



Universidad
Nacional
de Loja

Universidad Nacional de Loja

Facultad de la Energía, las Industrias y los Recursos
Naturales No Renovables

Carrera de Computación

**Prototipo de Software Antiplagio para la Carrera de
Ingeniería en Computación de la Universidad Nacional
de Loja, utilizando Redes Neuronales.**

Trabajo de Integración Curricular previa a
la obtención del título de Ingeniero en
Ciencias de la Computación

AUTOR:

Santiago Alexander Román Silva

Jaime Oswaldo Paqui Medina

DIRECTOR:

Ing. Oscar Miguel Cumbicus Pineda, Mg. Sc

Loja - Ecuador

2025

Certificación

Loja, 12 de marzo de 2024

Ing. Oscar Miguel Cumbicus Pineda, Mg. Sc.

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICO:

Que he revisado y orientado todo el proceso de elaboración del Trabajo de Integración Curricular denominado: **Prototipo de Software Antiplagio para la Carrera de Ingeniería en Computación de la Universidad Nacional de Loja, utilizando Redes Neuronales**, previo a la obtención del título de **Ingenieros en Ciencias de la Computación**, de autoría de los estudiantes **Jaime Oswaldo Paqui Medina con cédula de identidad N° 1900549170** y **Santiago Alexander Román Silva, con cédula de identidad N° 1106041989**, una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja, para el efecto, autorizo la presentación del mismo para su respectiva sustentación y defensa.

Ing. Oscar Miguel Cumbicus Pineda, Mg. Sc.

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Autoría

Nosotros, **Jaime Oswaldo Paqui Medina y Santiago Alexander Román Silva**, declaramos ser autores del presente Trabajo de Integración Curricular y eximimos expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos, de posibles reclamos y acciones legales, por el contenido del mismo. Adicionalmente aceptamos y autorizamos a la Universidad Nacional de Loja la publicación de mi Trabajo de Integración Curricular, en el Repositorio Digital Institucional – Biblioteca Virtual.

Firma:

Cédula de identidad: 1900823236

Fecha: 05 de marzo del 2025

Correo electrónico: jaime.paqui@unl.edu.ec

Teléfono: 0985606783

Firma:

Cédula de identidad: 1106041989

Fecha: 05 de marzo del 2025

Correo electrónico: santiago.roman@unl.edu.ec

Teléfono: 0990381254

Carta de autorización del estudiante

Nosotros, **Jaime Oswaldo Paqui Medina y Santiago Alexander Román Silva**, declaramos ser autores del Trabajo de Integración Curricular titulado “Prototipo de Software Anti plagio para la Carrera de **Ingeniería en Computación** de la Universidad Nacional de Loja, utilizando Redes Neuronales” como requisito para optar por el título de Ingenieros en Computación, autorizamos al Sistema Bibliotecario de la Universidad Nacional de Loja para que, con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido en el Repositorio Institucional. Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Institucional, en las redes de información del país y del exterior, con los cuales tenga convenio la Universidad. La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del trabajo de titulación que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los cuatro días del mes de mayo de dos mil veintidós.

Firma:

Firma:

Autor: Jaime Oswaldo Paqui Medina

Autor: Santiago A. Román Silva.

Cédula de identidad: 1900823236

Cédula de identidad: 1106041989

Fecha: 05 de marzo del 2025

Fecha: 05 de marzo del 2025

Correo electrónico:
jaime.paqui@unl.edu.ec

Correo electrónico:
santiago.roman@unl.edu.ec

Teléfono: 0986542760

Teléfono: 0990381254

DATOS COMPLEMENTARIOS:

Director del Trabajo de Titulación: Ing. Oscar Miguel Cumbicus Pineda, Mg. Sc.

Dedicatoria

Dedicamos este logro a todos aquellos que han compartido su apoyo y aliento a lo largo de este viaje académico. A nuestras familias y seres queridos, por su constante respaldo, paciencia y comprensión durante esta travesía. Sus palabras de aliento y cariño fueron un pilar fundamental que nos impulsó a dar lo mejor de nosotros en cada etapa. A nuestros profesores y mentores, cuya orientación y sabiduría guiaron nuestro camino. Sus conocimientos y guía nos permitieron explorar y comprender a profundidad el fascinante mundo de las redes neuronales en el análisis de plagio. A nuestros compañeros de clase, por compartir risas, debates y desafíos en cada paso del recorrido. Su colaboración y amistad enriquecieron nuestra experiencia y nos recordaron la importancia de la comunidad en el proceso de aprendizaje. Agradecemos especialmente a aquellos que participaron en la revisión y aporte de este trabajo, cuyas observaciones críticas contribuyeron significativamente a su mejora. Esta dedicatoria es un tributo a todos los elementos que se entrelazaron para hacer posible este logro. Nuestro trabajo no solo representa nuestro esfuerzo, sino también el resultado del apoyo inquebrantable de quienes creyeron en nosotros y en nuestra visión.

Santiago Alexander Román Silva

Jaime Oswaldo Paqui Medina

Agradecimiento

Agradecemos de corazón a todos quienes hicieron posible esta investigación y proyecto de tesis. Sus valiosas contribuciones y apoyo fueron fundamentales para este hito. A nuestras familias y seres queridos, fuente inagotable de apoyo y motivación. Sus palabras nos impulsaron a superar desafíos y perseverar en la búsqueda de conocimiento. Expresamos gratitud a los participantes de ensayos y encuestas. Sus aportes validaron y mejoraron resultados, fortaleciendo nuestro análisis. Reconocemos a fuentes bibliográficas y recursos académicos consultados. Su contribución enriqueció nuestra comprensión e investigación. Este trabajo no habría sido posible sin la colaboración y generosidad de todos mencionados. A todos, nuestro sincero agradecimiento por ser parte de este viaje y su contribución al éxito.

Santiago Alexander Román Silva

Jaime Oswaldo Paqui Medina

Índice de Contenidos

Portada.....	i
Certificación de director.....	ii
Autoría	iii
Carta de autorización del estudiante.....	iv
Dedicatoria	v
Agradecimiento.....	vi
Índice de Contenidos.....	vii
1. Título.....	1
2. Resumen	2
3. Introducción	4
4. Marco Teórico	6
4.1. Plagio	6
4.1.1. Tipos de Plagio.....	6
4.2. Redes Neuronales	7
4.2.1. Modelo estándar de neurona artificial	8
4.2.2. Arquitectura de una Red Neuronal.....	9
4.2.3. Ventajas de las Redes Neuronales.....	10
4.2.4. Aplicaciones con Redes Neuronales en el Procesamiento de Lenguaje Natural....	11
4.2.5. Tipos de Redes Neuronales en el Procesamiento de Lenguaje Natural (NLP).	12
4.2.6. Tabla Descriptiva de los tipos de redes neuronales	16
4.3. Modelo Latent Dirichlet Allocation (LDA).....	18
4.4. Procesamiento de Lenguaje Natural.....	18
4.4.1. Técnicas para el Procesamiento de Lenguaje Natural.	18
4.5. Tecnologías para el Desarrollo de Software.	19
4.5.1. Python.	19
4.5.2. Django	20
4.5.3. StarUML	20
4.5.4. Visual Studio Code	20
4.5.5. Lucidchart.....	21
4.5.6. Mysql.....	21
4.6. Metodología XP	21
4.6.1. Características XP	21
4.6.2. Proceso XP.....	22
4.6.3. Planeación.....	22

4.6.4. Diseño	23
4.6.5. Codificación	23
4.6.6. Pruebas	23
4.7. Modelo TAM	24
4.8. Modelo de Software 4+1	24
4.8.1. Vista Lógica	24
4.8.2. Vista de Desarrollo	24
4.8.3. Vista de Procesos	25
4.8.4. Vista Física	25
4.8.5. Vista de Escenarios	25
4.9. Librerías y Frameworks	25
4.9.1. Sckitilearn	25
4.9.2. Keras	25
4.9.3. Tensorflow	26
4.9.4. Spacy	26
4.9.5. Otras librerías Adicionales	26
4.10. Descripción de las Librerías para trabajar con Redes Neuronales	27
4.11. Trabajos Relacionados	30
4.11.1. TR1: Detección de plagio en documentos árabes: Enfoques, arquitectura y Sistemas.	30
4.11.2. TR2: Creación de un prototipo de sistema de detección de plagio con NLTK. 30	30
4.11.3. ... TR3: Diseño e implementación de un sistema de detección de Plagio mediante el método de similitud del coseno.	31
4.11.4. TR4: Diseño e Implementación de una herramienta en Studium para evitar los plagios en asignaturas de Ingeniería.	31
5. Metodología	32
5.1. Área de Estudio	32
5.2. Procedimiento.....	32
5.3. Métodos.....	34
5.4. Técnicas	34
5.5. Estándar	34
5.6. Materiales.....	34
6. Resultados	37
6.1. Objetivo 1: Diseñar el modelo de control anti plagio, mediante redes neuronales con Scikit-Learn.	37
Actividad 1.1: Diseñar el modelo mediante el lenguaje Python que marcara el marco de trabajo para el software anti plagio.	37

Actividad 1.2: Utilizar redes neuronales con Scikit-Learn para identificar coincidencias dentro los textos.....	45
6.2. Objetivo 2: Desarrollar el software anti plagio utilizando Django, aplicando el modelo de software 4+1.....	56
Actividad 1.1: Especificación de Requisitos de Software.....	56
Actividad 1.2: Diseño del Software anti plagio, aplicando el modelo 4 + 1.....	58
Actividad 1.3: Desarrollo del Software anti plagio, utilizando el framework Django.....	66
Actividad 1.4: Generar pruebas Funcionales del prototipo de software Antiplagio.....	69
Actividad 1.5: Generar pruebas de Aceptación.....	71
7. Discusión	77
8. Conclusiones	80
9. Recomendaciones	81
10. Bibliografía.....	83
11. Anexos	88

Índice de Tablas:

Tabla 1. Tabla Descriptiva [Fuente propia].	16
Tabla 2. Descripción de Librerías [Fuente propia].	27
Tabla 3. Recursos Humanos.	34
Tabla 4. Recursos de Software.	35
Tabla 5. Recursos de Hardware.	35
Tabla 6. Insumos.	36
Tabla 7. Casos de Prueba para el modelo.	43
Tabla 8. Casos de Prueba con red neuronal siamesa.	55
Tabla 9. Requisitos Funcionales.	56
Tabla 10. Requisitos No Funcionales.	57
Tabla 11. Casos de Pruebas Funcionales con el Prototipo de Software Antiplagio.	69
Tabla 12. Casos de Pruebas Funcionales con el Software PlagarismDetector.net.	70
Tabla 13. Casos de Prueba PlagarismDetector.net y Prototipo de Software Antiplagio.	70
Tabla 14. Tabla de preguntas para el modelo TAM.	72
Tabla 15. Valoraciones numéricas para cada respuesta escogida por el usuario.	73
Tabla 16. Cálculo del promedio final para la Aceptación en base a las variables FU y UP. 73	
Tabla 17. Cálculo del promedio final para el Rechazo en base a las variables FU y UP.	74
Tabla 18. Análisis de resultados en base a cada variable de medición.	74
Tabla A3 1. Entrevista.	95
Tabla A5 1. Proceso de planificación para cada iteración de la metodología XP en función de las historias de usuario.	140

Índice de Figuras:

Figura 1.	Proceso global en el sistema de una red neuronal.	8
Figura 2.	Modelo de neurona artificial standard.	9
Figura 3.	Arquitectura de una red Neuronal.	9
Figura 4.	Función sigmoidea.....	10
Figura 5.	Arquitectura de la Red Neuronal Recurrente.	12
Figura 6.	Arquitectura de la Red Neuronal Convolutiva.....	13
Figura 7.	Arquitectura de la Red Neuronal Transformer.....	14
Figura 8.	Arquitectura de la Red Neuronal GAN.	15
Figura 9.	Arquitectura de la Red Neuronal Siamesa.	16
Figura 10.	Representación gráfica del modelo LDA.....	18
Figura 11.	Proceso de la Metodología XP.	22
Figura 12.	Componentes Generales del detector de plagio [Fuente propia].....	38
Figura 13.	Función para la Limpieza de Archivos.	39
Figura 14.	Función para los Tópicos del Texto.	40
Figura 15.	Función para entrenar el modelo LDA.	41
Figura 16.	Función para Obtener la similitud del Coseno de 2 textos.	41
Figura 17.	Función para obtener plagio de Internet.	42
Figura 18.	Arquitectura de la Red Neuronal Siamesa [Fuente propia].	46
Figura 19.	DSpace de la Universidad Nacional de Loja.	47
Figura 20.	Biblioteca digital IEEE Xplore.	48
Figura 21.	Herramienta Nitro.	48
Figura 22.	Herramienta web DocTranslator.	49
Figura 23.	Herramienta web QuillBot.....	49
Figura 24.	Función para generar pares de textos.	50
Figura 25.	Función para la lectura de pares de textos.	51
Figura 26.	Función para generar el modelo entrenado.	52
Figura 27.	Función para calcular la similitud.	53
Figura 28.	Similitud usando el método del coseno.....	54
Figura 29.	Documento con los pares textos.....	54
Figura 30.	Diagrama de Clases	59
Figura 31.	Diagrama de secuencia.	60
Figura 32.	Diagrama de paquetes.	61
Figura 33.	Diagrama de actividades.	62
Figura 34.	Diagrama de despliegue.....	63
Figura 35.	Diagrama de caso de uso Analizar Plagio.	64

Figura 36.	Modelo relacional.	65
Figura 37.	Modelo Documento.....	66
Figura 38.	Vista para subir un documento.	67
Figura 39.	Plantilla para subir un documento.	68
Figura 40.	Interfaz subir un documento.	68
Figura 41.	Resultado de Facilidad de uso Percibida	75
Figura 42.	Resultado de Utilidad Percibida.....	76

Índice de Anexos:

Anexo 1. Descripción de las funciones principales para el análisis en documentos.	88
Anexo 2. Descripción de las funciones principales de la implementación de la red neuronal siamesa.	91
Anexo 3. Entrevista	95
Anexo 4. Especificación de requisitos de software según el estándar IEEE830.	97
Anexo 5. Interfases de usuario de los Requerimientos Funcionales.....	121
Anexo 6. Desarrollo del Objetivo 2.	140
Anexo 7. Plan de Pruebas Unitarias.....	157
Anexo 8. Plan de Pruebas Funcionales.....	164
Anexo 9. Manual de usuario del Administrador.	187
Anexo 10. Manual de usuario del Estudiante.	201
Anexo 11. Manual de usuario del Docente.....	220
Anexo 12. Evaluación de la experiencia de usuario en el uso del sistema Antiplagio	241
Anexo 13. Formulario con las respectivas preguntas, para realizar la encuesta correspondiente.	247
Anexo 14. Documento de Arquitectura de software con el modelo 4+1.....	250
Anexo 15. Certificación de traducción del resumen.....	277

1. Título

Prototipo de Software Antiplagio para la Carrera de Ingeniería en Computación de la Universidad Nacional de Loja, utilizando Redes Neuronales.

2. Resumen

Detectar el plagio es esencial para evitar el robo de conceptos y contenidos. Hoy en día existen muchas herramientas que permiten evitar el plagio, pero la mayoría son de pago y su acceso es limitado, razón por la cual son pocos los usuarios que los adquieren. Es por ello que este Trabajo de Integración Curricular (TIC) se ha llevado a cabo con el propósito de desarrollar un prototipo de software destinado a abordar la problemática de: ¿Como controlar el plagio mediante el desarrollo de un software Antiplagio para la carrera de Computación en la Universidad Nacional de Loja (UNL), mediante redes neuronales? Para abordar esta cuestión, se planteó dos objetivos específicos, llevados a cabo utilizando la metodología XP, que consta de cuatro fases: planificación, diseño, codificación y pruebas. Este TIC inicia con el primer objetivo sobre diseñar el modelo de control Antiplagio mediante redes neuronales, se tomó como punto de partida el trabajo relacionado TR2, dando como resultado la implementación del modelo. Para cumplir con el segundo objetivo de desarrollar el software Antiplagio utilizando el framework Django y aplicando el modelo 4 + 1, se utilizó la especificación de requisitos basada en el estándar IEEE830 junto con la metodología XP, obteniendo como resultado final el software Antiplagio. También se elaboró pruebas funcionales y pruebas de aceptación mediante el modelo TAM, con la finalidad de garantizar el correcto funcionamiento del software Antiplagio y que se cumpla con los requisitos funcionales establecidos, además de la elaboración de manuales de usuario para cada rol. De este modo se pudo dar respuesta a la pregunta de investigación, confirmando que mediante el software desarrollado se puede controlar plagio en los trabajos TIC, ya que este arroja resultados satisfactorios.

Palabras claves: Redes Neuronales, modelo TAM, IEEE830, Metodología XP, framework Django, modelo 4 + 1.

Abstract

Detecting and identify the plagiarism is essential work to avoid the theft of concepts and content. Today there are many tools that allow you to avoid plagiarism, but most are paid and their access is limited, for this motive few users purchase them. For this reason, this Information and Communication Technologies (ICT) has been carried out with the purpose of creating a software prototype aimed at solving the problem, ¿How to control plagiarism by developing anti-plagiarism software for the Computer Science degree at the National University of Loja (UNL), using neural networks? To address this problem, two specific objectives were set, carried out using the XP methodology, which consists of four phases: planning, design, coding and testing. This (ICT) Information and Communication Technologies begin with the first objective of designing the Anti-plagiarism control model using neural networks, therefore the related work TR2 was taken as a starting point, resulting in the implementation of the model. To comply with the second objective for the development of Anti-plagiarism software anti- plagiarism using the Django framework and applying the 4 + 1 model, it was decided to use the requirements specification based on the IEEE830 standard together with the XP methodology, obtaining as a final result the Anti-plagiarism software. Functional tests and acceptance tests were also carried out using the TAM model (technology acceptance model), in order to guarantee the correct functioning of the Anti-plagiarism software and that the established functional requirements are met, in addition to the preparation of user manuals for each role. This way the research question could be answered, confirming that the developed software can control plagiarism in PIC (curricular integration projects), since it way satisfactory results.

Keywords: Neural Networks, TAM model, IEEE830, XP Methodology, Django framework, 4 + 1 model.

3. Introducción

Entre las herramientas que pueden ayudar a mejorar la probidad académica de los estudiantes, aún desde los años pre-universitarios, se encuentran los software o herramientas "Antiplagio". No obstante, las principales herramientas Antiplagio son comerciales y necesitan de un pago anticipado, estas son: Turnitin, Urkund, Grammar, iThenticate, entre otras reduciendo así a las personas que tienen acceso a estas herramientas para detectar y corregir el problema del plagio académico [1].

En una encuesta aplicada a 10 docentes de la carrera de Computación de la Universidad Nacional de Loja (véase **Anexo 13**), se obtuvo resultados de 8 individuos, donde se indica que el 100% de los docentes encuestados manifiestan que sería muy beneficioso que se cree un software Antiplagio de la carrera, ya que este tipo de herramientas ayudaría a: "Garantizar que se respete la propiedad intelectual del autor, Facilidad para la revisión de tareas y Trabajos de Titulación, Evitar el exceso de plagio, Mejorar la calidad de los trabajos de docentes y alumnos en la parte de investigación, Evitar problemas legales dentro de este contexto".

Partiendo de este contexto, es necesario disponer de una herramienta de esta naturaleza, la cual sea fácil de utilizar, funcione correctamente, sea gratuito, ya que la mayoría de software que detectan plagio con resultados aceptables son de pago y los de acceso gratuito no realizan de manera correcta la revisión. Entonces a través de esta iniciativa, se evitará la prevención del plagio, generando resultados aceptables.

El objetivo de este proyecto es el desarrollo de un prototipo de software Antiplagio basado en redes neuronales para la carrera de Ingeniería en Computación de la Universidad Nacional de Loja. Se trata de una propuesta que busca, la disponibilidad de una herramienta de fácil acceso, de uso gratuito, ayude a contribuir la prevención del plagio académico en Proyectos de Integración Curricular (PIC), y que a su vez sea propio de la carrera. Para ello, fue necesario realizar un análisis a fin de diseñar un prototipo que sea útil en la detección del plagio.

En este trabajo se desarrolló un prototipo de software diseñado para detección del plagio. Por lo tanto, el Trabajo está estructurado de la siguiente manera. En la sección de Marco Teórico, se aborda la base teórica completa junto con los trabajos relacionados. Luego, se expone la Metodología, en la cual se detalla el campo de estudio y se describe el proceso que engloba las actividades del TIC, así como los métodos, técnicas y recursos humanos, además de los elementos de software, hardware e insumos utilizados. Dentro de la sección de Resultados se documenta la evidencia del avance del TIC, organizándose en actividades y subactividades correspondientes a los objetivos específicos siguientes: Diseñar el modelo

de control anti plagio, mediante redes Neuronales [2], y Desarrollar el software anti plagio utilizando el framework Django [3], aplicando el Modelo de software 4+1 [4]. En la sección de Discusión, lo autores del trabajo exponemos la evaluación fundamentada en los resultados obtenidos. Posteriormente, en la sección de Conclusiones, se resaltan las principales contribuciones surgidos tanto de los resultados como del análisis realizado en todo el proceso del proyecto. Por último, la sección de Recomendaciones se exponen sugerencias relevantes para el tema tratado y sirven como marco para trabajos futuros o avances en el campo de estudio.

4. Marco Teórico

4.1. Plagio

El plagio es una infracción de derechos de autor de cualquier trabajo producido al copiar ese trabajo sin el permiso de la persona que creó el trabajo, se puede cometer plagio de forma deliberada o de manera inconsciente. Definiciones y situaciones comunes en las cuales se comete este delito:

- El plagio ocurre cuando se toman ideas o palabras escritas por otros sin reconocer de forma directa el haberlo hecho
- Se produce al presentar como propio un trabajo de forma parcial o total sin ser el autor de dicho trabajo.
- Se considera que se comete plagio al copiar cualquier objeto de fondo, forma o incluso una simple frase [5].

4.1.1. Tipos de Plagio

Hay diversos tipos de plagio y diferentes formas de describirlos. Estos se clasifican en tres categorías y se explican los tipos de plagio que existen en cada una. Estas categorías se basan en la forma en que se llevan a cabo, el método utilizado y el propósito detrás de cometerlo [6].

a. Tipos de plagio por su forma

- **Auto-Plagio:** Ocurre cuando un autor reutiliza sus propios escritos y los presenta como una obra inédita u original, sin citar ni referenciar sus propias publicaciones anteriores. Se destaca que simplemente agregar una referencia no es suficiente, ya que no se informa al lector ni al editor sobre el alcance de la copia. Se plantea que el término puede generar un debate ético en el ámbito académico y se relaciona con el concepto de duplicidad. El problema se enfoca más en el aspecto ético que en el legal, ya que el autor presenta una publicación como un resultado de investigación original y no informa al editor o lector sobre la cantidad o magnitud de la copia, además de no referenciar la obra previamente publicada [7].

- **Falsa Autoría:** La falsa autoría en el contexto del plagio se refiere a la acción de atribuir erróneamente el nombre de una persona como el autor de un trabajo, como un artículo o ensayo, sin que esa persona haya realizado una contribución real en su creación. Esto implica engañar o falsificar la verdadera autoría del trabajo con el propósito de obtener reconocimiento, crédito o beneficios indebidos. Puede involucrar pagar a

alguien para que escriba el trabajo y luego presentarlo como propio, o usurpar el nombre de otra persona sin su conocimiento o consentimiento [8].

- **Envío Doble:** es una modalidad de plagio en la que se envía y publica un mismo trabajo o resultados de investigación en varios lugares, sin reconocer ni revelar adecuadamente la existencia de la duplicidad. En términos simples, implica presentar un trabajo idéntico como si fuera original y novedoso en múltiples publicaciones, sin mencionar que ya ha sido previamente publicado o presentado en otro sitio [9].

- **Robo de Material:** El robo de material dentro del plagio se refiere a la acción de tomar contenido de otras personas sin su autorización y sin otorgarles el crédito correspondiente. Esta forma de plagio implica apropiarse de ideas, palabras, frases, párrafos u cualquier otra forma de expresión creativa sin permiso ni reconocimiento adecuado al autor original [10].

b. Tipos de plagio mediante métodos.

- **Copiar y Pegar:** es una forma común de apropiación indebida de contenido. Consiste en copiar literalmente secciones de un texto original, ya sea en formato digital o impreso, y pegarlos en otro lugar sin realizar cambios. El copiar y pegar sin citar correctamente las fuentes o sin realizar modificaciones significativas representa una violación de los derechos de autor y de los principios éticos en el ámbito académico [5].

- **Parfraseo inapropiado:** El plagio inapropiado se caracteriza por la ausencia de integridad académica y ética, al hacer pasar como propio lo que en realidad pertenece a otra persona. Esta conducta transgrede los fundamentos de originalidad, honestidad y respeto a la propiedad intelectual. Además, ocasiona perjuicio tanto a los autores originales como a la comunidad académica en su conjunto, al distorsionar la justa distribución del reconocimiento y los méritos por las ideas y el trabajo intelectual [7].

- **Referencia falsa:** La referencia falsa en el contexto del plagio implica la inclusión de citas o referencias bibliográficas que no se corresponden con las fuentes reales utilizadas en un trabajo académico. Esta conducta deshonesto implica suministrar información engañosa o inventada sobre las fuentes citadas con el propósito de dar la apariencia de respaldo académico y validar las ideas expuestas [10].

4.2. Redes Neuronales

El concepto de red neuronal se utiliza para describir a una serie de modelos relacionados que comparten características similares y están definidos por un amplio rango de parámetros y una estructura flexible. Estos modelos surgieron a partir de investigaciones sobre el funcionamiento del cerebro, aunque con el tiempo se han desarrollado numerosos

modelos no biológicos. A pesar de esto, muchos términos utilizados en este contexto aún reflejan su origen biológico [2]. Una red neuronal se compone de unidades elementales de procesamiento (PE) que están interconectadas de una manera específica. El valor de las redes neuronales artificiales no solo radica en el modelo individual de cada elemento PE, sino también en la forma en que se conectan estos elementos procesadores. Por lo general, los PE se agrupan en niveles o capas, y una red neuronal se construye con una secuencia de capas que están conectadas entre sí de manera consecutiva [11].

4.2.1. Modelo estándar de neurona artificial

El modelo estándar de una neurona artificial basado en los principios establecidos en los trabajos de Rumelhart y McClelland (1986). Siguiendo estos principios, la neurona artificial estándar en su i -ésima forma se compone de

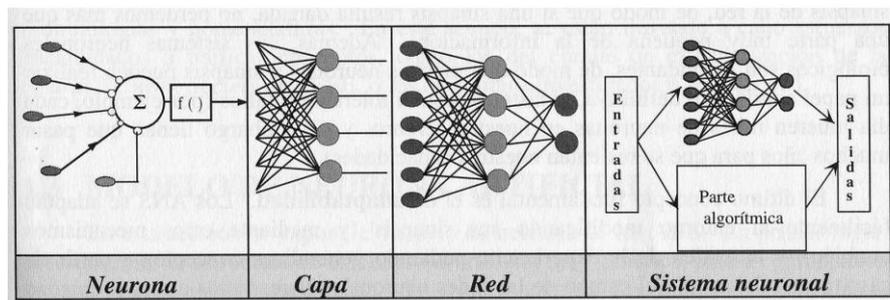


Figura 1. Proceso global en el sistema de una red neuronal [11].

- Se tiene un **conjunto de entradas** x_j y unos pesos sinápticos w_{ij} , donde j varía de 1 a n .
- También se cuenta con una **regla de propagación** h_i que se define utilizando dicho conjunto de entradas y pesos sinápticos. En resumen, la regla de propagación consiste en sumar el producto de cada entrada con su respectivo peso sináptico.
- Una **función de activación** se utiliza para representar tanto la salida de la neurona como su estado de activación. En resumen, la función de activación se aplica a la suma ponderada de las entradas y los pesos sinápticos, representada por h_i , para obtener la salida y el estado de activación de la neurona. En la figura 2 se presenta el modelo estándar de una neurona artificial descrito anteriormente.

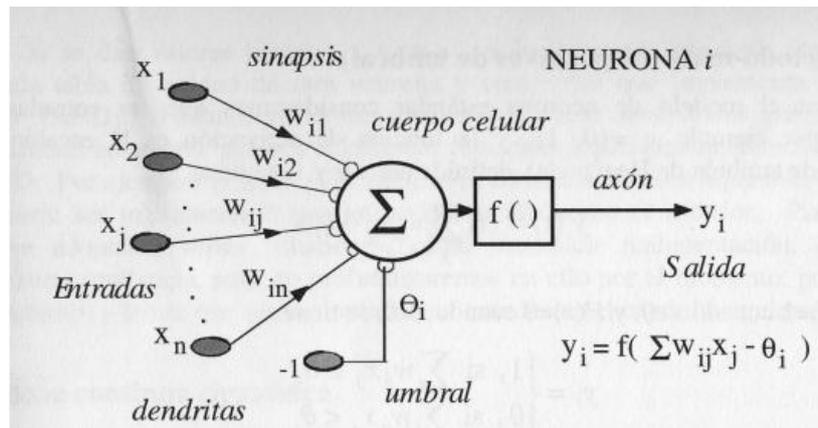


Figura 2. Modelo de neurona artificial standard [2].

4.2.2. Arquitectura de una Red Neuronal

La arquitectura de una red neuronal se refiere a la topología, estructura o patrón de conexiones de la red. En una red neuronal artificial, los nodos se conectan mediante sinapsis, y el comportamiento de la red está determinado por la estructura de estas conexiones sinápticas. Estas conexiones son direccionales, lo que significa que la información solo puede propagarse en un solo sentido, desde la neurona presináptica a la posináptica. En general, las neuronas se agrupan en unidades estructurales llamadas capas [13]. La red neuronal está formada por una o varias capas, y la forma en que estas capas están conectadas define la arquitectura de la red. En resumen, la arquitectura de una red neuronal se refiere a cómo están conectados los nodos mediante sinapsis, con la propagación de información en una dirección específica. Las neuronas se agrupan en capas, y el conjunto de estas capas constituye la red neuronal [14].

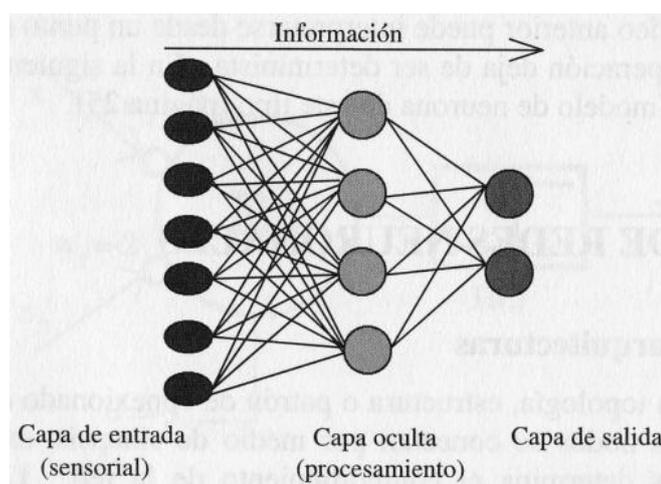


Figura 3. Arquitectura de una red Neuronal [14].

- a. **Capa Entrada:** Esta capa es la entrada de la red neuronal y representa los datos de entrada que se utilizarán para realizar la tarea específica. Por lo general, los datos de

entrada se representan como un vector numérico donde cada elemento del vector corresponde a una característica o atributo del dato. Cada elemento del vector se conecta con los nodos de la capa de entrada.

- b. Capa Oculta:** Es el componente clave en una red neuronal, situada entre la capa de entrada y la capa de salida. En una red tÍTICa, puede haber una o varias capas ocultas, compuestas por nodos o neuronas interconectadas. Cada nodo en una capa oculta está conectado con todos los nodos de la capa anterior y la siguiente, a través de conexiones conocidas como "pesos" [15].
- c. Capa Salida:** Esta capa es la última capa de la red neuronal y representa la salida de la red. Dependiendo del tipo de problema que se esté abordando, la capa de salida puede tener uno o varios nodos. Cada nodo en la capa de salida proporciona una estimación o predicción para una determinada variable objetivo o clase. La función de activación en la capa de salida puede variar según el tipo de problema, como una función sigmoide para problemas de clasificación binaria o una función softmax para problemas de clasificación multiclase [16].
- d. Función sigmoidea:** La función sigmoidea da valores que oscilan entre 0 y 1. La pendiente de la función de activación se altera al cambiar el valor de la entrada g . A continuación, se tiene la fórmula de la función sigmoidea [17].

$$f(x) = \frac{1}{1 + e^{-gx}}, \text{ con } x = gin_i - \Theta_i.$$

Figura 4. Función sigmoidea [17].

4.2.3. Ventajas de las Redes Neuronales

Las redes neuronales artificiales comparten muchas similitudes con el cerebro debido a su estructura y principios fundamentales. Por ejemplo, tienen la capacidad de aprender a partir de la experiencia, de generalizar conocimientos de casos previos a nuevos casos y de identificar características esenciales incluso en entradas que contienen información irrelevante [18].

- **Aprendizaje Adaptativo:** Las redes neuronales son capaces de aprender y adaptarse automáticamente a partir de los datos de entrada. Pueden ajustar sus pesos y conexiones internas para mejorar su rendimiento con el tiempo.
- **Capacidad de generalización:** Las redes neuronales pueden generalizar patrones y conocimientos aprendidos de casos anteriores a nuevos casos o situaciones similares.

Esto les permite hacer predicciones y tomar decisiones en escenarios no vistos anteriormente.

- **Procesamiento paralelo:** Las redes neuronales pueden realizar múltiples cálculos simultáneamente, lo que les permite procesar grandes cantidades de información de manera rápida y eficiente. Esto es especialmente útil para tareas de procesamiento de imágenes, reconocimiento de voz y procesamiento de datos masivos.
- **Adaptabilidad y flexibilidad:** Las redes neuronales pueden adaptarse a cambios en los datos de entrada y ajustarse a nuevas condiciones. Son capaces de aprender de forma continua y actualizarse para mantener un rendimiento óptimo.
- **Manejo de datos complejos:** Las redes neuronales pueden trabajar con datos no lineales y de alta dimensionalidad, lo que las hace adecuadas para abordar problemas complejos y extraer patrones ocultos en conjuntos de datos complejos.
- **Tolerancia a fallos y robustez:** Las redes neuronales tienen la capacidad de recuperarse y mantener su rendimiento incluso si algunos de sus componentes o conexiones se dañan o se pierden. Esto las hace resistentes a fallos y más confiables en entornos ruidosos o cambiantes [19].

4.2.4. Aplicaciones con Redes Neuronales en el Procesamiento de Lenguaje Natural.

Dentro del procesamiento de textos, las redes neuronales tienen varias aplicaciones que mejoran la comprensión y el análisis del contenido textual. Algunas de estas aplicaciones son las siguientes: [20]

- ❖ **Clasificación de texto:** Las redes neuronales se utilizan para clasificar textos en categorías específicas, como spam vs. no spam, sentimientos positivos vs. negativos, o temas específicos. Estas redes pueden aprender automáticamente patrones y características relevantes del texto para realizar una clasificación precisa [21].
- ❖ **Extracción de información:** Las redes neuronales pueden ser utilizadas para extraer información específica de textos, como nombres de personas, ubicaciones, fechas o eventos. Estas redes pueden aprender a identificar y etiquetar entidades y relaciones relevantes en el texto [22].
- ❖ **Traducción automática:** Las redes neuronales son ampliamente utilizadas en sistemas de traducción automática. Estas redes pueden aprender a traducir texto de un idioma a otro, capturando las sutilezas y las estructuras gramaticales de los diferentes idiomas para producir traducciones más precisas y naturales [23].
- ❖ **Generación de texto:** Las redes neuronales se utilizan para generar texto coherente y relevante en diversas aplicaciones. Pueden utilizarse para generar respuestas a

preguntas, resúmenes de textos, diálogos conversacionales o incluso para la creación de contenido generado por IA [24].

- ❖ **Modelado de lenguaje:** Las redes neuronales se utilizan para modelar el lenguaje y predecir palabras o secuencias de palabras en un contexto dado. Estos modelos de lenguaje pueden ser utilizados para la corrección gramatical, la autocorrección de texto, la sugerencia de palabras mientras se escribe, entre otros.
- ❖ **Análisis de sentimientos:** Las redes neuronales se utilizan para analizar el sentimiento expresado en un texto. Pueden clasificar si un texto tiene un sentimiento positivo, negativo o neutro, lo que es útil en aplicaciones como la minería de opiniones, la evaluación de comentarios de usuarios, etc.
- ❖ **Reconocimiento de voz:** Las redes neuronales también se utilizan en el reconocimiento de voz, donde se convierten las señales de audio en texto escrito. Estas redes pueden aprender a reconocer patrones de habla y convertirlos en texto comprensible [25].

4.2.5. Tipos de Redes Neuronales en el Procesamiento de Lenguaje Natural (NLP).

a. Redes Neuronales Recurrentes

Las Redes Neuronales Recurrentes (RNRs), también conocidas como Recurrent Neural Networks (RNN) en inglés, son modelos de redes neuronales artificiales (RNAs) en los cuales las conexiones entre las unidades están organizadas en forma de un ciclo dirigido [26]. Son especialmente adecuadas para el procesamiento de secuencias de texto, ya que pueden modelar la dependencia temporal entre las palabras. Son capaces de recordar información contextual pasada y utilizarla en el procesamiento de las palabras siguientes. Algunas variantes populares de RNN incluyen las LSTM (Long Short-Term Memory) y las GRU (Gated Recurrent Unit) [27].

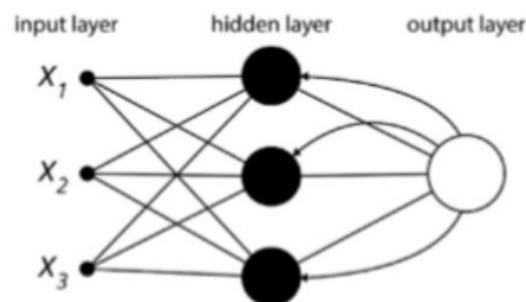


Figura 5. Arquitectura de la Red Neuronal Recurrente [27].

b. Redes neuronales convolucionales (CNN)

Las Redes Neuronales Convolucionales (CNNs), también conocidas como Convolutional Neural Networks (CNN) en inglés, son modelos ampliamente utilizados en el procesamiento de imágenes y la visión por computadora. Están diseñados de manera que imitan la estructura de la corteza visual de los animales [28]. Aunque las CNN se asocian comúnmente con el procesamiento de imágenes, también se utilizan en el procesamiento de texto. Las CNN también son útiles para tareas como la clasificación de texto, la extracción de información y la generación de texto [29].

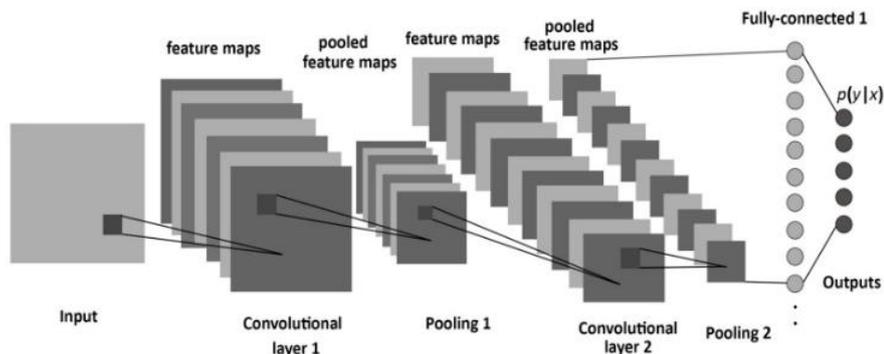


Figura 6. Arquitectura de la Red Neuronal Convolutiva [29].

c. Redes neuronales transformer

Las Redes Neuronales Transformer, o simplemente Transformers, fueron introducidas por primera vez en 2017 en un artículo llamado "Attention is All You Need". Los Transformers se basan en el mecanismo de atención, que permite a la red aprender y capturar relaciones entre diferentes partes de una secuencia de entrada [30]. Las redes transformer han revolucionado el procesamiento de lenguaje natural. Se basan en un mecanismo de atención que permite a la red capturar relaciones de largo alcance en las secuencias de texto. Han demostrado un rendimiento destacado en tareas como la traducción automática, la generación de texto y la comprensión del lenguaje natural [31].

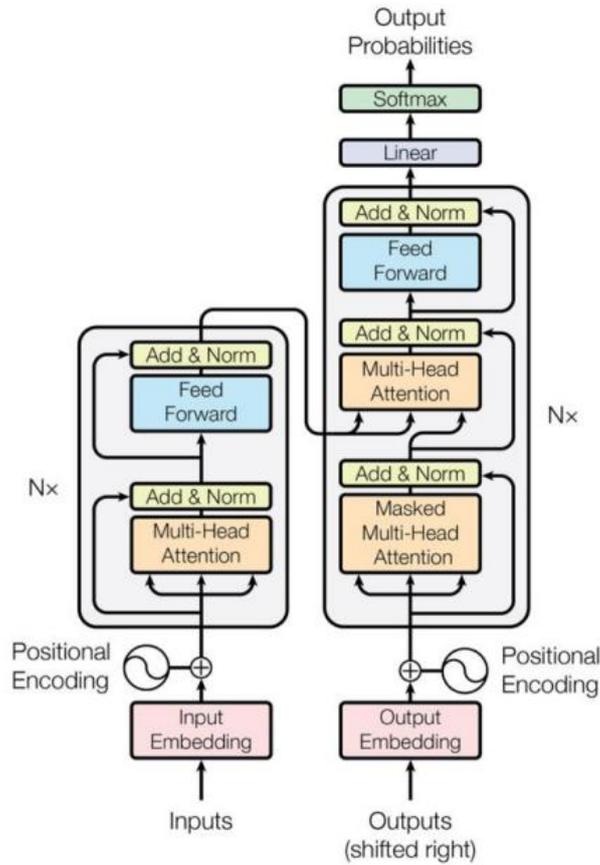


Figura 7. Arquitectura de la Red Neuronal Transformer [31].

d. Redes neuronales generativas (GAN)

Las Redes Generativas Antagónicas (GANs), conocidas como Generative Adversarial Networks en inglés, fueron presentadas en 2014 por Ian J. Goodfellow y su equipo en el artículo "Generative Adversarial Nets" [32]. Las GAN se utilizan para generar texto coherente y realista. Estas redes constan de un generador que produce muestras de texto y un discriminador que intenta distinguir entre textos reales y generados. Las GAN se aplican en la generación de texto creativo, el diálogo y la mejora de la fluidez y coherencia del lenguaje [33].

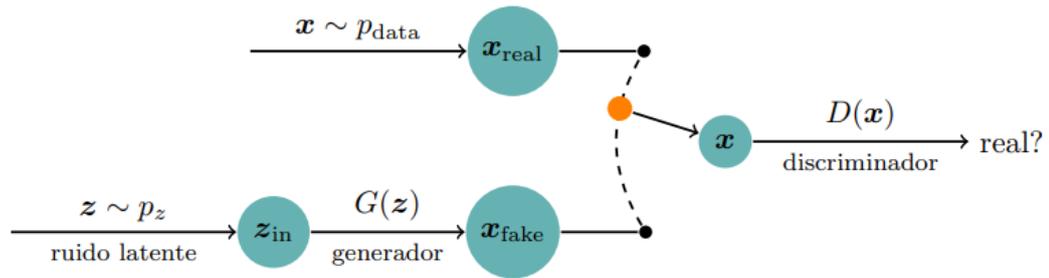


Figura 8. Arquitectura de la Red Neuronal GAN [33].

e. Redes neuronales Siamesas (SNNs)

Una red neuronal siamesa es un tipo de arquitectura de red neuronal utilizada en problemas de similitud o comparación de datos. Consiste en varias ramas o subredes idénticas que procesan las entradas de forma independiente y extraen características relevantes. Estas características se combinan en una capa de fusión para determinar la similitud entre los datos [34]. La salida de la red siamesa se utiliza para tareas como clasificación, reconocimiento facial o detección de anomalías. Esta arquitectura permite compartir conocimientos y aprender representaciones útiles de los datos, lo que puede mejorar el rendimiento y la generalización en comparación con otras estructuras de red neuronal [35].

Arquitectura General de la Red Neuronal Siamesa.

Se compone de dos redes neuronales idénticas que descubre la representación vectorial oculta de entrada. La similitud semántica de las representaciones vectoriales se determina comparando estas redes paralelas al final de la operación. Se alimentan mediante perceptrones y se entrenan mediante retro propagación de errores. La salida de una red neuronal siamesa es un indicador de la similitud de los vectores de entrada [36].

Entonces Sea $\{(x_i, y_i), i = 1, \dots, n\}$ un conjunto de datos que consiste en n vectores de características $x_i \in R^m$ de tamaño m , con etiquetas $y_i \in \{1, 2, \dots, C\}$. Vamos a construir un nuevo conjunto de entrenamiento $S = \{(x_i, x_j, z_{ij}), (i, j) \in K\}$ que consiste en pares de ejemplos x_i y x_j con etiquetas binarias $z_{ij} \in \{0, 1\}$ asignadas a ellos. Si ambos vectores de características x_i y x_j son semánticamente similares, es decir, pertenecen a la misma clase, entonces z_{ij} es 0. Si los vectores son semánticamente diferentes, es decir, corresponden a diferentes clases, entonces z_{ij} es 1. Por lo tanto, el conjunto de entrenamiento S se puede dividir en dos subconjuntos: un conjunto similar o positivo con $z_{ij} = 0$ y un conjunto disimilar o negativo con $z_{ij} = 1$. Supongamos que las nuevas representaciones de características de los ejemplos de entrada x_i y x_j son las salidas de la red neuronal siamesa (SNN), y se denotan como $h_i \in R^D$ y $h_j \in R^D$, respectivamente. La SNN realiza un mapeo tal que $h_i = f(x_i)$, que

intenta minimizar (maximizar) la distancia euclidiana $d(h_i, h_j)$ para el par de objetos similar (disimilar) tanto como sea posible [37]. La arquitectura estándar de la SNN se muestra en la **Figura 9**.

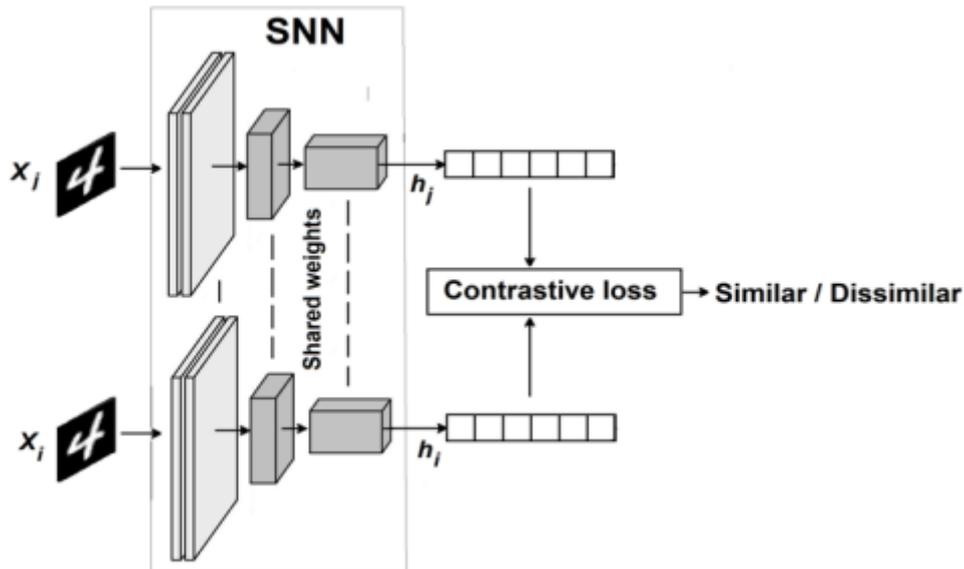


Figura 9. Arquitectura de la Red Neuronal Siamesa [37].

4.2.6. Tabla Descriptiva de los tipos de redes neuronales

En la siguiente tabla descriptiva se resume las características principales de cada tipo de red neuronal utilizada en el procesamiento de lenguaje natural, así como sus ventajas y aplicaciones más comunes. Cada tipo de red neuronal tiene sus propias fortalezas y es adecuado para diferentes tareas dentro del campo del procesamiento de lenguaje natural.

Tabla 1. Tabla Descriptiva [Fuente propia].

Tipo de Red Neuronal	Descripción	Ventajas	Aplicaciones
Redes Neuronales Recurrentes (RNN)	Modelan dependencias temporales en secuencias de texto	<ul style="list-style-type: none"> • Capturan la información contextual y la dependencia temporal • Flexibilidad para procesar secuencias de longitud variable • Útiles en tareas de clasificación y generación de texto 	<ul style="list-style-type: none"> • Traducción automática • Generación de texto • Análisis de sentimiento • Reconocimiento de voz

		<ul style="list-style-type: none"> • Manejan información secuencial y relaciones entre palabras 	
Redes Neuronales Convolucionales (CNN)	Extraen características locales en secuencias de texto	<ul style="list-style-type: none"> • Identifican patrones y características relevantes • Reducción de la dimensionalidad y procesamiento eficiente • Útiles para el análisis de texto basado en n-gramas • Adecuadas para tareas de clasificación y detección de texto 	<ul style="list-style-type: none"> • Extracción de información • Generación de texto • Resumen automático
Redes Neuronales Transformer	Modelan la relación entre palabras a través de la atención	<ul style="list-style-type: none"> • Capturan relaciones de largo alcance en secuencias de texto • Permiten una mejor comprensión del contexto global • Mayor capacidad de paralelización en el entrenamiento • Logran resultados sobresalientes en diversas tareas de NLP 	<ul style="list-style-type: none"> • Traducción automática • Generación de texto • Resumen automático • Comprensión del lenguaje natural
Red Neuronal Siamesa	Utiliza una arquitectura compartida para comparación o similitud de texto.	<ul style="list-style-type: none"> • Aprovecha el aprendizaje conjunto de pares de texto • Aprende representaciones de texto más robustas y discriminativas • Eficaz para problemas de clasificación y generación de texto • Puede manejar múltiples idiomas y dominios • Mejora la capacidad de generalización 	<ul style="list-style-type: none"> • Comparación de similitud de texto • Detección de plagio • Extracción de características • Reconocimiento de entidades nombradas • Clasificación de texto

La Red Neuronal Siamesa es una arquitectura especializada en problemas de comparación o similitud de texto. Utiliza dos o más ramas idénticas que comparten parámetros y aprenden representaciones de texto más robustas y discriminativas. Al utilizar el aprendizaje

conjunto de pares de texto, puede capturar características importantes y generar representaciones de texto más eficientes [38]. La Red Neuronal Siamesa es ampliamente utilizada en tareas como la comparación de similitud de texto, detección de plagio, extracción de características, reconocimiento de entidades nombradas y clasificación de texto. Además, su capacidad para manejar múltiples idiomas y dominios la hace muy versátil en diferentes aplicaciones del procesamiento de lenguaje natural [39].

4.3. Modelo Latent Dirichlet Allocation (LDA)

El algoritmo LDA (Latent Dirichlet Allocation), creado en 2003 por D. Blei, A. Ng y M. Jordan, es el primer método de Modelado de Temas. Se basa en variables ocultas que interactúan con datos observados. En este caso, los datos observados son cada uno de los textos y las variables ocultas son los tópicos de cada documento. La figura 10 muestra el modelo, donde cada nodo representa una variable aleatoria y las flechas las dependencias entre variables [40].

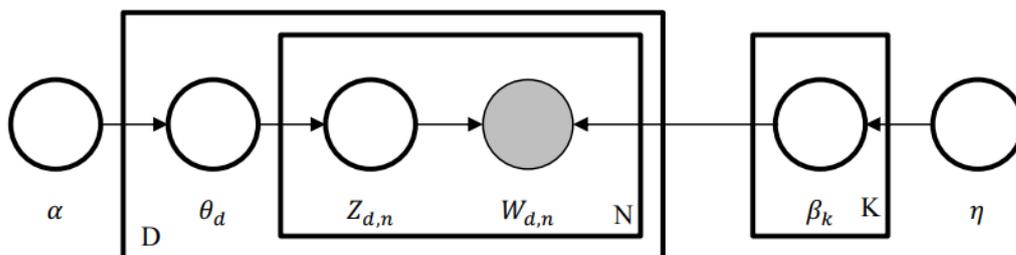


Figura 10. Representación gráfica del modelo LDA [40].

4.4. Procesamiento de Lenguaje Natural.

El Procesamiento de Lenguaje Natural (PLN) es una rama de la inteligencia artificial, su objetivo es hacer que las computadoras puedan entender, analizar y generar lenguaje humano de manera similar a los seres humanos. Para abordar la complejidad y ambigüedad del lenguaje, utiliza técnicas de aprendizaje automático y lingüística computacional [41].

4.4.1. Técnicas para el Procesamiento de Lenguaje Natural.

- **Tokenización**

Es una técnica que divide el texto a procesar según su propósito. Puede lograrse dividiendo el texto en frases o palabras, y estas frases se denominan tokens que forman un arreglo. En el ejemplo presentado, la frase "Hoy vamos a ir a la playa" se separaría en los siguientes tokens mediante la tokenización de palabras: ["Hoy", "vamos", "a", "ir", "a", "la", "playa"]. En las tareas de Procesamiento de Lenguaje Natural (PLN), cada palabra se convierte en un token único, lo que facilita el análisis y procesamiento del texto [42].

- **Normalización**

Esta técnica permite estandarizar y simplificar el texto para hacerlo consistente y más fácil de analizar. Se convierten todas las palabras a minúsculas, se eliminan signos de puntuación y caracteres especiales, y se pueden eliminar palabras vacías.

- **Lematización**

Esta técnica consiste en contar las palabras que se repiten en un diccionario de datos a partir de su lema o raíz lingüística. Devuelve las palabras a su forma base y convierte todas las variantes de una misma palabra a su lema. Por ejemplo, si se desea saber cuántas veces aparece la palabra "cerrado", la lematización contaría las ocurrencias de las palabras "cerrar", "cierra" y "cerrado", todas reducidas a su lema "cerrar" [43].

4.5. Tecnologías para el Desarrollo de Software.

4.5.1. Python.

Python es un lenguaje de programación de propósito general, interpretado, orientado an objetos y de uso generalizado con semántica dinámica. El nombre del lenguaje de programación Python proviene de Monty Python's Flying Circus, una comedia antigua de la BBC. Guido van Rossum nació en 1956 en Haarlem, Países Bajos, y lo creó. Por supuesto, Guido van Rossum no creó ni desarrolló todas las partes de Python. Miles de programadores, testers y usuarios han trabajado arduamente para que Python se haya extendido rápidamente por todo el mundo [44].

a. Por Qué Usar Python

Python se destaca por ser un lenguaje de programación fácil de aprender. La adquisición de habilidades en Python es significativamente más rápida que en muchos otros lenguajes.

- Python es fácil de enseñar: En comparación con otros lenguajes, la enseñanza requiere menos trabajo. Esto les permite a los maestros concentrarse más en enseñar técnicas de programación generales.
- Simple de usar: Se destaca por su facilidad de uso al escribir nuevo software, ya que con frecuencia es posible escribir código de forma más rápida con Python.
- Fácil de entender: Se destaca por su facilidad de comprensión porque el código escrito en Python con frecuencia resulta más fácil de entender.
- Fácil de adquirir: Es un lenguaje de programación gratuito y de código abierto, lo que significa que se puede descargar y usar de forma gratuita [45].

b. Características

- Python utiliza una sintaxis clara y fácil de leer, con un estilo de escritura que se asemeja al inglés que facilita el desarrollo y la colaboración.
- El lenguaje interpretado se ejecuta línea por línea sin compilación previa. Esto aumenta la flexibilidad y permite un ciclo de desarrollo más rápido.
- Tipado dinámico: permite cambiar el tipo de variables que se utilizan.
- Orientado a objetos: La creación de clases, la herencia de clases, la encapsulación de datos y otros conceptos de programación orientada a objetos se pueden realizar con él [46].
- Amplia biblioteca estándar: Python incluye una biblioteca estándar amplia y diversa que ofrece una variedad de módulos y funciones para realizar una variedad de tareas y facilita el desarrollo de aplicaciones.
- La portabilidad es amplia, lo que significa que los programas escritos en Python pueden funcionar en una variedad de sistemas operativos.
- Comunidad de desarrolladores activa: Python tiene una comunidad de desarrolladores muy activa y compasiva. La comunidad está constantemente trabajando para mejorar el lenguaje, y hay muchos recursos, bibliotecas y marcos de trabajo disponibles [47].

4.5.2. Django

Django es un marco de desarrollo web de código abierto basado en Python que permite la creación de aplicaciones web de manera rápida y efectiva. Sigue el patrón Modelo-Vista-Controlador (MVC), es conocido por su legibilidad del código, eficiencia y seguridad, y facilita tareas comunes del desarrollo web. Se utiliza con frecuencia para crear sitios web complejos y funcionales [3].

