastante flexibilidad

3.1.2.2 Ventajas del nuevo sistema respecto Quizgecko

- Carece de sistemas de suscripción que limitan las funcionalidades de la aplicación
- Dispone de un algoritmo claramente definido que es la base del sistema de generación de preguntas. En este caso Quizgecko, comenta que utiliza IA a partir de la más reciente ciencia de aprendizaje.
- Dispone de una API que puede servir a desarrolladores para otros tipos de sistemas, aporta mayor interoperabilidad.
- Permite determinar las palabras clave a partir de las cuales se generan las preguntas en cuestión. De esta forma el usuario sabe en todo momento cómo se crean las preguntas.
- Permite la realización de cuestionarios integrados en la propia aplicación.

3.1.3 OpExams (<https://opexams.com/free-questions-generator/>)

Al igual que las herramientas anteriores, esta permite la generación de preguntas a partir de texto y como novedad, también a partir de temáticas. Ofrece varios tipos de preguntas de opción múltiple, verdadero-falso, entre otras. Asimismo, tiene otras funcionalidades diferentes como un diseñador de exámenes, exámenes online, etc.

3.1.3.1 Ventajas OpExams

- Es notablemente rápido.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

- Parece que soporta muchos idiomas ya que ofrece un *combobox* bastante grande con multitud de idiomas a seleccionar.
- Permite varios tipos de preguntas.

3.1.3.2 Ventajas del sistema respecto a OpExams

- Define claramente el algoritmo que usa para la creación de preguntas, OpExams no habla en ningún momento de cómo se generan las preguntas y parece que extrae información extra de internet.
- Permite la preselección de las palabras clave más importantes.
- Permite el acceso a una API para desarrolladores.
- Carece de un sistema de suscripción que limite la funcionalidad del sistema.

3.2 Valoración de Alternativas de Solución

En este apartado se expondrán diferentes alternativas para llevar cabo la solución. Entre esto se verán diferentes tecnologías y enfoques valorando los ventajas e inconvenientes de cada uno.

3.2.1 Alternativas de enfoque del algoritmo

Las alternativas a la hora de determinar el enfoque del algoritmo a desarrollar se exponen en el apartado 2.1.3.

3.2.2 Desarrollo del backend de la aplicación

Implementar el algoritmo como una parte dentro del sistema es más sencillo que otras alternativas, sin embargo, desembocaría en una mayor rigidez del sistema. Tener el algoritmo como una parte más del sistema, impediría su uso en otras aplicaciones y reduciría su interoperabilidad.

Implementar el algoritmo como una API REST independiente de la interfaz de usuario ofrecería mayor flexibilidad en el sentido de que tanto el sistema como otros usuarios en otras aplicaciones pudieran aprovecharse de los beneficios de la búsqueda de *keywords*. Es por ello por lo que el reto de separar el sistema del algoritmo de búsqueda ofrece notables ventajas que se deben tener en cuenta. De esta misma forma se desacoplaría diversos aspectos, por ejemplo, el lenguaje de programación utilizado o la arquitectura de la aplicación que consuma esta API REST. Asimismo, este enfoque permite aprovechar la potencia de los servicios web para interconectar sistemas, a continuación, se plantean diferentes tecnologías factibles para la implementación del sistema.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

3.2.2.1 Flask

Flask es un *framework* de desarrollo web minimalista que permite crear aplicaciones web rápidamente. Se utiliza en muchos casos para el desarrollo de APIs de servicios a pesar de que permite el uso de motores de plantillas para renderizar datos.

- **Última versión:** 2.0.3
- **Desarrollador:** Armin Ronacher
- **Enlace:** <https://flask.palletsprojects.com/en/2.0.x/>

3.2.2.2 FastAPI

FastAPI a diferencia del anterior, está dedicado a la realización de APIs REST en Python. Es mucho más completo en este sentido que Flask (visto en el apartado anterior). Permite mayores opciones ya que autogenera documentación de las funciones de la API. Esta documentación además de estar renderizada automáticamente para las personas, también se serializa siguiendo las especificaciones OpenAPI que pueden ser procesada por máquinas (ver 14.4 sobre OpenAPI).

- **Última Versión:** 0.94.0
- **Desarrollador:** Sebastián Ramírez
- **Enlace:** <https://fastapi.tiangolo.com/>

3.2.2.3 Django

Django es un *framework* de desarrollo web escrito en Python, de código abierto lanzado en el año 2005. Utiliza el patrón de diseño modelo vista controlador (MVC) y, por mi experiencia, es muy similar en sus características a Spring.

- **Última Versión:** 4.1.3
- **Desarrolladores:** Adrian Holovaty, Simon Willison
- **Enlace:** <https://www.djangoproject.com/>

3.2.2.4 Decisión final de la tecnología para la implementación del backend

Podría hacerse una tabla comparando las características de cada uno:

	Django	FastAPI	Flask
Velocidad y rendimiento	En este aspecto sería el peor ya que está orientado al desarrollo de	Es el que mejor rendimiento ofrece debido principalmente a su enfoque asíncrono.	Es ligero y minimalista y tiene un buen rendimiento en escenarios simples.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

	aplicaciones web completas.		
Documentación y comunidad	Tiene una documentación amplia y la cantidad de tutoriales es amplia.	Carece de mucha documentación, sin embargo, la documentación existente es bastante clara y concisa.	Tiene menos documentación que Django, pero goza de un buen número de recursos disponibles.
Facilidad de desarrollo	Ofrece un desarrollo rápido, sobre todo para proyectos grandes. Es quizá poco adecuado para proyectos pequeños.	Tiene características modernas y es notablemente ágil lo que facilita el desarrollo.	Permite mayor libertad en la toma decisiones y puede ser bastante adecuado para proyectos pequeños.
Escalabilidad	Está diseñado para soportar la escalabilidad.	Ofrece una escalabilidad buena a pesar de ser limitada debido que es el más reciente.	Es fácilmente escalable.
Ecosistema y módulos adicionales	Es quizá el que más cantidad de módulos adicionales respecto de las otras opciones.	A pesar de ser novedoso, tiene bastantes bibliotecas básicas para integrar funcionalidades nuevas.	Tiene una gran cantidad de módulos y extensiones disponibles.

Tabla 3.1. Comparación tecnologías para implementación del backend.

Vista la comparación de las diferentes tecnologías disponibles, se ha optado por utilizar FastAPI. Es una tecnología reciente, así como desconocida para mí, especializada en la creación de API REST. Ofrece documentación fácil y rápida, permite también aspectos de autenticación, conexión con bases de datos, etc. Asimismo, es, a pesar de ser novedosa, escalable y permite integrar módulos fácilmente.

3.2.3 Desarrollo del frontend de la aplicación web

Para implementar el *frontend*, existen numerosas tecnologías que permiten la creación de aplicaciones web. A día de las tecnologías de mayor repercusión son aquellas que ofrecen la posibilidad de crear *Single Page Applications* (SPA). Las aplicaciones de una sola página son aquellas que minimizan las recargas del navegador, mostrando siempre una

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es
--	---	--	---

navegabilidad y una experiencia mucho más fluida para el usuario. Las que se barajan son las mostradas en los siguientes apartados.

3.2.3.1 Angular

Angular (o también Angular 2+ o Angular 2) es un *framework* de desarrollo de aplicaciones web escrito en TypeScript, es de código abierto, y goza del soporte de Google. Se basa como muchos otros *frameworks* en la modelo vista controlador (MVC).

- **Última Versión:** 15.0.4
- **Desarrollador:** Google y la comunidad
- **Enlace:** <https://angular.io/>

3.2.3.2 React

React es una librería JavaScript (no un *framework*) para construir interfaces de usuario. Se basa en el concepto de “componentes” los cuales se pueden combinar fácilmente. Muchos de estos componentes pueden encontrarse predefinidos en diferentes librerías que se exponen en el apartado 14.4 de los anexos de este documento.

- **Última Versión:** 18.1
- **Desarrollador:** Facebook y la comunidad
- **Enlace:** <https://react.dev/>

3.2.3.3 Vue.js

Es un *framework* de JavaScript de código abierto para la construcción de interfaces de usuario y aplicaciones de una sola página. Al igual que los comentados anteriormente también se basa en el uso de componentes, aunque ofrece paquetes extra de para múltiples usos como enrutamiento, estados, etc.

- **Última Versión:** 3.0.1
- **Desarrollador:** Evan You
- **Enlace:** <https://vuejs.org/>

3.2.3.4 Decisión final de la tecnología para la implementación del frontend

	Angular	React	Vue.js
Velocidad y rendimiento	Ofrece, al trabajar de forma diferente a los otros dos, un rendimiento	Es el que quizás ofrece mayor rendimiento y	Bastante eficiente y similar a React,

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

	ligeramente más pobre.	velocidad de renderizado.	para el renderizado de componentes.
Documentación y comunidad	Documentación amplia, donde la comunidad es activa y existen suficientes recursos.	Dada su popularidad, React tiene una documentación notablemente buena con gran cantidad de ejemplos.	La documentación abarca todos aquellos conceptos básicos de la tecnología.
Facilidad de desarrollo	Necesita de más horas de aprendizaje lo cual puede afectar al desarrollo.	Goza de simplicidad y facilidad de aprendizaje. Aspectos que facilitan el desarrollo.	Tiene una curva de aprendizaje suave, lo que ofrece una buena experiencia de desarrollo.
Escalabilidad	Está diseñado para permitir la escalabilidad.	Es de los más escalables, ya que permite dividir la aplicación en componentes fácilmente.	Es relativamente bueno, aunque ligeramente inferior a React.
Ecosistema y módulos adicionales	Contiene bastantes bibliotecas adicionales con gran cantidad de características.	React tiene como fuerte numerosos para el diseño de interfaces adaptables y de calidad.	Tienen numerosos módulos aunque menos que React y Angular.

Tabla 3.2. Comparación tecnologías para implementación del frontend.

Entre las tres tecnologías se ha decidido optar por implementar el *frontend* utilizando React, principalmente por su gran documentación y su uso ampliamente extendido y apoyado por una comunidad muy amplia. React es, además, una de las tecnologías con más necesidad de profesionales, con lo cual su demanda laboral, ha sido un aspecto clave en la toma de la decisión.

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

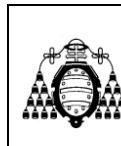
4 Planificación y Gestión del Proyecto

4.1 Planificación inicial del proyecto

4.1.1 Planificación inicial. WBS

El WBS (*Work Breakdown Structure*) o en español EDT (Estructura de Desglose de Trabajo) es una lista estructurada de todos los trabajos que hay que realizar en el proyecto. Esta lista de trabajos o tareas a realizar ha sido desarrollada utilizando la herramienta de Microsoft Project. Se ha estimado una duración de 300 horas basándose en los 12 créditos que compone el desarrollo de este y que figuran en los datos de la planificación académica de la titulación.

La fecha de comienzo del proyecto ha sido el 27/02/2023 y la fecha de fin estimada, teniendo en cuenta un calendario de proyecto de 5 horas diarias de lunes a viernes, es el 19/05/2023. Las etapas de mayor nivel en las que se ha dividido el WBS, así como una imagen del diagrama de Gantt se puede ver a continuación:



1	Planificación Trabajo Fin de Grado	300 horas	lun 27/02/23	vie 19/05/23	
1.1	Investigación del algoritmo	36 horas	lun 27/02/23	mié 08/03/23	
1.2	Terminada investigación algoritmo	0 días	mié 08/03/23	mié 08/03/23	2
1.3	Elaboración de planificación y presupuestos iniciales	22 horas	mié 08/03/23	mar 14/03/23	6
1.4	Terminada planificación y presupuestos iniciales	0 días	mar 14/03/23	mar 14/03/23	7
1.5	Estudio de viabilidad del sistema	6 horas	mar 14/03/23	mié 15/03/23	14
1.6	Terminado el estudio de viabilidad del sistema	0 días	mié 15/03/23	mié 15/03/23	15
1.7	Análisis del sistema	56 horas	mié 15/03/23	jue 30/03/23	18
1.8	Terminado análisis del sistema	0 días	jue 30/03/23	jue 30/03/23	19
1.9	Diseño del sistema	50 horas	vie 31/03/23	jue 13/04/23	25
1.10	Terminado el diseño del sistema	0 días	jue 13/04/23	jue 13/04/23	26
1.11	Implementación y pruebas del sistema	116 horas	vie 14/04/23	mié 17/05/23	31
1.12	Terminada la implementación y pruebas	0 días	mié 17/05/23	mié 17/05/23	32
1.13	Revisión de la documentación	14 horas	mié 17/05/23	vie 19/05/23	38
1.14	Terminado el resto de la documentación	0 días	vie 19/05/23	vie 19/05/23	39
2	Terminado el Trabajo Fin de Grado	0 días	vie 19/05/23	vie 19/05/23	1

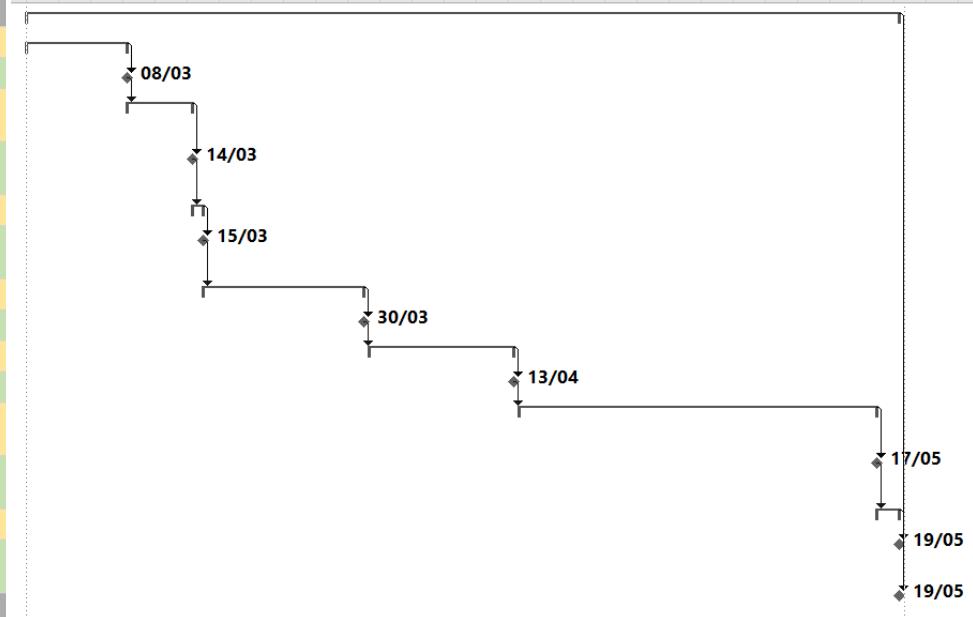


Figura 4.1. WBS con tareas resumen y diagrama de Gantt

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

El WBS con el diagrama de Gantt más detallado, al ser algo que en su naturaleza no deja de ser interactivo, se incluye adjunto en el fichero PlanificaciónInicialTFG.mpp. Las tareas serían las filas marcadas en amarillo y los hitos las líneas marcadas en verde. En los hitos ya que para completarse se han de completar las tareas, se han puesto en sus predecesoras a estas. De ahí las dependencias en el diagrama de Gantt. Con el fin de hacerlo más visual, en la Figura 4.2 se especifica el porcentaje de dedicación a cada una de las tareas resumen del WBS.

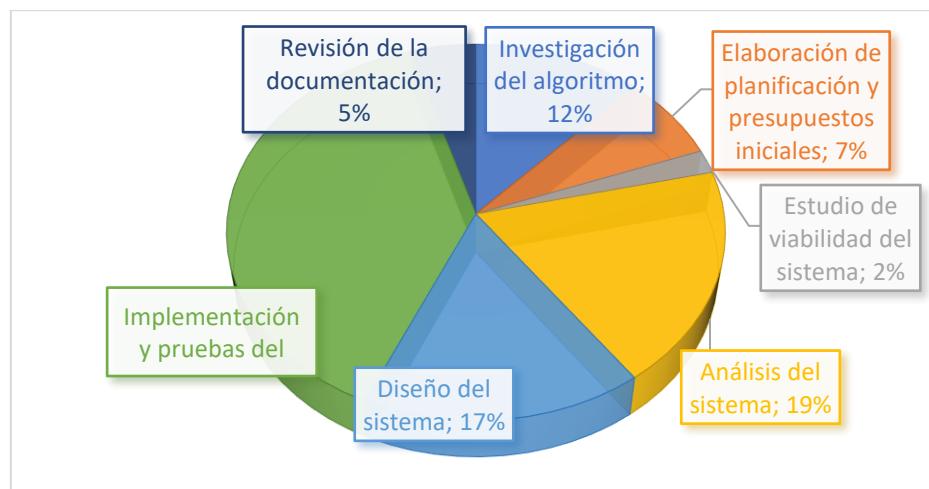


Figura 4.2. Porcentaje de dedicación a cada tarea en base a la planificación inicial.

En los siguientes subapartados se detallará mediante imágenes las diferentes subtareas que se han identificado para la elaboración del WBS.

4.1.1.1 Investigación del algoritmo

Esta etapa se centra en la investigación y desarrollo de un algoritmo para la extracción de palabras clave de textos. Al mismo tiempo de que se lleve a cabo su implementación, también se completará una evaluación de los resultados obtenidos para diferentes textos de prueba.

1.1	Investigación del algoritmo	36 horas	lun 27/02/23	mié 08/03/23
1.1.1	Desarrollo del contexto del problema	6 horas	lun 27/02/23	mar 28/02/23
1.1.2	Implementación del algoritmo	24 horas	mar 28/02/23	lun 06/03/23
1.1.3	Ánalisis de resultados	6 horas	mar 07/03/23	mié 08/03/23
1.2	Terminada investigación algoritmo	0 días	mié 08/03/23	mié 08/03/23

Figura 4.3. Desglose de tareas de la Investigación del algoritmo.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

4.1.1.2 Elaboración de planificación y presupuestos iniciales

Esta etapa se corresponde con la elaboración de una planificación del proyecto que permitirá llevar a cabo un seguimiento tanto del progreso como del coste y la duración del proyecto. A la vez también se llevará a cabo la elaboración de un presupuesto de costes y de cliente del proyecto.

1.3	Elaboración de planificación y presupuestos iniciales	22 horas	mié 08/03/23	mar 14/03/23
1.3.1	Planificación inicial	12 horas	mié 08/03/23	vie 10/03/23
1.3.1.1	Creación de EDT (WBS)	4 horas	mié 08/03/23	mié 08/03/23
1.3.1.2	Estimación de actividades	2 horas	jue 09/03/23	jue 09/03/23
1.3.1.3	Identificación de riesgos y plan de gestión de riesgos	6 horas	jue 09/03/23	vie 10/03/23
1.3.2	Presupuesto inicial	10 horas	vie 10/03/23	mar 14/03/23
1.3.2.1	Presupuesto de costes y cliente inicial	10 horas	vie 10/03/23	mar 14/03/23
1.4	Terminada planificación y presupuestos iniciales	0 días	mar 14/03/23	mar 14/03/23

Figura 4.4. Desglose de tareas Elaborar la planificación y presupuestos.

4.1.1.3 Estudio de viabilidad del sistema

Aquí, se ahondará en el análisis de sistemas similares, buscando características comunes a todos ellos, también se analizarán ventajas y desventajas de los sistemas respecto al propuesto.

1.5	Estudio de viabilidad del sistema	6 horas	mar 14/03/23	mié 15/03/23
1.5.1	Análisis de sistemas similares	2 horas	mar 14/03/23	mar 14/03/23
1.5.2	Valoración de alternativas de solución y selección final	4 horas	mié 15/03/23	mié 15/03/23
1.6	Terminado el estudio de viabilidad del sistema	0 días	mié 15/03/23	mié 15/03/23

Figura 4.5. Desglose de tareas del Estudio de viabilidad del sistema.

4.1.1.4 Análisis del sistema

El análisis del sistema es la fase en la que se desarrollarán tanto los requisitos de usuario y sistema. Se llevarán a cabo diagramas de contexto, diagramas de casos de uso, análisis

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

de escenarios, así como diagramas del sistema iniciales. El objetivo principal es delimitar los aspectos del sistema y tener una visión clara de lo que se va a implementar.

1.7	Análisis del sistema	56 horas	mié 15/03/23	jue 30/03/23
1.7.1	Determinar alcance del sistema	4 horas	mié 15/03/23	jue 16/03/23
1.7.2	Obtención de requisitos de sistema	18 horas	jue 16/03/23	mié 22/03/23
1.7.3	Ánalisis de casos de uso y escenarios	18 horas	mié 22/03/23	lun 27/03/23
1.7.4	Identificación de subsistemas	4 horas	lun 27/03/23	mar 28/03/23
1.7.5	Definición de interfaces de usuario	12 horas	mar 28/03/23	jue 30/03/23
1.8	Terminado análisis del sistema	0 días	jue 30/03/23	jue 30/03/23

Figura 4.6. Desglose de tareas del Análisis del sistema.

4.1.1.5 Diseño del sistema

El diseño del sistema es la etapa en la que se llevarán a cabo diagramas de componentes y de paquetes que permitan detallar más el sistema a implementar. Se aplicarán buenas prácticas para tener software mantenable. También se elaborará el diseño del modelo de datos así como el diseño del plan de pruebas.

1.9	Diseño del sistema	50 horas	vie 31/03/23	jue 13/04/23
1.9.1	Diagramas de arquitectura y componentes del sistema	22 horas	vie 31/03/23	jue 06/04/23
1.9.2	Diseño físico de datos	14 horas	jue 06/04/23	mar 11/04/23
1.9.3	Especificación del plan de pruebas	14 horas	mar 11/04/23	jue 13/04/23
1.10	Terminado el diseño del sistema	0 días	jue 13/04/23	jue 13/04/23

Figura 4.7. Desglose de tareas del Diseño del sistema.

4.1.1.6 Implementación y pruebas

Esta es la fase de mayor duración, en la que se transformarán los diseños y los análisis anteriores a un software funcional que implemente todos los detalles anteriores. Dada la importancia de las pruebas, estas se llevarán a cabo como parte de la implementación del sistema, evitando dejar estas para el final.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

1.11	↳ Implementación y pruebas del sistema	116 horas	vie 14/04/23	mié 17/05/23
1.11.1	Formación en tecnologías elegidas	4 horas	vie 14/04/23	vie 14/04/23
1.11.2	Implementación de la REST API	30 horas	vie 14/04/23	lun 24/04/23
1.11.3	Implementación y pruebas unitarias de la aplicación web	70 horas	lun 24/04/23	vie 12/05/23
1.11.4	Formularios de pruebas de usabilidad	6 horas	vie 12/05/23	lun 15/05/23
1.11.5	Pruebas de adaptabilidad	6 horas	mar 16/05/23	mié 17/05/23
1.12	Terminada la implementación y pruebas	0 días	mié 17/05/23	mié 17/05/23

Figura 4.8. Desglose de tareas de la Implementación y pruebas del sistema.

4.1.1.7 Revisión de la documentación

Esta etapa final permitirá completar aspectos importantes de la documentación, lo cual permitirá corregir y obtener una documentación que aporte valor y de calidad. Todo ello acorde al proyecto en su conjunto.

1.13	↳ Revisión de la documentación	14 horas	mié 17/05/23	vie 19/05/23
1.13.1	Documentación de la implementación (manuales)	2 horas	mié 17/05/23	mié 17/05/23
1.13.2	Documentación de la ejecución de las pruebas	4 horas	mié 17/05/23	jue 18/05/23
1.13.3	Planificación final	4 horas	jue 18/05/23	vie 19/05/23
1.13.4	Conclusiones y ampliaciones	2 horas	vie 19/05/23	vie 19/05/23
1.13.5	Revisión de los anexos	2 horas	vie 19/05/23	vie 19/05/23
1.14	Terminado el resto de la documentación	0 días	vie 19/05/23	vie 19/05/23

Figura 4.9. Desglose de tareas de la revisión de la documentación.

4.1.2 Riesgos

4.1.2.1 Plan de gestión de riesgos

El plan de gestión de riesgos puede verse de forma detallada en los anexos situados al final del trabajo. Concretamente en el apartado 14.1.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

4.1.2.2 Valoración de riesgos

En este apartado se detallan diferentes riesgos que afectan al proyecto. Los riesgos según la definición [21], son una serie de eventos que generan un efecto positivo o negativo sobre algún objetivo del proyecto, ya sea la calidad, el coste, la planificación o el alcance. En esta sección se describirán una serie de riesgos que pueden afectar al proyecto. Estos irán acompañados de un identificador, un título y una breve descripción, así como la categoría de estos.

4.1.2.2.1 *Identificación y análisis de riesgos*

ID	Nombre del Riesgo	Breve Descripción	Categoría
RIES01	Carga de trabajo de asignaturas	Compatibilizar el desarrollo de este trabajo con otras asignaturas del grado, puede generar retrasos y, por tanto, afectar al proyecto. Esto es debido principalmente a que para poder terminar con éxito es necesario aprobar todas las asignaturas.	Gestión de proyectos
RIES02	Adaptación a las tecnologías utilizadas	El uso de nuevas tecnologías como Python en el desarrollo de REST APIs o de React. Puede generar complicaciones a la hora de adaptarse y comenzar. Por ello es un riesgo que puede afectar al proyecto.	Técnico
RIES03	Comunicación con el tutor	Uno de los pilares clave en la realización del trabajo es la comunicación con el tutor. Puede darse el caso de que el tutor no esté disponible en ciertos momentos y puede afectar al proyecto.	Gestión de proyectos
RIES04	Indisponibilidad	Dado que concretamente en este proyecto solo se dispone de un único trabajador, puede ser que indisponibilidades temporales redunden en retrasos notables.	Organizacional
RIES05	Averías del equipo de trabajo	A lo largo del desarrollo los equipos y utilizados pueden sufrir averías. Esto puede desembocar en pérdidas de datos vitales, retrasos y demás factores de riesgo. Actualmente el equipo en el que se desarrolla tiene unos 5 años, y concretamente en este proyecto es el único equipo disponible.	Externo

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

RIES06	Problemas en la configuración del sistema	Al ser un sistema en el que se desarrollarán diversos componentes y módulos, existirá una cierta dificultad a la hora de configurar cada uno de ellos para que funcionen en conjunto. Con lo cual podrá derivar en retrasos y ser de riesgos para el proyecto.	Técnico
RIES07	Desconocimiento de la tecnología usada en Python	Dada la gran cantidad de tecnologías existentes, adaptarse a nuevas tecnologías nunca vistas puede ser costoso al inicio. Python es un lenguaje que se toca menos en el grado con lo cual requiere de un mínimo de adaptación y formación previa.	Técnico
RIES08	Sufrimiento de ataques informáticos	Un riesgo puede sufrir un ataque informático a nuestro equipo. Ya sean diferentes virus o similares que provoquen comportamientos indeseados. En este caso, el lugar de trabajo es una red doméstica sin excesivas medidas de seguridad, simplemente las establecidas por defecto por el operador de telefonía.	Externo
RIES09	Se deja de dar soporte a la tecnología	A lo largo del desarrollo puede pasar que React u otras librerías y <i>frameworks</i> utilizados se queden sin soporte. Esto desembocaría en un riesgo para el proyecto ya que implicaría reestructuraciones del proyecto. Actualmente hay otras tecnologías muy utilizadas como angular, vue.js que pueden hacer que React se utilice menos.	Externo
RIES10	Retrasos en la adquisición de algún producto	Aunque en este proyecto, no se requiere de muchos materiales específicos, es cierto que se pueden requerir de algunas máquinas virtuales o servidores dependientes de algún proveedor de servicios. En este caso si el proveedor se retrasa o se detecta algún problema podría ser de riesgo para el proyecto.	Externo

Tabla 4.1. Riesgos identificados durante el análisis.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

4.1.2.2.2 Priorización y probabilidades de impacto de los riesgos

Probabilidad	Muy Alta	0,90	0,05	0,14	0,27	0,50	0,81
	Alta	0,70	0,04	0,11	0,21	0,39	0,63
	Media	0,50	0,03	0,08	0,15	0,28	0,45
	Baja	0,30	0,02	0,05	0,09	0,17	0,27
	Muy Baja	0,10	0,01	0,02	0,03	0,06	0,09
		0,05	0,15	0,30	0,55	0,90	
		Inapreciable	Bajo	Medio	Alto	Crítico	
		0,05	0,15	0,30	0,55	0,90	
		Impacto					

Figura 4.10. Matriz de probabilidad e impacto.

A continuación, se evalúan cada una de las probabilidades de impactos obtenidas para cada riesgo. Estas probabilidades se obtienen a partir de la matriz de probabilidad e impacto anterior.

Código	Probabilidad	Impacto				Impacto
		Presupuesto	Planificación	Alcance	Calidad	
RIES01	Alta	Inapreciable	Crítico	Medio	Alto	0,63 (P) ²
RIES02	Alta	Bajo	Alto	Medio	Bajo	0,39
RIES03	Media	Bajo	Alto	Bajo	Medio	0,28
RIES04	Media	Bajo	Alto	Medio	Medio	0,28
RIES05	Baja	Alto	Medio	Bajo	Medio	0,17
RIES06	Baja	Inapreciable	Alto	Inapreciable	Inapreciable	0,17
RIES07	Baja	Bajo	Alto	Bajo	Medio	0,17
RIES08	Muy Baja	Inapreciable	Bajo	Bajo	Crítico	0,09
RIES09	Muy Baja	Bajo	Alto	Medio	Alto	0,06
RIES10	Muy Baja	Inapreciable	Alto	Medio	Inapreciable	0,06

Tabla 4.2. Registro de riesgos con probabilidades de impacto de cada riesgo.

4.1.2.3 Gestión de riesgos

4.1.2.3.1 Planificación de gestión de cada riesgo

Se muestran a continuación, diferentes estrategias para afrontar los riesgos.

Código	Nombre del Riesgo	Respuesta al riesgo	Estrategia
RIES01	Carga de trabajo de asignaturas	La carga de trabajo de otras asignaturas es algo que debemos aceptar y con lo que debemos convivir. Es algo que no se puede evitar y que es necesario para el éxito del proyecto. Se puede llevar una buena	Asumir el riesgo

² La (P) indica que el riesgo se debe priorizar dado que este tiene una probabilidad de impacto superior al 0,5.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

		organización, pero es inevitable que aparezca tarde o temprano.	
RIES02	Adaptación a las tecnologías utilizadas	En la informática el aprendizaje es constante por lo que la dificultad al inicio con nuevas tecnologías va a aparecer y debemos convivir con ello.	Asumir el riesgo
RIES03	Comunicación con el tutor	Mantener al tutor informado es responsabilidad nuestra con lo cual debemos mantener una comunicación con cierta antelación y bien planificada. Se debe evitar comunicaciones con prisas.	Mitigar el riesgo
RIES04	Indisponibilidad	Imprevistos de este estilo pueden surgir y no se puede hacer nada ante ellos. Accidentes o causas sobrevenidas aparecen en cualquier momento y debemos aceptar que puedan ocurrir.	Asumir el riesgo
RIES05	Averías del equipo de trabajo	Las averías en un equipo pueden suceder y es algo inevitable. Podemos cuidar la salud del equipo, pero no se puede hacer nada ante ello.	Asumir el riesgo
RIES06	Problemas en la configuración del sistema	Para ello se deberá anticipar cuanto antes el desarrollo de los módulos de esta forma, si surgen problemas tener un mayor tiempo de reacción en caso de que se atasquen ciertos aspectos.	Mitigar el riesgo
RIES07	Desconocimiento de la tecnología usada en Python	Anticipar lo antes posible videos, tutoriales de formación en este tipo de tecnologías. De esta forma podremos tener un mayor margen de tiempo para adaptarnos y convertirnos en “expertos” en este tipo de tecnologías.	Mitigar el riesgo
RIES08	Sufrimiento de ataques informáticos	Los ataques informáticos aparecen todos los días. Hay que ser responsable mantener configuraciones seguras del antivirus así como accesos a sitios con pocas garantías de seguridad. Optimizar la memoria del ordenador para evitar posible <i>malware</i> que se ejecuta en segundo plano.	Mitigar el riesgo
RIES09	Se deja de dar soporte a la tecnología	El uso de unas u otras tecnologías depende normalmente de grandes empresas que dan soporte técnico. El hecho de que una tecnología deje de tener soporte no es algo que podamos controlar.	Asumir el riesgo
RIES10	Retrasos en la adquisición de algún producto	En este caso, solo nos queda asumirlo ya que no se podrá hacer nada al respecto. Simplemente confiar en el proveedor.	Asumir el riesgo

Tabla 4.3. Estrategias de respuesta al riesgo.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

4.1.2.3.2 *Resolución y monitorización de riesgos*

Aunque es una fase del Plan de Gestión de riesgos visto en el Anexo 14.1, en este trabajo no se llevará una monitorización y una resolución de estos exhaustiva de estos.

4.1.3 Resumen presupuesto inicial

El presupuesto detallado tanto de cliente como de coste del proyecto pueden verse en el anexo 14.3.

4.1.3.1 Presupuesto de costes resumido

Código	Partida	Total
01	Partida 1: Investigación del algoritmo	1.404,00 €
02	Partida 2: Estudios y planificación inicial	1.092,00 €
03	Partida 3: Análisis del sistema	2.184,00 €
04	Partida 4: Diseño del sistema	1.950,00 €
05	Partida 5: Implementación y pruebas del sistema	4.524,00 €
06	Partida 6: Revisión de la documentación	546,00 €
07	Partida 7: Otros costes	36,00 €
Total Coste		11.736,00 €

Tabla 4.4. Presupuesto de costes resumido.

4.1.3.2 Presupuesto de cliente resumido

Código	Partida	Total cliente
01	Investigación del algoritmo	1.677,77 €
02	Estudios y planificación inicial	1.320,14 €
03	Análisis del sistema	2.381,73 €
04	Diseño del sistema	2.193,35 €
05	Implementación y pruebas del sistema	5.755,98 €
06	Revisión de la documentación	652,47 €
Total Coste sin IVA		13.981,44 €
IVA (21%)		2.936,10 €
Total Coste con IVA		16.917,55 €

Tabla 4.5. Presupuesto de cliente resumido.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

4.2 Planificación final del proyecto

La planificación final, ya que es similar a la inicial, se ha optado por no detallarla en la memoria. Han aparecido algunas tareas extra no identificadas al inicio y las horas han variado ligeramente. Se puede consultar en detalle en el fichero adjunto en los anexos entregados. Dicho fichero es el denominado PlanificaciónTFG.mpp.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

5 Análisis del Sistema de Información

5.1 Determinación del Alcance del Sistema

Además de lo ya comentado al inicio de este mismo trabajo en el apartado 1.1 en este apartado se delimitará hasta dónde llega sistema, es decir, su alcance. El sistema que se desarrollará tendrá dos actores principales que serán los usuarios anónimos y los usuarios autenticados, ambos con permisos y funcionalidades diferentes.