4.5.3. StarUML

Es una herramienta diseñada para el modelado de software, que se basa en los estándares UML (Lenguaje de Modelado Unificado) y MDA (Arquitectura Dirigida por Modelos) [48].

4.5.4. Visual Studio Code

Editor de código de Microsoft. Software libre y multiplataforma (Windows, GNU/Linux, macOS). Buena integración con Git, soporte de depuración y muchas extensiones para programar en varios lenguajes.

4.5.5. Lucidchart

Es una plataforma de creación de diagramas que permite la colaboración en tiempo real, permitiendo a múltiples usuarios trabajar juntos en un mismo documento.

4.5.6. Mysql

Es un sistema de gestión de bases de datos relacionales de código abierto con un enfoque cliente-servidor que permite crear y administrar bases de datos basadas en un modelo relacional.

4.6. Metodología XP

La programación extrema, también conocida como Extreme Programming (XP), es una metodología de desarrollo de software creada por Kent Beck, quien escribió el primer libro sobre el tema, *Extreme Programming Explained: Embrace Change* (1999) [49]. XP es uno de los enfoques ágiles más reconocidos para el desarrollo de software. Al igual que otros métodos ágiles, la programación extrema se distingue de las metodologías tradicionales al enfocarse más en la capacidad de adaptarse a los cambios en lugar de enfocarse en predecir los resultados [50].

4.6.1. Características XP

- Se implementa de forma dinámica a lo largo del ciclo de vida del software y tiene la capacidad de adaptarse a los cambios en los requisitos.
- Se enfatiza la importancia de los individuos y las interacciones por encima de los procesos y herramientas.
- Propone un enfoque iterativo en el que se siguen cuatro pasos: planificación, diseño, codificación y prueba.
- En cada iteración, se agregan nuevas funcionalidades al software. Esta metodología es especialmente adecuada para Trabajos con requisitos ambiguos y en constante cambio, así como para aquellos con un alto riesgo técnico [50].

4.6.2. Proceso XP

La programación extrema se basa en la orientación a objetos como el paradigma de desarrollo preferido, e implica un conjunto de reglas y prácticas que se aplican dentro de las actividades estructurales de planificación, diseño, codificación y pruebas [51].

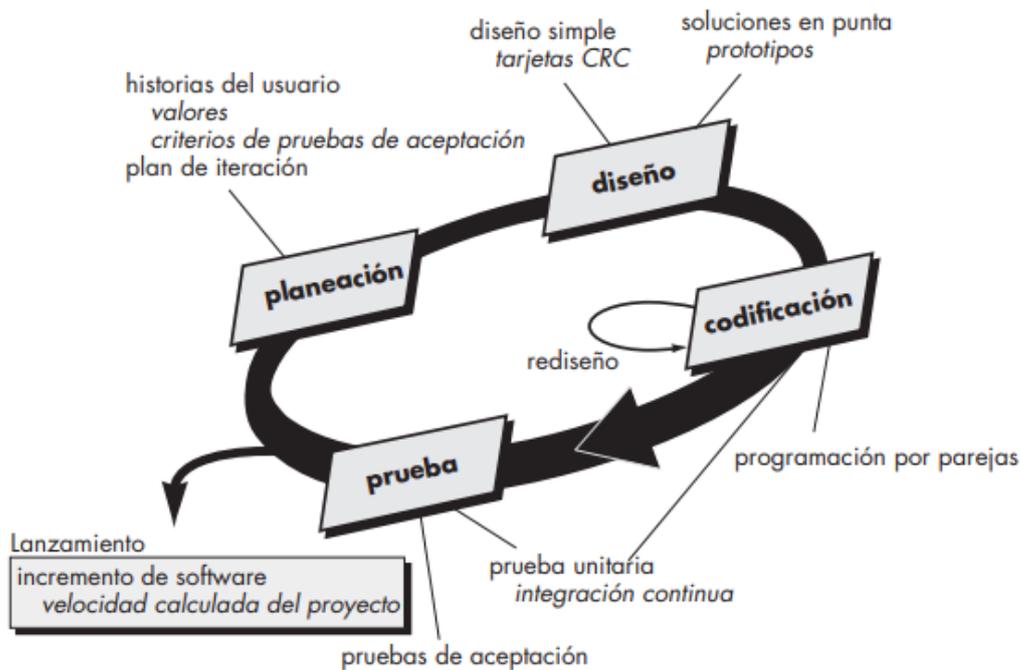


Figura 11. Proceso de la Metodología XP [51].

4.6.3. Planeación

Se inicia con una actividad de recopilación de requisitos, que tiene como objetivo comprender el contexto comercial del software y obtener una comprensión clara de las salidas deseadas. Esta recopilación se lleva a cabo a través de historias de usuarios, las cuales describen los resultados, características y funcionalidades necesarias del software en desarrollo. Estas historias son redactadas por el cliente y se agregan al conjunto de tareas pendientes del Trabajo [52].

a. Roles XP

En la metodología XP se adopta la noción de roles para organizar y asignar las diferentes actividades requeridas durante el desarrollo del Trabajo. Cada uno de estos roles es desempeñado por uno o más miembros del equipo.

- **Programador:** En el contexto de XP, el programador desempeña el rol de crear y ejecutar pruebas unitarias, así como desarrollar el código del sistema. Su

responsabilidad es garantizar que las pruebas unitarias sean efectivas y que el código producido cumpla con los requisitos establecidos [53].

- **Ciente:** El cliente desempeña un papel fundamental en la definición y validación de los requisitos del sistema. Su responsabilidad es redactar las historias de usuario, que describen las funcionalidades deseadas del sistema, y elaborar las pruebas funcionales para evaluar su implementación [54].
- **Encargado de pruebas (Tester):** Su función principal es colaborar estrechamente con el cliente en la elaboración de las pruebas funcionales que validarán la implementación del sistema [55].

4.6.4. Diseño

El diseño en XP se basa en el principio de "mantenlo sencillo" (MS). En lugar de optar por soluciones más complejas, siempre se prefiere un diseño simple. El diseño actúa como una guía para la implementación de una historia de acuerdo con su descripción. XP promueve el uso de tarjetas de calificación de contribuyente responsable (CRC) como una forma efectiva de desarrollar y mejorar el software [56].

4.6.5. Codificación

Durante la fase de codificación en XP, se sigue un enfoque específico. Uno de los aspectos más destacados en esta etapa es la programación en parejas, donde dos personas trabajan juntas en una estación de trabajo. Esto facilita la resolución de problemas en tiempo real y asegura la calidad del código a medida que se desarrolla [51].

- **Programación en Parejas:** En XP, se requiere que todo el código sea desarrollado por pares de programadores que trabajen juntos. Aunque esto podría parecer una reducción del 50% en la productividad inicialmente, según XP, no se experimenta tal pérdida. Al trabajar en parejas, se logra un diseño de mayor calidad, un código más organizado y con menos errores que si se trabajara de manera individual [57].

4.6.6. Pruebas

La metodología XP dedica una atención significativa a las pruebas, categorizándolas en distintos tipos y asignándoles funciones específicas. Además, establece directrices precisas sobre quién debe encargarse de realizar las pruebas, cuándo deben llevarse a cabo y cómo deben ser ejecutadas [58].

- **Pruebas Unitarias**

Las pruebas unitarias son aplicadas a todos los métodos relevantes de las clases en el Trabajo, asegurando que cada clase esté acompañada de su propio conjunto de pruebas. Es un requisito fundamental que ninguna clase sea liberada sin tener sus pruebas correspondientes. Este enfoque optimiza el trabajo del programador y garantiza la calidad del código resultante [56].

- **Pruebas de Aceptación**

Las pruebas de aceptación, también denominadas pruebas funcionales, son realizadas por el cliente con base en los requisitos establecidos en las historias de usuario. Estas pruebas permiten identificar y corregir posibles errores, asegurando que el software cumpla con los requisitos y expectativas del cliente [57].

4.7. Modelo TAM

El Modelo de Aceptación Tecnológica (TAM) es una teoría desarrollada para explicar cómo los usuarios adoptan y utilizan tecnologías de la información, se basa en dos constructos principales: [59]

- **Percepción de utilidad (PU):** Hace referencia a cómo los usuarios perciben si el sistema resultará beneficioso en su ámbito laboral o académico.
- **Facilidad de uso (FE):** Se refiere a la manera en que los usuarios perciben la facilidad con la que pueden aprender y utilizar el software [60].

4.8. Modelo de Software 4+1

Utiliza cinco vistas concurrentes para describir la arquitectura de un sistema: vista lógica, vista de desarrollo, vista física, vista de procesos y vista de escenarios. Esta combinación de enfoques facilita la comunicación entre equipos y la toma de decisiones arquitectónicas informadas al brindar una comprensión completa y detallada del software [4].

4.8.1. Vista Lógica

Esta vista se enfoca en representar las funcionalidades que el sistema ofrecerá a los usuarios finales, así como las interacciones y servicios que proporcionará para cumplir con los requisitos funcionales [61].

4.8.2. Vista de Desarrollo

En esta vista se representa el sistema desde el punto de vista del programador, enfocándose en la gestión del software. Es decir, se describe cómo se divide el sistema en componentes y las dependencias existentes entre ellos.

4.8.3. Vista de Procesos

Muestra cómo funciona el sistema durante su tiempo de ejecución. Brinda una comprensión de la funcionalidad y el comportamiento del software durante su funcionamiento, describiendo cómo los componentes que integran el sistema interactúan y se comunican entre sí [62].

4.8.4. Vista Física

Esta vista muestra cómo se despliega el software en el hardware y la infraestructura, representando los componentes físicos del sistema y sus conexiones. Es de gran importancia para garantizar el rendimiento y la disponibilidad del sistema.

4.8.5. Vista de Escenarios

Describen las interacciones entre objetos y procesos, y son esenciales para identificar y validar el diseño arquitectónico. Se utilizan también como base para probar un prototipo de la arquitectura [63].

4.9. Librerías y Frameworks

4.9.1. Scikit-learn

Scikit-learn es una biblioteca de Python que combina varios algoritmos avanzados de aprendizaje automático para abordar problemas supervisados y no supervisados de tamaño moderado. Su principal objetivo es poner el aprendizaje automático al alcance de personas sin conocimientos especializados empleando un lenguaje de programación de alto nivel y fácil de usar [64], scikit-learn incluye implementaciones populares de algoritmos de aprendizaje automático, como LIBSVM y LIBLINEAR. Además, otras bibliotecas de Python, como NLTK, han integrado interfaces compatibles con scikit-learn [65].

4.9.2. Keras

Keras es una API de alto nivel para redes neuronales, diseñada para facilitar la experimentación rápida. Ofrece una variedad de características:

- **Compatibilidad fluida:** Keras permite que el código se ejecute sin problemas tanto en la CPU como en la GPU.
- **Interfaz fácil de usar:** La API está diseñada para ser intuitiva, lo que facilita la creación rápida de prototipos de modelos de aprendizaje profundo.

- **Soporte integrado:** Keras proporciona soporte integrado para redes convolucionales (utilizadas en visión por computadora), redes recurrentes (para procesamiento de secuencias) y cualquier combinación de ambas.
- **Flexibilidad en las arquitecturas de red:** Keras admite diversas configuraciones de red, incluyendo modelos con múltiples entradas o salidas, compartición de capas y modelos compartidos. Esto lo hace adecuado para construir una amplia gama de modelos de aprendizaje profundo, desde redes de memoria hasta máquinas neuronales de Turing.
- **Compatibilidad con backends:** Keras puede ejecutarse en varios backends, como TensorFlow, CNTK o Theano. Esta flexibilidad permite a los usuarios elegir el backend más adecuado para sus necesidades específicas [66].

4.9.3. *Tensorflow*

TensorFlow es una biblioteca de código abierto ampliamente utilizada en el campo del aprendizaje automático (machine learning) y se utiliza principalmente para desarrollar, entrenar y desplegar modelos de aprendizaje automático [67]. Es una biblioteca que permite realizar cálculos numéricos de alto rendimiento utilizando gráficos computacionales. Estos gráficos representan las operaciones matemáticas como nodos y las variables como tensores multidimensionales, de ahí el nombre "TensorFlow" [68].

4.9.4. *Spacy*

Es una librería de software de código abierto para NLP. spaCy es un marco relativamente nuevo que se utiliza para implementar el NLP [69]. Una de las características más destacadas de spaCy es que incluye modelos lingüísticos y vectores de palabras preformados para más de 60 idiomas. A diferencia de sus predecesores más académicos, como NLTK, spaCy se centra principalmente en la producción y el envío de código [70].

4.9.5. *Otras librerías Adicionales*

- **numpy:** Para realizar operaciones matemáticas y numéricas eficientes.
- **googlesearch:** Proporciona funciones para realizar búsquedas en Google.
- **bs4 (BeautifulSoup):** Librería para analizar y extraer información de documentos HTML y XML.
- **nltk:** Librería de procesamiento de lenguaje natural que contiene herramientas y recursos para el análisis de texto.

4.10. Descripción de las Librerías para trabajar con Redes Neuronales

En la tabla 2 se presenta una descripción sobre las características principales de las librerías mencionadas, así como una razón sobre la elección de utilizar estas librerías en particular. También se incluye información sobre los lenguajes de programación compatibles, facilidad de uso y si proporcionan Modelos Pre-entrenados.

Tabla 2. Descripción de Librerías [Fuente propia].

Librerías	Características	Facilidad de Uso	Modelos Pre-entrenados	Lenguaje de Programación	Justificación
spaCy	<ul style="list-style-type: none"> • Spacy es una biblioteca de alta velocidad de procesamiento de lenguaje natural de código abierto. • Tokenización, lematización, etiquetado gramatical, reconocimiento de entidades nombradas y muchas otras funciones comunes de procesamiento de texto están disponibles. • Se basa en Python y se concentra en la eficiencia y el rendimiento. • Proporciona modelos previamente entrenados para una variedad de idiomas que se pueden utilizar para realizar tareas específicas de (NLP). 	Ofrece una API intuitiva y bien documentada	Si, proporciona modelos pre-entrenados para más de 60 idiomas	Python	<ul style="list-style-type: none"> • Eficiencia y velocidad en el análisis y procesamiento de texto. • Amplia gama de modelos pre-entrenados para diferentes tareas de NLP. • Ofrece una API intuitiva y bien documentada, lo que facilita su implementación.
scikit-learn	<ul style="list-style-type: none"> • Scikit-learn es una biblioteca de aprendizaje automático basada en Python que también incorpora características para el NLP. • Proporciona herramientas para la extracción de 	Proporciona una interfaz coherente y fácil de usar	-No	Python	<ul style="list-style-type: none"> • Amplia colección de algoritmos de aprendizaje automático para tareas de clasificación y regresión. • Proporciona una interfaz fácil de usar

	<p>características de texto.</p> <ul style="list-style-type: none"> • Los algoritmos de aprendizaje supervisado y no supervisado se ofrecen para la clasificación y otras tareas de procesamiento de texto. • Ofrece herramientas para la selección de parámetros y la evaluación de modelos. 				<p>y coherente para construir modelos de NLP.</p> <ul style="list-style-type: none"> • Documentación detallada y abundante con ejemplos prácticos y casos de uso.
Keras	<ul style="list-style-type: none"> • Keras es una biblioteca de aprendizaje profundo que se basa en Python y está disponible de código abierto. • Keras se puede ejecutar sobre una variedad de bibliotecas de cálculo numérico, como TensorFlow y Theano, lo que le permite aprovechar todas las características de estas bibliotecas de cálculo numérico que ya existen. • Keras admite una variedad de arquitecturas de redes neuronales que se pueden utilizar en tareas de procesamiento natural de la información (NLP), incluidos modelos de atención, redes neuronales Siamesas, redes neuronales convolucionales (CNN) y redes neuronales recurrentes (RNN). 	<p>Permite una fácil definición y entrenamiento de modelos de aprendizaje profundo</p>	-No	Python	<ul style="list-style-type: none"> • Interfaz de alto nivel para construir y entrenar modelos de aprendizaje profundo. • Permite la creación rápida de arquitecturas de redes neuronales para tareas de NLP. • Amplia comunidad de usuarios y abundante documentación con ejemplos y guías paso a paso.

TensorFlow	<ul style="list-style-type: none"> • TensorFlow es una biblioteca de aprendizaje automático y profundo de código abierto. • Ofrece una plataforma completa para el desarrollo y el entrenamiento de modelos de aprendizaje automático, como el NLP. • Ofrece una variedad de API y herramientas para crear y ejecutar redes neuronales. • Permite el desarrollo de modelos de alto y bajo nivel a través de Keras y su API de grafo computacional. • Es popular entre la comunidad de investigación y cuenta con una amplia red de bibliotecas complementarias. 	Ofrece una amplia gama de herramientas y recursos para el desarrollo de modelos de aprendizaje automático	-No	Python, C++, Java, JavaScript, Swift, y otros	<ul style="list-style-type: none"> • Framework versátil y escalable para el desarrollo de modelos de aprendizaje automático en NLP. • Soporte para la creación de redes neuronales profundas y su entrenamiento en grandes volúmenes de datos. • Amplio ecosistema de herramientas y recursos para el NLP.
Gensim	<ul style="list-style-type: none"> • El modelado de temas y el procesamiento de texto a gran escala son los dominios principales de Gensim. • Hace posible la implementación efectiva de algoritmos de modelado como Latent Dirichlet Allocation (LDA) y el análisis semántico oculto (LSA). • Además, Gensim incluye herramientas como Word2Vec y FastText para crear y entrenar modelos de representación de palabras. • Es una biblioteca ligera y fácil de usar centrada en la escalabilidad y el rendimiento. 	Proporciona algoritmos eficientes para la extracción de temas y creación de modelos de word embeddings (incrustaciones de palabras)	- No	Python	<ul style="list-style-type: none"> • Enfoque en el modelado de temas y la vectorización de texto. • Proporciona algoritmos eficientes para la extracción de temas y creación de modelos de word embeddings. • Documentación detallada y guías de uso para implementar técnicas de NLP específicas.

-
- La búsqueda de información, la comparación de documentos y la extracción de temas son tareas de NLP en las que Gensim es particularmente útil.
-

4.11. Trabajos Relacionados

4.11.1. TR1: Detección de plagio en documentos árabes: Enfoques, arquitectura y Sistemas.

El estudio de la detección del plagio ha suscitado últimamente una gran atención. Aunque los sistemas de detección de plagio pueden utilizarse con muchos idiomas diferentes, funcionan mejor cuando se crean específicamente para un idioma determinado, teniendo en cuenta sus cualidades especiales. La eficiencia de estos sistemas viene determinada en última instancia por la eficacia de sus principales módulos de procesamiento, la optimización de consultas y la reducción de documentos, que tienen un impacto significativo en los tiempos de respuesta y en la calidad de los resultados. En este ensayo, analizan varios métodos y proporcionen una arquitectura y un sistema para encontrar plagios en documentos árabes. Los cuatro niveles de los módulos principales de procesamiento de la arquitectura web propuesta permiten a los usuarios revisar los textos y analizar los resultados mediante una interfaz intuitiva [71].

4.11.2. TR2: Creación de un prototipo de sistema de detección de plagio con NLTK.

Este trabajo aborda el problema del plagio en entornos académicos y propone el desarrollo de un sistema de detección de plagio específico. Aunque existen métodos de detección de plagio, ninguno es óptimo para abordar la paráfrasis o la detección de la idea subyacente. Por lo tanto, proponen desarrollar un prototipo capaz de analizar textos copiados de internet y señalar los párrafos originales, incluso cuando hayan sufrido cambios superficiales. Este prototipo se estructura en módulos y utiliza conocimientos adquiridos en el área de Lenguas Aplicadas. Las tecnologías específicas utilizadas para llevar a cabo el Trabajo, incluyen el lenguaje de programación Python, el corpus Wikicorpus, el entorno de desarrollo Anaconda y Spyder, así como las bibliotecas de código de Python NLTK (Natural Language Toolkit) y WordNet, junto con otras bibliotecas auxiliares. El objetivo es proporcionar una herramienta efectiva para detectar y prevenir el plagio, mejorando la integridad académica y evitando sanciones por plagio [72].

4.11.3. TR3: Diseño e implementación de un sistema de detección de Plagio mediante el método de similitud del coseno.

El objetivo de este Trabajo de investigación es crear un sistema llamado Kipcheck que pueda calcular ecuaciones de palabras y evaluar la tasa de plagio de un documento utilizando el método del coseno. Para evitar sanciones académicas, esta evaluación tiene como objetivo garantizar que los documentos creados tengan un nivel mínimo de fraude. Kipcheck realiza tres pruebas de aplicación del sistema: cálculo máximo de palabras procesables, comparación de cálculos realizados con la aplicación y de forma manual, y verificación de la consistencia de los resultados de cálculo de la aplicación en base a esquemas diferentes [73].

4.11.4. TR4: Diseño e Implementación de una herramienta en Studium para evitar los plagios en asignaturas de Ingeniería.

El objetivo del Trabajo es crear una solución universal y efectiva para todos los lenguajes de programación para las asignaturas de grado. Esta solución, que se basa en Studium, que permitirá al profesor activar fácilmente un sistema anticopia al crear tareas o entregas. Para reducir los gastos utilizan software libre. Los estudiantes podrán solicitar un informe sobre el análisis del sistema anticopia para ver si su tarea es similar a la de sus compañeros; sin embargo, esta función será opcional y solo estará disponible después de enviar la tarea. El sistema de detección de plagio opera utilizando un enfoque científico conocido como "Algoritmos Locales para la Huella Digital de Documentos" [74].

5. Metodología

5.1. Área de Estudio

El Trabajo de Integración Curricular se llevó a cabo en la Universidad Nacional de Loja, específicamente en la Facultad de Energía, como parte del programa de la Carrera de Ingeniería en Computación. Esta iniciativa se desarrolló en colaboración con el cliente principal y director del Trabajo, el Ing. Oscar Miguel Cumbicus Pineda.

5.2. Procedimiento

El proceso para alcanzar el objetivo general del TT se detalla a continuación, mencionando cada fase de la metodología XP, empleada para cada objetivo específico:

1. **Objetivo 1:** Diseñar el modelo de control anti plagio, mediante redes Neuronales.

a. **Fase 1 Planificación:**

- Se hizo una búsqueda sobre los tipos de Redes Neuronales (Véase el marco teórico la sección **Tipos de redes neuronales**)
- Se realizó una búsqueda de librerías para trabajar con las redes neuronales (Véase el marco teórico la sección **Librerías y Frameworks**)

b. **Fase 2 Diseño:**

- Se generó un diagrama el mismo que presenta los componentes del detector de plagio (Véase resultados la **Actividad 1.1**).
- Seguidamente se diseñó la arquitectura de la red neuronal siamesa, la misma que fue seleccionada para trabajar con la detección de plagio (Véase en resultados la **Actividad 1.2**).
- Se generó un dataset de textos, que consistió en recopilar y crear conjuntos de datos que incluían textos originales y potencialmente plagiados. Estos textos fueron extraídos del repositorio del Dspace de la UNL (Véase en resultados la **Actividad 1.2**).

c. **Fase 3 Codificación:**

- Se llevó a cabo la codificación de los componentes y funciones necesarias para el análisis de plagio. Esto involucro la lectura y limpieza de archivos, conversión de los documentos en un formato adecuado, análisis mediante técnicas de vectorización y cálculo de similitud del coseno (Véase en resultados la **Actividad 1.1**).

- Se codificó la red neuronal siamesa, así como su configuración en las capas de entrada, procesamiento y salida (Véase en resultados la **Actividad 1.2**).

d. Fase 4 Pruebas:

- Se realizaron pruebas unitarias, con documentos extraídos de la web para verificar el funcionamiento de las funciones implementadas en la fase de codificación (Véase en resultados la **Actividad 1.1** y **Actividad 1.2**).

2. Objetivo 2: Desarrollar el software anti plagio utilizando el framework Django, aplicando el Modelo de software 4+1.

El desarrollo para este objetivo, se encuentra detallado en el **Anexo 6**.

a. Fase 1 Planificación:

- Se realizó una entrevista al cliente del software con la finalidad de generar las Historias de usuario (véase **Anexo 3**).
- Se empleó la norma IEEE830 para la elaboración de la documentación de requerimientos (véase **Anexo 4**).
- A partir del documento de especificación de requisitos (véase **Anexo 5** sección **Fase Planificación**), se establecieron las distintas iteraciones, cada una de ellas con sus correspondientes historias de usuario definidas.

b. Fase 2 Diseño:

- Se diseñó la arquitectura del software anti plagio (véase **Anexo 5** sección **Fase de Diseño**).
- Se empleó el modelo 4+1, para generar la vista lógica, vista de despliegue, vista de procesos, vista física y vista de escenarios, junto con el modelo relacional de la base de datos (véase **Anexo 5** sección **Fase de Diseño**).

c. Fase 3 Codificación:

- Se llevó a cabo una explicación acerca de las normas de codificación, se elaboraron las historias de usuario correspondientes a cada una de las iteraciones planificadas. (véase **Anexo 5** sección **Fase de Codificación**).

d. Fase 4 Pruebas:

- Se genero un plan de pruebas unitarias con la finalidad de validar cada una de las funciones implementadas en el prototipo de software anti plagio (véase **Anexo 7**).

5.3. Métodos

a) Estudio de caso.

Es un método de investigación que analiza situaciones específicas, con el objetivo de obtener un conocimiento profundo del fenómeno estudiado, mediante la recopilación de datos de diversas fuentes de evidencia. De esta manera, se buscaron casos similares en los cuales se haya llevado a cabo la implementación de un modelo de software con el fin de detectar y prevenir el plagio académico. (Véase sección de **Trabajos Relacionados**).

5.4. Técnicas

- **Entrevistas:** Mediante esta técnica, se logró recopilar información valiosa del cliente del software, y de esta manera generar las Historias de usuario para el software anti plagio.
- **Reuniones:** Esta técnica de reuniones resultó beneficiosa, ya que permitió la participación activa del cliente del software que es uno de los involucrados en el Trabajo de titulación. A través de estas reuniones, se logró revisar los avances en cada etapa del Trabajo, lo que facilitó recibir comentarios y críticas constructivas.

5.5. Estándar

- IEEE 830: Consiste en un conjunto de sugerencias para la descripción de los requerimientos del software, su uso permitió cumplir con el segundo objetivo específico del Trabajo de Titulación.

5.6. Materiales

- **Recursos Humanos:** El desarrollo de este Trabajo de Titulación fue llevado a cabo por los participantes que se detallan en la tabla 3.

Tabla 3. Recursos Humanos.

Nombre	Descripción
Santiago Alexander Román Silva	Estudiante a cargo de la ejecución del Trabajo.

Jaime Oswaldo Paqui Medina	Estudiante a cargo de la ejecución del Trabajo.
Ing. Oscar Miguel Cumbicos Pineda	Docente director del Trabajo de Trabajo de Titulación

- **Recursos de Software y Hardware:** Durante la elaboración del Trabajo de Titulación se utilizaron los recursos de hardware y software detallados en la tabla 4 y 5, respectivamente.

Tabla 4. Recursos de Software.

Recurso	Descripción	Link de acceso
Google Drive	Servicio que permitió almacenar y trabajar de forma colaborativa con los diferentes documentos.	https://drive.google.com/drive/folders/1mGyMFmJINMPw4VJaY6cP1rvOzDWO3n2t?usp=sharing
Zoom	Plataforma de videoconferencia en línea que permitió participar en reuniones virtuales, trabajo colaborativo, retroalimentación del TT, etc.	https://zoom.us/
Lucidchart	Software que permitió desarrollar la arquitectura general.	https://www.lucidchart.com
StarUML	Software que permitió desarrollar los diferentes diagramas.	https://staruml.io
Visual Studio Code	Software que permitió elaborar el código fuente	https://code.visualstudio.com
Django	Framework que permitió desarrollar el software web anti plagio.	https://www.djangoproject.com
Mysql	Sistema de gestión de base de datos que permitió almacenar, gestionar y manipular los datos del software.	https://www.mysql.com

Tabla 5. Recursos de Hardware.

Recurso	Descripción
Laptop	Equipo principal para el desarrollo del TT.

- **Insumos:** La tabla 6 presenta los elementos requeridos para llevar a cabo el Trabajo de Trabajo de Titulación.

Tabla 6. Insumos.

Insumos	Descripción
Internet	Internet se empleó para diversas actividades, tales como la comunicación, investigación y el acceso a diferentes herramientas.
Software: FirmaEC	Este software se utilizó para firmar las diferentes secciones del Trabajo de Titulación tales como: Entrevista, Especificación de Requisitos de Software, Manuales de Usuario, etc.

6. Resultados

En este apartado se proporcionan los resultados alcanzados durante la ejecución del Trabajo de Integración Curricular, en base a la metodología XP.

6.1. Objetivo 1: Diseñar el modelo de control anti plagio, mediante redes neuronales con Scikit-Learn.

Actividad 1.1: Diseñar el modelo mediante el lenguaje Python que marcara el marco de trabajo para el software anti plagio.

En esta actividad se realizó una búsqueda sobre las técnicas para el preprocesamiento de los datos como son: tokenización, normalización y lematización (véase sección **Técnicas para el Procesamiento de Lenguaje Natural.**). Para la parte de tokenización esta técnica nos sirvió para descomponer los textos en partes más pequeñas como por ejemplo frases. La normalización esta técnica nos sirvió para convertir todas las letras a minúsculas, eliminando así las diferencias entre mayúsculas y minúsculas, excluir caracteres especiales, tildes, etc. Por último, la lematización nos sirvió para convertir las palabras a su forma canónica, es decir, a su forma fundamental y reconocible en el lenguaje.

Seguidamente se generó un diagrama con los componentes que formaran parte del detector de plagio (véase **Figura 12**). El flujo de este diagrama inicializa con el usuario, ya sea docente o estudiante, deberá subir el documento que desea analizar y activará la función de inicio de análisis del documento. Durante esta etapa, el documento experimentará un procesamiento general en el cual se extraerá el texto en bruto. Luego se llevará a cabo la fase de limpieza, tokenización y normalización del texto obtenido, lo que permitirá identificar los tópicos que componen el documento y facilitará la comparación con otros Trabajos de Titulación (TT), constituyendo así la primera parte de los resultados. Posteriormente, se llevará a cabo la búsqueda de similitudes en internet utilizando el texto procesado, representando la segunda parte de los resultados. Finalmente, se comprobarán ambos resultados para generar los valores de similitud finales. Este proceso culminará en la generación de la salida, que consistirá en el documento final con los valores de plagio identificados.

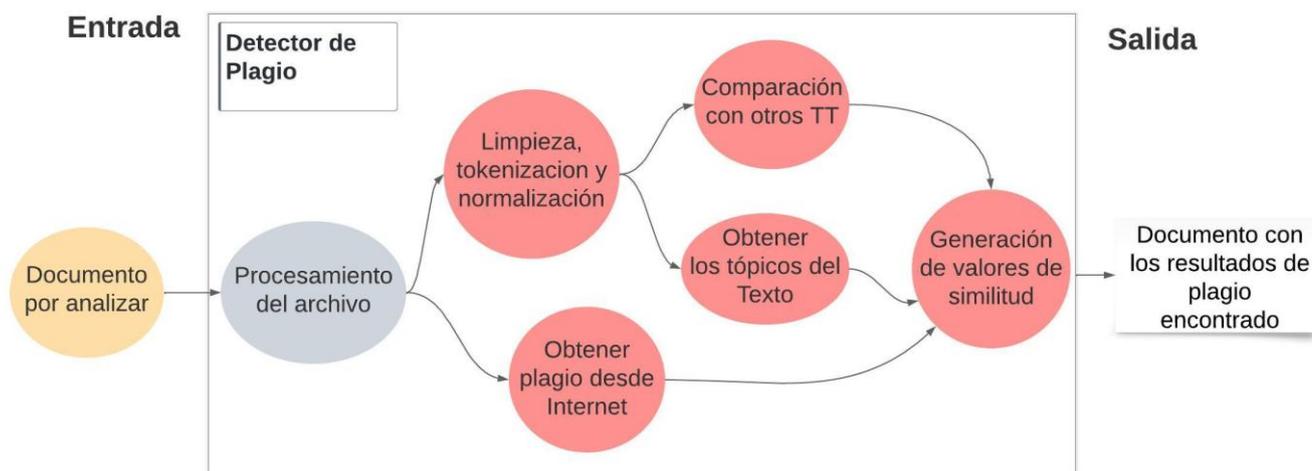


Figura 12. Componentes Generales del detector de plagio [Fuente propia].

Posteriormente se procedió a codificar los componentes y las diferentes funciones principales que implicaban: lectura de los archivos a comparar, limpieza de los archivos, conversión de los documentos ingresados a un formato adecuado para el procesamiento de código, análisis de los documentos mediante técnicas de vectorización y el cálculo de la similitud del coseno para obtener el porcentaje de similitud entre ellos. Para visualizar el código completo generado visite el link: <https://github.com/SantiagoRoman26/Plagio>.

De entre las funciones principales se presenta en la **Figura 13** la función limpieza, la misma que permite limpiar el texto extrayendo todo carácter especial propias del idioma español, se eliminan espacios en blanco y se normalizan saltos de línea para obtener un formato más consistente. Posteriormente, se corrigen caracteres especiales, como las vocales acentuadas y comillas. Además, se reemplaza el espacio en blanco especial '\u200b' por un espacio normal, luego se procede a normalizar el texto mediante la biblioteca 'unicodedata' esto con la finalidad de que sea más fácil de analizar el texto por el algoritmo. Seguidamente, el texto se divide en oraciones mediante la tokenización de oraciones usando la biblioteca NLTK (Natural Language Toolkit) usando la función propia del idioma español. Se filtran las oraciones que consisten únicamente en un punto y se ajusta la puntuación al final de las oraciones. Finalmente, se realiza una segunda fase de limpieza, donde se agrupan las oraciones que forman un párrafo, considerando la lógica de inicio de oraciones en minúsculas. Todo este proceso resulta en una lista de oraciones mejoradas en términos de formato y coherencia.

```

def limpieza(archivo):
    archivo_limpio = re.sub(r'\n+', '\n', archivo.strip())
    archivo_limpio = re.sub('\n', ' ', archivo_limpio.strip())
    archivo_limpio = re.sub(r'[.][.]+', '.', archivo_limpio.strip())
    archivo_limpio = re.sub(r'[ ]+', ' ', archivo_limpio.strip())
    archivo_limpio = re.sub('á', 'a', archivo_limpio.strip())
    archivo_limpio = re.sub('é', 'e', archivo_limpio.strip())
    archivo_limpio = re.sub('í', 'i', archivo_limpio.strip())
    archivo_limpio = re.sub('ó', 'o', archivo_limpio.strip())
    archivo_limpio = re.sub('ú', 'u', archivo_limpio.strip())
    archivo_limpio = re.sub("'", '', archivo_limpio.strip())
    archivo_limpio = re.sub('"', '', archivo_limpio.strip())
    archivo_limpio = re.sub('\u200b', ' ', archivo_limpio.strip())

    archivo_limpio = unicodedata.normalize("NFKD", archivo_limpio.strip())

    oraciones = sent_tokenize(archivo_limpio.strip(), "spanish")
    oraciones_limpias = []
    for oracion in oraciones:
        if oracion.strip() != '.':
            if oracion.strip().endswith('.'):
                oracion_a_agregar = oracion[:-1]
            else:
                oracion_a_agregar = oracion
            oraciones_limpias.append(oracion_a_agregar.strip())

    i=0
    j=0

    oraciones_mas_limpias = []
    while i < len(oraciones_limpias):
        if i == 0:
            oraciones_mas_limpias.append(oraciones_limpias[0])
        else:
            palabras_oracion = word_tokenize(oraciones_limpias[i])
            if palabras_oracion[0].islower():
                oraciones_mas_limpias[j] += " " + oraciones_limpias[i]
            else:
                j += 1
                oraciones_mas_limpias.append(oraciones_limpias[i])
        i += 1
    return oraciones_mas_limpias

```

Figura 13. Función para la Limpieza de Archivos.

Así mismo en la **Figura 14** se presenta la función para obtener el tema principal del documento mediante un procedimiento para identificar los temas dominantes en un texto, esto se realiza mediante el uso de la técnica de Modelado de Tópicos con Latent Dirichlet Allocation (LDA). Inicialmente se carga el modelo de procesamiento de lenguaje natural en español ("es_core_news_sm") utilizando la biblioteca spaCy, luego se preparan hilos de ejecución para procesar archivos de referencia y el texto de prueba de manera simultánea. Luego, se utiliza LDA para modelar los tópicos en los archivos de referencia posteriormente limpiados,

empleando un diccionario y un corpus generado a partir de los documentos procesados. Esto resulta en forma de palabras clave asociadas al tóxico identificado.

```
def obtener_tema_del_texto(archivo_test, sw, cantidad_de_topicos):
    log.info("TOPICOS | Obteniendo topicos del texto ...")
    log.debug("TOPICOS | Preparando archivos para modelo LDA ...")
    nlp = spacy.load('es_core_news_sm')

    hilos_preparar_archivos_para_lda = list()
    for archivo in archivos_referencia_limpios:
        hilo_preparar_archivo_para_lda = threading.Thread(target=preparar_archivo_referencia_para_lda,
                                                         args=(archivo, nlp, sw,))
        hilos_preparar_archivos_para_lda.append(hilo_preparar_archivo_para_lda)
        hilo_preparar_archivo_para_lda.start()

    texto_preparado_test = preparar_texto_para_lda(archivo_test, nlp, sw)

    for index, thread in enumerate(hilos_preparar_archivos_para_lda):
        thread.join()

    log.debug("TOPICOS | Archivos de referencia preparados")
    log.debug("TOPICOS | Corriendo algoritmo LDA para archivos de referencia ...")
    diccionario = corpora.Dictionary(textos_preparados_referencia)
    corpus = [diccionario.doc2bow(texto) for texto in textos_preparados_referencia]
    modelo_lda = gensim.models.LdaMulticore(corpus, num_topics=10, id2word=diccionario, passes=2)

    log.debug("TOPICOS | Algoritmo LDA para archivos de referencia finalizado")

    log.debug("TOPICOS | Corriendo algoritmo LDA para archivos de test ...")
    indice, score = sorted(modelo_lda[diccionario.doc2bow(texto_preparado_test)], key=lambda tup: -1 * tup[1])[0]

    topico_con_mas_score.extend(
        [palabra.split("*")[1].replace("\", \"") for palabra in modelo_lda.print_topic(indice, cantidad_de_topicos).split(" + ")])

    log.info(f"TOPICOS | Topicos del texto: {topico_con_mas_score}")
```

Figura 14. Función para los Tópicos del Texto.

Seguidamente se presenta en la **Figura 15** las funciones que permiten tratar el texto del documento para ser usado para el modelo LDA. Primero se extraen los sustantivos lematizados de cada oración, se eliminan las palabras irrelevantes según una lista de palabras vacías (stopwords), y se filtran palabras numéricas. El resultado es una lista de palabras sustantivas lematizadas, luego el texto original se procesa, y el resultado se almacena en una estructura de datos adecuada para el análisis posterior con el algoritmo LDA.

```

def preparar_texto_para_lda(archivo, nlp, sw):
    archivo_mas_limpio = []
    for oracion in archivo:
        sustantivos = [token.lemma_ for token in nlp(oracion.lower()) if token.pos_ in ['NOUN'] and len(token.text) > 2]
        oracion_mas_limpia = [palabra for palabra in sustantivos if not palabra in sw and not str(palabra).isnumeric()]
        archivo_mas_limpio = archivo_mas_limpio + [palabra for palabra in oracion_mas_limpia if str(palabra) != '']
    return archivo_mas_limpio

def preparar_archivo_referencia_para_lda(archivo, nlp, sw):
    archivo_preparado = preparar_texto_para_lda(archivo.texto, nlp, sw)
    textos_referencia.append(ArchivoTxt(archivo.nombre, archivo.extension, archivo_preparado))
    textos_preparados_referencia.append(archivo_preparado)

```

Figura 15. Función para entrenar el modelo LDA.

En la **Figura 16**, se muestra la función que permite obtener la similitud entre dos textos, primero, se convierte un texto en un vector de términos usando el conteo de palabras, utilizando una expresión regular para extraer las palabras del texto y luego generando un contador de frecuencias de términos, seguidamente se implementa el cálculo de similitud del coseno entre los dos vectores representativos de términos, luego se calcula el numerador como la suma de los productos de las frecuencias de términos coincidentes, y se determina el denominador como el producto de las normas euclidianas de los vectores. Este enfoque proporciona una medida de similitud entre los textos, donde valores más altos indican mayor similitud semántica.

```

def obtener_similitud_del_coseno(vector1, vector2):
    interseccion = set(vector1.keys()) & set(vector2.keys())
    numerador = sum([vector1[x] * vector2[x] for x in interseccion])

    sum1 = sum([vector1[x] ** 2 for x in vector1.keys()])
    sum2 = sum([vector2[x] ** 2 for x in vector2.keys()])
    denominador = math.sqrt(sum1) * math.sqrt(sum2)

    if not denominador:
        return 0.0
    else:
        return float(numerador) / denominador

def string_a_vector(texto):
    palabras = re.compile(r'\w+').findall(texto)
    return Counter(palabras)

def obtener_similitud(texto1, texto2):
    vector1 = string_a_vector(texto1)
    vector2 = string_a_vector(texto2)
    return obtener_similitud_del_coseno(vector1, vector2)

```

Figura 16. Función para Obtener la similitud del Coseno de 2 textos.

En la **Figura 17** se muestra la función que permite realizar una búsqueda en internet de la oración preparada para obtener el porcentaje de similitud. Para ello se utiliza un motor de búsqueda usando un determinado número de enlaces relevantes a partir de la consulta preparada. Posteriormente, para cada URL recuperada, se extrae el texto HTML asociado y se somete a un proceso de limpieza para mejorar la coherencia y calidad del contenido. Luego, se compara la oración preparada de la consulta con cada oración en el texto limpio, utilizando una medida de similitud del coseno. Para conocer sobre la descripción de los componentes implementados (visite el **Anexo 1**).

```
def obtener_oracion_mas_parecida_de_internet(oracion, oracion_preparada, sw, cantidad_de_links, buscar_en_pdfs):
    mayor_porcentaje = 0.0
    oracion_mas_parecida = ''
    url_donde_se_encontro = ''
    archivo_donde_se_encontro = ''
    ubicacion_dentro_de_la_lista = 0

    mutex.acquire()
    for url in search(oracion_preparada, tld="com.ar", num=cantidad_de_links, stop=cantidad_de_links, pause=2):
        time.sleep(0.002)
        mutex.release()
        if (not buscar_en_pdfs) and url.endswith(".pdf") or str(url).endswith(".pdf/"):
            mutex.acquire()
            continue
        else:
            texto = obtener_html_como_texto(url)
            if texto != '':
                archivo = limpieza(texto)
                for oracion_a_comparar in archivo:
                    oracion_a_comparar_preparada = preparar_oracion(oracion_a_comparar, sw)
                    if oracion_a_comparar_preparada is None:
                        continue
                    similitud = obtener_similitud(oracion_preparada, oracion_a_comparar_preparada)
                    if similitud > 0.8:
                        mayor_porcentaje = similitud
                        oracion_mas_parecida = oracion_a_comparar
                        url_donde_se_encontro = url
                        archivo_donde_se_encontro = archivo
                        ubicacion_dentro_de_la_lista = int(archivo.index(oracion_a_comparar))
                        break
                    elif similitud > mayor_porcentaje:
                        mayor_porcentaje = similitud
                        oracion_mas_parecida = oracion_a_comparar
                        url_donde_se_encontro = url
                        archivo_donde_se_encontro = archivo
                        ubicacion_dentro_de_la_lista = int(archivo.index(oracion_a_comparar))
            if mayor_porcentaje > 0.8:
                mutex.acquire()
                break
        mutex.acquire()
    time.sleep(0.002)
    mutex.release()
    ubicacion_principio = sum(map(len, map(word_tokenize, archivo_donde_se_encontro[:ubicacion_dentro_de_la_lista])))
    ubicacion_fin = ubicacion_principio + len(word_tokenize(oracion_mas_parecida))
    ubicacion_donde_se_encontro = f"({ubicacion_principio}, {ubicacion_fin})"
    porcentajes_de_aparicion_internet.append((oracion, oracion_mas_parecida, mayor_porcentaje, url_donde_se_encontro))
```

Figura 17. Función para obtener plagio de Internet.

Finalmente se generó pruebas unitarias (véase **Tabla 7**) con textos extraídos de la web, con el objetivo de verificar el funcionamiento de las funciones implementadas para el análisis en documentos.

Tabla 7. Casos de Prueba para el modelo.

Caso de prueba	Descripción	Texto 1	Texto 2	Resultado esperado																		
Caso 1	Plagio completo: ambos textos son idénticos	"Prueba 1"	"Prueba 2"	<p>Tópicos del texto: <i>fruta, verdura, riesgo, salud, enfermedad</i></p> <p>Análisis de plagio Total de 21 plagios encontrados Porcentaje de plagio general: 100%</p> <table border="1"> <thead> <tr> <th>Oración plagiada</th> <th>Oración original</th> <th>Lugar donde se encontró</th> </tr> </thead> <tbody> <tr> <td>Las frutas y verduras son una parte esencial de una dieta sana</td> <td>Las frutas y verduras son una parte importante de una dieta sana</td> <td>Prueba 2.pdf</td> </tr> <tr> <td>Estan repletas de vitaminas, minerales y antioxidantes que pueden ayudarle a proteger su cuerpo de las enfermedades</td> <td>Estan repletas de vitaminas, minerales y antioxidantes que pueden ayudarle a proteger su cuerpo de las enfermedades</td> <td>Prueba 2.pdf</td> </tr> <tr> <td>Comer fruta y verdura tambien puede ayudarte a mantener un peso saludable, reducir el riesgo de padecer enfermedades cardiacas, derrames cerebrales y cancer, y mejorar tu salud y bienestar general</td> <td>Comer frutas y verduras tambien puede ayudarte a mantener un peso saludable, reducir el riesgo de padecer enfermedades cardiacas, derrames cerebrales y cancer, y mejorar tu salud y bienestar general</td> <td>Prueba 2.pdf</td> </tr> <tr> <td>Comer frutas y verduras tiene muchos beneficios</td> <td>Comer frutas y verduras tiene muchos beneficios</td> <td>Prueba 2.pdf</td> </tr> <tr> <td>Mejora de la salud del corazon: Las frutas y verduras son bajas en grasas saturadas y colesterol, y tienen un alto contenido en fibra</td> <td>Mejora de la salud del corazon: Las frutas y verduras son bajas en grasas saturadas y colesterol, y tienen un alto contenido en fibra</td> <td>Prueba 2.pdf</td> </tr> </tbody> </table>	Oración plagiada	Oración original	Lugar donde se encontró	Las frutas y verduras son una parte esencial de una dieta sana	Las frutas y verduras son una parte importante de una dieta sana	Prueba 2.pdf	Estan repletas de vitaminas, minerales y antioxidantes que pueden ayudarle a proteger su cuerpo de las enfermedades	Estan repletas de vitaminas, minerales y antioxidantes que pueden ayudarle a proteger su cuerpo de las enfermedades	Prueba 2.pdf	Comer fruta y verdura tambien puede ayudarte a mantener un peso saludable, reducir el riesgo de padecer enfermedades cardiacas, derrames cerebrales y cancer, y mejorar tu salud y bienestar general	Comer frutas y verduras tambien puede ayudarte a mantener un peso saludable, reducir el riesgo de padecer enfermedades cardiacas, derrames cerebrales y cancer, y mejorar tu salud y bienestar general	Prueba 2.pdf	Comer frutas y verduras tiene muchos beneficios	Comer frutas y verduras tiene muchos beneficios	Prueba 2.pdf	Mejora de la salud del corazon: Las frutas y verduras son bajas en grasas saturadas y colesterol, y tienen un alto contenido en fibra	Mejora de la salud del corazon: Las frutas y verduras son bajas en grasas saturadas y colesterol, y tienen un alto contenido en fibra	Prueba 2.pdf
Oración plagiada	Oración original	Lugar donde se encontró																				
Las frutas y verduras son una parte esencial de una dieta sana	Las frutas y verduras son una parte importante de una dieta sana	Prueba 2.pdf																				
Estan repletas de vitaminas, minerales y antioxidantes que pueden ayudarle a proteger su cuerpo de las enfermedades	Estan repletas de vitaminas, minerales y antioxidantes que pueden ayudarle a proteger su cuerpo de las enfermedades	Prueba 2.pdf																				
Comer fruta y verdura tambien puede ayudarte a mantener un peso saludable, reducir el riesgo de padecer enfermedades cardiacas, derrames cerebrales y cancer, y mejorar tu salud y bienestar general	Comer frutas y verduras tambien puede ayudarte a mantener un peso saludable, reducir el riesgo de padecer enfermedades cardiacas, derrames cerebrales y cancer, y mejorar tu salud y bienestar general	Prueba 2.pdf																				
Comer frutas y verduras tiene muchos beneficios	Comer frutas y verduras tiene muchos beneficios	Prueba 2.pdf																				
Mejora de la salud del corazon: Las frutas y verduras son bajas en grasas saturadas y colesterol, y tienen un alto contenido en fibra	Mejora de la salud del corazon: Las frutas y verduras son bajas en grasas saturadas y colesterol, y tienen un alto contenido en fibra	Prueba 2.pdf																				

Caso 2 No es plagio: "Prueba "Prueba
 los textos son 1" 2"
 completamente
 diferentes

Tópicos del texto: *cerebro, lóbulo, cuerpo, órgano, información*

Análisis de plagio

Total de 0 plagios encontrados

Porcentaje de plagio general: 0%

Oración plagiada	Oración original	Lugar donde se encontró
------------------	------------------	-------------------------

Caso 3 Plagio parcial: "Prueba "
 los textos 1" Prueba
 tienen similitud 2"
 pero no son
 idénticos

Tópicos del texto: *fruta, verdura, riesgo, salud, enfermedad*

Análisis de plagio

Total de 21 plagios encontrados

Porcentaje de plagio general: 80%

Oración plagiada	Oración original	Lugar donde se encontró
Las frutas y verduras son una parte importante de una dieta sana	Las frutas y verduras son una parte esencial de una dieta sana	Prueba1.pdf
Estan repletas de vitaminas, minerales y antioxidantes que pueden ayudarle a proteger su cuerpo de las enfermedades	Estan repletas de vitaminas, minerales y antioxidantes que pueden ayudarle a proteger su cuerpo de las enfermedades	Prueba1.pdf
Comer frutas y verduras tambien puede ayudarte a mantener un peso saludable, reducir el riesgo de padecer enfermedades cardiacas, derrames cerebrales y cancer, y mejorar tu salud y bienestar general	Comer fruta y verdura tambien puede ayudarte a mantener un peso saludable, reducir el riesgo de padecer enfermedades cardiacas, derrames cerebrales y cancer, y mejorar tu salud y bienestar general	Prueba1.pdf
Comer frutas y verduras tiene muchos beneficios	Comer frutas y verduras tiene muchos beneficios	Prueba1.pdf
Mejora de la salud del corazon: Las frutas y verduras son bajas en grasas saturadas y colesterol, y tienen un alto contenido en fibra	Mejora de la salud del corazon: Las frutas y verduras son bajas en grasas saturadas y colesterol, y tienen un alto contenido en fibra	Prueba1.pdf
Esto las convierta en una	Esto las convierta en una	

Caso 4 Textos similares: los textos tienen una alta similitud, pero no son plagio "Prueba 1" "Prueba 2"

Tópicos del texto: *cerebro, lobulo, organo, cuerpo, informacion*

Análisis de plagio

Total de 5 plagios encontrados

Porcentaje de plagio general: 16%

Oración plagiada	Oración original	Lugar donde se encontró
El cerebro humano es un <u>organ</u> altamente complejo y fundamental en nuestro cuerpo	El cerebro humano es un <u>organ</u> complejo y fascinante	Prueba 3.pdf
Es responsable de una amplia gama de funciones, desde el procesamiento de pensamientos y emociones hasta el control de nuestros <u>movimientos y sentidos</u>	Es responsable de todo, desde nuestros pensamientos y emociones hasta nuestros movimientos y sentidos	Prueba 3.pdf
<u>Aqui</u> te presento algunos datos adicionales sobre el cerebro:	He <u>aqui</u> algunos datos adicionales sobre el cerebro:	Prueba 3.pdf
El cerebro tiene un peso aproximado de 1,5 kg, representando alrededor del 2% de nuestro peso <u>corporal</u>	El cerebro pesa alrededor de 1,5 kg y constituye aproximadamente el 2% del peso corporal	Prueba 3.pdf
A lo largo de nuestra vida, el cerebro <u>continua</u> cambiando y <u>adaptandose</u>	El cerebro sigue cambiando y <u>adaptandose</u> a lo largo de nuestra vida	Prueba 3.pdf

Actividad 1.2: Utilizar redes neuronales con Scikit-Learn para identificar coincidencias dentro los textos.

Esta actividad se inicializo con una búsqueda de los tipos de Redes Neuronales, la misma que consistió en:

- Realizar una búsqueda de los diversos tipos de redes neuronales que podrían ser utilizadas en el Trabajo propuesto (véase sección **Tipos de Redes Neuronales en el Procesamiento de Lenguaje Natural**).
- Seleccionar la red neuronal: En base a una tabla descriptiva realizada (véase sección **Tabla Descriptiva de los tipos de redes neuronales**.) donde se detalla las características, ventajas y aplicaciones de los diferentes tipos de redes neuronales, se decidió trabajar con la Red Neuronal Siamesa, para más información sobre esta red neuronal (véase sección **Redes neuronales Siamesas**).

A si mismo se realizó una búsqueda de las diversas librerías y frameworks para trabajar con redes neuronales, mediante una tabla descriptiva se detalla las características, facilidad de uso, modelos pre-entrenados y lenguaje de programación con los que trabaja cada una de estas librerías (véase sección **Descripción de las Librerías para trabajar con Redes Neuronales**).

Posteriormente con el objetivo de crear un modelo que permita determinar similitudes en textos, se diseñó la Arquitectura de la Red Neuronal Siamesa (véase la **Figura 18**) la misma que se compone de tres fases principales:

- **Fase de entrada (Input):** En esta fase se introducen los datos (texto1 y texto2) en la red. Representa la fase inicial, cuando los datos son recibidos y procesados por las capas subsiguientes.
- **Fase de procesamiento (Processing):** En esta fase, los datos de entrada pasan por varias capas, se transforman y se manipulan para aprender representaciones significativas. La capa de incrustación, las capas de aplanamiento, la capa de concatenación, la capa de deserción y las capas densas, son algunas de las capas involucradas en esta fase. Para hacer predicciones o capturar medidas de documentos plagiados o no plagiados, estas capas procesan colectivamente los datos de entrada, extraen características pertinentes y realizan cálculos.
- **Fase de resultados (Output):** Es la conclusión del modelo de red neuronal, la produce la última capa densa, que utiliza las representaciones aprendidas en la fase de procesamiento para producir la salida deseada. La fase de resultados proporciona las probabilidades de clase de predicciones binarias en un problema de clasificación binaria.

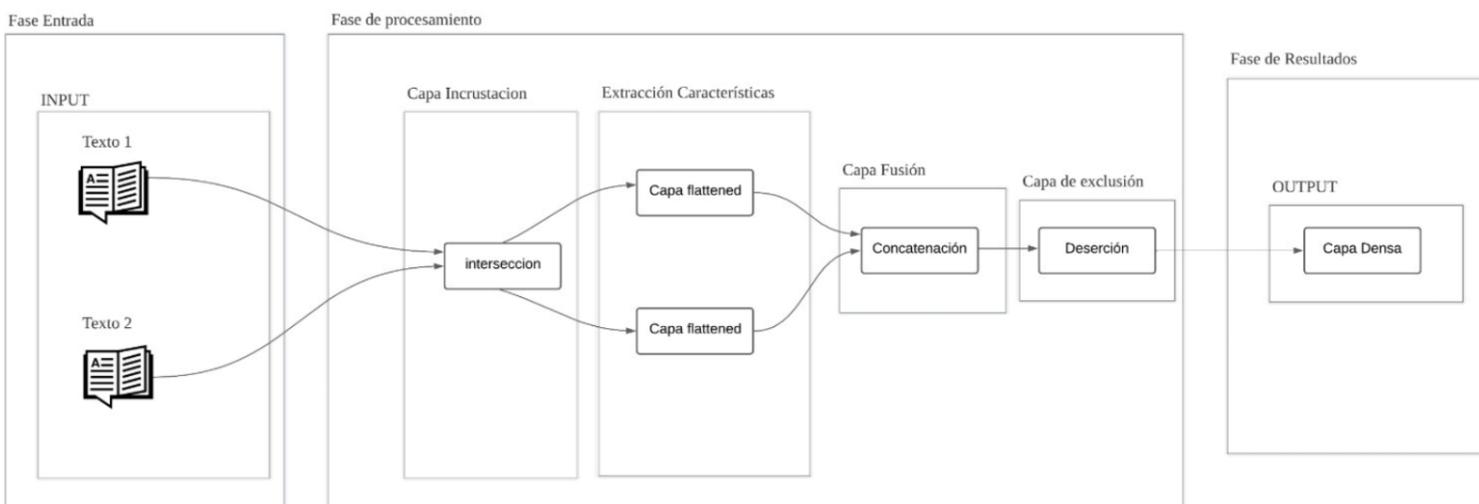


Figura 18. Arquitectura de la Red Neuronal Siamesa [Fuente propia].

Seguidamente se generó un dataset de textos para entrenar a la red neuronal, lo que implicó la recopilación de datos donde incluyen una variedad de textos originales y potencialmente plagiados.

- Se obtuvieron textos originales del repositorio DSpace de la Universidad Nacional de Loja (UNL) de la carrera de Computación (véase **Figura 19**), así como de la biblioteca digital IEEE Xplore (véase **Figura 20**), para este caso se realizó una búsqueda avanzada mediante la siguiente cadena de búsqueda: ("All Metadata":neural networks) AND ("All Metadata":machine learning) AND ("All

Repositorio Digital - Universidad Nacional de Loja / Facultad de Energía, las Industrias y los Recursos Naturales no Renovables

TRABAJOS DE TITULACION

AEIRNNR : [1408] Página de inicio de la colección

Buscar

Fecha Publicación Autor Título Materia

Suscribirse para recibir un correo electrónico cada vez que se introduzca un ítem en esta colección. [Suscribir](#) [RSS](#) [L.O.](#) [RSS](#) [L.O.](#) [RSS](#)

Elementos (mostrados por Fecha de envío en Descendente orden): 1 a 20 de 1408 [Siguiete >](#)

Fecha de publicación	Título	Autor(es)
26-jun-2023	Diseño de una red SOHO segura mediante el empleo de un honeypot en la red de la empresa SIITE para mitigar ataques cibernéticos	<i>Tucker Yepez, John Jossimar; González Ortega, Wellington Rafael</i>
27-jun-2023	Modelo de red neuronal transformer para responder preguntas frecuentes acerca de becas e incentivos de los estudiantes de la Carrera de Computación de la Universidad Nacional de Loja Transformer neural network model to answer frequently asked questions about Scholarships and Incentives for students of the Computing Career of the Universidad Nacional de Loja	<i>Cumbicus Pineda, Oscar Miguel; Quevedo Reyes, Evelyn Juliana</i>
26-jun-2023	Sistema de irrigación utilizando IoT para cultivos de la Universidad Nacional de Loja Irrigation system using IoT for crops at the National University of Loja.	<i>Guamán Quinche, Edwin René; Minga Medina, Angel Favian; Nole Reyes, Alex Santiago</i>

Otras opciones relacionadas

Autor

- Cabrera Abad, Jaime Geovanny (3)
- Aguilar Betancourt, Julia Patricia (2)
- Blacio Aguilar, Milton Gualberto (2)
- Cañar Lalangui, Janeth Alexandra,... (2)
- Chicaiza Songor, Gabriela Paulina (2)
- Chuico Muñoz, Alex Lenin, Miguel ... (2)
- Cuenca Satama, Anderson Rigoberto (2)
- Flores Gaona, Carlos Eduardo (2)
- González Capa, María Paola (2)
- González Ortega, Wellington Rafael (2)

[next >](#)

Título

Metadata":natural language processing).

Figura 19. DSpace de la Universidad Nacional de Loja.

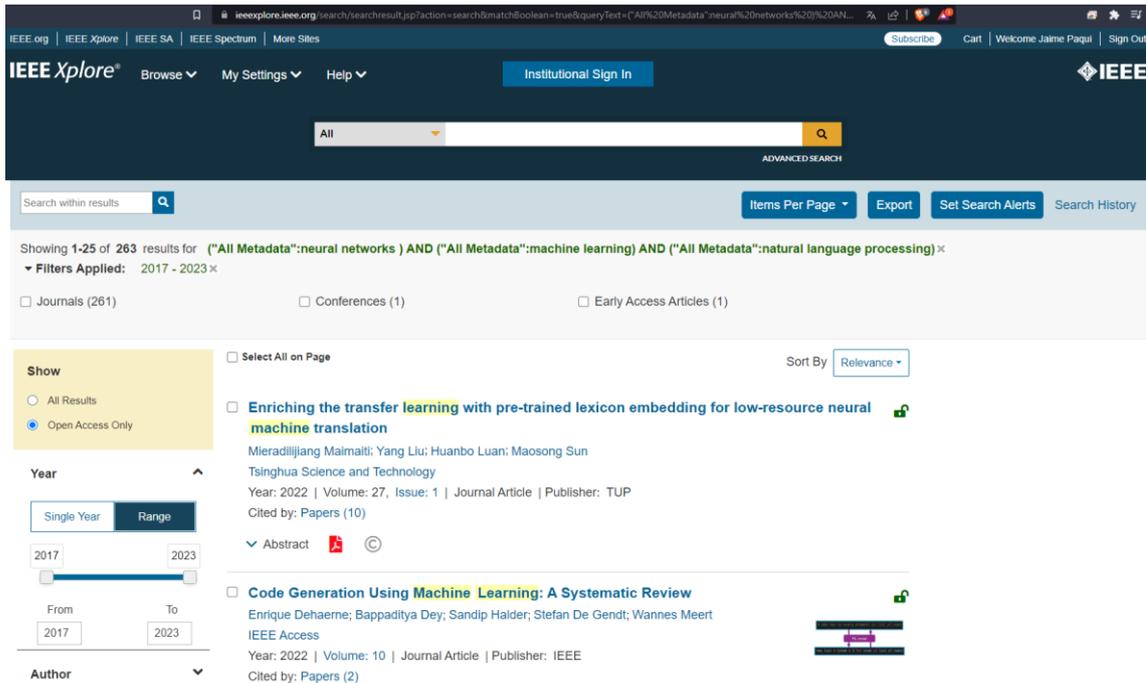


Figura 20. Biblioteca digital IEEE Xplore.

- Se extrajo el texto de los documentos descargados utilizando la herramienta Nitro (véase Figura 21). Es importante destacar que, en el caso de los documentos del IEEE Xplore, fue necesario traducirlos al español, ya que todos los documentos de dicha biblioteca están en inglés, para ello se utilizó la herramienta web DocTranslator (véase Figura 22).

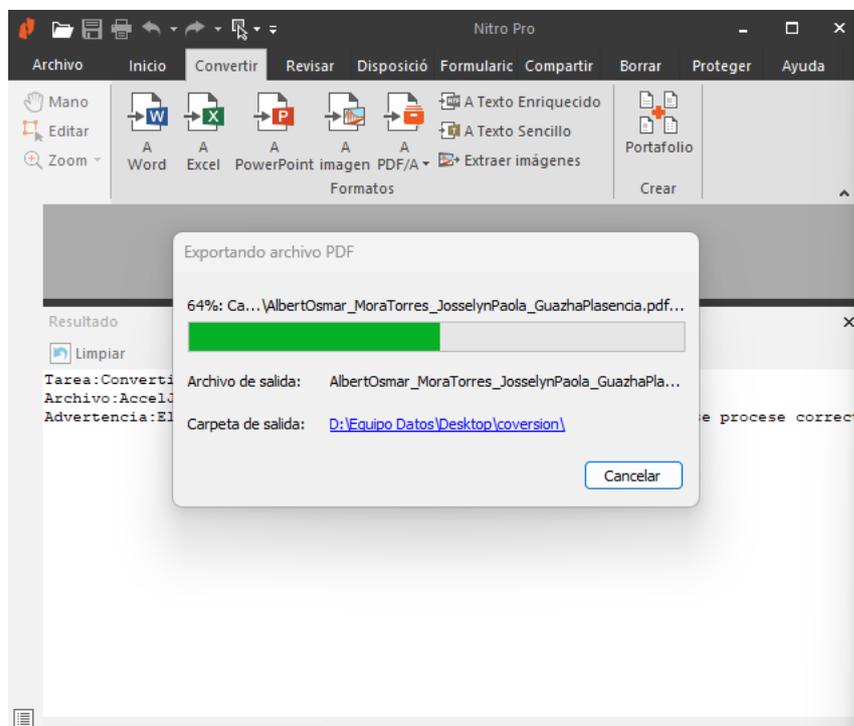


Figura 21. Herramienta Nitro.

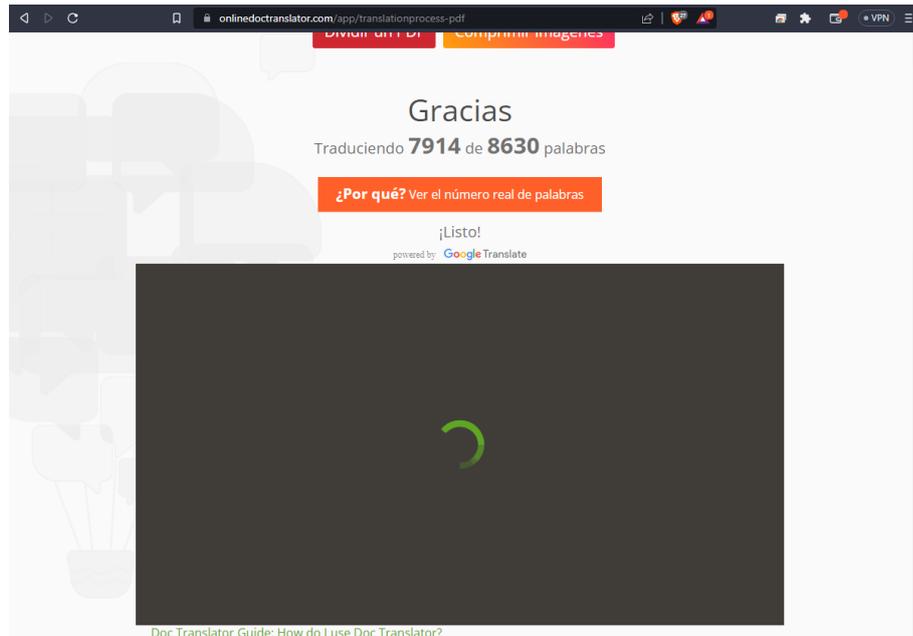


Figura 22. Herramienta web DocTranslator.

- Se generaron dos conjuntos diferentes de datos con los textos extraídos: uno que consistió en un 80% de documentos parafraseados y otro que representó el 20%. Esto permitió contar con una mayor variedad de valores de prueba en el dataset. Para realizar el parafraseo se utilizó la herramienta web QuillBot (véase **Figura 23**).

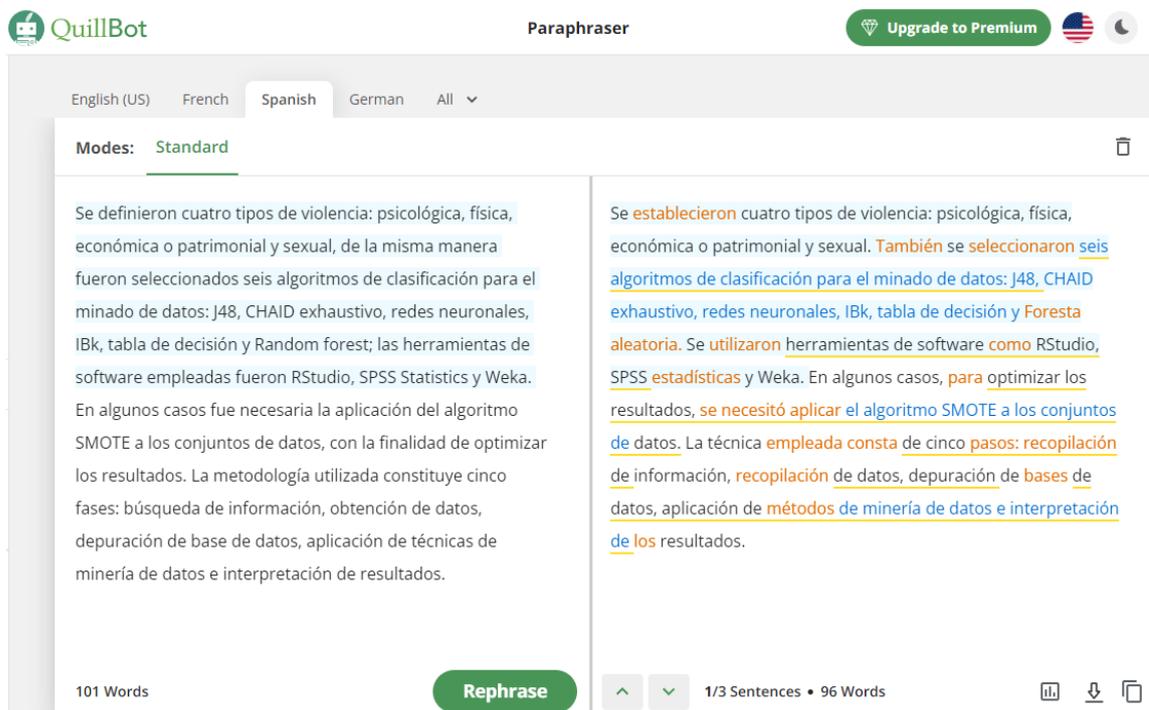


Figura 23. Herramienta web QuillBot.

Posteriormente se codificó la Red Neuronal Siamesa, lo que implicó la importación de Librerías y Módulos: Las librerías que se utilizaron para la codificación de la red neuronal son: Scikit-learn, numpy, googlesearch, bs4 (BeautifulSoup), nltk, TfidfVectorizer (de sklearn.feature_extraction.text), metodos_de_similitud, helper y procesamiento_de_archivos. Para más información de estas librerías (véase sección **Librerías y Frameworks**).

- **metodos_de_similitud**: Módulo personalizado que contiene métodos para calcular la similitud entre textos.
- **helper**: Módulo personalizado que contiene funciones y variables de apoyo para el procesamiento de archivos y modelos.
- **procesamiento_de_archivos**: Módulo personalizado que contiene funciones para la limpieza y procesamiento de archivos de texto.
- **TfidfVectorizer (de sklearn.feature_extraction.text)**: Implementa la técnica de vectorización TF-IDF para convertir texto en características numéricas.

La función **generar_pares_textos** (véase **Figura 24**) representa un método apto para la detección de similitud entre textos a partir de un conjunto de textos. Utiliza la medida de similitud del coseno para evaluar la proximidad semántica entre los textos, generando pares de documentos con sus respectivos puntajes de similitud. Estos resultados se almacenan en un archivo de texto denominado "pares_textos.txt". El código utiliza la clase TfidfVectorizer de scikit-learn para transformar los documentos de texto en vectores TF-IDF, y posteriormente calcula la similitud del coseno entre estos vectores.

```
def generar_pares_textos(pares_textos_path, documentos): #fuera
#cuidado, posible implementacion de deteccion de plagio para proseguir
sample_files = [doc.nombre + doc.extension for doc in documentos]
sample_contents = ["".join(doc.texto) for doc in documentos]
sample_contents_lemmatized = sample_contents

vectorize = lambda Text: TfidfVectorizer(max_features=10000, ngram_range=(1, 2), sublinear_tf=True, smooth_idf=True).fit_transform(Text).toarray()
similarity = lambda doc1, doc2: cosine_similarity([doc1, doc2])

vectors = vectorize(sample_contents_lemmatized)
s_vectors = list(zip(sample_files, vectors))

results = set()
for sample_a, text_vector_a in s_vectors:
    new_vectors = s_vectors.copy()
    current_index = new_vectors.index((sample_a, text_vector_a))
    del new_vectors[current_index]
    for sample_b, text_vector_b in new_vectors:
        sim_score = similarity(text_vector_a, text_vector_b)[0][1]
        sample_pair = sorted((sample_a, sample_b))
        score = sample_pair[0], sample_pair[1], sim_score
        results.add(score)

with open(os.path.join(pares_textos_path, "pares_textos.txt"), 'w') as f:
    for data in results:
        print(data, file=f)
```

Figura 24. Función para generar pares de textos.

La función `leer_pares_textos` (véase **Figura 25**) aprovecha los datos de la función anterior para crear una lista de tuplas denominada "text_pairs" que almacena información sobre los pares de documentos y sus respectivos puntajes de similitud. Luego, se procede a la extracción de información específica de cada tupla, incluyendo los nombres de los documentos y sus puntajes de similitud. Finalmente, la función devuelve tres listas: "train_text1" y "train_text2" las cuales contienen los textos correspondientes a los pares de documentos, y "train_similarity" que almacena los puntajes de similitud asociados.

```
def leer_pares_textos(pares_textos_path, documentos):
    text_pairs = []
    train_text1 = []
    train_text2 = []
    train_similarity = []
    with open(pares_textos_path + "pares_textos.txt", "r") as file: #ojo
        for line in file:
            line = line.strip().strip('(').replace("'", "")
            texto1, texto2, sim = line.split(", ")
            doc1 = texto1
            doc2 = texto2
            similarity = float(sim)
            text_pairs.append((doc1, doc2, similarity))

    for pair in text_pairs:
        doc1 = pair[0]
        doc2 = pair[1]
        similarity = pair[2]

        # Leer el texto en los documentos
        documento1 = documento_por_nombre(documentos, doc1)
        documento2 = documento_por_nombre(documentos, doc2)

        if documento1 and documento2:
            # Ambos documentos encontrados, se pueden acceder a sus atributos
            text1 = documento1.texto
            text2 = documento2.texto

            train_text1.append(text1)
            train_text2.append(text2)
            train_similarity.append(similarity)
        else:
            # Alguno de los documentos no se encontró, manejar el caso en consecuencia
            print(f"Error: No se encontró el documento para la pareja {doc1} y {doc2}")

    train_similarity = np.array(train_similarity)

    return train_text1, train_text2, train_similarity
```

Figura 25. Función para la lectura de pares de textos.

La función **generar_modelo_entrenado** (véase **Figura 26**) implementa la construcción y el entrenamiento de la red neuronal, en primer lugar, se establece el número máximo de palabras a considerar y se inicializan listas para almacenar los textos y puntajes de similitud provenientes de pares de documentos. La función de entrenamiento "fit" es invocada con los datos de entrenamiento, incluyendo las secuencias de palabras representadas numéricamente y los puntajes de similitud. Finalmente, el modelo entrenado, junto con el tokenizer y los datos de entrenamiento, se almacenan en la lista "modelo_entrenado" para su posterior uso. El código utiliza la biblioteca Keras para construir una red neuronal con capas de Embedding compartidas para representar los textos, seguidas de capas Flatten y Concatenate para fusionar las representaciones. Posteriormente, se aplican capas de Dropout y Dense para la reducción de overfitting y la predicción binaria de similitud, respectivamente. El modelo se compila utilizando la función de pérdida binary_crossentropy y el optimizador Adam.

```
def generar_modelo_entrenado(pares_textos_path):

    # Ingresar el numero maximo de palabras a considerar
    max_words = 10000
    train_text1 = []
    train_text2 = []
    train_similarity = []
    documentos = archivos_referencia_limpios
    generar_pares_textos(pares_textos_path, documentos)
    train_text1, train_text2, train_similarity = leer_pares_textos(pares_textos_path, documentos)

    tokenizer = Tokenizer(num_words=max_words)
    tokenizer.fit_on_texts(train_text1 + train_text2)

    train_sequences1 = tokenizer.texts_to_sequences(train_text1)
    train_sequences2 = tokenizer.texts_to_sequences(train_text2)

    train_data1 = pad_sequences(train_sequences1)
    train_data2 = pad_sequences(train_sequences2)

    embedding_dim = 100 #200
    hidden_units = 128 #256
    dropout_rate = 0.2

    input1 = Input(shape=(train_data1.shape[1],))
    input2 = Input(shape=(train_data2.shape[1],))

    shared_embedding_layer = Embedding(max_words, embedding_dim)

    embedded1 = shared_embedding_layer(input1)
    embedded2 = shared_embedding_layer(input2)

    flattened1 = Flatten()(embedded1)
    flattened2 = Flatten()(embedded2)

    concatenated = Concatenate()([flattened1, flattened2])
    dropout = Dropout(dropout_rate)(concatenated)
    output = Dense(hidden_units, activation='relu')(dropout)
    output = Dense(1, activation='sigmoid')(output)
```

Figura 26. Función para generar el modelo entrenado.

La función **calcular_similitud** (véase **Figura 27**) representa la aplicación práctica del modelo de aprendizaje, extrae el modelo, el tokenizer, y los datos de entrenamiento almacenados previamente en la lista "modelo_entrenado" para tokenizar los textos de entrada, convierte las secuencias de palabras en datos de entrada compatibles con el modelo mediante la técnica de padding, y finalmente, utiliza el modelo para predecir la similitud entre los dos textos dados.

```
def calcular_similitud( text1, text2):
    model, tokenizer, train_data1, train_data2 = modelo_entrenado[0]
    text1_sequence = tokenizer.texts_to_sequences([text1])
    text2_sequence = tokenizer.texts_to_sequences([text2])
    text1_data = pad_sequences(text1_sequence, maxlen=train_data1.shape[1])
    text2_data = pad_sequences(text2_sequence, maxlen=train_data2.shape[1])

    similarity = model.predict([text1_data, text2_data])[0][0]
    return similarity
```

Figura 27. Función para calcular la similitud.

La explicación de los valores específicos utilizados para las funciones que se puede apreciar en la **Figura 24** y **Figura 26**, lo puede visualizar en el **Anexo 2**, junto con la descripción de las funciones principales de la red neuronal siamesa. Así mismo para conocer la descripción de las Capas implementadas de esta red neuronal (visite el **Anexo 2** en la sección "Descripción de las Capas Codificadas").

Finalmente se utilizó el dataset generado para las pruebas unitarias. Para hacer más sencillo el control de los documentos se los llamó documento (1), documento (2), etc. Usando hasta una cantidad de 35 documentos inicialmente para generar nuestro dataset inicial. Una vez que se obtuvo un dataset inicial, se procedió a obtener los pares de textos de todos los documentos. Los pares de textos representan una combinación de 2 elementos sin repetición en la que se agrega adicionalmente la similitud que existe entre ambos documentos. Para obtener la similitud inicial se hizo uso de la similitud de coseno (véase **Figura 28**) utilizando los métodos que proporciona scikit-learn, dándonos así la base del entrenamiento del modelo de red neuronal a usar.

```

def check_plagiarism_and_save_results(documentos):
    sample_files = [doc.nombre + doc.extension for doc in documentos]
    sample_contents = [doc.texto for doc in documentos]
    sample_contents_lemmatized = sample_contents

    vectorize = lambda Text: TfidfVectorizer(max_features=10000, ngram_range=(1, 2), sublinear_tf=True, smooth_idf=True).fit_transform(Text).toarray()
    similarity = lambda doc1, doc2: cosine_similarity([doc1, doc2])

    vectors = vectorize(sample_contents_lemmatized)
    s_vectors = list(zip(sample_files, vectors))

    results = set()
    for sample_a, text_vector_a in s_vectors:
        new_vectors = s_vectors.copy()
        current_index = new_vectors.index((sample_a, text_vector_a))
        del new_vectors[current_index]
        for sample_b, text_vector_b in new_vectors:
            sim_score = similarity(text_vector_a, text_vector_b)[0][1]
            sample_pair = sorted((sample_a, sample_b))
            score = sample_pair[0], sample_pair[1], sim_score
            results.add(score)

oraciones_list = texto_oraciones.split('\n')
vectorize_oraciones = vectorize(oraciones_list)
pair_similarities = []
for i in range(len(oraciones_list)):
    for j in range(i+1, len(oraciones_list)):
        sim_score = similarity(vectorize_oraciones[i], vectorize_oraciones[j])[0][1]
        pair_similarity = oraciones_list[i], oraciones_list[j], sim_score
        pair_similarities.append(pair_similarity)

with open('pares_oraciones.txt', 'w', encoding='utf-8') as f:
    for data in pair_similarities:
        print(data, file=f)
with open('pares_textos.txt', 'w') as f:
    for data in results:
        print(data, file=f)

```

Figura 28. Similitud usando el método del coseno.

Una vez que se obtenga los pares de textos, todos con su respectiva similitud como se puede visualizar en la **Figura 29**, se podrá comenzar a darle los datos a la red neuronal y generar su modelo.



```

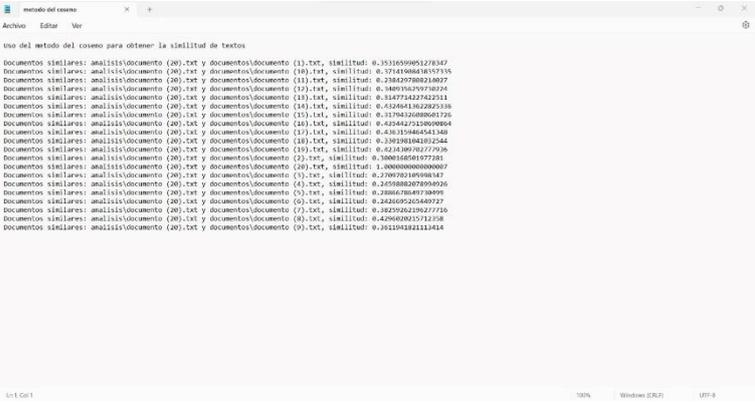
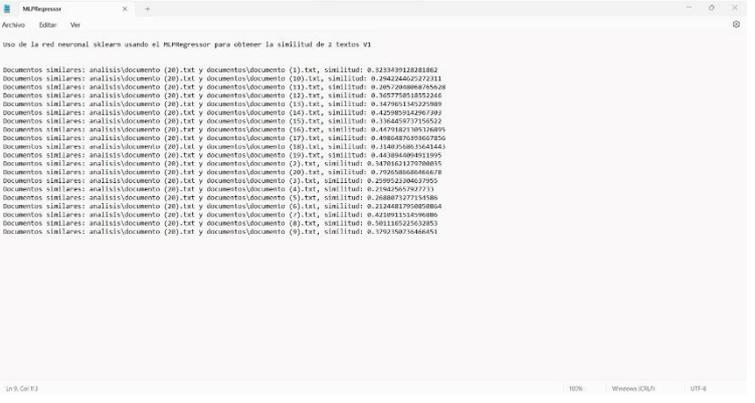
('documento (13).txt', 'documento (18).txt', 0.07163613386893444)
('documento (20).txt', 'documento (7).txt', 0.06813165699637105)
('documento (17) - copia.txt', 'documento (3).txt', 0.04559440026169893)
('documento (13).txt', 'documento (2).txt', 0.035944485270378884)
('documento (12) - copia.txt', 'documento (5).txt', 0.08746913538240711)
('documento (14).txt', 'documento (9) - copia.txt', 0.08053174867897675)
('documento (18).txt', 'documento (19).txt', 0.0876110799421545)
('documento (19).txt', 'documento (3).txt', 0.04714171578362813)
('documento (16).txt', 'documento (19).txt', 0.06940164242705148)
('documento (13).txt', 'documento (6).txt', 0.03935071469504841)
('documento (12).txt', 'documento (7).txt', 0.05623732759478666)
('documento (14) - copia.txt', 'documento (18).txt', 0.06759342710977195)
('documento (4).txt', 'documento (9) - copia.txt', 0.05971970278226339)
('documento (4) - copia.txt', 'documento (4).txt', 0.9999999999999997)
('documento (1).txt', 'documento (7).txt', 0.07740428610984872)
('documento (19).txt', 'documento (2) - copia.txt', 0.049618287131270145)
('documento (13) - copia.txt', 'documento (9) - copia.txt', 0.15676672329640004)
('documento (1).txt', 'documento (9).txt', 0.09544230715012683)
('documento (7).txt', 'documento (9).txt', 0.12212994317705465)
('documento (11) - copia.txt', 'documento (3) - copia.txt', 0.03498164824889194)
('documento (13).txt', 'documento (7) - copia.txt', 0.13412721206806336)
('documento (16).txt', 'documento (7).txt', 0.05466276137609114)
('documento (15).txt', 'documento (3) - copia.txt', 0.039153543104604595)
('documento (2) - copia.txt', 'documento (9) - copia.txt', 0.033644293980617475)
('documento (15).txt', 'documento (7) - copia.txt', 0.07954406951479268)
('documento (20).txt', 'documento (5).txt', 0.05973252011236662)
('documento (4).txt', 'documento (8).txt', 0.05626733206795698)
('documento (1).txt', 'documento (11).txt', 0.07966596089155764)
('documento (10).txt', 'documento (16).txt', 0.15428288764331444)
('documento (13) - copia.txt', 'documento (4) - copia.txt', 0.06598479059658444)
('documento (14) - copia.txt', 'documento (8).txt', 0.0799288440240755)
('documento (19).txt', 'documento (7) - copia.txt', 0.13795567485883736)
('documento (1) - copia.txt', 'documento (7).txt', 0.07740428610984872)
('documento (15).txt', 'documento (17).txt', 0.08176649405333933)
('documento (13).txt', 'documento (9).txt', 0.15676672329640004)
('documento (12).txt', 'documento (17) - copia.txt', 0.08977639596542496)
('documento (14).txt', 'documento (6) - copia.txt', 0.028069406347084838)
('documento (12).txt', 'documento (8) - copia.txt', 0.08376573984597077)
('documento (4) - copia.txt', 'documento (5) - copia.txt', 0.031835451006757796)
('documento (8).txt', 'documento (9) - copia.txt', 0.08436232833530698)

```

Figura 29. Documento con los pares textos.

Finalmente, cuando toda la información de pares textos pueda ser usada para la red neuronal, se hicieron diferentes tipos de pruebas (véase **Tabla 8.**) para mostrar la efectividad de los datos.

Tabla 8. Casos de Prueba con red neuronal siamesa.

Caso de prueba	Descripción	Resultado esperado
Caso 1	Generar los valores de similitud usando el método del coseno	 <pre> uso del metodo del coseno para obtener la similitud de textos Documentos similares: analisis\documento (20).txt y documentos\documento (1).txt, similitud: 0.33316599951278347 Documentos similares: analisis\documento (20).txt y documentos\documento (10).txt, similitud: 0.3734388841857335 Documentos similares: analisis\documento (20).txt y documentos\documento (11).txt, similitud: 0.21842778282148077 Documentos similares: analisis\documento (20).txt y documentos\documento (12).txt, similitud: 0.3489336297978224 Documentos similares: analisis\documento (20).txt y documentos\documento (13).txt, similitud: 0.147734227622311 Documentos similares: analisis\documento (20).txt y documentos\documento (14).txt, similitud: 0.433461362282338 Documentos similares: analisis\documento (20).txt y documentos\documento (15).txt, similitud: 0.437842368864318 Documentos similares: analisis\documento (20).txt y documentos\documento (16).txt, similitud: 0.433427212009084 Documentos similares: analisis\documento (20).txt y documentos\documento (17).txt, similitud: 0.433439484541349 Documentos similares: analisis\documento (20).txt y documentos\documento (18).txt, similitud: 0.338181841812544 Documentos similares: analisis\documento (20).txt y documentos\documento (19).txt, similitud: 0.433438932775938 Documentos similares: analisis\documento (20).txt y documentos\documento (2).txt, similitud: 0.330016505772281 Documentos similares: analisis\documento (20).txt y documentos\documento (20).txt, similitud: 1.0000000000000007 Documentos similares: analisis\documento (20).txt y documentos\documento (3).txt, similitud: 0.2300202189598147 Documentos similares: analisis\documento (20).txt y documentos\documento (4).txt, similitud: 0.26588882079594926 Documentos similares: analisis\documento (20).txt y documentos\documento (5).txt, similitud: 0.288627687787809 Documentos similares: analisis\documento (20).txt y documentos\documento (6).txt, similitud: 0.2823695205449727 Documentos similares: analisis\documento (20).txt y documentos\documento (7).txt, similitud: 0.432525242277735 Documentos similares: analisis\documento (20).txt y documentos\documento (8).txt, similitud: 0.429662015731238 Documentos similares: analisis\documento (20).txt y documentos\documento (9).txt, similitud: 0.38233821131414 </pre>
Caso 2	Generar los valores de similitud utilizando una red neuronal Multi-Capa	 <pre> uso de la red neuronal sklearn usando el MLPRegressor para obtener la similitud de 2 textos v1 Documentos similares: analisis\documento (20).txt y documentos\documento (1).txt, similitud: 0.333439113388383 Documentos similares: analisis\documento (20).txt y documentos\documento (10).txt, similitud: 0.28424602272311 Documentos similares: analisis\documento (20).txt y documentos\documento (11).txt, similitud: 0.3872868695638 Documentos similares: analisis\documento (20).txt y documentos\documento (12).txt, similitud: 0.30575781855246 Documentos similares: analisis\documento (20).txt y documentos\documento (13).txt, similitud: 0.34780116229988 Documentos similares: analisis\documento (20).txt y documentos\documento (14).txt, similitud: 0.42282542282393 Documentos similares: analisis\documento (20).txt y documentos\documento (15).txt, similitud: 0.33644273755522 Documentos similares: analisis\documento (20).txt y documentos\documento (16).txt, similitud: 0.407918138138895 Documentos similares: analisis\documento (20).txt y documentos\documento (17).txt, similitud: 0.488487836073505 Documentos similares: analisis\documento (20).txt y documentos\documento (18).txt, similitud: 0.338485881548444 Documentos similares: analisis\documento (20).txt y documentos\documento (19).txt, similitud: 0.443894094931905 Documentos similares: analisis\documento (20).txt y documentos\documento (2).txt, similitud: 0.38786217708095 Documentos similares: analisis\documento (20).txt y documentos\documento (20).txt, similitud: 0.792058486846478 Documentos similares: analisis\documento (20).txt y documentos\documento (3).txt, similitud: 0.22952138637955 Documentos similares: analisis\documento (20).txt y documentos\documento (4).txt, similitud: 0.25842562727731 Documentos similares: analisis\documento (20).txt y documentos\documento (5).txt, similitud: 0.30887077316186 Documentos similares: analisis\documento (20).txt y documentos\documento (6).txt, similitud: 0.2324481759828864 Documentos similares: analisis\documento (20).txt y documentos\documento (7).txt, similitud: 0.4238112408088 Documentos similares: analisis\documento (20).txt y documentos\documento (8).txt, similitud: 0.501116521512853 Documentos similares: analisis\documento (20).txt y documentos\documento (9).txt, similitud: 0.392359784668451 </pre>

Caso 3 Generar los valores de similitud usando la red neuronal

```

Resultado
Archivo Editar Ver

documentos\documento (1).txt: Similarity = 0.4883885085582733
documentos\documento (10).txt: Similarity = 0.6758876442909241
documentos\documento (11).txt: Similarity = 0.2374669760465622
documentos\documento (12).txt: Similarity = 0.20378799736499786
documentos\documento (13).txt: Similarity = 0.22001080214977264
documentos\documento (14).txt: Similarity = 0.14442754638195938
documentos\documento (15).txt: Similarity = 0.32997689187526703
documentos\documento (16).txt: Similarity = 0.2044076770544052
documentos\documento (17).txt: Similarity = 0.35683968871831894
documentos\documento (18).txt: Similarity = 0.1555425226688385
documentos\documento (19).txt: Similarity = 0.4234309702777936
documentos\documento (2).txt: Similarity = 0.2177808987524986
documentos\documento (20).txt: Similarity = 0.598757078886032104
documentos\documento (3).txt: Similarity = 0.2909002103992347
documentos\documento (4).txt: Similarity = 0.34598082078994926
documentos\documento (5).txt: Similarity = 0.2886678649730499
documentos\documento (6).txt: Similarity = 0.2356219266429727
documentos\documento (7).txt: Similarity = 0.2236675265589728
documentos\documento (8).txt: Similarity = 0.3225926819627616
documentos\documento (9).txt: Similarity = 0.3511981831123419
  
```

Caso 4 Generar los valores de similitud usando la red neuronal cambiando las constantes de la red neuronal

```

Resultado
Archivo Editar Ver

uso de la red neuronal cambiando los datos de embedding_size = 500 y hidden_units = 500

Documento 1: Similarity = 0.0007270402405256
Documento 10: Similarity = 0.0000340104041510
Documento 11: Similarity = 0.0000740007400000
Documento 12: Similarity = 0.0000000000000000
Documento 13: Similarity = 0.0012022270792224
Documento 14: Similarity = 0.0000000000000000
Documento 15: Similarity = 0.0017000000000000
Documento 16: Similarity = 0.0000000000000000
Documento 17: Similarity = 0.0000000000000000
Documento 18: Similarity = 0.0010000000000000
Documento 19: Similarity = 0.0000000000000000
Documento 2: Similarity = 0.0010000000000000
Documento 20: Similarity = 0.0010000000000000
Documento 3: Similarity = 0.0010000000000000
Documento 4: Similarity = 0.0010000000000000
Documento 5: Similarity = 0.0010000000000000
Documento 6: Similarity = 0.0010000000000000
Documento 7: Similarity = 0.0010000000000000
Documento 8: Similarity = 0.0010000000000000
Documento 9: Similarity = 0.0010000000000000
  
```

6.2. Objetivo 2: Desarrollar el software anti plagio utilizando Django, aplicando el modelo de software 4+1.

En esta sección se muestra los resultados alcanzados para el segundo objetivo del Trabajo de Titulación, para visualizar más detalladamente visite el **Anexo 6**.

Actividad 1.1: Especificación de Requisitos de Software.

En base al Estándar IEEE830 se identificaron y documentaron los requisitos funcionales (Véase **Tabla 9**) y no funcionales (Véase **Tabla 10**), así como las historias de usuario (véase **Anexo 4**), del software anti plagio.

Tabla 9. Requisitos Funcionales.

Nro. de Requisito	Nombre del requisito
RF01	Registrar usuarios.
RF02	Iniciar sesión
RF03	Validar registro de usuario.

RF04	Ver lista de usuarios.
RF05	Cambiar estado de la cuenta de los usuarios.
RF06	Editar cuentas usuarios.
RF07	Eliminar cuentas usuarios.
RF08	Analizar plagio por el Estudiante
RF09	Analizar plagio por el Docente
RF10	Compartir resultado de análisis de plagio.
RF11	Ver lista de documentos analizados por el usuario.
RF12	Ver resultado de análisis de plagio.
RF13	Editar resultado de análisis de plagio
RF14	Eliminar resultado de análisis de plagio
RF15	Descargar resultado de análisis de plagio.

Tabla 10. Requisitos No Funcionales.

Nro. de Requisito	Nombre del requisito	Descripción
RNF01	Desempeño.	El sistema deberá asegurar el funcionamiento del sistema informático para todos los usuarios, permitiendo la consulta y actualización constante y simultánea de la información almacenada o los registros realizados, sin afectar el tiempo de respuesta.
RNF02	Mantenibilidad	La implementación del sistema deberá tener la capacidad de ser fácilmente modificado o actualizado sin causar problemas o complicaciones adicionales.
RNF03	Usabilidad	El sistema debe tener una interfaz fácil de usar, amigable con el usuario de manera que los usuarios puedan manejarlo sin dificultades.
RNF04	Alerta de errores	El sistema debe ser capaz de mostrar mensajes de error cuando el usuario utilice incorrectamente algunas funcionalidades del software.
RNF05	Seguridad	El sistema deberá limitar el acceso únicamente a usuarios que hayan sido autenticados y que cuenten con los permisos necesarios para utilizar el sistema.
RNF06	Disponibilidad.	El sistema debe estar operativo sin interrupciones durante todos los días de la semana.

Para obtener información más detallada de los requisitos funcionales y no funcionales del sistema, los cuales han sido validados junto con sus correspondientes historias de usuario (véase el **Anexo 4**).

Actividad 1.2: Diseño del Software anti plagio, aplicando el modelo 4 + 1.

En esta actividad se procedió a definir la arquitectura (véase **Anexo 5** sección **Fase de Diseño**.) y los componentes del sistema, se utilizó el modelo de software 4 + 1 para generar las diferentes vistas del software anti plagio: Vista Lógica, Vista de Despliegue, Vista de Procesos, Vista Física y Vista de Escenarios (Véase **Anexo 14**). Así mismo se elaboró un modelo relacional en donde incluyen las tablas con sus respectivos datos y relaciones.

Vista Lógica

- Diagrama de clases.

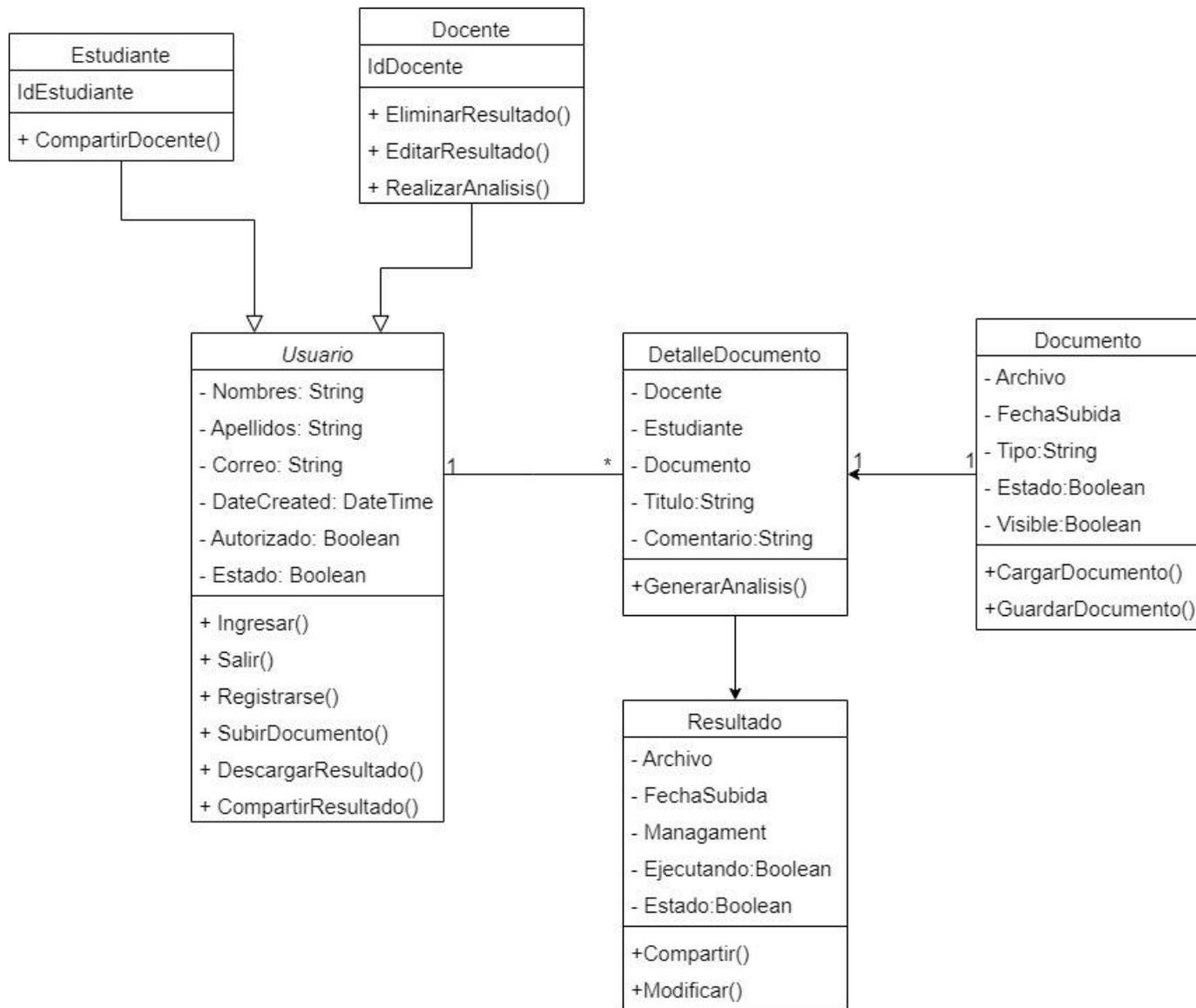


Figura 30. Diagrama de Clases

- Diagrama de Secuencia.

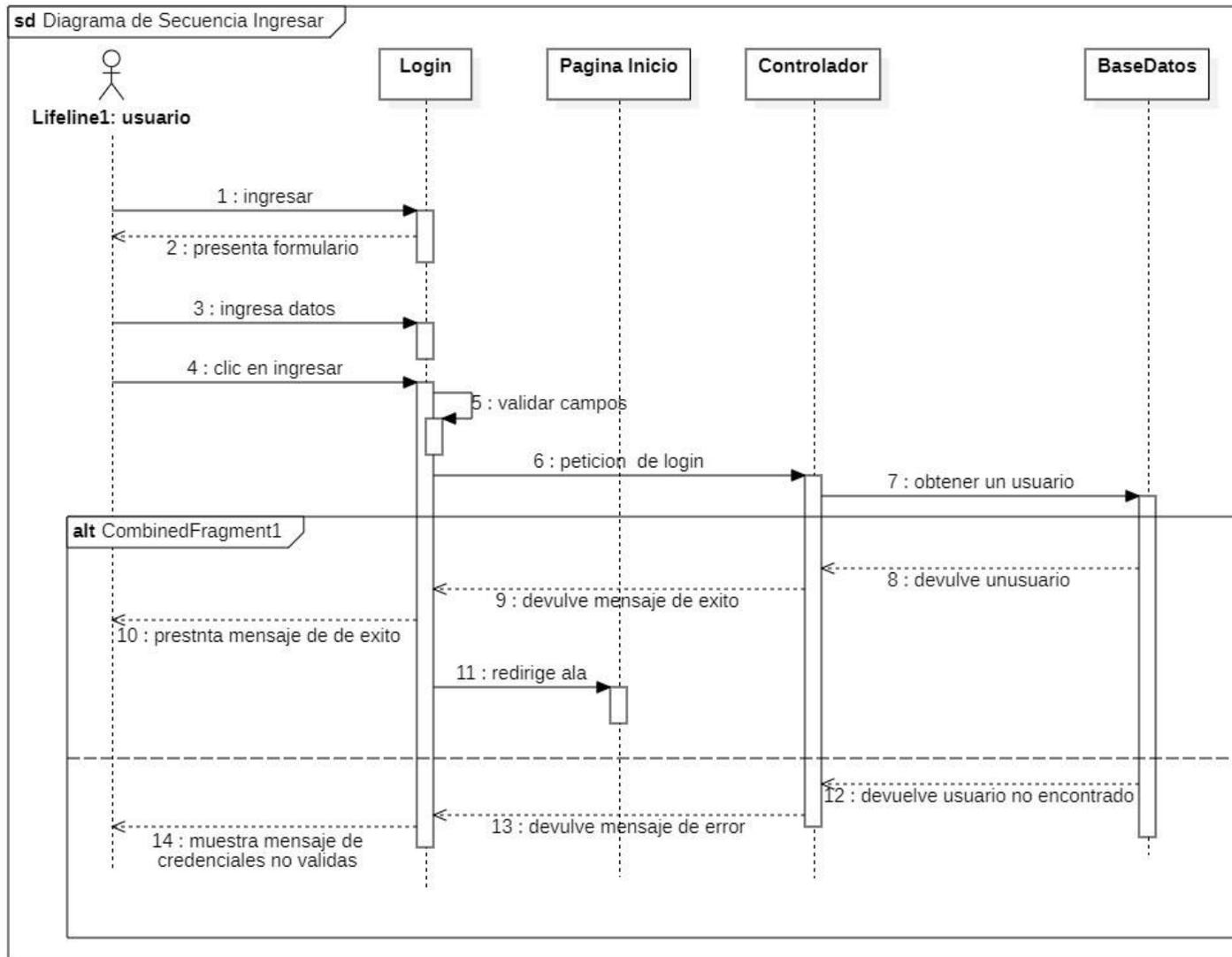


Figura 31. Diagrama de secuencia.

Vista de Despliegue.

- Diagrama de Paquetes.

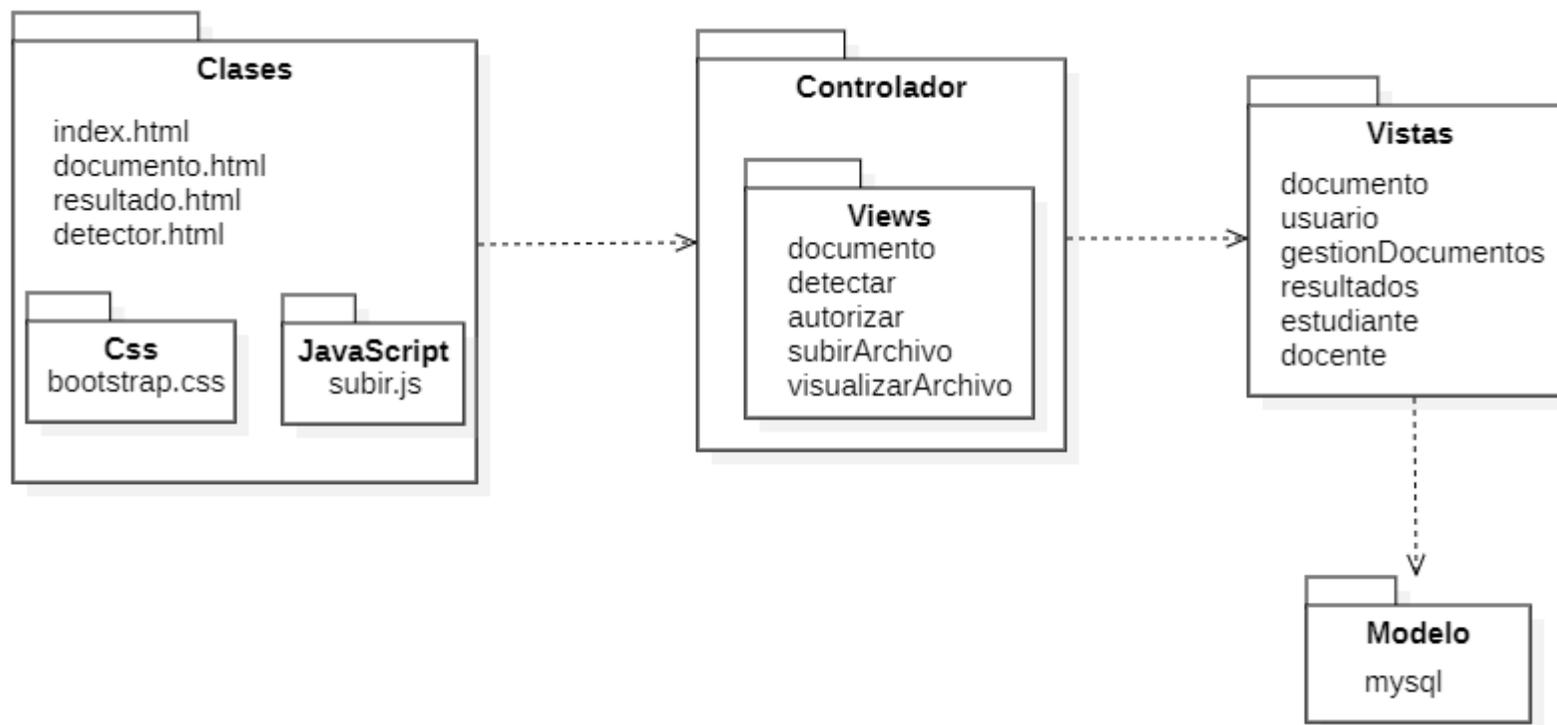


Figura 32. Diagrama de paquetes.

Vista de Procesos.

- Diagrama de Actividades.

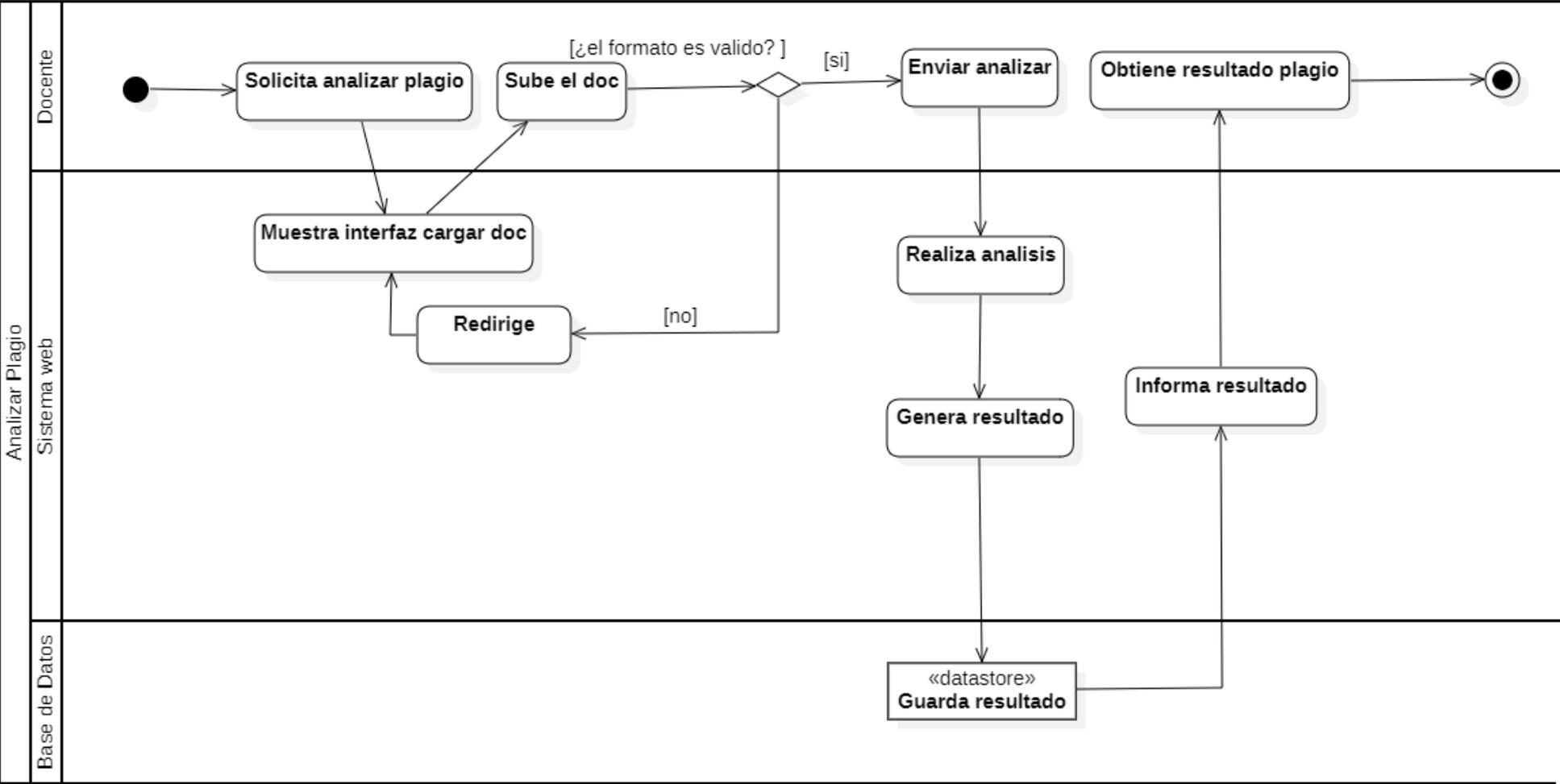


Figura 33. Diagrama de actividades.

Vista Física.

- Diagrama de despliegue.

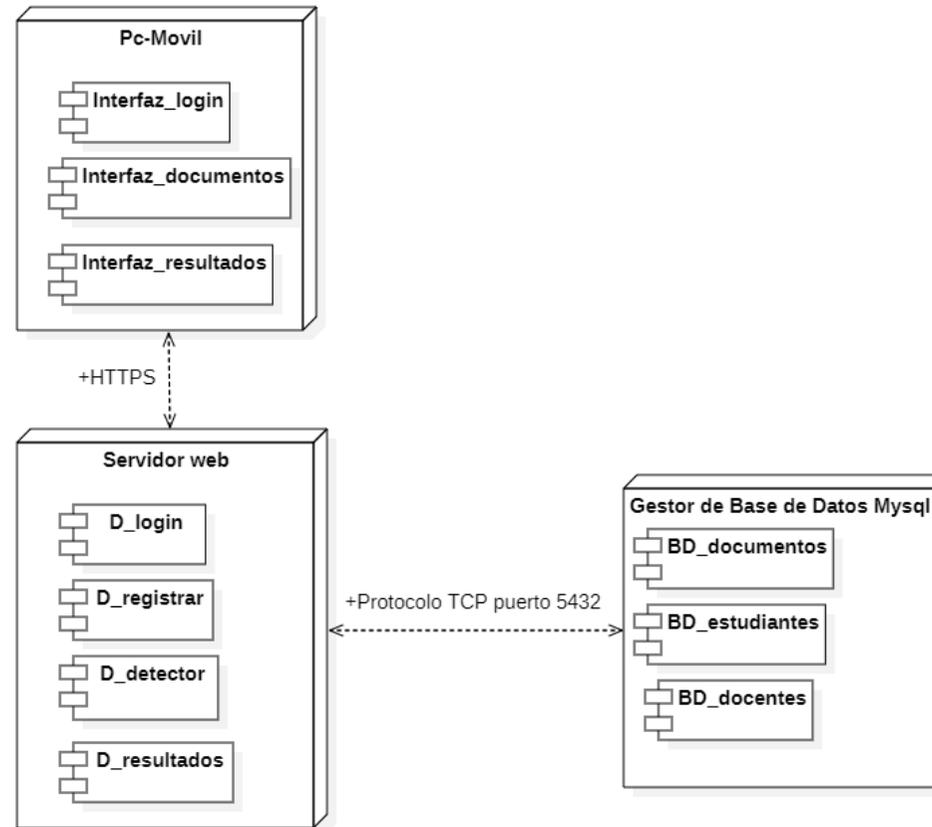


Figura 34. Diagrama de despliegue.