Por un lado los usuarios anónimos solo podrán acceder a vistas generales de la aplicación como la página principal, así como otras de interés general. Asimismo, estos usuarios anónimos también podrán acceder a la vista de unirse a test o cuestionario (serían los alumnos). Estos cuestionarios serán los que hayan sido creados por los usuarios autenticados.

Por otro lado, los usuarios que hayan entrado en sesión ya sea por medio de registro o de inicio de sesión, dispondrán del resto de funcionalidades. Estas funcionalidades serán las de introducir texto para su procesamiento, búsqueda de palabras clave (así como edición de estas), gestión de preguntas de examen. En cada una de estas fases tendrá la posibilidad de modificar tanto el texto a procesar como palabras clave y las preguntas generadas por el sistema. Del mismo modo, también podrán llevar a cabo una gestión de los cuestionarios o tests que generen a partir de las preguntas donde podrán visualizar los resultados obtenidos por los usuarios que realicen el cuestionario.

5.2 Identificación de Actores del Sistema

Los principales actores que interactuarán en el sistema son los que siguen a continuación:

- **Usuario anónimo:** este actor representará a cualquier usuario que alcance el sitio web. Este actor tendrá una funcionalidad limitada, podrá iniciar sesión o registrarse y verá solo ciertas vistas restringidas. En principio podrá ser cualquier usuario incluidos alumnos que quieran contestar a los cuestionarios.
- **Usuario autenticado:** este actor dispondrá de la totalidad de las funcionalidades. Será aquel usuario que disponga de una cuenta de usuario y haya entrado en sesión por medio de un registro o inicio de sesión. Podrían ser los profesores que decidan crear sus cuestionarios.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

5.3 Establecimiento de Requisitos

5.3.1 Obtención de los Requisitos del Sistema

Para la obtención de requisitos, se ha estructurado este apartado siguiendo la especificación IEEE 830 disponible en [22]. Cada requisito se detalla a continuación.

5.3.1.1 Requisitos de interfaces externas

5.3.1.1.1 *Requisitos de interfaces de usuario*

RINTERFACES_USUARIO1. El sistema será accesible para los usuarios en los navegadores que siguen a continuación:

RINTERFACES_USUARIO1.1. Google Chrome.

RINTERFACES_USUARIO1.2. Microsoft Edge.

RINTERFACES_USUARIO1.3. Mozilla Firefox.

RINTERFACES_USUARIO2. El sistema proporcionará una interfaz adaptable como mínimo a las siguientes resoluciones de dispositivos³.

RINTERFACES_USUARIO2.1. 390x844 (Dispositivos móviles).

RINTERFACES_USUARIO2.2. 768x1024 (Tabletas).

RINTERFACES_USUARIO2.3. 1920x1080 (Monitores).

RINTERFACES_USUARIO3. El sistema estará internacionalizado para los siguientes idiomas:

RINTERFACES_USUARIO3.1. Español.

RINTERFACES_USUARIO3.2. Inglés.

5.3.1.2 Requisitos funcionales

5.3.1.2.1 *Página principal*

RPRINCIPAL1. El sistema mostrará la vista principal a todos los usuarios ya sean anónimos o identificados.

³ Estas resoluciones se corresponden con las más populares en los últimos 12 meses en Europa para cada tipo de dispositivo. Fuente: <https://gs.statcounter.com>.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

5.3.1.2.2 *Contacto*

RCONTACTO1. El sistema permitirá el acceso a los datos de contacto a todos los usuarios. Los datos serán los siguientes:

RCONTACTO1.1. Email.

RCONTACTO1.2. Nombre y apellidos.

RCONTACTO1.3. Perfil de GitHub.

RCONTACTO2. El sistema tendrá accesible la información de contacto desde cualquier punto de la aplicación.

5.3.1.2.3 *Página documentación*

RDOCUMENTACION1. El sistema permitirá el acceso a la documentación disponible a todos los usuarios ya sean anónimos o estén identificados.

RDOCUMENTACION1.1. El sistema redirigirá a los usuarios a las páginas de la documentación del sistema.

RDOCUMENTACIÓN2. El sistema permitirá el acceso a documentación interactiva en diferentes plataformas.

RDOCUMENTACIÓN3. La documentación tendrá un formato procesable por máquinas.

5.3.1.2.4 *Gestión de usuarios*

RUSUARIOS1. El sistema permitirá a los usuarios anónimos registrarse.

RUSUARIOS1.1. El usuario anónimo introducirá los siguientes datos:

RUSUARIOS1.1.1. Nombre (Obligatorio)

RUSUARIOS1.1.2. Primer apellido (Obligatorio)

RUSUARIOS1.1.3. Segundo apellido (Opcional)

RUSUARIOS1.1.4. Email (Obligatorio)

RUSUARIOS1.1.4.1. El sistema validará que el email no esté asociado a otra cuenta registrada.

RUSUARIOS1.1.4.2. El sistema validará que el email no sea vacío.

RUSUARIOS1.1.4.3. El sistema validará que el email tenga el formato típico de un email: <texto>@<texto> sin espacios, siendo los fragmentos de <texto> alfanuméricos.

RUSUARIOS1.1.5. Contraseña (Obligatorio)

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

RUSUARIOS1.1.5.1. El sistema validará que la contraseña contenga mínimo seis caracteres.

RUSUARIOS1.1.5.2. El sistema validará que la contraseña contenga mínimo una letra mayúscula.

RUSUARIOS1.1.5.3. El sistema validará que la contraseña contenga mínimo una letra minúscula.

RUSUARIOS1.1.5.4. El sistema validará que la contraseña contenga mínimo un dígito numérico.

RUSUARIOS1.1.6. Confirmación de la contraseña (Obligatorio)

RUSUARIOS1.1.6.1. El sistema validará que la contraseña confirmada sea idéntica a la contraseña (ver RUSUARIOS1.1.5).

RUSUARIOS1.1.7. El sistema, si no pasa alguna de las validaciones, mostrará un mensaje al usuario impidiendo el registro

RUSUARIOS1.1.8. Si se pasan todas las validaciones, el sistema guardará los datos del usuario en la base de datos.

RUSUARIOS2. El sistema permitirá a los usuarios registrados iniciar sesión.

RUSUARIOS2.1. El sistema solicitará los siguientes datos obligatorios:

RUSUARIOS2.1.1. Email.

RUSUARIOS2.1.2. Contraseña.

RUSUARIOS2.2. El sistema validará que el par email-contraseña figure en la base de datos asociado a un usuario.

RUSUARIOS2.2.1. En caso de que no figure en la base de datos, el sistema mostrará un mensaje al usuario impidiendo el inicio de sesión.

RUSUARIOS2.2.2. En caso de que sí figure en la base de datos, el usuario pasará a estar autenticado.

RUSUARIOS3. El sistema debe permitir a los usuarios identificados salir de sesión.

RUSUARIOS3.1. Si sucede algún fallo durante la salida de sesión, el sistema mostrará un mensaje indicando el problema.

RUSUARIOS4. El sistema permitirá los usuarios visualizar sus datos personales.

5.3.1.2.5 *Procesamiento de texto*

RPROCESAMIENTO1. El sistema permitirá solo a los usuarios autenticados introducir texto o ficheros.

RPROCESAMIENTO2. El sistema será capaz de procesar texto introducido directamente.

RPROCESAMIENTO3. El sistema permitirá la carga de ficheros por parte del usuario.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

RPROCESAMIENTO3.1. El sistema admitirá ficheros txt.

RPROCESAMIENTO4. El sistema verificará que el texto o el contenido del fichero no sea vacío.

RPROCESAMIENTO4.1. Si alguno es vacío, el sistema impedirá realizar el procesado del texto.

RPROCESAMIENTO5. El sistema permitirá introducir un texto de prueba.

RPROCESAMIENTO6. El sistema permitirá eliminar directamente todo el texto introducido.

5.3.1.2.6 *Gestión de palabras clave*

RKEYWORD1. El sistema deberá ser capaz de mostrar la lista de palabras clave identificadas una vez se halla procesado el texto.

RKEYWORD2. El sistema permitirá editar la lista de palabras clave (ver requisito RKEYWORD1).

RKEYWORD2.1. El usuario autenticado tendrá la opción de añadir nuevas palabras clave a la lista.

RKEYWORD2.1.1. El sistema validará que estas palabras añadidas no sean vacías.

RKEYWORD2.1.2. El sistema validará que la palabra no sea repetida.

RKEYWORD2.1.3. El sistema validará que la palabra introducida esté en el texto introducido.

RKEYWORD2.1.4. Si no pasa alguna de las validaciones, el sistema impedirá añadir la palabra a la lista.

RKEYWORD2.2. El usuario autenticado podrá editar cada una de las palabras clave de la lista.

RKEYWORD2.2.1. El sistema validará que estas palabras editadas no sean vacías.

RKEYWORD2.2.2. El sistema validará que estas palabras editadas estén en el texto.

RKEYWORD2.2.3. El sistema validará que estas palabras editadas no estén repetidas con las demás.

RKEYWORD2.2.4. Si no pasa alguna de las validaciones, el sistema impedirá editar dicha palabra.

RKEYWORD2.3. El usuario autenticado tendrá la opción de eliminar las palabras clave de la lista.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

RKEYWORD2.3.1. El sistema validará que esté al menos una palabra seleccionada para eliminar.

RKEYWORD2.3.2. Si no hay ninguna seleccionada, el sistema impedirá que se eliminen palabras.

RKEYWORD3. El usuario podrá seleccionar palabras clave de la lista, (ver RKEYWORD1).

RKEYWORD3.1. El usuario deberá seleccionar al menos una palabra clave para generar las preguntas.

RKEYWORD3.2. El sistema permitirá seleccionar todas las palabras clave automáticamente.

RKEYWORD3.3. El sistema mostrará en tiempo real el número de palabras clave seleccionadas.

RKEYWORD4. El sistema permitirá al usuario filtrar la lista de palabras clave con un sistema de búsqueda.

RKEYWORD4.1. El sistema validará el texto introducido y retornará la lista de palabras clave que contengan dicho texto.

RKEYWORD4.2. Si no coincide ninguna, el sistema indicará al usuario que no se ha encontrado ningún resultado.

RKEYWORD5. El sistema permitirá modificar el número de preguntas a generar por cada palabra.

RKEYWORD5.1. Por defecto este número de preguntas a generar por cada palabra será de 1.

RKEYWORD5.2. El sistema impedirá que el usuario introduzca un número que sea estrictamente menor que 0.

RKEYWORD6. El sistema mostrará en tiempo real el número máximo de preguntas totales a generar.

5.3.1.2.7 Gestión de preguntas

RPREGUNTAS1. El sistema generará preguntas a partir del texto original y las palabras clave encontradas.

RPREGUNTAS2. El usuario autenticado podrá editar cada una de las preguntas.

RPREGUNTAS2.1. El usuario autenticado podrá cambiar el enunciado de la pregunta.

RPREGUNTAS2.1.1. El sistema validará que el enunciado no sea vacío.

RPREGUNTAS2.2. El usuario autenticado podrá editar las opciones de la pregunta.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

RPREGUNTAS2.2.1. El sistema validará que las opciones editadas no sean vacías.

RPREGUNTAS2.3. El usuario autenticado podrá añadir opciones de la pregunta.

RPREGUNTAS2.3.1. El sistema validará que las opciones añadidas no sean vacías.

RPREGUNTAS2.4. El usuario autenticado podrá eliminar opciones de la pregunta.

RPREGUNTAS2.5. Si alguna de las validaciones no se cumple, el sistema impedirá realizar la operación al usuario.

RPREGUNTAS3. El usuario podrá exportar las preguntas a un fichero con los siguientes formatos disponibles.

RPREGUNTAS3.1. Ficheros docx.

RPREGUNTAS3.2. Ficheros pdf.

RPREGUNTAS3.3. Ficheros txt en formato Aiken.

RPREGUNTAS4. El sistema generará como mínimo una pregunta por palabra clave seleccionada.

RPREGUNTAS4.1. El sistema evitará que las preguntas sean repetidas.

RPREGUNTAS4.2. En caso de que haya preguntas repetidas, el sistema avisará al usuario de las preguntas repetidas.

RPREGUNTAS4.3. El sistema impedirá que no se generen preguntas para una palabra clave seleccionada.

RPREGUNTAS5. Si para una palabra se quiere más de una pregunta, el sistema obtendrá el máximo de preguntas diferentes.

RPREGUNTAS5.1. En caso de que se alcance ese máximo, el sistema mostrará un mensaje indicando que no se han podido generar el número requerido de preguntas.

RPREGUNTAS6. El usuario podrá añadir nuevas preguntas.

RPREGUNTAS6.1. El sistema generará un esqueleto por defecto a la pregunta que el usuario podrá cambiar.

RPREGUNTAS6.2. El sistema hará las comprobaciones pertinentes al editar el esqueleto de la pregunta (ver RPREGUNTAS2).

RPREGUNTAS6.3. El sistema permitirá añadir preguntas duplicadas de algunas existentes.

RPREGUNTAS7. El usuario podrá eliminar preguntas de la lista.

RPREGUNTAS8. El usuario podrá ordenar las preguntas generadas por el sistema.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

R PREGUNTAS9. El sistema mostrará el número de preguntas propuestas.

R PREGUNTAS9.1. El sistema actualizará este número cada vez que se elimine una pregunta.

R PREGUNTAS9.2. El sistema actualizará este número cada vez que se añada una pregunta.

5.3.1.2.8 *Gestión de cuestionarios*

RTESTS1. El usuario autenticado tendrá la opción de generar cuestionarios a partir de las preguntas generadas.

RTESTS1.1. El sistema permitirá al usuario autenticado habilitar o deshabilitar el salto de preguntas a la hora de crear los cuestionarios.

RTESTS1.2. El sistema permitirá al usuario autenticado habilitar o deshabilitar el retorno de retroalimentación o *feedback* de los cuestionarios.

RTESTS1.3. El sistema comprobará que haya al menos una pregunta para poder generar el cuestionario.

RTESTS1.4. Si se pasan todas las comprobaciones el sistema almacenará en la base de datos la información del test creado:

RTESTS1.4.1. Id de test.

RTESTS1.4.2. Fecha de creación.

RTESTS1.4.3. Usuario creador.

RTESTS1.4.4. Salto de preguntas activado.

RTESTS1.4.5. Retorno de retroalimentación.

RTESTS1.5. El sistema almacenará la base de datos la información de cada pregunta:

RTESTS1.5.1. Id del cuestionario al que pertenece.

RTESTS1.5.2. Id pregunta.

RTESTS1.5.3. Enunciado de la pregunta.

RTESTS1.6. El sistema guardará en la base de datos la información de cada opción de cada pregunta:

RTESTS1.6.1. Id de la pregunta a la que pertenece.

RTESTS1.6.2. Id opción.

RTESTS1.6.3. Valor de la opción.

RTESTS1.6.4. Si es correcta o no la opción.

RTESTS1.7. Si no sucede ningún error, el sistema mostrará un mensaje indicando que se ha creado correctamente el cuestionario.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

RTESTS1.7.1. El sistema mostrará un código de unión al cuestionario.

RTESTS1.7.1.1. El código podrá ser copiado.

RTESTS1.7.2. El sistema mostrará el enlace de unión al cuestionario.

RTESTS1.7.2.1. El enlace podrá ser escaneado.

RTESTS1.7.2.2. El enlace podrá ser copiado.

RTESTS2. El sistema permitirá a los usuarios anónimos unirse a los cuestionarios disponibles.

RTESTS2.1. El usuario anónimo tendrá la opción de unirse a cuestionarios. Para ello introducirá los siguientes datos:

RTESTS2.1.1. Id del cuestionario.

RTESTS2.1.2. Id de alumno.

RTESTS2.2. El sistema comprobará que el exista el cuestionario introducido en el sistema.

RTESTS2.3. El sistema comprobará que el usuario no haya terminado el cuestionario.

RTESTS2.4. El sistema comprobará que el cuestionario no esté cerrado.

RTESTS2.5. Si alguna de estas validaciones no se cumple el sistema impedirá que el usuario se una al cuestionario.

RTESTS2.6. El sistema una vez se una el usuario almacenará la información del usuario en la base de datos:

RTESTS2.6.1. Id de usuario (alumno).

RTESTS2.6.2. Id de test al que se une.

RTESTS2.7. El sistema comprobará si el usuario tiene el cuestionario empezado y sin terminar.

RTESTS2.7.1. En ese caso el sistema enviará un mensaje a la base de datos para obtener los datos de las últimas respuestas realizadas.

RTESTS3. El sistema permitirá responder las preguntas a los usuarios que se hayan unido al cuestionario.

RTESTS3.1. El sistema almacenará en la base de datos la respuesta contestada cada vez que el usuario modifique una opción.

RTESTS3.2. El sistema permitirá pasar de una pregunta a otra de distintas formas:

RTESTS3.2.1. Saltando de una a otra si la opción está habilitada (ver RTESTS1.4).

RTESTS3.2.2. Hacia la inmediatamente siguiente.

RTESTS3.2.3. Hacia la inmediatamente anterior.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

RTESTS3.3. El sistema mostrará un mapa informando del estado del cuestionario.

RTESTS3.3.1. El sistema indicará el total de las preguntas.

RTESTS3.3.2. El sistema indicará la pregunta en la que se encuentra el usuario.

RTESTS3.3.3. El sistema indicará las preguntas que se hayan contestado.

RTESTS3.4. El sistema mostrará el id de usuario con el que se está en el cuestionario.

RTESTS3.5. El sistema permitirá limpiar la pregunta seleccionada.

RTESTS4. El sistema permitirá a los usuarios autenticados monitorizar el estado de los cuestionarios que creen.

RTESTS4.1. El sistema mostrará para cada cuestionario la siguiente información:

RTESTS4.1.1. Id del test.

RTESTS4.1.2. Fecha de creación del cuestionario.

RTESTS4.1.3. Si está cerrado o no el cuestionario

RTESTS4.1.4. Identificadores de los usuarios que se han unido al cuestionario.

RTESTS4.1.4.1. Si no hay usuarios que se hayan unido, el sistema indicará que no hay usuarios.

RTESTS4.2. El usuario autenticado podrá abrir/cerrar los cuestionarios.

RTESTS4.2.1. Cada vez que el usuario cambie el estado del cuestionario, el sistema enviará a la base de datos el nuevo estado.

RTESTS4.3. El usuario autenticado podrá visualizar en tiempo real los datos de los usuarios que están unidos al cuestionario.

RTESTS4.3.1. Estado que podrá ser “En progreso”, “Suspensión” o “Aprobado”.

RTESTS4.3.2. Puntuación.

RTESTS4.3.3. Id del usuario.

RTESTS4.3.4. Número de preguntas contestadas.

RTESTS4.4. El sistema permitirá de actualizar los datos.

RTESTS4.4.1. El sistema enviará una petición a la base de datos obteniendo los datos asociados a los cuestionarios creados por el usuario.

RTESTS4.4.2. Si no sucede ningún error los datos se actualizarán.

RTESTS4.4.3. Si sucede algún error, el sistema informará al usuario del problema.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

RTESTS4.5. El usuario podrá ordenar por los datos de los usuarios unidos al cuestionario.

RTESTS4.6. El usuario podrá visualizar el código del cuestionario para escanear en todo momento.

RTESTS5. El sistema permitirá enviar el resultado en cualquier momento.

RTESTS5.1. Cuando se envíe el resultado el sistema calculará la nota del cuestionario.

RTESTS5.2. El sistema enviará el resultado a la base de datos con la información siguiente:

RTESTS5.1.1. Puntuación en el cuestionario.

RTESTS5.1.2. Afirmando que ha terminado el cuestionario.

RTESTS5.2. El sistema comprobará que el cuestionario no esté cerrado antes de enviar.

RTESTS5.3. Si no sucede ningún error al enviar el cuestionario, el sistema mostrará *feedback* o no en función de si está o no habilitado (ver RTESTS1.4).

RTESTS5.3.1. Si no se muestra *feedback*, el usuario verá la confirmación pudiendo volver a la página principal.

RTESTS5.3.2. Si se muestra *feedback*, el sistema mostrará las preguntas acertadas, falladas, no contestadas junto con la puntuación obtenida.

RTESTS5.3.3. El sistema calculará la nota en base diez según los siguientes criterios:

RTESTS5.3.3.1. Pregunta no contestada cero puntos.

RTESTS5.3.3.2. Pregunta acertada suma un punto.

RTESTS5.3.3.3. Pregunta fallada resta uno partido del número de opciones.

RTESTS5.3.4. El sistema guardará la nota sobre diez en la base de datos.

RTESTS5.3.5. Si la nota obtenida es negativa, el sistema almacenará cero como puntuación obtenida.

RTESTS5.4. Si sucede algún error el sistema indicará el motivo por el que no se puede enviar el resultado.

5.3.1.3 Atributos del sistema

5.3.1.3.1 Requisitos de seguridad

RSEGURIDAD1: El sistema almacenará las contraseñas de los usuarios cifradas.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

RSEGURIDAD2: El sistema controlará accesos vía URL a vistas no permitidas para un determinado usuario.

RSEGURIDAD2.1: Si se produce un acceso no autorizado se mostrará un mensaje de impidiendo la acción al usuario.

RSEGURIDAD3. La comunicación con el sistema se hará por medio del protocolo https.

5.4 Especificación de Casos de Uso y Escenarios

5.4.1 Diagrama de contexto

Este diagrama de contexto, a continuación, representa los casos de uso a alto nivel de forma que se delimita el sistema.

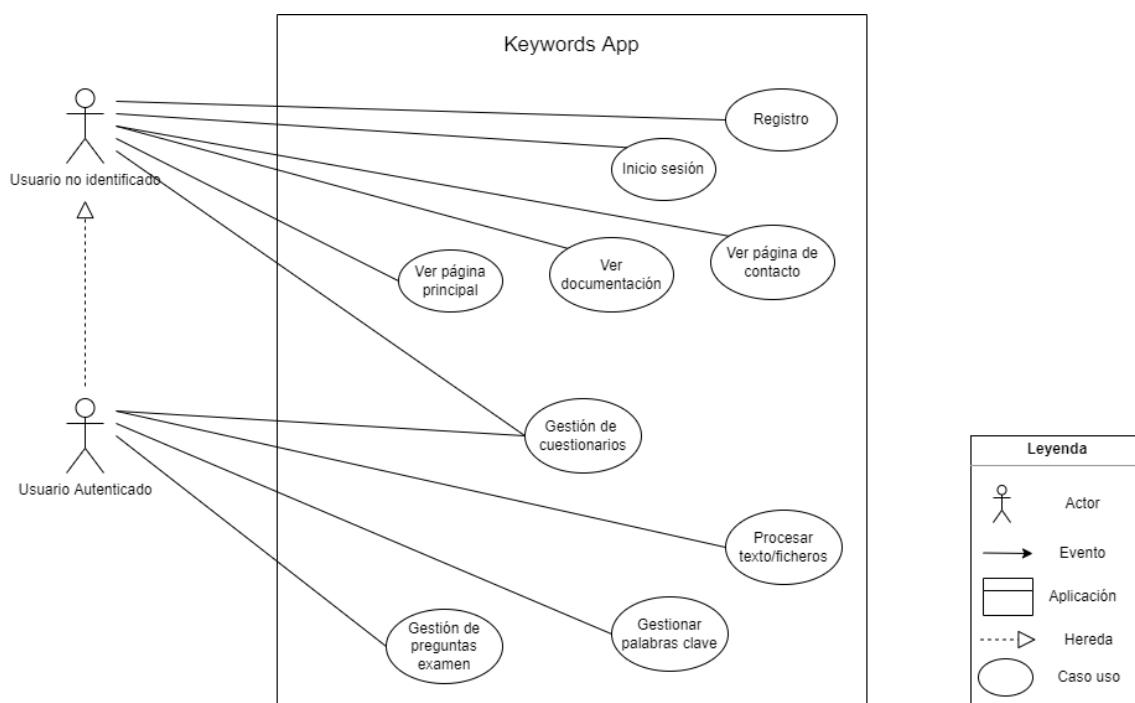


Figura 5.1. Diagrama de contexto de la aplicación

5.4.2 Especificación de casos de uso

Además, a continuación, se especifican las descripciones de casos de uso junto a pequeños diagramas que detallan aún más el diagrama de contexto anterior. Asimismo, se detallan trazabilidades con los requisitos mencionados anteriormente.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

5.4.2.1 Ver página principal

Nombre del caso de uso:	Ver página principal
Quien lo inicia:	Usuario no identificado o Usuario identificado
Requisitos que satisface:	RPRINCIPAL1
Descripción:	
Cualquier usuario autenticado o no, podrá visualizar la página principal de la aplicación. En esta página principal se incluirá un menú que permitirá acceder al resto de opciones. Estas opciones serán diferentes en base al tipo de usuario.	

Tabla 5.1. Caso de uso de ver la página principal.

5.4.2.2 Ver página de documentación

Nombre del caso de uso:	Ver página de documentación
Quien lo inicia:	Usuario no identificado o Usuario identificado
Requisitos que satisface:	RDOCUMENTACION1
Descripción:	
Cualquier usuario autenticado o no, podrá visualizar la página de la documentación de la aplicación accesible desde el menú.	

Tabla 5.2. Caso de uso de ver la página de documentación.

5.4.2.3 Ver información de contacto

Nombre del caso de uso:	Ver página de contacto
Quien lo inicia:	Usuario no identificado o Usuario identificado
Requisitos que satisface:	RCONTACTO1
Descripción:	
Cualquier usuario autenticado o no, podrá visualizar la información de contacto desde cualquier punto de la aplicación.	

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

Tabla 5.3. Caso de uso de la página de contacto.

5.4.2.4 Gestión de usuarios

Nombre del caso de uso:	Inicio de sesión
Quien lo inicia:	Usuario no identificado
Requisitos que satisface:	RUSUARIOS2 y subrequisititos
Descripción:	<p>Cualquier usuario que no se haya autenticado previamente tendrá disponible la opción de iniciar sesión. Para ello el usuario deberá introducir el usuario y contraseña con el que se haya registrado previamente.</p> <p>El sistema comprobará que el par usuario-contraseña introducido por el usuario figure en la base de datos. En caso afirmativo, el usuario iniciará sesión y dispondrá del resto de funcionalidades para un usuario autenticado. En caso negativo, se mostrará un mensaje impidiendo el inicio de sesión.</p>

Tabla 5.4. Caso de uso de inicio de sesión de usuarios.

Nombre del caso de uso:	Registro
Quien lo inicia:	Usuario no identificado
Requisitos que satisface:	RUSUARIOS1 y subrequisititos
Descripción:	<p>Cualquier usuario que no se haya autenticado previamente tendrá disponible la opción de registrarse. Deberá introducir los datos especificados en los requisitos sobre gestión de usuarios (apartado 5.3.1.2.4).</p> <p>El sistema se encargará de validar los datos. Si no sucede ningún error, se creará una nueva entrada en la base de datos con las credenciales del usuario y este entrará en sesión. En caso de que exista algún error, se mostrará un mensaje de error indicando el motivo.</p>

Tabla 5.5. Caso de uso de registro de usuarios.

5.4.2.5 Procesamiento del texto



Figura 5.2. Diagrama de casos de uso de procesamiento del texto.

Nombre del caso de uso:	Carga ficheros
Quien lo inicia:	Usuario identificado
Requisitos que satisface:	Requisitos de RPROCESAMIENTO3
Descripción:	
El usuario autenticado tendrá disponible la opción de cargar ficheros. El sistema validará el formato del fichero y el contenido. En caso de que suceda algún error, el sistema informará al usuario del problema.	

Tabla 5.6. Caso de uso de carga de ficheros.

Nombre del caso de uso:	Carga de texto
Quien lo inicia:	Usuario identificado
Requisitos que satisface:	Requisitos de RPROCESAMIENTO2 y RPROCESAMIENTO 5
Descripción:	
El usuario autenticado tendrá disponible la opción de cargar el texto directamente. El sistema validará el texto introducido. En caso de que suceda algún error, el sistema informará al usuario del problema. Además, el sistema permitirá cargar un texto de prueba por si el usuario no dispone de ninguno.	

Tabla 5.7. Caso de uso de carga de texto.

Nombre del caso de uso:	Eliminar todo el texto
Quien lo inicia:	Usuario identificado

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

Requisitos que satisface:	Requisitos de RPROCESAMIENTO6
Descripción:	
El usuario autenticado tendrá disponible la opción de cargar el texto directamente. El sistema validará el texto introducido. En caso de que suceda algún error, el sistema informará al usuario del problema.	

Tabla 5.8. Caso de uso de eliminar todo el texto.

5.4.2.6 Gestión de palabras clave

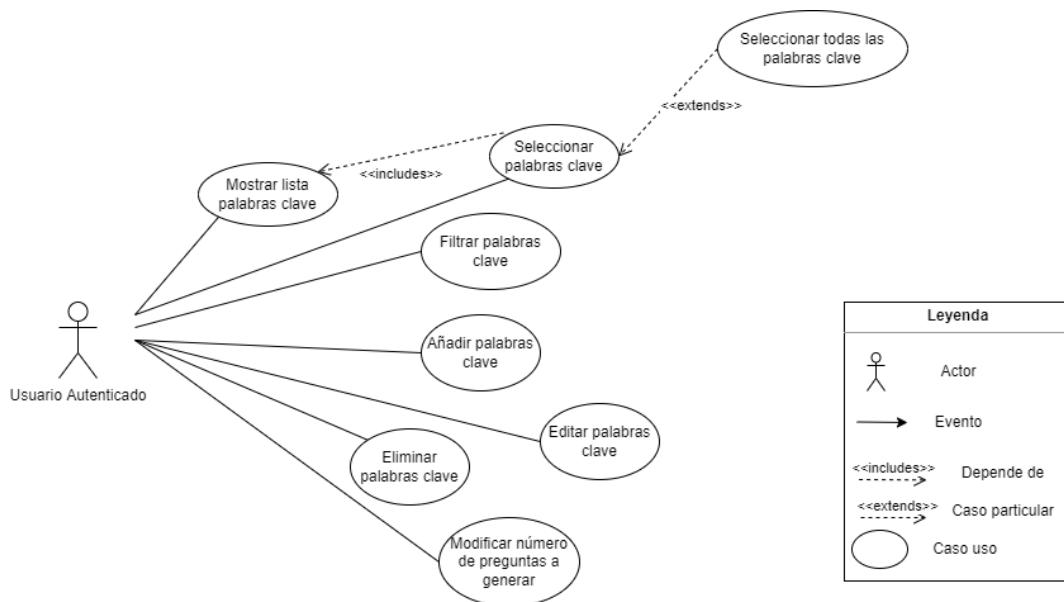


Figura 5.3. Diagrama de casos de uso para la gestión de palabras clave.

Nombre del caso de uso:	Mostrar lista palabras clave
Quien lo inicia:	Sistema
Requisitos que satisface:	Requisitos de RKEYWORD1
Descripción:	
El sistema a partir del texto mostrará una lista de palabras clave al usuario que hayan sido generadas a partir del texto. Además estas servirán posteriormente para mostrar las preguntas.	

Tabla 5.9. Caso de uso de mostrar palabras clave.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

Nombre del caso de uso:	Añadir palabras clave
Quien lo inicia:	Usuario identificado
Requisitos que satisface:	Requisitos de RKEYWORD2.1
Descripción:	
El usuario autenticado una vez haya procesado el texto, podrá añadir nuevas palabras clave para usar después en la gestión de preguntas. Estas palabras clave que sean añadidas no deben ser vacías ni repetidas, el sistema se encargará de comprobarlo.	

Tabla 5.10. Caso de uso de añadir palabras clave.

Nombre del caso de uso:	Editar palabras clave
Quien lo inicia:	Usuario identificado
Requisitos que satisface:	Requisitos de RKEYWORD2.2
Descripción:	
El usuario autenticado una vez haya procesado el texto, podrá editar las palabras clave para usar después en la gestión de preguntas. Estas palabras clave editadas no deben ser vacías, el sistema se encargará de comprobarlo.	

Tabla 5.11. Caso de uso de editar palabras clave.

Nombre del caso de uso:	Eliminar palabras clave
Quien lo inicia:	Usuario identificado
Requisitos que satisface:	Requisitos de RKEYWORD2.3
Descripción:	
El usuario autenticado una vez haya procesado el texto, podrá eliminar palabras clave que haya seleccionado para usar después en la gestión de preguntas. Estas palabras clave que sean añadidas deberán estar presentes en el texto, el sistema se encargará de comprobarlo.	

Tabla 5.12. Caso de uso de eliminar palabras clave.

Nombre del caso de uso:	Seleccionar palabras clave
Quien lo inicia:	Usuario identificado

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

Requisitos que satisface:	Requisitos de RKEYWORD3
Descripción:	
El usuario podrá seleccionar las palabras clave que considere oportunas y de mayor relevancia para la posterior generación de preguntas. Deberá seleccionar al menos una palabra clave para poder generar preguntas. Además, un caso, particular de este caso de uso es el que se indica para seleccionar todas de forma simultánea.	

Tabla 5.13. Caso de uso de Seleccionar palabras clave.

Nombre del caso de uso:	Filtrar palabras clave
<b b="" inicia:<="" lo="" quien="">	Usuario identificado
Requisitos que satisface:	Requisitos de RKEYWORD4
Descripción:	
El usuario podrá a través de un sistema de búsqueda filtrar en base a las palabras que crea más interesantes. El usuario proporcionará el texto y la lista de palabras clave se actualizará de manera dinámica.	

Tabla 5.14. Caso de uso de filtrar palabras clave.

Nombre del caso de uso:	Modificar número de preguntas a generar
<b b="" inicia:<="" lo="" quien="">	Usuario identificado
Requisitos que satisface:	Requisitos de RKEYWORD5
Descripción:	
El usuario podrá decidir cuántas preguntas quiere generar por cada palabra clave, el usuario podrá incrementar o decrementar el número así como introducirlo directamente. El sistema cada vez que el usuario modifique este número en cada una de las palabras actualizará el número total de preguntas a generar.	

Tabla 5.15. Caso de uso de modificar número de preguntas a generar.