Vista de Escenarios.

- Diagramas de casos de uso.

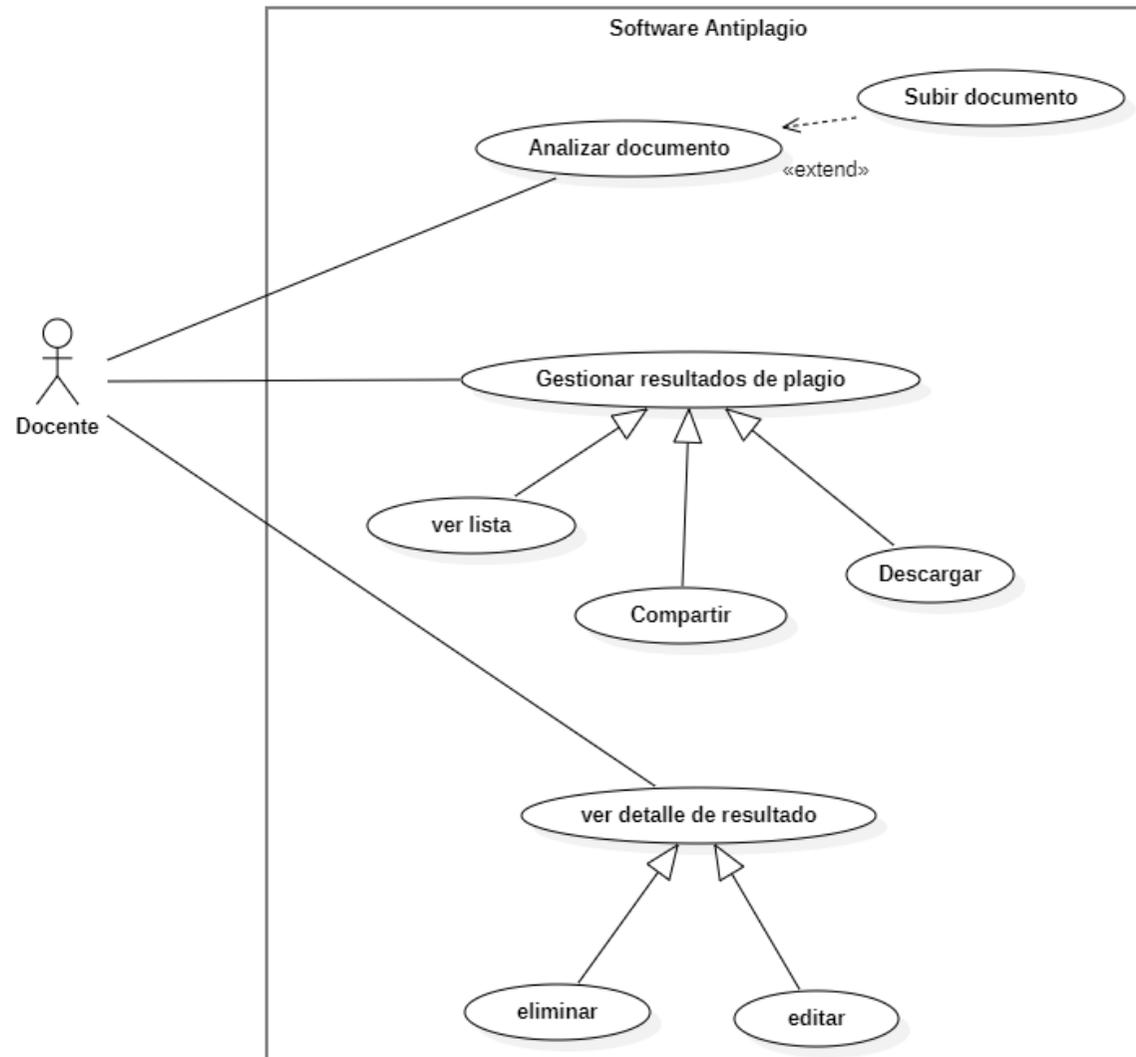


Figura 35. Diagrama de caso de uso Analizar Plagio.

Así mismo, se creó el modelo relacional (véase **Figura 36.**) para la base de datos del sistema de detección de plagio. Este modelo incorpora las tablas con sus datos correspondientes y las relaciones entre ellas.

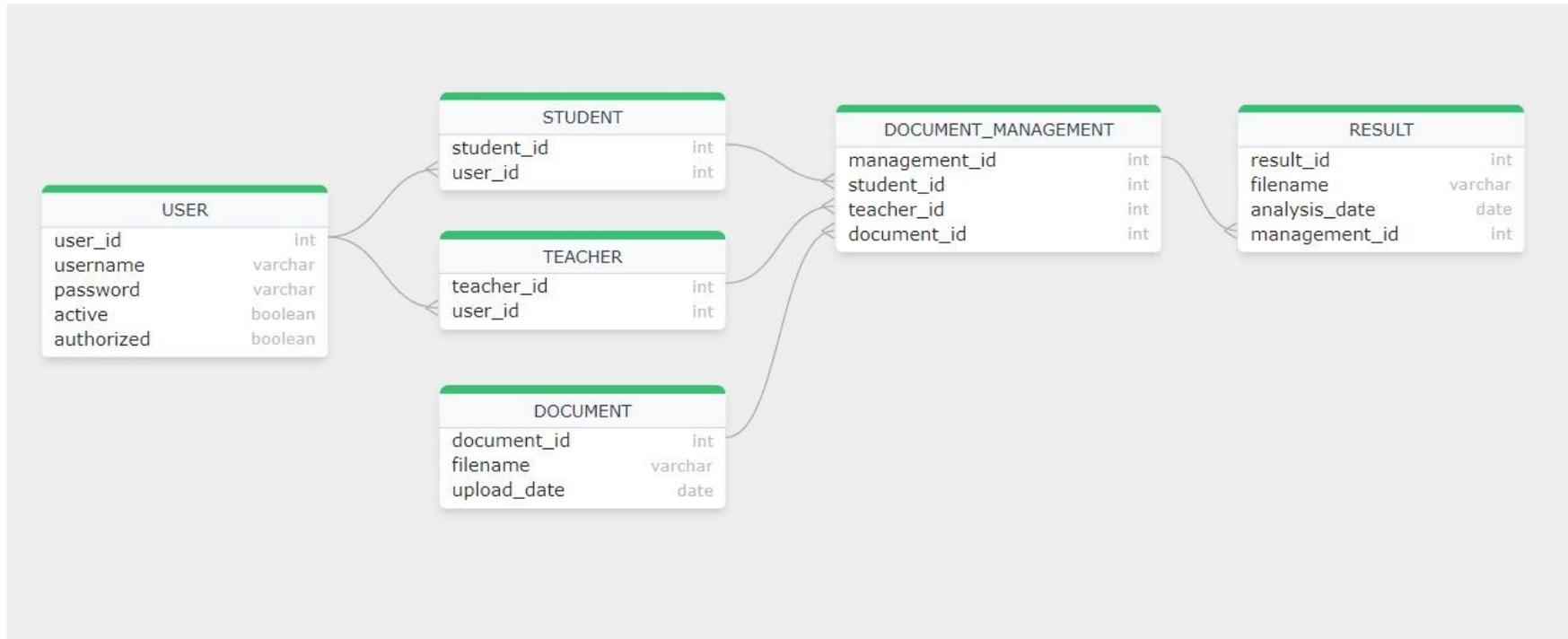


Figura 36. Modelo relacional.

Actividad 1.3: Desarrollo del Software anti plagio, utilizando el framework Django.

En esta actividad se codificó todo el software anti plagio, para ello se presenta una de las funcionalidades implementadas. Además, se incluye una breve explicación sobre la codificación. Para obtener información detallada sobre las demás funcionalidades realizadas en cada iteración, consulte el **Anexo 6** en la sección de Codificación.

El desarrollo del software siguió el patrón de diseño MVT característico de Django, el cual está compuesto de:

1. Modelo: Es responsable de la estructura de la base de datos MySQL, que define cómo se organizan y almacenarán los datos.

En la **Figura 37** se presenta la clase Documento, el mismo que será utilizado para su respectivo análisis de plagio, se coloca la lógica del modelamiento de la base de datos lo que permitirá almacenar el documento.

```
class Documento(models.Model):
    documento_id = models.AutoField(primary_key = True)
    archivo = models.FileField(upload_to='pdfs/')
    fecha_subida = models.DateTimeField(auto_now_add=True)
    # usuario = models.ForeignKey(Usuario, on_delete = models.CASCADE)
    tipo = models.CharField(max_length=50, default='pdf')
    estado = models.BooleanField(default = True) # el estado representa si se
    visible = models.BooleanField(default = False) # el unico documento con es
```

Figura 37. Clase Documento.

2. Vista: Es responsable de obtener los datos necesarios del modelo y decidir qué plantilla se debe usar para presentar la información al usuario.

A si mismo se presenta la implementación para subir un documento (véase **Figura 38.**) para analizar plagio, en donde se define la lógica con la que trabajara el modelo y el template en la que se mostrara al usuario para subir el documento.

```

@login_required
def subir_archivo(request):
    user = request.user
    usuario = Usuario.objects.get(correo=user.email)
    print(usuario.usuario_id)
    if usuario.estado:
        if request.method == 'POST':
            documento = Documento()
            archivo = request.FILES['archivo']
            ext = os.path.splitext(archivo.name)[-1].lower()
            if ext == '.pdf': ...
            elif ext in ['.doc', '.docx']: ...
            elif ext == '.txt': ...
            else: ...

            # documento.usuario = usuario
            documento.archivo = archivo
            documento.visible = True
            gestion = GestionDocumentos()

            if user.groups.filter(name = "docente").exists() or user.groups.filter(name = "admin").exists(): ...

            elif user.groups.filter(name = "estudiante").exists(): ...
            else: ...

            # return redirect(visualizar_archivo, documento.documento_id)
            response_data = {'redirect_url': reverse('visualizar_archivo', args=[gestion.gestion_id])}
            return JsonResponse(response_data)
            # else:
            #     return render(request, 'documento/subir_archivo.html', {'mensaje': 'Usuario no encontrado'}) #ojo
        return render(request, 'documento/subir_archivo.html')
    else :
        return render(request, 'homepage.html')

```

Figura 38. Vista para subir un documento.

3. Plantilla: Se encarga de definir cómo se verá la información que se muestra al usuario en la interfaz web, contiene el diseño y la estructura de la página, permite subir un documento para que posteriormente se guarde en la base de datos. Seguidamente se tiene la implementación (véase **Figura 39**) para subir un documento para su posterior análisis.

```
{% load static %}
<!DOCTYPE html>
<html>

<head>
  <title>Subir Archivo</title>
  <link rel="stylesheet" type="text/css" href="{% static 'css/documento/subir.css' %}">
</head>

<body>
  <div class="container">
    {% if mensaje %}
    <p>{{ mensaje }}</p>
    {% endif %}
    <h1>Subir Archivo</h1>
    <form method="POST" enctype="multipart/form-data">
      {% csrf_token %}
      <input type="file" name="archivo">
      <input type="submit">
    </form>
    <div id="progress-bar-container">
      <div id="progress-bar"></div>
    </div>
  </div>

  <script src="{% static 'js/documento/subi.js' %}"></script>
</body>

</html>
```

Figura 39. Plantilla para subir un documento.

Subir Archivo

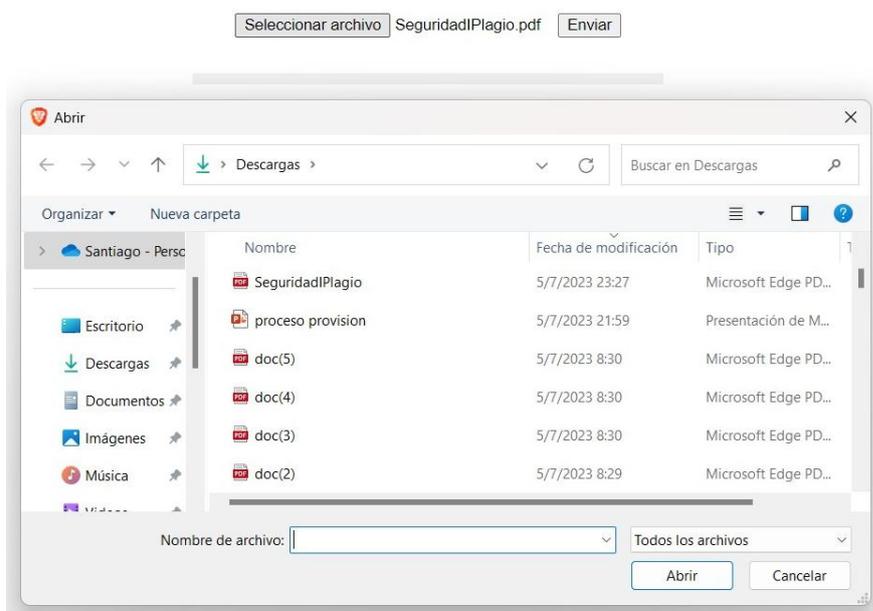


Figura 40. Interfaz subir un documento.

Actividad 1.4: Generar pruebas Funcionales del prototipo de software Antiplagio.

En el **Anexo 9** se presenta un plan de pruebas funcionales del prototipo de software Antiplagio. Así mismo a continuación se presenta en la **Tabla 11** las respectivas pruebas realizadas del software Antiplagio buscando comparar los resultados del software con una herramienta gratuita de internet.

Tabla 11. Casos de Pruebas Funcionales con el Prototipo de Software Antiplagio.

Resultados con el software Antiplagio			
Documento	Tiempo de ejecución	Oraciones plagiadas encontradas	Porcentaje total obtenido
Anteproyecto Jessica Guazha	38 min	29 oraciones	3 %
Anteproyecto Daniel Novillo, Brayan Guaman	37 min	14 oraciones	2 %
Anteproyecto Alexis Armijos	40 min	36 oraciones	4 %
Anteproyecto Carlos Ordoñez	32 min	40 oraciones	5 %
Anteproyecto Kevin Cumbicus, Byron Herrera	26 min	31 oraciones	4 %
Anteproyecto Jean Ochoa y Edgar Andrés Soto	43 min	52 oraciones	2 %
Anteproyecto Angel Martinez	24 min	38 oraciones	5 %
Anteproyecto Beiker Santorum	30 min	60 oraciones	7 %
Anteproyecto Tandazo Ramírez, Junior Alexander	25 min	5 oraciones	1 %
Anteproyecto Vázquez Minchala, Bladimir Alejandro	42 min	12 oraciones	2 %

Los resultados que presento el software Antiplagio usando 10 documentos PIC fueron bastante alentadores teniendo en cuenta que la mayoría de oraciones plagiadas encontradas en la ejecución corresponden a otros documentos PIC mostrando que este tiene mayor preferencia a los plagios de este tipo.

Tabla 12. Casos de Pruebas Funcionales con el Software PlagiarismDetector.net

Resultados con un software gratuito (PlagiarismDetector.net)			
Documento	Tiempo de ejecución	Oraciones plagiadas encontradas	Porcentaje total obtenido
Anteproyecto Jessica Guazha	20 min	35 oraciones	6 %
Anteproyecto Daniel Novillo, Brayan Guaman	19 min	16 oraciones	4 %
Anteproyecto Alexis Armijos	18 min	41 oraciones	5 %
Anteproyecto Carlos Ordoñez	19 min	31 oraciones	3 %
Anteproyecto Kevin Cumbicus, Byron Herrera	14 min	16 oraciones	5 %
Anteproyecto Jean Ochoa y Edgar Andrés Soto	34 min	83 oraciones	5 %
Anteproyecto Angel Martinez	28 min	56 oraciones	6 %
Anteproyecto Beiker Santorum	19 min	21 oraciones	3 %
Anteproyecto Tandazo Ramírez, Junior Alexander	20 min	27 oraciones	5 %
Anteproyecto Vázquez Minchala, Bladimir Alejandro	31 min	80 oraciones	7%

La ejecución en este programa nos muestra resultados idóneos a los que se buscan llegar, este programa demuestra menores tiempos de ejecución y en su gran mayoría una elevada cantidad de plagio identificada, sin embargo, este tipo de software no tiene acceso a los documentos PIC de la Universidad Nacional de Loja y por su defecto solo realiza búsqueda por medio de internet para encontrar las similitudes, esto tiende a reducir la efectividad del programa a la hora de encontrar similitudes que puedan generar un mayor problema de plagio para la institución.

Tabla 13. Casos de Prueba PlagiarismDetector.net y Prototipo de Software Antiplagio

Comparación de resultados entre ambos softwares						
#	Antiplagio		(PlagiarismDetector.net)		% de efectividad	
Documento	Tiempo de ejecución	Porcentaje de similitud obtenido	Tiempo de ejecución	Porcentaje de similitud obtenido	%Efectividad Tiempo de ejecución	%Efectividad de las similitudes obtenidas
Documento 1	38 min	3 %	20 min	6 %	- 90 %	50 %
Documento 2	37 min	2 %	19 min	4 %	- 94.73 %	50 %
Documento 3	40 min	4 %	18 min	5 %	- 122.2 %	80 %
Documento 4	32 min	5 %	19 min	3 %	- 68.42 %	166 %
Documento 5	26 min	4 %	14 min	5 %	- 85.71 %	80 %
Documento 6	43 min	2 %	34 min	5 %	- 26.47 %	40 %

Documento 7	24 min	5 %	28 min	6 %	14,28 %	83 %
Documento 8	30 min	7 %	19 min	3 %	- 57.89 %	233 %
Documento 9	25 min	1 %	20 min	5 %	- 25 %	20 %
Documento 10	42 min	2 %	31 min	7%	- 35.48 %	28 %
Total					- 59,16 %	83 %

El análisis de la eficacia tanto en términos de tiempo de ejecución como en la identificación de similitudes durante la ejecución del software antiplagio se realizó empleando fórmulas específicas. Para calcular el porcentaje de efectividad con respecto al tiempo, se aplicó la fórmula $\left(1 - \frac{\text{Tiempo de ejecución de antiplagio}}{\text{Tiempo de ejecución de PlagarismDetector.net}}\right) * 100$ lo que resultó en un porcentaje promedio de -59.16 % de efectividad. Este resultado indica que el software antiplagio presenta un tiempo de ejecución aproximadamente un 60% más prolongado que otros detectores de plagio convencionales.

Por otro lado, el porcentaje de efectividad en relación con el porcentaje de similitudes detectadas durante la ejecución se determinó mediante la fórmula $\left(\frac{\text{Porcentaje de antiplagio}}{\text{Porcentaje de PlagarismDetector.net}}\right) * 100$ obteniendo así un porcentaje promedio del 83 % de efectividad. Este resultado señala que el software antiplagio logra identificar un 83% de las similitudes que se detectan utilizando otros detectores de plagio convencionales.

Actividad 1.5: Generar pruebas de Aceptación.

En esta actividad se aplicó del modelo TAM. Para ello se presenta los pasos a seguir para Evaluar el software Antiplagio, mediante este modelo.

1. Definir el Sistema web.

Prototipo de software para detección de plagio: "**Antiplagio**"

Descripción: Antiplagio es un prototipo de software para la detección de plagio diseñado para la Universidad Nacional de Loja. Su objetivo principal es identificar el plagio en trabajos TIC.

Características:

- **Análisis de similitud:** Antiplagio utiliza un algoritmo de comparación para buscar similitudes entre el texto del documento analizado y una amplia base de datos de fuentes académicas, literarias y en línea.

- Informes detallados: Antiplagio genera informes que muestra los resultados del análisis de plagio, destacando las secciones sospechosas de plagio y proporcionando enlaces directos a las fuentes originales.
- Interfaz amigable: La interfaz de usuario de Antiplagio es intuitiva y fácil de usar, lo que permite a los usuarios cargar y analizar documentos sin complicaciones.

2. Identificar las variables del modelo TAM: El TAM se basa en dos constructores principales:

a. Percepción de utilidad (PU): Se refiere a que si el sistema de detección de plagio les será útil en su trabajo o estudio. Es decir, si creen que les ayudará a identificar y prevenir el plagio de manera efectiva.

b. Facilidad de uso (FE): Se refiere a la facilidad de aprender y utilizar el sistema de detección de plagio. Cuanto más fácil de usar perciban el sistema, más probable será su adopción.

3. Diseñar un cuestionario y Medir las variables denominadas constructores.

En la **Tabla 11**, se presenta las variables identificadas junto con el cuestionario y los ítems denominados indicadores.

Tabla 14. Tabla de preguntas para el modelo TAM.

Variables	Indicadores	Cuestionario
Facilidad de uso percibida (FUP)	FUP1	¿Cree usted que el proceso de registro o inicio de sesión en Antiplagio es sencillo y sin complicaciones?
	FUP2	¿Crees que es sencillo cargar tus documentos en Antiplagio para realizar el análisis de plagio?
	FUP3	¿Le resulta sencillo navegar por las diferentes secciones dentro del sistema Antiplagio?
	FUP4	¿Encuentras que la interfaz de Antiplagio es fácil de entender y utilizar?
	FUP5	¿Cree usted que el sistema Antiplagio le permite realizar el análisis de plagio con pocos pasos y sin complicaciones innecesarias?
	FUP6	¿Le resulta cómodo interpretar los resultados de plagio generados por el sistema Antiplagio?
Utilidad percibida (UP)	UP1	¿Cree usted que el sistema Antiplagio le será útil para identificar posibles casos de plagio en los trabajos académicos?
	UP2	¿Cree usted que el sistema Antiplagio te proporcionará información útil y detallada sobre las fuentes o citas que podrían estar relacionadas con el plagio?
	UP3	¿Cree usted que el sistema Antiplagio es una herramienta accesible y adecuada para estudiantes y profesores en la Carrera de Computación?

UP4	¿Consideras que el acceso gratuito a Antiplagio es una ventaja significativa en comparación con otras herramientas de detección de plagio de pago?
UP5	¿Cree usted que el sistema Antiplagio ofrece un análisis de resultados claro y comprensible para ayudarte a mejorar la calidad de tus trabajos académicos?
UP6	¿Cree usted que el sistema Antiplagio es una herramienta valiosa para la carrera de computación?

Para el cuestionario utilizaremos una escala de opción única 1 o 2, donde 1 representa " de acuerdo " y 2 representa " en desacuerdo ", así mismo se le agrego un valor para cada opción, lo cual se puede apreciar en la **Tabla 12**.

Tabla 15. Valoraciones numéricas para cada respuesta escogida por el usuario.

Respuesta	Valor
de acuerdo	12,5
en desacuerdo	0

4. Seleccionar una muestra de usuarios y recolectar datos: Se eligió un grupo representativo de usuarios que utilizarán el sistema de detección de plagio, en este caso serán docentes y estudiantes de la carrera de Computación.

Se tiene una muestra de 8 usuarios incluido estudiantes y docentes, que han utilizado el sistema. Se distribuyo el cuestionario diseñado (véase **Anexo 12**) en el paso 3, a los usuarios para recopilar sus percepciones sobre la utilidad y facilidad de uso del sistema.

Tabla 16. Cálculo del promedio final para la Aceptación en base a las variables FU y UP.

Variables	Facilidad de Uso	Utilidad de uso Percibida
P1	87,5% = 0,875	100% = 1,0
P2	75% = 0,75	87,5% = 0,875
P3	75% = 0,75	100% = 1,0
P4	75% = 0,75	87,5% = 0,875
P5	62,5% = 0,625	87,5% = 0,875
P6	62,5% = 0,625	100% = 1,0
Suma Total:	4,375	5,625
Promedio:	0,7292	0,9375

Promedio en porcentaje:	72,92%	93,75%
--------------------------------	--------	--------

Tabla 17. Cálculo del promedio final para el Rechazo en base a las variables FU y UP.

Variables	Facilidad de Uso	Utilidad de uso Percibida
P1	12,5% = 0,125	0% = 0
P2	25% = 0,25	12,5% = 0,125
P3	25% = 0,25	0% = 0
P4	25% = 0,25	12,5% = 0,125
P5	37,5% = 0,375	12,5% = 0,125
P6	37,5% = 0,375	0% = 0
Suma Total:	1,625	0,375
Promedio:	0,2708	0,0625
Promedio en porcentaje:	27,08%	6,25%

Tabla 18. Análisis de resultados en base a cada variable de medición.

Pregunta	1	2	3	4	5	6	7	8	Aceptación	Rechazo
FU	P1	✓	✓	✓	✓	✗	✓	✓	87,5%	12,5%
	P2	✓	✓	✓	✗	✗	✓	✓	75%	25%
	P3	✗	✗	✓	✓	✓	✓	✓	75%	25%
	P4	✓	✓	✓	✗	✗	✓	✓	75%	25%
	P5	✗	✗	✗	✓	✓	✓	✓	62,5%	37,5%
	P6	✓	✓	✗	✗	✗	✓	✓	62,5%	37,5%
	Promedio Final:									72,92%
UP	P1	✓	✓	✓	✓	✓	✓	✓	100%	0%
	P2	✓	✓	✓	✓	✓	✗	✓	87,5%	12,5%
	P3	✓	✓	✓	✓	✓	✓	✓	100%	0%
	P4	✓	✓	✓	✗	✓	✓	✓	87,5%	12,5%
	P5	✗	✓	✓	✓	✓	✓	✓	87,5%	12,5%
	P6	✓	✓	✓	✓	✓	✓	✓	100%	0%
	Promedio Final:									93,75%

5. Analizar los resultados:

Facilidad de Uso (FU).

Los resultados indican que los usuarios consideran que Antiplagio es fácil de usar, ya que la calificación promedio para esta sección es de **72,92%**. Esto muestra que los usuarios encuentran la interfaz del sistema intuitiva y sencilla de navegar. Una alta facilidad de uso es crucial para la adopción exitosa del sistema, ya que reduce las barreras de entrada y hace que los usuarios se sientan más cómodos al utilizarlo.

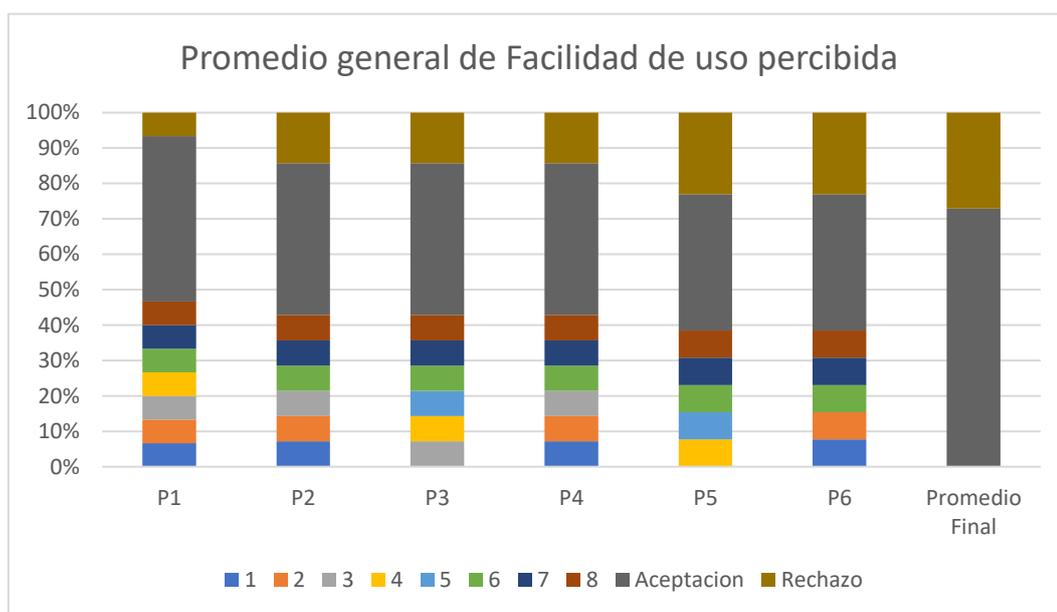


Figura 41. Resultado de Facilidad de uso Percibida

Utilidad de uso Percibida (UP).

Los resultados muestran que los usuarios perciben que Antiplagio es útil para identificar posibles casos de plagio en sus trabajos académicos, ya que la calificación promedio para esta sección es de **93,75%**. Esto sugiere que los usuarios valoran la capacidad del sistema para ayudarles a garantizar la originalidad y la integridad de sus trabajos. Una alta percepción de utilidad es un factor importante para la aceptación del sistema, lo que indica que los usuarios están dispuestos a utilizarlo para mejorar la calidad de sus investigaciones.

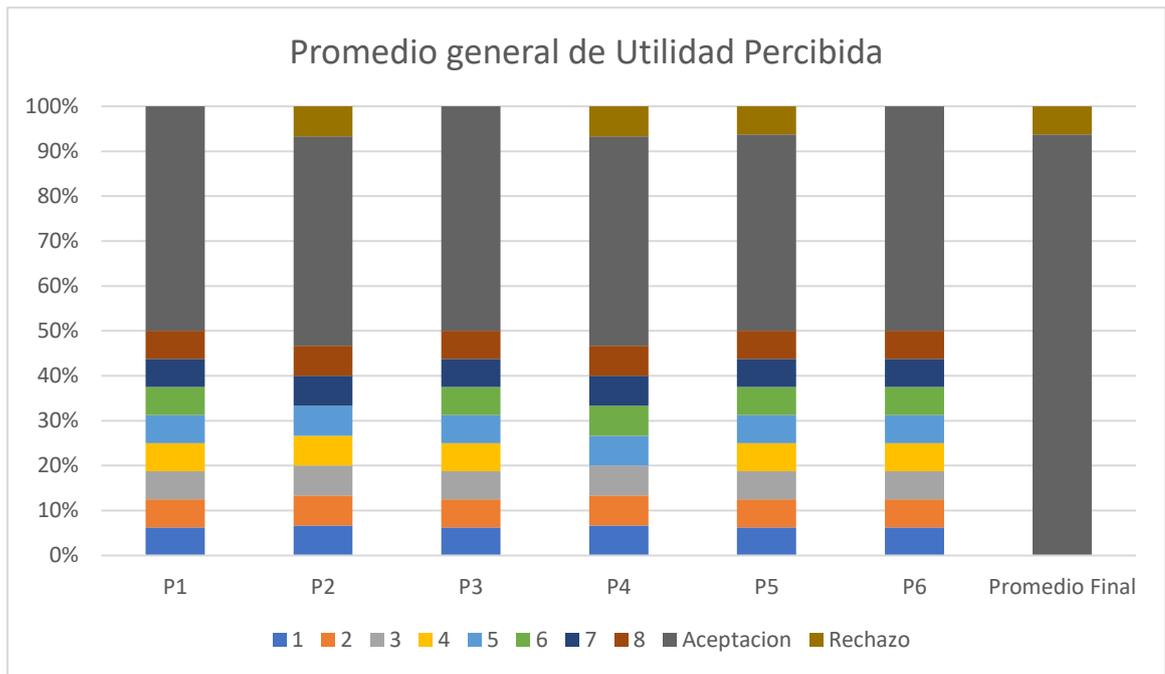


Figura 42. Resultado de Utilidad Percibida.

7. Discusión

Para dar respuesta a la pregunta de investigación: ¿Como se logrará controlar el plagio mediante el desarrollo de un software de control Antiplagio para la carrera de ingeniería en Computación en la Universidad Nacional de Loja, mediante el uso de redes neuronales? El proceso se inició mediante el desarrollo de un modelo detector utilizando el lenguaje Python, El punto de partida para la creación de este modelo fue el Trabajo Relacionado [72], el cual presentaba otra solución al problema de investigación planteado. No obstante, es relevante señalar que este modelo estaba diseñado exclusivamente para el idioma inglés. Para adaptarlo a nuestras necesidades, se procedió a una reestructuración del modelo, al mismo tiempo que garantizamos su correcto funcionamiento a través de pruebas unitarias. Los resultados de estas pruebas, detallados en la sección 4, demostraron de manera satisfactoria la eficacia del modelo adaptado. Con la finalidad de mejorar la precisión del modelo, se incorporó una red neuronal siamesa, apoyándonos en el Trabajo Relacionado [73]. Dicha red fue programada y sometida a un proceso de entrenamiento mediante trabajos PIC obtenidas del repositorio de la UNL. Los resultados obtenidos, expuestos en la sección 5, confirmaron su rendimiento satisfactorio y su capacidad para potenciar la detección de casos de plagio.

El desarrollo del software web Antiplagio, basado en el framework Django, que forma parte del segundo objetivo, representa una mejora con respecto al trabajo relacionado [72] que se tomó como base, ya que este se limita exclusivamente a un modelo y carece del componente web. Al implementar el modelo detector, potenciado por las redes neuronales previamente diseñadas con el componente web, se comprueba la funcionalidad correcta del software desarrollado para el análisis de plagio, lo cual se puede constatar en el plan de pruebas funcionales elaborado en el **Anexo 8**. Además, la accesibilidad del software a través de una interfaz web agregó un valor significativo al Trabajo debido a su utilidad y facilidad de uso. En resumen, la incorporación del componente web y el funcionamiento del modelo detector respaldado por las redes neuronales realzan la contribución y la utilidad del software desarrollado.

Con la finalidad de evaluar el software se utilizó en modelo de aceptación tecnológica TAM, centrándose en la Facilidad de Uso percibida (FU) y Utilidad de Uso percibida (UP) del sistema Antiplagio, con los usuarios encuestados tanto estudiantes como docentes de la carrera de Computación de la Universidad Nacional de Loja. En la sección de Resultados del objetivo 3, en relación a la Facilidad de Uso percibida (FU), los resultados muestran que una gran proporción de usuarios considera que es fácil utilizar el sistema de control atiplagio, con una aceptación de 72,92% totalmente de acuerdo, del promedio general de todas las preguntas. Sin embargo, es importante mencionar que se identificó un grupo minúsculo de

usuarios, los cuales estuvieron en desacuerdo, con la facilidad del uso del sistema, de entre los más críticos se identificó que el sistema no les resulta cómodo para interpretar los resultados de plagio ((FU5) con un rechazo de 37,5%) y que el sistema no les permite realizar el análisis en pocos pasos ((FU6) con un rechazo de 37,5%), obteniendo, así como promedio general de todas las interrogantes un rechazo de 27,08% en desacuerdo.

En cuanto a la Utilidad de uso Percibida (UP), los resultados arrojan que una gran cantidad de usuarios considera que la herramienta es útil en la detección de plagio (PU1), valiosa para la carrera de computación (PU6), accesible y adecuada para estudiantes y profesores (PU3), obteniendo una aceptación del 93,75% del promedio general de todas las preguntas realizadas. Sin embargo se identificó un grupo minúsculo de usuarios, los cuales estuvieron en desacuerdo, con la utilidad del sistema, de entre los más críticas se identificó que el sistema no les proporciona información útil ((UP2) con un rechazo de 12,5%), no es una ventaja el acceso gratuito, en comparación con otras herramientas de pago ((UP4) con un rechazo de 12,5%), y que el sistema no ofrece un análisis de resultados claro y comprensible para ayudar a mejorar la calidad de los trabajos ((UP5) con un rechazo de 12,5%). obteniendo, así como promedio general de todas las interrogantes un rechazo de 6,25% en desacuerdo.

Otras soluciones, como se puede observar en los Trabajos relacionados [71],[72]. [73],[74] si bien se realiza la implementación de un prototipo de software para detectar plagio, no se evidencia la presencia de una interfaz gráfica que permita la visualización de los resultados. Esta carencia dificulta la interpretación y comprensión de dichos resultados. Además, cabe destacar que estas soluciones carecen de una integración con redes neuronales. Por el contrario, el presente trabajo de titulación, mediante la implementación del componente web, se pudo corroborar su utilidad y facilidad de uso. Este aspecto fue verificado a través de la prueba de aceptación mencionada previamente. Asimismo, las pruebas unitarias y funcionales realizadas confirmaron el correcto desempeño del sistema implementado.

Es importante mencionar que la solución [72], se limita exclusivamente al idioma inglés, careciendo además de un componente web que facilite al usuario la comprensión de las funcionalidades que el prototipo pretende cubrir. Además, no se incluye un plan de pruebas unitarias que permita verificar el adecuado funcionamiento de cada componente programado. De manera similar, la ausencia de pruebas funcionales impide validar todo el funcionamiento del prototipo, Por último, la carencia de pruebas de aceptación implica que no se realiza una validación por parte de los usuarios para comprobar si el software desarrollado cumple con sus expectativas.

Por lo tanto, la solución planteada es mejor respecto a los trabajos relacionados, ya que garantiza que se pueda hacer un análisis de plagio satisfactorio, disponer de resultados entendibles y que la herramienta es de gran utilidad en un 93,75%. Además, aborda deficiencias presentes en los trabajos relacionados al incluir en el presente PIC un prototipo de software Antiplagio con una interfaz gráfica amigable, que cuenta con pruebas de aceptación para confirmar su pertinencia en cuanto a su facilidad de uso y utilidad, cubriendo así las carencias previamente mencionadas.

8. Conclusiones

- Se logro desarrollar un modelo de control Antiplagio utilizando redes neuronales implementadas con Scikit-Learn y técnicas de Procesamiento de Lenguaje Natural, como la Tokenizacion, Lemantizacion y Normalizacion, junto con el cálculo de similitud mediante el uso del método del coseno, lo que permitió obtener una herramienta útil para detectar plagio en trabajos y documentos académicos. Durante el proceso de desarrollo, se entrenó el modelo utilizando un Dataset generado con documentos PIC de la carrera de computación de la UNL. Posteriormente, se evaluó su rendimiento con un conjunto de pruebas independientes, obteniendo un resultado permisible. Cabe destacar que este modelo es básico y tiene sus limitaciones, ya que puede encontrar desafíos en situaciones donde se utilice técnicas de reescritura avanzada.
- La aplicación del estandar IEEE830 para la Especificación de Requisitos de software resultó fundamental para identificar tanto los requisitos funcionales como los no funcionales. Al complementar con las historias de usuario definidas, se pudo obtener una definición detallada de las funcionalidades necesarias en el desarrollo del software Antiplagio.
- La utilización de la metodología XP (Programacion Extrema), junto con el software SLDC propio de la carrera de Computación de la Universidad Nacional de Loja (UNL) durante el proceso de desarrollo del software Antiplagio, desempeñó un papel fundamental en la planificación y organización efectiva del Trabajo, permitiendo así complementar el desarrollo de todos los requisitos identificados. Durante la fase de diseño de la metodología XP se aplicó el modelo 4+1 para satisfacer los requerimientos del software.
- La evaluación de la ejecución del software detector de plagio mediante la prueba de aceptación basada en el modelo TAM ha arrojado resultados positivos. Estos resultados se derivan de las encuestas que se centraron en la variable de Facilidad de Uso (FU) y revelaron un porcentaje de aceptación del 72.92%. Además, en relación a la Utilidad de uso Percibida (UP), se obtuvo un porcentaje de aceptación del 93.75%. Estos datos reflejan la utilidad y la facilidad de uso del software para los usuarios, respaldando la viabilidad de implementar el software como una herramienta para afrontar los desafíos de la detección de plagio en los Proyectos de Integración Curricular.

9. Recomendaciones

- Verificar constantemente el correcto funcionamiento del modelo detector para así asegurarnos de su eficacia y precisión en la detección. Esto implica realizar pruebas en diversos escenarios y condiciones, identificando posibles fallos y refinando el modelo en caso necesario.
- Emplear un enfoque ágil como Extreme Programming (XP) en Trabajos de desarrollo de software, incorporando la participación activa y la retroalimentación de los clientes. Esta estrategia asegura la flexibilidad del proceso de desarrollo, inmune a las influencias de cambios inesperados. Esto, a su vez, garantiza entregas puntuales y en línea con las necesidades expresadas por los usuarios.
- Ampliación de Fuentes de Información: Dado que el proceso de búsqueda inicial de información relacionada con el desarrollo de redes neuronales para procesamiento del lenguaje resultó desafiante debido a la escasez de ejemplos en el idioma español, se recomienda ampliar la búsqueda en fuentes académicas, conferencias y revistas especializadas incluyendo búsquedas en otros tipos de idiomas. Se sugiere fomentar la colaboración y el intercambio de conocimientos por medio de plataformas en línea, foros y comunidades relacionadas principalmente con el procesamiento del lenguaje natural y las redes neuronales podría brindar soluciones innovadoras y superar desafíos técnicos.
- Exploración de Alternativas Tecnológicas: Dado que el desarrollo del prototipo de software se centró en el uso de redes neuronales siamesas, se podría investigar y considerar la integración de otras arquitecturas de redes neuronales y técnicas avanzadas de procesamiento del lenguaje. Esta exploración podría enriquecer la funcionalidad del prototipo y aumentar su precisión en la detección de similitudes. Además de llevar a cabo una evaluación continua del prototipo de software en entornos reales y con un grupo más amplio de usuarios permitirá identificar posibles mejoras, optimizar la usabilidad y abordar cualquier problema técnico o de experiencia de usuario que pueda surgir.
- Perspectiva Multidisciplinaria: Aunque el enfoque se centró en la Ingeniería en Computación, se sugiere explorar colaboraciones interdisciplinarias con expertos en áreas como lingüística, ética en la tecnología y pedagogía. Esta perspectiva multidisciplinaria podría enriquecer la comprensión de los problemas de plagio, sus implicaciones y contribuir a la mejora continua del prototipo.

Trabajos futuros

- Exploración de Redes Neuronales Alternativas: Para mejorar la precisión y eficiencia del prototipo de software, se recomienda explorar una variedad de arquitecturas de redes neuronales más allá de las redes neuronales siamesas. Investigar modelos como LSTM, GRU o transformers podría ofrecer nuevas perspectivas para el procesamiento del lenguaje natural y la detección de similitudes en textos.
- Implementación de Algoritmos de Refinamiento: Como mejora adicional, se podría considerar la implementación de algoritmos de refinamiento que ayuden a detectar plagio más sutil y sofisticado. Esto podría implicar el uso de técnicas de procesamiento de lenguaje natural más avanzadas y la combinación de diferentes modelos de redes neuronales para un análisis más completo.

Limitaciones

- Una de las principales limitaciones radicó en la falta de métodos que emplearan redes neuronales para la detección de plagio, especialmente en el contexto del idioma español. Dado que la mayoría de los modelos y enfoques de detección de similitudes textuales se centran en el idioma inglés, se enfrentó un desafío al adaptar estos métodos al español.
- Otra limitante crucial en este trabajo estuvo relacionada con los recursos computacionales disponibles. La insuficiencia de estos recursos generó una demora significativa en la realización de las pruebas iniciales de la red neuronal. La complejidad de los análisis entre textos extensos demandó un extenso período de tiempo, lo cual impactó negativamente en la eficiencia y rapidez de los procesos.

10. Bibliografía

- [1] D. D. Arce, "Herramientas 'Antiplagio': ¿son confiables? Estudio de casos.," *EduTec. Revista Electrónica de Tecnología Educativa*, no. 61, pp. a365–a365, 2017.
- [2] "¿Qué es una red neuronal? - Documentación de IBM." <https://www.ibm.com/docs/es/spss-statistics/29.0.0?toTIC=networks-what-is-neural-network> (accessed Jun. 18, 2023).
- [3] Á. Guerrero Sánchez, "Desarrollo de una aplicación web para un gimnasio en Django," Universitat Politècnica de València, 2022.
- [4] J. A. O. HERNÁNDEZ, "DESARROLLO DE TRABAJOS DE SOFTWARE," *Tecnológico de Estudios Superiores, Ciudad de México*, 2011.
- [5] J. Fernando Sánchez-Vega, M. Montes Y Gómez, and P. Rosso, "Identificación de plagio parafraseado incorporando estructura, sentido y estilo de los textos," Feb. 2016.
- [6] Y. B. Roka, "Plagiarism: Types, causes and how to avoid this worldwide problem," *Nepal Journal of Neuroscience*, vol. 14, no. 3, pp. 2–6, 2017.
- [7] R. Ramírez Bacca, H. David, and J. Patiño, "Plagio y 'auto-plagio'. Una reflexión Plagiarism and Self-plagiarism. A Reflexion," Jul. 2016.
- [8] U. Y. Sociedad, E. Enrique, and E. Freire, "EL PLAGIO UN FLAGELO EN EL ÁMBITO ACADÉMICO ECUATORIANO," May 2020.
- [9] Sandra Timal López and Francisco Sánchez Espinoza, "El plagio en el contexto del derecho de autor," Universidad Autónoma de Puebla, México, 2016.
- [10] A. Soto Rodríguez, "E-Ciencias de la Información El plagio y su impacto a nivel académico y profesional," Jan. 2012.
- [11] X. B. Olabe, "REDES NEURONALES ARTIFICIALES Y SUS APLICACIONES".
- [12] P. Larrañaga, "Tema 8. Redes Neuronales".
- [13] I. N. Da Silva *et al.*, *Artificial neural network architectures and training processes*. Springer, 2017.
- [14] Marcos, "Introducción a las Redes de Neuronas Artificiales", Accessed: Jun. 18, 2023. [Online]. Available: <http://sabia.tic.udc.es/>
- [15] "Conceptos básicos (redes neuronales) - Documentación de IBM." <https://www.ibm.com/docs/es/spss-modeler/saas?toTIC=networks-basics-neural> (accessed Jun. 18, 2023).
- [16] Antonio J. Serrano, Emilio Soria, and Jose D. Maritn, *REDES NEURONALES ARTIFICIALES*.
- [17] C. Alberto Ruiz Marta Susana Basualdo Autor and D. Jorge Matich, "Cátedra: Informática Aplicada a la Ingeniería de Procesos-Orientación I Redes Neuronales: Conceptos Básicos y Aplicaciones".

- [18] M. M. Mijwel, "Artificial neural networks advantages and disadvantages," Retrieved from LinkedIn [https://www.linkedin.com/pulse/artificial-neuralnet Work](https://www.linkedin.com/pulse/artificial-neuralnet-work), 2018.
- [19] D. J. Matich, "Neural Networks: Basic Concepts and Applications," *National University of Technology—Faculty Regional Rosario. Group of Applied Research in the Chemical Engineering (GIAIQ)*, 2001.
- [20] K. Kumar and G. S. M. Thakur, "Advanced applications of neural networks and artificial intelligence: A review," *International journal of information technology and computer science*, vol. 4, no. 6, p. 57, 2012.
- [21] M. Campos Mocholl'i, "Clasificación de textos basada en redes neuronales," Universitat Politècnica de València, 2021.
- [22] J. Avendaño, "Redes Neuronales Aplicadas al Análisis de Datos," *Revista de Información Tecnológica y Sociedad*, vol. 391, 2009.
- [23] F. Casacuberta Nolla and Á. Peris Abril, "Traducción automática neuronal," *Tradumática*, no. 15, pp. 66–74, 2017.
- [24] M. Gregoryev, "Generación de textos en ruso mediante técnicas de Aprendizaje Automático para la industria del lenguaje," Universitat Politècnica de València, 2022.
- [25] D. H. Milone, "Reconocimiento automático del habla con redes neuronales artificiales," *Ciencia, Docencia y Tecnología*, vol. 16, no. 31, pp. 261–322, 2005.
- [26] I. Bonet Cruz, S. Salazar Martinez, A. R. Abed, G. Abalo, and M. M. G. Lorenzo, "Redes neuronales recurrentes para el análisis de secuencias Recurrent neural network for sequences analysis".
- [27] J. Pérez Guerrero Sevilla, J. de, T. por, and R. Pino Mejías, "REDES RECURRENTEES."
- [28] "Siamese Network : Architecture and Applications in Computer Vision Hengliang Luo," 2014. [Online]. Available: <http://en.wikipedia.org/wiki/Siamese>
- [29] L. Mera-Jiménez and J. F. Ochoa-Gómez, "Redes neuronales convolucionales para la clasificación de componentes independientes de rs-fMRI," *Tecnológicas*, vol. 24, no. 50, p. e1626, Jan. 2021, doi: 10.22430/22565337.1626.
- [30] A. : Javier, S. Gozalo, and F. Díaz Gómez, "Análisis del estado del arte de la generación de texto con redes neuronales mediante modelos de Transformer."
- [31] G. García Subies Tutor and F. Serradilla García Madrid, "Trabajo Fin de Máster Modelos de Transformers para la Clasificación de Texto."
- [32] L. Laura, R. Calcagni, and F. Ronchetti, "Redes Generativas Antagónicas y sus aplicaciones."
- [33] J. de la Torre, "REDES GENERATIVAS ADVERSARIAS (GAN) FUNDAMENTOS TEÓRICOS Y APLICACIONES SURVEY," 2023.

- [34] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese Neural Networks for One-shot Image Recognition."
- [35] "Contenido Lista de figuras IV."
- [36] D. Chicco, "Siamese neural networks: An overview," *Artificial neural networks*, pp. 73–94, 2021.
- [37] L. V Utkin, M. S. Kovalev, and E. M. Kasimov, "An explanation method for Siamese neural networks."
- [38] "Contenido Lista de figuras IV."
- [39] Mario Campos Mocholi, "Clasificación de textos basadas en redes neuronales".
- [40] M. Silvestre Gómez, "Implementación de asignación jerárquica latente de Dirichlet para modelado de temas," 2018.
- [41] A. C. Vásquez, J. P. Quispe, A. M. Huayna, and others, "Procesamiento de lenguaje natural," *Revista de investigación de Sistemas e Informática*, vol. 6, no. 2, pp. 45–54, 2009.
- [42] V. Escudero González, "Plataforma para procesado de lenguaje natural como servicio," 2022.
- [43] J. G. Valladares Cedillo, "Análisis de sentimientos para textos cortos en español, una revisión del estado del arte," 2022.
- [44] Cisco Networking Academy, "Cisco Skills For All," Apr. 23, 2023. <https://skillsforall.com/es/launch?id=728abfb2-e134-4203-9365-b10e675e9971&tab=curriculum&view=62449c7c-a047-5c0f-8a08-640ed53efaac> (accessed May 30, 2023).
- [45] K. R. Srinath, "Python—the fastest growing programming language," *International Research Journal of Engineering and Technology*, vol. 4, no. 12, pp. 354–357, 2017.
- [46] B. Tutorial, "tutorialspoint. com," *Tutorial Point*, 2017.
- [47] R. G. Duque, "Python PARA TODOS", Accessed: Jun. 18, 2023. [Online]. Available: <http://mundogeek.net/tutorial-python/>
- [48] "Ingeniería del Software StarUML", Accessed: Jul. 24, 2023. [Online]. Available: <http://cnx.org/content/m15092/latest/>
- [49] K. Beck, "Praise for Extreme Programming Explained, Second Edition."
- [50] Kent. Beck and M. Fowler, *Planning extreme programming*. Addison-Wesley, 2001.
- [51] Roger s. Pressman, *Ingenieria del software-Presman 7ma edicion 2010*.
- [52] N. Matallana Torres and J. Antonio Torres Benavides PRESIDENTE Maria Ysabel Aranguri Garcia Manuel Gregorio Leon Tenorio, "Aplicación web utilizando la metodología de diagnóstico logístico para apoyar el proceso de gestión de pedidos en una universidad privada de la región Lambayeque PRESENTADA POR INGENIERO DE SISTEMAS Y COMPUTACIÓN APROBADA POR."

- [53] D. P. Ramírez, "METODOLOGÍAS ÁGILES. ¿ CÌMO DESARROLLO UTILIZANDO XP?," 2008.
- [54] T. Dudziak, "Extreme programming an overview," *Methoden und Werkzeuge der Softwareproduktion WS*, vol. 1999, pp. 1–28, 2000.
- [55] L. Crispin and T. House, *Testing extreme programming*. Addison-Wesley Professional, 2003.
- [56] O. A. Pérez, "Cuatro enfoques metodológicos para el desarrollo de Software RUP–MSF–XP–SCRUM," *Inventum*, vol. 6, no. 10, pp. 64–78, 2011.
- [57] Luis Miguel Echeverry and Luz Elena Delgado, "CASO PRÁCTICO DE LA METODOLOGÍA ÁGIL XP AL DESARROLLO DE SOFTWARE," Universidad Tecnológica de Pereira , 2007.
- [58] Y. Blanco, I. E. Blanco, C. Arletys, and I. L. Robles, "Metodologías de desarrollo de software (XP)."
- [59] D. A. Leyton Soto, "Extensión al modelo de aceptación de tecnología TAM, para ser aplicado a sistemas colaborativos, en el contexto de pequeñas y medianas empresas," 2013.
- [60] J. HIDALGO, M. VASQUEZ, L. BRAVO, F. BURGOS, and Y. VARGAS, "Modelo de aceptación de tecnología TAM en NextCloud. Caso de estudio Escuela Computación e Informática," *Revista Espacios*, vol. 40, no. 21, 2019.
- [61] A. L. Rivera Landa, "Importancia de la Documentación de la Arquitectura de Software para el Desarrollo de Sistemas: Caso Practico Sistema de Conciliación de Ineval.," 2016.
- [62] C. Larman, *Applying UML and patterns: an introduction to object oriented analysis and design and interative development*. Pearson Education India, 2012.
- [63] L. B. P. C. R. Kazman, "Software Architecture in Practice Third Edition." 2013.
- [64] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in python Machine Learning for neuroimaging View project Scikit-learn: Machine Learning in Python Gaël Varoquaux Bertrand Thirion Vincent Dubourg Alexandre Passos PEDREGOSA, VAROQUAUX, GRAMFORT ET AL. Matthieu Perrot," 2011. [Online]. Available: <http://scikit-learn.sourceforge.net>.
- [65] Gavin. Hackeling, *Mastering Machine Learning with scikit-learn*. Packt Publishing, 2014.
- [66] Carlos Antona Cortes, "HERRAMIENTAS MODERNAS EN REDES NEURONALES: LA LIBRERIA KERAS," 2017.
- [67] M. Scarpino, *TensorFlow For Dummies*. John Wiley & Sons, 2018.
- [68] A. A. Estévez Acosta and others, "Desarrollo e implementación de un sistema de identificación de signos de la lengua de señas mexicana basado en redes neuronales y el sistema embebido Jetson Nano," *REPOSITORIO NACIONAL CONACYT*, 2022.
- [69] B. Tutorial, "tutorialspoint. com," *Tutorial Point*, 2017.

- [70] D. Altinok, *Mastering spaCy: An end-to-end practical guide to implementing NLP applications using the Python ecosystem*. Packt Publishing Ltd, 2021.
- [71] B. Kahloula and J. Berri, "Plagiarism Detection in Arabic Documents: Approaches, Architecture and Systems.," *Journal of Digital Information Management*, vol. 14, no. 2, 2016.
- [72] O. A. Rivero Ortega, "Creación de un prototipo de sistema de detección de plagio con NLTK," 2020.
- [73] R. Fauzi, M. Iqbal, and T. Haryanti, "Design and Implementation of a Final Project Plagiarism Detection System Using Cosine Similarity Method," *Int. J. Appl. Inf. Technol. Vol*, vol. 5, no. 02, 2021.
- [74] G. Villarrubia González *et al.*, "Diseño e implementación de una herramienta en Studium para evitar los plagios en asignaturas de Ingeniería," 2018.

11. Anexos

Anexo 1. Descripción de las funciones principales para el análisis en documentos.

a. Procesamiento de archivos

El algoritmo comienza obteniendo todos los archivos de entrenamiento y el archivo de test y procesándolos para su posterior manejo. Se levantan todos los archivos de su correspondiente directorio y, los tipos de archivos soportados se parsean (análisis de sintaxis) a texto y se guardan en una clase ArchivoTxt, que funciona como una Struct de C, es decir, solamente es para almacenar cómodamente los datos para ser utilizados, en este caso un Struct de Nombre (nombre del archivo), Extensión, y Texto. Los tipos de archivos no soportados se descartan. Luego de levantarlos se limpian todos los archivos y se obtiene de cada uno de ellos un array de oraciones, que será mucho mejor para que los procesen los siguientes componentes. Dividir el texto en oraciones se hace con la función `sent_tokenize` de `nltk` al que se le puede especificar el lenguaje en el que está el texto para que se tengan en cuenta además de puntos, signos de pregunta, exclamación y demás a la hora de dividir.

La limpieza de los archivos está separada en hilos (uno por archivo) para que se realice más rápido. Esta limpieza esta realizada con expresiones regulares e implica:

- Convertir múltiples enteros (`\n+`) en un entero solo.
- Convertir un entero (`\n`) en un punto (para que `sent_tokenize` tome a los enteros como puntos).
- Convertir múltiples espacios a un espacio solo.
- Letras con tilde a su correspondiente letra sin tilde.
- Normalizar todo a unicode para que no queden caracteres extraños que puedan haber aparecido al momento del parseo, como emoticones, o imágenes. Esto se hizo con la librería `unicodedata2`.

Una vez que realizamos la limpieza de los archivos, obtenemos del archivo “Texto excluido de plagio.txt” que tiene como dijimos antes, texto que no nos gustaría que se analizara el plagio.

b. Tópicos del Texto

Para este componente esperamos a que terminen todos los hilos de procesamiento de archivos para poder arrancarlo, ya que los necesitaremos. Para obtener el tema o tópicos del texto se utilizó `Topic Modelling`, que se basa en obtener un modelo estadístico de los tópicos de un texto, también llamadas palabras clave, temas principales, etc. Se eligió utilizar el modelo LDA para realizar el `Topic Modelling` sobre el texto a analizar. El modelo de

Asignación Latente de Dirichlet (LDA) es un modelo de obtención de tópicos el cual se entrena con ciertos archivos. El modelo ve a cada archivo como una serie de categorías o tópicos distribuidos en base a la distribución de Dirichlet (familia de distribuciones de probabilidad continuas multivariada según Wikipedia). Este modelo está relacionado con Bag of Words, ya que no le importan el orden en el que estén las palabras sino la cantidad de las mismas en el texto.

c. Obtención de los tópicos del texto.

Básicamente lo primero que hacemos es preparar cada archivo de referencia para entrenar el modelo LDA. A cada archivo se prepara en un hilo aparte para tener paralelismo de tareas. "Prepararlo para LDA" consiste en quedarse solamente con los sustantivos del texto y lematizados, es decir, con su palabra raíz (si la palabra es jirafas, lematizada sería jirafa). Esto se lo realiza con spacy, con el modelo de es_core_new_sm. Luego con la librería Gensim, se crea un diccionario que contiene la cantidad de veces que aparece una palabra en los archivos de entrenamiento y con Gensim doc2bow se crea una lista que contiene la cantidad de palabras de cada archivo y el número de veces aparecen. Ahora ya se tiene listo todo lo que se necesita para entrenar el modelo LDA, por lo que se lo entrena. Luego de entrenarlo se le envía un archivo que desconoce el modelo (el archivo del cual queremos saber sus tópicos) y se aplica al modelo entrenado. Esto me va a retornar varias listas de posibles tópicos junto con un puntaje asociado, por lo que nos quedamos con la lista de tópicos que más puntaje tenga, y con los primeros n tópicos, siendo n la cantidad de tópicos que definimos en el archivo de configuración.

d. Comparación con otros TT

Tanto para este componente como el de Plagio de Internet, para la parte de comparar una oración con otra para saber si es plagio o no, se utilizó el método de Similitud Del Coseno, en el cual se obtienen resultados suficientemente favorables para lo que se requiere lograr. La similitud del coseno es una medida de la similitud existente entre dos vectores en un espacio que posee un producto interior con el que se evalúa el valor del coseno del ángulo comprendido entre ellos. Para obtener el plagio de otros trabajos prácticos, lo que hacemos es dado nuestro archivo del cual queremos detectar plagios ya previamente separado en oraciones, preparamos cada oración para ser comparada y la comparamos con cada oración (preparada para comparar) de cada archivo de entrenamiento (los otros Trabajos).

e. Obtener Plagio de Internet

Lo que realiza este componente es bastante similar al anterior componente, salvo que no depende de que el componente del procesamiento de los archivos de entrenamiento haya terminado ya que no los necesita, por lo que comenzará a ejecutarse una vez que el archivo de test esté procesado. Básicamente lo que hacemos es dado nuestro archivo del cual queremos detectar plagios ya previamente separado en oraciones, preparamos cada oración para ser comparada y la buscamos en Google con la API de googlesearch. Entonces la oración ya está preparada para ser comparada como se mencionó en el componente anterior. Luego, de cada oración, nos quedamos con los primeros n links que resultaron de buscarla en Google, siendo n el número que definimos en el archivo de configuración para el campo cantidad_de_links. Guardamos el html de cada una de esas páginas y las convertimos a texto con la herramienta de BeautifulSoup. De cada oración se va obteniendo el porcentaje de similitud según el algoritmo mencionado anteriormente y se va guardando en una lista como (oracion, oracion_mas_parecida, mayor_porcentaje, url_donde_se_encontro, ubicacion_dentro_de la página).

Anexo 2. Descripción de las funciones principales de la implementación de la red neuronal siamesa.

- **Función generar_pares_textos:** Esta función recibe una ruta **pares_textos_path**, que se guardará en un archivo de texto con los pares de textos y sus similitudes. Además, se le envía una lista de documentos que incluye los textos de los documentos que se deben comparar. La función vectoriza los textos de los documentos mediante la técnica TF-IDF. Luego, determina la similitud del coseno de cada par de textos y almacena los resultados en el conjunto de resultados. Por último, guarda los pares de textos y sus equivalentes en el archivo pares_textos.txt.

La elección de un vectorizador TF-IDF con una **longitud máxima de 10.000** y un rango de **n-gramas de 1 a 2** en la función de vectorización lambda se justifica por el equilibrio entre precisión y generalización que proporciona. Al establecer una longitud máxima más pequeña, se evita la captura de detalles precisos, pero poco relevantes que podrían conducir al sobreajuste del modelo. Esto garantiza que el modelo se ajuste de manera más adecuada a los datos de entrenamiento y generalice mejor a nuevos datos. Además, esta elección también tiene en cuenta consideraciones de eficiencia computacional al limitar la dimensionalidad de los vectores y captura información contextual relevante mediante la inclusión de bigramas. En resumen, la elección de estos parámetros busca obtener un equilibrio óptimo entre precisión, generalización y eficiencia computacional en el proceso de vectorización del texto.

- **La función leer_pares_textos:** Lee pares de textos y sus similitudes desde el archivo pares_textos.txt, que se encuentra en la ruta pares_textos_path. Se debe ir sobre cada línea del archivo, realizar el procesamiento necesario para extraer los textos y la similitud, y luego agregarlos a las listas train_text1, train_text2 y train_similarity, respectivamente. Luego, vuelve a enviar estas listas.
- **La función generar_modelo_entrenado:** Responsable de generar y entrenar el modelo utilizando los pares de textos que están disponibles. Para generar los pares de textos y calcular sus similitudes, primero llama a la función generar_pares_textos. Para obtener listas de textos y similitudes, utilice la función leer_pares_textos. Luego, utiliza la clase Tokenizer de Keras para procesar los textos y convertirlos en secuencias numéricas. A continuación, utilice la función pad_sequences para rellenar las secuencias para que tengan la misma longitud. Definió el modelo de una red neuronal siamesa con capas de inserción,

concatenación, salida y densidad. El optimizador Adam y la función de pérdida de crossentropía binaria se utilizan para compilar el modelo. Por último, entrena el modelo con los datos de entrenamiento `train_data1` y `train_data2` y las etiquetas de similitud `train_similarity`.

- **La función `calcular_similitud`:** Utiliza el modelo entrenado para encontrar qué tan similares son dos textos. Tiene dos textos recibidos: `texto1` y `texto2`. Primero, encuentra el modelo entrenado, el tokenizer y los datos de entrenamiento en la lista `modelo_entrenado`. Luego, procesa los datos de entrada `text1_data` y `text2_data` utilizando el tokenizer y el padding. Por último, el modelo se utiliza para predecir la similitud entre los textos y devolver el valor de similitud.

Explicación de los valores específicos utilizados la función `generar modelo entrenado`.

- **`max_words`:** El valor máximo de palabras a tener en cuenta en el vocabulario es de 10.000. La reducción del número de palabras reduce la dimensionalidad de los datos y aumenta la eficiencia computacional. Además, se selecciona un valor lo suficientemente grande para incluir una amplia gama de palabras pertinentes en el texto.
- **`embedding_dim`:** El parámetro `embedding_dim` se refiere a la dimensión del espacio de incrustación utilizado para representar las palabras. Un valor de 100 indica que cada palabra se representará en un espacio de 100 dimensiones. Esta dimensión suele ser suficiente para capturar características semánticas importantes en el texto. Entonces que sucedería si se aumenta o disminuye este valor:
 - Si se aumenta el valor de `embedding_dim`, un valor demasiado grande puede aumentar la complejidad computacional y requerir más memoria para almacenar los vectores de incrustación.
 - Si se reduce el valor de `embedding_dim`. Esto podría resultar en una pérdida de información y una disminución en la capacidad de capturar elementos semánticos complejos en el texto.

Es importante encontrar un equilibrio entre la capacidad de representación del modelo y los recursos computacionales disponibles para lograr un rendimiento óptimo.

- **`hidden_units`:** Representa el número de unidades en la capa oculta de la red neuronal. Se selecciona un valor de 128 para equilibrar la capacidad de aprendizaje

del modelo y la complejidad computacional. Puede modificarse según el tamaño y las necesidades del conjunto de datos. Entonces que sucedería si se aumenta o disminuye este valor:

- Incrementar el número de unidades en la capa oculta puede causar sobreajuste, perdiendo la capacidad de generalizar en nuevos datos. Esto se debe a la mayor complejidad y el riesgo de ajuste excesivo, lo que afecta negativamente el rendimiento del modelo. Es esencial encontrar un equilibrio adecuado en la elección del número de unidades para evitar estos problemas y permitir un mejor rendimiento en los datos.
 - Reducir el valor puede disminuir la capacidad de aprendizaje y la capacidad de capturar relaciones más complejas en los datos. La elección adecuada de `hidden_units` depende de la complejidad del problema y de la cantidad de datos disponibles.
- **dropout_rate:** Muestra la cantidad de unidades que se desactivan aleatoriamente durante el entrenamiento para evitar el sobreajuste. El valor de 0,2 indica que en cada paso de entrenamiento se desactiva el 20% de las unidades. Esto ayuda a regularizar el modelo y mejorar su generalización. Entonces que sucedería si se aumenta o disminuye este valor:
 - Si se aumenta puede llevar a un sub-ajuste, donde el modelo no aprende lo suficiente y no se ajusta adecuadamente a los datos de entrenamiento.
 - Si se disminuye se desactivan menos neuronas durante el entrenamiento, lo que aumenta la dependencia de neuronas individuales y aumenta la probabilidad de que el modelo se sobreajuste.
 - **optimizer:** El optimizador Adam se utiliza para ajustar los pesos del modelo durante el entrenamiento. La tasa de ajuste de los pesos está controlada por un rendimiento (`lr`) de 0.001. Entonces que sucedería si se aumenta o disminuye este valor:
 - Si aumenta el proceso de entrenamiento puede volverse inestable y los resultados del modelo pueden ser poco confiables o deficientes.
 - Si disminuye puede hacer que el proceso de entrenamiento sea más lento. El modelo necesitará más iteraciones y tiempo para ajustar sus parámetros y aprender de los datos de entrenamiento. Esto no sería conveniente cuando se esté trabajando con datos muy grandes o cuando se tenga limitaciones de tiempo.
 - **epochs:** El número de veces que el modelo recorrerá todo el conjunto de entrenamiento durante el proceso de entrenamiento, el valor de 20 se selecciona

como un número suficiente de iteraciones para que el modelo aprenda y ajuste sus parámetros, pero sin gastar demasiado tiempo en el proceso.

- **batch_size:** Representa el número de muestras de entrenamiento que se procesan antes de que se actualicen los parámetros del modelo. En este caso, se utiliza un tamaño de lote de 32, lo que significa que se procesarán 32 muestras a la vez antes de actualizar los pesos del modelo. El tamaño del lote se determina en función de la eficiencia computacional y la capacidad de memoria.

Descripción de las Capas Codificadas

- **Capa de entrada:** Recibe los pares de texto que se deben comparar.
- **Capa de Embedding:** Para representar las palabras de manera numérica, la capa de embedding convierte las secuencias de palabras en vectores densos de tamaño fijo.
- **Capa de Flatten:** Aplana los vectores de embedding para que puedan ser concatenados.
- **Capa de Concatenación:** Concatena los vectores de inserción de los dos textos en un vector único.
- **Capa Dropout:** Utiliza una técnica de regularización conocida como salida para evitar el sobreajuste y mejorar la generalización del modelo.
- **Capas densas (Fully Connected):** Aplica capas densas (totalmente conectadas) para realizar la clasificación final de la similitud entre los textos.
- **Capa de salida:** La capa de salida produce el resultado final del modelo, que es la predicción de la similitud entre los textos. Dado que se trata de un problema de clasificación binaria donde se busca predecir la similitud (valor entre 0 y 1) entre los textos, la función de activación utilizada en la última capa es una función sigmoide.

Anexo 3. Entrevista

Entrevista para el Prototipo de Software anti plagio para la carrera de Ingeniería en Computación de la Universidad Nacional de Loja, utilizando redes neuronales.

Tabla A3 1. Entrevista

Entrevistado:	Ing. Oscar Miguel Cumbicus Pineda Docente de la Carrea de Computación		
Entrevistado:	Ing. Oscar Miguel Cumbicus Pineda Cliente Principal		
Fecha:	24-04-2023 15H30-16H30	Duración:	1:00
Responsables:	Jaime Oswaldo Paqui Medina y Santiago Alexander Román Silva Autores del Trabajo de Titulación		
ENTREVISTA			
Pregunta 1	¿Quiénes harán uso del software?		
Respuesta	Docentes y estudiantes		
Pregunta 2	¿Qué características considera que son necesarias en una herramienta de detección de plagio para el ámbito académico?		
Respuesta	<ul style="list-style-type: none"> • La forma de ingreso debe ser fácil, permitiendo seleccionar si se trata de un docente o alumno. • Debe permitir la detección en diferentes tipos de fuentes • Debe tener compatibilidad con diferentes formatos. • Debe generar informes detallados. • Debe poseer un CRUD de los documentos subidos. 		
Pregunta 3	¿En qué formato estarán los documentos que se analizarán en el software?		
Respuesta	PDF		
Pregunta 4	¿Qué nivel de similitud se considera aceptable antes de que un documento sea considerado plagio?		
Respuesta	<ul style="list-style-type: none"> • 0-5% aceptable • 6-15% bajo • 16-25% medio • 26-100% alto 		
Pregunta 5	¿Cómo le gustaría que el software muestre los resultados del análisis de similitud? ¿En forma de porcentaje, gráficos, informes detallados, etc.?		
Respuesta	<ul style="list-style-type: none"> • detallada • porcentaje 		
Pregunta 6	¿Considera necesario que el software permitiera a los usuarios guardar sus documentos analizados para una referencia futura?		
Respuesta	Si, el sistema debe guardar los documentos en una carpeta para llevar un mejor control de los documentos.		

Pregunta 7	¿Desea que el software genere un informe detallado sobre los resultados del análisis de plagio?
Respuesta	sí, que genere un Historial de los documentos analizados
Pregunta 8	¿Considera necesario que el software tenga la capacidad de permitir que el docente comparta los resultados del análisis de plagio?
Respuesta	Si, mediante correo electrónico.
Responsables (Firma Digital)	
Cliente Principal:	Ing. Oscar Miguel Cumbicus Pineda (Director de Tesis)
Autores:	Jaime Oswaldo Paqui Medina (Tesisista) Santiago Alexander Roman Silva (Tesisista)

Anexo 4. Especificación de requisitos de software según el estándar IEEE830.

Sistema de anti plagio para la carrera de Computación de la Universidad Nacional de Loja

Elaborado por:

Jaime Oswaldo Paqui Medina

Santiago Alexander Román Silva

Revisado por:

Ing. Oscar Miguel Cumbicus Pineda

Aprobado por:

Ing. Oscar Miguel Cumbicus Pineda

TABLA DE CONTENIDO

Historial de versiones	99
Información del Trabajo.....	99
1. Introducción	100
1.2. Propósito	100
1.3. Alcance del Producto.....	100
1.3. Referencias.....	101
1.4. Personal involucrado en el Desarrollo del Sistema.....	101
1.4.1. Roles de usuarios.....	102
1.4.2. Entorno Operativo	102
1.4.2.1. Funcionamiento de auditoría.....	102
1.4.2.2. De control de acceso	102
2. Requerimientos de alto nivel	102
2.1. Requerimientos Funcionales.....	102
3. Requerimientos no Funcionales	111
4. Historias de usuario	114
5. Aprobaciones.....	120

Historial de versiones

Versión	Fecha	Responsable	Descripción de la modificación
1	06-04-2023	Ing. Oscar Miguel Cumbicus	Documento inicial
2	04-05-2023	Ing. Oscar Miguel Cumbicus	Nuevos Requerimientos e Historias de usuario

Información del Trabajo

Tabla 1. Trabajo para la UNL.

Unidad requirente	Carrera de Computación
Nombre del Trabajo	Sistema de anti plagio para la carrera de Computación de la Universidad Nacional de Loja.
Fecha de elaboración	01-04-2023
Solicitante / Patrocinador principal	Carrera de Computación - Universidad Nacional de Loja

1. Introducción

En el presente documento se describe la Especificación de Requisitos de Software (ERS) del “**Sistema de Detección de Plagio.**”. La presente ERS se basa en estándares IEEE 830, tomando como base para la especificación de los requisitos, encuestas realizadas a los Docentes de la Carrera de Computación de la Universidad Nacional de Loja.

1.2. Propósito

En la Carrera de Computación de la Universidad Nacional de Loja, se ha identificado la necesidad de mejorar los procesos de análisis de documentos de Trabajos de Titulación (TT) con el fin de detectar plagio. Por tanto, se prevé la implementación de un sistema de detección de plagio de código abierto, específico para la carrera, que permita a los integrantes de la misma tener un mayor control sobre el plagio en los documentos. Con ello se busca optimizar el proceso de análisis de documentos y garantizar la integridad académica en la realización de los TT.

1.3. Alcance del Producto

La información y el uso, son gestionados por los integrantes de la Carrera de Computación de la UNL, esta área podrá, administrar y evaluar la funcionalidad del Sistema, de acuerdo a las necesidades requeridas.

El presente Trabajo de sistema tiene como alcance desarrollar el sistema que permita:

- Cargar documentos: El sistema debe permitir al usuario cargar documentos en diferentes formatos (PDF, Word, TXT, etc.) para ser comparados con otros documentos.
- Análisis de plagio: El sistema debe realizar una comparación exhaustiva entre los documentos cargados y las fuentes de comparación seleccionadas, utilizando técnicas de análisis de similitud para detectar posibles plagios.
- Alertas de detección de plagio: El sistema debe ser capaz de generar alertas de detección de plagio y mostrar los resultados al usuario.
- Gestión de resultados: El usuario debe poder gestionar los resultados del análisis de plagio, como guardarlos para su posterior uso o eliminarlos.

1.3. Referencias

Como referencia para el desarrollo del sistema de detección de plagio se tiene:

Tabla 2. Referencias

Referencia	Título	Ruta
Encuesta	Sistema de anti plagio para la Universidad Nacional de Loja	https://docs.google.com/forms/d/1P-WUkaI6ruc02s2FtNMeSIF3JyHWP0ZGM6ygmAPvTHI/edit

1.4. Personal involucrado en el Desarrollo del Sistema

Tabla 3. Personal Involucrado

Nombre	Jaime Oswaldo Paqui Medina
Rol	Usuario del producto
Categoría Profesional	Estudiantes de la Carrera de la UNL
Responsabilidad	Estudiante a cargo de la ejecución del Trabajo
Información de contacto	jaime.paqui@unl.edu.ec

Tabla 4. Personal Involucrado

Nombre	Santiago Alexander Román Silva
Rol	Usuario del producto
Categoría Profesional	Estudiantes de la Carrera de la UNL
Responsabilidad	Estudiante a cargo de la ejecución del Trabajo
Información de contacto	santiago.roman@unl.edu.ec

Tabla 5. Personal Involucrado

Nombre	Oscar Miguel Pineda Cumbicus
Rol	Cliente del producto
Categoría Profesional	Docente de la Carrera de la UNL
Responsabilidad	Docente director del Trabajo de Trabajo de Titulación
Información de contacto	oscar.cumbicus@unl.edu.ec

1.4.1. Roles de usuarios

- **Administrador:** encargado de gestionar el sistema, configurar las opciones de análisis y definir los usuarios y sus permisos de acceso.
- **Docente:** persona que utiliza el sistema para cargar y analizar documentos en busca de posibles plagios.
- **Estudiante:** persona que utiliza el sistema para cargar los documentos y visualizar los resultados.
- **Desarrollador:** encargado de mantener y actualizar el sistema, solucionar problemas técnicos y asegurar su correcto funcionamiento.

1.4.2. Entorno Operativo

1.4.2.1. Funcionamiento de auditoría

Se almacenarán los registros de actividad o historial de logs de las distintas acciones que ejecuten los usuarios, respecto a la ejecución de una evaluación.

1.4.2.2. De control de acceso

El sistema debe controlar los permisos que tiene cada usuario para su accesibilidad de una manera correcta, de tal forma que pueda acceder la información que le corresponde de acuerdo a su rol.

2. Requerimientos de alto nivel

2.1. Requerimientos Funcionales

Tabla 6. RF01

Identificación del requerimiento:	RF01
Nombre del Requerimiento:	Registrar usuarios
Descripción del requerimiento:	El sistema debe permitir al usuario poder registrarse.

Características:	<p>En el registro el usuario debe seleccionar su rol:</p> <ul style="list-style-type: none"> • Estudiante • Docente <p>Para el registro el usuario debe disponer de los siguientes datos:</p> <ul style="list-style-type: none"> • Nombres • Usuario • Apellidos • Correo • Contraseña • Confirmar contraseña
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03 • RNF04 • RNF05 • RNF08
Prioridad del requerimiento:	Alta

Tabla 7. RF02

Identificación del requerimiento:	RF02
Nombre del Requerimiento:	Iniciar sesión.
Descripción del requerimiento:	El sistema debe permitir a los usuarios iniciar sesión.
Características:	<ul style="list-style-type: none"> • La cuenta del usuario debe estar validada por el administrador para poder iniciar sesión. • El usuario debe utilizar sus credenciales registradas como correo institucional y contraseña para iniciar sesión.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF02 • RNF03 • RNF04 • RNF05
Prioridad del requerimiento:	Alta

Tabla 8. RF03

Identificación del requerimiento:	RF03
Nombre del Requerimiento:	Validar registro de usuarios.
Descripción del requerimiento:	El sistema debe permitir al administrador validar o rechazar el registro de un usuario.
Características:	Para validar o rechazar el registro del usuario el administrador debe disponer de los siguientes datos: <ul style="list-style-type: none"> • Validar • Rechazar
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03 • RNF04 • RNF05 • RNF08
Prioridad del requerimiento:	Alta

Tabla 9. RF04

Identificación del requerimiento:	RF04
Nombre del Requerimiento:	Ver lista de usuarios.
Descripción del requerimiento:	El sistema debe permitir al administrador visualizar a todos los usuarios registrados.
Características:	El sistema debe contar con una lista de usuarios donde el administrador dispondrá de los siguientes datos: <ul style="list-style-type: none"> • Nombre • Apellido • Correo • Rol
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03 • RNF04 • RNF05

	<ul style="list-style-type: none"> • RNF08
Prioridad del requerimiento:	Alta

Tabla 10. RF05

Identificación del requerimiento:	RF05
Nombre del Requerimiento:	Cambiar rol de cuentas de usuario.
Descripción del requerimiento:	El sistema debe permitir al administrador modificar el rol de la cuenta de los usuarios registrados.
Características:	El administrador podrá cambiar el rol de cuenta, para ello deberá disponer las opciones: <ul style="list-style-type: none"> • Administrador • Docente. • Estudiante
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03 • RNF04 • RNF05 • RNF08
Prioridad del requerimiento:	Alta

Tabla 11. RF06

Identificación del requerimiento:	RF06
Nombre del Requerimiento:	Editar usuarios
Descripción del requerimiento:	El sistema debe permitir al administrador modificar los datos de los usuarios.

Características:	El administrador podrá editar datos de los usuarios registrados tales como: <ul style="list-style-type: none"> • Nombre • Apellido • Correo Electrónico.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03 • RNF04 • RNF05 • RNF08
Prioridad del requerimiento:	Alta

Tabla 12. RF07

Identificación del requerimiento:	RF07
Nombre del Requerimiento:	Eliminar usuarios
Descripción del requerimiento:	El sistema debe permitir al administrador eliminar a los usuarios registrados.
Características:	El administrador podrá eliminar a los usuarios registrados, ya sea que estos usuarios estén con la cuenta activa o inactiva.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03 • RNF04 • RNF05 • RNF08
Prioridad del requerimiento:	Alta

Tabla 13. RF08

Identificación del requerimiento:	RF08
--	------

Nombre del Requerimiento:	Analizar plagio por el Estudiante.
Descripción del requerimiento:	El sistema permitirá al estudiante, agregar documentos al sistema para su respectivo análisis.
Características:	<ul style="list-style-type: none"> • El sistema permitirá al estudiante cargar un documento a la vez en los siguientes formatos: PDF y TXT. • El usuario estudiante debe disponer de los siguientes datos: <ul style="list-style-type: none"> ○ Correo electrónico (destino donde le llegara el resultado de plagio). ○ Descripción. ○ Analizar plagio. • El sistema debe ser capaz de detectar plagio en diferentes tipos de fuentes, como libros, artículos, revistas de internet, páginas web y trabajos TIC. • Una vez terminado el análisis se presenta el resultado automáticamente.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03 • RNF04 • RNF05
Prioridad del requerimiento:	Alta

Tabla 14. RF09

Identificación del requerimiento:	RF09
Nombre del Requerimiento:	Analizar plagio por el Docente.
Descripción del requerimiento:	El sistema permitirá al docente, agregar documentos al sistema para su respectivo análisis.
Características:	<ul style="list-style-type: none"> • El sistema permitirá al docente cargar un documento a la vez en los siguientes formatos: PDF y TXT. • El sistema debe ser capaz de detectar plagio en diferentes tipos de fuentes, como libros, artículos, revistas de internet, páginas web y trabajos TIC. • Una vez terminado el análisis se presenta el resultado automáticamente.

Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03 • RNF04 • RNF05
Prioridad del requerimiento:	Alta

Tabla 15. RF10

Identificación del requerimiento:	RF10
Nombre del Requerimiento:	Compartir resultados de análisis de plagio.
Descripción del requerimiento:	El sistema debe permitir a los usuarios compartir el análisis del documento plagiado.
Características:	<ul style="list-style-type: none"> • En la lista de resultados de plagio el usuario podrá compartir dicho resultado para ello debe disponer de los siguientes datos: <ul style="list-style-type: none"> ○ Correo electrónico ○ Descripción ○ Compartir
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03 • RNF04 • RNF05
Prioridad del requerimiento:	Alta

Tabla 16. RF11

Identificación del requerimiento:	RF11
Nombre del Requerimiento:	Ver listado de documentos analizados por el usuario.
Descripción del requerimiento:	El usuario podrá visualizar todos los documentos que hayan sido analizados.

Características:	El sistema debe mostrar el nombre de los documentos con las respectivas opciones: <ul style="list-style-type: none"> • Eliminar. • Ver análisis detallado. • Descargar
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF02 • RNF03 • RNF04 • RNF05
Prioridad del requerimiento:	Alta

Tabla 17. RF12

Identificación del requerimiento:	RF12
Nombre del Requerimiento:	Ver resultados de análisis de plagio.
Descripción del requerimiento:	El sistema debe mostrar el resultado de análisis de plagio.
Características:	El usuario podrá visualizar: <ul style="list-style-type: none"> • Porcentaje de plagio • Párrafo original • Párrafo plagiado • Lugar donde se detectó el plagio.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF02 • RNF03 • RNF04 • RNF05
Prioridad del requerimiento:	Alta

Tabla 18. RF13

Identificación del requerimiento:	RF13
Nombre del Requerimiento:	Editar resultados de análisis de plagio.
Descripción del requerimiento:	El sistema debe permitir al docente editar el detalle del análisis de plagio.
Características:	El docente dispondrá de un botón que le permitirá modificar los párrafos plagiados, esto con la finalidad de bajar el nivel de plagio.
Requerimiento o NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03 • RNF04 • RNF05
Prioridad del requerimiento:	Alta

Tabla 19. RF14

Identificación del requerimiento:	RF14
Nombre del Requerimiento:	Eliminar resultados de análisis de plagio.
Descripción del requerimiento:	El sistema debe permitir al usuario eliminar el resultado del análisis de plagio.
Características:	El docente podrá eliminar el resultado de plagio, para ello dispondrá de una opción de eliminar.
Requerimiento o NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03 • RNF04 • RNF05
Prioridad del requerimiento:	Alta

Tabla 20. RF15

Identificación del requerimiento:	RF15
Nombre del Requerimiento:	Descargar resultados de análisis de plagio.
Descripción del requerimiento:	El sistema debe permitir al usuario descargar los resultados obtenidos.
Características:	El usuario podrá descargar el resultado de plagio, para ello dispondrá de una opción de descargar.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03 • RNF04 • RNF05
Prioridad del requerimiento:	Alta

3. Requerimientos no Funcionales

Tabla 20. RNF01

Identificación del requerimiento:	RNF01
Nombre del Requerimiento:	Desempeño
Características:	El sistema asegurará a los usuarios un rendimiento fiable en lo que respecta a los datos almacenados en el sistema, brindando así una garantía de confiabilidad.
Descripción del requerimiento:	Asegurar el funcionamiento del sistema informático para todos los usuarios, permitiendo la consulta y actualización constante y simultánea de la información almacenada o los registros realizados, sin afectar el tiempo de respuesta.
Prioridad del requerimiento:	Alta

Tabla 21. RNF02

Identificación del requerimiento:	RNF02
Nombre del Requerimiento:	Mantenibilidad.
Características:	Facilidad para ser modificado.
Descripción del requerimiento:	La implementación del sistema deberá tener la capacidad de ser fácilmente modificado o actualizado sin causar problemas o complicaciones adicionales.
Prioridad del requerimiento:	Alta

Tabla 22. RNF03

Identificación del requerimiento:	RNF03
Nombre del Requerimiento:	Usabilidad
Características:	Facilidad para ser usado por el usuario final.
Descripción del requerimiento:	El sistema debe tener una interfaz fácil de usar, amigable con el usuario de manera que los usuarios puedan manejarlo sin dificultades.
Prioridad del requerimiento:	Alta

Tabla 23. RNF04

Identificación del requerimiento:	RNF04
Nombre del Requerimiento:	Alertas de errores
Características:	Presentar mensajes de error.

Descripción del requerimiento:	El sistema debe ser capaz de mostrar mensajes de error cuando el usuario utilice incorrectamente algunas funcionalidades del software.
Prioridad del requerimiento:	Alta

Tabla 24. RNF05

Identificación del requerimiento:	RNF05
Nombre del Requerimiento:	Seguridad
Características:	El sistema debe contar con medidas de seguridad para evitar el acceso no autorizado a los documentos cargados y garantizar la integridad de los resultados del análisis de plagio.
Descripción del requerimiento:	El sistema deberá limitar el acceso únicamente a usuarios que hayan sido autenticados y que cuenten con los permisos necesarios para utilizar el sistema.
Prioridad del requerimiento:	Alta

Tabla 25. RNF06

Identificación del requerimiento:	RNF06
Nombre del Requerimiento:	Disponibilidad
Características:	El sistema debe estar operativo sin interrupciones durante todos los días de la semana.
Descripción del requerimiento:	El sistema debe estar siempre disponible para los usuarios, asegurando que esté diseñado de manera apropiada para enfrentar posibles fallas en cualquiera de sus partes.
Prioridad del requerimiento:	Alta

4. Historias de usuario

ID: H001		Registrar usuarios.	
Usuario / Rol	Administrador, Docente, Estudiante	Prioridad	Alta
		Puntos estimados	8
Descripción	Deseo poder registrarme en el sistema.		
Nro.	Escenario	Criterio de aceptación	
1	Registrar Usuarios	<p>El sistema debe permitir seleccionar un rol de usuario ya sea como docente o estudiante.</p> <p>El sistema debe permitir ingresar nombre, apellido, correo y contraseña del usuario.</p> <p>El sistema debe validar que el correo electrónico sea único y que pertenezca a la Universidad Nacional de Loja.</p> <p>El sistema debe mostrar un mensaje de error apropiado si algún campo requerido no se completa o si se ingresan datos no válidos.</p> <p>Después de enviar el formulario, el sistema debe almacenar los detalles del usuario en un estado "pendiente" o "no verificado" hasta que sean validados por el administrador.</p> <p>El sistema debe enviar una notificación al administrador cuando un nuevo usuario se registre y requiera validación.</p>	

ID: H005		Iniciar sesión.	
Usuario / Rol	Administrador, Docente y Estudiante	Prioridad	Alta
		Puntos estimados	8
Descripción	Deseo poder iniciar sesión para acceder a las funcionalidades de la plataforma. El inicio de sesión deberá permitir a los usuarios ingresar su correo electrónico y contraseña para autenticarse en el sistema.		

Nro.	Escenario	Criterio de aceptación
1	El usuario ingresa al sistema con sus credenciales (nombre de usuario y contraseña)	<p>Debe contar con una interfaz amigable con el usuario</p> <p>Debe tener la opción olvide mi contraseña para que los usuarios puedan recuperarla en caso de ser necesario.</p> <p>La autenticación deberá ser segura y protegida mediante medidas de seguridad adecuadas para proteger la información de los usuarios.</p>

ID: H002		Validar registro de usuarios.	
Usuario / Rol	Administrador	Prioridad	Alta
		Puntos estimados	8
Descripción	Deseo poder validar el registro de usuarios.		
Nro.	Escenario	Criterio de aceptación	
1	Validar registro de usuarios.	<p>El administrador debe tener acceso a una interfaz para revisar y validar los registros de los usuarios.</p> <p>El administrador debe poder ver la información del usuario para validar el registro.</p> <p>El administrador debe tener la capacidad de aprobar el registro del usuario pendiente.</p> <p>Si el administrador aprueba el registro, el sistema debe actualizar el estado del usuario a "verificado" y enviar una notificación al usuario informándole que su cuenta ha sido validada.</p> <p>Si el administrador rechaza el registro, el sistema debe eliminar el registro del usuario y enviar una notificación al usuario explicando el motivo del rechazo.</p> <p>El sistema debe proporcionar una interfaz intuitiva y fácil de usar para el administrador, facilitando la revisión y validación eficiente de los registros de usuarios pendientes.</p>	

ID: H004		Administrar usuarios.	
Usuario / Rol	Administrador	Prioridad	Alta
		Puntos estimados	8
Descripción	Deseo poder gestionar a los usuarios registrados, así mismo quiero asignar permisos de acceso basado en roles (Docente y Estudiante) para controlar las acciones permitidas en función de los permisos asignados.		
Nro.	Escenario	Criterio de aceptación	
1	Administración de usuarios	<p>Debe contar con un registro de los usuarios registrados en el sistema.</p> <p>Debe poder gestionar a los usuarios registrados (eliminar, cambiar estado de cuenta, editar, asignar rol.)</p>	

ID: H006		Analizar plagio por el Estudiante	
Usuario / Rol	Estudiante	Prioridad	Alta
		Puntos estimados	8
Descripción	Deseo poder analizar un documento en búsqueda de plagio.		
Nro.	Escenario	Criterio de aceptación	
1	Realizar el análisis de plagio	<p>El sistema debe permitir subir documentos en formatos PDF y TXT, para que el estudiante pueda realizar el respectivo análisis.</p> <p>El sistema debe mostrar un formulario para que el estudiante ingrese el correo electrónico del docente a quien se enviara el resultado de plagio.</p> <p>El sistema debe detectar plagio en diferentes tipos de fuentes como libros, artículos, revistas de internet, páginas web y trabajos TIC.</p> <p>El sistema deberá mostrar automáticamente el resultado de plagio, al finalizar el análisis.</p> <p>El sistema deberá guardar los documentos analizados para una referencia futura, lo que será útil para recordar</p>	

	qué documentos se han analizado previamente.
--	--

ID: H007		Analizar plagio por el Docente	
Usuario / Rol	Docente	Prioridad	Alta
		Puntos estimados	8
Descripción	Deseo poder analizar un documento en búsqueda de plagio.		
Nro.	Escenario	Criterio de aceptación	
1	Realizar el análisis de plagio	<p>El sistema debe permitir subir documentos en formatos PDF y TXT, para que los usuarios puedan realizar el respectivo análisis.</p> <p>El sistema debe detectar plagio en diferentes tipos de fuentes como libros, artículos, revistas de internet, páginas web y trabajos TIC, lo que ayudará a garantizar la integridad académica del trabajo presentado.</p> <p>El sistema deberá mostrar automáticamente el resultado de plagio, al finalizar el análisis.</p> <p>El sistema deberá guardar los documentos analizados para una referencia futura, lo que será útil para recordar qué documentos se han analizado previamente.</p>	

ID: H008		Compartir resultados de análisis de plagio.	
Usuario / Rol	Docente, Estudiante.	Prioridad	Alta
		Puntos estimados	8
Descripción	Deseo poder compartir el resultado de mi análisis de plagio para cualquier usuario.		
Nro.	Escenario	Criterio de aceptación	
1	Compartir análisis de plagio.	<p>Se debe ingresar una dirección de correo electrónico a quien desea compartir el resultado del análisis y agrega una pequeña descripción sobre el análisis de plagio.</p> <p>El sistema debe permitir al usuario poder compartir un</p>	

		<p>resultado a la vez.</p> <p>El sistema deberá contar con una opción de "compartir resultados" que permita al usuario seleccionar el informe de análisis que desea enviar por correo electrónico.</p> <p>El sistema deberá adjuntar automáticamente el informe de análisis en formato PDF al correo electrónico enviado.</p>
--	--	---

ID: H010		Ver listado de documentos analizados por el usuario.	
Usuario / Rol	Docente y Estudiante	Prioridad	Alta
		Puntos estimados	8
Descripción	Deseo visualizar todos los documentos a los que se ha hecho el respectivo análisis.		
Nro.	Escenario	Criterio de aceptación	
1	Ver todos los documentos analizados	El sistema debe mostrar una lista de los documentos analizados, donde el usuario podrá gestionar estos documentos como (eliminar, visualizar análisis).	

ID: H011		Ver resultado de análisis de plagio	
Usuario / Rol	Docente y Estudiante	Prioridad	Alta
		Puntos estimados	8
Descripción	Deseo poder ver de forma detallada el análisis de plagio que se ha realizado anteriormente a los documentos.		
Nro.	Escenario	Criterio de aceptación	
1	Ver el detalle del análisis de plagio.	El usuario dispondrá de una lista de documentos analizados, en el cual podrá acceder a cada una de ellos para ver de forma más detallada el análisis de plagio.	

		El sistema debe mostrar al usuario el porcentaje de plagio, el párrafo original, el párrafo plagiado, fecha del análisis y lugar donde fue detectado el plagio.
--	--	---

ID: H012	Reducir el nivel de plagio de los documentos.		
Usuario / Rol	Docente	Prioridad	Alta
		Puntos estimados	8
Descripción	Deseo poder reducir el nivel de plagio de los documentos analizados, como por ejemplo excluyendo párrafos plagiados.		
Nro.	Escenario	Criterio de aceptación	
1	Reducir el nivel de plagio.	Se debe reducir el nivel de plagio, modificando el resultado de análisis de plagio. Se puede excluir uno o muchos párrafos plagiados.	

ID: H013	Descargar resultado de análisis de plagio		
Usuario / Rol	Investigador	Prioridad	Alta
		Puntos estimados	8
Descripción	Deseo poder descargar los resultados de análisis de plagio.		
Nro.	Escenario	Criterio de aceptación	
1	Descargar Resultados	El sistema deberá contar con una opción de "Exportar o Descargar" que le permita al usuario poder realizar la respectiva descarga de manera satisfactoria. El sistema debe asegurar que los usuarios puedan exportar los resultados en formato PDF para que puedan guardarlos o compartirlos fácilmente con otras personas.	

5. Aprobaciones

Nombre y Apellido	Cargo / Rol	Firma
Ing. Oscar Miguel Cumbicus Pineda	Cliente del producto	

Anexo 5. Interfases de usuario de los Requerimientos Funcionales.

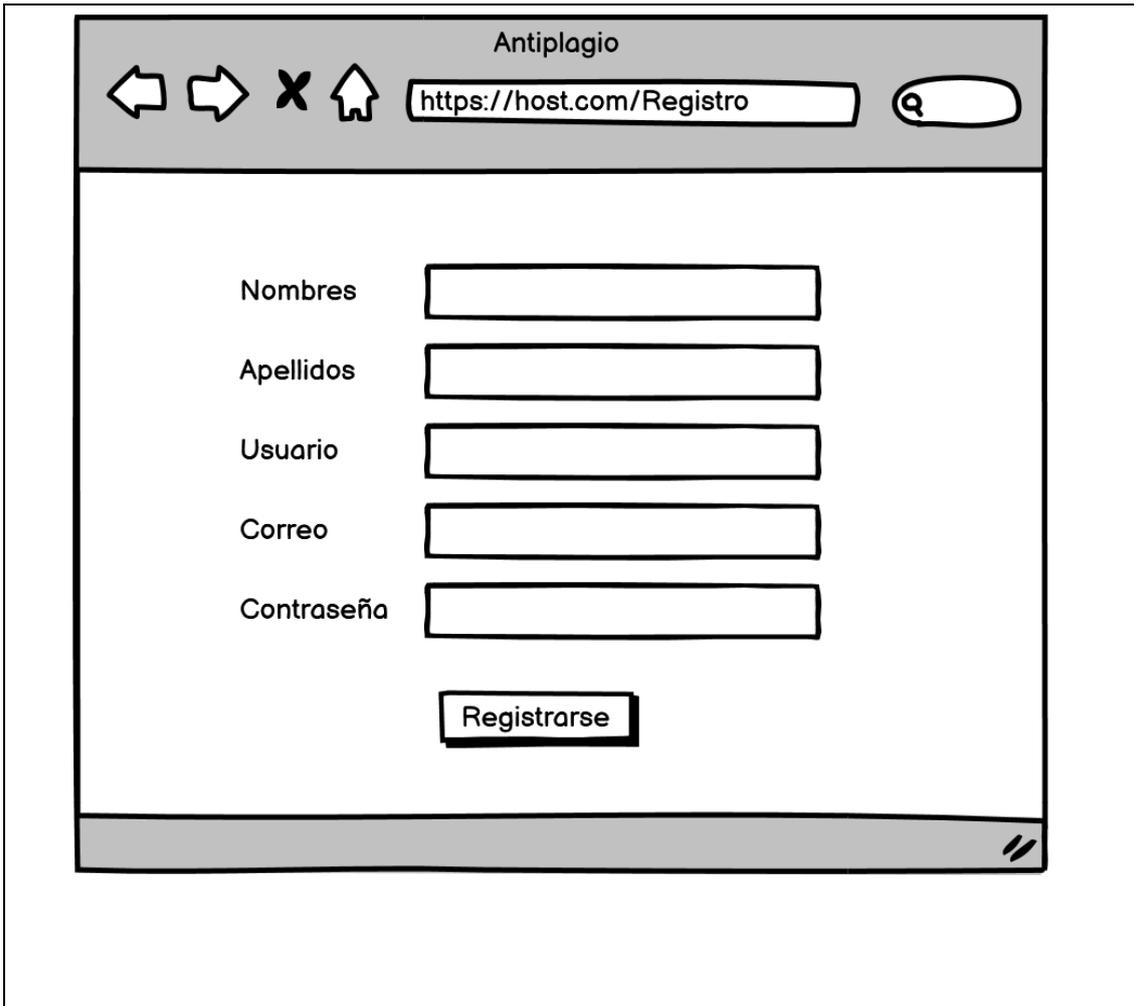
Requisito Funcional Registrar Usuarios

Tabla A4 1. CASO DE USO REGISTRAR USUARIOS

Identificador	CU01
Nombre	Registro de usuarios
Precondición	El usuario debe poseer un correo que pertenezca a la UNL.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario da clic en registrar que la parte superior derecha. 2. Se presenta un formulario donde debe ingresar sus datos. 3. El usuario da clic el botón continuar. 4. El usuario selecciona un rol dando clic en unos de los botones ya sea estudiante o docente. 5. Al usuario se le presenta una página donde le mostrará un mensaje avisando que su cuenta será validada. 6. Al usuario le llega por correo electrónico la validación o rechazo de registro de cuenta.
Flujo alternativo	<ol style="list-style-type: none"> 1. Si no dispone de un correo electrónico válido, el formulario mostrará un mensaje de error. 2. Si no se llena el formulario correctamente o si se ingresan datos ya existentes, se presenta el mensaje de error.
Requisito	RF01

Tabla A4 2. INTERFAZ REGISTRAR USUARIO

Interfaz Registrar usuario		
Nro. Interfaz	1	
Estado de pruebas	Aceptada sin cambios	si
	Aceptada con cambios	
	Rediseñar	
Requisito funcional	RF01	
Observaciones		



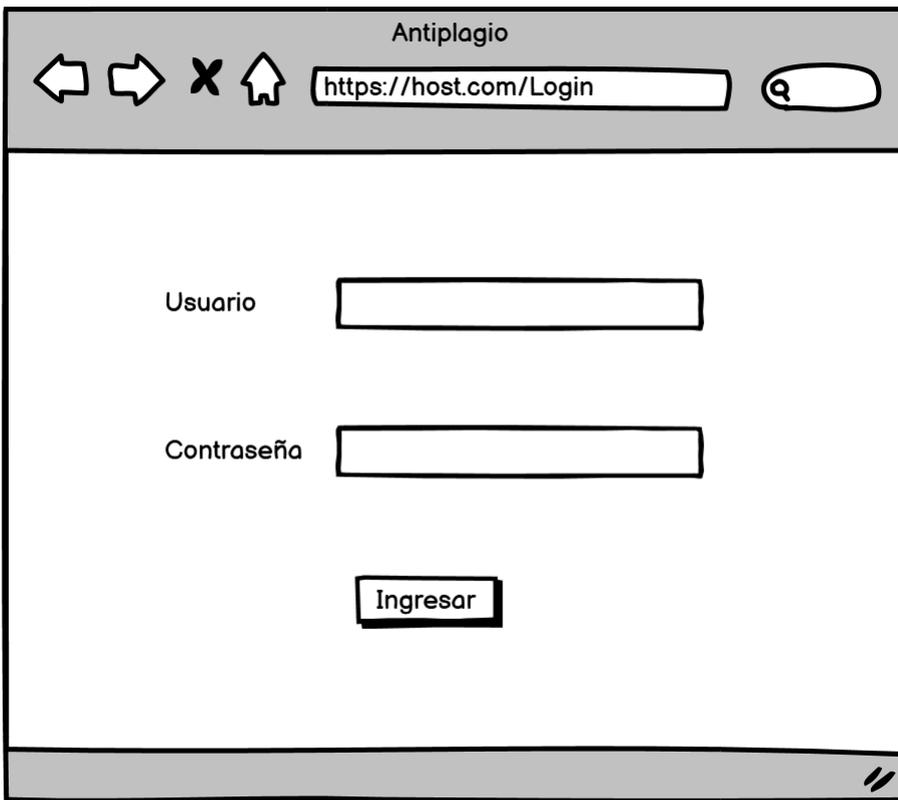
Requisito Funcional Iniciar Sesión

Tabla A5 1.

Tabla A4 3. CASO DE USO INGRESAR AL SISTEMA

Identificador	CU02
Nombre	Iniciar sesión
Precondición	El usuario debe tener una cuenta registrada y validada por el administrador.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario ingresa al botón iniciar sesión ubicado en la parte superior derecha de la página de inicio. 2. Se le presenta al usuario un formulario. 3. El usuario ingresa sus credenciales: correo y contraseña. 4. El usuario selecciona el botón iniciar sesión.
Flujo alternativo	<ol style="list-style-type: none"> 1. Si no se llena el formulario correctamente o si se ingresan datos ya existentes, se presenta el mensaje de error.
Requisito	RF02

Tabla A4 4. INTERFAZ INGRESAR AL SISTEMA

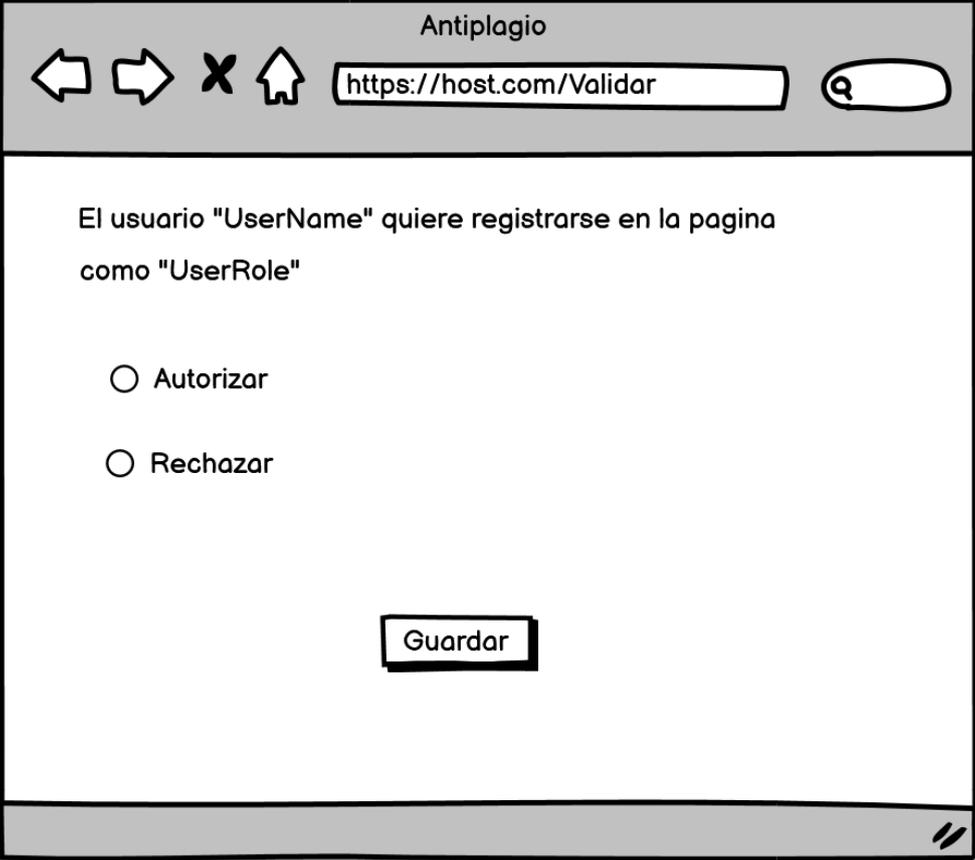
Interfaz ingresar al sistema		
Nro. Interfaz	1	
Estado de pruebas	Aceptada sin cambios	si
	Aceptada con cambios	
	Rediseñar	
Requisito funcional	RF02	
Observaciones		
		

Requisito Funcional Validar registro de usuario

Tabla A4 5. CASO DE USO VALIDAR REGISTRO USUARIO

Identificador	CU03
Nombre	Validar o rechazar registro
Precondición	El administrador debe tener los permisos necesarios. El administrador debe estar autenticado en el sistema.
Flujo principal	<ol style="list-style-type: none"> 1. Al administrador le llega la notificación por correo de que un nuevo usuario solicita la validación de su registro. 2. El administrador abre el correo electrónico y verifica la información del usuario. 3. El administrador proceder a realizar la acción de validar o rechazar.
Flujo alternativo	<ol style="list-style-type: none"> 1. Si el administrador no valida al usuario, este no podrá acceder al sistema.
Requisito	RF03

Tabla A4 6. INTERFAZ VALIDAR REGISTRO DE USUARIO

Interfaz Validar registro de usuario		
Nro. Interfaz	1	
Estado de pruebas	Aceptada sin cambios	si
	Aceptada con cambios	
	Rediseñar	
Requisito funcional	RF03	
Observaciones		
		

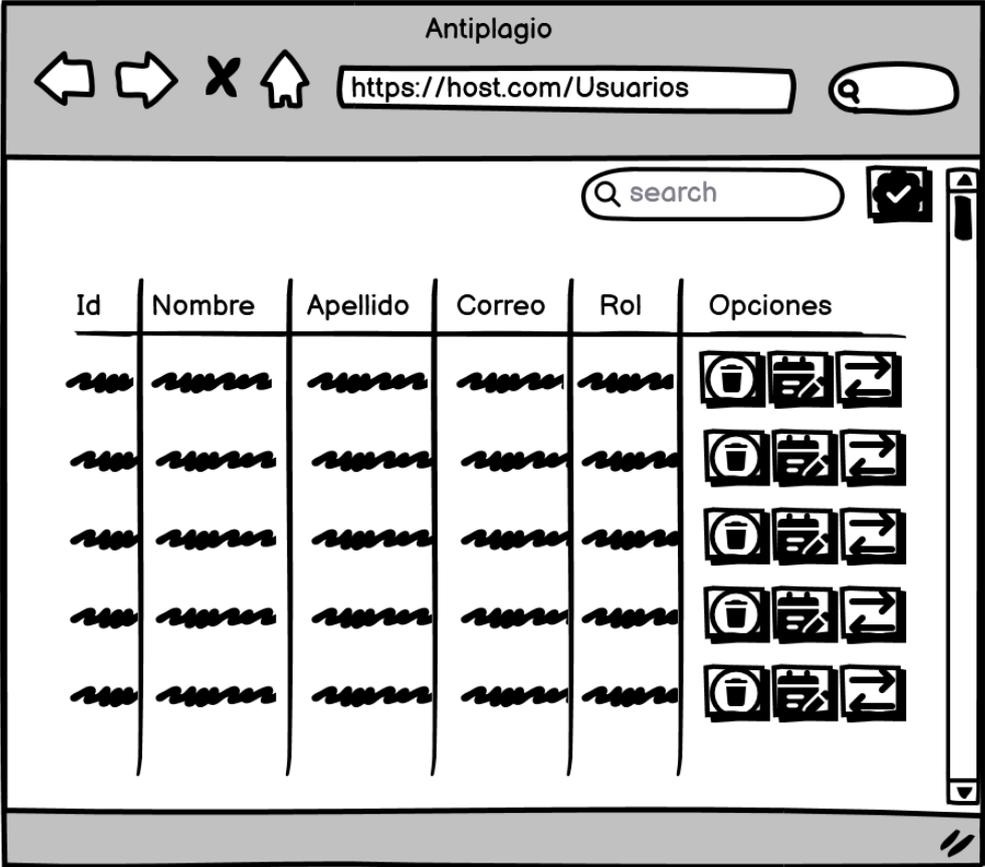
Requisito Funcional Ver lista de usuarios

Tabla A4 7. CASO DE USO VER LISTA DE USUARIOS

Identificador	CU04
Nombre	Ver lista de usuarios
Precondición	El administrador debe estar autenticado en el sistema. El administrador debe tener los permisos necesarios.
Flujo principal	1. El administrador ingresa al sistema con sus credenciales.

	2. El sistema le muestra al administrador una lista con todos los usuarios registrados
Flujo alternativo	1. Si no hay usuarios registrados se le presentara al administrador una lista vacía.
Requisito	RF04

Tabla A4 8. INTERFAZ VER LISTA DE USUARIOS

Interfaz ver lista de usuarios		
Nro. Interfaz	1	
Estado de pruebas	Aceptada sin cambios	si
	Aceptada con cambios	
	Rediseñar	
Requisito funcional	RF04	
Observaciones		
		

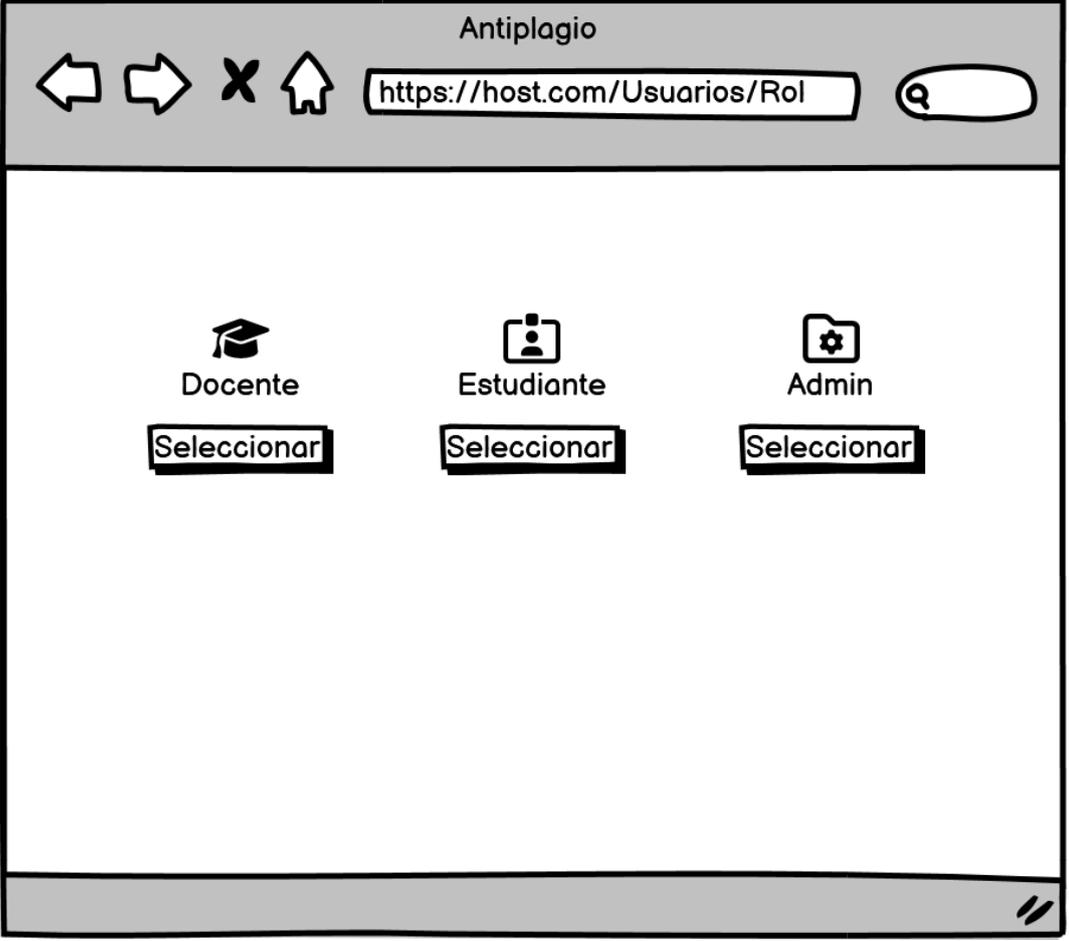
Requisito Funcional Cambiar rol del usuario

Tabla A4 9. CASO DE USO CAMBIAR ROL DEL USUARIO

Identificador	CU05
Nombre	Cambiar el rol de cuentas de usuario.
Precondición	El administrador debe estar autenticado en el sistema. El administrador debe tener los permisos necesarios.
Flujo principal	1. El administrador ingresa al sistema con sus credenciales.