5.4.2.7 Gestión de preguntas



Figura 5.4. Diagrama de casos de uso para la gestión de preguntas de examen.

Nombre del caso de uso:	Editar respuestas
Quien lo inicia:	Usuario identificado
Requisitos que satisface:	Requisitos de RPREGUNTAS2.2
Descripción:	
El usuario podrá editar las opciones propuestas por el sistema. Podrá cambiar el texto de cada una de las opciones con el fin de perfeccionar al máximo los resultados propuestos.	

Tabla 5.16. Caso de uso de Editar respuestas.

Nombre del caso de uso:	Añadir respuestas
Quien lo inicia:	Usuario identificado
Requisitos que satisface:	Requisitos de RPREGUNTAS2.3
Descripción:	
El sistema permitirá añadir nuevas opciones a cada pregunta por parte del usuario identificado. El sistema verificará que las nuevas opciones no sean vacías.	

Tabla 5.17. Caso de uso de Añadir respuestas.

Nombre del caso de uso:	Eliminar respuestas
--------------------------------	---------------------

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

Quien lo inicia:	Usuario identificado
Requisitos que satisface:	Requisitos de RPREGUNTAS2.4
Descripción:	
El sistema permitirá eliminar las opciones propuestas por el sistema. De esta forma se podrán corregir algunas opciones que no tengan sentido, sean incorrectas o demasiado obvias.	

Tabla 5.18. Caso de uso de Eliminar respuestas.

Nombre del caso de uso:	Editar enunciados
Quien lo inicia:	Usuario identificado
Requisitos que satisface:	Requisitos de RPREGUNTAS2.1
Descripción:	
El usuario podrá editar los enunciados de las preguntas en caso de que detecte algún fallo o decida mejorarlos. El sistema validará que estos no sean vacíos, en ese caso impedirá realizar la acción.	

Tabla 5.19. Caso de uso de Editar enunciados.

Nombre del caso de uso:	Exportar preguntas
Quien lo inicia:	Usuario identificado
Requisitos que satisface:	Requisitos de RPREGUNTAS3
Descripción:	
El usuario podrá exportar las preguntas que desee a ficheros de texto a través de la opción que proporciona el sistema. Concretamente se podrán exportar las preguntas a formato docx, txt y pdf.	

Tabla 5.20. Caso de uso de Exportar preguntas de examen.

Nombre del caso de uso:	Mostrar preguntas
Quien lo inicia:	Usuario identificado
Requisitos que satisface:	Requisitos de RPREGUNTAS1, RPREGUNTAS4

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

Descripción:

El sistema mostrará la lista de preguntas en base a la lista de palabras clave seleccionadas. El sistema mostrará cada pregunta con 4 opciones. Las preguntas se intentarán que no sean repetidas, en caso de que no sea posible el sistema avisará al usuario de que puede haber preguntas repetidas.

Tabla 5.21. Caso de uso de mostrar preguntas.

Nombre del caso de uso:	Añadir preguntas
Quien lo inicia:	Usuario identificado
Requisitos que satisface:	Requisitos de RPREGUNTAS6
Descripción:	
El usuario podrá añadir nuevas preguntas por defecto de forma que pueda crear sus propias preguntas. Además como caso particular también podrá duplicar preguntas existentes o que hayan sido generadas por el sistema.	

Tabla 5.22. Caso de uso de Añadir preguntas.

Nombre del caso de uso:	Ordenar preguntas
Quien lo inicia:	Usuario identificado
Requisitos que satisface:	Requisitos de RPREGUNTAS8
Descripción:	
El usuario podrá ordenar las preguntas de la lista. De esta forma podrá organizar la lista poniendo arriba o abajo las que considere más importantes.	

Tabla 5.23. Caso de uso de Ordenar preguntas.

Nombre del caso de uso:	Eliminar preguntas
Quien lo inicia:	Usuario identificado
Requisitos que satisface:	Requisitos de RPREGUNTAS7
Descripción:	

El usuario podrá eliminar las preguntas de la lista. De esta forma podrá descartar aquellas preguntas que se hayan generado con errores o no sean necesarias para el usuario.

Tabla 5.24. Caso de uso de eliminar preguntas.

5.4.2.8 Gestión de cuestionarios

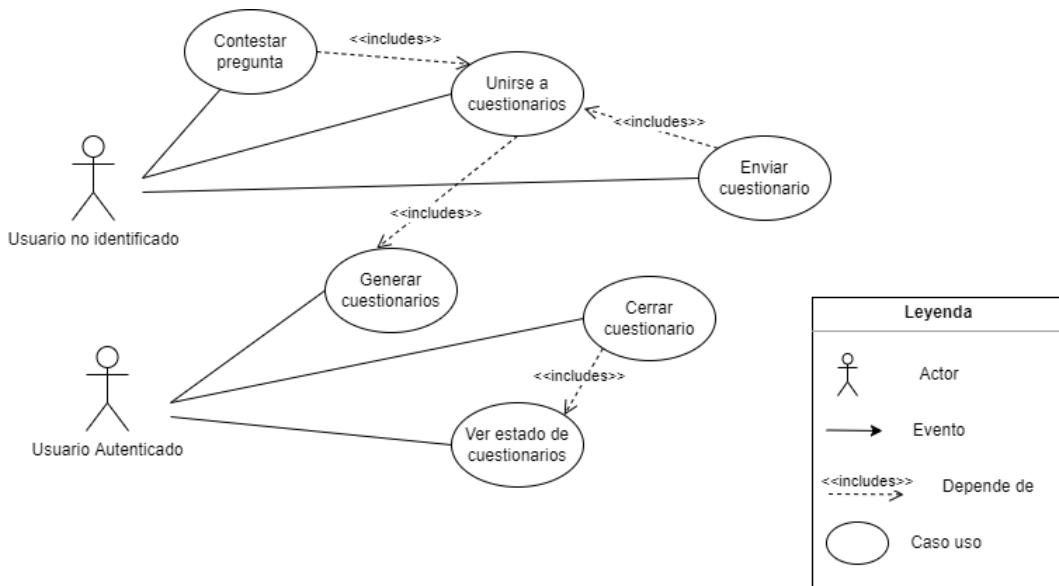


Figura 5.5. Diagrama de casos de uso para la gestión de cuestionarios.

Nombre del caso de uso:	Generar cuestionarios
Quien lo inicia:	Usuario identificado
Requisitos que satisface:	Requisitos de RTESTS1
Descripción:	
El usuario autenticado, una vez genere las preguntas que desee, podrá generar test con esas preguntas que se almacenarán en la base de datos. El sistema permitirá configurar estos cuestionarios habilitando o no saltos de pregunta o habilitando la retroalimentación al finalizar.	
El sistema retornará el código del test así como la posibilidad de facilitar el escaneo para unirse.	

Tabla 5.25. Caso de uso de generar cuestionarios.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

Nombre del caso de uso:	Ver estado de cuestionarios
Quien lo inicia:	Usuario identificado
Requisitos que satisface:	Requisitos de RTESTS4
Descripción:	<p>El sistema permitirá ver el estado de los cuestionarios que haya creado. Mostrará el id del cuestionario, así como la fecha de creación. También mostrará todos los alumnos que se hayan unido mostrando en tiempo real el número de preguntas contestadas y la puntuación.</p>

Tabla 5.26. Caso de uso de ver estado de cuestionarios.

Nombre del caso de uso:	Cerrar cuestionarios
Quien lo inicia:	Usuario identificado
Requisitos que satisface:	Requisitos de RTESTS4.2
Descripción:	<p>El sistema permitirá a los usuarios autenticados que hayan creado los cuestionarios cerrarlos o abrirlos en función de cómo deseen. El sistema enviará a la base de datos un mensaje para actualizar el estado. Si no sucede ningún error, el sistema mostrará un mensaje indicando que se ha modificado correctamente.</p>

Tabla 5.27. Caso de uso de cerrar cuestionarios.

Nombre del caso de uso:	Unirse a test
Quien lo inicia:	Usuario no identificado
Requisitos que satisface:	Requisitos de RTESTS2
Descripción:	<p>Un usuario anónimo podrá unirse a los cuestionarios por medio del enlace, el código o escaneando un código de acceso. El sistema comprobará que el usuario no haya terminado el cuestionario, y que el cuestionario exista. Si se da alguna situación, el sistema impedirá la unión al cuestionario.</p> <p>Si no sucede ningún error, el sistema añadirá los datos del usuario a la base de datos (en caso de que no haya entrado ya al cuestionario). Por otro lado, si ha entrado ya pero</p>

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

no ha enviado el cuestionario, el sistema cargará el estado anterior en el que el usuario se salió del cuestionario.

Tabla 5.28. Caso de uso de unirse a cuestionarios.

Nombre del caso de uso:	Contestar pregunta	
Quien lo inicia:	Usuario no identificado	
Requisitos que satisface:	Requisitos de RTESTS3	
Descripción:	<p>El usuario podrá contestar cada una de las preguntas seleccionando la opción correcta. Contestar una pregunta depende de que el usuario se haya podido unir o no al cuestionario. Cada vez que se conteste una pregunta el sistema actualizará en la base datos la opción elegida de forma que no se envíen todas al final. Además, se actualizará el estado del mapa de las preguntas marcándolas como contestadas.</p>	

Tabla 5.29. Caso de uso de Contestar pregunta.

Nombre del caso de uso:	Enviar a cuestionarios	
Quien lo inicia:	Usuario no identificado	
Requisitos que satisface:	Requisitos de RTESTS5	
Descripción:	<p>El sistema permitirá a los usuarios que se hayan unido al cuestionario, enviar las preguntas que hayan contestado. El sistema enviará los datos a la base de datos y se calculará la nota correspondiente a la selección del usuario.</p> <p>Una vez enviado el sistema mostrará o no <i>feedback</i> en función de la configuración del cuestionario, en caso de que sí, se indicará la puntuación obtenida así como las preguntas falladas y las acertadas. Si por otro lado no está el <i>feedback</i> activado, el sistema mostrará un mensaje indicando que se ha enviado y permitirá volver al inicio.</p>	

Tabla 5.30. Caso de uso de enviar cuestionarios.

5.4.3 Análisis de escenarios

Los escenarios nos permiten conocer cómo funcionan los casos de uso especificados anteriormente. Normalmente, especifican las interacciones entre los usuarios y el sistema analizando el camino normal, así como alternativas. Los escenarios desarrollados son los que siguen en los siguientes apartados.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

5.4.3.1 Ver página principal

VER PÁGINA PRINCIPAL	
NUMERACIÓN	1
PRECONDICIONES	-
POSCONDICIONES	-
QUIEN LO COMIENZA	<i>Usuario anónimo o identificado</i>
QUIEN LO FINALIZA	<i>Usuario anónimo o identificado</i>
DESCRIPCIÓN	<i>El usuario accede a la aplicación web y visualiza la página principal del sitio con la información general.</i>

Tabla 5.31. Escenario de ver página principal

5.4.3.2 Ver página documentación

VER PÁGINA DOCUMENTACIÓN	
NUMERACIÓN	2
PRECONDICIONES	-
POSCONDICIONES	-
QUIEN LO COMIENZA	<i>Usuario anónimo o identificado</i>
QUIEN LO FINALIZA	<i>Usuario anónimo o identificado</i>
DESCRIPCIÓN	<i>El usuario accede a la aplicación web, visualiza la página principal del sitio con la información general. Posteriormente decide acudir al menú de opciones y acceder a la documentación disponible. La aplicación muestra toda la documentación.</i>

Tabla 5.32. Escenario de ver página de documentación.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

5.4.3.3 Ver página contacto

VER PÁGINA CONTACTO	
NUMERACIÓN	3
PRECONDICIONES	-
POSCONDICIONES	-
QUIEN LO COMIENZA	<i>Usuario anónimo o identificado</i>
QUIEN LO FINALIZA	<i>Usuario anónimo o identificado</i>
DESCRIPCIÓN	<i>El usuario accede a la aplicación web, visualiza la página principal del sitio con la información general. Posteriormente decide acudirá ver la información de contacto para interesarse por la implementación del sistema.</i>

Tabla 5.33. Escenario de ver página de contacto.

5.4.3.4 Gestión de usuarios

REGISTRO	
NUMERACIÓN	4
PRECONDICIONES	-
POSCONDICIONES	-
QUIEN LO COMIENZA	<i>Usuario anónimo</i>
QUIEN LO FINALIZA	<i>Sistema</i>
DESCRIPCIÓN	<p><i>Un usuario anónimo accede a la aplicación web. Acude al menú de opciones y decide registrarse. Introduce todos los datos necesarios:</i></p> <ul style="list-style-type: none"> • <i>Nombre</i> • <i>Apellidos</i> • <i>Email</i> • <i>Contraseña y confirmación de contraseña</i>

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

VARIACIONES (ESCENARIOS SECUNDARIOS)

El sistema pasa las validaciones y añade un nuevo usuario registrado. El sistema muestra que se ha registrado correctamente.

1. *El usuario utiliza un email correspondiente a otro usuario ya registrado en el sistema. En este caso no pasaría las validaciones y el sistema impedirá el registro.*
2. *El usuario introduce la contraseña y la confirmación de contraseña diferente. No pasaría las validaciones y mostraría un error.*
3. *El usuario introduce una contraseña menor de 6 caracteres o sin ninguna mayúscula o si ningún dígito. El sistema impedirá realizar el registro.*

Tabla 5.34. Escenario de registro.

INICIO DE SESIÓN	
NUMERACIÓN	5
PRECONDICIONES	-
POSCONDICIONES	-
QUIEN LO COMIENZA	<i>Usuario anónimo</i>
QUIEN LO FINALIZA	<i>Sistema</i>
DESCRIPCIÓN	<p><i>Un usuario anónimo registrado con anterioridad en el sistema accede al sistema. Acude al menú de opciones y decide iniciar sesión. Introduce todos los datos necesarios:</i></p> <ul style="list-style-type: none"> • <i>Email</i> • <i>Contraseña</i> <p><i>El sistema pasa las validaciones de que exista el par email-contraseña. El sistema muestra que se ha iniciado sesión correctamente.</i></p>
VARIACIONES (ESCENARIOS SECUNDARIOS)	<ol style="list-style-type: none"> 1. <i>El usuario utiliza un email que no existe en el sistema. El sistema mostrará un mensaje de inicio de sesión fallido.</i> 2. <i>El usuario introduce un email que existe en la base de datos, pero una contraseña que no</i>

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

corresponde al email en cuestión. El sistema mostrará un mensaje de inicio de sesión fallido.

Tabla 5.35. Escenario de inicio de sesión.

5.4.3.5 Procesamiento del texto

CARGAR TEXTO/FICHERO	
NUMERACIÓN	6
PRECONDICIONES	<i>El usuario debe haber iniciado sesión con anterioridad</i>
POSCONDICIONES	-
QUIEN LO COMIENZA	<i>Usuario identificado</i>
QUIEN LO FINALIZA	<i>Sistema</i>
DESCRIPCIÓN	<i>Un usuario identificado con email diego@email.com introduce un texto o fichero sobre historia de España para procesar y buscar las palabras clave. El sistema procesará el texto y validará que no sea vacío, posteriormente retornará la lista de palabras clave identificadas.</i>
VARIACIONES (ESCENARIOS SECUNDARIOS)	<ol style="list-style-type: none"> 1. <i>El usuario introduce un texto o un fichero sin texto. El sistema impedirá que se pueda procesar el texto.</i> 2. <i>El usuario adjunta un fichero no válido y que no se puede procesar. El sistema mostrará un mensaje indicando el motivo.</i> 3. <i>El usuario no tiene ningún fichero ni ningún texto, decide acudir a la opción de cargar texto de prueba para probar la aplicación. Finalmente carga un texto de prueba.</i>

Tabla 5.36. Escenario de carga de texto/fichero.

5.4.3.6 Gestión de palabras clave

AÑADIR PALABRAS CLAVE

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

NUMERACIÓN	7.1
PRECONDICIONES	<i>El usuario debe haber introducido un texto para procesar previamente.</i>
POSCONDICIONES	-
QUIEN LO COMIENZA	<i>Usuario identificado</i>
QUIEN LO FINALIZA	<i>Sistema</i>
DESCRIPCIÓN	<i>Un usuario identificado con email diego@email.com ha procesado un texto. El sistema muestra la lista de palabras clave identificadas. El usuario decide añadir una palabra más que considera importante y no se ha encontrado. El sistema comprueba la palabra a añadir y la añade correctamente.</i>
VARIACIONES (ESCENARIOS SECUNDARIOS)	<ol style="list-style-type: none"> 1. <i>El usuario introduce una palabra vacía. El sistema impedirá la acción de añadir la palabra.</i> 2. <i>El usuario introduce una palabra que ya existe en la lista de palabras, el sistema impedirá que se añada.</i> 3. <i>El usuario introduce una palabra que no está en el texto, el sistema impedirá que se añada.</i>

Tabla 5.37. Escenario de añadir de palabras clave.

ELIMINAR PALABRAS CLAVE	
NUMERACIÓN	7.2
PRECONDICIONES	<i>El usuario debe haber introducido un texto para procesar previamente.</i>
POSCONDICIONES	-
QUIEN LO COMIENZA	<i>Usuario identificado</i>
QUIEN LO FINALIZA	<i>Sistema</i>
DESCRIPCIÓN	<i>Un usuario identificado con email diego@email.com ha procesado un texto. El sistema muestra la lista de palabras clave identificadas. El usuario decide</i>

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

VARIACIONES (ESCENARIOS SECUNDARIOS)	<p><i>seleccionar dos palabras que no considera importantes. El usuario hace clic en eliminar.</i></p> <p><i>1. El usuario no selecciona ninguna palabra, el sistema debe impedir que pueda eliminar palabras de la lista.</i></p>
--	--

Tabla 5.38. Escenario de eliminar palabras clave.

EDITAR PALABRAS CLAVE	
NUMERACIÓN	7.3
PRECONDICIONES	<i>El usuario debe haber introducido un texto para procesar previamente.</i>
POSCONDICIONES	-
QUIEN LO COMIENZA	<i>Usuario identificado</i>
QUIEN LO FINALIZA	<i>Sistema</i>
DESCRIPCIÓN	<i>Un usuario identificado con email diego@email.com ha procesado un texto. El sistema muestra la lista de palabras clave identificadas. El usuario decide editar una palabra que está mal escrita. El usuario cambia el texto y confirma la edición.</i>
VARIACIONES (ESCENARIOS SECUNDARIOS)	<p><i>1. El usuario edita la palabra y la deja vacía, el sistema impedirá que se puedan añadir palabras vacías a la lista.</i></p>

Tabla 5.39. Escenario de editar palabras clave.

SELECCIONAR PALABRAS CLAVE	
NUMERACIÓN	7.4
PRECONDICIONES	<i>El usuario debe haber introducido un texto para procesar previamente.</i>
POSCONDICIONES	<i>El número de palabras clave seleccionado se verá modificado.</i>
QUIEN LO COMIENZA	<i>Usuario identificado</i>
QUIEN LO FINALIZA	<i>Usuario identificado</i>

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

DESCRIPCIÓN	<i>Un usuario identificado con email diego@email.com ha procesado un texto. El sistema muestra la lista de palabras clave identificadas. El usuario decide seleccionar 5 de las 10 palabras clave encontradas. El usuario pasa a generar las preguntas.</i>
VARIACIONES (ESCENARIOS SECUNDARIOS)	<ol style="list-style-type: none"> 1. <i>El usuario no selecciona ninguna palabra clave, el sistema impedirá al usuario continuar ya que debe seleccionar al menos una palabra para continuar.</i> 2. <i>El usuario considera que son todas importantes y decide seleccionar todas. El sistema marca como seleccionadas todas las palabras clave.</i>

Tabla 5.40. Escenario de seleccionar palabras clave.

FILTRO DE BUSQUEDA DE PALABRAS CLAVE	
NUMERACIÓN	7.5
PRECONDICIONES	<i>El usuario debe haber introducido un texto para procesar previamente.</i>
POSCONDICIONES	-
QUIEN LO COMIENZA	<i>Usuario identificado</i>
QUIEN LO FINALIZA	<i>Usuario identificado</i>
DESCRIPCIÓN	<i>Un usuario identificado con email diego@email.com ha procesado un texto. El sistema muestra la lista de palabras clave identificadas. La lista es muy larga y el usuario decide filtrar utilizando el sistema de búsqueda. Busca solo las que contienen la cadena “die”, el sistema retorna la lista correspondiente.</i>
VARIACIONES (ESCENARIOS SECUNDARIOS)	<ol style="list-style-type: none"> 1. <i>El usuario introduce la cadena “ejemplo”, no hay ninguna palabra que contenga dicha cadena en la lista con lo cual el sistema no retorna ninguna palabra. El sistema muestra que no se ha encontrado ninguna palabra.</i>

Tabla 5.41. Escenario de filtro de búsqueda de palabras clave.

MODIFICAR NÚMERO DE PREGUNTAS A GENERAR
--

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

NUMERACIÓN	7.6
PRECONDICIONES	<i>El usuario debe haber introducido un texto para procesar previamente.</i>
POSCONDICIONES	<i>El número de preguntas a generar por cada palabra se modificará.</i>
QUIEN LO COMIENZA	<i>Usuario identificado</i>
QUIEN LO FINALIZA	<i>Usuario identificado</i>
DESCRIPCIÓN	<i>Un usuario identificado con email diego@email.com ha procesado un texto. El sistema muestra la lista de palabras clave identificadas. El usuario considera que la palabra “África” es muy importante y decide generar 3 preguntas de ella.</i>
VARIACIONES (ESCENARIOS SECUNDARIOS)	<ol style="list-style-type: none"> 1. <i>El usuario incrementa el número de 1 a 3 y deselecciona el resto de las palabras.</i> 2. <i>El usuario introduce el número de preguntas a generar directamente.</i> 3. <i>El usuario intenta introducir un número negativo o un texto. El sistema impedirá que se introduzca el texto.</i>

Tabla 5.42. Escenario de modificar número de preguntas para cada palabra.

5.4.3.7 Gestión de preguntas

MOSTRAR PREGUNTAS	
NUMERACIÓN	8.1
PRECONDICIONES	<i>El usuario debe haber seleccionado palabras clave con anterioridad para generar las preguntas.</i>
POSCONDICIONES	-
QUIEN LO COMIENZA	<i>Usuario identificado</i>
QUIEN LO FINALIZA	<i>Sistema</i>
DESCRIPCIÓN	<i>Un usuario identificado con email diego@email.com genera una serie de preguntas. A partir de las palabras</i>

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

VARIACIONES (ESCENARIOS SECUNDARIOS)

clave “Silk” y “Whooll”. El sistema muestra dos preguntas diferentes, una diferente para cada una de las palabras clave.

- El usuario selecciona muchas palabras clave. No es posible generar una pregunta para cada palabra con lo cual se generan preguntas repetidas. El sistema avisará al usuario de que pueden existir preguntas repetidas.

Tabla 5.43. Escenario de mostrar preguntas.

EDITAR ENUNCIADOS	
NUMERACIÓN	8.2
PRECONDICIONES	<i>El usuario debe haber seleccionado palabras clave con anterioridad para generar las preguntas.</i>
POSCONDICIONES	-
QUIEN LO COMIENZA	<i>Usuario identificado</i>
QUIEN LO FINALIZA	<i>Sistema</i>
DESCRIPCIÓN	<i>Un usuario identificado con email diego@email.com genera una serie de preguntas. Decide modificar los enunciados de una de ellas. El sistema validará que no sea vacío.</i>
VARIACIONES (ESCENARIOS SECUNDARIOS)	<ol style="list-style-type: none"> El usuario decide borrar todo el texto del enunciado de la pregunta, el usuario impedirá esta acción avisando al usuario.

Tabla 5.44. Escenario de modificar preguntas.

AÑADIR RESPUESTAS	
NUMERACIÓN	8.3
PRECONDICIONES	<i>El usuario debe haber seleccionado palabras clave con anterioridad para generar las preguntas.</i>
POSCONDICIONES	-
QUIEN LO COMIENZA	<i>Usuario identificado</i>

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

QUIEN LO FINALIZA	<i>Sistema</i>
DESCRIPCIÓN	<i>Un usuario identificado con email diego@email.com genera una serie de preguntas. Decide añadir nuevas respuestas a una pregunta porque considera que las generadas son insuficientes. El sistema valida que no sea vacía y añade la nueva opción.</i>
VARIACIONES (ESCENARIOS SECUNDARIOS)	<ol style="list-style-type: none"> 1. <i>El usuario decide añadir una respuesta con un texto vacío, el sistema mostrará un mensaje impidiendo que se añada la nueva opción.</i>

Tabla 5.45. Escenario de añadir respuestas.

ELIMINAR RESPUESTAS	
NUMERACIÓN	8.4
PRECONDICIONES	<i>El usuario debe haber seleccionado palabras clave con anterioridad para generar las preguntas.</i>
POSCONDICIONES	-
QUIEN LO COMIENZA	<i>Usuario identificado</i>
QUIEN LO FINALIZA	<i>Sistema</i>
DESCRIPCIÓN	<i>Un usuario identificado con email diego@email.com genera una serie de preguntas. Decide eliminar alguna respuesta a una pregunta porque considera que las generadas son demasiadas o son demasiado fáciles.</i>
VARIACIONES (ESCENARIOS SECUNDARIOS)	-

Tabla 5.46. Escenario de eliminar respuesta.

EXPORTAR PREGUNTAS	
NUMERACIÓN	8.5
PRECONDICIONES	<i>El usuario debe haber seleccionado palabras clave con anterioridad para generar las preguntas.</i>
POSCONDICIONES	-

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

QUIEN LO COMIENZA	<i>Usuario identificado</i>
QUIEN LO FINALIZA	<i>Sistema</i>
DESCRIPCIÓN	<i>Un usuario identificado con email diego@email.com genera una serie de preguntas. Decide exportar las preguntas generadas a un fichero para después incluirlas en uno de sus exámenes. Exporta las preguntas tanto en formato docx, como txt y pdf.</i>
VARIACIONES (ESCENARIOS SECUNDARIOS)	-

Tabla 5.47. Escenario de exportar preguntas.

ORDENAR PREGUNTAS	
NUMERACIÓN	8.6
PRECONDICIONES	<i>El usuario debe haber seleccionado palabras clave con anterioridad para generar las preguntas.</i>
POSCONDICIONES	-
QUIEN LO COMIENZA	<i>Usuario identificado</i>
QUIEN LO FINALIZA	<i>Usuario identificado</i>
DESCRIPCIÓN	<i>Un usuario identificado con email diego@email.com genera un par de preguntas. Hay preguntas desordenadas con lo cual decide poner arriba las más importantes y debajo las menos.</i>
VARIACIONES (ESCENARIOS SECUNDARIOS)	-

Tabla 5.48. Escenario de ordenar preguntas.

AÑADIR PREGUNTAS	
NUMERACIÓN	8.7
PRECONDICIONES	<i>El usuario debe haber seleccionado palabras clave con anterioridad para generar las preguntas.</i>

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

POSCONDICIONES	-
QUIEN LO COMIENZA	<i>Usuario identificado</i>
QUIEN LO FINALIZA	<i>Sistema</i>
DESCRIPCIÓN	<i>Un usuario identificado con email diego@email.com genera una serie de preguntas. Estas no son suficientes y decide inventarse nuevas preguntas que considera importantes. Añade dos nuevas preguntas y las edita.</i>
VARIACIONES (ESCENARIOS SECUNDARIOS)	<ol style="list-style-type: none"> 1. <i>El usuario ve que hay preguntas generadas muy similares y que quiere añadir. El usuario decide duplicar la pregunta y añadirla a la lista.</i>

Tabla 5.49. Escenario de añadir nuevas preguntas.

ELIMINAR PREGUNTAS	
NUMERACIÓN	8.8
PRECONDICIONES	<i>El usuario debe haber seleccionado palabras clave con anterioridad para generar las preguntas.</i>
POSCONDICIONES	-
QUIEN LO COMIENZA	<i>Usuario identificado</i>
QUIEN LO FINALIZA	<i>Sistema</i>
DESCRIPCIÓN	<i>Un usuario identificado con email diego@email.com genera una serie de preguntas. Algunas de las preguntas las considera irrelevantes y decide eliminarlas.</i>
VARIACIONES (ESCENARIOS SECUNDARIOS)	-

Tabla 5.50. Escenario de eliminar preguntas.

5.4.3.8 Gestión de cuestionarios

GENERAR CUESTIONARIO

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

NUMERACIÓN	9.1
PRECONDICIONES	<i>El usuario debe tener al menos una pregunta creada.</i>
POSCONDICIONES	-
QUIEN LO COMIENZA	<i>Usuario identificado</i>
QUIEN LO FINALIZA	<i>Sistema</i>
DESCRIPCIÓN	<i>Un usuario identificado con email diego@email.com genera una serie de preguntas. Decide crear un cuestionario con estas preguntas. Este cuestionario lo decide configurar por defecto dando feedback y permitiendo el salto de preguntas. El cuestionario se genera correctamente y se muestra el código del cuestionario en la pantalla para que se puedan unir usuarios.</i>
VARIACIONES (ESCENARIOS SECUNDARIOS)	<ol style="list-style-type: none"> 1. <i>El usuario activa el feedback pero no el salto de preguntas.</i> 2. <i>El usuario habilita el salto pero no el feedback.</i> 3. <i>El usuario deshabilita ambas opciones.</i> 4. <i>La sesión del usuario expira, se muestra el mensaje de error</i>

Tabla 5.51. Escenario de generar cuestionario.

VER ESTADO DE CUESTIONARIOS	
NUMERACIÓN	9.2
PRECONDICIONES	-
POSCONDICIONES	-
QUIEN LO COMIENZA	<i>Usuario identificado</i>
QUIEN LO FINALIZA	<i>Sistema</i>
DESCRIPCIÓN	<i>Un usuario identificado con email diego@email.com. Dicho usuario crea un cuestionario con 5 preguntas y les pasa el código a sus alumnos. El usuario accede a la vista de ver resultados y monitoriza el estado del cuestionario. Visualiza que un alumno lleva contestadas 5 preguntas y ve sus credenciales.</i>

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

VARIACIONES (ESCENARIOS SECUNDARIOS)	1. <i>No hay ningún usuario en el cuestionario. El sistema mostrará que no hay usuarios que se hayan unido al cuestionario.</i>
--	---

Tabla 5.52. Escenario de ver estado de cuestionario.

CERRAR CUESTIONARIO	
NUMERACIÓN	9.3
PRECONDICIONES	<i>El usuario debe tener al menos un test creado.</i>
POSCONDICIONES	-
QUIEN LO COMIENZA	<i>Usuario identificado</i>
QUIEN LO FINALIZA	<i>Sistema</i>
DESCRIPCIÓN	<i>Un usuario identificado con email diego@email.com. Dicho usuario crea un cuestionario con 10 preguntas y les pasa el código a sus alumnos. El usuario accede a la vista de ver resultados y monitoriza el estado del cuestionario. Se da cuenta de que el cuestionario está mal creado, con lo cual decide cerrarlo.</i>
VARIACIONES (ESCENARIOS SECUNDARIOS)	-

Tabla 5.53. Escenario de cerrar cuestionario.

UNIRSE A CUESTIONARIO	
NUMERACIÓN	9.4
PRECONDICIONES	-
POSCONDICIONES	-
QUIEN LO COMIENZA	<i>Usuario no identificado</i>
QUIEN LO FINALIZA	<i>Sistema</i>
DESCRIPCIÓN	<i>Un usuario no identificado, es alumno de una asignatura de la universidad. Recibe un código “prueba-test-repositorios-información”, el usuario accede a la vista para entrar al cuestionario e introduce</i>

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

VARIACIONES (ESCENARIOS SECUNDARIOS)	<p><i>el código y su identificador personal “UO276406”. Se une al test correctamente.</i></p> <ol style="list-style-type: none"> <i>El código del cuestionario no existe, el sistema mostrará un mensaje de error.</i> <i>El usuario ya ha terminado el cuestionario e intenta volver a hacerlo, se muestra un mensaje de error.</i> <i>El usuario escanea el código y accede directamente a la vista de entrar al cuestionario donde pone su UO personal.</i> <i>El usuario hace clic en el enlace proporcionado y accede a la vista del cuestionario e introduce su UO personal.</i> <i>El usuario había dejado a medias el cuestionario y decide volver a entrar, introduce de nuevo los datos y el sistema reanuda el cuestionario en el estado anterior.</i>
--	--

Tabla 5.54. Escenario de unirse a cuestionario.

CONTESTAR A PREGUNTA	
NUMERACIÓN	9.5
PRECONDICIONES	<i>El usuario debe haber entrado al cuestionario.</i>
POSCONDICIONES	-
QUIEN LO COMIENZA	<i>Usuario no identificado</i>
QUIEN LO FINALIZA	<i>Sistema</i>
DESCRIPCIÓN	<i>Un usuario no identificado, se encuentra en la primera pregunta de un cuestionario sobre sistemas operativos. El usuario marca la respuesta c), el sistema almacena mediante una petición a la base de datos la opción marcada.</i>
VARIACIONES (ESCENARIOS SECUNDARIOS)	<ol style="list-style-type: none"> <i>El usuario tras haber contestado decide limpiar la pregunta ya que prefiere no contestarla. El sistema actualiza la elección del usuario en la base de datos.</i>

Tabla 5.55. Escenario de contestar a pregunta.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

ENVIAR CUESTIONARIO

NUMERACIÓN	9.5
PRECONDICIONES	<i>El usuario debe haberse unido a un cuestionario</i>
POSCONDICIONES	-
QUIEN LO COMIENZA	<i>Usuario no identificado</i>
QUIEN LO FINALIZA	<i>Sistema</i>
DESCRIPCIÓN	<i>Un usuario no identificado está contestando preguntas de un cuestionario de Diseño del Software. El usuario termina y decide enviar el cuestionario. El cuestionario se ha configurado para obtener dar feedback, el usuario aprueba con 9 preguntas acertadas de 10. El sistema muestra la nota y las preguntas acertadas y falladas tras haber guardado en la base de datos los resultados.</i>
VARIACIONES (ESCENARIOS SECUNDARIOS)	<ol style="list-style-type: none"> 1. <i>El usuario autenticado decide cerrar antes de que se haya enviado el cuestionario. El sistema mostrará un mensaje indicando que no se puede enviar el cuestionario.</i> 2. <i>El cuestionario está configurado para no mostrar feedback con lo cual muestra un mensaje de que se ha enviado y permite la opción de volver a la página principal.</i>

Tabla 5.56. Enviar cuestionario.