	<ol style="list-style-type: none"> 2. El sistema le muestra al administrador una lista con todos los usuarios registrados. 3. El administrador selecciona el botón cambiar rol. 4. El sistema le muestra un mensaje de confirmación. 5. El administrador selecciona el botón cambiar rol para efectuar los cambios o sino selecciona cancelar para no efectuar los cambios.
Flujo alternativo	<ol style="list-style-type: none"> 1. Si no hay usuarios registrados se le presentara al administrador una lista vacía.
Requisito	RF05

Tabla A4 10. INTERFAZ CAMBIAR ROL DEL USUARIO

Interfaz cambiar estado de cuenta de los usuarios		
Nro. Interfaz	1	
Estado de pruebas	Aceptada sin cambios	si
	Aceptada con cambios	
	Rediseñar	
Requisito funcional	RF05	
Observaciones		
		

Requisito Funcional Editar usuarios

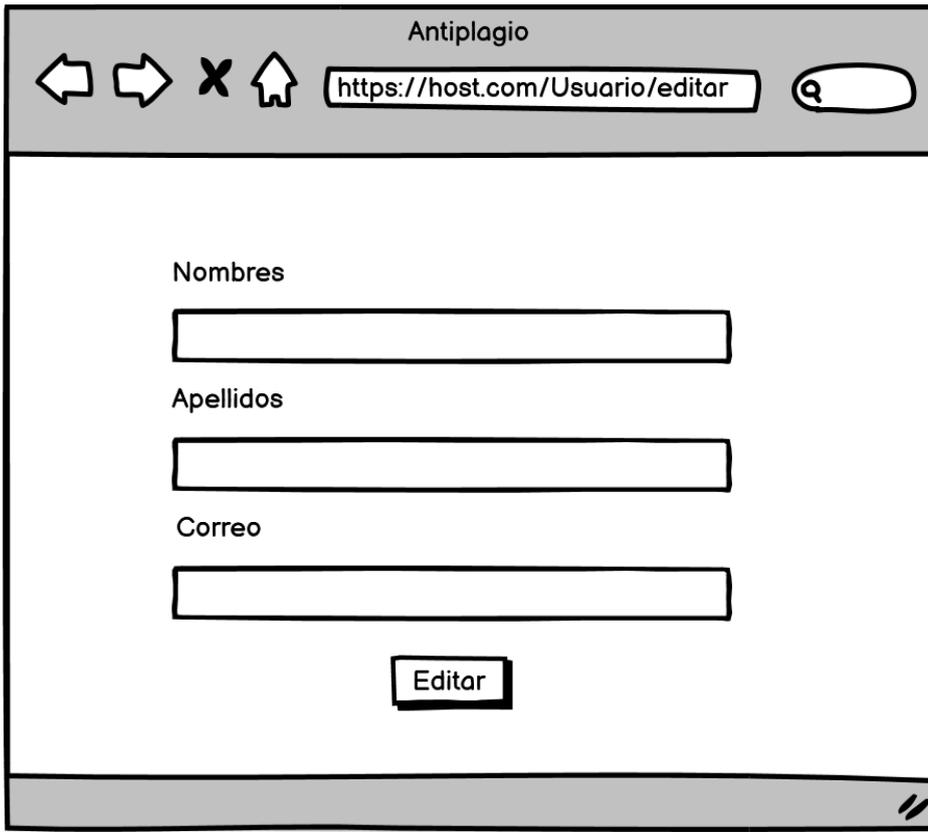
Tabla A4 11. CASO DE USO EDITAR USUARIOS

Identificador	CU06
Nombre	Editar usuarios
Precondición	El administrador debe estar autenticado en el sistema. El administrador debe tener los permisos necesarios.
Flujo principal	<ol style="list-style-type: none"> 1. El administrador ingresa al sistema con sus credenciales. 2. El sistema le muestra al administrador una lista con todos los usuarios registrados. 3. El administrador selecciona el botón editar. 4. El sistema le muestra un formulario. 5. El administrador ingresara los datos a modificar. 6. Posteriormente seleccionara el botón modificar usuario
Flujo alternativo	<ol style="list-style-type: none"> 1. Si no hay usuarios registrados se le presentara al administrador una lista vacía.
Requisito	RF06

Tabla A4 12. INTERFAZ EDITAR USUARIOS

Interfaz editar usuarios		
Nro. Interfaz	1	
Estado de pruebas	Aceptada sin cambios	si
	Aceptada con cambios	
	Rediseñar	
Requisito funcional	RF06	

Observaciones



Requisito Funcional Eliminar usuarios

Tabla A4 13. CASO DE USO ELIMINAR USUARIOS

Identificador	CU07
Nombre	Eliminar usuarios
Precondición	El administrador debe estar autenticado en el sistema. El administrador debe tener los permisos necesarios.
Flujo principal	<ol style="list-style-type: none">1. El administrador ingresa al sistema con sus credenciales.2. El sistema le muestra al administrador una lista con todos los usuarios registrados.3. El administrador selecciona el botón eliminar.4. El sistema le muestra un mensaje de confirmación.5. El administrador selecciona el botón eliminar para efectuar los cambios o sino selecciona cancelar para cancelar la operación.
Flujo alternativo	<ol style="list-style-type: none">1. Si no hay usuarios registrados se le presentara al administrador una lista vacía.
Requisito	RF07

Tabla A4 14. INTERFAZ ELIMINAR USUARIOS

Interfaz eliminar usuarios		
Nro. Interfaz	1	
Estado de pruebas	Aceptada sin cambios	si
	Aceptada con cambios	
	Rediseñar	
Requisito funcional	RF07	
Observaciones		

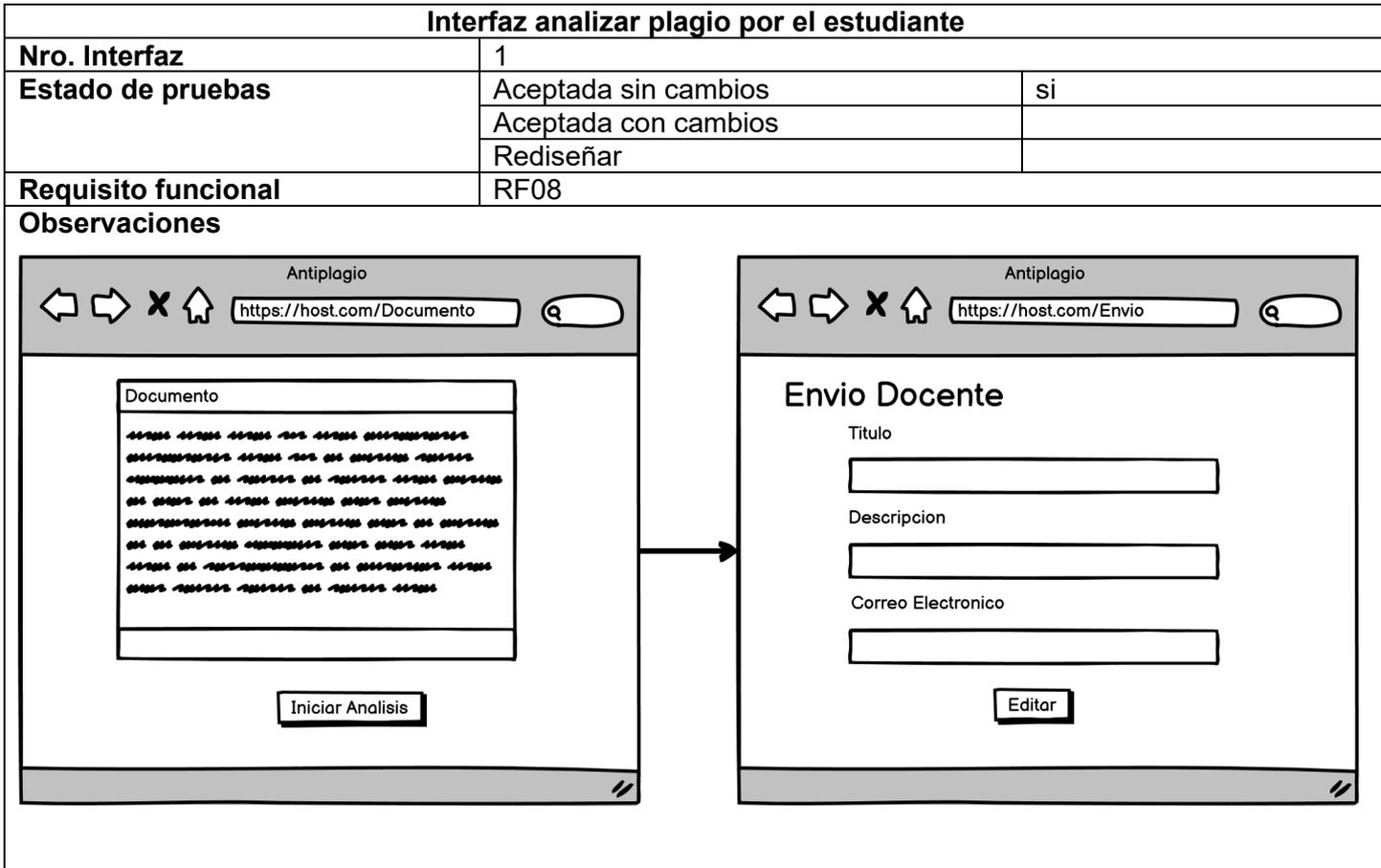
Requisito Funcional Analizar plagio por el Estudiante

Tabla A4 15. CASO DE USO ANALIZAR PLAGIO POR EL ESTUDIANTE

Identificador	CU08
Nombre	Analizar plagio por el estudiante
Precondición	El estudiante debe estar autenticado en el sistema. El estudiante debe subir un documento ya sea en formato pdf o txt.
Flujo principal	<ol style="list-style-type: none"> 1. El estudiante ingresa a la página documentos, ubicada en la página principal. 2. El estudiante selecciona el botón seleccionar documento. 3. El estudiante selecciona el botón subir documento.

	<ol style="list-style-type: none"> 4. El estudiante ingresa el correo electrónico del docente al cual le llegara el resultado de plagio, adjunto a esto también ingresa una breve descripción. 5. El estudiante selecciona el botón analizar plagio. 6. El sistema analiza el plagio y genera el resultado automáticamente. 7. Al estudiante se le muestra en pantalla el resultado de plagio
Flujo alterno	<ol style="list-style-type: none"> 1. Si el estudiante no está autenticado en el sistema se le mostrara la pantalla de inicio de sesión. 2. Si el estudiante intenta subir un documento diferente del formato pdf y txt, se le mostrara un mensaje de error. 3. Si el usuario estudiante selecciona un correo que no pertenezca a un docente, se le mostrara un mensaje de error.
Requisito	RF08

Tabla A4 16. INTERFAZ ANALIZAR PLAGIO POR EL ESTUDIANTE

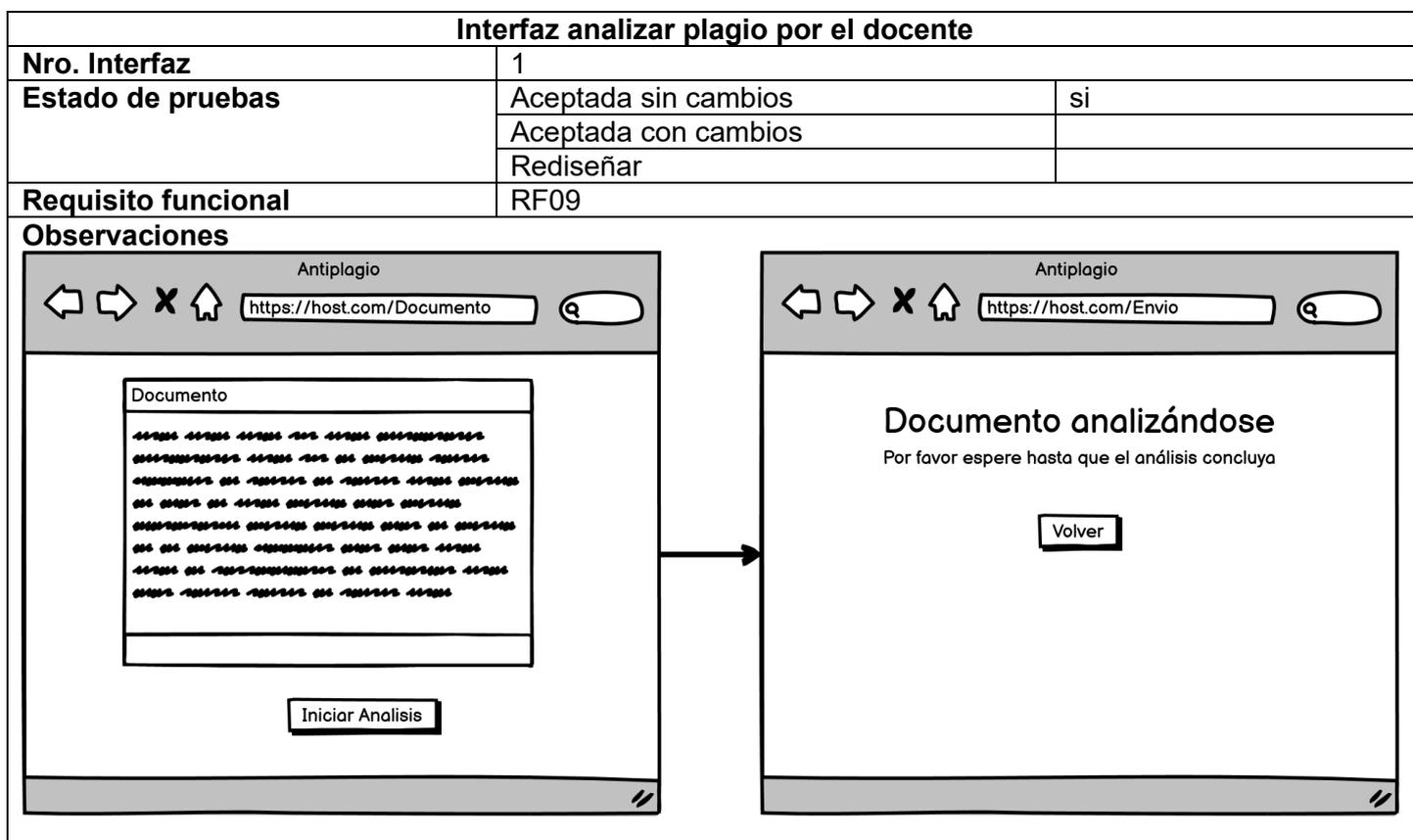


Requisito Funcional Analizar plagio por el Docente

Tabla A4 17. CASO DE USO ANALIZAR PLAGIO POR EL DOCENTE

Identificador	CU09
Nombre	Analizar plagio por el docente
Precondición	El docente debe estar autenticado en el sistema. El docente debe subir un documento ya sea en formato pdf o txt.
Flujo principal	<ol style="list-style-type: none"> 1. El docente ingresa a la página documentos, ubicada en la página principal. 2. El docente selecciona el botón seleccionar documento. 3. El docente selecciona el botón subir documento. 4. El docente selecciona el botón analizar plagio. 5. El sistema analiza el plagio y genera el resultado automáticamente. 6. Al docente se le muestra en pantalla el resultado de plagio
Flujo alterno	<ol style="list-style-type: none"> 7. Si el docente no está autenticado en el sistema se le mostrara la pantalla de inicio de sesión. 8. Si el docente intenta subir un documento diferente del formato pdf y txt, se le mostrara un mensaje de error.
Requisito	RF09

Tabla A4 18. INTERFAZ ANALIZAR PLAGIO POR EL DOCENTE



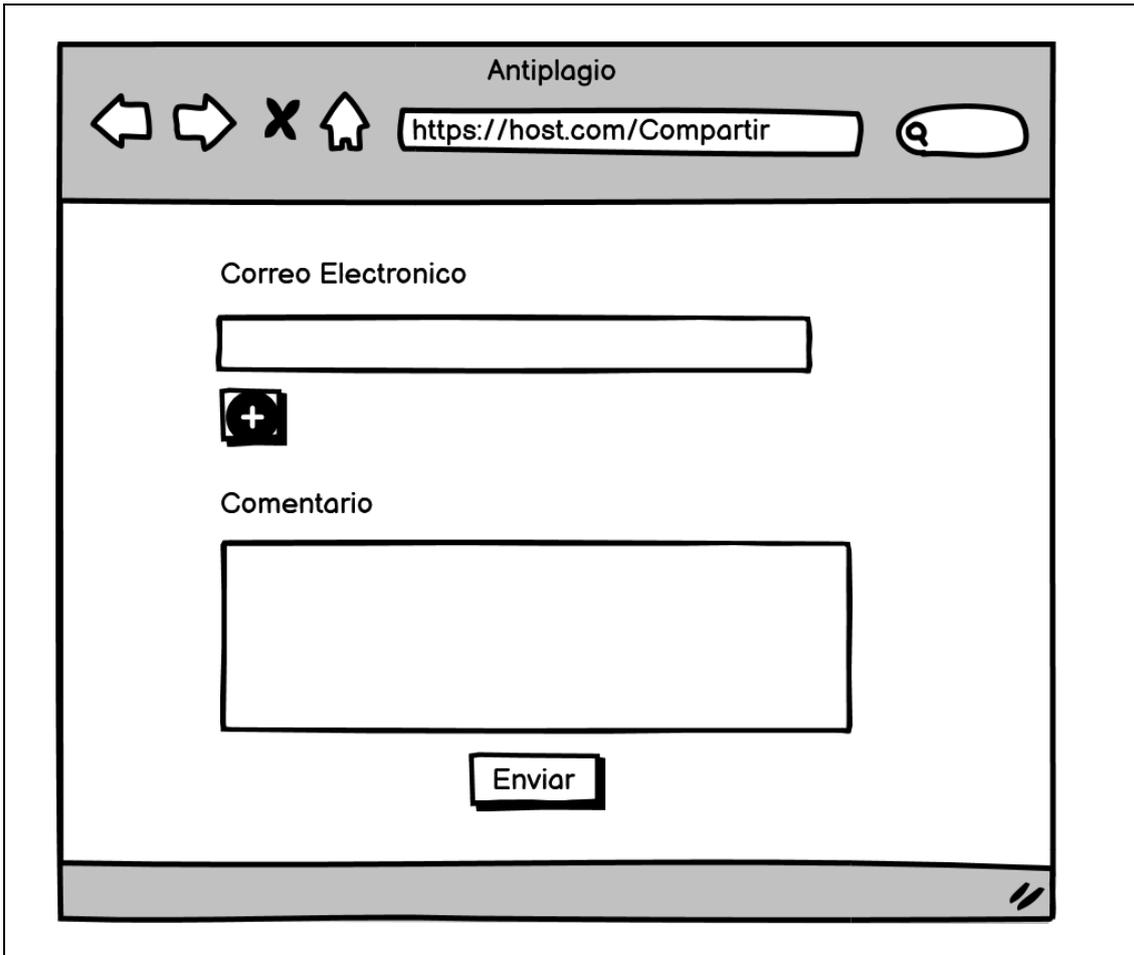
Requisito Funcional Compartir Resultado de análisis de plagio.

Tabla A4 19. CASO DE USO COMPARTIR RESULTADO DE PLAGIO

Identificador	CU10
Nombre	Compartir resultados de plagio
Precondición	El usuario debe estar autenticado en el sistema. El usuario debe poseer un análisis de plagio realizado anteriormente.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario ingresa a la sección resultados de plagio, ubicado en la página principal. 2. Al usuario se le presenta una lista con los documentos analizados. 3. El usuario da clic en revisión para entrar en detalle del resultado. 4. Al usuario se le presenta el detalle del resultado de plagio. 5. El usuario da clic en el botón compartir. 6. Al usuario se le presenta un formulario. 7. El usuario ingresa el correo electrónico correspondiente a quien enviar. 8. El usuario da clic botón enviar.
Flujo alternativo	<ol style="list-style-type: none"> 1. Si el usuario no está autenticado en el sistema se le mostrara la pantalla de inicio de sesión. 2. Si el usuario no dispone de un análisis de plagio, se le presenta una lista vacía.
Requisito	RF10

Tabla A4 20. INTERFAZ COMPARTIR RESULTADO DE PLAGIO

Interfaz compartir resultado de plagio		
Nro. Interfaz	1	
Estado de pruebas	Aceptada sin cambios	si
	Aceptada con cambios	
	Rediseñar	
Requisito funcional	RF10	
Observaciones		

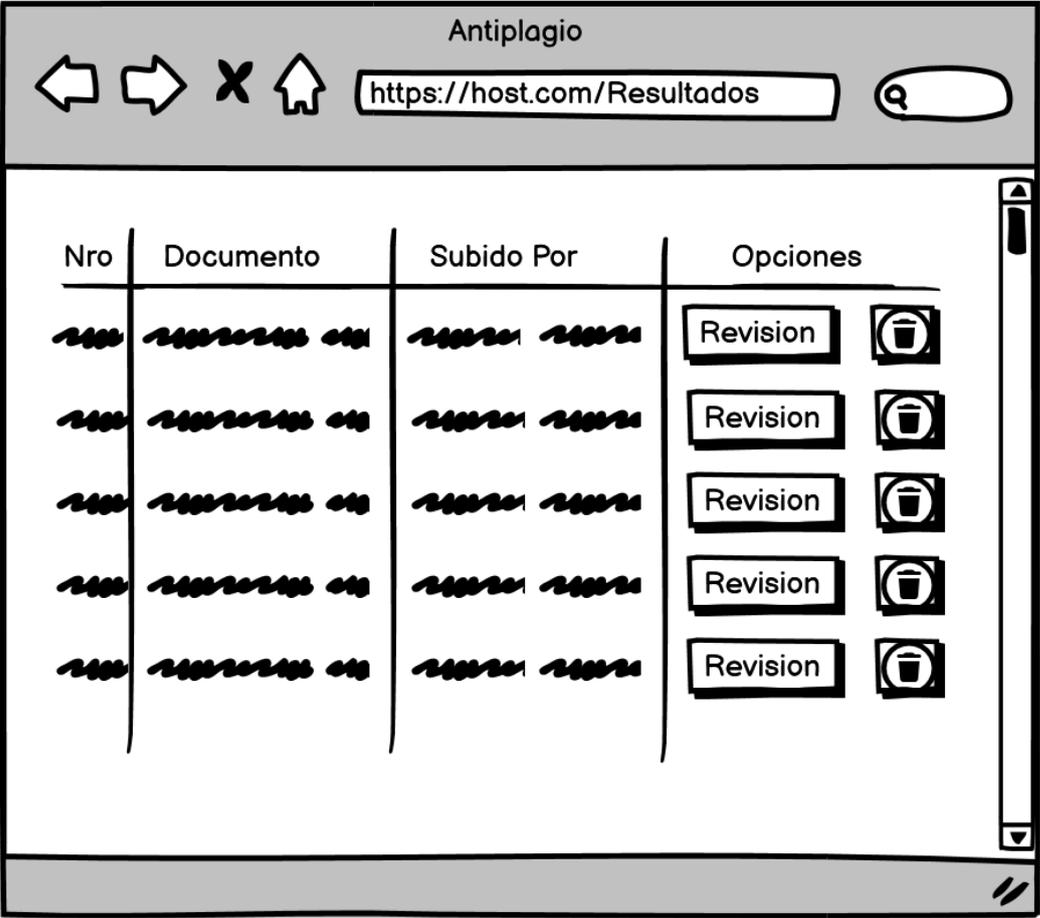


Requisito Funcional Ver lista de Documentos analizados por el usuario

Tabla A4 21. CASO DE USO VER LISTA DE DOCUMENTOS ANALIZADOS

Identificador	CU11
Nombre	Ver listado de documentos analizados por el usuario.
Precondición	El usuario debe estar autenticado en el sistema.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario ingresa a la sección Resultados. 2. Al usuario se le presenta una lista con los resultados.
Flujo alternativo	<ol style="list-style-type: none"> 1. Si el usuario no está autenticado en el sistema se le mostrara la pantalla de inicio de sesión. 2. Si el usuario no dispone de un análisis de plagio, se le presenta una lista vacía.
Requisito	RF11

Tabla A4 22. INTERFAZ VER LISTA DE DOCUMENTOS ANALIZADOS

Interfaz ver lista de documentos analizados		
Nro. Interfaz	1	
Estado de pruebas	Aceptada sin cambios	si
	Aceptada con cambios	
	Rediseñar	
Requisito funcional	RF11	
Observaciones		
		

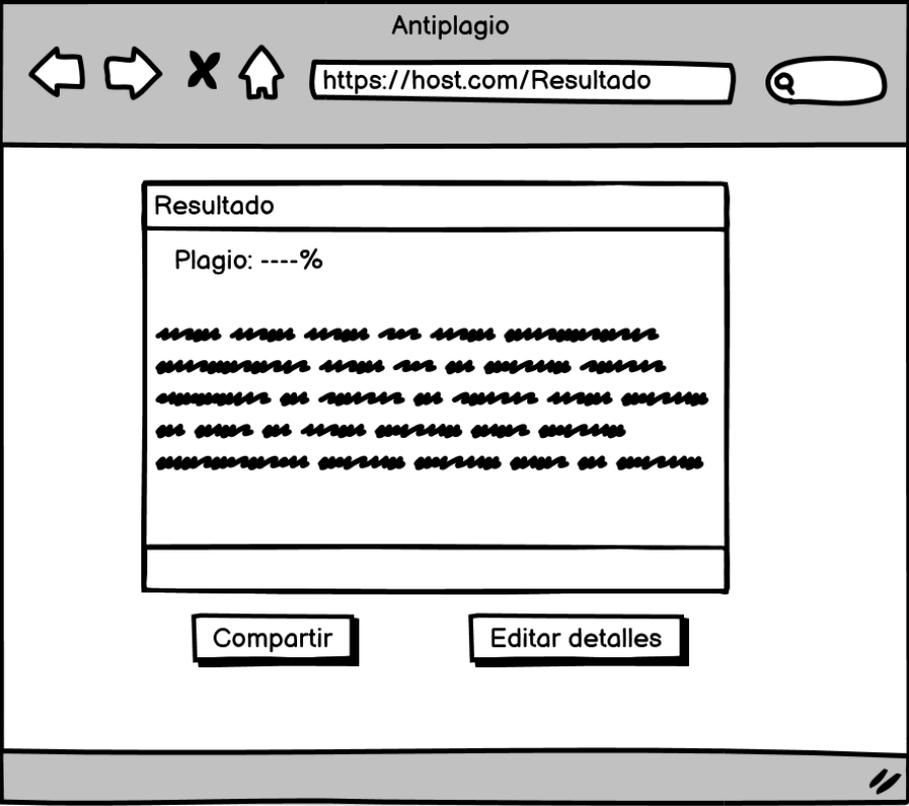
Requisito Funcional Ver análisis detallado de documentos plagiados por el usuario.

Tabla A4 23. CASO DE USO VER RESULTADO DETALLADO DE ANÁLISIS DE PLAGIO

Identificador	CU12
Nombre	Ver análisis detallado de los documentos plagiados por el usuario.
Precondición	El usuario debe estar autenticado en el sistema. El usuario debe poseer un análisis de plagio realizado anteriormente.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario ingresa a la sección Resultados. 2. Al usuario se le presenta una lista con los resultados. 3. El usuario da clic en la sección revisión ubicado en la columna acciones. 4. Al usuario se le presenta en pantalla el resultado de plagio detallado.

Flujo alternativo	<ol style="list-style-type: none"> 1. Si el usuario no está autenticado en el sistema se le mostrara la pantalla de inicio de sesión. 2. Si el usuario no dispone de un análisis de plagio previo, se le presenta una lista vacía.
Requisito	RF12

Tabla A4 24. INTERFAZ VER RESULTADO DETALLADO DE ANÁLISIS DE PLAGIO

Interfaz ver resultado detallado de análisis de plagio		
Nro. Interfaz	1	
Estado de pruebas	Aceptada sin cambios	si
	Aceptada con cambios	
	Rediseñar	
Requisito funcional	RF12	
Observaciones		

Requisito Funcional Editar detalle de los documentos plagiados

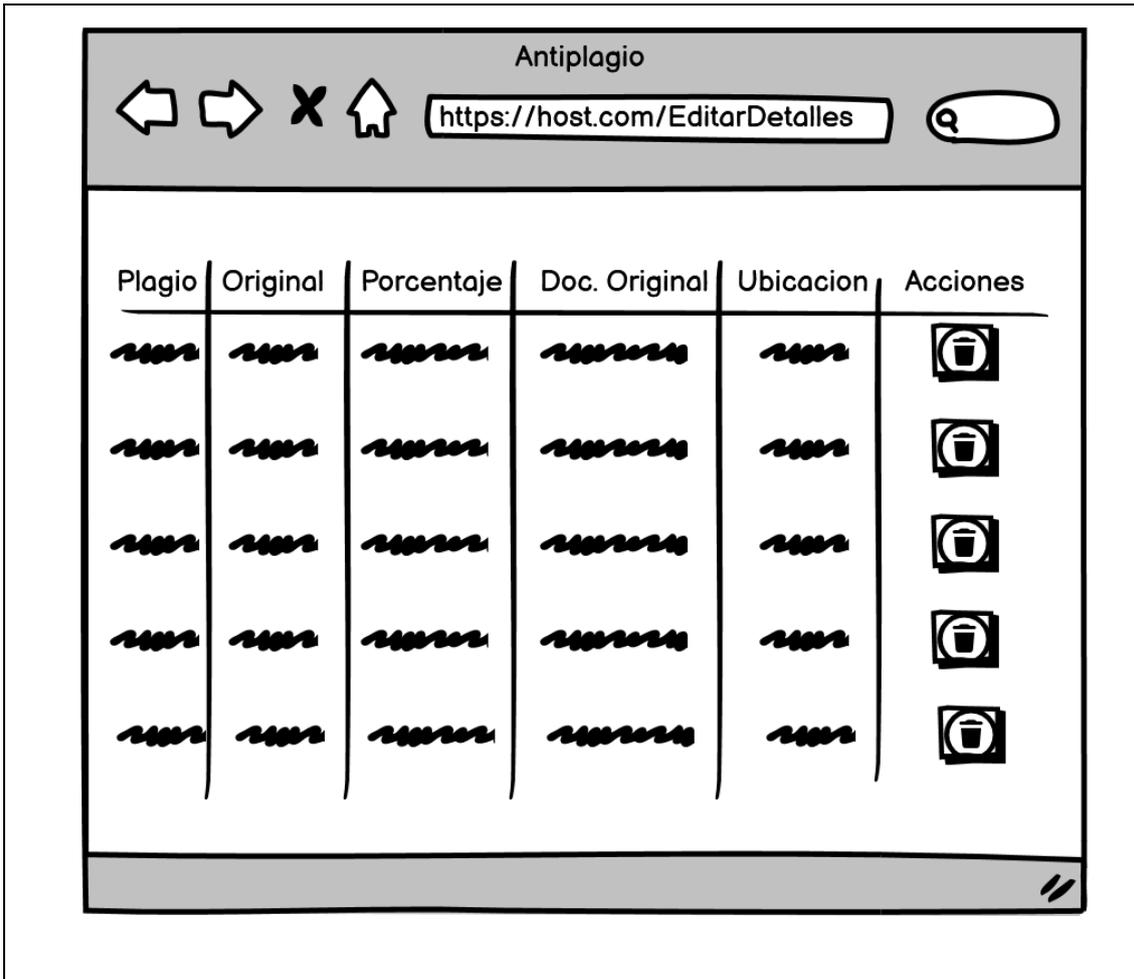
Tabla A4 25. CASO DE USO EDITAR RESULTADO DETALLADO DE PLAGIO

Identificador	CU13
Nombre	Editar detalle de los documentos plagiados
Precondición	<p>El usuario debe estar autenticado en el sistema. El usuario debe poseer un análisis de plagio realizado anteriormente. Solo el usuario docente dispondrá de la opción editar detalle.</p>
Flujo principal	<ol style="list-style-type: none"> 1. El usuario ingresa a la sección Resultados. 2. Al usuario se le presenta una lista con los resultados.

	<ol style="list-style-type: none"> 3. El usuario da clic en la sección revisión ubicado en la columna acciones. 4. Al usuario se le presenta en pantalla el resultado de plagio detallado. 5. El usuario da clic en el botón editar. 6. Al usuario se le presentara un botón en cada párrafo plagiado. 7. El usuario da clic en el párrafo que desea eliminar para reducir el plagio, este párrafo se eliminara automáticamente.
Flujo alterno	<ol style="list-style-type: none"> 1. Si el usuario no está autenticado en el sistema se le mostrara la pantalla de inicio de sesión. 2. Si el usuario no dispone de un análisis de plagio previo, se le presenta una lista vacía.
Requisito	RF13

Tabla A4 26. INTERFAZ EDITAR RESULTADO DETALLADO DE PLAGIO

Interfaz editar resultado detallado de plagio		
Nro. Interfaz	1	
Estado de pruebas	Aceptada sin cambios	si
	Aceptada con cambios	
	Rediseñar	
Requisito funcional	RF13	
Observaciones		

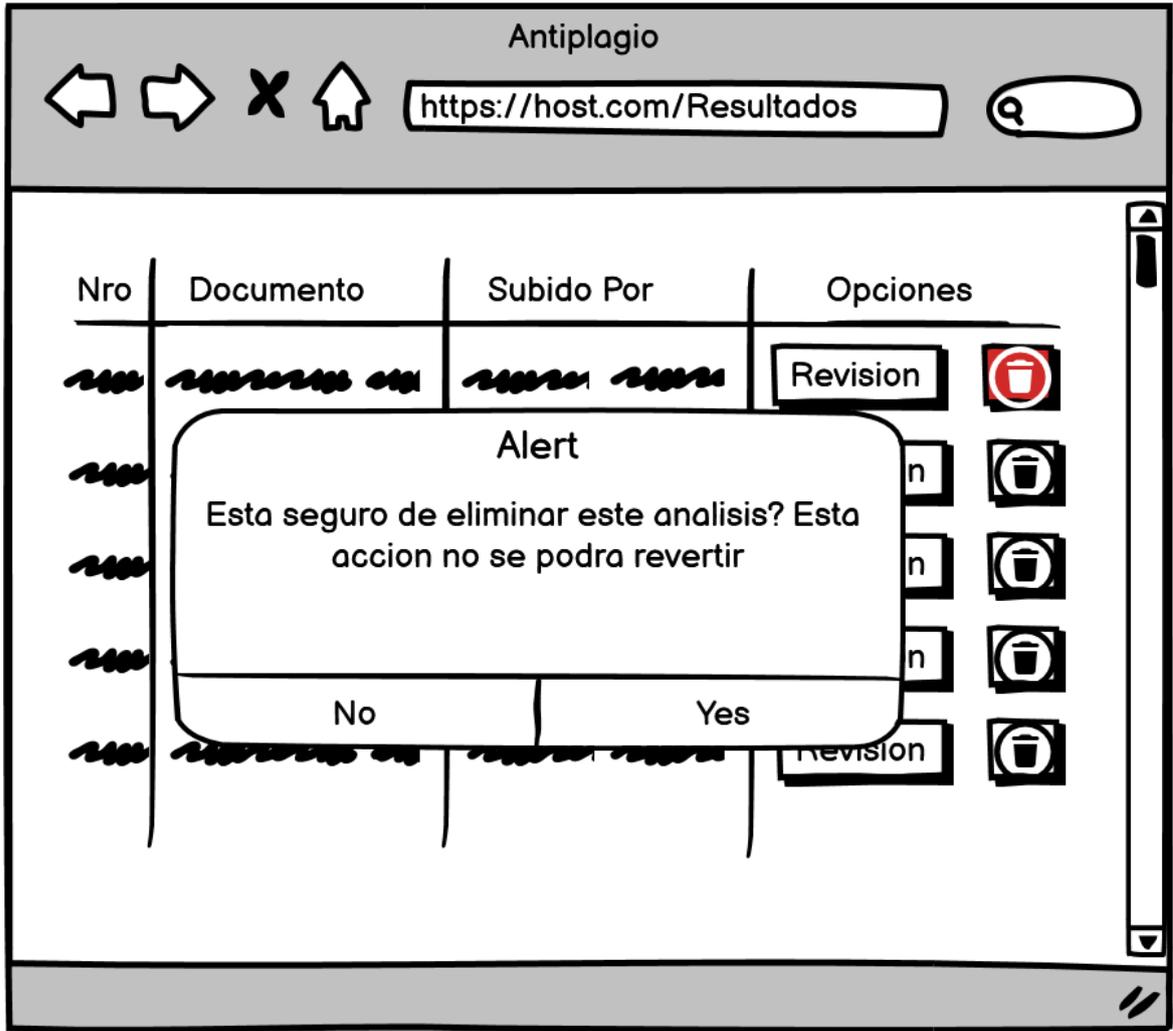


Requisito Funcional Eliminar resultado de los documentos plagiados

Tabla A4 27. CASO DE USO ELIMINAR RESULTADO DE LOS DOCUMENTOS PLAGIADOS

Identificador	CU14
Nombre	Eliminar detalle de los documentos plagiados
Precondición	El usuario debe estar autenticado en el sistema. El usuario debe poseer un análisis de plagio realizado anteriormente. Solo el usuario docente dispondrá de la opción editar detalle.
Flujo principal	8. El usuario ingresa a la sección Resultados. 9. Al usuario se le presenta una lista con los resultados. 10. El usuario da clic en la sección eliminar ubicado en la columna acciones. 11. Al usuario se le presentara un mensaje de confirmación. 12. El usuario selecciona el botón eliminar.
Flujo alternativo	1. Si el usuario no está autenticado en el sistema se le mostrara la pantalla de inicio de sesión. 2. Si el usuario no dispone de un análisis de plagio previo, se le presenta una lista vacía.
Requisito	RF14

Tabla A4 28. INTERFAZ ELIMINAR RESULTADO DE LOS DOCUMENTOS PLAGIADOS

Interfaz eliminar resultado de los documentos plagiados		
Nro. Interfaz	1	
Estado de pruebas	Aceptada sin cambios	si
	Aceptada con cambios	
	Rediseñar	
Requisito funcional	RF14	
Observaciones	 <p>The screenshot shows a web browser window titled 'Antiplagio' with the URL 'https://host.com/Resultados'. Below the browser is a table with columns: 'Nro', 'Documento', 'Subido Por', and 'Opciones'. The table contains five rows of data, each with a trash can icon in the 'Opciones' column. An 'Alert' dialog box is overlaid on the table, containing the text: 'Esta seguro de eliminar este analisis? Esta accion no se podra revertir'. The dialog has 'No' and 'Yes' buttons at the bottom.</p>	

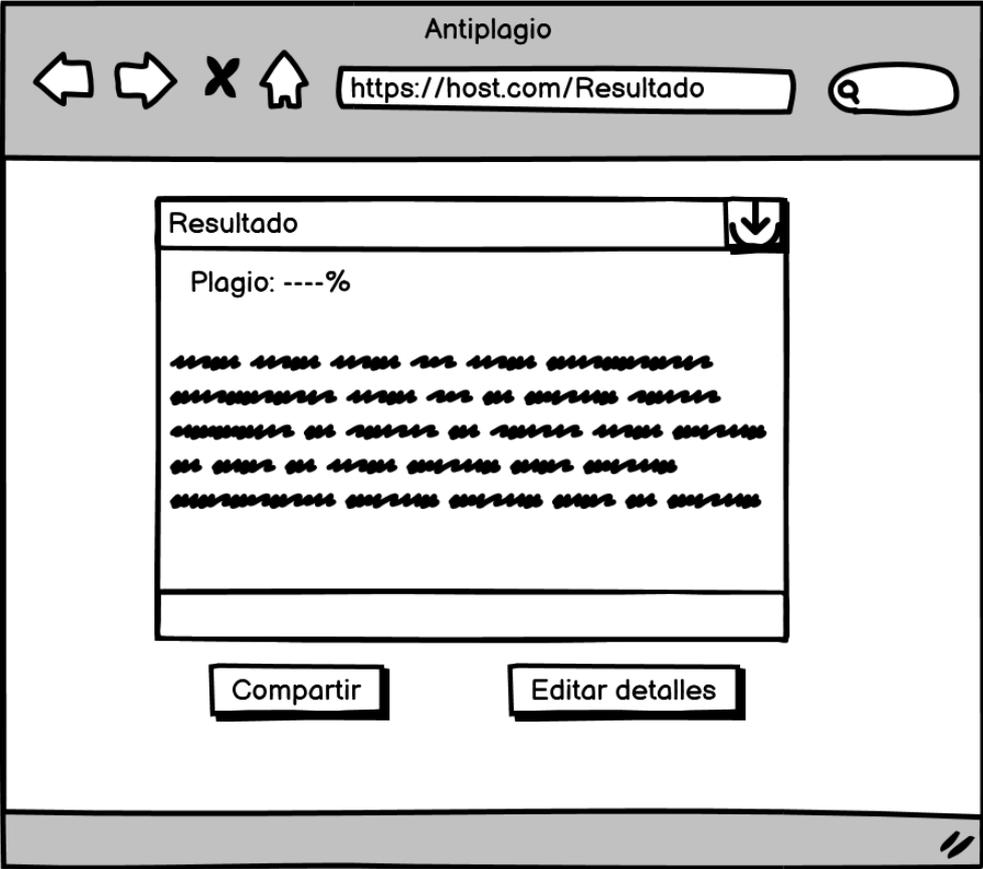
Requisito Funcional Descargar Resultado de plagio.

Tabla A4 29. CASO DE USO DESCARGAR RESULTADO DE PLAGIO

Identificador	CU15
Nombre	Descargar resultados de plagio
Precondición	El usuario debe estar autenticado en el sistema. El usuario debe poseer un análisis de plagio realizado anteriormente. Solo el usuario docente dispondrá de la opción editar detalle.

Flujo principal	<ol style="list-style-type: none"> 1. El usuario ingresa a la sección Resultados. 2. Al usuario se le presenta una lista con los resultados. 3. El usuario da clic en la sección descargar ubicado en la columna acciones. 4. El resultado se descarga automáticamente.
Flujo alternativo	<ol style="list-style-type: none"> 1. Si el usuario no está autenticado en el sistema se le mostrara la pantalla de inicio de sesión. 2. Si el usuario no dispone de un análisis de plagio previo, se le presenta una lista vacía.
Requisito	RF15

Tabla A4 30. INTERFAZ DESCARGAR RESULTADO DE PLAGIO

Interfaz descargar resultado de plagio		
Nro. Interfaz	1	
Estado de pruebas	Aceptada sin cambios	si
	Aceptada con cambios	
	Rediseñar	
Requisito funcional	RF15	
Observaciones		
 <p>The screenshot shows a web browser window titled 'Antiplagio'. The address bar contains 'https://host.com/Resultado'. The main content area has a header 'Resultado' with a download icon. Below it, the text 'Plagio: ----%' is displayed, followed by several lines of illegible, garbled text. At the bottom of the content area, there are two buttons: 'Compartir' and 'Editar detalles'.</p>		

Anexo 6. Desarrollo del Objetivo 2.

Desarrollo del Objetivo 2: Desarrollo del software anti plagio utilizando Django y aplicando del modelo de software 4+1.

Trabajo: Prototipo de Software anti plagio para la carrera de Ingeniería en Computación de la Universidad Nacional de Loja, utilizando redes neuronales.

Desarrollo del software anti plagio utilizando Django y aplicando del modelo de software 4+1.

En esta sección se presenta las actividades que se llevaron a cabo para cumplir con el segundo objetivo del Trabajo de Titulación, proporcionando un desglose de las fases de la metodología XP y aplicando el modelo 4+1, para el diseño del software anti plagio.

1. Fase Planificación

Para esta fase se realizó una entrevista con el cliente principal del software y con ello poder generar las Historias de Usuario, para de esta manera definir las actividades necesarias para cada iteración (véase **Tabla A5 1**).

Tabla A5 1. Proceso de planificación para cada iteración de la metodología XP en función de las historias de usuario.

Iteración	Historia de Usuario
1	Registrar usuarios.
	Iniciar sesión.
	Autorizar registro.
2	Validar registro de usuarios.
3	Analizar plagio por el docente.
	Analizar plagio por el Estudiante.
4	Compartir resultados de análisis de plagio.
	Ver listado de documentos analizados por el usuario.
5	Ver resultado de análisis de plagio
	editar resultado de análisis de plagio.
	Descargar resultado de análisis de plagio.
6	Ver listado de usuarios
	Cambiar el estado de cuentas de usuario.

	Editar usuarios.
	Eliminar usuarios.

Para llevar a cabo la planificación, se empleó la herramienta desarrollada específicamente por la Universidad Nacional de Loja. En esta herramienta, se incorporaron todas las iteraciones, asignando un tiempo mínimo para la ejecución de cada una de ellas. Este período podría ser ajustado según el progreso realizado en el Trabajo. Después de concluir cada iteración, se llevaba a cabo una reunión con el cliente para validar la implementación de cada historia de usuario y, en caso necesario efectuar correcciones.

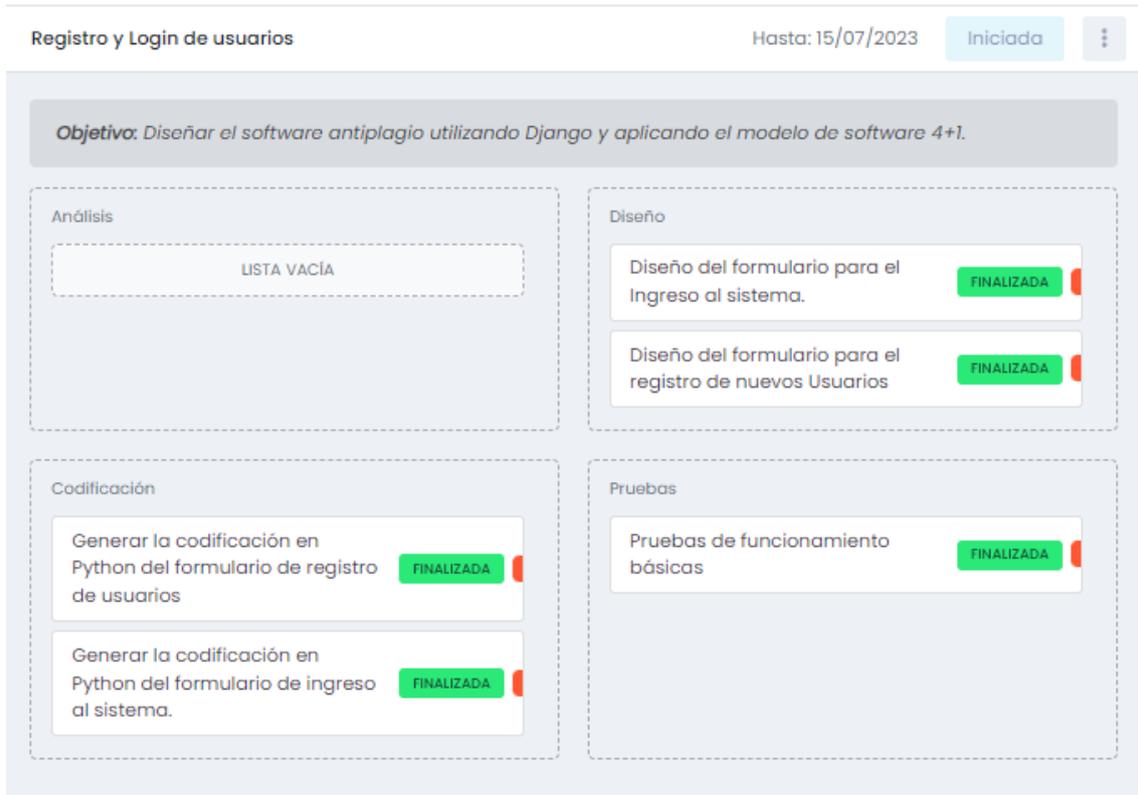


Figura A5 1. Tareas de la Iteración 1

Validar y Rechazar registro Hasta: 17/07/2023 Iniciada ⋮

Objetivo: Diseñar el software antiplagio utilizando Django y aplicando el modelo de software 4+1.

<p>Análisis</p> <p>Iniciar un análisis de como se debe validar los registros para nuevos usuarios en una plataforma web FINALIZADA</p>	<p>Diseño</p> <p>Diseño del formulario para el validar registro de Usuarios FINALIZADA</p>
<p>Codificación</p> <p>Generar la codificación en Python de la validación de usuarios FINALIZADA</p>	<p>Pruebas</p> <p>Pruebas de funcionamiento básicas FINALIZADA</p>

Figura A5 2. Tareas de la Iteración 2

Analizar plagio Hasta: 18/07/2023 Iniciada ⋮

Objetivo: Diseñar el software antiplagio utilizando Django y aplicando el modelo de software 4+1.

<p>Análisis</p> <p>Análisis de como incrustar el código para el análisis de plagio en el proyecto Django FINALIZADA</p>	<p>Diseño</p> <p>Diseñar como deberá ser el flujo para realizar un análisis de plagio FINALIZADA</p>
<p>Codificación</p> <p>Codificar la inserción del código para el análisis de plagio en Django FINALIZADA</p>	<p>Pruebas</p> <p>Generar pruebas de funcionamiento basico FINALIZADA</p>

Figura A5 3. Tareas de la Iteración 3

Compartir y ver listado de documentos analizados Hasta: 19/07/2023 Iniciada ⋮

Objetivo: Diseñar el software antiplagio utilizando Django y aplicando el modelo de software 4+1.

<p>Análisis</p> <p>Realizar un análisis de como se debe mostrar los resultados a los usuarios FINALIZADA</p>	<p>Diseño</p> <p>Diseñar un modelo que permita a los usuarios ver en una lista todos los análisis de plagio que hayan realizado FINALIZADA</p>
<p>Codificación</p> <p>Codificar la inserción del código para el análisis de plagio en Django FINALIZADA</p>	<p>Pruebas</p> <p>Pruebas de funcionamiento basicas FINALIZADA</p>

Figura A5 4. Tareas de la Iteración 4

Ver análisis detallado, editar análisis y descargar resultados Hasta: 20/07/2023 Iniciada ⋮

Objetivo: Diseñar el software antiplagio utilizando Django y aplicando el modelo de software 4+1.

<p>Análisis</p> <p>Analizar como se debe ver un documento detallado de análisis de plagio FINALIZADA</p>	<p>Diseño</p> <p>Diseñar como debe verse un análisis detallado en Django FINALIZADA</p>
<p>Codificación</p> <p>Codificar la generación del resultado con detalles FINALIZADA</p>	<p>Pruebas</p> <p>Pruebas de funcionamiento basicas FINALIZADA</p>

Figura A5 5. Tareas de la Iteración 5



Figura A5 6. Tareas de la Iteración 6

2. Fase de Diseño

Se procedió a crear y diseñar la arquitectura de software para la detección de plagio en esta sección. Se tuvieron en cuenta las tecnologías seleccionadas para su desarrollo, las cuales fueron adaptadas a medida que avanzaba la implementación del Trabajo.

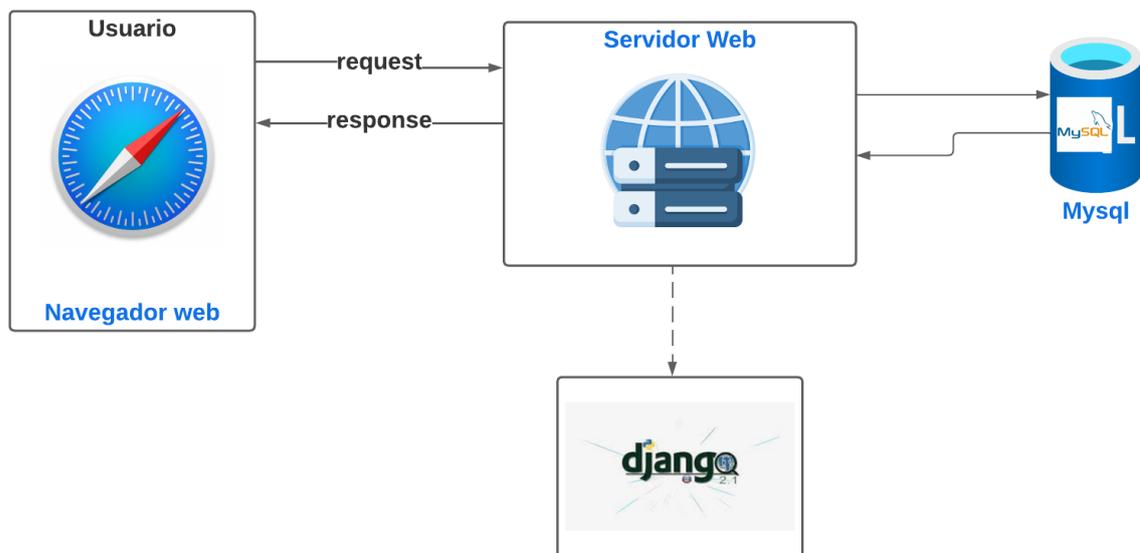


Figura A5 7. Arquitectura del Software

La Arquitectura de Software para el modelo 4 + 1 se presenta en el **Anexo 14**

3. Fase de Codificación

Para la codificación se utilizó la herramienta Visual Studio Code en conjunto con la base de datos MySQL y el framework de desarrollo web Django. Además de ello se definieron algunas normas de codificación para su desarrollo.

Normas de codificación.

Como medida recomendable, se decidió emplear normas en el desarrollo del software con el objetivo de poseer una buena comprensión del código y tener una implementación adecuada a lo largo de todo el proceso de desarrollo.

Models

- Los nombres de los modelos para la Base de Datos de MySQL se escribieron con la primera letra en mayúscula para cada una de las palabras y el resto en minúsculas.
- Los nombres de los atributos de los modelos se escribieron en minúsculas.
- Para la mayoría de los nombres tanto de modelos y atributos se escribieron en el idioma
- español para tener mejor control de los datos.

Views

- Para el nombre de los métodos de las clases se escribieron la primera palabra en minúscula y el resto todo en mayúsculas.
- Para los atributos de los métodos se escriben con minúsculas toda la palabra y si existía más de una palabra este fue separada por un guion bajo.

URLS

- Para el nombre de las rutas de las urls se escribieron todo en minúsculas, los mismos que si existían más de una palabra están fueron separadas por un guion bajo.
- Se agruparon por secciones asociando los CRUD para los distintos modelos.
- Para los nombres de las rutas se escriben con minúsculas toda la palabra y si existía
- más de una palabra fue separada por un guion bajo.

Forms

- Los nombres para las clases se escribieron con la primera letra en mayúscula para cada una de las palabras todas comenzando con la palabra Formulario.
- Para el nombre de los métodos se escribieron todo en minúsculas, los mismos que si existían más de una palabra las cuales se iniciaban con la primera letra en mayúscula.

- Para los atributos de los métodos se escriben con minúsculas toda la palabra y si existía más de una palabra esta se iniciaban con la primera letra en mayúscula.

3.1. Iteración 1

Para la primera iteración la historia de usuario se centra en el registro de usuarios y el ingreso al sistema.