5.5 Identificación de Subsistemas a Nivel de Arquitectura de la Aplicación

En este apartado se expondrán los subsistemas que forman parte del sistema. De esta forma se podrá analizar la estructura general del mismo. Esta representación está simplificada, ya que es fase de análisis, se expondrá en más detalle en el capítulo de diseño:

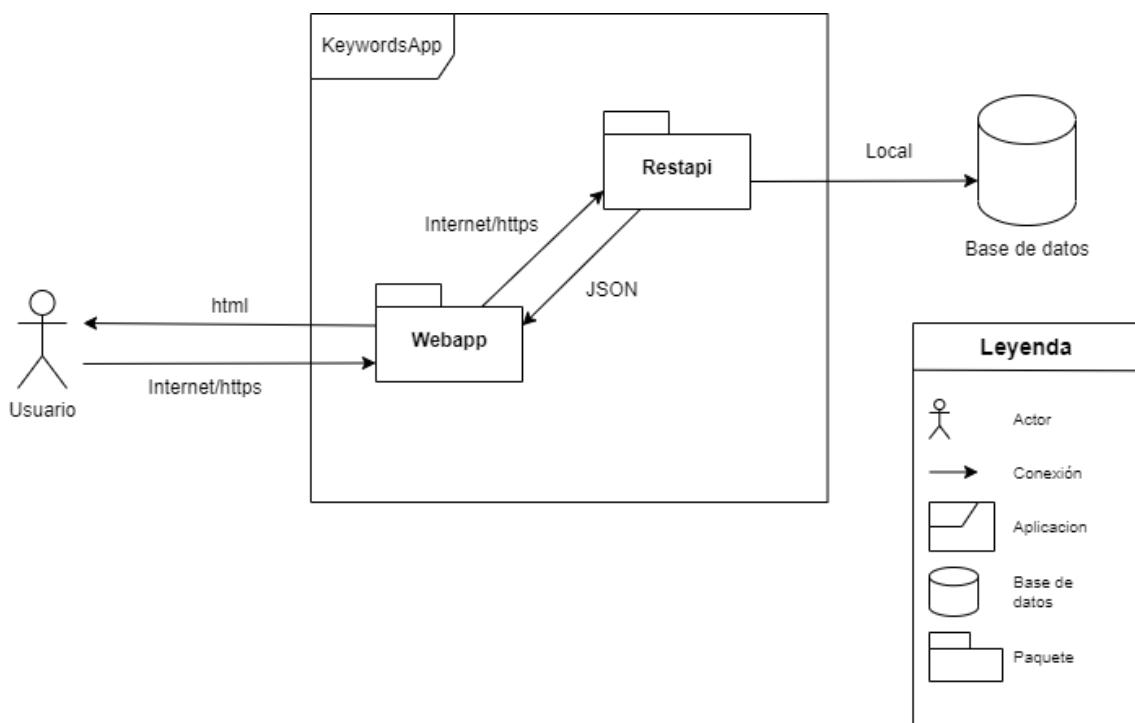


Figura 5.6. Subsistemas de arquitectura con interfaces entre estos.

5.5.1 Descripción de los Subsistemas a Nivel de Arquitectura

A continuación, se describen los diferentes subsistemas identificados en la figura anterior. Estos serían los subsistemas a nivel de arquitectura de la aplicación, los cuales se considerarán para implementar:

- **Webapp:** Corresponde a todos los componentes que conforman parte de la interfaz de usuario. Tal y como se puede apreciar será el paquete con el que interactúe el usuario.
- **Restapi:** Es el subsistema que contiene la infraestructura de comunicación con la base de datos y se encarga de suministrar los datos que requiera la parte de interfaz de usuario. A este subsistema de la aplicación podrá accederse tanto a través de la interfaz de usuario como a través de peticiones http. Será independiente del subsistema de la webapp.

La webapp se comunicará con el subsistema restapi a través de peticiones http, esto permitirá que puedan estar distribuidos y separados en máquinas diferentes. La restapi a su vez enviará sus datos usando el formato JSON (*Java Script Object Notation*). Esta también se encargará de guardar la información y comunicarse con la base de datos.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

5.6 Definición de Interfaces de Usuario

5.6.1 Descripción de la Interfaz

En este apartado de descripción de la interfaz se mostrarán una serie de esquemas de las pantallas de la aplicación de forma que se pueda identificar rápidamente la disposición de los elementos en la aplicación. A continuación, se muestran los correspondientes bocetos con algunas explicaciones de cada boceto:

Menú de opciones principal

Email:

Contraseña:

Iniciar Sesión

Footer

Figura 5.7. Boceto de la página de inicio de sesión.

Esta vista, permitirá la identificación de los usuarios en la aplicación. Estos iniciarán sesión mediante sus credenciales. Una vez se inicie sesión, los usuarios podrán acceder a sus datos y la información almacenada en la base de datos, asimismo, también podrán acceder a la creación de cuestionarios.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

Signin

← → C

Menú de opciones principal

Nombre:

Primer apellido:

Segundo apellido:

Email:

Contraseña:

Repita su contraseña:

Registrarse

Footer

Figura 5.8. Boceto de la página de registro.

La vista de registro está destinada al alta de nuevos usuarios en el sistema. Para ello deberán introducir sus datos personales, email y contraseña. El sistema se encargará de hacer las comprobaciones oportunas como que no sean vacíos, el email no esté ya registrado o que la contraseña cumpla ciertos requisitos básicos de seguridad.

Home

← → C

Menú de opciones principal

Página principal

En esta sección se mostrará información general de la página web como por ejemplo:

- Tips generales de uso
- Ventajas de la aplicación
- Datos de cómo funciona y tecnologías utilizadas

Footer

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

Figura 5.9. Boceto de la página principal.

La página principal será introductoria para los usuarios que accedan a la aplicación. Contendrá información del uso así como una breve descripción del sistema. Estará orientada a dar una sensación amigable a los usuarios y será la primera toma de contacto de estos con el sistema.



Figura 5.10. Boceto de la vista de procesamiento del texto.

Esa vista será el primer paso a la hora de crear los exámenes, los usuarios en esta vista introducirán el texto a partir del cual obtendrán las palabras clave. Podrán cargar ficheros de texto, y escribir texto. Una vez haya texto cargado, el sistema lo procesará y obtendrá aquellas palabras de mayor relevancia según el algoritmo desarrollado en el apartado 2.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

The mockup illustrates a user interface for keyword management. At the top, there's a header bar with a logo, the school name, and the title 'ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO'. Below the header is the author's information: 'DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es'. The main area is titled 'Menú de opciones principal'. It includes a note about a visual schema for navigation and a list of identified keywords ('keyword1' through 'keyword8'). A sidebar provides options to edit, add, or remove keywords. The bottom section is labeled 'Footer'.

Figura 5.11. Boceto de vista de gestión de palabras clave.

Esta vista permite la selección de las palabras clave que los usuarios quieren para llevar a cabo la creación de preguntas. Cada una de las palabras se marcará como seleccionada. También habrá opciones para añadir, eliminar e incluso modificar las palabras de la lista haciendo doble clic. Asimismo, el sistema permitirá volver a la página de procesamiento de texto por si el usuario detecta algún error en el texto previamente introducido.

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

Preguntas

Menú de opciones principal

Esquema visual de pasos que tiene que seguir el usuario, servirá de ayuda para indicarle dónde se encuentra

Enunciado de la pregunta1

- Opción1
- Opción2
- ...

Enunciado de la pregunta2

- Opción1
- Opción2
- ...

...

Cada pregunta podrá ser editada por el usuario

Menú de opciones de exportación de las preguntas

Footer

Figura 5.12. Boceto de la página de gestión de preguntas.

La vista de gestión de preguntas mostrará la lista de preguntas propuestas por el sistema. El usuario tendrá la opción para eliminar preguntas, añadir nuevas y duplicar las existentes. Asimismo, también podrá modificar las opciones con doble clic, añadir nuevas opciones y eliminar existentes. Asimismo, el usuario podrá reordenar estas preguntas con un sistema de *drag and drop*.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

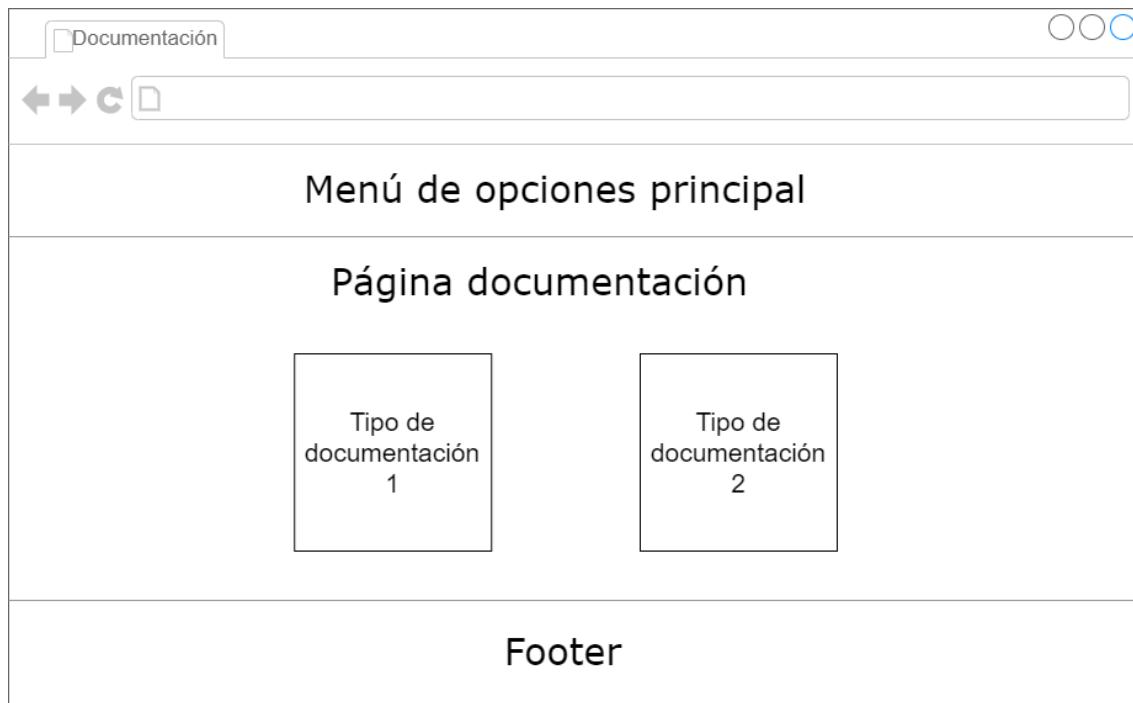


Figura 5.13. Boceto de página de documentación.

La página de documentación permite acceder a dos enlaces que contienen la documentación de la restapi. Esta página está más orientada a aquellos usuarios que buscan atacar directamente la restapi sin usar la webapp. Habrá dos tipos de documentación, muy similares, para ello el usuario hará clic y el sistema le redirigirá a la página que contiene la documentación necesaria.

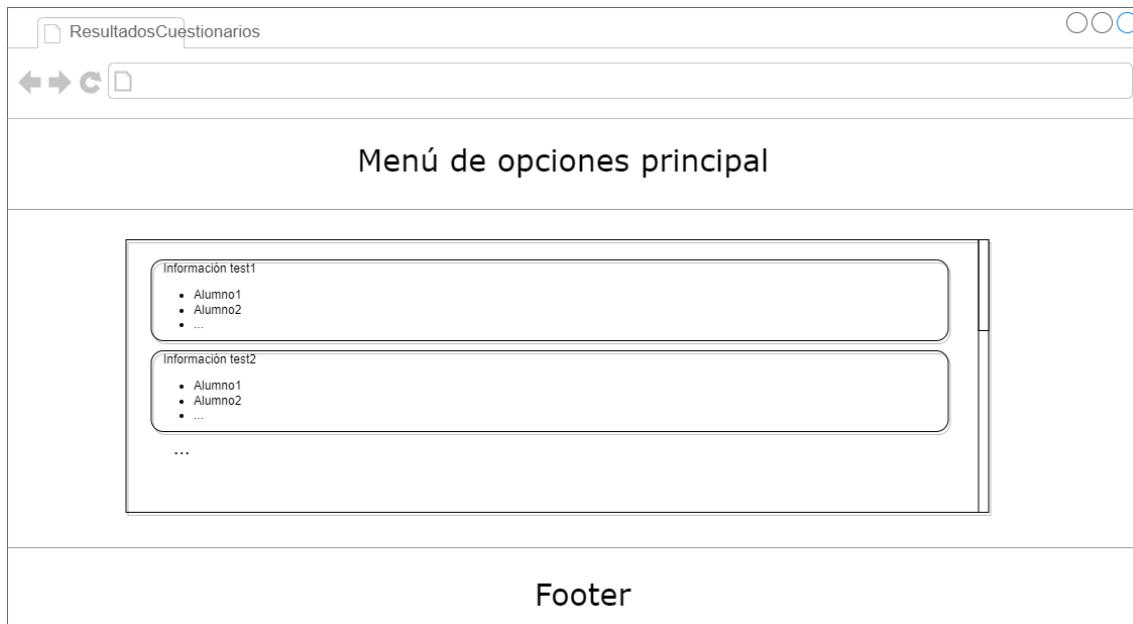


Figura 5.14. Boceto de vista de ver resultados de cuestionarios.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

La vista de ver los resultados de los cuestionarios estará disponible para los usuarios autenticados. Estos podrán desde esta vista monitorizar, aquellos cuestionarios que hayan creado, los alumnos que se han unido al test así como las notas obtenidas por los propios usuarios.

Podrán acceder al código de acceso de cada cuestionario, podrán actualizar los resultados para monitorizar en tiempo real todas aquellas cuestiones interesantes, asimismo, también podrán abrir o cerrar el cuestionario según proceda para restringir o no el acceso de más usuarios.

UnirseCuestionario

← → C □

Menú de opciones principal

Id test: _____

Id usuario: _____

Unirse a test

Footer

Figura 5.15. Boceto de vista de entrar a cuestionario.

Esta vista estará disponible para los usuarios que no se hayan autenticado, concretamente para los alumnos que tengan que unirse a un cuestionario. Podrán acceder directamente desde el menú de navegación o a través de un código de acceso.

Para unirse a un cuestionario el sistema verificará el código de acceso, así como el identificador del usuario. Presumiblemente, este id sería algo como el UO del alumno o similar que permita obtener la información del usuario.



Menú de opciones principal

Enunciado pregunta...

- Opción1
- Opción2
- Opción3
- Opción4

Mapa de preguntas

Footer

Figura 5.16. Boceto de la vista de contestar pregunta en el cuestionario.

Esta vista estará disponible cuando un usuario acceda a la realización de un cuestionario. El usuario podrá pasar de una pregunta a otra, enviar el cuestionario, o saltar preguntas utilizando el mapa de preguntas (en función de la configuración del cuestionario). Cada vez que se conteste una pregunta el sistema guardará la pregunta. Del mismo modo, esta se marcará como contestada. Una vez se envíe el cuestionario, el sistema calculará la nota y mostrará el *feedback* o no en función de la configuración del cuestionario.

5.6.2 Diagrama de Navegabilidad

En este apartado se representa de forma esquemática la navegabilidad entre ventanas de la aplicación. Es decir, se indica a dónde se puede acceder y desde donde:

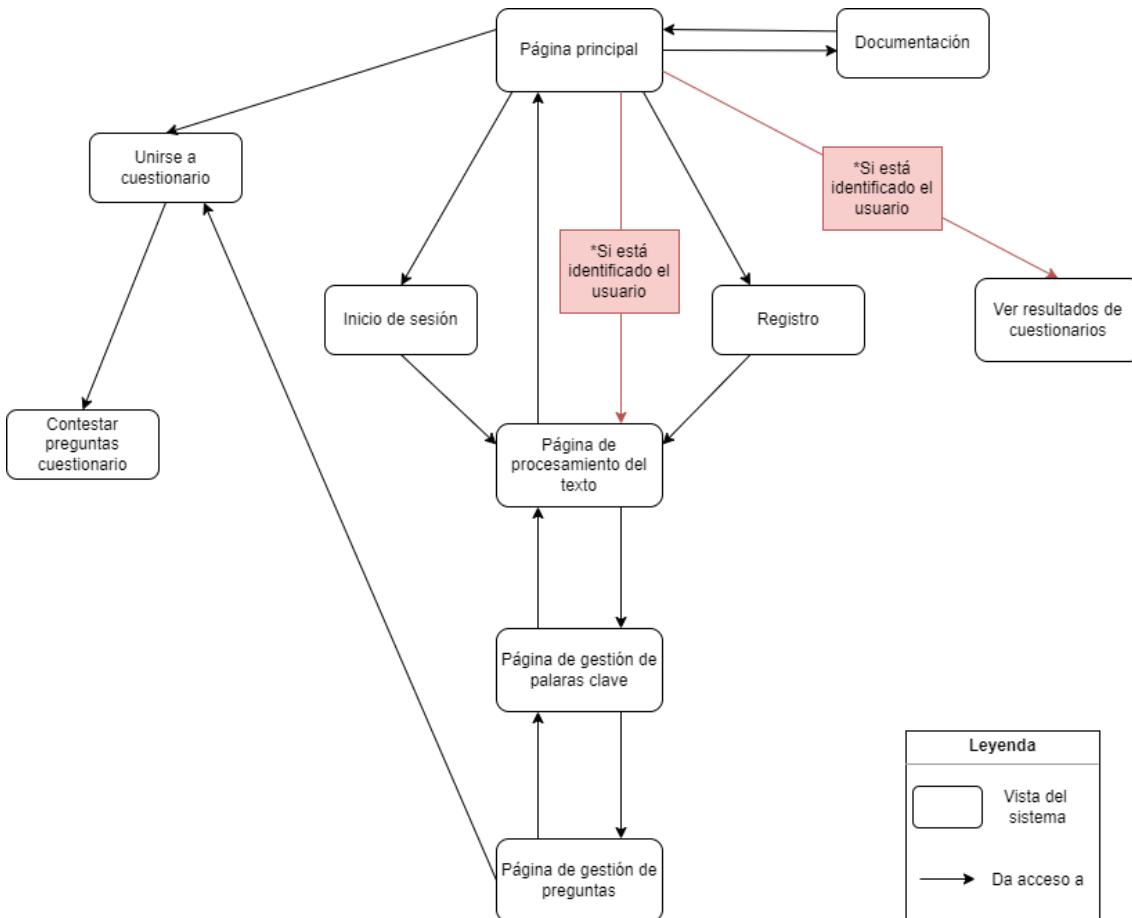


Figura 5.17. Diagrama de navegabilidad del sistema.

5.7 Especificación del plan de pruebas

En este apartado se detallan las pruebas que se llevarán a cabo a lo largo de la implementación:

- Estáticas:** Se realizarán pruebas estáticas de código utilizando análisis. Esto permitirá detectar diferentes tipos de defectos en el código que puedan ser corregidos de forma que se reduzca la deuda técnica lo más posible. Se usará la herramienta SonarCloud sincronizada con el repositorio en GitHub.
- Unitarias:** Estas pruebas funcionales se realizarán sobre el módulo de la restapi. Es decir, se probarán las peticiones que ofrece la restapi con el fin de verificar su funcionamiento así como el modelo de dominio. Estarán disponibles en un directorio “/restapi/tests” y estarán documentadas en el código. Se realizarán utilizando la librería de Python, TestClient y estarán automatizadas con el comando pytest. Para ello, se ha seguido la documentación disponible en la documentación oficial [23].

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

- **Sistema – Funcionales:** Pruebas funcionales (*end-to-end*) realizadas desde la interfaz web donde se probarán los escenarios principales de la aplicación. Se realizarán pruebas automáticas y manuales. Para ello se usará la librería Puppeteer para probar aplicaciones de forma similar a Selenium. Las situaciones de prueba que se contemplan se especifican en el apartado de 6.4. Asimismo, estas pruebas se detallan en los ficheros de la webapp que terminan en “.test.js”.
- **Sistema – Pruebas con usuarios:** Se realizarán pruebas con usuarios utilizando cuestionarios. Para ello se utilizarán diferentes tipos de usuarios los cuales nos den *feedback* sobre las posibles dificultades que tengan al usar el sistema.
- **Sistema – Adaptabilidad:** Pruebas de adaptabilidad en diferentes resoluciones para las vistas del sistema. Estas se harán de forma manual utilizando las funcionalidades que ofrecen diferentes herramientas como es <http://responsivetesttool.com>. Estas se harán para comprobar la disposición en los diferentes elementos del sitio, ya que en muchos casos se accederá a través de dispositivos móviles para realizar los cuestionarios.

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

6 Diseño del Sistema de Información

6.1 Diseño de Físico de datos

6.1.1 Descripción del SGBD utilizado

Para el desarrollo de la aplicación web en cuestión se ha utilizado un Sistema Gestor de Bases de Datos (SGBD) relacional, en concreto, durante el desarrollo se ha utilizado SQLite. Este SGBD es el más utilizado en el mundo, es multiplataforma y muy ligero de forma que permite una gran portabilidad.

Asimismo SQLite fue lanzado en el año 2000. Con lo cual goza de una madurez más que notable e ideal en cuanto a fiabilidad se refiere, al igual que el resto de los sistemas relacionales de persistencia. Asimismo, es un SGBD ideal para sitios web con un tráfico medio. Es capaz de soportar entre 400.000 y 500.000 peticiones http al día [24]. Se puede ver mucha más información sobre SQLite en el enlace disponible en la referencia [24].

6.1.2 Integración del SGDB en el sistema

La integración del Sistema Gestor de Bases de Datos en la aplicación web se ha hecho a través de la librería SQLAlchemy de Python. A través de esta librería, utilizando su ORM (*Object Relational Mapper*) se ha generado el modelo de la base de datos. Este modelo es sobre el que se autogeneran todas las relaciones y tablas del sistema. De esta forma cada vez que se arranca la API REST, se generan, si no lo están, todas las relaciones y tablas.

Asimismo, este sistema nos permite una persistencia automática de objetos en Python, similar a como se haría en Java utilizando JPA. Toda la documentación sobre SQLAlchemy está disponible en [25].

Aun así para entender de forma simple cómo funciona un ORM en cualquier lenguaje podemos verlo como una caja negra que funciona de la siguiente forma:

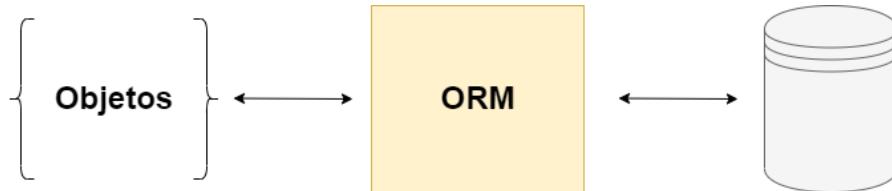


Figura 6.1. Funcionamiento ORM.

Un ORM, tal y como se ve en la figura anterior, actúa como intermediario que permite traducir objetos (datos de un lenguaje de programación) a datos persistentes que son entendidos por un sistema gestor de bases datos relacional.



6.1.3 Diagrama E-R

El diagrama entidad relación muestra una abstracción de las entidades de la base de datos del sistema. Se indican tanto las claves primarias (PK) como las claves foráneas (FK) las cuales permiten identificar únicamente a una fila de cada tabla o referenciar a otras filas de otras tablas respectivamente.

Asimismo, hay que destacar que una clave primaria puede estar formada por varios campos, no solo por uno. Es especialmente interesante en el caso de “TestStudent” y “StudentQuestion” que son tablas que surgen por relaciones N-N entre las entidades Student-Test y Student-Question respectivamente.

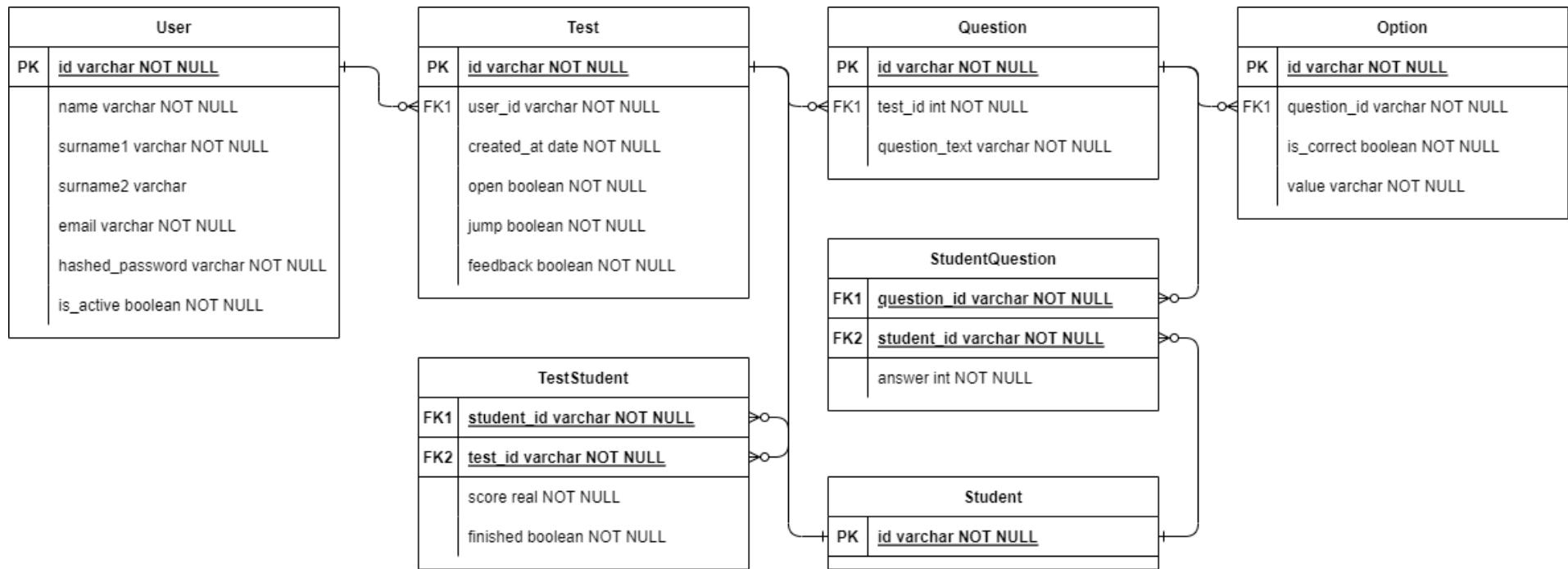


Figura 6.2. Diagrama Entidad-Relación

6.2 Arquitectura de Módulos del Sistema

6.2.1 Diagrama de componentes Webapp

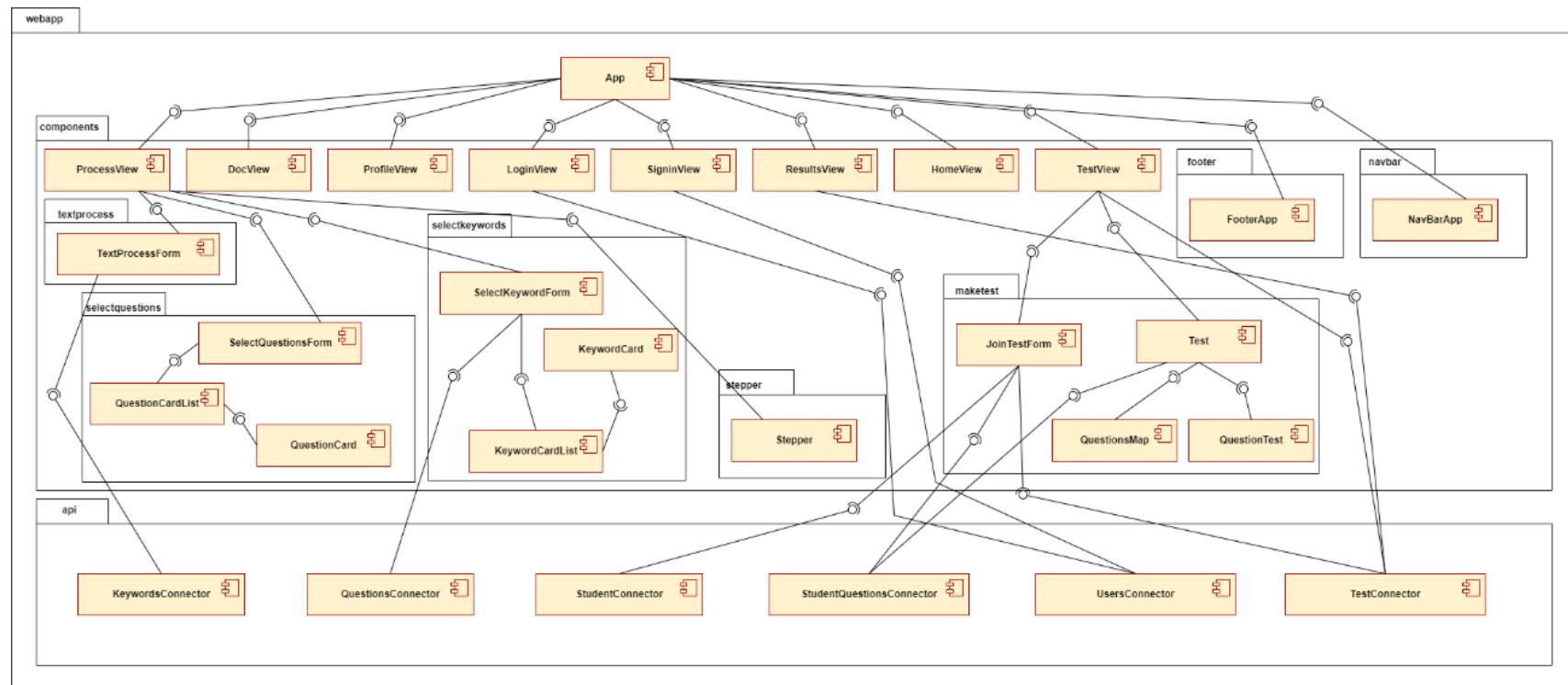


Figura 6.3. Diagrama de componentes de la webapp.

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

- **App:** es el componente de React sobre el que se cargan las diferentes vistas y se configuran el resto de las rutas del sistema.
- **Components:** en este paquete se encuentran los componentes de React que se renderizan en la interfaz de usuario.
 - **HomeView:** es el componente que renderiza la página principal.
 - **DocView:** componente que renderiza la vista de documentación de la API.
 - **ProfileView:** componente que renderiza los datos del perfil del usuario.
 - **LoginView y Sign View:** son componentes que contienen los formularios de inicio de sesión y registro respectivamente.
 - **ResultsView:** es el componente encargado de mostrar al profesor los resultados de los alumnos que se unen a cada cuestionario.
 - **FooterApp:** componente con el pie de página de la aplicación web.
 - **NavBarApp:** componente con el menú de navegación del sistema.
 - **ProcessView:** componente para el proceso de creación de preguntas.
 - **Stepper:** contiene la guía de pasos durante el proceso de procesar un texto, seleccionar las palabras clave, etc.
 - **TextProcessForm:** componente que contiene el formulario para cargar un texto sobre el que buscar las palabras clave.
 - **SelectKeywordForm:** contiene el formulario sobre el que se pueden manipular las palabras clave encontradas.
 - **KeywordCardList:** componente con la lista de palabras clave encontradas (estará formada por muchas KeywordCards).
 - **KeywordCard:** componente que contiene información de cada una de las palabras.
 - **SelectQuestionForm:** contiene el formulario sobre el que se pueden manipular las preguntas propuestas.
 - **QuestionCardList:** componente con la lista de preguntas propuestas (cada pregunta será una QuestionCard).
 - **QuestionCard:** componente que contiene información de cada una de las preguntas.
- **Api:** en este paquete se encuentran diferentes clases que permiten la conexión con la restapi mediante peticiones http. De esta forma la webapp solo se comunica desde este punto con la restapi, desacoplando estos dos subsistemas.
 - **KeywordsConnector:** hace peticiones para la búsqueda de palabras clave.
 - **QuestionsConnector:** hace peticiones para la generación de preguntas.
 - **StudentConnector:** hace peticiones en relación con los usuarios que se unen a los cuestionarios.
 - **StudentQuestionsConnector:** permite hacer peticiones con información de cada pregunta que responde un alumno.
 - **UsersConnector:** hace peticiones de alta e inicio de sesión de usuarios.
 - **TestConnector:** hace peticiones basadas en los cuestionarios que crea el usuario.

6.2.2 Diagrama de componentes Restapi

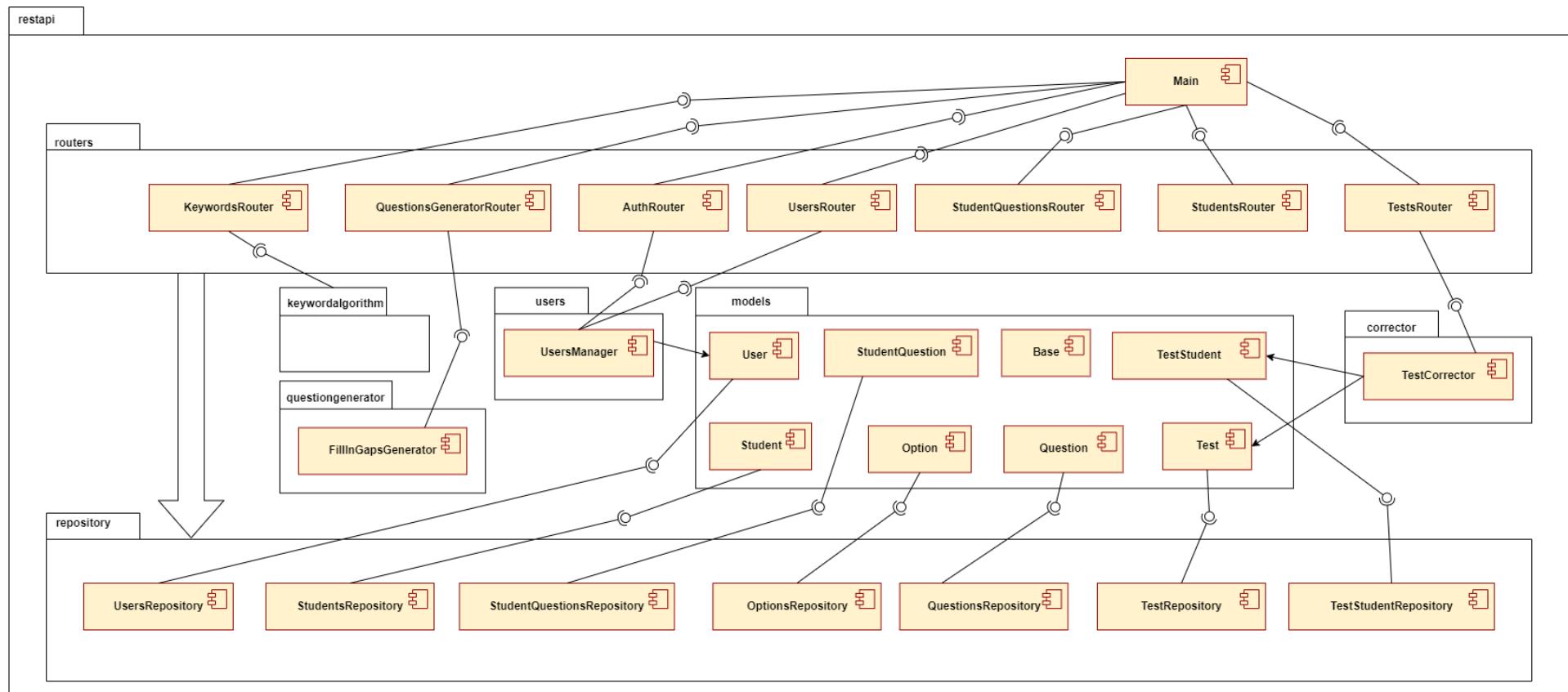


Figura 6.4. Diagrama de componentes de la restapi.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

- **Main:** este componente es el fichero Python, que se encarga de arrancar la restapi así como hacer todas las inicializaciones correspondientes.
- **Routers:** en este paquete estarán los manejadores de cada una de las peticiones que se pueden hacer a la restapi. Básicamente, divididos según la ruta. Estos ficheros serán los que se encarguen de enviar peticiones a los ficheros repository.
 - **KeywordsRouter:** maneja las rutas referidas al manejo y extracción de palabras clave.
 - **QuestionsGeneratorRouter:** maneja las peticiones para la generación de preguntas, principalmente a partir de las palabras clave anteriores y el texto.
 - **AuthRouter:** permite la gestión de rutas de autorización de usuarios.
 - **UsersRouter:** permite el manejo de rutas para el alta, eliminación, y demás opciones de los usuarios.
 - **StudentQuestionsRouter:** manejador para añadir o actualizar respuestas de estudiantes en cuestionarios concretos.
 - **TestsRouter:** permite manejar rutas para la gestión de cuestionarios, añadir nuevos, corregir, modificar, etc.
- **Models:** este paquete contendrá los ficheros con código Python que sirven para generar la estructura de la base de datos, así como el manejo de la persistencia. Existe un fichero para cada una de las tablas de la base de datos. Se puede ver información de las relaciones entre cada modelo en el diagrama entidad-relación de la Figura 6.2.
- **Repository:** este paquete contendrá diferentes ficheros con funciones que permiten el acceso a la base de datos. Serán los únicos ficheros que se comuniquen con la base de datos. Existirá un fichero para cada una de las tablas de la base de datos.
- **Users:** Este paquete contendrá la lógica de negocio para el manejo de usuarios, principalmente aspectos de configuración.
- **QuestionGenerator:** este paquete contendrá clases Python con la lógica de negocio para generar preguntas.
 - **FillInGapsGenerator:** contiene la lógica de negocio para la creación de preguntas de llenar con la palabra que falta. Añadir nuevos tipos de preguntas implicaría crear otro fichero similar.
- **KeywordAlgorithm:** Este paquete contiene la lógica de negocio para generar las palabras clave, se corresponde con el diagrama expuesto en la Figura 2.5. Diagrama de componentes del algoritmo.
- **Corrector:** este paquete contendrá clases en código Python con la lógica de negocio para llevar a cabo la corrección de cuestionarios. Este paquete recibirá las respuestas de un estudiante concreto para un cuestionario específico y calculará la nota del estudiante.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

6.3 Diseño de la API REST

En este apartado, se explicarán el diseño de las rutas de la restapi, con los posibles retornos y demás datos de cada una. Todas ellas están organizadas según la funcionalidad.

6.3.1 Gestión de palabras clave

6.3.1.1 Encontrar palabras clave

- **Ruta:** /api/v1.0/keywords/find
- **Método HTTP:** POST
- **Parámetros:**

```
{
  "text_body": "string"
}
```

- **Retorno**
 - Lista de palabras clave (código 200)
 - Error de validación (código 422)
 - No autorizado (401)

6.3.2 Gestión de preguntas

6.3.2.1 Generar preguntas

- **Ruta:** /api/v1.0/questions/generate
- **Método HTTP:** POST
- **Parámetros:**

```
{
  "text_body": "string",
  "keywords_selected": [
    {
      "value": "string",
      "numberOfQuestions": 0
    }
  ]
}
```

- **Retorno:**
 - lista de preguntas con las respectivas opciones (código 200)
 - Error de validación (código 422)
 - No autorizado (401)

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

6.3.3 Gestión de usuarios

6.3.3.1 Inicio de sesión

- **Ruta:** /api/v1.0/auth/login
- **Método HTTP:** POST
- **Parámetros:**

```
{
  "username": "string",
  "password": "string",
}
```

- **Retorno:**
 - Access Token (200)
 - Credenciales incorrectas (código 400)
 - Error de validación (código 422)

6.3.3.2 Registro

- **Ruta:** /api/v1.0/auth/register
- **Método HTTP:** POST
- **Parámetros:**

```
{
  "email": "user@example.com",
  "password": "string",
  "is_active": true,
  "is_superuser": false,
  "is_verified": false,
  "name": "string",
  "surname1": "string",
  "surname2": "string"
}
```

- **Retorno:**
 - Datos del usuario (201)
 - Credenciales incorrectas (código 400)
 - Error de validación (código 422)

6.3.3.3 Salir de sesión

- **Ruta:** /api/v1.0/auth/logout
- **Método HTTP:** POST
- **Parámetros:** -

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

- **Retorno:**
 - Sale correctamente (200)
 - Falta el token (código 401)
 - No autorizado (código 401)

6.3.4 Gestión de tests

6.3.4.1 Añadir test

- **Ruta:** /api/v1.0/test/add
- **Método HTTP:** POST
- **Parámetros:**

```
{
  "questions": [
    {
      "question": "string",
      "options": [
        {
          "value": "string",
          "correct": true
        }
      ]
    }
  ],
  "jump": true,
  "feedback": true
}
```

- **Retorno:**
 - Creado correctamente (201)
 - Error de validación (código 422)
 - No autorizado (código 401)

6.3.4.2 Obtener tests usuario

- **Ruta:** /api/v1.0/test/find
- **Método HTTP:** GET
- **Parámetros:**
- **Retorno:**
 - Lista de la información de los cuestionarios del usuario (200)
 - No autorizado (código 401)

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

6.3.4.3 Obtener test concreto

- **Ruta:** /api/v1.0/test/{test_id}
- **Método HTTP:** GET
- **Parámetros:** test_id
- **Retorno:**
 - Información del test concreto (200)

6.3.4.4 Corregir test

- **Ruta:** /api/v1.0/test/check
- **Método HTTP:** POST
- **Parámetros:**

```
{
  "testId": "string",
  "studentId": "string",
  "selection": [
    0
  ]
}
```

- **Retorno:**
 - Enviado correctamente (201)
 - Error de validación (código 422)

6.3.4.5 Obtener resultados

- **Ruta:** /api/v1.0/test/find/results
- **Método HTTP:** GET
- **Parámetros:** -
- **Retorno:**
 - Resultados cuestionarios (200)
 - No autorizado (código 401)

6.3.4.6 Cambiar estado test

- **Ruta:** /api/v1.0/test/changestatus
- **Método HTTP:** POST
- **Parámetros:**

```
{
  "id": "string",
  "open": true
}
```

- **Retorno:**
 - Cambiado estado (200)
 - Error de validación (código 422)
 - No autorizado (código 404)

6.3.4.7 Añadir estudiante a test

- **Ruta:** `//api/v1.0/students/add/{student_id}/test/{test_id}`
- **Método HTTP:** GET
- **Parámetros:** student_id, test_id
- **Retorno:**
 - Añadido correctamente (201)
 - Error de validación (código 422)

6.4 Diagrama de despliegue

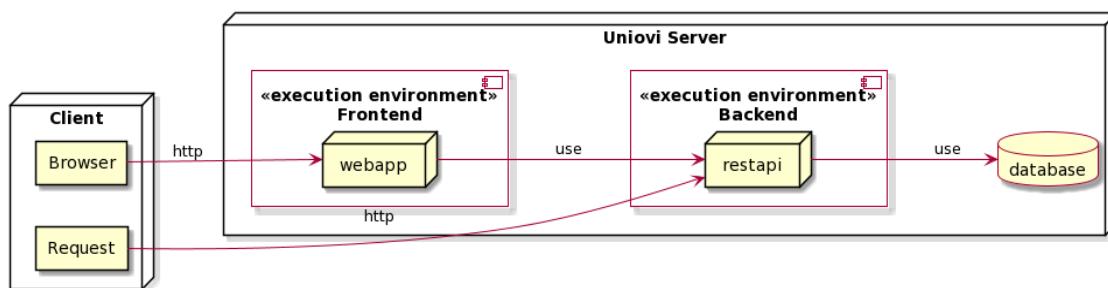


Figura 6.5. Diagrama de despliegue del sistema.

El servidor en el que se despliega el sistema será un servidor en la intranet de la universidad, con lo cual para acceder al sistema será necesario estar conectado la red corporativa ya sea desde una VPN o mismamente desde la propia red wifi.

En este servidor, estarán instaladas todas las dependencias para el correcto funcionamiento tanto de la webapp como de la restapi. La webapp y la restapi a pesar de estar en la misma máquina, se ejecutarán en dos procesos diferentes y en dos puertos diferentes que escucharán las diferentes peticiones. Los clientes ya sea mediante un navegador web o con peticiones http se conectarán a la webapp o a la restapi respectivamente. Finalmente, la restapi utilizará la base de datos a la cual se conectará para hacer labores de persistencia de datos.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

6.5 Diseño del Plan de Pruebas

Es recomendable para entender el diseño del plan de pruebas y conocer ciertos conceptos de pruebas en ingeniería del software, echar un ojo al anexo del apartado 14.4 donde se describe el proceso de diseño seguido.

6.5.1 Pruebas Unitarias

6.5.1.1 Restapi

Estas pruebas se realizarán automatizadas para las rutas más importantes de la restapi y se documentan en el propio código. Concretamente en el directorio “/restapi/tests”.

6.5.2 Pruebas de Sistema

6.5.2.1 Pruebas Funcionales (end-to-end)

Estas se dividirán según los diferentes casos de uso de mayor funcionalidad identificados en la fase de análisis, se subrayan además aquellas situaciones de prueba que se han automatizado con la librería Puppeteer. Por otro, lado, aquellas situaciones no cubiertas con pruebas automatizadas se realizarán de forma manual:

Gestión de usuarios	
P001: Inicio de sesión	
Condiciones de prueba y clases de equivalencia	Situaciones de prueba
<ul style="list-style-type: none"> • Email <ul style="list-style-type: none"> ○ Email válido ○ Email no existente • Contraseña <ul style="list-style-type: none"> ○ Contraseña válida ○ Contraseña incorrecta 	<ul style="list-style-type: none"> • Combinación parcial <ul style="list-style-type: none"> ○ Email válido <ul style="list-style-type: none"> ▪ <u>Contraseña válida</u> ▪ <u>Contraseña incorrecta</u> ○ <u>Email no válido</u>
P002: Registro	
Condiciones de prueba y clases de equivalencia	Situaciones de prueba



<ul style="list-style-type: none"> • Email <ul style="list-style-type: none"> ○ Email ya registrado ○ Email sin formato correcto ○ Email válido • Contraseña <ul style="list-style-type: none"> ○ Sin una mayúscula ○ Con menos de 6 caracteres ○ Sin un dígito ○ Contraseña válida • Confirmación de contraseña <ul style="list-style-type: none"> ○ Coincide con la contraseña ○ No coincide con la contraseña 	<ul style="list-style-type: none"> • Combinación parcial <ul style="list-style-type: none"> ○ <u>Email ya registrado</u> ○ Email válido <ul style="list-style-type: none"> ▪ <u>Contraseña sin mayúscula</u> ▪ <u>Contraseña con menos de 6 caracteres</u> ▪ <u>Contraseña sin un dígito</u> ▪ <u>Contraseña con espacio en blanco</u> ▪ Contraseña válida <ul style="list-style-type: none"> • Confirmación de contraseña coincide • <u>Confirmación de contraseña no coincide</u>
--	---

Tabla 6.1. Situaciones de prueba para la gestión de usuarios.

Procesamiento de texto	
P003: Carga de fichero	
Condiciones de prueba y clases de equivalencia	Situaciones de prueba
<ul style="list-style-type: none"> • Tipo de fichero <ul style="list-style-type: none"> ○ Válido (txt) ○ No válido (Otro) • Longitud de texto en fichero <ul style="list-style-type: none"> ○ Vacío ○ No vacío 	<ul style="list-style-type: none"> • Combinación parcial <ul style="list-style-type: none"> ○ Válido <ul style="list-style-type: none"> ▪ Vacío ▪ No vacío ○ No válido
P004: Eliminar todo el texto	
Condiciones de prueba y clases de equivalencia	Situaciones de prueba
<ul style="list-style-type: none"> • Texto escrito <ul style="list-style-type: none"> ○ Vacío ○ No vacío 	<ul style="list-style-type: none"> • Estrategia minimizada <ul style="list-style-type: none"> ○ <u>Texto Vacío</u> ○ <u>Texto No vacío</u>
P005: Carga de texto	

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

Condiciones de prueba y clases de equivalencia	Situaciones de prueba
<ul style="list-style-type: none"> Tipo de entrada de texto <ul style="list-style-type: none"> Escrito (por teclado) Texto de prueba 	<ul style="list-style-type: none"> Estrategia minimizada <ul style="list-style-type: none"> <u>Escrito (por teclado)</u> <u>Texto de prueba</u>

Tabla 6.2. Situaciones de prueba para el procesamiento del texto.

Gestión de palabras clave	
P006: Búsqueda de palabras clave	
Condiciones de prueba y clases de equivalencia	Situaciones de prueba
<ul style="list-style-type: none"> Filtro búsqueda <ul style="list-style-type: none"> Vacio Coincide con al menos una palabra No coincide con ninguna palabra 	<ul style="list-style-type: none"> Estrategia minimizada <ul style="list-style-type: none"> <u>Vacio</u> <u>Coincide con al menos una palabra</u> <u>No coincide con ninguna palabra</u>
P007: Eliminar palabras clave	
Condiciones de prueba y clases de equivalencia	Situaciones de prueba
<ul style="list-style-type: none"> Número de seleccionadas <ul style="list-style-type: none"> Alguna Todas 	<ul style="list-style-type: none"> Estrategia minimizada <ul style="list-style-type: none"> <u>Alguna eliminada</u> <u>Todas eliminadas</u>
P008: Añadir palabras clave	
Condiciones de prueba y clases de equivalencia	Situaciones de prueba
<ul style="list-style-type: none"> Palabra existente en la lista <ul style="list-style-type: none"> Existe en la lista No existe en la lista Palabra existente en el texto <ul style="list-style-type: none"> Existe en el texto No existe en el texto 	<ul style="list-style-type: none"> Combinación parcial <ul style="list-style-type: none"> <u>Existe en la lista</u> <u>No existe en la lista</u> <ul style="list-style-type: none"> <u>Existe en el texto</u> <u>No existe en el texto</u>
P009: Editar palabras clave	

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

Condiciones de prueba y clases de equivalencia	Situaciones de prueba
<ul style="list-style-type: none"> • Palabra existente en la lista <ul style="list-style-type: none"> ○ Existente en la lista ○ No existe en la lista • Palabra existente en el texto <ul style="list-style-type: none"> ○ Existe en el texto ○ No existe en el texto 	<ul style="list-style-type: none"> • Combinación parcial <ul style="list-style-type: none"> ○ Existe en la lista ○ No existe en la lista <ul style="list-style-type: none"> ▪ Existe en el texto ▪ No existe en el texto
P010: Modificar número de preguntas a generar por palabras clave	
Condiciones de prueba y clases de equivalencia	Situaciones de prueba
<ul style="list-style-type: none"> • Tipo carácter <ul style="list-style-type: none"> ○ Numérico ○ No numérico • Valor carácter numérico <ul style="list-style-type: none"> ○ > 0 ○ $= 0$ ○ < 0 	<ul style="list-style-type: none"> • Combinación parcial <ul style="list-style-type: none"> ○ <u>No numérico</u> ○ Numérico <ul style="list-style-type: none"> ▪ ≥ 0 ▪ $= 0$ ▪ ≤ 0
P011: Seleccionar palabras clave	
Condiciones de prueba y clases de equivalencia	Situaciones de prueba
<ul style="list-style-type: none"> • Número de palabras clave seleccionadas <ul style="list-style-type: none"> ○ Ninguna ○ Al menos una seleccionada ○ Todas 	<ul style="list-style-type: none"> • Estrategia minimizada <ul style="list-style-type: none"> ○ <u>Ninguna seleccionada</u> ○ <u>Al menos una seleccionada</u> ○ <u>Todas</u>

Tabla 6.3. Situaciones de prueba para el procesamiento de palabras clave.

Gestión de preguntas	
P012: Mostrar preguntas	
Condiciones de prueba y clases de equivalencia	Situaciones de prueba



<ul style="list-style-type: none">• Existen preguntas repetidas<ul style="list-style-type: none">○ Si○ No• Preguntas insuficientes para palabra<ul style="list-style-type: none">○ Si○ No	<ul style="list-style-type: none">• Combinación total<ul style="list-style-type: none">○ Existen preguntas repetidas<ul style="list-style-type: none">▪ <u>Hay preguntas insuficientes para palabra</u>▪ <u>No hay preguntas insuficientes para palabra</u>○ No existen preguntas repetidas<ul style="list-style-type: none">▪ <u>Hay preguntas insuficientes para palabra</u>▪ <u>No hay preguntas insuficientes para palabra</u>
--	---

P013: Editar enunciados

Condiciones de prueba y clases de equivalencia	Situaciones de prueba
<ul style="list-style-type: none">• Longitud del nuevo enunciado<ul style="list-style-type: none">○ Mayor○ Menor○ 0	<ul style="list-style-type: none">• Estrategia minimizada<ul style="list-style-type: none">○ <u>Más grande</u>○ <u>Más pequeño</u>○ <u>0</u>

P014: Editar opciones

Condiciones de prueba y clases de equivalencia	Situaciones de prueba
<ul style="list-style-type: none">• Modifica opciones<ul style="list-style-type: none">○ Añade una nueva○ Elimina una existente○ Modifica el texto de una	<ul style="list-style-type: none">• Estrategia minimizada<ul style="list-style-type: none">○ <u>Añade una nueva</u>○ Elimina una existente○ Modifica el texto de una

P015: Añadir preguntas

Condiciones de prueba y clases de equivalencia	Situaciones de prueba
<ul style="list-style-type: none">• Forma de añadir pregunta<ul style="list-style-type: none">○ Duplicar existente○ Añadir nueva	<ul style="list-style-type: none">• Estrategia minimizada<ul style="list-style-type: none">○ <u>Duplicar pregunta existente</u>

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

	<input type="radio"/> Añadir nueva pregunta
P016: Eliminar preguntas	
Condiciones de prueba y clases de equivalencia	Situaciones de prueba
<ul style="list-style-type: none"> • Número de preguntas existentes <ul style="list-style-type: none"> <input type="radio"/> Ninguna <input type="radio"/> Alguna 	<ul style="list-style-type: none"> • Estrategia minimizada <ul style="list-style-type: none"> <input type="radio"/> Ninguna pregunta existente <input type="radio"/> Alguna pregunta existente
P017: Ordenar preguntas	
Condiciones de prueba y clases de equivalencia	Situaciones de prueba
<ul style="list-style-type: none"> • Nueva posición pregunta <ul style="list-style-type: none"> <input type="radio"/> Hacia arriba <input type="radio"/> Hacia abajo <input type="radio"/> Hacia posición no válida 	<ul style="list-style-type: none"> • Estrategia minimizada <ul style="list-style-type: none"> <input type="radio"/> Hacia arriba <input type="radio"/> Hacia abajo <input type="radio"/> Hacia posición no válida
P018: Exportar preguntas	
Condiciones de prueba y clases de equivalencia	Situaciones de prueba
<ul style="list-style-type: none"> • Número de preguntas a exportar <ul style="list-style-type: none"> <input type="radio"/> Ninguna <input type="radio"/> Al menos 1 • Formato de exportación <ul style="list-style-type: none"> <input type="radio"/> Txt <input type="radio"/> Pdf <input type="radio"/> Docx 	<ul style="list-style-type: none"> • Combinación parcial <ul style="list-style-type: none"> <input type="radio"/> Ninguna <input type="radio"/> Al menos 1 <ul style="list-style-type: none"> ▪ Txt ▪ Pdf ▪ Docx

Tabla 6.4. Situaciones de prueba para la gestión de preguntas.

Gestión de cuestionarios	
P019: Generar cuestionario	
Condiciones de prueba y clases de equivalencia	Situaciones de prueba



<ul style="list-style-type: none">• Alguna pregunta modificada<ul style="list-style-type: none"><input type="radio"/> Si<input type="radio"/> No• Número de preguntas<ul style="list-style-type: none"><input type="radio"/> Ninguna<input type="radio"/> Al menos 1	<ul style="list-style-type: none">• Combinación parcial<ul style="list-style-type: none"><input type="radio"/> Al menos 1 pregunta<ul style="list-style-type: none">▪ Alguna pregunta modificada▪ No se modifican preguntas<input type="radio"/> Ninguna pregunta
---	--

P020: Ver estado cuestionarios

Condiciones de prueba y clases de equivalencia	Situaciones de prueba
<ul style="list-style-type: none">• Número de cuestionarios creados<ul style="list-style-type: none"><input type="radio"/> Ninguno<input type="radio"/> Al menos uno• Usuarios en cuestionario<ul style="list-style-type: none"><input type="radio"/> Ninguno<input type="radio"/> Al menos uno• Estados usuarios<ul style="list-style-type: none"><input type="radio"/> Aprobado<input type="radio"/> Suspens<input type="radio"/> En progreso	<ul style="list-style-type: none">• Combinación parcial<ul style="list-style-type: none"><input type="radio"/> Al menos uno<ul style="list-style-type: none">▪ <u>Ningún usuario</u>▪ Al menos un usuario<ul style="list-style-type: none">• <u>Aprobado</u>• <u>Suspens</u>• <u>En</u> <u>progreso</u><input type="radio"/> <u>Ninguno</u>

P021: Cerrar/Abrir cuestionario

Condiciones de prueba y clases de equivalencia	Situaciones de prueba
<ul style="list-style-type: none">• Cambio estado de cuestionario<ul style="list-style-type: none"><input type="radio"/> Abrir<input type="radio"/> Cerrar	<ul style="list-style-type: none">• Estrategia minimizada<ul style="list-style-type: none"><input type="radio"/> Abrir<input type="radio"/> Cerrar

P022: Unirse a cuestionario

Condiciones de prueba y clases de equivalencia	Situaciones de prueba
<ul style="list-style-type: none">• Cuestionario cerrado<ul style="list-style-type: none"><input type="radio"/> Si<input type="radio"/> No• Estado cuestionario usuario<ul style="list-style-type: none"><input type="radio"/> Finalizado<input type="radio"/> En progreso	<ul style="list-style-type: none">• Combinación parcial<ul style="list-style-type: none"><input type="radio"/> <u>No existe test</u><input type="radio"/> Existe test<ul style="list-style-type: none">▪ <u>Cerrado</u>▪ Abierto• <u>Finalizado</u>

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

<ul style="list-style-type: none"> <input type="radio"/> No se ha unido • Código test introducido <ul style="list-style-type: none"> <input type="radio"/> No existente <input type="radio"/> Existe 	<ul style="list-style-type: none"> • <u>En progreso</u> • No unido
P023: Contestar a pregunta	
Condiciones de prueba y clases de equivalencia	Situaciones de prueba
<ul style="list-style-type: none"> • Opción contestada <ul style="list-style-type: none"> <input type="radio"/> Opción válida <input type="radio"/> Limpia la contestación 	<ul style="list-style-type: none"> • Estrategia minimizada <ul style="list-style-type: none"> <input type="radio"/> Opción válida <input type="radio"/> Limpia contestación
P024: Enviar cuestionario	
Condiciones de prueba y clases de equivalencia	Situaciones de prueba
<ul style="list-style-type: none"> • Cuestionario con <i>feedback</i> activado <ul style="list-style-type: none"> <input type="radio"/> Si <input type="radio"/> No • Nota obtenida <ul style="list-style-type: none"> <input type="radio"/> Aprueba <input type="radio"/> Suspende • Cuestionario cerrado <ul style="list-style-type: none"> <input type="radio"/> Si <input type="radio"/> No 	<ul style="list-style-type: none"> • Combinación parcial <ul style="list-style-type: none"> <input type="radio"/> Cerrado <input type="radio"/> Abierto <ul style="list-style-type: none"> ▪ Activado <i>feedback</i> <ul style="list-style-type: none"> • Aprueba • Suspende ▪ No activado <i>feedback</i>

Tabla 6.5. Situaciones de prueba para la gestión de cuestionarios.

6.5.2.2 Pruebas con Usuarios

Condiciones de prueba y clases de equivalencia	Situaciones de prueba
<ul style="list-style-type: none"> • Tipo de usuario <ul style="list-style-type: none"> <input type="radio"/> Conocimiento informático bajo (usuario 3) <ul style="list-style-type: none"> ▪ Creación de cuestionario ▪ Unión y envío de cuestionario <input type="radio"/> Conocimiento informático medio <input type="radio"/> Conocimiento informático alto • Tarea para realizar 	<ul style="list-style-type: none"> • Combinación total <ul style="list-style-type: none"> <input type="radio"/> Conocimiento informático bajo (usuario 3) <ul style="list-style-type: none"> ▪ Creación de cuestionario ▪ Unión y envío de cuestionario <input type="radio"/> Conocimiento informático medio (usuario 2)



<ul style="list-style-type: none">○ Creación de cuestionario○ Unión y envío de cuestionario	<ul style="list-style-type: none">■ Creación de cuestionario■ Unión y envío de cuestionario○ Conocimiento informático alto (usuario 1)<ul style="list-style-type: none">■ Creación de cuestionario■ Unión y envío de cuestionario
--	--

Tabla 6.6. Situaciones de prueba para las pruebas con usuarios.

Estas se combinan totalmente y se obtiene que cada usuario deberá realizar las dos tareas, con lo cual obtenemos unas 6 situaciones de prueba. Estas se ejecutarán completando cuestionarios de evaluación de los siguientes apartados, los cuales se basan en los disponibles en la referencia [26].

6.5.2.2.1 Cuestionario previo

¿Qué edad tienes? -
¿Usa un ordenador frecuentemente? <ol style="list-style-type: none">1. Todos los días2. Varias veces a la semana3. Ocasionalmente4. Nunca o casi nunca
¿Qué tipo de actividades realiza con el ordenador? <ol style="list-style-type: none">1. Es parte de mi trabajo o profesión2. Lo uso básicamente para ocio3. Solo empleo aplicaciones estilo Office4. Únicamente leo el correo y navego ocasionalmente
¿Ha usado alguna vez software como el de esta prueba? <ol style="list-style-type: none">1. Sí, he empleado software similar2. No, aunque si empleo otros programas que me ayudan a realizar tareas similares3. No, nunca
¿Qué busca Vd. Principalmente en un programa?

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

1. Que sea fácil de usar
2. Que sea intuitivo
3. Que sea rápido
4. Que tenga todas las funciones necesarias

Tabla 6.7. Diseño de cuestionario previo.

6.5.2.2.2 Cuestionario posterior

Preguntas Cortas				
Funciones de la Aplicación				
Facilidad de Uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Sabe dónde está dentro de la aplicación?</i>				
<i>¿Existe ayuda para las funciones en caso de que tenga dudas?</i>				
<i>¿Le resulta sencillo el uso de la aplicación?</i>				
Funcionalidad	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Funciona cada tarea como Vd. Espera?</i>				
<i>¿El tiempo de respuesta de la aplicación es muy grande?</i>				
Calidad del Interfaz				
Aspectos gráficos	Muy Adecuado	Adecuado	Poco Adecuado	Nada Adecuado
<i>El tipo y tamaño de letra es</i>				
<i>Los iconos e imágenes usados son</i>				
<i>Los colores empleados son</i>				
Diseño de la Interfaz	Si	No	A veces	
<i>¿Le resulta fácil de usar?</i>				
<i>¿El diseño de las pantallas es claro y atractivo?</i>				
<i>¿Cree que el programa está bien estructurado?</i>				
Observaciones				
Cualquier comentario del usuario				

Tabla 6.8. Cuestionario posterior.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

6.5.2.2.3 Cuestionario del supervisor de la prueba

El supervisor de la prueba vigilará las interacciones del usuario con el sistema y este se encargará de ir anotando todos los aspectos que considere oportunos. Anotará, aspectos en los que se queda atascado, sensaciones del usuario de prueba y dificultades visibles.

Asimismo, en ningún momento prestará ayuda al usuario de prueba. Finalmente anotará si el usuario se rinde y en qué punto lo hace.

6.5.2.3 Pruebas de Adaptabilidad

Como vemos, las resoluciones de pantalla más populares para cada tipo de dispositivo según el sitio web *statcounter* (<https://gs.statcounter.com>) en los últimos 12 meses en Europa son:

- Monitores: 1920x1080
- Tabletas: 768x1024
- Dispositivo móvil: 390x844

Condiciones de prueba y clases de equivalencia	Situaciones de prueba
<ul style="list-style-type: none"> • Tamaño de pantalla <ul style="list-style-type: none"> ○ 390x844 ○ 768x1024 ○ 1920x1080 • Vistas de la aplicación <ul style="list-style-type: none"> ○ Formulario inicio de sesión ○ Formulario registro ○ Formulario de unirse a cuestionario ○ Vista de realización de cuestionario 	<ul style="list-style-type: none"> • Combinación total <ul style="list-style-type: none"> ○ 390x844 <ul style="list-style-type: none"> ▪ Formulario inicio de sesión ▪ Formulario registro ▪ Formulario de unirse a cuestionario ▪ Vista de realización de cuestionario ○ 768x1024 <ul style="list-style-type: none"> ▪ Formulario inicio de sesión ▪ Formulario registro ▪ Formulario de unirse a cuestionario ▪ Vista de realización de cuestionario ○ 1920x1080 <ul style="list-style-type: none"> ▪ Formulario inicio de sesión ▪ Formulario registro ▪ Formulario de unirse a cuestionario ▪ Vista de realización de cuestionario

Tabla 6.9. Situaciones de prueba para pruebas de adaptabilidad.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

7 Implementación del Sistema de Información

7.1 Estándares y normas seguidos

7.1.1 Guías PEP de Python

Dada la importancia de buenas prácticas en la elaboración y desarrollo de código, con el fin de que este sea de la mejor calidad y lo más mantenible posible, se han intentado seguir diversas guías y normas de referencia. Entre estas para la implementación de soluciones en Python se ha optado por seguir las guías PEPs (*Python Enhancement Proposals*) [27] las cuales dictan diferentes aspectos que debe tener un buen código en Python. Por ejemplo, uso de *CamelCase* para clases o separar espacios de funciones con barras bajas.

7.2 Lenguajes de programación

Aquí se enumeran los diferentes lenguajes de programación utilizados en el desarrollo del proyecto.

7.2.1 Python

Este lenguaje de programación, que a diferencia de Java u otros similares, es interpretado, se ha utilizado para la implementación del algoritmo y su posterior estudio.

También se ha utilizado para la implementación del *backend* del sistema. Concretamente, se ha utilizado la versión 3.11.2.

7.2.2 JavaScript

JavaScript es un lenguaje de programación muy utilizado en desarrollo web, este lenguaje está estandarizado en el estándar. Asimismo, JavaScript goza de muchas características ya que es multiparadigma. Un paradigma de programación no deja de ser una clasificación del lenguaje programación. JavaScript permite paradigma orientado a objetos, funcional e imperativo.

Al igual que Python también es interpretado y no compilado como sí lo es Java. Se interpreta del lado del cliente (es decir, en el navegador no en el servidor como sí hace PHP) y se ha utilizado para la programación de la interfaz de usuario interactiva con la librería React.

Asimismo, para la gestión de paquetes se ha usado el popular sistema de gestión de paquetes npm (*Node Packet Manager*) en su versión 8.13.2.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

7.3 Herramientas usadas para el desarrollo

7.3.1 Visual Studio Code

Visual Studio Code es un editor de código altamente personalizable con múltiples opciones y funcionalidades. Es un editor que se ha utilizado para el desarrollo de la parte del *frontend* así como del *backend*. Principalmente se ha optado por este entorno ya que es un muy ligero y versátil en el desarrollo de aplicaciones web.

- **Versión:** 1.76.2
- **Desarrollador:** Microsoft
- **Enlace:** <https://code.visualstudio.com/>

Visual Studio Code también ha servido como administrador de bases de datos con sus extensiones disponibles. Estos permiten manejar y atacar un SGBD al mismo tiempo que se desarrolla el sistema. Para ello se ha utilizado la extensión siguiente para el manejo de bases de datos relacionales SQLite:

- **Enlace:** <https://github.com/yy0931/sqlite3-editor>

7.3.2 Mozilla Firefox

Se ha utilizado para visualizar y monitorizar la consola de depuración durante el desarrollo de la interfaz de usuario de la aplicación.

- **Versión Utilizada:** 113.0.1
- **Desarrollador:** Fundación Mozilla
- **Enlace:** <https://www.mozilla.org/es-ES/firefox/new/>

7.3.3 PyCharm Professional

Utilizado para el desarrollo de código en Python durante el capítulo 2. Es un IDE de JetBrains con licencia de pago pero que se puede utilizar con fines educativos a través de la cuenta de correo de la Universidad de Oviedo.

- **Versión Utilizada:** 2022.3.3
- **Desarrollador:** JetBrains
- **Enlace:** <https://www.jetbrains.com/pycharm>

7.3.4 Global Protect

GlobalProtect es una plataforma de VPN (red privada virtual en español) diseñada para inspeccionar tanto el tráfico entrante como el saliente y proteger el acceso a datos a través

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

de conexiones segura. Usando el portal de la universidad *portalgp.uniovi.es* y las credenciales de la Universidad de Oviedo podemos conectarnos a la red de la universidad.