ACTIVIDADES	RESPONSABLES	FECHA INICIO	FECHA FIN	ESTADO	ATRASO
Diseño del formulario para el registro de nuevos Usuarios	Jaime Oswaldo Paqui Medina Santiago Roman	14/07/2023	21/07/2023	Finalizada	Ninguno
Generar la codificación en Python del formulario de registro de usuarios	Santiago Roman	14/07/2023	21/07/2023	Finalizada	Ninguno
Diseño del formulario para el Ingreso al sistema.	Jaime Oswaldo Paqui Medina Santiago Roman	06/07/2023	14/07/2023	Finalizada	2 días

RESUMEN GENERAL	
Actividades Resueltas	5 actividades
Actividades Sin Resolver	0 actividades
Actividades con retraso	3 actividades
Actividades Descartadas	0 actividad

ACTIVIDADES DESCARTADAS
No hay actividades descartadas

Figura A5 24. Resultados del Sprint 1.

```

class RegistrarWizardView(SessionWizardView):
    form_list = [
        ("registro", FormularioRegistrar),
        ("rol", FormularioRol),
    ]

    def get_template_names(self):
        return [TEMPLATES[self.steps.current]]

    def done(self, form_list, **kwargs):
        formulario_registro_data = self.get_cleaned_data_for_step("registro")
        formulario_rol_data = self.get_cleaned_data_for_step("rol")
        # Procesar formulario de registro
        usuario = formulario_registro_data['username']
        clave = formulario_registro_data['password']
        correo = formulario_registro_data['email']
        nombres = formulario_registro_data['first_name']
        apellidos = formulario_registro_data['last_name']

        user = User.objects.create_user(usuario, correo, clave)
        user.last_name = apellidos
        user.first_name = nombres
        user.save()

        usuario_model = Usuario()
        usuario_model.apellidos = apellidos
        usuario_model.nombres = nombres
        usuario_model.correo = correo
        usuario_model.estado = False
        usuario_model.save()

        # Procesar formulario de rol
        rol = formulario_rol_data['rol']
        if rol == 'estudiante': ...
        elif rol == 'docente': ...
        else:
            estudiante_model = Estudiante()
            estudiante_model.usuario = usuario_model
            estudiante_model.save()

        return HttpResponseRedirect(reverse('completado'))

```

Figura A5 25. Método para registrar usuarios.

```

def autenticar(request):
    if request.method == 'POST':
        formulario = FormularioLogin(request.POST)
        if formulario.is_valid():
            usuario = request.POST['username']
            clave = request.POST['password']
            user = authenticate(username = usuario, password=clave)

            if user is not None:
                if user.is_active:
                    usuario = Usuario.objects.get(correo=user.email)
                    if usuario.estado:
                        login(request, user)
                        next_url = request.GET.get('next') # Obtener la URL a la que se intentó acceder
                        if next_url:
                            return HttpResponseRedirect(next_url)
                        else:
                            return HttpResponseRedirect(reverse('homepage'))
                    else:
                        return HttpResponseRedirect(reverse('no_activo'))
                else:
                    return HttpResponseRedirect(reverse('no_activo'))
            else:
                messages.add_message(request, messages.INFO, 'Usuario o Clave incorrectas')
            # else:
            #     messages.add_message(request, messages.INFO, 'Demasiados intentos fallidos. Cuenta bloqueada')

        else:
            formulario = FormularioLogin()
        context = {
            'formulario': formulario
        }
    return render (request, 'login/login.html', context)

```

Figura A5 26. Método para autenticar al usuario.

3.2. Iteración 2

Para la segunda iteración la historia de usuario se centra en validar registro de usuario, una vez que ingresa los datos en el formulario.

ACTIVIDADES	RESPONSABLES	FECHA INICIO	FECHA FIN	ESTADO	ATRASO
Diseño del formulario para el validar registro de Usuarios	Jaime Oswaldo Paqui Medina Santiago Roman	14/07/2023	21/07/2023	Finalizada	Ninguno
Generar la codificación en Python de la validación de usuarios	Santiago Roman	14/07/2023	21/07/2023	Finalizada	Ninguno
Pruebas de funcionamiento básicas	Jaime Oswaldo Paqui Medina	14/07/2023	15/07/2023	Finalizada	1 días

RESUMEN GENERAL	
Actividades Resueltas	4 actividades
Actividades Sin Resolver	0 actividades
Actividades con retraso	1 actividad
Actividades Descartadas	0 actividad

ACTIVIDADES DESCARTADAS
No hay actividades descartadas

Figura A5 27. Resultados del Sprint 2.

```

class FormularioAutorizacion(forms.Form):
    opciones = [('autorizar', 'Autorizar'), ('rechazar', 'Rechazar')]
    seleccion = forms.ChoiceField(widget=forms.RadioSelect, choices=opciones, required=True)

```

Figura A5 28. Formulario para validar el registro de usuarios.

3.3. Iteración 3

Para la tercera iteración la historia de usuario se centra analizar plagio por el estudiante y docente, donde el usuario deberá subir un documento para su posterior análisis.

ACTIVIDADES	RESPONSABLES	FECHA INICIO	FECHA FIN	ESTADO	ATRASO
Diseñar como deberá ser el flujo para realizar un análisis de plagio	Santiago Roman	14/07/2023	21/07/2023	Finalizada	Ninguno
Generar pruebas de funcionamiento basico	Jaime Oswaldo Paqui Medina	14/07/2023	21/07/2023	Finalizada	Ninguno
Codificar la inserción del código para el análisis de plagio en Django	Santiago Roman	14/07/2023	21/07/2023	Finalizada	Ninguno

RESUMEN GENERAL	
Actividades Resueltas	4 actividades
Actividades Sin Resolver	0 actividades
Actividades con retraso	0 actividades
Actividades Descartadas	0 actividad

ACTIVIDADES DESCARTADAS
No hay actividades descartadas

Figura A5 29. Resultados del Sprint 3.

```

@login_required
def gestionEstudiante (request,gestion_id):
    user = request.user
    usuario_global = Usuario.objects.get(correo=user.email)
    if usuario_global.estado:
        formulario_gestion = FormularioGestion(request.POST)
        if request.method == 'POST':
            if formulario_gestion.is_valid():
                datos_gestion = formulario_gestion.cleaned_data
                #verificar que el email del docente exista
                email_docente = datos_gestion.get('email')
                if Usuario.objects.filter(correo=email_docente).exists():
                    #verificar si el usuario esta registrado como docente
                    print(" ##### ")
                    print(" verificar si el correo esta asociado a un docente ")
                    print(" ##### ")
                    usuario = Usuario.objects.get(correo=email_docente)
                    if Docente.objects.filter(usuario = usuario).exists() or user.groups.filter(name = "docente").exists():
                        print("aqui")
                        listaGestion = GestionDocumentos.objects.get (gestion_id = gestion_id)
                        documento = listaGestion.documento

                        listaGestion.titulo = datos_gestion.get('titulo')
                        listaGestion.comentario = datos_gestion.get('comentario')
                        listaGestion.docente = Docente.objects.get(usuario = usuario)
                        listaGestion.save()
                        documento.visible = True
                        documento.save()
                        #detectar' gestion.gestion_id
                        #return render(request, 'gestion/exito.html',locals())
                        return HttpResponseRedirect(reverse('detectar', args=[listaGestion.gestion_id]))
                    else:
                        return HttpResponseRedirect(reverse('homepage'))
                else:
                    return HttpResponseRedirect(reverse('homepage'))

            return render(request, 'gestion/gestionEstudiante.html',locals())

    return render(request, 'gestion/gestionEstudiante.html',locals())
else:
    return render(request, 'homepage.html')

```

Figura A5 30. Método para análisis de plagio por el estudiante.

```

@login_required
def detectar(request, gestion_id):
    user = request.user
    usuario = Usuario.objects.get(correo=user.email)
    if usuario.estado:

        gestion = GestionDocumentos.objects.get(gestion_id = gestion_id)
        # documento = Documento.objects.get(documento_id = gestion.documento_id)
        documento = gestion.documento #ojo
        directorio_archivo = documento.archivo.path

        resultado = Resultado()
        resultado.management = gestion
        resultado.ejecutando = True
        resultado.save()

        # documento_generado, nombre= main.main()
        documento_generado, nombre, plagio, informacion = main.main(directorio_archivo)
        with open(documento_generado, 'rb') as archivo:
            archivo_django = ContentFile(archivo.read(), name=nombre)

        resultado.ejecutando = False
        resultado.archivo.save(nombre, archivo_django)
        resultado.estado = True
        resultado.plagio = plagio
        resultado.informacion = informacion
        resultado.save()

        return HttpResponseRedirect(reverse('revision', args=[resultado.resultado_id]))

    return render(request, 'homepage.html')

```

Figura A5 31. Método para análisis de plagio por el docente.

3.4. Iteración 4

Para la cuarto iteración la historia de usuario se centra en ver todos los documentos analizados por el usuario y compartir resultado de análisis de plagio, una vez terminado el análisis.

ACTIVIDADES	RESPONSABLES	FECHA INICIO	FECHA FIN	ESTADO	ATRASO
Diseñar un modelo que permita a los usuarios ver en una lista todos los análisis de plagio que hayan realizado	Jaime Oswaldo Paqui Medina Santiago Roman	14/07/2023	21/07/2023	Finalizada	Ninguno
Codificar la inserción del código para el análisis de plagio en Django	Santiago Roman	14/07/2023	21/07/2023	Finalizada	Ninguno
Pruebas de funcionamiento basicas	Jaime Oswaldo Paqui Medina	14/07/2023	15/07/2023	Finalizada	4 días

RESUMEN GENERAL	
Actividades Resueltas	4 actividades
Actividades Sin Resolver	0 actividades
Actividades con retraso	1 actividad
Actividades Descartadas	0 actividad

ACTIVIDADES DESCARTADAS
No hay actividades descartadas

Figura A5 32. Resultados del Sprint 4.

```

@login_required
def index(request):
    user = request.user
    usuario = Usuario.objects.get(correo=user.email)
    if usuario.estado:
        if user.groups.filter(name = "docente").exists() or user.groups.filter(name = "admin").exists():
            cond = False
            docente = Docente.objects.get(usuario = usuario)
            listaGestion = GestionDocumentos.objects.filter(docente = docente)
            listaResultado = []
            for gestion in listaGestion:
                try:
                    print(gestion.gestion_id)
                    resultado = Resultado.objects.get(management=gestion)

                    listaResultado.append(resultado)
                    if gestion.estudiante:
                        cond = True
                except Resultado.DoesNotExist:
                    print("el resultado esta procesandose, o no existe en la base de datos")
            elif user.groups.filter(name = "estudiante").exists():
                estudiante = Estudiante.objects.get(usuario = usuario)
                listaGestion = GestionDocumentos.objects.filter(estudiante = estudiante)
                listaResultado = []
                for gestion in listaGestion:
                    try:
                        print(gestion.gestion_id)
                        if gestion.docente is not None:
                            resultado = Resultado.objects.get(management=gestion)
                            listaResultado.append(resultado)
                            cond = True
                        else:
                            print('No tiene asignado un profesor para realizar el plagio.')
                    except Resultado.DoesNotExist:
                        print("el resultado esta procesandose, o no existe en la base de datos")
                pass
            # busqueda = request.POST.get("busqueda")
            return render(request, 'plagio/index.html', locals())
    return render(request, 'homepage.html')

```

Figura A5 33. Método para ver la lista de Documentos analizados por el usuario.

```

@login_required
def compartirDocumento(request, resultado_id):
    user = request.user
    usuario = Usuario.objects.get(correo=user.email)
    nombres = usuario.nombres + " " + usuario.apellidos
    resultado = Resultado.objects.get(resultado_id = resultado_id)
    if request.method == 'POST':
        # Obtener los correos electrónicos ingresados en el formulario
        correos = request.POST.getlist('correo')

        # Obtener el comentario ingresado en el formulario
        comentario = request.POST.get('comentario', '')

        archivo_adjunto = resultado.archivo

        # Por ejemplo, enviar correos electrónicos a las direcciones ingresadas
        emailCompartir(correos, comentario, nombres, archivo_adjunto)
        return redirect(index)
    return render(request, 'plagio/compartir.html', locals())

```

Figura A5 34. Método para compartir resultado de análisis de plagio.

3.5. Iteración 5

Para la quinta iteración la historia de usuario se centra en ver análisis detallado de documentos plagiados por el usuario, editar detalle de los documentos plagiados y eliminar resultado de los documentos plagiados, una vez terminado el análisis.

ACTIVIDADES	RESPONSABLES	FECHA INICIO	FECHA FIN	ESTADO	ATRASO
Diseñar como debe verse un análisis detallado en Django	Jaime Oswaldo Paqui Medina	14/07/2023	21/07/2023	Finalizada	Ninguno
Codificar la generación del resultado con detalles	Santiago Roman	14/07/2023	21/07/2023	Finalizada	Ninguno
Pruebas de funcionamiento basicas	Jaime Oswaldo Paqui Medina	14/07/2023	15/07/2023	Finalizada	4 días

RESUMEN GENERAL	
Actividades Resueltas	4 actividades
Actividades Sin Resolver	0 actividades
Actividades con retraso	1 actividad
Actividades Descartadas	0 actividad

ACTIVIDADES DESCARTADAS
No hay actividades descartadas

Figura A5 35. Resultados del Sprint 5.

```
@login_required
def revision(request,resultado_id):
    #cambio agregar logicca de documento publico
    user = request.user
    usuario = Usuario.objects.get(correo=user.email)
    if usuario.estado:
        resultado = Resultado.objects.get(resultado_id = resultado_id)
        gestion = GestionDocumentos.objects.get(resultado = resultado)
        return render(request,'plagio/Success.html',locals())
    return
```

Figura A5 36. Método para ver análisis detallado de documentos plagiados por el usuario.

```

@login_required
def detalleDocumento(request, resultado_id):
    user = request.user
    resultado = Resultado.objects.get(resultado_id = resultado_id)
    plagio = resultado.plagio
    if user.groups.filter(name = "docente").exists() or user.groups.filter(name = "admin").exists():
        listaOracion=[]
        listaPlagio=[]
        listaPorcentaje=[]
        listaUrl=[]
        listaUbicacion=[]
        for oracion, plagio, porcentaje, url, ubicacion in plagio:
            listaOracion.append(oracion)
            listaPlagio.append(plagio)
            listaPorcentaje.append(porcentaje)
            listaUrl.append(url)
            listaUbicacion.append(ubicacion)

        matriz = []
        print("plagio",len(listaOracion))
        for i in range(len(listaOracion)):
            fila = [listaOracion[i], listaPlagio[i], listaPorcentaje[i], listaUrl[i],listaUbicacion[i]]
            matriz.append(fila)
        num_oraciones = resultado.informacion['total_oraciones']
        context = {
            'matriz': matriz,
            'resultado': resultado,
            'num_oraciones' : num_oraciones,
        }

        return render(request, 'plagio/matrizDetalle.html', context)
    else:
        return render(request, 'login/forbidden.html', locals())

```

Figura A5 37. Método para editar detalle de los documentos plagiados.

```

@login_required
def eliminarResultado(request, resultado_id):
    user = request.user
    if user.groups.filter(name = "docente").exists() or user.groups.filter(name = "admin").exists():
        try:
            resultado = Resultado.objects.get(resultado_id = resultado_id)
            gestion = resultado.management
            resultado.delete()
            gestion.delete()
            return redirect(index)
        except Exception as e:
            print("error")
            return redirect(index)

    else:
        return render(request, 'login/forbidden.html', locals())

```

Figura A5 38. Método para eliminar resultado de los documentos plagiados

3.6. Iteración 6

Para la sexta iteración la historia de usuario se centra en la gestión de cuentas de usuario, aquí el administrador podrá: cambiar rol de la cuenta de los usuarios, editar usuarios y eliminar usuarios.

ACTIVIDADES	RESPONSABLES	FECHA INICIO	FECHA FIN	ESTADO	ATRASO
Diseño del control del administrador	Santiago Roman	14/07/2023	21/07/2023	Finalizada	Ninguno
Codificar los crud y acciones necesarias para el control de usuarios por parte del administrador	Santiago Roman	14/07/2023	21/07/2023	Finalizada	Ninguno
Pruebas de funcionamiento basicas	Jaime Oswaldo Paqui Medina	14/07/2023	15/07/2023	Finalizada	4 días

RESUMEN GENERAL	
Actividades Resueltas	4 actividades
Actividades Sin Resolver	0 actividades
Actividades con retraso	1 actividad
Actividades Descartadas	0 actividad

ACTIVIDADES DESCARTADAS
No hay actividades descartadas

Figura A5 39. Resultados del Sprint 6.

```

@login_required
def cambiarRolUsuario(request, usuario_id):
    user = request.user
    if user.groups.filter(name = "admin").exists():
        usuario = Usuario.objects.get(usuario_id=usuario_id)
        user = User.objects.get(email = usuario.correo)
        formulario_rol = FormularioRol(request.POST)
        if request.method == 'POST':
            if formulario_rol.is_valid():
                formulario_rol_data = formulario_rol.cleaned_data
                rol = formulario_rol_data['rol']
                rolAnterior, idRol = verificarRolAnterior(usuario.usuario_id)
                if rol == 'estudiante' and rolAnterior !=1: ...
                elif rol == 'docente' and rolAnterior !=2: ...
                elif rol == 'admin' and rolAnterior !=3: ...
            else:
                print("error al usar el boton o de rol repetido")
                message=("error rol repetido.")
                url = reverse('usuarios') + f'?message={message}'
                return redirect(url)
                message=("rol cambiado con exito")
                url = reverse('usuarios') + f'?message={message}'
                return redirect(url)
            message=("error en el formulario")
            url = reverse('usuarios') + f'?message={message}'
            return redirect(url)
        return render (request, 'usuarios/rol.html', locals())
    else:
        return render(request, 'login/forbidden.html', locals())

```

Figura A5 40. Método para Cambiar rol de la cuenta de los usuarios.

```

@login_required
def modificarUsuario(request, usuario_id):
    user = request.user
    usuario = Usuario.objects.get(correo=user.email)
    # if user.groups.filter(name = "admin").exists() or usuario.usuario_id = usuario_id:
    usuario = Usuario.objects.get(usuario_id=usuario_id)
    if request.method == 'GET':
        formulario_usuario = FormularioUsuario(instance = usuario)
    else:
        formulario_usuario = FormularioUsuario(request.POST, instance = usuario)
        if formulario_usuario.is_valid():
            #ORM
            formulario_usuario.save()
            return redirect(index)
        return render (request, 'usuarios/modificar.html', locals())
    # else:
    #     return render(request, 'login/forbidden.html', locals())

```

Figura A5 41. Método para editar usuarios.

```

@login_required
def eliminarUsuario(request, usuario_id):
    user = request.user
    if user.groups.filter(name = "admin").exists():
        try:
            rolAnterior, idRol = verificarRolAnterior(usuario_id)
            if rolAnterior == 2:
                docente = Docente.objects.get(docente_id = idRol)
                docente.delete()
            usuario = Usuario.objects.get(usuario_id=usuario_id)
            usuario.delete()
            return redirect(index)
        except ProtectedError as e:
            # Manejo del error
            registros_asociados = e.protected_objects
            message = f"No se puede eliminar el Docente debido a registros asociados en GestionDocumentos: {registros_asociados}"
            url = reverse('usuarios') + f'?message={message}'
            return redirect(url)
    else:
        return render(request, 'login/forbidden.html', locals())

```

Figura A5 42. Método para eliminar usuarios.

4. Fase de Pruebas

En el **Anexo 7** se tiene el plan de Pruebas unitarias.

Plan de Pruebas Unitarias

Trabajo: Prototipo de Software Antiplagio para la Carrera de Ingeniería en Computación de la Universidad Nacional de Loja, utilizando Redes.

Versión: 1.0

Fecha: 31/07/2023

Autor: Jaime Paqui y Santiago Román.

Correo electrónico: jaime.paqui@unl.edu.ec, santiago.roman@unl.edu.ec

Hoja de control

Organismo	Universidad Nacional de Loja		
Trabajo	Prototipo de Software Antiplagio para la Carrera de Ingeniería en Computación de la Universidad Nacional de Loja, utilizando Redes.		
Entregable	Planes de Pruebas Unitarias		
Autor	Jaime Paqui y Santiago Román.		
Versión/Edición	1.0	Fecha Versión	28/07/2021
Aprobado por	Oscar Miguel Cumbicus Pineda.	Fecha Aprobación	31/07/2021
		N.º Total de Páginas	8

Registro de cambios

Versión doc.	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
1.0	Versión inicial del Plan de Pruebas Unitarias.	Jaime Oswaldo Paqui Medina	26/07/2023

Control de distribución

Nombre y Apellidos
Jaime Oswaldo Paqui Medina.
Santiago Alexander Román Silva.
Oscar Miguel Cumbicus Pineda.

1.	Introducción	160
1.1	Objetivo	160
1.2	Propósito	160
2.	Definición de los casos de pruebas.....	160
3.	Glosario	163

1. Introducción

1.1 Objetivo

El objetivo de este documento es verificar la funcionalidad correcta de las funciones implementados en el prototipo de software anti plagio, esperando así que cada función genere un resultado esperado y se comporte de manera correcta en distintos escenarios.

1.2 Propósito

Comprobar el correcto funcionamiento del prototipo de software anti plagio, para asegurar que este cumpla con los requisitos del usuario. Definición de los casos de pruebas.

Para esta sección se procede a describir cada uno de los casos de prueba identificados:

Número del caso de prueba	Componente	Descripción del caso	Prerrequisitos
CP01	Usuario	Validar que los usuarios puedan registrarse en el sistema.	Correo electrónico de la UNL.
CP02	Usuario	Validar que los usuarios puedan ingresar al sistema	Disponer de una cuenta en el sistema Antiplagio.
CP03	Usuario	Validar el registro del usuario.	Disponer del rol admin y los permisos necesarios.
CP04	Usuario	Verificar que se pueda modificar el rol de un usuario.	Disponer de una cuenta en el sistema Antiplagio.
CP05	Usuario	Verificar que se pueda eliminar un usuario.	Disponer de una cuenta en el sistema Antiplagio.
CP06	Archivo	Validar que se analice un documento en búsqueda de plagio.	Disponer de un documento.
CP07	Archivo	Validar que se pueda editar un resultado de plagio.	Disponer de un resultado de plagio realizado por el sistema.
CP08	Archivo	Verificar que se pueda eliminar un resultado de plagio	Verificar que se pueda eliminar un resultado de plagio

CP01

No	Descripción	Método	Datos de Entrada	¿ok?	Observaciones
1	El usuario ingresa los datos en el formulario de registro, posteriormente selecciona el rol y se verifica que no exista una cuenta de usuario.	RegistrarWizarView()	Formulario: <ul style="list-style-type: none"> • nombre • apellido • usuario • contraseña • rol 	✓	N/A

CP02					
No	Descripción	Método	Datos de Entrada	¿ok?	Observaciones
1	El usuario ingresa sus credenciales, posteriormente el sistema valida la autenticidad del usuario.	autenticar()	Formulario: <ul style="list-style-type: none"> • username • password 	✓	N/A

CP03					
No	Descripción	Método	Datos de Entrada	¿ok?	Observaciones
1	El administrador ingresa al sistema y valida el registro del usuario.	validarUsuario()	Formulario de Registro de usuario.	✓	N/A

CP04					
No	Descripción	Método	Datos de Entrada	¿ok?	Observaciones
1	El administrador ingresa al sistema y modifica el rol del usuario.	cambiarRol()	Usuario registrado con rol de estudiante.	✓	N/A

CP05					
No	Descripción	Método	Datos de Entrada	¿ok?	Observaciones
1	El administrador ingresa al sistema y elimina una cuenta de usuario.	eliminarUsuario()	Usuario registrado.	✓	N/A

CP06					
No	Descripción	Método	Datos de Entrada	¿ok?	Observaciones
1	El usuario sube un documento al sistema y se verifica que realice el análisis de plagio.	analizar()	Documento PDF.	✓	N/A

CP07					
No	Descripción	Método	Datos de Entrada	¿ok?	Observaciones
1	El usuario selecciona un documento analizado y se verifica que se pueda eliminar los párrafos plagiados.	editarAnálisis()	Documento analizado.	✓	N/A

CP08					
No	Descripción	Método	Datos de Entrada	¿ok?	Observaciones
1	El usuario selecciona un documento analizado y se verifica que se pueda eliminar el resultado de plagio.	eliminarAnálisis()	Documento analizado.	✓	N/A

2. Glosario

En esta sección se presenta los términos usados en el documento junto con su correspondiente significado.

Término	Descripción
PDF	Formato de documento portátil
UNL	Universidad Nacional de Loja.

3. Aprobaciones

Nombre y Apellido	Cargo / Rol	Firma
Ing. Oscar Miguel Cumbicus Pineda	Cliente del producto	

Plan de Pruebas Funcionales

Trabajo: Prototipo de Software Antiplagio para la Carrera de Ingeniería en Computación de la Universidad Nacional de Loja, utilizando Redes.

Versión: 1.0

Fecha: 31/07/2023

Autor: Jaime Paqui y Santiago Román

Correo electrónico: jaime.paqui@unl.edu.ec , santiago.roman@unl.edu.ec

Hoja de control

Organismo	Universidad Nacional de Loja		
Trabajo	Prototipo de Software Antiplagio para la Carrera de Ingeniería en Computación de la Universidad Nacional de Loja, utilizando Redes.		
Entregable	Planes de Pruebas Funcionales.		
Autor	Jaime Paqui y Santiago Román.		
Versión/Edición	1.0	Fecha Versión	31/07/2021
Aprobado por	Oscar Miguel Cumbicus Pineda.	Fecha Aprobación	
		N.º Total de Páginas	

Registro de cambios

Versión doc.	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
1.0	Versión inicial del Plan de Pruebas Funcionales	Jaime Oswaldo Paqui Medina y Santiago Alexander Román Silva.	31/07/2023

Control de distribución

Nombre y Apellidos
Jaime Oswaldo Paqui Medina
Santiago Alexander Román Silva
Oscar Miguel Cumbicus Pineda, Mg. Sc

1.	INTRODUCCIÓN	167
1.1	Objetivo.....	167
1.2	Alcance	167
2.	TRAZABILIDAD DE CASOS DE PRUEBAS – REQUISITOS.....	168
3.	DEFINICIÓN DE LOS CASOS DE PRUEBAS	169
4.	ESTRATEGIA DE EJECUCIÓN DE PRUEBAS	183
5.	GLOSARIO	184
6.	BIBLIOGRAFÍA Y REFERENCIAS.....	185
7.	APROBACIONES.....	186

1. INTRODUCCIÓN

1.1 Objetivo

El presente documento tiene como objetivo reunir casos de prueba para garantizar que prototipo de software anti plagio, cumpla con los requisitos del usuario. Para cada caso de prueba se tiene de una breve descripción, junto con una matriz que relaciona los casos con los requisitos funcionales. A si mismo se tiene una estrategia, para llevare a cabo la ejecución de pruebas de manera correcta.

1.2 Alcance

Los casos de prueba están destinados a ser evaluados por el docente Oscar Miguel Cumbicus Pineda, director del Trabajo de Titulación (TT), docente de la Carrera de Computación de la UNL. Los estudiantes Jaime Oswaldo Paqui Medina y Santiago Alexander Román Silva que son quienes generan y registran los casos de prueba.

2. TRAZABILIDAD DE CASOS DE PRUEBAS – REQUISITOS

En la siguiente tabla se presenta cómo se relacionan los casos de pruebas definidos junto con los requisitos funcionales descritos en la especificación de requisitos. Cada fila representa un caso de prueba definido, mientras que las columnas representan los requisitos funcionales. La marca "x" indica la correspondencia existente entre ellos.

	RF-01	RF-02	RF-03	RF-06	RF-07	RF-08	RF-10	RF-13
CP01	X							
CP02	X							
CP03	X							
CP04			X					
CP05		X						
CP06		X						
CP07		X						
CP08		X		X				
CP09					X			
CP10					X			
CP11		X				X		
CP12		X				X		
CP13		X				X		
CP14						X		X

3. DEFINICIÓN DE LOS CASOS DE PRUEBAS

En esta sección se proporciona una descripción detallada de todos los casos de pruebas identificados para evaluar el funcionamiento del sistema. También se muestra los pasos a seguir para ejecutar cada caso de prueba.

Registrar un nuevo usuario	CP01	
	¿Prueba de despliegue?	No
Descripción: Se verificará que se pueda registrar un usuario, que no disponga de ninguna cuenta de usuario.		
Prerrequisitos <ul style="list-style-type: none">• Ingresar a la página de inicio.• Ingresar en Registrarse.• Disponer de un correo de la UNL.		
Pasos: <ol style="list-style-type: none">1. Hacer clic en “registrarse”.2. Llenar el formulario con sus datos.3. Hacer clic en siguiente.4. Seleccionar el rol.5. Hacer clic en “registrarse”.6. Esperar el mensaje de aviso de validación de la cuenta.7. Revisar el correo electrónico la validación de la cuenta.		
Resultado esperado: <ul style="list-style-type: none">• Mensaje de aviso de validación.• Mensaje de confirmación de la cuenta.		
Resultado obtenido: <Resultado obtenido de la ejecución del caso de prueba> <ul style="list-style-type: none">• Mensaje de aviso de validación. <div data-bbox="375 1234 976 1760" style="border: 1px solid #ccc; padding: 20px; text-align: center; background-color: #f9f9f9;"><p>EL REGISTRO SE HA COMPLETADO</p><p>La cuenta está pendiente de confirmación. Le llegará un correo electrónico cuando su cuenta esté activa.</p><p>Cerrar</p></div>		

Mensaje de confirmación de la cuenta

¡Registro de cuenta exitoso!

Gracias test1 test1 por registrarte en nuestro sitio. Tu cuenta ha sido autorizada con éxito.

[Iniciar sesión](#)

Registro con correo electrónico existente.	CP02	
	¿Prueba de despliegue?	No
Descripción: Se verificará que se pueda registrar un usuario con correo electrónico repetido.		
Prerrequisitos <ul style="list-style-type: none">• Ingresar a la página de inicio.• Ingresar en Registrarse.• Disponer de un correo de la UNL.		
Pasos: <ol style="list-style-type: none">1. Hacer clic en “registrarse”.2. Ingresar un correo electrónico que ya está registrado en el sistema.3. Hacer clic en “registrarse”.		
Resultado esperado: <ul style="list-style-type: none">• Mensaje que muestre que correo electrónico ya está en uso.		
Resultado obtenido: <Resultado obtenido de la ejecución del caso de prueba>		

Crear Una Cuenta

 Nombre

 Apellido

 Usuario

 Email

• **Email is already in use.**

 Contraseña

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

Continuar

Registrar un usuario con datos incompletos.	CP03	
	¿Prueba de despliegue?	No
Descripción: Se verificará que se pueda realizar un registro con información incompleta.		
Prerrequisitos <ul style="list-style-type: none"> • Ingresar a la página de inicio. • Ingresar en Registrarse. • Disponer de un correo de la UNL. 		
Pasos: <ol style="list-style-type: none"> 3. Hacer clic en “registrarse”. 4. Dejar algunos campos vacíos. 5. Hacer clic en siguiente. 		
Resultado esperado: <ul style="list-style-type: none"> • Mensaje de aviso, que indica que se debe completar todos los campos. 		
Resultado obtenido: <Resultado obtenido de la ejecución del caso de prueba>		
 <p>The screenshot shows a registration form titled "Crear Una Cuenta". It has four input fields: "Nombre" (filled with "emanuel"), "Apellido" (filled with "emanuel"), "Usuario" (empty), and "Contraseña" (empty). A red error message "Completa este campo" with an exclamation mark icon is shown above the "Usuario" field. Below it, a red bullet point states "• Email is already in use." The "Contraseña" field has a list of requirements: it cannot be similar to other personal information, must be at least 8 characters, cannot be a common key, and cannot be purely numeric. A blue "Continuar" button is at the bottom.</p>		

Validar cuenta de usuario	CP04	
	¿Prueba de despliegue?	No
Descripción: Se verificará que el administrador pueda validar una cuenta de usuario correctamente.		
Prerrequisitos <ul style="list-style-type: none"> • Ingresar a la página de inicio. • Ingresar en “Iniciar sesión”. 		
Pasos: 1. Hacer clic en “Autorización”. 2. Hacer clic en siguiente.		
Resultado esperado: <ul style="list-style-type: none"> • Mensaje de usuario autorizado. 		
Resultado obtenido: <Resultado obtenido de la ejecución del caso de prueba>		
		

Inicio de sesión	CP05	
	¿Prueba de despliegue?	No
Descripción: Se verificará que se pueda ingresar al sistema.		
Prerrequisitos <ul style="list-style-type: none"> • Ingresar a la página de inicio. • Ingresar en Iniciar sesión. • Disponer de una cuenta de usuario. 		
Pasos: 6. Hacer clic en “Iniciar sesión”. 7. Ingresar sus credenciales (nombre de usuario y contraseña). 8. Hacer clic iniciar.		
Resultado esperado: <ul style="list-style-type: none"> • Visualizar en pantalla bienvenido, junto con el nombre de usuario. 		
Resultado obtenido: <Resultado obtenido de la ejecución del caso de prueba>		



Inicio de sesión con credenciales no validas.	CP06	
	¿Prueba de despliegue?	No
Descripción: Se verificará que el sistema valide el ingreso con credenciales no validas.		
Prerrequisitos <ul style="list-style-type: none"> • Ingresar a la página de inicio. • Ingresar en Iniciar sesión. • Disponer de una cuenta de usuario. 		
Pasos: <ol style="list-style-type: none"> 1. Hacer clic en “Iniciar sesión”. 2. Ingresar sus credenciales (nombre de usuario y contraseña). 3. Hacer clic iniciar. 		
Resultado esperado: <ul style="list-style-type: none"> • Mensaje de aviso, credenciales incorrectas. 		
Resultado obtenido: <Resultado obtenido de la ejecución del caso de prueba>		

Bienvenido

 Usuario

 Contraseña

Usuario o Clave incorrectas

[¿Olvidaste tu contraseña?](#)

Ingresar

Recuperar contraseña	CP07	
	¿Prueba de despliegue?	No
Descripción: Se verificará que se pueda restablecer la contraseña, en caso de olvidarla.		
Prerrequisitos <ul style="list-style-type: none"> Ingresar a la página de inicio. Ingresar en Iniciar sesión. Disponer de una cuenta de usuario. 		
Pasos: <ol style="list-style-type: none"> 1. Hacer clic en “Iniciar sesión”. 2. Hacer clic en “¿Olvidó su contraseña?”. 3. Ingresar correo electrónico. 4. Hacer clic en “Restablecer contraseña”. 5. Revisar el correo electrónico donde llega la nueva contraseña. 		

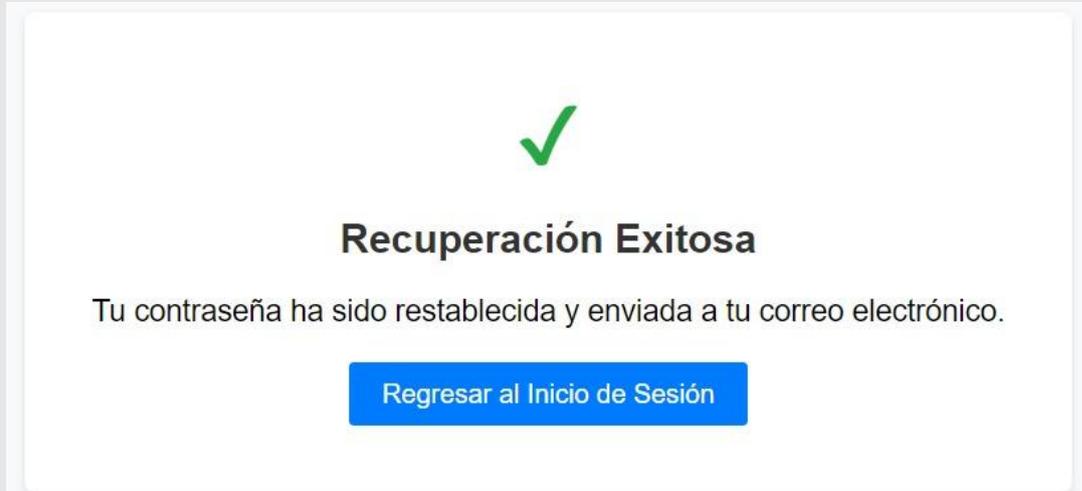
6. Ingresar al sistema con la contraseña nueva.

Resultado esperado:

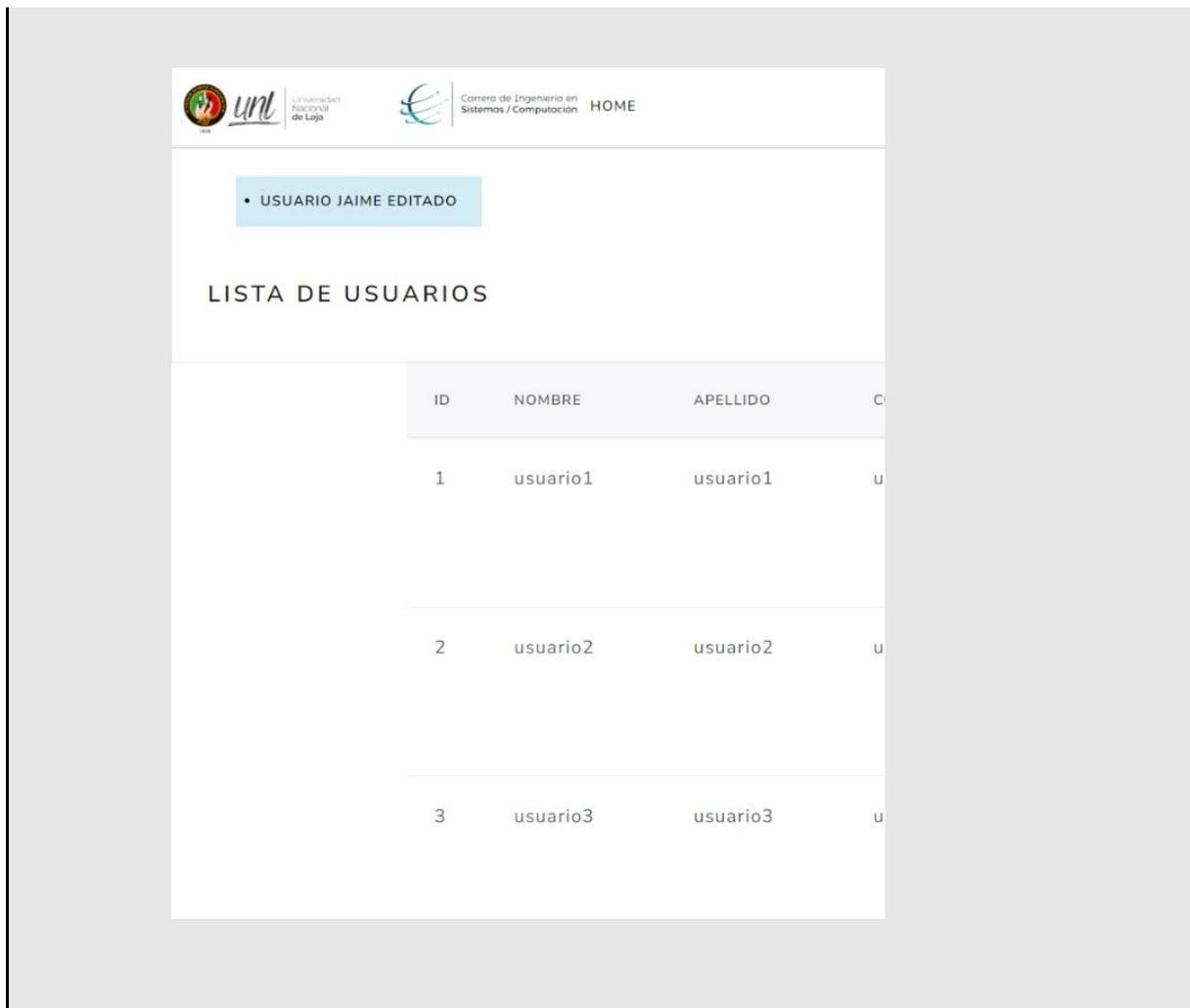
- Mensaje de recuperación exitosa.

Resultado obtenido:

<Resultado obtenido de la ejecución del caso de prueba>



Edición exitosa de la información de un usuario	CP08	
	¿Prueba de despliegue?	No
Descripción: Se verificará que se pueda realizar una modificación de datos de los usuarios.		
Prerrequisitos <ul style="list-style-type: none">• Ingresar a la página de inicio.• Ingresar en Iniciar sesión como administrador.		
Pasos: <ol style="list-style-type: none">7. Seleccionar un usuario existente y hacer clic en “Editar”.8. Ingresar la información a modificar.9. Hacer clic en “Modificar usuario”.		
Resultado esperado: <ul style="list-style-type: none">• Se confirma los cambios con un mensaje usuario editado.		
Resultado obtenido: <Resultado obtenido de la ejecución del caso de prueba>		



Eliminación exitosa de un usuario	CP09	
	¿Prueba de despliegue?	No
Descripción: Se verificará que se pueda eliminar un usuario registrado.		
Prerrequisitos <ul style="list-style-type: none"> • Ingresar a la página de inicio. • Ingresar en Iniciar sesión como administrador. 		
Pasos: <ol style="list-style-type: none"> 1. Seleccionar un usuario existente y hacer clic en “Eliminar”. 2. Confirmar la eliminación del usuario haciendo clic en “Eliminar usuario”. 		
Resultado esperado: <ul style="list-style-type: none"> • Mensaje de confirmación para eliminar. • Se confirma los cambios con un mensaje de usuario eliminado. 		
Resultado obtenido: <Resultado obtenido de la ejecución del caso de prueba>		



¿Estás seguro, esta acción no se podrá revertir?

Esta acción eliminará el usuario registrado.

Eliminar usuario

Cancelar

• USUARIO JAIME ELIMINADO

LISTA DE USUARIOS

ID	NOMBRE	APELLIDO	CORREO
1	usuario1	usuario1	usuario1@usuario.com
2	usuario2	usuario2	usuario2@usuario.com
3	usuario3	usuario3	usuario3@usuario.com

Cargar documentos de forma exitosa.	CP11	
	¿Prueba de despliegue?	No
Descripción: Se verificará que se pueda subir un documento al sistema para su posterior análisis.		
Prerrequisitos <ul style="list-style-type: none"> • Ingresar a la página de inicio. • Ingresar en Iniciar sesión. 		
Pasos: <ol style="list-style-type: none"> 3. Hacer clic en documentos. 4. Hacer clic en seleccionar documento. 5. Seleccionar el documento. 6. Hacer clic en abrir. 		
Resultado esperado: <ul style="list-style-type: none"> • Se muestra en pantalla el nombre del documento seleccionado. 		
Resultado obtenido: <Resultado obtenido de la ejecución del caso de prueba> <div style="border: 1px solid gray; padding: 10px; text-align: center; margin: 10px 0;"> <h2 style="margin: 0;">SUBIR ARCHIVO</h2> <div style="display: flex; justify-content: center; align-items: center; gap: 20px;"> <div style="background-color: #007bff; color: white; padding: 5px 15px; border-radius: 5px;">Seleccionar Archivo</div> CloudCOriginal.pdf </div> <div style="display: flex; justify-content: center; align-items: center; gap: 20px; margin-top: 10px;"> <div style="background-color: #dc3545; color: white; padding: 5px 15px; border-radius: 5px;">Cancelar</div> <div style="background-color: #007bff; color: white; padding: 5px 15px; border-radius: 5px;">Subir Archivo</div> </div> </div>		

Análisis de plagio completado	CP12	
	¿Prueba de despliegue?	No
Descripción: Se verificará que se detecte plagio con documento idéntico.		
Prerrequisitos <ul style="list-style-type: none"> • Ingresar a la página de inicio. • Ingresar en Iniciar sesión. 		
Pasos: <ol style="list-style-type: none"> 1. Hacer clic en documentos. 2. Hacer clic en seleccionar documento. 3. Seleccionar el documento. 4. Hacer clic en abrir. 5. Hacer clic en enviar. 6. Ingresar los campos para el envío de resultado al docente. 		

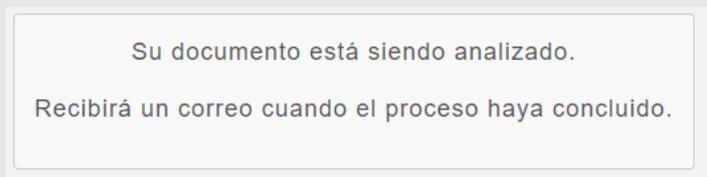
7. Hacer clic en enviar.

Resultado esperado:

- Se presenta en pantalla mensaje de aviso que el documento está siendo analizado.
- En la sección resultados se muestra el resultado de plagio

Resultado obtenido:

<Resultado obtenido de la ejecución del caso de prueba>



Numero	Documento	Subido Por	Revision
1	Introducción_2_XeZokhk.pdf	jaimo jaimo	Revision Eliminar

Excluir párrafos plagiados	CP13	
	¿Prueba de despliegue?	No
Descripción: Se verificará que se pueda reducir el porcentaje de plagio.		
Prerrequisitos <ul style="list-style-type: none">• Ingresar a la página de inicio.• Ingresar en Iniciar sesión.		
Pasos: <ol style="list-style-type: none">1. Hacer clic en “revisión”.2. Seleccionar el resultado de plagio.3. Hacer clic en “Editar”.4. Hacer clic en Eliminar párrafo.		
Resultado esperado: <ul style="list-style-type: none">• Se muestra en pantalla como el porcentaje de plagio disminuye.		
Resultado obtenido: <Resultado obtenido de la ejecución del caso de prueba>		
Plagio general		

TABLA DE PLAGIOS

Porcentaje de plagio: 100.00%

num oraciones del documento: 16

ORACIÓN PLAGIADA	ORACIÓN ORIGINAL	PORCENTAJE DE PLAGIO	LUGAR DONDE SE ENCONTRÓ	UBICACIÓN	ACCIONES
El concepto de la computacion en la nube se basa en la utilizacion de servidores remotos y compartidos para almacenar y procesar datos	El concepto de la computacion en la nube se basa en la utilizacion de servidores remotos y compartidos para almacenar y procesar datos	100,00%	CloudCPlagio.pdf	(54, 77)	ELIMINAR
Uno de los principales beneficios de la computacion en la nube es su capacidad de escalar	Uno de los principales beneficios de la computacion en la nube es su capacidad de escalar	100,00%	CloudCPlagio.pdf	(116, 132)	ELIMINAR
Esta tecnologia ha brindado a individuos y empresas la posibilidad de aprovechar recursos informaticos de manera flexible y escalable, sin la necesidad de invertir en infraestructur a costosa	Esta tecnologia ha brindado a individuos y empresas la posibilidad de aprovechar recursos informaticos de manera flexible y escalable, sin la necesidad de invertir en infraestructur a costosa	100,00%	CloudCPlagio.pdf	(25, 54)	ELIMINAR
La computacion en la nube ha causado un impacto significativo en la manera en que almacenamos, accedemos y procesamos informacion en la era digital	La computacion en la nube ha causado un impacto significativo en la manera en que almacenamos, accedemos y procesamos informacion en la era digital	100,00%	CloudCPlagio.pdf	(0, 25)	ELIMINAR

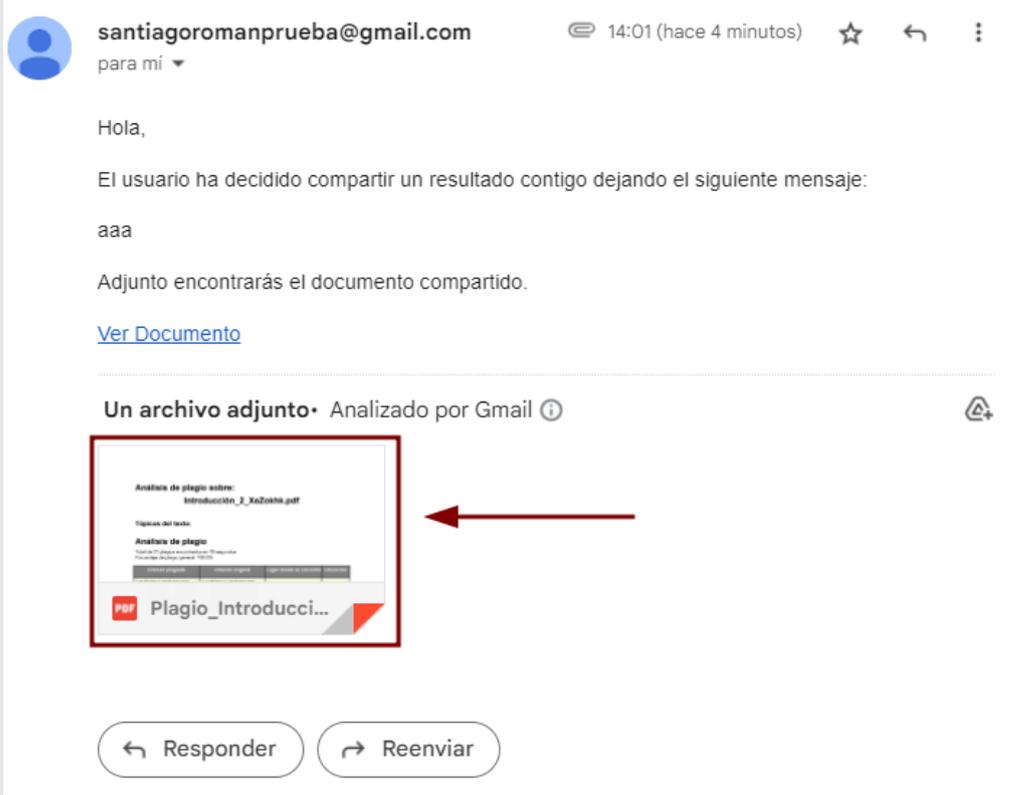
Plagio Excluido

TABLA DE PLAGIOS

Porcentaje de plagio: 18.75%

num oraciones del documento: 16

ORACIÓN PLAGIADA	ORACIÓN ORIGINAL	PORCENTAJE DE PLAGIO	LUGAR DONDE SE ENCONTRÓ	UBICACIÓN	ACCIONES
Ademas, la computacion en la nube ha permitido el surgimiento de servicios de software como servicio (SaaS), que proporcionan acceso a aplicaciones y herramientas de alta calidad sin necesidad de instalarlas localmente	Ademas, la computacion en la nube ha permitido el surgimiento de servicios de software como servicio (SaaS), que proporcionan acceso a aplicaciones y herramientas de alta calidad sin necesidad de instalarlas localmente	100,00%	CloudCPlagio.pdf	(280, 317)	ELIMINAR
Su flexibilidad, escalabilidad y disponibilidad han permitido a personas y organizaciones aprovechar al maximo la tecnologia sin incurrir en costos excesivos	Su flexibilidad, escalabilidad y disponibilidad han permitido a personas y organizaciones aprovechar al maximo la tecnologia sin incurrir en costos excesivos	100,00%	CloudCPlagio.pdf	(342, 364)	ELIMINAR
En resumen, la computacion en la nube ha tenido un impacto significativo en la forma en que utilizamos y accedemos a los recursos informaticos	En resumen, la computacion en la nube ha tenido un impacto significativo en la forma en que utilizamos y accedemos a los recursos informaticos	100,00%	CloudCPlagio.pdf	(317, 342)	ELIMINAR

Envió exitoso de resultado de plagio por correo electrónico.	CP14	
	¿Prueba de despliegue?	No
Descripción: Se verificará que se pueda reducir el porcentaje de plagio.		
Prerrequisitos <ul style="list-style-type: none"> • Ingresar a la página de inicio. • Ingresar en Iniciar sesión. 		
Pasos: <ol style="list-style-type: none"> 5. Hacer clic en “revisión”. 6. Seleccionar el resultado de plagio. 7. Hacer clic en “Enviar”. 8. Ingresar los datos solicitados. 9. Hacer clic en “Compartir”. 		
Resultado esperado: <ul style="list-style-type: none"> • El documento se puede visualizar en el correo electrónico destino, junto a un mensaje. 		
Resultado obtenido: <Resultado obtenido de la ejecución del caso de prueba> <div data-bbox="248 869 1279 1664" style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;">  <p>The screenshot shows an email interface. At the top, it says 'santiagoromanprueba@gmail.com' and '14:01 (hace 4 minutos)'. The body of the email says 'Hola, El usuario ha decidido compartir un resultado contigo dejando el siguiente mensaje: aaa Adjunto encontrarás el documento compartido. Ver Documento'. Below this is a section for an attachment: 'Un archivo adjunto • Analizado por Gmail'. The attachment is a PDF file named 'Plagio_Introducci...' with a thumbnail showing a plagiarism analysis report. A red arrow points to the thumbnail. At the bottom of the email view are buttons for 'Responder' and 'Reenviar'.</p> </div>		

4. ESTRATEGIA DE EJECUCIÓN DE PRUEBAS

En esta sección se menciona los casos de prueba, las mismas que se ejecutan en tres ciclos.

	Ciclo 1	Ciclo 2	Ciclo 3
CP01	X		
CP02	X		
CP03	X		
CP04	X		
CP05	X		
CP06		X	
CP07		X	
CP08		X	
CP09		X	
CP10		X	
CP11			X
CP12			X
CP13			X
CP14			X

5. GLOSARIO

En esta sección se presenta los términos usados en el documento junto con su correspondiente significado.

Término	Descripción
UNL	Universidad Nacional de Loja
TT	Trabajo de Titulación

6. BIBLIOGRAFÍA Y REFERENCIAS

Referencia	Título
Ref. 1	Documentación de Especificación de Requisitos.
Ref. 2	Platilla del Plan de Pruebas Funcionales del portal del marco de desarrollo de software de la Junta de Andalucía.

7. APROBACIONES

Nombre y Apellido	Cargo / Rol	Firma
Ing. Oscar Miguel Cumbicus Pineda	Cliente del producto	

Anexo 9. Manual de usuario del Administrador.

Sistema de anti-Plagio para la carrera de Computación de la Universidad Nacional de Loja

Elaborado por:

Jaime Oswaldo Paqui Medina

Santiago Alexander Román Silva

Revisado por:

Ing. Oscar Miguel Cumbicus Pineda

Aprobado por:

Ing. Oscar Miguel Cumbicus Pineda

Mayo 2023

TABLA DE CONTENIDO

Historial de versiones	189
Información del Trabajo.....	189
1. Introducción.....	190
2. Acceso al Sistema.....	190
3. Gestionar Usuarios.....	194
4. Firmas de Responsabilidad	200

Historial de versiones

Versión	Fecha	Responsable	Descripción de la modificación
1	06-04-2023	Ing. Oscar Miguel Cumbicus	Documento inicial

Información del Trabajo

Tabla 1. Trabajo para la UNL.

Nombre del Trabajo	Sistema de anti-Plagio para la carrera de Computación de la Universidad Nacional de Loja.
Fecha de elaboración	01-04-2023
Solicitante / Patrocinador principal	Carrera de Computación - Universidad Nacional de Loja

1. Introducción

En el presente documento se tiene la documentación adecuada para guiar a los usuarios en el uso adecuado del **Sistema Antiplagio**, pone a disposición del lector, el siguiente documento donde constan las diversas funcionalidades que tiene el sistema para el rol de Administrador, con el fin de que puedan agregar usuarios, modificar usuarios, validar registro de usuarios, cambiar rol de los usuarios, entre otros.

2. Acceso al Sistema

1. Abrir el navegador y dirigir <https://Antiplagiodocker.azurewebsites.net/>
2. A continuación, en la parte superior derecha hay un menú denominado **[Ingresar]** hacer clic para ingresar.

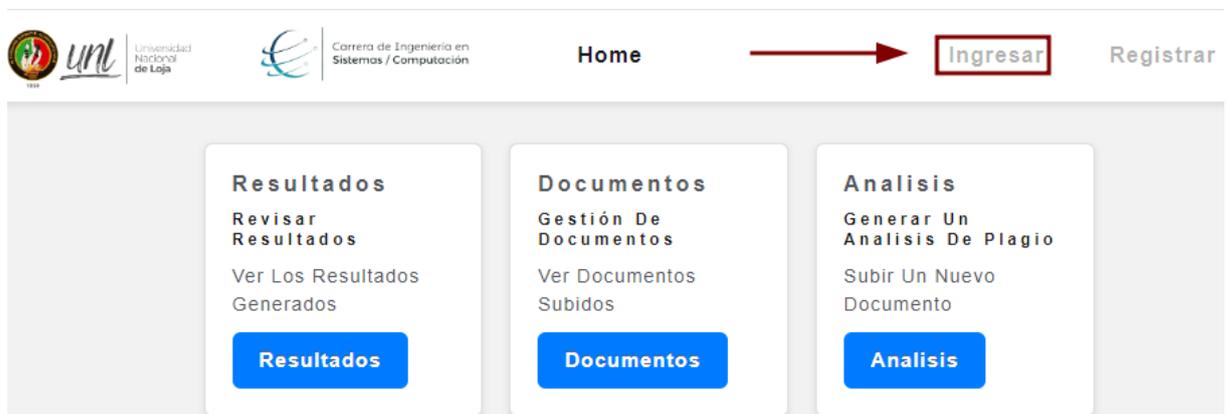


Imagen 1: Página principal.