- **Versión Utilizada:** 6.0.3-38
- **Desarrollador:** Palo Alto Networks
- **Enlace:** <https://www.paloaltonetworks.es/products/globalprotect>

7.3.5 Escritorio Remoto Windows

Utilizado para la conexión a máquinas remotas a través de la red de la universidad usando la plataforma de VPN del apartado 7.3.4.

- **Versión Utilizada:** 10.2.3005.0
- **Desarrollador:** Microsoft Corporation
- **Enlace:** Viene de serie en ordenadores Windows

7.3.6 Git

Git un sistema de control de versiones distribuido que permite gestionar todos los cambios realizados sobre distintos ficheros. En git, al ser distribuido, tenemos repositorios locales y repositorios remotos.

- **Versión Utilizada:** 2.35.3
- **Desarrollador:** Linus Torvalds
- **Enlace:** <https://git-scm.com/downloads>

Normalmente los repositorios remotos se almacenan en servicios web. En el caso de este trabajo se ha utilizado la plataforma GitHub que el mismo tiempo permite múltiples funcionalidades que facilitan la gestión de cambios.

- **Enlace a mi perfil de GitHub:** <https://github.com/uo276406>

7.4 Consideraciones de seguridad

7.4.1 Token de acceso

A la hora de acceder a una cuenta de un usuario, este recibe un token de seguridad. Este token de seguridad está implementado usando el estándar jwt, (*JSON Web Token*). Este se utiliza a la hora de hacer peticiones de la webapp a la restapi, ya que este se envía en la cabecera de la petición de forma que se pueda identificar fácilmente al usuario que realiza la petición. Este token de seguridad tiene la principal ventaja de evitar el almacenamiento de datos del cliente en el servidor ofreciendo un mayor nivel de

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

seguridad. Del mismo modo también permite proteger rutas para que solo puedan ser accedidas por usuarios autenticados en el sistema.

Este token además tiene un tiempo de expiración, una vez este tiempo de expiración termina, el usuario debe volver a iniciar sesión para renovarlo. Esta medida también permite detectar tiempos de inactividad y demás posibilidades que resulten en un problema de seguridad.

```
async getKeywords(text, accessToken) {
  return await fetch(apiendpoint + "/api/v1.0/keywords/find", {
    method: "POST",
    headers: {
      "Content-Type": "application/json",
      "Authorization": "Bearer " + accessToken,
    },
    body: JSON.stringify({
      text_body: text,
    }),
  }).then((response) => response.json());
```

Figura 7.1. Ejemplo de envío de petición con token de acceso.

7.4.2 Almacenamiento de contraseñas

En cuanto al almacenamiento de contraseñas se refiere, el sistema utiliza el algoritmo BCrypt. El sistema previamente al almacenamiento de los datos de los usuarios realiza un encriptado de las contraseñas evitando que estas se almacenen con texto plano.

El algoritmo BCrypt está ampliamente utilizado y es el algoritmo utilizado por defecto en múltiples distribuciones de Linux. Este algoritmo, al igual que muchos otros incorpora para la encriptación tanto un *hash* como un *salt* (estos valores se utilizan en el ámbito de la ciberseguridad como cadenas de caracteres que son entradas del algoritmo y se usan durante la encriptación). En Python, este algoritmo está implementado en la librería Passlib, la cual es de gran popularidad y está ampliamente probada.

Otro aspecto relacionado con las contraseñas es que a la hora de hacer un registro, el sistema fuerza al usuario a introducir una contraseña robusta. Estas condiciones se especifican en el apartado 5.3.1.2.4 sobre requisitos de sistema, y es una medida de seguridad también aplicada en el sistema. Las condiciones son las siguientes:

- Mínimo 6 caracteres.
- Mínimo 1 dígito.
- Mínimo 1 minúscula.
- Mínimo 1 mayúscula.

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

8 Ejecución de las Pruebas

8.1 Pruebas Estáticas

8.1.1 Análisis estático

Estas pruebas se ejecutan de forma automatizada cada vez que se crea una *pull request* (petición de cambio) en el repositorio. Esto nos permitirá revisar el código previamente a la confirmación de los cambios. Permite analizar, aspectos de calidad del código relacionados con malas prácticas, variables no usadas, y demás que pueden acumularse e incrementar la deuda técnica del proyecto.

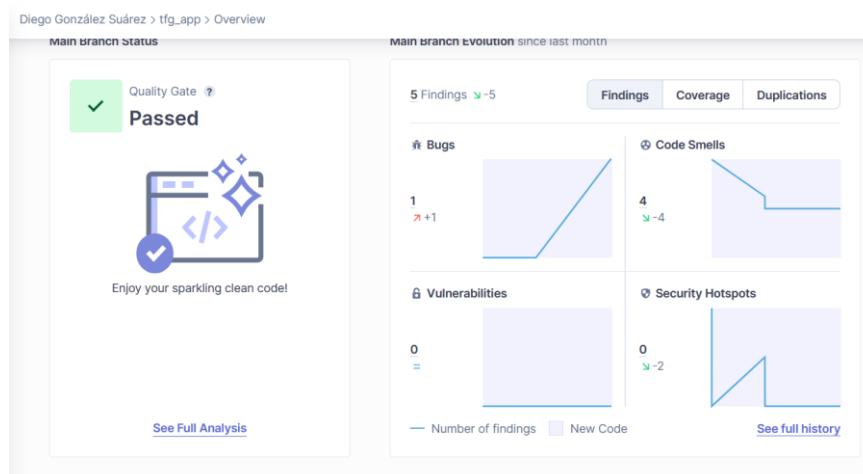


Figura 8.1. Vista del panel de administración de SonarCloud.

Cada vez que se realice el análisis obtendremos una valoración de “*Quality Gate*” que nos informará sobre si se pasa o no el análisis. En caso de que no, nos indicará los aspectos a corregir y cómo corregirlos. También se puede ver información sobre la calidad del código en el inicio del repositorio en GitHub.

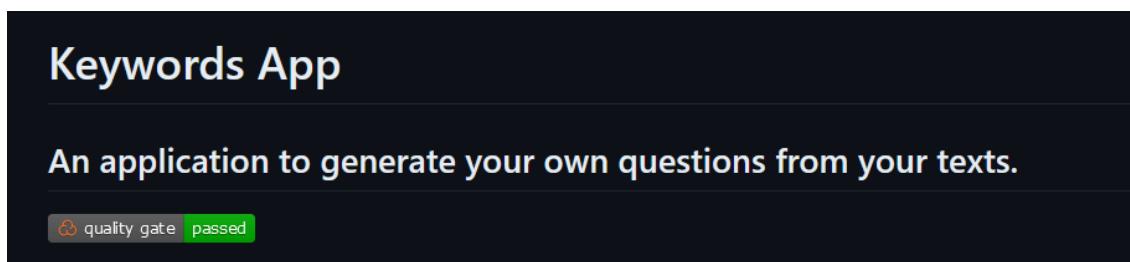


Figura 8.2. *Quality Gate* en el repositorio (https://github.com/uo276406/tfg_app).

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

8.2 Pruebas Unitarias

8.2.1 Restapi

La ejecución de las pruebas unitarias de la restapi se ejecuta utilizando el comando *pytest* en el directorio /restapi. Y se obtiene un resultado tal que así:

```
● (.venv) PS C:\Users\diego\OneDrive\Escritorio\tfg_app\restapi> pytest
=====
test session starts =====
platform win32 -- Python 3.11.2, pytest-7.3.1, pluggy-1.0.0
rootdir: C:\Users\diego\OneDrive\Escritorio\tfg_app\restapi
plugins: anyio-3.6.2
collected 26 items

tests\test_keywordsrouter.py .. [ 7%]
tests\test_main.py . [ 11%]
tests\test_questionsgeneratorrouter.py .. [ 19%]
tests\test_testrouter.py ..... [ 61%]
tests\test_usersrouter.py ..... [100%]

===== warnings summary =====
models\test.py:8
  C:\Users\diego\OneDrive\Escritorio\tfg_app\restapi\models\test.py:8: PytestCollectionWarning: cannot collect test class 'Test' because
it has a __init__ constructor (from: tests\test_testrouter.py)
    class Test(Base):

-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
===== 26 passed, 1 warning in 31.99s =====
○ (.venv) PS C:\Users\diego\OneDrive\Escritorio\tfg_app\restapi> []
```

Figura 8.3. Resultado de ejecución de pruebas unitarias de la restapi.

8.3 Pruebas del Sistema

8.3.1 Pruebas Funcionales (end-to-end)

Para la ejecución de las pruebas funcionales de interfaz es necesario estar en el directorio “/webapp” y ejecutar el comando “npm test”. Este comando detectará todos los ficheros cuya extensión es “.test.js” e irá ejecutando cada test de forma automática. Realmente, el comando “npm test”, se puede ver descrito en el fichero “package.json”, siendo este comando el siguiente:

```
"test": "react-scripts test",
```

Una vez ejecutemos “npm test”, tendremos un panel de opciones como este:

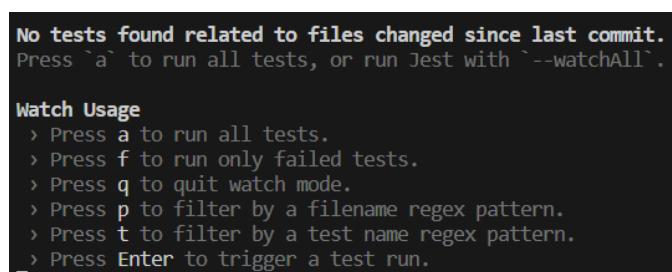


Figura 8.4. Panel de opciones de ejecución de tests.

A continuación hacemos clic en el teclado en la “a” y se ejecutarían las pruebas.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

Además para la ejecución de estas pruebas es necesario utilizar la base de datos de pruebas denominada “testing.db” incluida en el código fuente de los anexos. Esta base de datos contiene filas específicas para cubrir diferentes situaciones de prueba con lo cual es importante especificarla en el fichero “/restapi/.env” con la siguiente línea:

```
DATABASE="sqlite+aiosqlite:///testingdb.db"
```

El resultado de la ejecución se muestra a continuación:

```
Test Suites: 8 passed, 8 total
Tests:      34 passed, 34 total
Snapshots:  0 total
Time:       844.115 s
Ran all test suites.
```

Figura 8.5. Resultado ejecución pruebas funcionales de la webapp.

En ocasiones, la ejecución de las pruebas puede dar como resultado que algunos tests fallen, este se debe a tiempos de espera que se agotan. Esto puede deberse a la carga del ordenador ya que en ocasiones pueden abrirse muchas pestañas del navegador para ejecutar las pruebas.

8.3.2 Pruebas con Usuarios

8.3.2.1 Usuario 1 (Conocimiento alto)

8.3.2.1.1 Cuestionario previo

¿Qué edad tienes?
21
¿Usa un ordenador frecuentemente?
Todos los días
¿Qué tipo de actividades realiza con el ordenador?
Es parte de mi trabajo o profesión
¿Ha usado alguna vez software como el de esta prueba?
No, nunca
¿Qué busca Vd. Principalmente en un programa?
Que sea fácil de usar
Que sea intuitivo
Que sea rápido

Tabla 8.1. Cuestionario previo usuario 1.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

8.3.2.1.2 Cuestionario posterior

Preguntas Cortas				
Funciones de la Aplicación				
Facilidad de Uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
¿Sabe dónde está dentro de la aplicación?	X			
¿Existe ayuda para las funciones en caso de que tenga dudas?		X		
¿Le resulta sencillo el uso de la aplicación?	X			
Funcionalidad	Siempre	Frecuentemente	Ocasionalmente	Nunca
¿Funciona cada tarea como Vd. Espera?	X			
¿El tiempo de respuesta de la aplicación es muy grande?		X		
Calidad del Interfaz				
Aspectos gráficos	Muy Adecuado	Adecuado	Poco Adecuado	Nada Adecuado
El tipo y tamaño de letra es	X			
Los iconos e imágenes usados son	X			
Los colores empleados son	X			
Diseño de la Interfaz	Si	No	A veces	
¿Le resulta fácil de usar?	X			
¿El diseño de las pantallas es claro y atractivo?		X		
¿Cree que el programa está bien estructurado?		X		
Observaciones				
-				

Tabla 8.2. Cuestionario posterior del usuario 1.

8.3.2.1.3 Anotaciones supervisor

- El usuario en algunos casos espera demasiado en la pantalla de carga para la generación de preguntas durante el proceso de creación de exámenes.
- El usuario se confunde con los botones para pasar a la siguiente pregunta, cuando se llega a la última pregunta, el botón de siguiente está activo y se confunde

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

ligeramente ya que no cambia de pregunta. Después se fija en el mapa y ve que está en la última y ya no usa el botón.

8.3.2.2 Usuario 2 (Conocimiento medio)

8.3.2.2.1 Cuestionario previo

¿Qué edad tienes?
29
¿Usa un ordenador frecuentemente?
Todos los días
¿Qué tipo de actividades realiza con el ordenador?
Varias veces a la semana
¿Ha usado alguna vez software como el de esta prueba?
No, aunque si empleo otros programas que me ayudan a realizar tareas similares
¿Qué busca Vd. Principalmente en un programa?
Que sea intuitivo
Que sea rápido
Que tenga todas las funciones necesarias

Tabla 8.3. Cuestionario previo del usuario 2.

8.3.2.2.2 Cuestionario posterior

Preguntas Cortas				
Funciones de la Aplicación				
Facilidad de Uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Sabe dónde está dentro de la aplicación?</i>		X		
<i>¿Existe ayuda para las funciones en caso de que tenga dudas?</i>	X			
<i>¿Le resulta sencillo el uso de la aplicación?</i>	X			
Funcionalidad	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Funciona cada tarea como Vd. Espera?</i>	X			

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

<i>¿El tiempo de respuesta de la aplicación es muy grande?</i>	X
Calidad del Interfaz	
Aspectos gráficos	Muy Adecuado Adecuado Poco Adecuado Nada Adecuado
<i>El tipo y tamaño de letra es</i>	X
<i>Los iconos e imágenes usados son</i>	X
<i>Los colores empleados son</i>	X
Diseño de la Interfaz	Si No A veces
<i>¿Le resulta fácil de usar?</i>	X
<i>¿El diseño de las pantallas es claro y atractivo?</i>	X
<i>¿Cree que el programa está bien estructurado?</i>	X
Observaciones	
La aplicación creada es de gran utilidad para favorecer la tarea de un docente y ayudarle a generar test para evaluar de la materia a los alumnos.	

Tabla 8.4. Cuestionario posterior del usuario 2.

8.3.2.2.3 Anotaciones supervisor

- Hace clic en comenzar y tarda en darse cuenta de que tiene que hacer el registro.
- Le cuesta diferenciar ligeramente las palabras seleccionadas de las no seleccionadas. Echa en falta que se marque más la diferencia de color.
- No utiliza el mapa para saltar preguntas y opta por pasar de una en una.

8.3.2.3 Usuario 3 (Conocimiento bajo)

8.3.2.3.1 Cuestionario previo

¿Qué edad tienes?
56
¿Usa un ordenador frecuentemente?
Nunca o casi nunca
¿Qué tipo de actividades realiza con el ordenador?
Únicamente leo el correo y navego ocasionalmente
¿Ha usado alguna vez software como el de esta prueba?
No, nunca

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

¿Qué busca Vd. Principalmente en un programa?

Que sea fácil de usar

Tabla 8.5. Cuestionario previo del usuario 3.

8.3.2.3.2 Cuestionario posterior

Preguntas Cortas				
Funciones de la Aplicación				
Facilidad de Uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Sabe dónde está dentro de la aplicación?</i>		X		
<i>¿Existe ayuda para las funciones en caso de que tenga dudas?</i>	X			
<i>¿Le resulta sencillo el uso de la aplicación?</i>		X		
Funcionalidad				
Aspectos gráficos	Muy Adecuado	Adecuado	Poco Adecuado	Nada Adecuado
<i>El tipo y tamaño de letra es</i>	X			
<i>Los iconos e imágenes usados son</i>		X		
<i>Los colores empleados son</i>		X		
Calidad del Interfaz				
Diseño de la Interfaz	Si	No	A veces	
<i>¿Le resulta fácil de usar?</i>			X	
<i>¿El diseño de las pantallas es claro y atractivo?</i>		X		
<i>¿Cree que el programa está bien estructurado?</i>	X			
Observaciones				
-				

Tabla 8.6. Cuestionario posterior del usuario 3.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

8.3.2.3.3 *Anotaciones supervisor*

- Consigue procesar el texto correctamente pero le cuesta diferenciar las palabras seleccionadas de las que no lo están.
- A la hora de contestar preguntas, decide borrar una respuesta pero no ve el botón para limpiar la respuesta.
- Tampoco utiliza el salto de preguntas, pasa varias de una en una.

8.3.2.4 Conclusiones y posibles cambios

Como conclusiones, no se aprecian excesivos problemas por parte de los usuarios ya que consiguen realizar las tareas de forma satisfactoria. Es cierto, que algunos detalles no son capaces de verlos, pero es lógico y asumible ya que es la primera vez que se enfrentan a una aplicación de este estilo.

La primera tarea es en la que más problemas se aprecian, consiguen registrarse e iniciar sesión sin problemas pero les cuesta ligeramente entender el concepto de tener que seleccionar las palabras clave. Además les cuesta distinguir aquellas que están seleccionadas y algunas funcionalidades que se han implementado. Echan en falta algunas ayudas como por ejemplo *tooltips* que mejoren su experiencia de usuario.

Por lo general, los usuarios quedan satisfechos con el uso de la aplicación, entienden el funcionamiento general y son capaces de llevar a cabo las labores que se les han sido asignadas.

El 100% de los usuarios consideró que estaba bien estructurada la aplicación. Además otras estadísticas interesantes, es que el 66% de los usuarios consideraron que los colores eran al menos adecuados. Solo uno de los usuarios los consideró que ocasionalmente eran adecuados. Del mismo modo ningún usuario los consideró no adecuados. De la misma forma para el tamaño y tipo de letra, el 66% de los usuarios lo consideró muy adecuado. En el gráfico siguiente se puede ver en detalle:

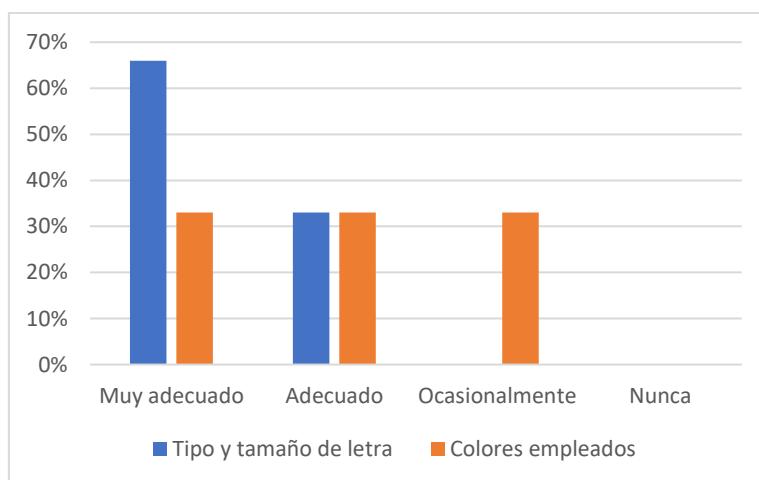


Figura 8.6. Gráfico con encuestas sobre colores empleados y tamaño y tipo de letra.

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

Los cambios que se implementarían, basándonos en los datos obtenidos serían los siguientes:

- Añadir mayor contraste a las palabras seleccionadas.
- Añadir ayudas a la hora de dar más información en la realización de ciertas actividades. Un ejemplo es el número de preguntas a generar por cada palabra.
- Mejora de la navegabilidad por las preguntas de los cuestionarios, ya que muchos usuarios optan por pasar de una en una y perder más tiempo, en lugar de saltar navegando por el mapa de navegación.
- Desactivación de los botones de “Siguiente” y “Anterior”, en la última y primera pregunta de los cuestionarios respectivamente.
- Mejora en la selección de palabras clave, permite hacer clic en toda la palabra.
- Modificación del botón de “Enviar cuestionario” disponible desde todas las preguntas.
- Aumentado tamaño de fuente en las opciones durante la realización de los cuestionarios.

8.3.3 Pruebas de Adaptabilidad

Para llevar a cabo la ejecución de las pruebas se ha utilizado el sitio web <http://responsivetesttool.com> que permite la vista en múltiples resoluciones del sistema.

8.3.3.1 Dispositivo móvil 390x844

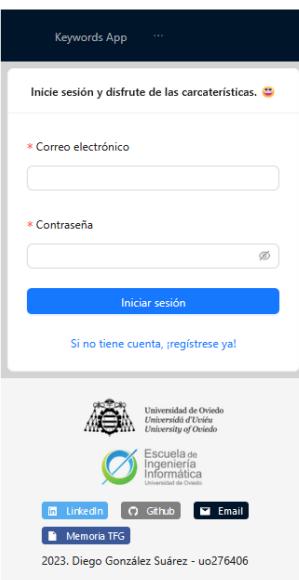


Figura 8.7. Formulario de inicio de sesión adaptable a dispositivo móvil.

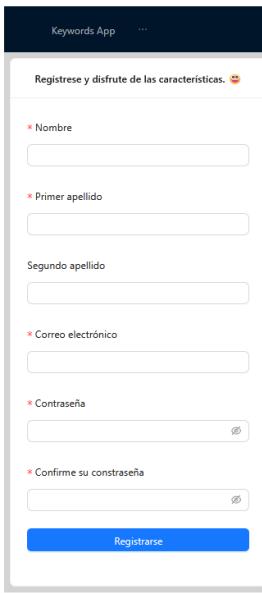


Figura 8.8. Formulario de registro adaptable a dispositivo móvil.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

Keywords App ...

Únase a un test para responder a las pre...

* Id del test

* Id alumno

Unirse a test

Universidad de Oviedo
Universitat d'Uviéu
University of Oviedo

Escuela de
Ingeniería
Informática
Universidad de Oviedo

LinkedIn Github Email
Memoria TFG

2023. Diego González Suárez - uo276406

Keywords App ...

Ha entrado en el test como UO276406

Resumen En progreso

The Americas, Europe, and _____ Before 1492

INTRODUCTION

Globalization, the ever-increasing interconnectedness, is not a new phenomenon, but it accelerated when western Europeans discovered the riches of the East

south
 africa
 lanka
 kenya

< Pregunta anterior > Siguiente pregunta

1 2 3 4

> Enviar test

Figura 8.9. Formulario de unirse a test adaptable a dispositivo móvil.

Figura 8.10. Vista de realización de cuestionarios adaptable a dispositivo móvil.

8.3.3.2 Tableta 768x1024

Keywords App ...

Inicie sesión y disfrute de las características. 😊

* Correo electrónico:

* Contraseña:

Iniciar sesión

Si no tiene cuenta, ¡regístrate ya!

Universidad de Oviedo
Universitat d'Uviéu
University of Oviedo

Escuela de
Ingeniería
Informática
Universidad de Oviedo

LinkedIn Github Email Memoria TFG

2023. Diego González Suárez - uo276406

Figura 8.11. Formulario inicio de sesión adaptable tableta.



Keywords App Unirse a test Documentación API Idioma ...

Registrese y disfrute de las características. 😊

* Nombre:

* Primer apellido:

Segundo apellido:

* Correo electrónico:

* Contraseña: ⚡

* Confirme su contraseña: ⚡

Registrarse

Universidad de Oviedo
Universitat d'Uviéu
University of Oviedo

Escuela de
Ingeniería
Informática
Universidad de Oviedo

[LinkedIn](#) [Github](#) [Email](#) [Memoria TFG](#)

2023. Diego González Suárez - uo276406

Figura 8.12. Formulario de registro adaptable a tableta.

Keywords App Unirse a test Documentación API Idioma ...

Únase a un test para responder a las preguntas. 😊

* Id del test:

* Id alumno:

Unirse a test

Universidad de Oviedo
Universitat d'Uviéu
University of Oviedo

Escuela de
Ingeniería
Informática
Universidad de Oviedo

[LinkedIn](#) [Github](#) [Email](#) [Memoria TFG](#)

2023. Diego González Suárez - uo276406

Figura 8.13. Formulario de unirse a test adaptable a tableta.



Figura 8.14. Vista de realización de cuestionarios adaptable a tableta.

8.3.3.3 Monitor 1920x1080

Figura 8.15. Formulario de inicio de sesión adaptable a monitor.



Keywords App

Unirse a test Documentación API Idioma Iniciar sesión Registrarse

Registrese y disfrute de las características. 😊

* Nombre:

* Primer apellido:

Segundo apellido:

* Correo electrónico:

* Contraseña: Mostrar

* Confirme su contraseña: Mostrar

Registrarse

Universidad de Oviedo
Escuela de Ingeniería Informática
LinkedIn GitHub Email Memoria TFG
2023. Diego González Suárez - uo276406

Figura 8.16. Formulario de registro de usuarios adaptable a monitor.

Keywords App

Unirse a test Documentación API Idioma Iniciar sesión Registrarse

Únase a un test para responder a las preguntas. 😊

* Id del test:

* Id alumno:

Unirse a test

Universidad de Oviedo
Escuela de Ingeniería Informática
LinkedIn GitHub Email Memoria TFG
2023. Diego González Suárez - uo276406

Figura 8.17. Formulario de unirse a test adaptable a monitor.

Keywords App

Unirse a test Documentación API Idioma Iniciar sesión Registrarse

Ha entrado en el test como UsuarioAccesibilidad Resumen • En progreso

But when the Silk Road, the long _____ from China to the Mediterranean, became costlier and
① more dangerous to travel, Europeans searched for a more efficient and inexpensive trade route over
water, initiating the development of what we now call the Atlantic World

terminus
 overland trading route
 rail
 road

< Pregunta anterior > Siguiente pregunta

Universidad de Oviedo
Escuela de Ingeniería Informática
LinkedIn GitHub Email Memoria TFG
2023. Diego González Suárez - uo276406

Figura 8.18. Vista de realización de cuestionarios adaptable a monitor.

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

9 Manuales de Usuario

9.1 Manual para el desarrollador

9.1.1 Descarga del código fuente

Para descargar el código fuente tenemos varias opciones:

- **Descargar Git (ver apartado 7.3.6) y ejecutar el comando:**

```
git clone https://github.com/uo276406/tfg_app.git
```

- **Descargar el fichero comprimido del proyecto desde GitHub (https://github.com/uo276406/tfg_app)**

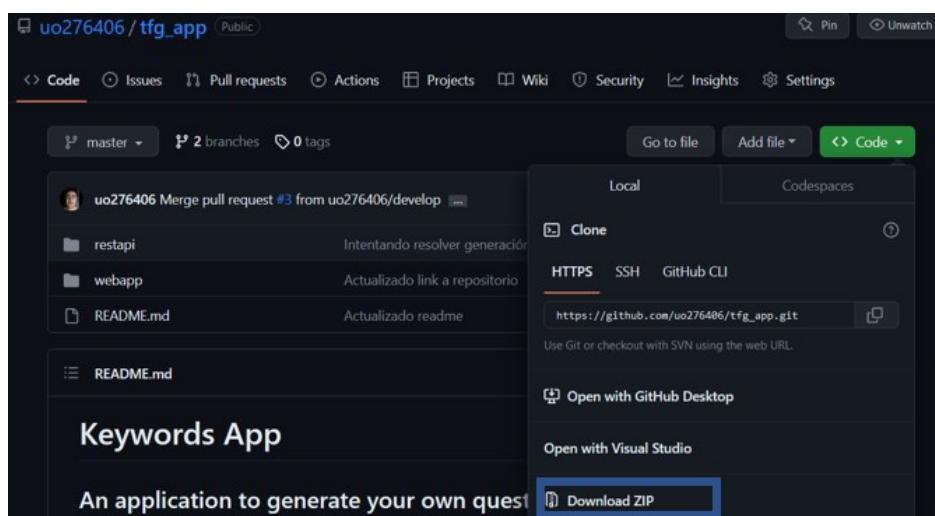


Figura 9.1. Descarga proyecto comprimido de GitHub.

- **Descargar el fichero comprimido de la última versión del repositorio**

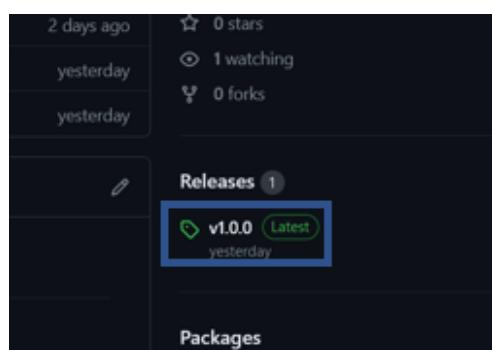


Figura 9.2. Descarga de la última *release* del repositorio.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

- **Descargar el código adjunto en los anexos del trabajo:** En este caso el paso siguiente sobre la creación de ficheros adicionales no sería necesario ya que ya irían incluidos en el código fuente.

9.1.2 Creación de ficheros adicionales

Del mismo modo también son necesarios algunos ficheros extra. Principalmente, estos ficheros están relacionados con variables del entorno (*Environment variables*) que necesita el sistema para funcionar. Estos ficheros serían los siguientes:

Añadir un fichero denominado ".env" en el directorio "/restapi" con el contenido:

```
DATABASE="sqlite+aiosqlite:///development.db"
SECRET="SECRET_KEY"
SECONDS=7200
```

También es necesario crear un fichero denominado "development.db" en el directorio restapi (Este fichero sería la base de datos referenciada en la primera línea del fichero ".env" de la restapi). Este fichero se puede leer con diferentes programas y extensiones de Visual Studio Code para ver la estructura de la base de datos.

Añadir otro igual denominado ".env" al directorio "/webapp" con el contenido:

```
REACT_APP_API_URL=http://localhost:8000
REACT_APP_WEBAPP_URL=http://localhost:3000
REACT_APP_REDOC=http://localhost:8000/redoc
REACT_APP_SWAGGER=http://localhost:8000/docs
```

9.1.3 Instalación de la REST API

El primer paso para instalar la REST API o *backend* de la aplicación es tener instalado Python (si no lo tenemos ya). En el apartado 7.2.1 se describen algunas características. En este proyecto se ha utilizado la versión 3.11.2. Para instalarlo no tenemos que hacer nada más que acceder al sitio web de descargas, <https://www.python.org/downloads/>.

Ahí podremos descargar versiones específicas, así como la última disponible. Para acceder a versiones concretas debemos ver el apartado “*Looking for a specific release?*”:

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

3.7 security 2018-06-27 2023-06-27 PEP 537

Looking for a specific release?

Python releases by version number:

Release version	Release date	Click for more
Python 3.10.11	April 5, 2023	Download Release Notes
Python 3.11.3	April 5, 2023	Download Release Notes
Python 3.10.10	Feb. 8, 2023	Download Release Notes
Python 3.11.2	Feb. 8, 2023	Download Release Notes
Python 3.11.1	Dec. 6, 2022	Download Release Notes
Python 3.10.9	Dec. 6, 2022	Download Release Notes
Python 3.9.16	Dec. 6, 2022	Download Release Notes

[View older releases](#)

Figura 9.3. Descarga de versión concreta de Python.

Una vez hacemos clic en “download”, accedemos a la página de descargas y vamos al final del sitio, descargaremos un ejecutable en función de las características de nuestro equipo:

Files						
Version	Operating System	Description	MD5 Sum	File Size	GPG	Sigstore
Gzipped source tarball	Source release		f6b5226ccba5ae1ca9376aab0bf673	26437858	SIG	CRT SIG
XZ compressed source tarball	Source release		a957cffb58a89303b62124896881950b	19893284	SIG	CRT SIG
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later	e038c3d5cee8c5210735a764d3f36f5a	42835777	SIG	CRT SIG
Windows embeddable package (32-bit)	Windows		64853e569d7cb0d154779300ff9c9b6	9574852	SIG	CRT SIG
Windows embeddable package (64-bit)	Windows		ae7de44ecbe2d3a37dbde3ce669d31b3	10560465	SIG	CRT SIG
Windows embeddable package (ARM64)	Windows		747090b80a52e8bbcb5cb65f78fee575	9780864	SIG	CRT SIG
Windows installer (32-bit)	Windows		2123016702bbb45688baedc3695852f4	24155760	SIG	CRT SIG
Windows installer (64-bit)	Windows	Recommended	4331ca54d9eacdbe6e97d6ea63526e57	25325400	SIG	CRT SIG
Windows installer (ARM64)	Windows	Experimental	040ab03501a65cc26bd340323bb1972e	24451768	SIG	CRT SIG

Figura 9.4. Versión de ejecutable de Python a descargar.

Una vez se haya completado la descarga hacemos doble clic y seguimos los pasos del instalador. Esto nos instalará en el equipo la versión de Python sin más, esto es, sin ningún paquete adicional ni nada por el estilo. Sin embargo, la REST API, necesita de diversos módulos para su funcionamiento. Por ejemplo, de módulos para crear un servidor web como uvicorn u otros como numpy para cuestiones matemáticas. Estos módulos vienen todos especificados en el fichero /restapi/requirements.txt.

Para instalar todos estos módulos usaremos el gestor de paquetes de Python que nos permite gestionar las dependencias. Para ello abriremos un terminal en el directorio /restapi y usaremos el comando siguiente:

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

Esperamos a que se instale y con esto ya tendríamos instalados todos los paquetes necesarios y podríamos correr la REST API con la siguiente orden que lanzará la restapi en el servidor web:

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

```
uvicorn.exe main:app --reload
```

Este servidor web escuchará peticiones en la dirección <http://localhost:8000> y se puede consultar la documentación de la API REST en las direcciones <http://localhost:8000/docs> y <http://localhost:8000/redoc>.

9.1.4 Instalación de la Webapp

La webapp está desarrollada usando JavaScript con la librería React. Asimismo, tanto React como las demás librerías se han descargado usando el gestor de paquetes NPM (*Node Package Manager*). Para ello previamente debemos tener instalado en nuestro equipo Node.js. Node.js es un entorno de desarrollo en tiempo de ejecución para JavaScript.

En este proyecto se ha utilizado la versión 18.12.1 de Node y la versión 8.13.2 de npm. Ambas se pueden instalar en el siguiente enlace <https://nodejs.org/es/download> ya que vienen juntas.



Figura 9.5. Página de descarga de Node.js junto con npm.

Una vez descargado el instalador ya podríamos seguir los pasos por defecto y completar el proceso. Una vez terminado, podríamos ejecutar los comandos siguientes para comprobar que se ha instalado correctamente.

```
node -v
npm -v
```

Terminada la instalación, se deben instalar paquetes necesarios para el funcionamiento del sistema (similar al caso anterior con la restapi cuando se instalaron los paquetes necesarios). En este caso los paquetes se especifican en el fichero */webapp/package.json*. Para instalarlos ejecutamos dentro del directorio */webapp* lo siguiente:

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

```
npm install
```

Esperamos a que se terminan de instalar todos y podemos ejecutar la aplicación:

```
npm start
```

Aunque las ampliaciones son infinitas, a continuación se comentan algunas posibles modificaciones que pueden ser de utilidad para futuros desarrolladores.

9.1.5 Añadir nuevos tipos de preguntas

Actualmente, la aplicación solo soporta la creación de preguntas del tipo “rellenar con la palabra que falta”. Sin embargo, es posible que en un futuro se quieran añadir nuevos formatos de pregunta para ello habría que seguir los siguientes pasos:

1. Añadir un nuevo generador de preguntas en el directorio `/restapi/questiongenerator`:

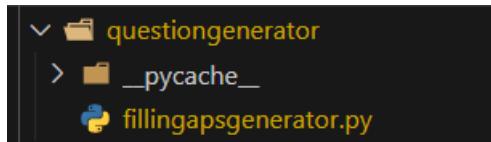


Figura 9.6. Ubicación generadores de preguntas.

2. Este directorio contiene un generador de un tipo de preguntas concretas y contiene una clase con un método llamado “generate_questions” que recibe el texto completo y las palabras clave del texto. Añadir un nuevo tipo de pregunta significaría añadir una nueva clase que reescriba dicho método.

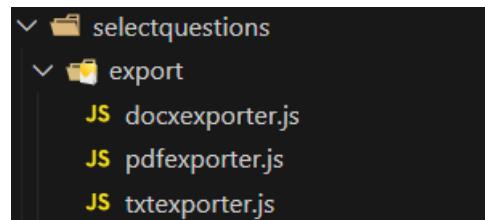
```
def generate_questions(self, text, keywords_selected):
```

Figura 9.7. Cabecera del método de generar preguntas.

9.1.6 Añadir nuevos formatos de exportación de preguntas

Por ahora la aplicación permite la exportación de preguntas en tres formatos que son “.docx”, “.txt (AIKEN)”, “.pdf”. Es posible que surjan nuevos y se deseen añadir otros tipos de exportación para ello se pueden seguir los pasos siguientes:

1. En el directorio “/webapp/src/components/selectquestions/export” se encuentran los diferentes tipos de exportadores.



	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

Figura 9.8. Ubicación de exportadores de preguntas.

- Cada tipo de exportador conoce cómo exportar a cada tipo de fichero. Cada uno contiene una clase con un método denominado “export” que recibe la lista de preguntas. Añadir un nuevo formato significaría añadir una nueva clase con dicho método “export”.

```
class TxtExporter {
    export(questions) {
```

Figura 9.9. Cabecera clase y método de exportación de preguntas.

- Para añadir un nuevo botón para la exportación, habría que acudir al fichero denominado “SelectQuestionsForm.jsx” y crear un nuevo botón con el siguiente esquema, donde al evento “onClick”, se pasa una llamada al método “exportTo” con el exportador concreto a utilizar:

```
<Button
  style={buttonDocxStyle}
  icon={<FileTextOutlined />}
  onClick={() => exportTo(new DocxExporter())}
  disabled={questions.length === 0}
>
  docx
</Button>
```

Figura 9.10. Esquema botones de exportación.

9.1.7 Añadir nuevos idiomas

Aunque ahora está solo internacionalizada para inglés y español puede ser interesante añadir nuevos idiomas. Para ello es sencillo, deberemos acudir al directorio “/public/i18n” y crear un nuevo fichero “.json” con el código del idioma a utilizar:



Figura 9.11. Directorio donde se ubican los ficheros de internacionalización.

Habrá que crear, un fichero con los nuevos valores del idioma en cuestión para cada propiedad. Una vez se tenga el idioma, podemos añadir la opción al menú de navegación desde el fichero “NavBarApp.jsx”.

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

9.2 Manual de usuario

9.2.1 Acceso al sistema

Hay dos opciones para acceder, la primera es de forma local ejecutando el sistema en nuestra propia máquina. Para ello es necesario seguir los pasos de instalación del manual del desarrollador en el apartado 9.1.

Otra opción es, acceder al sistema (restapi y webapp) desplegado en la red interna de la Universidad de Oviedo. Concretamente en una máquina Windows Server 2019 cuya dirección es 156.35.95.118. Para ello es necesario estar conectado a través de una VPN o simplemente estando conectado a la red wifi de la universidad.

Asimismo, tanto la webapp como la restapi, se encuentran desplegadas en diferentes puertos. Las respectivas direcciones son las siguientes:

- **Webapp:** <http://156.35.95.118:3000/> (**Recomendable**)
- **Restapi:** <http://156.35.95.118:8000/>

9.2.2 Cambio de idioma

La aplicación está en inglés y en español, y esto se puede modificar desde el menú de navegación:

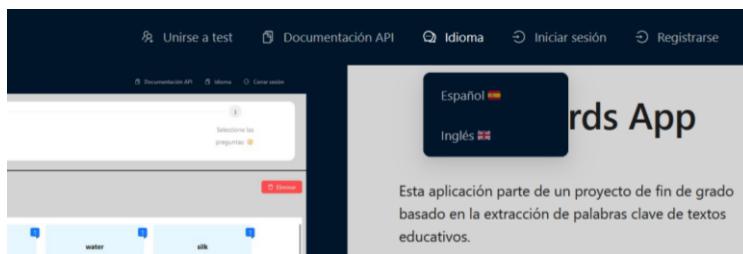
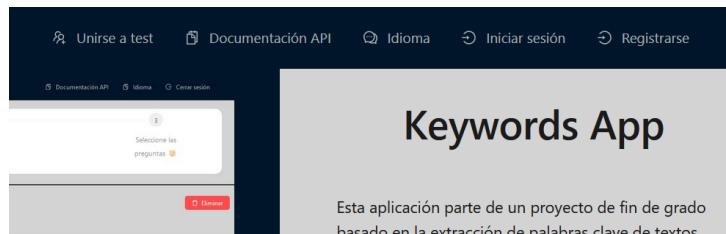


Figura 9.12. Cambio de idioma.

9.2.3 Autenticación de usuarios

La aplicación permite tanto el registro como el inicio de sesión de usuarios, ambas opciones están disponibles en el panel de navegación ubicado en la parte superior derecha.



	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

Figura 9.13. Acceso a registro e inicio de sesión.

El inicio de sesión inicio de sesión solicita tanto el email como la contraseña, avisando en caso de que estos sean incorrectos, el formulario es el siguiente y se encuentra en la ruta “/login”:

Inicie sesión para disfrutar de todas las caraterísticas. 😊

* Correo electrónico:

* Contraseña:

Iniciar sesión

[Si no tiene cuenta, ¡regístrese ya!](#)

Figura 9.14. Formulario de inicio de sesión.

El registro requiere de más datos los cuales se ven en el siguiente formulario, en la ruta “/signin”:

Registrese, si no tiene cuenta, para disfrutar de todas las características. 😊

* Nombre:

* Primer apellido:

Segundo apellido:

* Correo electrónico:

* Contraseña:

* Confirme su contraseña:

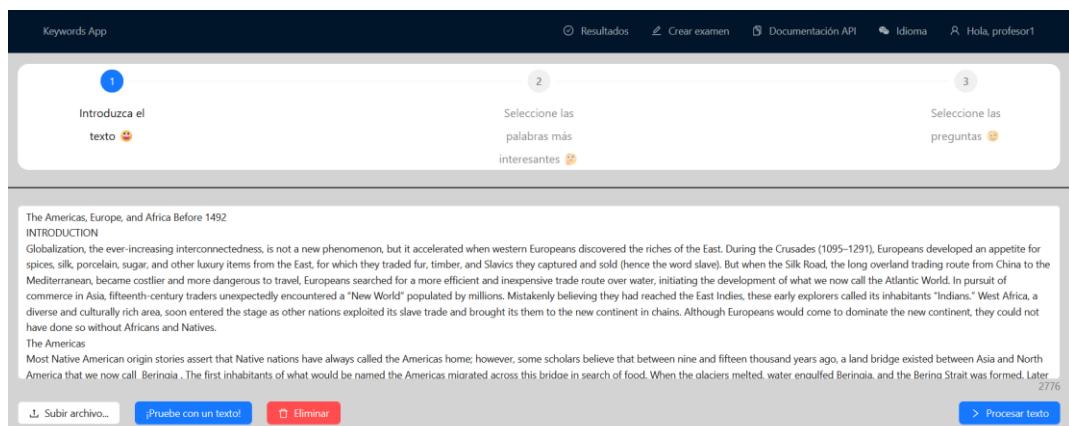
Registrar

Figura 9.15. Formulario de registro de usuarios.

Una vez se inicia sesión o se registra un usuario el sistema redirige a la vista principal de la aplicación. Además, las opciones del menú de navegación variarán, ya que se habilitará nuevas opciones disponibles solo para usuarios autenticados.

9.2.4 Procesamiento de textos

Los usuarios autenticados pueden procesar textos en la ruta “/process”, a la cual se puede acceder desde el menú de navegación en la opción “Crear examen”, así como desde el botón comenzar de la página principal.



The screenshot shows the 'Keywords App' interface. At the top, there's a navigation bar with links to 'Resultados', 'Crear examen', 'Documentación API', 'Idioma', and 'Hola, profesor1'. Below the navigation, three numbered steps are displayed:

1. Introduzca el texto 😊
2. Seleccione las palabras más interesantes 😊
3. Seleccione las preguntas 😊

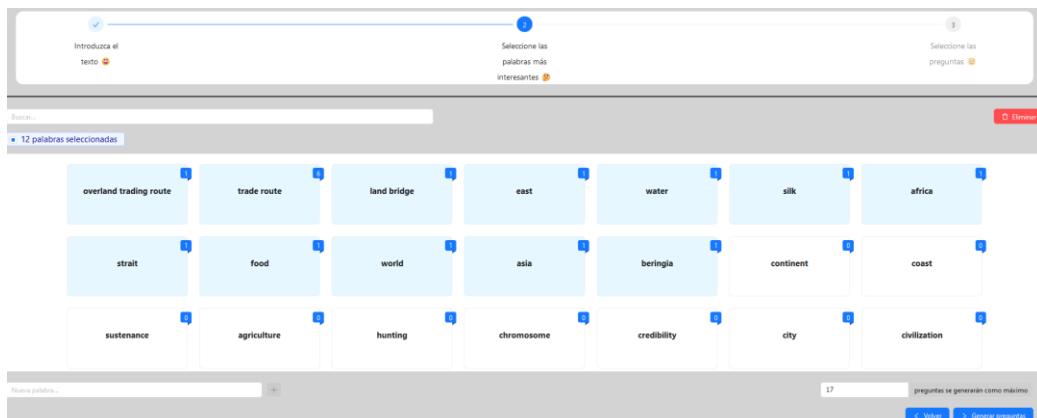
The main content area contains a historical text titled 'The Americas, Europe, and Africa Before 1492' from 'INTRODUCTION'. The text discusses the Silk Road and the discovery of the Americas. Below the text, there are buttons for 'Subir archivo...', '¡Pruebe con un texto!', 'Eliminar', and 'Procesar texto'. A file size indicator '2776' is also visible.

Figura 9.16. Vista de procesamiento de texto.

Se puede probar con un texto de ejemplo, subiendo un fichero de texto y una vez lo tengamos listo podemos pasar a la fase de procesamiento de texto. Se puede visualizar el paso en el que nos encontramos en la parte superior, justo debajo del menú de navegación.

9.2.5 Gestión de palabras clave

Una vez se procesa el texto, el sistema cargará mientras busca las palabras clave, las cuales se mostrarán de la siguiente forma:



The screenshot shows the 'Keywords App' interface after processing the text. It displays a grid of selected keywords with their counts and options to edit or delete them. The keywords listed are:

overland trading route	trade route	land bridge	east	water	silk	africa
strait	food	world	asia	beringia	continent	coast
sustenance	agriculture	hunting	chromosome	credibility	city	civilization

At the bottom, there are buttons for 'Nuevo palabra...', '+', '17', 'preguntas se generarán como máximo', '<> Volver', and '> Generar preguntas'.

Figura 9.17. Selección de palabras clave.

Por defecto marcará el 30% de las palabras encontradas. Asimismo se pueden seleccionar todas y se pueden buscar palabras. Ambas opciones están la parte superior izquierda. Por otro lado también se pueden añadir nuevas palabras desde la parte inferior izquierda.

Para cada palabra se puede modificar el texto haciendo doble clic y también se puede editar el número de preguntas a generar por cada palabra.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

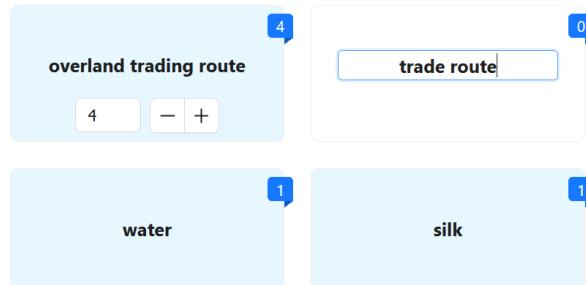


Figura 9.18. Edición de palabras y modificación del número de preguntas a generar.

9.2.6 Gestión de preguntas

Tras haber seleccionado las palabras clave, ahora se pueden seleccionar las preguntas que queramos.

Figura 9.19. Vista de preguntas propuestas.

En la parte inferior también tenemos diferentes opciones de exportación de las preguntas. Estas se pueden hacer a diferentes formatos.

Las preguntas, se pueden editar, tanto el enunciado como las opciones. Para el enunciado se puede editar haciendo clic en el lápiz al final y para las opciones con doble clic, asimismo, también se pueden añadir y eliminar opciones.

Se pueden añadir nuevas preguntas haciendo clic en el botón de “Añadir nueva pregunta...” o duplicando las existentes. Cada pregunta también se puede eliminar haciendo clic en el cubo de la parte derecha de cada pregunta.



Figura 9.20. Añadir preguntas.

También existe un sistema de *drag and drop* para ordenar las preguntas:

Figura 9.21. Sistema de *drag and drop* para ordenar preguntas.

9.2.7 Gestión de cuestionarios

9.2.7.1 Creación de cuestionarios

Los usuarios una vez decidan las mejores preguntas, tendrán la opción de crear sus propios cuestionarios, esta opción se encuentra en la parte inferior derecha de la vista de gestión de preguntas y solo estará disponible si hay al menos una pregunta. Asimismo, estos cuestionarios se pueden configurar utilizando las opciones especificadas con checkboxes para mostrar *feedback* o activar el salto de preguntas. Ambas se encuentran justo encima del botón de generar preguntas.

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

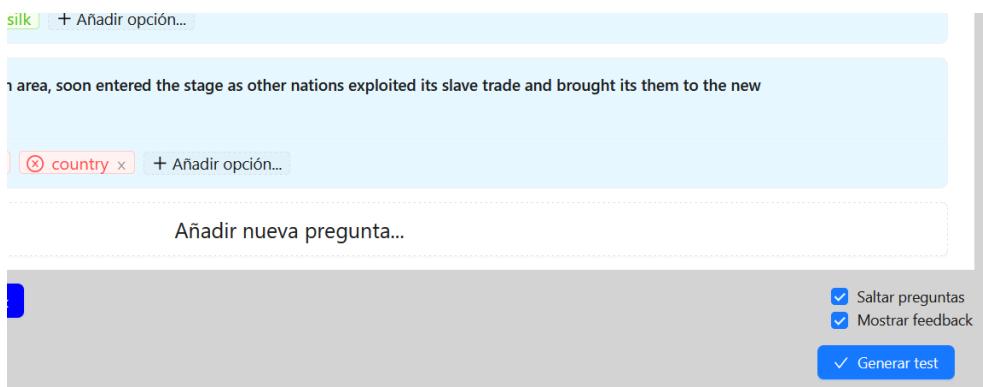


Figura 9.22. Botón de generación de cuestionarios.

Una vez se genera el cuestionario, la aplicación genera un código QR así como el id y el enlace del cuestionario que servirá para unirse.

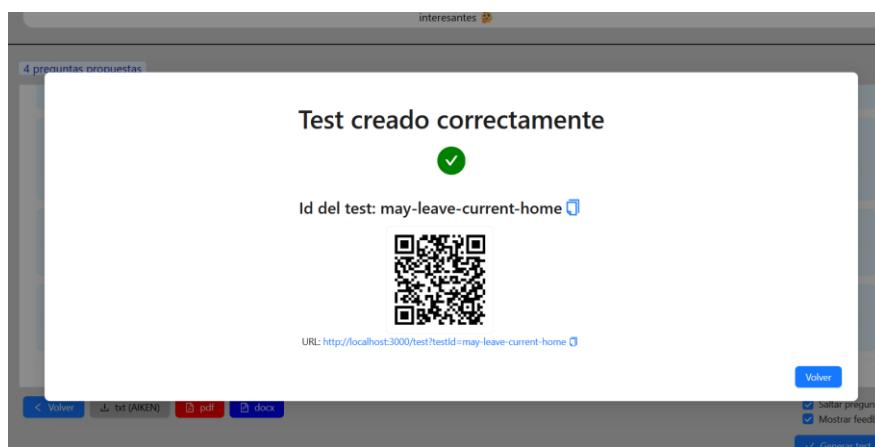


Figura 9.23. Cuestionario o test creado correctamente.

9.2.7.2 Gestión de resultados

Aquellos usuarios que hayan creado y publicado sus cuestionarios podrán visualizar los resultados de aquellas personas que los hagan. Esta vista está disponible desde el menú de navegación para los usuarios autenticados en la opción “Resultados”, en la ruta “/results”.

Desde esta vista podrá ver en tiempo real los usuarios que están haciendo el cuestionario así como las preguntas que van contestando.



Figura 9.24. Vista de resultados de cuestionarios.

Desde esta vista cada cuestionario puede ser abierto o cerrado (para que entren o no más usuarios) así como también se puede actualizar el estado. Los profesores, además pueden ver sus cuestionarios y visualizar el código QR de acceso al cuestionario.

9.2.7.3 Unirse a cuestionarios

Los usuarios anónimos, que serían presumiblemente los alumnos, serían los que se unirían a los cuestionarios publicados. El profesor les haría llegar el id del cuestionario o les proporcionaría el código QR o el enlace. Para ello, introducirían los siguientes datos en la vista de “unirse a test” accesible desde el menú de navegación, en la ruta “/test”.

Figura 9.25. Formulario para unirse a test

El sistema, tras haber comprobado los datos, permitiría a los usuarios ver las preguntas y unirse al cuestionario con una vista similar a la siguiente:

Figura 9.26. Vista de realización del cuestionario.

El sistema permite pasar preguntas de una en una así como el movimiento en el mapa de navegación del cuestionario que se muestra arriba a la derecha. Este mapa indica, además

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

la pregunta en la que se encuentra el usuario. Una vez se termina el cuestionario, el sistema podrá dar o no *feedback* en función de si está o no activada la opción.

Ha entrado en el test como
UsuarioManual
Puntuación del test
2.75/4 – 6.88/10
Resumen
Aprobado

During the Crusades (1095–1291), Europeans developed an appetite for spices, _____, porcelain, sugar, and other luxury items from the East, for which they traded fur, timber, and Slavics they captured and sold (hence the word slave)

1

cloth
 chiffon
 linen
 silk 

" West _____, a diverse and culturally rich area, soon entered the stage as other nations exploited its slave trade and brought its them to the new continent in chains

1

africa 
 continent
 zimbabwe
 country

 Q1
 Q2
 Q3
 X4

[Volver a la página principal](#)

Figura 9.27. Vista de retroalimentación del cuestionario.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

10 Conclusiones y ampliaciones

10.1 Conclusiones

En cuanto a estos meses de trabajo, este proyecto ha conseguido ahondar en numerosos aspectos y conceptos importantes en cuanto a la extracción de palabras clave se refiere. El algoritmo propuesto ofrece buenos resultados en comparación con otros algoritmos de enfoques diferentes.

Este trabajo también ha sido especialmente interesante, no solo por el algoritmo propuesto si no por la herramienta desarrollada que utiliza dicho algoritmo. Ofrecer una herramienta permite probarlo y darle, de algún modo, cierta utilidad a la propuesta. El desarrollo de esta herramienta ha permitido conocer nuevos *frameworks* y librerías de gran popularidad hoy en día. Asimismo, mis habilidades de desarrollo de aplicaciones web se han reforzado de una forma notable, ya que se ha implementado un perfil de aplicación diferente a muchas otras típicas como tiendas online, o redes de usuarios, etc.

Se han llevado a cabo y se han documentado todas las fases importantes a seguir a lo largo de un proyecto, tanto planificación como presupuesto, diseño, análisis e implementación. Esto ha permitido obtener como resultado una documentación completa y útil del proyecto. La herramienta utilizada, ofrece una gran interoperabilidad con muchas otras aplicaciones ya que ofrece una API REST de servicios que ofrece multitud de utilidades. Del mismo modo, también ofrece una interfaz web de usuario, intuitiva, adaptable y de calidad, pensada para realizar las tareas de la forma más eficiente posible siempre pensando en la experiencia del usuario.

Tanto la interfaz de usuario como la API REST de servicios cumplen notablemente con los objetivos planteados al principio del proyecto siendo ambas completamente independientes. Para la implementación se ha intentado seguir al máximo las mejores prácticas en cuanto a mantenibilidad se refiere, siguiendo al mismo tiempo las recomendaciones de los *frameworks* y librerías utilizadas.

10.2 Ampliaciones

En este apartado es interesante destacar algunas ampliaciones para llevar a cabo en un futuro. Entre estas se pueden comentar algunas opciones interesantes:

- Ampliar el algoritmo para la búsqueda de palabras clave tanto en inglés como en español, a día hoy la búsqueda de palabras clave del algoritmo solo se puede garantizar que es efectiva para textos en inglés.
- Mejorar la vista de resultados de los alumnos disponible para usuarios registrados. Exportaciones de notas, vista de las preguntas contestadas, resúmenes más detallados, etc. Esta ampliación pasaría por darle una vuelta y darle, a esta vista, posibilidades más avanzadas a los usuarios.

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

- Desplegar la aplicación en una máquina virtual a través de un proveedor de servicios en la nube como Azure, Google Cloud, AWS u Oracle Cloud que permita tener el sistema disponible a todo el mundo y no solo a personal conectado a la red de la Universidad de Oviedo.
- Implementar un sistema de integración continua que permita realizar despliegues y ejecución de pruebas automáticos. Esta ampliación ofrecería una gran agilidad, sin embargo, requiere de una configuración compleja y precisa.
- Implementar un sistema de recuperación de contraseñas para usuarios registrados, así como modificación de datos del perfil.
- Añadir nuevos idiomas en cuanto a internacionalización de la aplicación se refiere, con el fin de hacerla más accesible. Se podrían añadir idiomas con gran número de hablantes como son el chino, el árabe o el hindi. Idiomas que hoy en día están en auge y gozan de un número notable de hablantes.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

11 Referencias bibliográficas

- [1] J. Pascual Espada, J. Solís Martínez, I. Cid Rico, and L. E. Velasco Sánchez, “Extracting keywords of educational texts using a novel mechanism based on linguistic approaches and evolutive graphs,” *Elsevier*, 2022, Accessed: Jan. 26, 2023. [Online]. Available: <https://doi.org/10.1016/j.eswa.2022.118842>
- [2] S. Beliga, A. Meštrović, and S. Martinčić-Ipšić, “An overview of graph-based keyword extraction methods and approaches,” *Journal of information and organizational sciences*, vol. 39, no. 1, pp. 1–20, 2015.
- [3] “RAKE (Rapid Automatic Keyword Extraction) algorithm,” Sep. 15, 2021. <https://pypi.org/project/rake-nltk/> (accessed Mar. 15, 2023).
- [4] “YAKE (Yet Another Keyword Extractor) algorithm.” <https://github.com/LIAAD/yake> (accessed Mar. 15, 2023).
- [5] D. Gayo Avello, “Sistemas de información para la Web. Primer bloque de materiales: ‘Desde Memex a la Web Semántica.’”
- [6] K. Zhang, H. Xu, J. Tang, and J. Li, “Keyword extraction using support vector machine,” in *Advances in Web-Age Information Management: 7th International Conference, WAIM 2006, Hong Kong, China, June 17-19, 2006. Proceedings* 7, Springer, 2006, pp. 85–96.
- [7] Y. Uzun, “Keyword extraction using naive bayes,” in *Bilkent University, Department of Computer Science, Turkey* www.cs.bilkent.edu.tr/~guvenir/courses/CS550/Workshop/Yasin_Uzun.pdf, 2005.
- [8] Y. Kim *et al.*, “Validation of deep learning natural language processing algorithm for keyword extraction from pathology reports in electronic health records,” *Sci Rep*, vol. 10, no. 1, pp. 1–9, 2020.
- [9] W. Xu and G. Nong, “A study for extracting keywords from data with deep learning and suffix array,” *Multimed Tools Appl*, vol. 81, no. 5, pp. 7419–7437, 2022.
- [10] M. Tang, P. Gandhi, M. A. Kabir, C. Zou, J. Blakey, and X. Luo, “Progress notes classification and keyword extraction using attention-based deep learning models with BERT,” *arXiv preprint arXiv:1910.05786*, 2019.
- [11] “KeyBERT Algorithm.” Accessed: Jun. 01, 2023. [Online]. Available: <https://github.com/MaartenGr/KeyBERT>
- [12] L. Page, S. Brin, R. Motwani, and T. Winograd, “The PageRank Citation Ranking: Bringing Order to the Web,” 1998.
- [13] R. Mihalcea and P. Tarau, “TextRank: Bringing Order into Texts,” 2004.
- [14] Christian. Laville, *From yesterday to tomorrow: history and citizenship education, secondary cycle one, student textbook A*. Graficor, Chenelière éducation, 2008.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

- [15] D. M. Kennedy, L. Cohen, and T. A. Bailey, “The American Peageant,” 2011. [Online]. Available: www.cengage.com/history/kennedy/ampageant14e
- [16] B. Walsh, “Modern World History Cambridge IGCSE ® BEN WALSH 11:58,” 2013.
- [17] P. S. Corbett, V. Jansen, J. M. Lund, T. Pfannestiel, S. Waskiewicz, and P. Vickery, *U.S. History*. OpenStax, 2014. Accessed: Feb. 15, 2023. [Online]. Available: <https://openstax.org/details/books/us-history?Book%20details>
- [18] “F1 Score.” <https://en.wikipedia.org/wiki/F-score> (accessed Apr. 06, 2023).
- [19] “Summa - Textrank.” Accessed: Jun. 01, 2023. [Online]. Available: <https://github.com/summanlp/textrank>
- [20] “Yake Algorithm.” Accessed: Jun. 01, 2023. [Online]. Available: <https://pypi.org/project/yake/>
- [21] Project Management Institute, *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*, 5th ed. 2013.
- [22] “IEEE Guide for Software Requirements Specifications,” *IEEE Std 830-1984*, pp. 1–26, 1984, doi: 10.1109/IEEESTD.1984.119205.
- [23] “Testing in FastAPI Systems.” <https://fastapi.tiangolo.com/tutorial/testing/> (accessed May 27, 2023).
- [24] “SQLite.” <https://sqlite.org/index.html> (accessed May 18, 2023).
- [25] “SQLAlchemy Docs.” <https://www.sqlalchemy.org/> (accessed May 20, 2023).
- [26] J. M. Redondo López, “Documentos-modelo para Trabajos de Fin de Grado/Master de la Escuela de Informática de Oviedo,” Jun. 17, 2019. https://www.researchgate.net/publication/327882831_Plantilla_de_Proyectos_de_Fin_de_Carrera_de_la_Escuela_de_Informatica_de_Oviedo (accessed Feb. 23, 2023).
- [27] G. van Rossum, “Python Docs.” <https://www.python.org/doc/essays/styleguide/> (accessed Mar. 12, 2023).
- [28] A. A. Juan Fuente and B. López Pérez, *Guía de Aprendizaje de la asignatura de Dirección y Planificación de Proyectos Informáticos*. Universidad de Oviedo, 2022.
- [29] B. W. Boehm, “Software risk management: principles and practices,” *IEEE Softw.*, vol. 8, no. 1, pp. 32–41, 1991.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

12 Glosario

API REST	<i>Una API es una pieza de código que permite comunicación entre diferentes aplicaciones. REST no es un estándar, es más bien un estilo de arquitectura para sistemas distribuidos. Se suele utilizar combinado con la representación JSON y es una alternativa a los servicios SOAP basados en XML.</i>
Backend	<i>Es un término en inglés que hace referencia a una capa de representación, que en el desarrollo de aplicaciones informáticas hace referencia a la parte de acceso a datos que no ve el usuario.</i>
Caso de uso	<i>Representan qué hace el sistema. Los diagramas de casos de uso se utilizan para representar las funcionalidades del sistema desde el punto de vista del usuario.</i>
Clave foránea (FK)	<i>En el ámbito de las bases de datos relacionales, una clave foránea o ajena es una columna o conjunto de estas que referencia a columnas de otra tabla. Esto permite en muchos casos la navegabilidad entre diferentes tablas.</i>
Clave primaria (PK)	<i>En el ámbito de las bases de datos relacionales una clave primaria es un campo o una combinación de campos que permite identificar únicamente a una fila de una tabla.</i>
Diagrama de componentes	<i>Es un tipo de diagrama que permite ver la estructura de las diferentes partes del sistema. Permite analizar e identificar cómo se relacionan entre sí.</i>
Diagrama de navegabilidad	<i>Es un diagrama que representa cómo se relacionan las diferentes vistas o páginas de una aplicación.</i>
Diagrama E-R (Entidad-Relación)	<i>Un diagrama E-R permite conocer cómo se interrelacionan las entidades de un modelo de datos.</i>
Escenario	<i>Se utiliza en conjunto con los casos de uso para representar cómo funciona el sistema, así como la forma en que el usuario interactúa con el sistema.</i>
Framework	<i>Un framework hace referencia a “entorno de trabajo”, en el caso de la informática se suele hacer referencia a tecnologías que facilitan enormemente tareas de desarrollo de aplicaciones. Por ejemplo, Spring Boot es un framework que facilita mucho el desarrollo de aplicaciones web.</i>

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

Frontend	<i>Es un término en inglés que hace referencia a una capa de representación, que en el desarrollo de aplicaciones informáticas hace referencia a la parte que ve el usuario, es decir la interfaz de usuario.</i>
JSON	<i>Significa (JavaScript Object Notation) y es un formato para el intercambio de datos entre diferentes sistemas.</i>
Lenguaje compilado	<i>Un lenguaje compilado es aquel que necesita de un compilador. Un compilador no deja de ser un traductor que a partir de un lenguaje de alto nivel genera instrucciones de bajo nivel capaces de ser procesadas por una máquina. En este tipo de lenguajes, primero se hace la compilación y después se ejecuta. Un ejemplo podría ser Java.</i>
Lenguaje interpretado	<i>Un lenguaje interpretado necesita de un intérprete. Este tipo de lenguaje se sirven del intérprete para ir ejecutando el código fuente instrucción por instrucción, una a una. En este trabajo se utilizan tanto JavaScript como Python.</i>
ORM	<i>Técnica que facilita la conversión de datos entre un lenguaje de programación y el uso de una base de datos relacional.</i>
Requisito	<i>Los requisitos son una especificación de lo que se debe implementar. Son descripciones de cómo debe comportarse el sistema, o de una propiedad o atributo del sistema. Estos deben tener una serie de características, ya que deben ser concisos, no ambiguos, no dependientes de tecnología, etc.</i>
SGBD	<i>Un SGBD (Sistema Gestor de Base de Datos) es un software que permite la administración y gestión de bases de datos. Este tipo de programas permiten visualizar, modificar y leer la información por medio de consultas.</i>



13 Índice alfabético

A

Actores · 56
Alcance · 50, 56, 62, 177, 178
API REST · 6, 13, 17, 18, 39, 41, 107, 114, 151, 162, 166, 187

B

Backend · 166

C

Caso de uso · 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79
Clase equivalencia · 187
Clave foránea · 166
Clave primaria · 166
Condición de prueba · 187

D

Diagrama de casos de uso · 69, 71, 73, 77
Diagrama de componentes · 28, 110, 112, 113
Diagrama de contexto · 66, 67
Diagrama de despliegue · 118
Diagrama de Navegabilidad · 104
Diagrama Entidad-Relación · 109

E

Escenario · 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 166

F

F1 Score · 30, 35, 165
FastAPI · 40, 41, 165, 187, 188
Framework · 166
Frontend · 167

G

Grafo · 21, 23, 24, 25, 26, 27, 29

K

Keywords · 17, 18, 19, 20, 21, 27, 31, 32, 39, 114, 164, 188, 189

L

Lenguaje compilado · 167
Lenguaje interpretado · 167

M

Manual · 148, 154

O

ORM · 10, 107, 167

P

Petición de cambio · 134
Plan de pruebas · 48, 105, 119, 179, 180
Precisión · 29, 31, 32, 33, 34
Presupuesto · 52, 54, 173, 174, 175, 176, 178, 180, 182, 184, 185, 186, 187
Productividad · 171, 172
Pruebas con Usuarios · 126
Pruebas de Adaptabilidad · 129, 142
Pruebas de Sistema · 119
Pruebas Estáticas · 134
Pruebas Funcionales (end-to-end) · 119, 135
Pruebas Unitarias · 119, 135

R

React · 17, 18, 42, 43, 50, 51, 130, 151, 188
Recall · 29, 30, 31, 32, 33, 34
Requisito · 56, 60
Restapi · 96, 112, 119, 135, 154
Riesgo · 52

S

Score · 21, 28, 35, 36, 165



Seguridad · 51, 53, 66, 98, 132, 133, 170, 171

Sensitivity · 29

SGBD · 107, 131, 167

Sistema Gestor de Bases de Datos · 107

Situación de prueba · 187

T

TextRank · 17, 18, 23, 32, 33, 34, 35, 164

Token de acceso · 132

W

Webapp · 96, 110, 151, 154

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

14 Anexos

14.1 Plan de gestión de riesgos

El plan de gestión de riesgos permite conocer los procesos a la hora de llevar a cabo el manejo de los riesgos. Ya sea desde su identificación y análisis hasta su monitorización y control. Existen muchas metodologías que podemos seguir, en este caso se seguirá la propuesta en [28] que es una combinación de la propuesta del PMBOK [21] y de Boehm [29]. Aun así, en este trabajo se centrará en la Valoración de Riesgos, así como en la Planificación de la Gestión de cada Riesgo, dejando un poco de lado la Resolución y Monitorización.

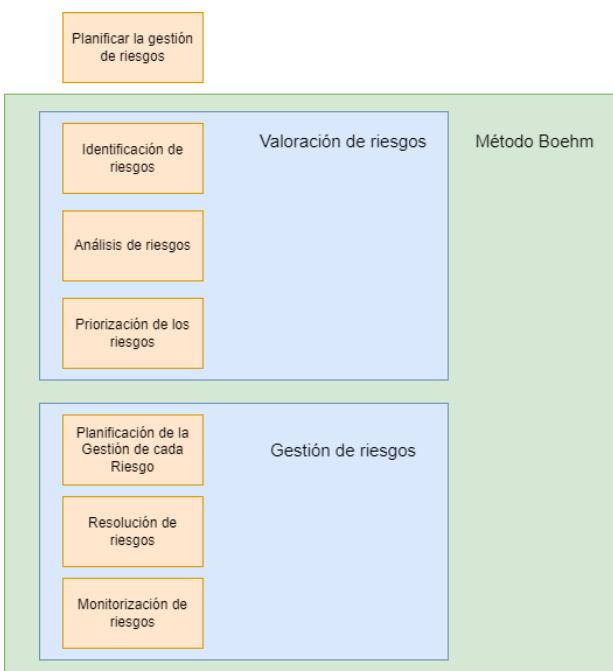


Figura 14.1. Metodología para el plan de gestión de riesgos.

Cada una de las fases se puede resumir en lo siguiente en base a lo expuesto en [28]:

- **Planificar la Gestión de riesgos:** Decidir las políticas a aplicar en la gestión de riesgos.
- **Valoración de Riesgos:** Está compuesta por las fases de Identificación, Análisis y Priorización de Riesgos, tal y como se aprecia en la imagen superior. Se realiza al inicio del desarrollo (con posibilidad de iterar varias).
- **Gestión de Riesgos:** Está compuesta por las fases de Planificación de la Gestión de cada Riesgo, Resolución de Riesgos y Monitorización de Riesgos como se ve en la imagen superior. Estas fases se realizan en parte antes del desarrollo, pero tienen una clara importancia durante el desarrollo.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

14.1.1 Valoración de riesgos

14.1.1.1 Identificación de Riesgos

En este paso se recopilan posibles riesgos que pueden afectar al proyecto. Se utilizarán diferentes técnicas:

- **Técnicas de recopilación de información:** principalmente técnicas de tormentas de ideas (*brainstorming*).
- **Análisis de supuestos:** se utilizarán hipótesis que se puedan extraer al ámbito del proyecto de forma que ayuden a identificar nuevos riesgos.

De la identificación de riesgos se obtendrá un registro de riesgos con la siguiente información para cada riesgo:

- **Identificador**
- **Nombre**
- **Descripción**
- **Categoría**

14.1.1.2 Análisis de riesgos

Esta fase del plan de gestión de riesgos se determinará una probabilidad para cada uno de los riesgos identificados. Para ello se parte del registro obtenido anteriormente y se analizará cada uno de los riesgos en base a una evaluación de probabilidad e impacto de cada riesgo.

14.1.1.3 Priorización de riesgos

En esta etapa se hará la ordenación de la lista de riesgos. En este apartado se partirá del registro anterior y se usará la técnica de matriz de probabilidad e impacto. De esta etapa obtendríamos el registro obtenido en la fase anterior pero priorizado por la importancia de cada uno de los riesgos.

14.1.2 Gestión de riesgos

14.1.2.1 Planificación de la gestión de cada riesgo

En esta etapa a partir del registro anterior se aplicarán diferentes estrategias para riesgos o amenazas. Se determinará la estrategia a seguir. Entre las estrategias que se barajarán serán las siguientes:

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

- **Eliminar del riesgo:** esta técnica consiste en eliminar los factores que pueden producir el riesgo.
- **Transferir el riesgo:** esta técnica se basa en delegar los riesgos en terceros mediante la contratación de servicios o seguros.
- **Mitigar el riesgo:** aquí se busca identificar los factores que inducen el riesgo y tenerlos controlados para que sus efectos sean mínimos.
- **Asumir el riesgo:** en este caso se opta por aceptar el riesgo y convivir con él ya que no se puede hacer nada para evitarlo.

Se obtiene como salida una estrategia para encarar cada uno de los riesgos junto con una descripción justificando esta estrategia.

14.1.2.2 Resolución de riesgos

Esta fase no se desarrollará ampliamente en el trabajo. Aun así, la resolución de riesgos es la fase en la que se medita si cada uno de los riesgos se ha solucionado en comparación a la estrategia planteada al inicio.

14.1.2.3 Monitorización de riesgos

Aunque esta fase tampoco se desarrollará ampliamente en el trabajo. La monitorización y el control de los riesgos se centra en seguimiento de los riesgos a lo largo del proyecto. De aquí se persigue obtener una actualización del registro de riesgos.

14.2 *Modelo de empresa supuesto*

En este apartado se describe el modelo de empresa supuesto para la realización de este proyecto. Para ello se ha supuesto que el único de miembro de la empresa es un solo Ingeniero de Software, dada la naturaleza del trabajo.

14.2.1 Costes de personal

Personal	Núm.	Sueldo bruto anual	Coste salarial anual ⁴	Total
Ingeniero de Software	1	32.000,00 €	40.640,00 €	40.640,00 €
Total	1			40.640,00 €

⁴ El coste salarial es lo que le cuesta el empleado a la empresa, para ello se ha determinado que es el sueldo bruto más un 27% del mismo, de esta forma se cubre lo relativo a impuestos por la seguridad social, bonus y demás pagas extras entre otros aspectos de índole financiera

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

Tabla 14.1. Información de costes del personal supuesto en la empresa.

14.2.2 Productividad del personal

Personal	Total	Productividad (%)	Coste Directo	Coste indirecto (%)	Coste Indirecto
Ingeniero de Software	40.640,00 €	70%	28.448,00 €	30%	12.192,00 €
Total	40.640,00 €		28.448,00 €		12.192,00 €

Tabla 14.2. Productividad y relación de costes directos e indirectos en salarios.

En este caso se define, para la empresa, la productividad de los trabajadores. En este caso para el único trabajador se ha supuesto un porcentaje de productividad del 70%, quizás un poco bajo, pero se ha supuesto ya que este trabajo se compaginado con asignaturas del grado.

Los costes indirectos son aquellos costes que no se relacionan estrechamente a la producción, pero que son necesarios para que se lleve a cabo. Otros costes

Aquí se detallan otro tipo de costes indirectos necesarios para la empresa, en este caso se busca simular y darle cierto contexto a la empresa.

Servicio	Coste mes	Coste año
Limpieza	80,00 €	960,00 €
Asesoría legal	100,00 €	1.200,00 €
Seguros	200,00 €	2.400,00 €
Auditorias de seguridad	100,00 €	1.200,00 €
Material de oficina	20,00 €	240,00 €
Gasto de agua	60,00 €	720,00 €
Gasto de electricidad	200,00 €	2.400,00 €
Controles de calidad	100,00 €	1.200,00 €
Correos y mensajería	10,00 €	120,00 €
Gastos de mantenimiento	150,00 €	1.800,00 €
Impuestos varios	100,00 €	1.200,00 €

13.440,00 €

Tabla 14.3. Costes indirectos para el funcionamiento de la empresa supuesta.

También existen otros costes aplicados a las licencias del software o equipos utilizados para la realización del proyecto. Para los equipos se ha decidido optar por amortización ya que, al ser un material adquirido, este se amortiza pasado un tiempo, en este caso se ha puesto 4 años. En el caso de las licencias se optaría por alquiler ya que va por suscripción mensual.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

Equipo/Licencia	Unid.	Precio	Coste total	Coste año	Tipo	Plazo
Portátil	1	1.400,00 €	1.500,00 €	375,00 €	Amortización	4
Licencia JetBrains PyCharm	1	65,00 €	65,00 €	780,00 €	Alquiler	
Licencia Microsoft Project 2019	1	27,00 €	27,00 €	324,00 €	Alquiler	
Licencia Office 365	1	11,70 €	11,70 €	140,40 €	Alquiler	

1.619,40 €

Tabla 14.4. Costes indirectos relativos a materiales para el desarrollo del proyecto.

14.2.3 Horas productivas y precio hora

En la tabla siguiente se muestran tanto las horas de trabajo anuales como las horas realmente productivas en base al porcentaje de productividad de 70% así como un precio/hora que permite obtener unos valores de facturación total de la empresa de 64.166,80€.

Personal	Prod. (%)	Horas/año	Horas productivas/año (por persona)	Horas productivas/año (totales)	Precio hora (con beneficios ⁵)	Facturación	Precio hora (sin beneficios) ⁶
Ingeniero de Software	70%	2008 ⁷	1405,6	1405,6	51,53 €	72.435,45 €	39,00 €

1405,6

72.435,45 €

Tabla 14.5. Horas productivas y precio por hora con y sin beneficios.

14.2.4 Resumen modelo de empresa

En la siguiente tabla figura el total de costes directos, especificado en la Tabla 14.2. Productividad y relación de costes directos e indirectos en salarios., los costes indirectos que son la suma de los especificados en la Tabla 14.2, la Tabla 14.3 y la Tabla 14.4. Asimismo, la suma de costes directos e indirectos con su porcentaje de beneficio y el coste total en el que se agrega todos los costes (directos e indirectos) y los beneficios.

Del mismo modo se especifica la facturación posible en base al precio por hora (con beneficios) establecido en la Tabla 14.5.

⁵ Suponemos que queremos obtener unos beneficios del 25%.

⁶ Será el precio/hora que se use para el cálculo del presupuesto costes.

⁷ Horas anuales tomadas como referencia del calendario del año 2019.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

Concepto	Importe
Total costes directos	28.448,00 €
Total costes indirectos	27.251,40 €
Suma costes directos e indirectos	55.699,40 €
Beneficio deseado (25%)	13.924,85 €
Coste total	69.624,25 €
Facturación posible en base a horas de producción	72.435,45 €
Margen entre coste total y facturación	3,88%

Tabla 14.6. Resumen de modelo de la empresa.

14.3 Detalle del presupuesto

Para calcular el presupuesto inicial se han utilizado las siguientes partidas del proyecto:

- **Partida 1:** Investigación del algoritmo
- **Partida 2:** Estudios y planificación inicial
- **Partida 3:** Análisis del sistema
- **Partida 4:** Diseño del sistema
- **Partida 5:** Implementación y pruebas del sistema
- **Partida 6:** Revisión de la documentación
- **Partida 7:** Otros costes

A continuación, se irá haciendo la descripción detallada del presupuesto por cada partida. Aunque los cálculos en detalle se adjuntan en los anexos⁸.

14.3.1 Presupuesto de costes

14.3.1.1 Partida 1: Investigación del algoritmo

La partida 1 contienen lo relativo a investigación, entre lo que destacan tareas de elaboración del contexto del problema, implementación y análisis de resultados.

I1	I2	Descripción
01		Investigación del algoritmo
	001	Desarrollo del contexto del problema
	002	Implementación del algoritmo
	003	Análisis de resultados

Tabla 14.7. Tareas necesarias para la Partida 1.

⁸ El detalle se puede ver en el documento adjuntado en los anexos denominado “PresupuestoTFG.xlsx”.

I1	I2	I3	Descripción	Cantidad	Unidades	Precio	Subtotal (3)	Subtotal (2)	Total
01			Investigación del algoritmo						1.404,00 €
	001		Desarrollo del contexto del problema					234,00 €	
		01	Ingeniero de Software	6	horas	39,00 €	234,00 €		
	002		Implementación del algoritmo					936,00 €	
		02	Ingeniero de Software	24	horas	39,00 €	936,00 €		
	003		Análisis de resultados					234,00 €	
		01	Ingeniero de Software	6	horas	39,00 €	234,00 €		

Total	1.404,00 €
-------	-------------------

Tabla 14.8. Presupuesto de costes de la partida de 1 de investigación del algoritmo.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

14.3.1.2 Partida 2: Estudios y planificación inicial

La partida 2 consta de las siguientes tareas correspondientes a la planificación y presupuestos iniciales, así como los estudios de viabilidad:

I1	I2	I3	Descripción
01			Elaboración de presupuestos y planificación inicial
	001		Planificación inicial
		01	Creación de EDT (WBS)
		02	Estimación de actividades
		03	Identificación de riesgos y plan de gestión de riesgos
	002		Presupuesto inicial
	001		Presupuesto de costes inicial
02			Estudio de viabilidad del sistema
	001		Análisis de sistemas similares
	002		Valoración de alternativas de solución y selección final

Tabla 14.9. Tareas correspondientes a la Partida 2.

I1	I2	I3	I4	Descripción	Cantidad	Unidades	Precio	Subtotal (4)	Subtotal (3)	Subtotal (2)	Total
01				Elaboración de presupuestos y planificación inicial							858,00 €
	001			Planificación inicial							468,00 €
		01		Creación de EDT (WBS)							156,00 €
			001	Ingeniero de Software	4	horas	39,00 €	156,00 €			
		02		Estimación de actividades							78,00 €
			001	Ingeniero de Software	2	horas	39,00 €	78,00 €			
		03		Identificación de riesgos y plan de gestión de riesgos							234,00 €
			001	Ingeniero de Software	6	horas	39,00 €	234,00 €			
	002			Presupuesto inicial							390,00 €
		01		Presupuesto de costes inicial							390,00 €
			001	Ingeniero de Software	10	horas	39,00 €	390,00 €			
02				Estudio de viabilidad del sistema							234,00 € 234,00 €
	001			Análisis de sistemas similares							78,00 €
		01		Ingeniero de Software	2	horas	39,00 €	78,00 €			
	002			Valoración de alternativas de solución y selección final							156,00 €
		01		Ingeniero de Software	4	horas	39,00 €	156,00 €			
Total											1092,00 €

Tabla 14.10. Presupuesto de costes para partida 2 de estudios y planificación.

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

14.3.1.3 Partida 3: Análisis del sistema

La partida de análisis del sistema consta de las siguientes tareas:

I1	I2	Descripción
01		Análisis del sistema
	001	Determinar el alcance del sistema
	002	Obtención de requisitos del sistema
	003	Ánálisis de escenarios y casos de uso
	004	Identificación de subsistemas
	005	Definición de interfaces de usuario

Tabla 14.11. Tareas relativas a la Partida 3.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

I1	I2	I3	Descripción	Cantidad	Unidades	Precio	Subtotal (3)	Subtotal (2)	Total
01			Análisis del sistema						2.184,00 €
	001		Determinar el alcance del sistema					156,00 €	
		01	Ingeniero de Software	4	horas	39,00 €	156,00 €		
	002		Obtención de requisitos del sistema					702,00 €	
		01	Ingeniero de Software	18	horas	39,00 €	702,00 €		
	003		Análisis de escenarios y casos de uso					702,00 €	
		01	Ingeniero de Software	18	horas	39,00 €	702,00 €		
	004		Identificación de subsistemas					156,00 €	
		01	Ingeniero de Software	4	horas	39,00 €	156,00 €		
	005		Definición de interfaces de usuario					468,00 €	
		01	Ingeniero de Software	12	horas	39,00 €	468,00 €		
							Total	2.184,00 €	

Tabla 14.12. Presupuesto de costes para la partida 3 de análisis del sistema.

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

14.3.1.4 Partida 4: Diseño del sistema

La partida de diseño consta de las siguientes tareas a incluir en el presupuesto:

I1	I2	Descripción
01		Diseño del sistema
	001	Diagramas de arquitectura y componentes del sistema
	002	Diseño físico de datos
	003	Diseño de la interfaz
	004	Especificación del plan de pruebas

Tabla 14.13. Tareas relativas a la partida 4 de diseño del sistema.

I1	I2	I3	Descripción	Cantidad	Unidades	Precio	Subtotal (3)	Subtotal (2)	Total
01			Diseño del sistema						1.950,00 €
	001		Diagramas de arquitectura y componentes del sistema					858,00 €	
		01	Ingeniero de Software	22	horas	39,00 €	858,00 €		
	002		Diseño físico de datos					546,00 €	
		01	Ingeniero de Software	14	horas	39,00 €	546,00 €		
	003		Especificación del plan de pruebas					546,00 €	
		01	Ingeniero de Software	14	horas	39,00 €	546,00 €		
Total									1.950,00 €

Tabla 14.14. Presupuesto de costes para la partida 4 de diseño del sistema

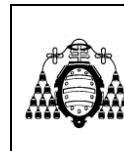
	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

14.3.1.5 Partida 5: Implementación y pruebas del sistema

La partida de implementación y pruebas consta de las siguientes tareas a incluir en el presupuesto:

I1	I2	Descripción
01		Implementación y pruebas del sistema
	001	Formación en desarrollo de REST API aplicaciones web
	002	Implementación de la REST API
	003	Implementación de pruebas unitaria y de la aplicación web
	004	Formularios de pruebas de usabilidad
	005	Otros tipos de pruebas

Tabla 14.15. Tareas relativas a la partida 5.



I1	I2	I3	Descripción	Cantidad	Unidades	Precio	Subtotal (3)	Subtotal (2)	Total
01			Implementación y pruebas del sistema						4.524,00 €
	001		Formación en desarrollo de REST API apps web					156,00 €	
		01	Ingeniero de Software	4	horas	39,00 €	156,00 €		
	002		Implementación de la REST API					1.170,00 €	
		01	Ingeniero de Software	30	horas	39,00 €	1.170,00 €		
	003		Implementación de pruebas unitaria y de la aplicación web					2.730,00 €	
		01	Ingeniero de Software	70	horas	39,00 €	2.730,00 €		
	004		Formularios de pruebas de usabilidad					234,00 €	
		01	Ingeniero de Software	6	horas	39,00 €	234,00 €		
	005		Otros tipos de pruebas					234,00 €	
		01	Ingeniero de Software	6	horas	39,00 €	234,00 €		

Total	4.524,00 €
--------------	-------------------

Tabla 14.16. Presupuesto de costes para partida 5 de implementación y pruebas.

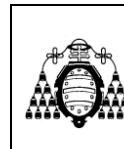
	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

14.3.1.6 Partida 6: Revisión de la documentación

La partida de revisión de la documentación consta de las siguientes tareas a incluir en el presupuesto:

I1	I2	Descripción
01		Documentación restante
	001	Documentación de la implementación (manuales)
	002	Documentación de la ejecución de las pruebas
	003	Planificación final
	004	Conclusiones y ampliaciones
	005	Revisión de los anexos

Tabla 14.17. Tareas relativas a la partida 6.



I1	I2	I3	Descripción	Cantidad	Unidades	Precio	Subtotal (3)	Subtotal (2)	Total
01			Revisión de la documentación						546,00 €
	001		Documentación de la implementación					78,00 €	
		01	Ingeniero de Software	2	horas	39,00 €	78,00 €		
	002		Documentación de la ejecución de las pruebas					156,00 €	
		01	Ingeniero de Software	4	horas	39,00 €	156,00 €		
	003		Planificación final					156,00 €	
		01	Ingeniero de Software	4	horas	39,00 €	156,00 €		
	004		Conclusiones y ampliaciones					78,00 €	
		01	Ingeniero de Software	2	horas	39,00 €	78,00 €		
	005		Revisión anexos					78,00 €	
		01	Ingeniero de Software	2	horas	39,00 €	78,00 €		
								Total	546,00 €

Tabla 14.18. Presupuesto de costes de la partida 6 de documentación restante.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

14.3.1.7 Partida 7: Otros costes

En esta partida se especifican otros costes relativos a la realización del proyecto. Son costes más indirectos que no están directamente relacionados con la realización del proyecto en sí, si no con aspectos como los desplazamientos necesarios para llevar a cabo el proyecto. Se puede ver en la siguiente tabla:

I1	I2	I3	Descripción	Cantidad	Unidades	Precio	Subtotal (3)	Subtotal (2)	Total
01			Otros costes						36,00 €
	001		Viajes (desplazamientos a la universidad)					36,00 €	
		01	Desplazamiento en tren (15 días x 12 km/día)	180	km	0,20 €	36,00 €		
							Total	36,00 €	

Tabla 14.19. Presupuesto de costes de la partida 7 de otros costes

14.3.1.8 Presupuesto de costes (Agregado)

Aquí se muestra un resumen agregado del presupuesto de costes del proyecto, en esta tabla se incluye un código para cada partida con el coste total de la misma. Se muestra al final de la tabla el coste total.

Código	Partida	Total
01	Partida 1: Investigación del algoritmo	1.404,00 €
02	Partida 2: Estudios y planificación inicial	1.092,00 €
03	Partida 3: Análisis del sistema	2.184,00 €
04	Partida 4: Diseño del sistema	1.950,00 €
05	Partida 5: Implementación y pruebas del sistema	4.524,00 €
06	Partida 6: Revisión de la documentación	546,00 €
07	Partida 7: Otros costes	36,00 €
Total Coste		11.736,00 €

Tabla 14.20. Presupuesto de costes agregado.

14.3.2 Presupuesto de cliente

El presupuesto de cliente es el presupuesto que se le envía al cliente. Este presupuesto es el que tiene incluidos los porcentajes de beneficios (25% en el caso de este proyecto).

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

Asimismo, en este presupuesto no se pueden mostrar partidas relativas a “Otros costes” ya que no son de interés para el proyecto. En este caso se mostrarían los presupuestos de las partidas de cliente que son las siguientes:

Código	Partida
01	Investigación del algoritmo
02	Estudios y planificación inicial
03	Análisis del sistema
04	Diseño del sistema
05	Implementación y pruebas del sistema
06	Revisión de la documentación

Tabla 14.21. Partidas e ítems de cliente.

Entre estos presupuestos de costes de las partidas debemos repartir los siguientes costes:

Repartir entre las partidas de cliente	
Beneficios para obtener (25% del presupuesto de costes total)	2.934,00 €
Otros costes	36,00 €
Total a repartir	2.970,00 €

Tabla 14.22. Coste para repartir entre las demás partidas.

Para repartirlo lo que se utiliza es una ponderación de forma que se reparta equitativamente entre los costes de cada partida. Para ello ponderamos este total a repartir entre cada partida, lo que nos da un resultado tal que así:

Código	Partida	Total coste	Ponderación	Total cliente
01	Investigación del algoritmo	1.404,00 €	273,77 €	1.677,77 €
02	Estudios y planificación inicial	1.092,00 €	228,14 €	1.320,14 €
03	Análisis del sistema	2.184,00 €	197,73 €	2.381,73 €
04	Diseño del sistema	1.950,00 €	243,35 €	2.193,35 €
05	Implementación y pruebas del sistema	4.524,00 €	1.231,98 €	5.755,98 €
06	Revisión de la documentación	546,00 €	106,47 €	652,47 €

Tabla 14.23. Presupuesto de coste con ponderaciones y presupuesto cliente.

Ahora el presupuesto de cliente resumido y el que se le entregaría, sería el siguiente:

Código	Partida	Total cliente
01	Investigación del algoritmo	1.677,77 €
02	Estudios y planificación inicial	1.320,14 €
03	Análisis del sistema	2.381,73 €
04	Diseño del sistema	2.193,35 €
05	Implementación y pruebas del sistema	5.755,98 €
06	Revisión de la documentación	652,47 €

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

Total Coste sin IVA	13.981,44 €
IVA (21%)	2.936,10 €
Total Coste con IVA	16.917,55 €

Tabla 14.24. Presupuesto de cliente resumido.

14.4 Diseño de las pruebas

Para el diseño se ha utilizado una estrategia de partición en clases de equivalencia de diferentes condiciones de prueba que se han identificado. Estas clases de equivalencia se pueden combinar de muchas formas y son las que nos permiten obtener las diferentes situaciones de prueba. Se pueden hacer combinaciones parciales, combinaciones totales, entre otras más específicas como *base choice*.

Ejemplo con combinación total:

- **Condición de prueba 1**
 - Clase equivalencia 11
 - Clase equivalencia 12
- **Condición de prueba 2**
 - Clase equivalencia 21
 - Clase equivalencia 22
- Combinación total (**Condición de prueba 1 – 2**)
 - Clase equivalencia 11
 - Clase equivalencia 21 (*Situación de prueba 1*)
 - Clase equivalencia 11
 - Clase equivalencia 22 (*Situación de prueba 2*)
 - Clase equivalencia 12
 - Clase equivalencia 21 (*Situación de prueba 3*)
 - Clase equivalencia 12
 - Clase equivalencia 22 (*Situación de prueba 4*)

Las situaciones de prueba nos permiten después saber qué hay que probar, estas serán las que tendremos que cubrir con los diferentes casos de prueba que se crean. Estas situaciones de prueba se identifican por las de menor nivel y se puede obtener toda la información de una situación de prueba yendo de forma ascendente hacia las de mayor nivel.

14.5 Especificación OpenAPI (OAS)

Además de la facilidad para creación de API REST, el *framework* FastAPI ofrece ventajas a la hora de la generación automática de documentación en base al código desarrollado. Además, FastAPI nos ofrece interfaces de usuario interactivas de documentación de

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

forma automática que pueden integrarse fácilmente en nuestra aplicación. OpenAPI es un estándar que dicta una serie de normas que abogan por la unificación a la hora de documentar las API REST, todo ello con el fin último de lograr la mayor similitud entre todas. Esta apoyado por múltiples empresas de referencia en el sector de la informática como son Cisco, SAP o IBM entre otras. Se puede ver más información en este enlace <https://www.openapis.org/>. Esta especificación con OpenAPI no deja de dar como resultado un fichero JSON de información sobre el sistema que puede ser interpretado por una máquina fácilmente.

14.6 Librerías de componentes en React

Tal y como se comenta en el apartado 3.2.3 sobre tecnologías para el desarrollo del *frontend*, React se basa en componentes. Crear componentes desde cero implicaría una mayor carga de trabajo. Además, una vez se ha implementado un componente, se puede reutilizar, lo que da gran ventaja y versatilidad a las librerías de componentes. En este proyecto se ha utilizado la librería de componentes *Ant design*.

- **Enlace:** <https://ant.design/>

A pesar de que se ha usado esta, existen muchas otras de gran popularidad y muy similares como son las siguientes:

- **Mui:** <https://mui.com/>
- **React Boostrap:** <https://react-bootstrap.github.io/>
- **Fluent UI:** <https://developer.microsoft.com/en-us/fluentui>

Muchas de estas librerías ofrecen funcionalidades que agilizan la tarea de crear interfaces de usuario. Ya que la mayoría de sus componentes están orientados a diseños responsivos y adaptables a diferentes resoluciones de pantalla.

14.7 ChatGPT en la búsqueda de keywords

14.7.1 ¿Qué es ChatGPT?

ChatGPT es una herramienta desarrollada por OpenAI que está muy de moda hoy en día. ChatGPT no deja de ser un modelo de aprendizaje automático entrenado con datos de hasta 2021 (al menos a fecha de 14/03/2023).

Permite mantener una conversación, responder preguntas y resolver dudas de forma similar a cómo haríamos en una conversación con una persona. El gran *hándicap* de estas herramientas basadas en modelos de aprendizaje es que dependen enormemente del conjunto de datos utilizado para su entrenamiento, el cual, nunca representa al máximo el mundo real. Por este motivo y a pesar de su gran utilidad hay que desconfiar de cualquier

	ESCUELA DE INGENIERÍA INFORMÁTICA		<i>ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS</i> TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

duda que nos resuelva ya que en muchas ocasiones son respuestas erróneas, confusas o imprecisas que debemos contrastar.

14.7.2 ¿Cómo probar ChatGPT en la búsqueda de *keywords*?

En este apartado se busca explicar el procedimiento que se ha seguido para probar la búsqueda de palabras clave usando esta herramienta. En los siguientes puntos se especifican los pasos:

1. Nos aseguramos de que conoce el libro sobre el que vamos a probar. Para ello le hacemos una pregunta indicándole la referencia bibliográfica la cual permite conocer el libro de forma precisa. Se hace de esta forma ya que otra opción sería pegar el texto, pero, la aplicación permite textos de pocos caracteres con lo cual se hace difícil. Otra opción sería pegar por trozos, pero esto desvirtuaría el experimento respecto al resto de opciones.

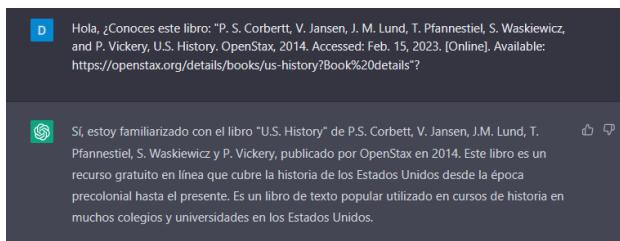


Figura 14.2. Pregunta a GPT del libro a partir de referencia bibliográfica.

2. Le hacemos una pregunta para que nos obtenga una lista de palabras clave en inglés, de esta forma podemos comprobar y contrastar cuántas palabras encuentra y cuántas no. En ocasiones, basta con poner simplemente el número de capítulo pero en otros casos hay que indicarle el título del capítulo. Hay que tener en cuenta que al darle el título del capítulo puede ser que tienda más a basarse en la temática y no tanto en el texto.



Figura 14.3. Listado de palabras clave encuentra.

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

Una vez tenemos la lista podemos comparar ya con la lista de palabras clave real y la propuesta en la imagen anterior, evaluando las métricas especificadas en el apartado 2.3 sobre análisis de resultados.

14.8 Gestores de referencias: Mendeley

Dada la parte de investigación que se incluye en el apartado 2 y la necesidad de hacer referencia a diferentes artículos científicos y páginas web, hace que gestionar las referencias sea complejo. Para ello se ha utilizado la herramienta Mendeley que permite incluir y manejar las referencias de forma automática. A esta herramienta se puede acceder utilizando las credenciales de la universidad. Se pueden crear colecciones de citas bibliográficas, se pueden editar cada una de las citas e importar nuevas citas. Asimismo se puede modificar muy rápidamente el estilo de la citación.

La aplicación Mendeley de escritorio utilizada es la siguiente:

- **Versión utilizada:** 2.80.1
- **Desarrollador:** Mendeley Lt.
- **Enlace:** <https://www.mendeley.com/download-reference-manager/windows>

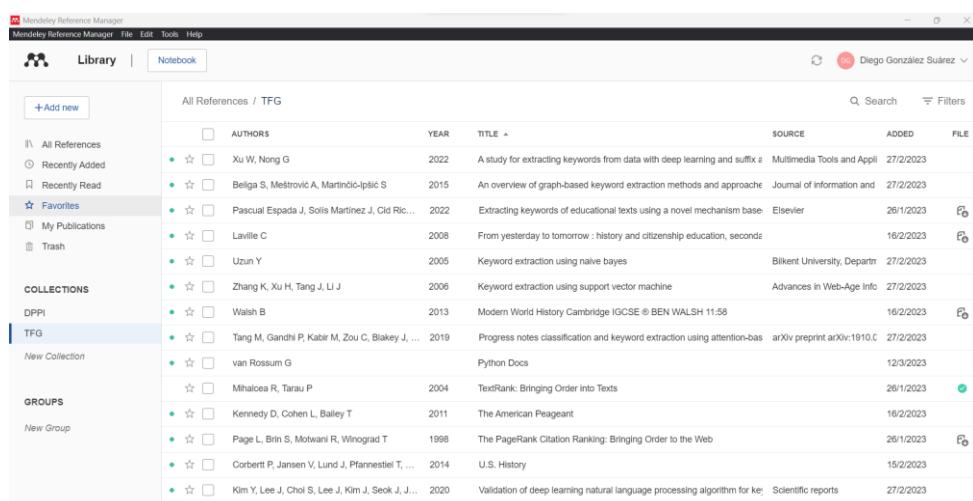


Figura 14.4. Vista de la aplicación de escritorio de Mendeley.

La extensión de Word de Mendeley utilizada es la siguiente:

- **Versión Utilizada:** 1.56.0
- **Desarrollador:** Elsevier Lt.
- **Enlace:** <https://appsource.microsoft.com/es/product/office/wa104382081?exp=ubp8>

	ESCUELA DE INGENIERÍA INFORMÁTICA		ALGORITMO PARA LA EXTRACCIÓN DE KEYWORDS DE TEXTOS EDUCATIVOS TRABAJO FIN DE GRADO
DIEGO GONZÁLEZ SUÁREZ – uo276406@uniovi.es			

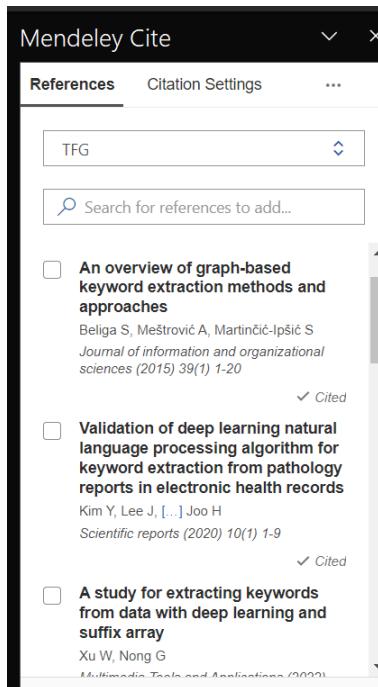


Figura 14.5. Extensión de Mendeley para Word.

14.9 Contenido entregado en los anexos

El contenido entregado en los anexos del trabajo es un fichero comprimido denominado “Anexos.zip” cuyo contenido es el siguiente:

- **PresupuestoTFG.xlsx:** es un documento de Excel que contiene el detalle con el cálculo del presupuesto, el cual también se describe en la memoria.
- **PlanificacionInicialTFG.mpp:** contiene la planificación realizada al inicio del proyecto, la cual conforma la estructura de desglose de trabajo. Se detalla también en la memoria.
- **PlanificacionTFG.mpp:** contiene la planificación final del proyecto. Concretamente es el fichero que ha servido para llevar a cabo el seguimiento del proyecto.
- **Declaracion.pdf:** contiene la declaración de que la obra es completamente original. Este documento está firmado digitalmente con el certificado emitido por la FNMT.
- **Un directorio denominado “tfq_app”:** este contiene el código fuente de la aplicación desarrollada. Contiene dos subdirectorios denominados “webapp” y “restapi”, cada uno contiene el código fuente respectivo a dicho subsistema de la aplicación.