3. Posteriormente el sistema solicitará ingresar las credenciales de acceso.

Imagen 2: Formulario para ingresar credenciales.

4. A continuación, ingresar credenciales en el formulario y hacer clic en el botón **[Ingresar]**.



Bienvenido

Usuario

Contraseña

[¿Olvidaste tu contraseña?](#)

Ingresar

Imagen 3: Ingresar al sistema.

5. Posteriormente se le presenta la página principal, dándole la bienvenida al usuario.



Detector De Plagio

Bienvenido Admin!

Admin
Ingresar Al Admin
Configurar Las Opciones De Django
Admin

Usuarios
Gestionar Usuarios
Gestionar Los Usuarios.
Usuario

Imagen 4: Página Inicio

6. Para recuperar la contraseña, repetir el paso 1,2 y 3.

7. Posteriormente, hacer clic en **[Olvidaste tu contraseña]**.

The image shows a login interface with the title "Bienvenido". It contains two input fields: "Usuario" (with a person icon) and "Contraseña" (with a lock icon). Below the password field is a link that says "¿Olvidaste tu contraseña?". This link is highlighted with a red rectangular box, and a red arrow points to it from the left. Below the link is a blue button labeled "Ingresar".

Imagen 5: Recuperar contraseña.

8. A continuación, ingresar el correo electrónico y hacer clic en **[Recuperar Contraseña]**.

The image shows a password recovery form with the title "Recuperar Contraseña". It features a label "✉ Correo electrónico" above a text input field. The input field is highlighted with a red rectangular box, and a red arrow points to it from the right. Below the input field is a blue button labeled "Recuperar Contraseña". This button is also highlighted with a red rectangular box, and a red arrow points to it from the right.

Imagen 6: Formulario recuperar contraseña.

9. Posterior se presenta un mensaje de recuperación exitosa, hacer clic en **[Regresar al Inicio de Sesión]** para regresar a la página de inicio.



Recuperación Exitosa

Tu contraseña ha sido restablecida y enviada a tu correo electrónico.

[Regresar al Inicio de Sesión](#)

Imagen 6: Mensaje de confirmación.

10.Revisar el correo electrónico para consultar su nueva contraseña, hacer clic en **[Iniciar Sesión]**.

Recuperación de Contraseña

Hemos recibido una solicitud para restablecer la contraseña de su cuenta.
Su nueva contraseña generada es:

pM09Mm7MH3



Le recomendamos cambiar su contraseña después de iniciar sesión.

Si no solicitó esta acción, se recomienda comunicarse con un administrador.

[Iniciar Sesión](#)



No responda a este correo electrónico. Si necesita ayuda, contáctenos en support@tusitio.com.

Imagen 7: Contraseña nueva.

11.Posterior el sistema le redirige al inicio de sesión, donde podrá ingresar con su nueva contraseña.

3. Gestionar Usuarios

- Paso1: Para gestionar a los usuarios, hacer clic en **[Usuario]** en la página principal del paso 5 del acceso al sistema.



Imagen 8: Gestión de usuarios.

- Paso2: Posteriormente se presenta la lista de usuarios con sus respectivas acciones.

Id	Nombre	Apellido	Correo	Rol	Opciones
1	santiago	roman	santiago.roman@unl.edu.ec	admin	
2	Jhames	Paqui	jaispaqui@gmail.com	estudiante	
3	Allison	Jaramillo	allisonjaramillo2114@gmail.com	docente	
4	Santiago	Roman	santiagorman123@gmail.com	docente	
5	Ale	Ale	ale@gmail.com	estudiante	

Imagen 9: Lista de usuarios.

a. Validar registro de usuario.

- Hacer clic en el icono , que se muestra en la lista de usuarios, mencionado en el paso 2.

- A continuación, se presenta una lista de usuarios registrados, a la espera de la validación de la cuenta.

Lista De Usuarios				
Id	Nombre	Apellido	Correo	Detalles
6	test1	test1	teste1@test.com	Autorizar

Imagen 10: Lista de usuarios.

- Posteriormente, para validar el registro de usuario hacer clic en **[Autorizar]**, ubicado en la columna detalles.

Lista De Usuarios				
Id	Nombre	Apellido	Correo	Detalles
6	test1	test1	teste1@test.com	Autorizar

Imagen 11: Autorizar registro de usuario.

- Seguidamente, se presenta la información del usuario junto con 2 opciones de autorización, para validar el registro del usuario:

AUTORIZACIÓN PARA UN DOCENTE

Nombre: test1

Apellidos: test1

Correo: teste1@test.com

Opción de autorización:

Autorizar

Rechazar

[ENVIAR](#)

Imagen 12: Autorización del usuario.

- Opción 1: Para validar el registro de usuario seleccionar **Autorizar**, posteriormente hacer clic en **[Enviar]**.

AUTORIZACIÓN PARA UN DOCENTE

Nombre: test1
Apellidos: test1
Correo: teste1@test.com

Opción de autorización:

Autorizar ←
 Rechazar

ENVIAR ←

Imagen 13: Autorizar registro.

- Opción 2: Para rechazar el registro de usuario seleccionar **Rechazar**, posteriormente hacer clic en **[Enviar]**.

AUTORIZACIÓN PARA UN DOCENTE

Nombre: test1
Apellidos: test1
Correo: teste1@test.com

Opción de autorización:

Autorizar
 Rechazar ←

ENVIAR ←

Imagen 14: Rechazar registro.

b. **Eliminar usuario.**

- Hacer clic en el icono , que se muestra en la lista de usuarios, mencionado en el paso 2.
- A continuación, se presenta un mensaje de confirmación para eliminar al usuario, hacer clic en **[Eliminar usuario]** para ejecutar la acción eliminar, caso contrario hacer clic en **[Cancelar]** para rechazar la acción.

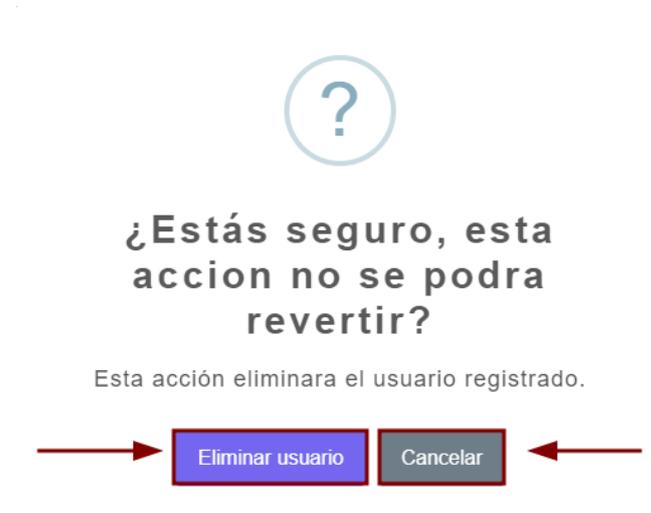


Imagen 15: Eliminar usuario.

c. Editar usuario.

- Hacer clic en el icono , que se muestra en la lista de usuarios, mencionado en el paso 2.
- Seguidamente se presenta un formulario, donde deberá ingresar la información nueva del usuario.
- Una vez ingresado los datos nuevos hacer clic en **[Modificar Usuario]**.

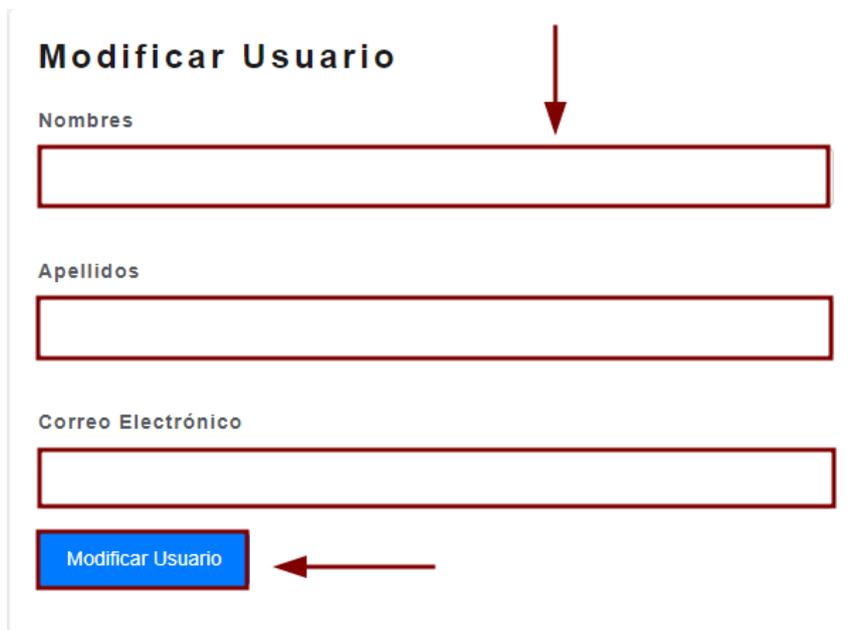
Un formulario con el título "Modificar Usuario". Tiene tres campos de entrada de texto: "Nombres", "Apellidos" y "Correo Electrónico". Cada campo está rodeado por un recuadro rojo. Debajo de los campos, hay un botón azul con el texto "Modificar Usuario". Una flecha roja apunta desde la parte superior hacia el campo "Nombres", y otra flecha roja apunta desde la derecha hacia el botón "Modificar Usuario".

Imagen 16: Eliminar usuario.

d. Cambiar rol del usuario

- Hacer clic en el icono , que se muestra en la lista de usuarios, mencionado en el paso 2.
- A continuación, se presenta un mensaje de confirmación para cambiar el rol del usuario, hacer clic en [**Cambiar Rol**] para ejecutar la acción, caso contrario hacer clic en [**Cancelar**] para rechazar la acción.

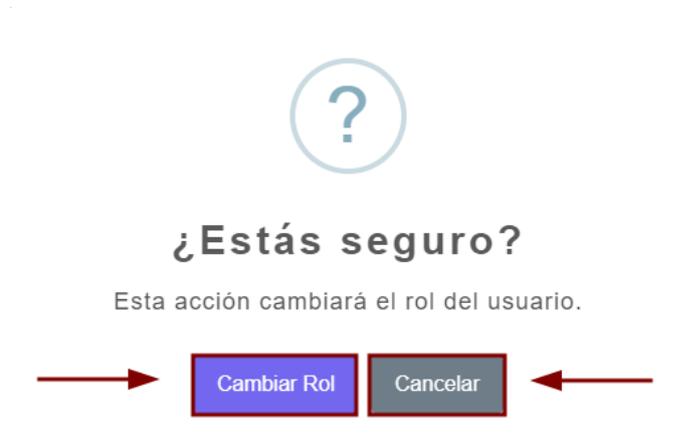


Imagen 17: Mensaje de confirmación.

- Al hacer clic en **Cambiar Rol**, se presenta todos los roles de usuario, hacer clic en [**Seleccionar**] para cambiar el rol del usuario.



Imagen 18: Cambiar rol del usuario.

e. **Buscar usuarios**

- Ingresar el nombre del usuario que desea buscar, en la barra de búsqueda.



Imagen 18: Buscar usuario.

- Hacer clic en el icono , que se muestra en la lista de usuarios, mencionado en el paso 2 para efectuar la búsqueda.

Id	Nombre	Apellido	Correo	Rol	Opciones
1	santiago	roman	santiago.roman@unl.edu.ec		  
4	Santiago	Roman	santiagorman123@gmail.com		  

Imagen 18: Lista de usuarios.

4. Firmas de Responsabilidad

Acción	Funcionario	Firma
Elaborado por:	Jaime Oswaldo Paqui Medina Tesista	
	Santiago Alexander Román Silva Tesista	
Revisado por:	Oscar Miguel Cumbicus Pineda Cliente del Software	
Aprobado por:	Oscar Miguel Cumbicus Pineda Cliente del Software	

Anexo 10. Manual de usuario del Estudiante.

Sistema de anti-Plagio para la carrera de Computación de la Universidad Nacional de Loja

Elaborado por:

Jaime Oswaldo Paqui Medina

Santiago Alexander Román Silva

Revisado por:

Ing. Oscar Miguel Cumbicus Pineda

Aprobado por:

Ing. Oscar Miguel Cumbicus Pineda

TABLA DE CONTENIDO

Historial de versiones	189
Información del Trabajo.....	189
1. Introducción.....	190
2. Acceso al Sistema.....	190
3. Gestionar Usuarios.....	194
4. Firmas de Responsabilidad	200

Historial de versiones

Versión	Fecha	Responsable	Descripción de la modificación
1	06-04-2023	Ing. Oscar Miguel Cumbicus	Documento inicial

Información del Trabajo

Tabla 1. Trabajo para la UNL.

Nombre del Trabajo	Sistema de anti-Plagio para la carrera de Computación de la Universidad Nacional de Loja.
Fecha de elaboración	01-04-2023
Solicitante / Patrocinador principal	Carrera de Computación - Universidad Nacional de Loja

1. Introducción

En el presente documento se tiene la documentación adecuada para guiar a los usuarios en el uso adecuado del **Sistema Antiplagio**, pone a disposición del lector, el siguiente documento donde constan las diversas funcionalidades que tiene el sistema para el rol del Estudiante, con el fin de que puedan agregar usuarios, modificar usuarios, validar registro de usuarios, cambiar rol de los usuarios, entre otros.

2. Registrarse

1. Abrir el navegador y dirigir <https://Antiplagiodocker.azurewebsites.net/>
2. A continuación, en la parte superior derecha hay un menú denominado **[Registrar]** hacer clic para registrarse.

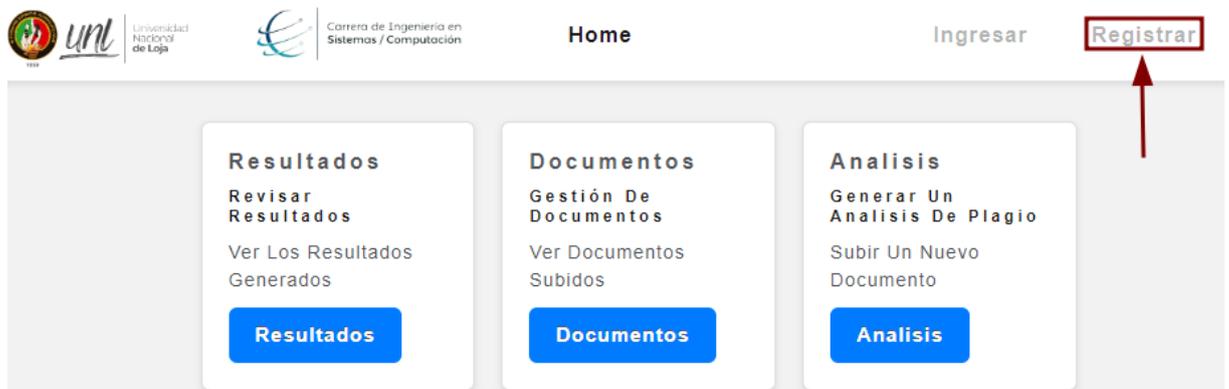


Imagen 1: Página principal.

3. Posteriormente se presenta un formulario, donde deberá ingresar los datos solicitados, hacer clic en **[Continuar]** una vez ingresado los datos. Para la contraseña tener en consideración los mensajes de aviso.

Crear Una Cuenta

 Nombre

 Apellido

 Usuario

 Email

 Contraseña

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

Continuar

Imagen 2: Formulario de registro.

4. A continuación, hacer clic en **[Seleccionar]** en el rol de estudiante, para completar el proceso de registro.



Imagen 3: Rol para el registro.

5. Posteriormente se presenta un mensaje, donde se menciona que el proceso se ha completado y que está a la espera de validación de su cuenta.



Imagen 4: Mensaje de aviso.

6. Revisar su correo electrónico para verificar si el administrador le valida la cuenta.

3. Acceso al Sistema

1. Abrir el navegador y dirigir <https://Antiplagiodocker.azurewebsites.net/>
2. A continuación, en la parte superior derecha hay un menú denominado **[Ingresar]** hacer clic para ingresar.

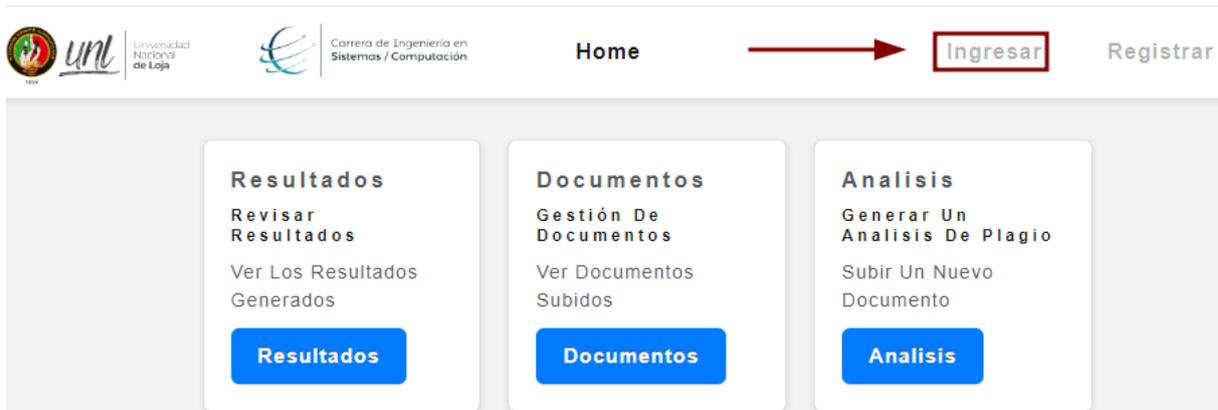


Imagen 5: Página principal.

3. Posteriormente el sistema solicitará ingresar las credenciales de acceso.

Imagen 6: Formulario para ingresar credenciales.

4. A continuación, ingresar credenciales en el formulario y hacer clic en el botón **[Ingresar]**.



Bienvenido

Usuario

Contraseña

¿Olvidaste tu contraseña?

Ingresar

Imagen 7: Ingresar al sistema.

5. Posterior se le presenta la página principal, dándole la bienvenida al usuario.



UNL Universidad Nacional de La Plata Carrera de Ingeniería en Sistemas / Computación Home Salir Cambiar Contraseña

Detector De Plagio

Bienvenido Jaime!

Resultados
Revisar Resultados
Ver Los Resultados Generados
Resultados

Documentos
Gestión De Documentos
Ver Documentos Subidos
Documentos

Análisis
Generar Un Analisis De Plagio
Subir Un Nuevo Documento
Análisis

Imagen 8: Página principal

6. En caso de haber olvidado su contraseña podrá recuperar, para ello repetir el paso 1, 2 y 3.

7. A continuación, hacer clic en **[Olvidaste tu contraseña]**.



The image shows a login form titled "Bienvenido". It contains two input fields: "Usuario" and "Contraseña". Below the "Contraseña" field is a link labeled "¿Olvidaste tu contraseña?" which is highlighted with a red box and a red arrow pointing to it. Below the link is a blue button labeled "Ingresar".

Imagen 9: Recuperar contraseña.

8. Seguidamente deberá ingresar el correo electrónico y hacer clic en **[Recuperar Contraseña]**.



The image shows a form titled "Recuperar Contraseña". It has a checkbox labeled "Correo electrónico" which is checked. Below the checkbox is an input field for the email address, highlighted with a red box and a red arrow pointing to it. Below the input field is a blue button labeled "Recuperar Contraseña", also highlighted with a red box and a red arrow pointing to it.

Imagen 10: Formulario recuperar contraseña.

9. Se presenta un mensaje de recuperación exitosa, hacer clic en **[Regresar al Inicio de Sesión]** para regresar a la página de inicio.

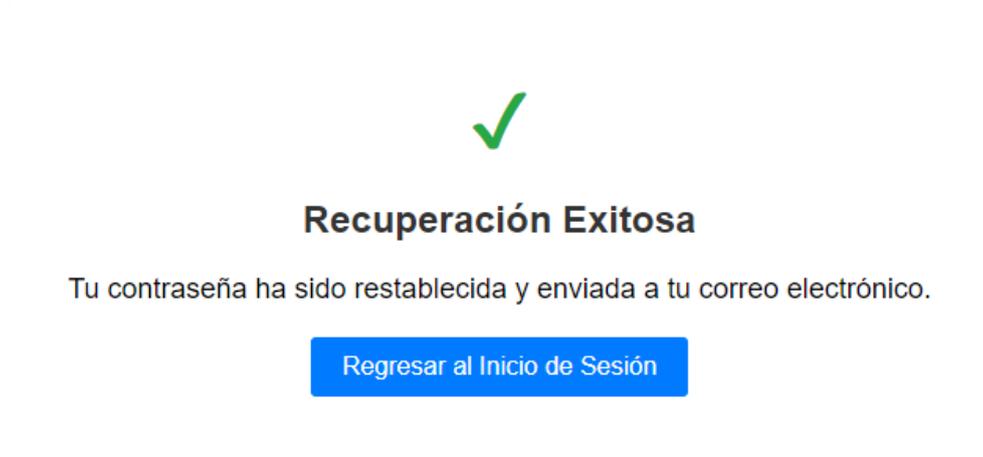


Imagen 11: Mensaje de confirmación.

10. Revisar el correo electrónico para consultar su nueva contraseña, hacer clic en **[Iniciar Sesión]**.

Recuperación de Contraseña

Hemos recibido una solicitud para restablecer la contraseña de su cuenta.
Su nueva contraseña generada es:

pM09Mm7MH3 ←

Le recomendamos cambiar su contraseña después de iniciar sesión.

Si no solicitó esta acción, se recomienda comunicarse con un administrador.

Iniciar Sesión ←

No responda a este correo electrónico. Si necesita ayuda, contáctenos en support@tusitio.com.

Imagen 12: Contraseña nueva.

11. Posterior el sistema le redirige al inicio de sesión, donde podrá ingresar con su nueva contraseña.

4. Cambiar contraseña

- En la página principal del paso 5 de acceder al sistema, hacer clic en **[Cambiar contraseña]** la parte superior derecha.



Imagen 13: Cambiar contraseña.

- Posteriormente se presenta un formulario, donde deberá ingresar la antigua clave y la nueva clave, hacer clic en **[Continue]**, para completar el proceso de modificación de la clave.

Imagen 14: Formulario para cambiar contraseña.

- Finalmente, el sistema le redirige a la página principal, junto con un mensaje de aviso, mencionando que su contraseña fue actualizada.



Imagen 15: Mensaje de aviso.

5. Análisis.

a. Realizar análisis de plagio

- Paso1: Cargar documento, hacer clic en [Análisis] en la página principal del paso 5 del acceso al sistema.



Imagen 16: Análisis de Plagio.

- Paso 2: Posteriormente hacer clic en [Seleccionar Archivo].



Imagen 17: Seleccionar documento.

- Paso 3: Seleccionar el documento a cargar en el sistema y luego hacer clic en **[Abrir]**.

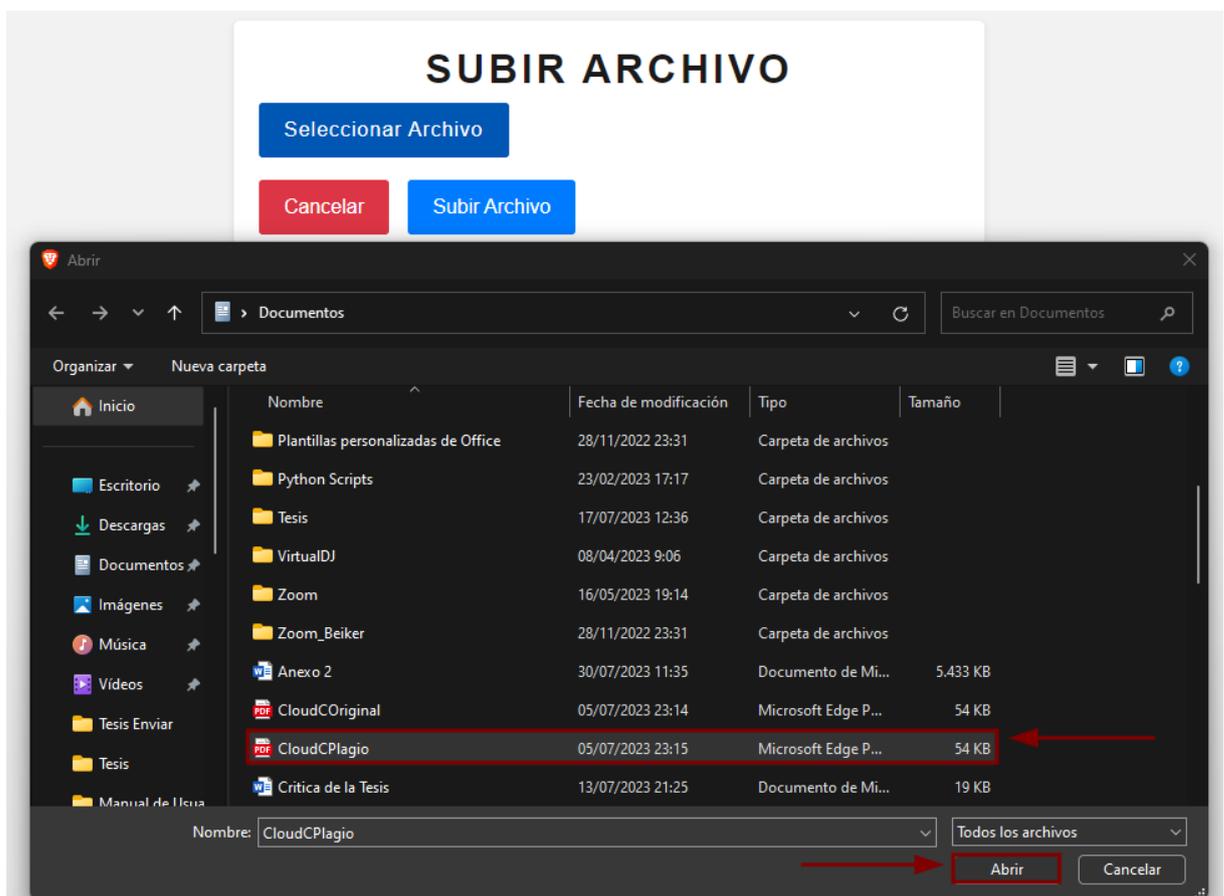


Imagen 18: Seleccionar documento.

- Paso 4: Seguidamente hacer clic en **[Subir Archivo]**.

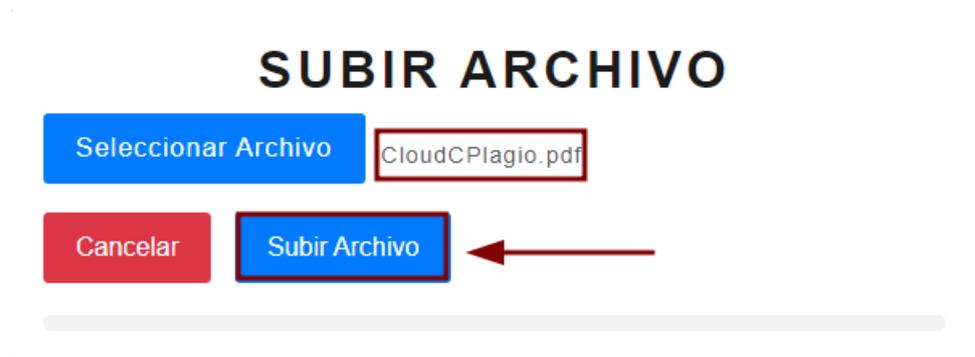


Imagen 19: Seleccionar documento.

- Paso 5: A continuación, hacer clic en **[Iniciar análisis]**.

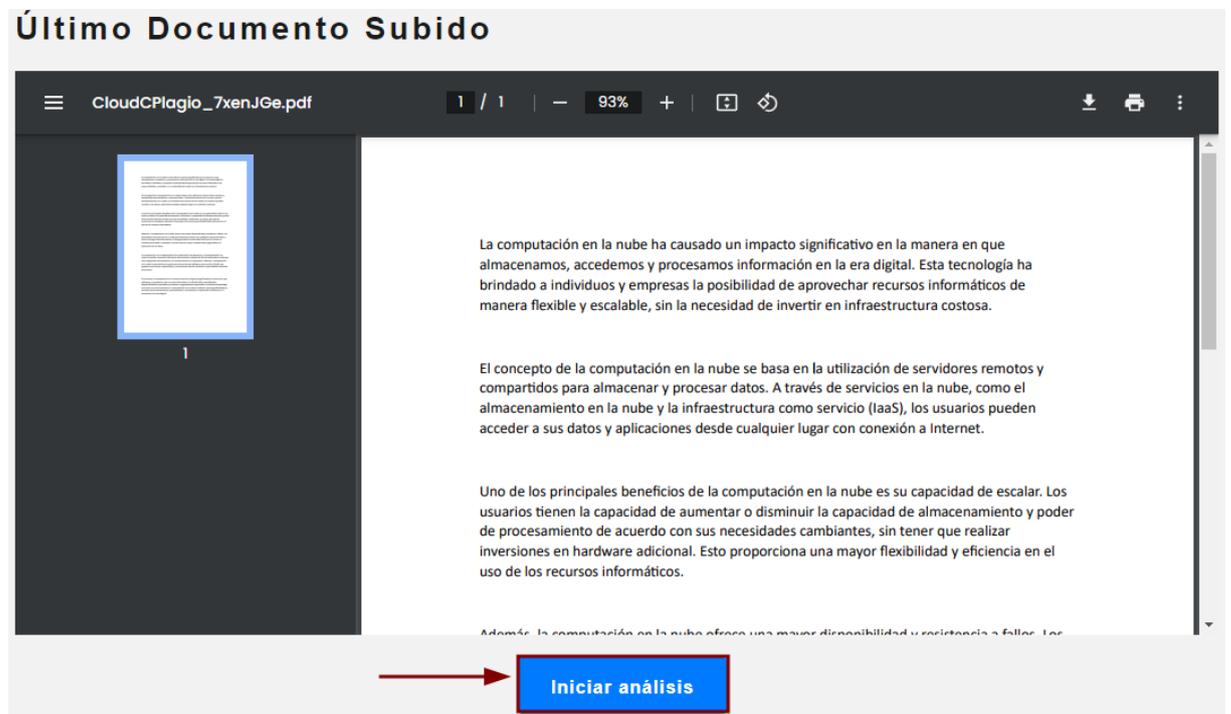


Imagen 20: Iniciar análisis.

- Seguidamente se presenta un formulario, donde deberá ingresar la información del docente, al cual le llegará el resultado de plagio, hacer clic en **[Enviar]** para empezar el análisis, en caso de cancelar esta acción hacer clic en **[Cancelar]**.

Envío Docente

Título

Descripción

Correo Electrónico

Enviar Cancelar

Imagen 21: Formulario para envío de resultado de plagio.

- Seguidamente se presenta un mensaje, para informar que ha empezado el análisis, así mismo se le informa que el resultado le llegara a su correo, hacer clic en **[Regresar]**, para volver a la página principal.

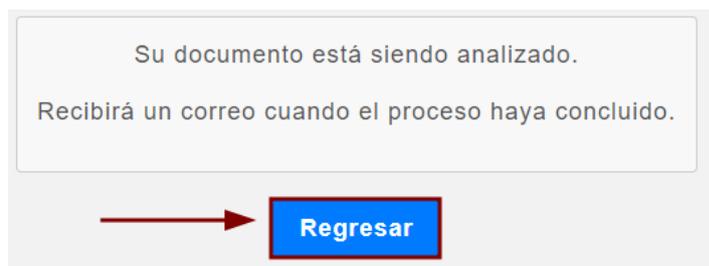


Imagen 22: Mensaje de aviso del análisis de plagio.

- Finalmente revisar el correo electro para verificar que el análisis a concluido con éxito.

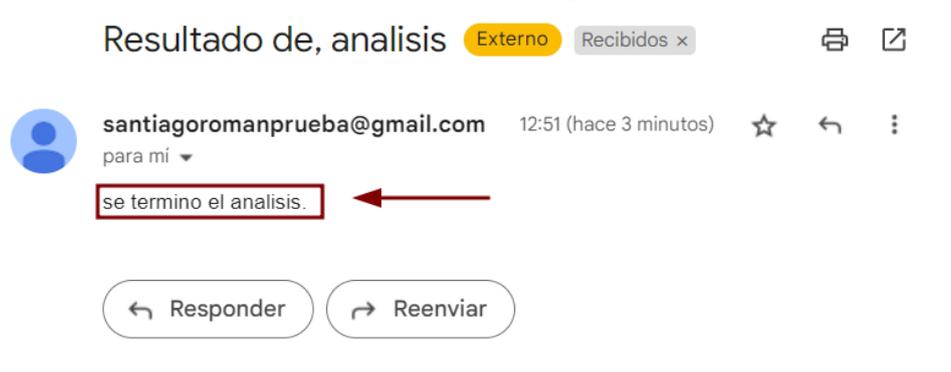


Imagen 23: Mensaje de confirmación de análisis de plagio.

b. Ver resultados de plagio.

- Paso 1: Para ver el resultado de plagio hacer clic en [**Resultados**], en la página principal del paso 5 del acceso al sistema.



Imagen 24: Resultado de plagio.

- Paso 2: A continuación, se presenta una lista de todos los resultados de plagio generados.

Numero	Documento	Titulo	Comentario	Subido Por	Acciones
1	leccion1_6flwWOZ.pdf	Proyecto PIC	Sección Introducción	oswaldo medina	Revision
2	leccionU02.pdf	PIC	Sección Resumen	oswaldo medina	Revision

Imagen 25: Lista de resultados de plagio.

- Paso 3: Seguidamente hacer clic en [**Revision**], para visualizar el resultado de plagio.

Numero	Documento	Titulo	Comentario	Subido Por	Acciones
1	leccion1_6flwWOZ.pdf	Proyecto PIC	Sección Introducción	oswaldo medina	Revision
2	leccionU02.pdf	PIC	Sección Resumen	oswaldo medina	Revision

Imagen 26: Revisión de resultado de plagio.

- Paso 4: Finalmente, se le presenta el resultado de plagio generado.

DOCUMENTO ANALIZADO EXITOSAMENTE

☰ (anonymous)
1 / 1 | 108% + | 📄 ↻

Análisis de plagio sobre:
leccion1_6flwWOZ.pdf

Análisis de plagio

Total de 0 plagios encontrados en 24 segundos
Porcentaje de plagio general: 0%

Oración plagiada	Oración original	Lugar donde se encontró	Ubicación

Compartir
Editar Detalles

Imagen 27: Resultado de plagio.

- c. **Compartir análisis de plagio.**
 - Repetir todos los pasos del **literal b** de la sección Análisis.
 - A continuación, en el paso 4 hacer clic en [**Compartir**].

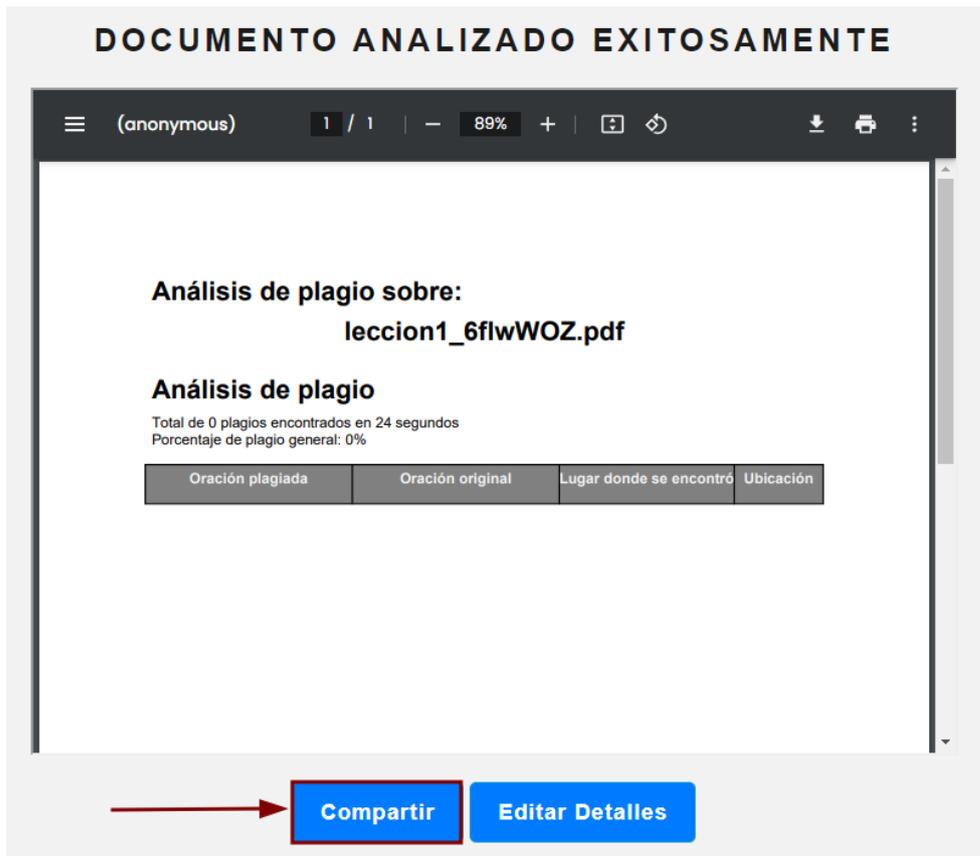


Imagen 28: Compartir resultado de plagio.

- Finalmente llenar los campos solicitados, y hacer clic en [Compartir].

Compartir Documento

Correo electrónico

+ ↑

Comentario

Compartir ← ↑

Imagen 29: Formulario compartir resultado de plagio.

6. Firmas de Responsabilidad

Acción	Funcionario	Firma
Elaborado por:	Jaime Oswaldo Paqui Medina Tesista	
	Santiago Alexander Román Silva Tesista	
Revisado por:	Oscar Miguel Cumbicus Pineda Cliente del Software	
Aprobado por:	Oscar Miguel Cumbicus Pineda Cliente del Software	

Anexo 11. Manual de usuario del Docente.

Sistema de anti-Plagio para la carrera de Computación de la Universidad Nacional de Loja

Elaborado por:

Jaime Oswaldo Paqui Medina

Santiago Alexander Román Silva

Revisado por:

Ing. Oscar Miguel Cumbicus Pineda

Aprobado por:

Ing. Oscar Miguel Cumbicus Pineda

Mayo 2023

TABLA DE CONTENIDO

Historial de versiones	222
Información del Trabajo.....	222
1. Introducción.....	223
2. Registrarse.....	223
3. Acceso al Sistema.....	225
4. Cambiar contraseña	229
5. Gestión de documentos.....	231
6. Firmas de Responsabilidad	240

Historial de versiones

Versión	Fecha	Responsable	Descripción de la modificación
1	06-04-2023	Ing. Oscar Miguel Cumbicus	Documento inicial
2	11-04-2023	Ing. Oscar Miguel Cumbicus	Actualizar las capturas de pantalla

Información del Trabajo

Tabla 1. Trabajo para la UNL.

Nombre del Trabajo	Sistema de anti-Plagio para la carrera de Computación de la Universidad Nacional de Loja.
Fecha de elaboración	01-04-2023
Solicitante / Patrocinador principal	Carrera de Computación - Universidad Nacional de Loja

1. Introducción

En el presente documento se tiene la documentación adecuada para guiar a los usuarios en el uso adecuado del **Sistema Antiplagio**, pone a disposición del lector, el siguiente documento donde constan las diversas funcionalidades que tiene el sistema para el rol de Docente, con el fin de que puedan agregar usuarios, modificar usuarios, validar registro de usuarios, cambiar rol de los usuarios, entre otros.

2. Registrarse

1. Abrir el navegador y dirigir <https://Antiplagiodocker.azurewebsites.net/>
2. A continuación, en la parte superior derecha hay un menú denominado **[Registrar]** hacer clic para registrarse.

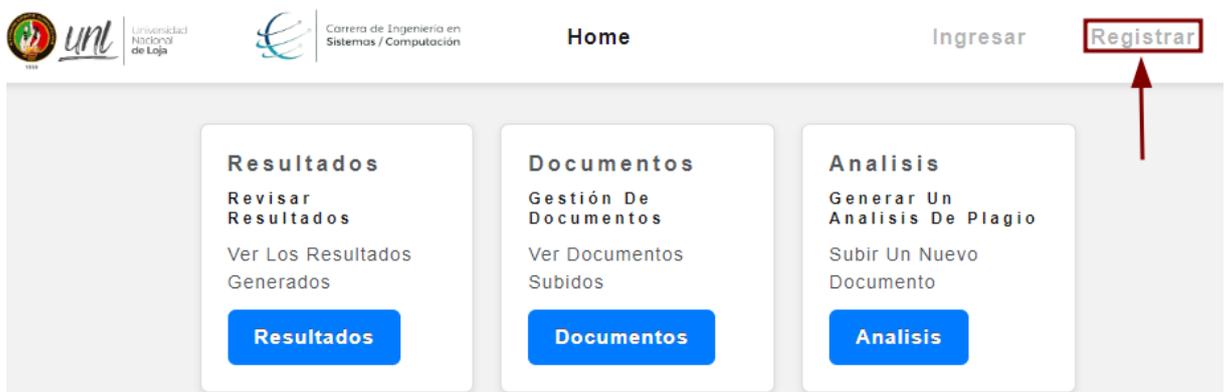


Imagen 1: Página principal.

3. Posteriormente se presenta un formulario, donde deberá ingresar los datos solicitados, hacer clic en **[Continuar]** una vez ingresado los datos. Para la contraseña tener en consideración los mensajes de aviso.

Crear Una Cuenta

 Nombre

 Apellido

 Usuario

 Email

 Contraseña

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

Continuar

Imagen 2: Formulario de registro.

4. A continuación, hacer clic en **[Seleccionar]** en el rol de docente, para completar el proceso de registro.



Imagen 3: Rol para el registro.

5. Posteriormente se presenta un mensaje, donde se menciona que el proceso se ha completado y de que está a la espera de que le validen su cuenta.



Imagen 4: Mensaje de aviso.

6. Revisar su correo electrónico para verificar si el administrador le validó la cuenta.

3. Acceso al Sistema

1. Abrir el navegador y dirigir <https://Antiplagiodocker.azurewebsites.net/>
2. A continuación, en la parte superior derecha hay un menú denominado **[Ingresar]** hacer clic para ingresar.

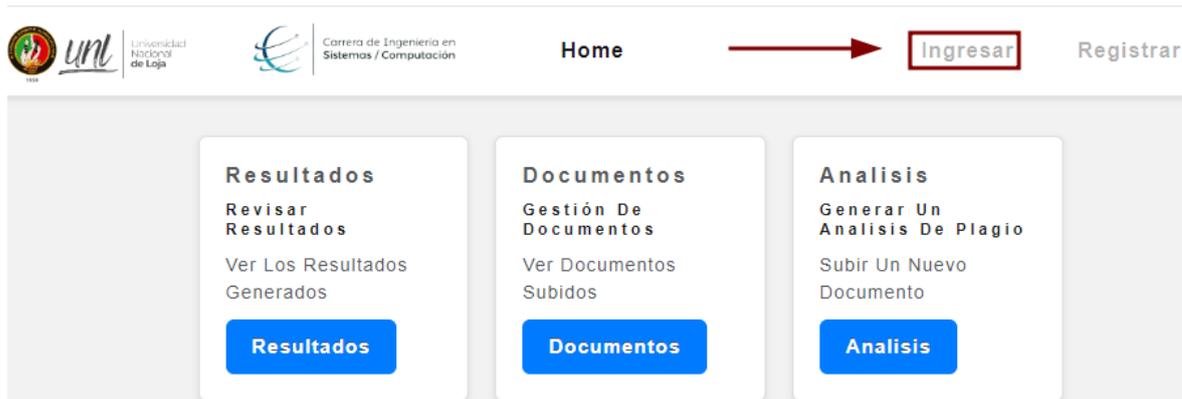


Imagen 5: Página principal.

3. Posteriormente el sistema solicitará ingresar las credenciales de acceso.

Imagen 6: Formulario para ingresar credenciales.

4. A continuación, ingresar credenciales en el formulario y hacer clic en el botón **[Ingresar]**.

Imagen 7: Ingresar al sistema.

5. Posterior se le presenta la página principal.



Imagen 8: Página principal

6. En caso de haber olvidado su contraseña podrá recuperar, para ello repetir el paso 1, 2 y 3.

7. A continuación, hacer clic en **[Olvidaste tu contraseña]**.

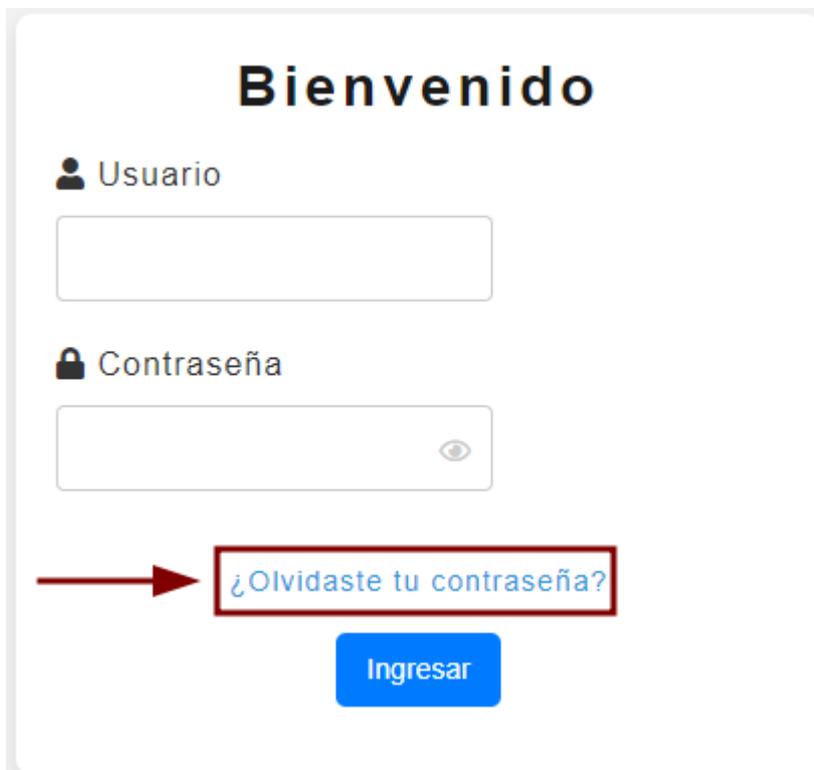


Imagen 9: Recuperar contraseña.

8. A continuación, ingresar el correo electrónico y hacer clic en **[Recuperar Contraseña]**.



Recuperar Contraseña

Correo electrónico

[Recuperar Contraseña](#)

Detailed description: This is a password recovery form. At the top, it has the title 'Recuperar Contraseña'. Below the title is a checked checkbox labeled 'Correo electrónico'. Underneath is a text input field. Below the input field is a blue button with the text 'Recuperar Contraseña'. Two red arrows point to the input field and the button from the right side.

Imagen 10: Formulario recuperar contraseña.

9. Posterior se presenta un mensaje de recuperación exitosa, hacer clic en **[Regresar al Inicio de Sesión]** para regresar a la página de inicio.

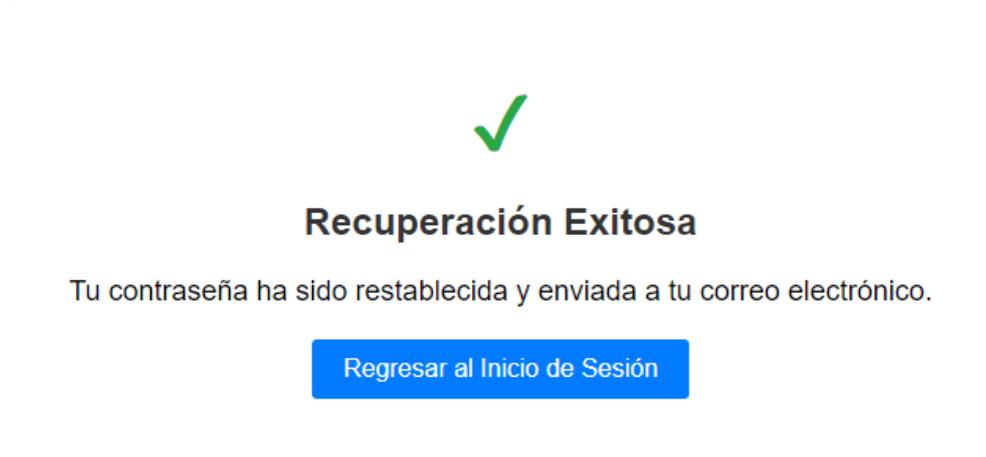


Imagen 11: Mensaje de confirmación.

10. Revisar el correo electrónico para consultar su nueva contraseña, hacer clic en **[Iniciar Sesión]**.

Recuperación de Contraseña

Hemos recibido una solicitud para restablecer la contraseña de su cuenta.
Su nueva contraseña generada es:

pM09Mm7MH3

Le recomendamos cambiar su contraseña después de iniciar sesión.

Si no solicitó esta acción, se recomienda comunicarse con un administrador.

Iniciar Sesión

No responda a este correo electrónico. Si necesita ayuda, contáctenos en support@tusitio.com.

Imagen 12: Contraseña nueva.

11. Posterior el sistema le redirige al inicio de sesión, donde podrá ingresar con su nueva contraseña.

4. Cambiar contraseña

- En la página principal del paso 5 de acceder al sistema, hacer clic en **[Cambiar contraseña]** la parte superior derecha.

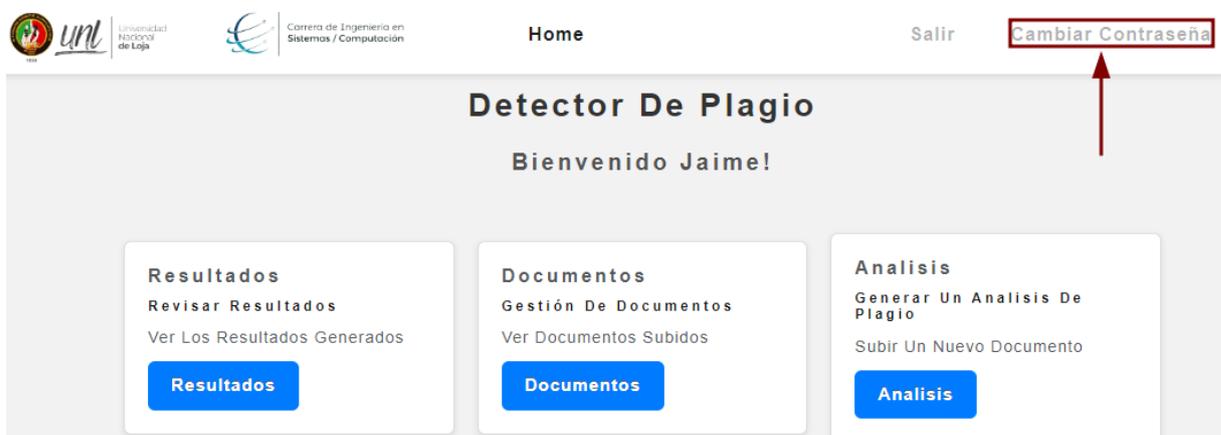


Imagen 13: Cambiar contraseña.

- Posteriormente se presenta un formulario, donde deberá ingresar la antigua clave y la nueva clave, hacer clic en **[Continue]**, para completar el proceso de modificación de la clave.

Imagen 14: Formulario para cambiar contraseña.

- Finalmente, el sistema le redirige a la página principal, junto con un mensaje de aviso, mencionando que su contraseña fue actualizada.

Imagen 10: Mensaje de aviso.

5. Análisis.

a. Realizar análisis de plagio

- Paso1: Cargar documento, hacer clic en **[Documentos]** en la página principal del paso 5 del acceso al sistema.



Imagen 15: Documentos.

- Paso 2: Posteriormente hacer clic en **[Seleccionar Archivo]**.



Imagen 16: Seleccionar documento.

- Paso 3: Seleccionar el documento a cargar en el sistema y luego hacer clic en **[Abrir]**.

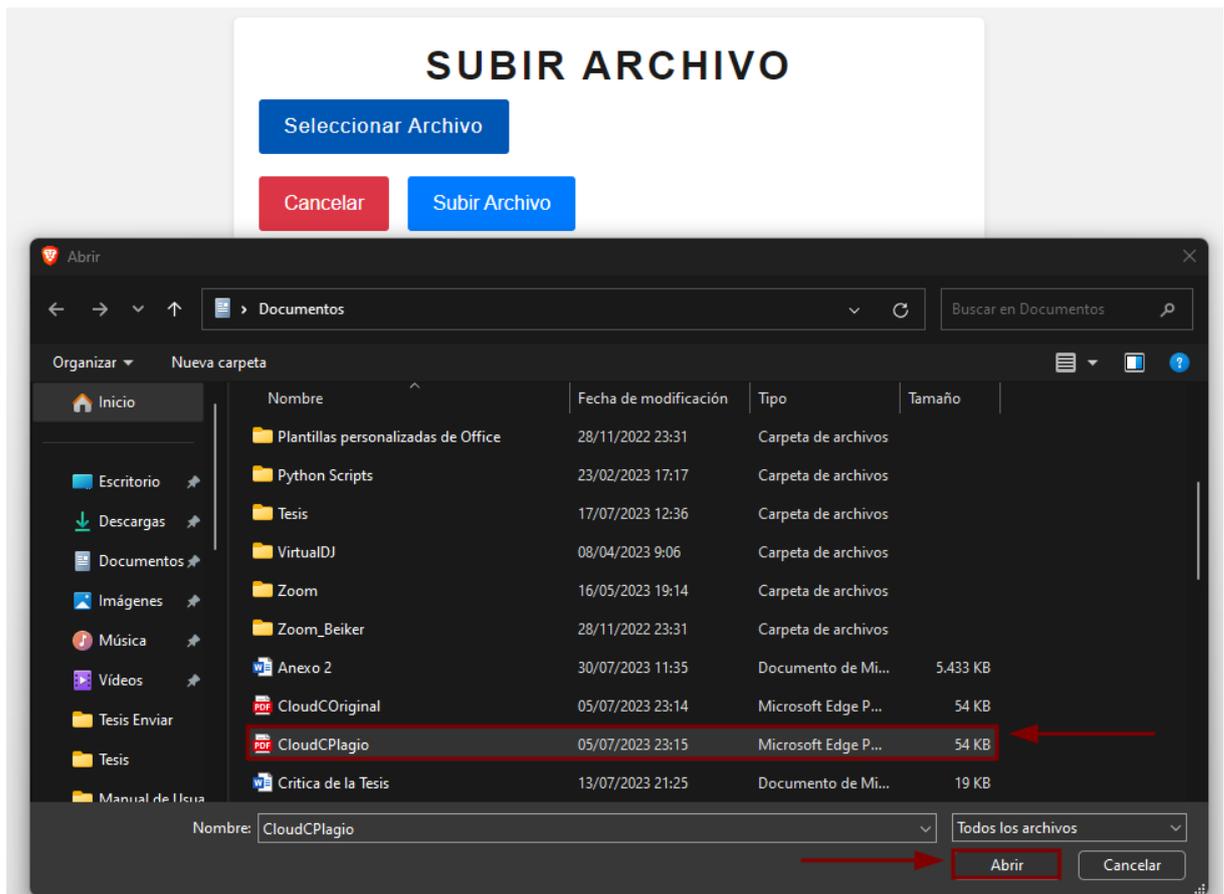


Imagen 17: Seleccionar documento.

- Paso 4: Seguidamente hacer clic en [Subir Archivo].

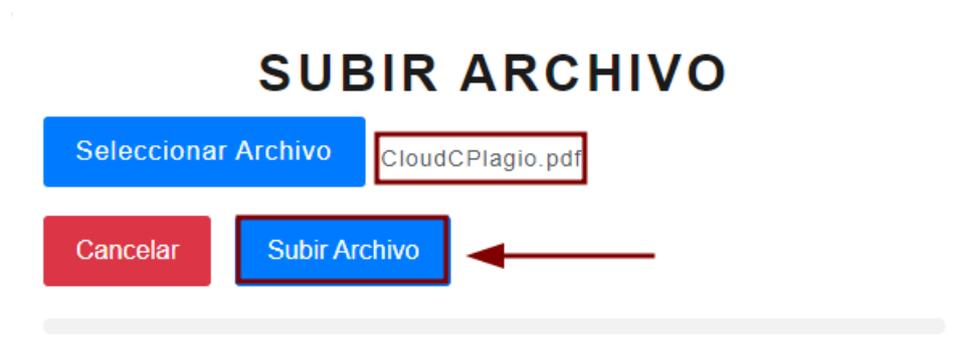


Imagen 18: Seleccionar documento.

- Paso 5: A continuación, hacer clic en [Iniciar análisis].

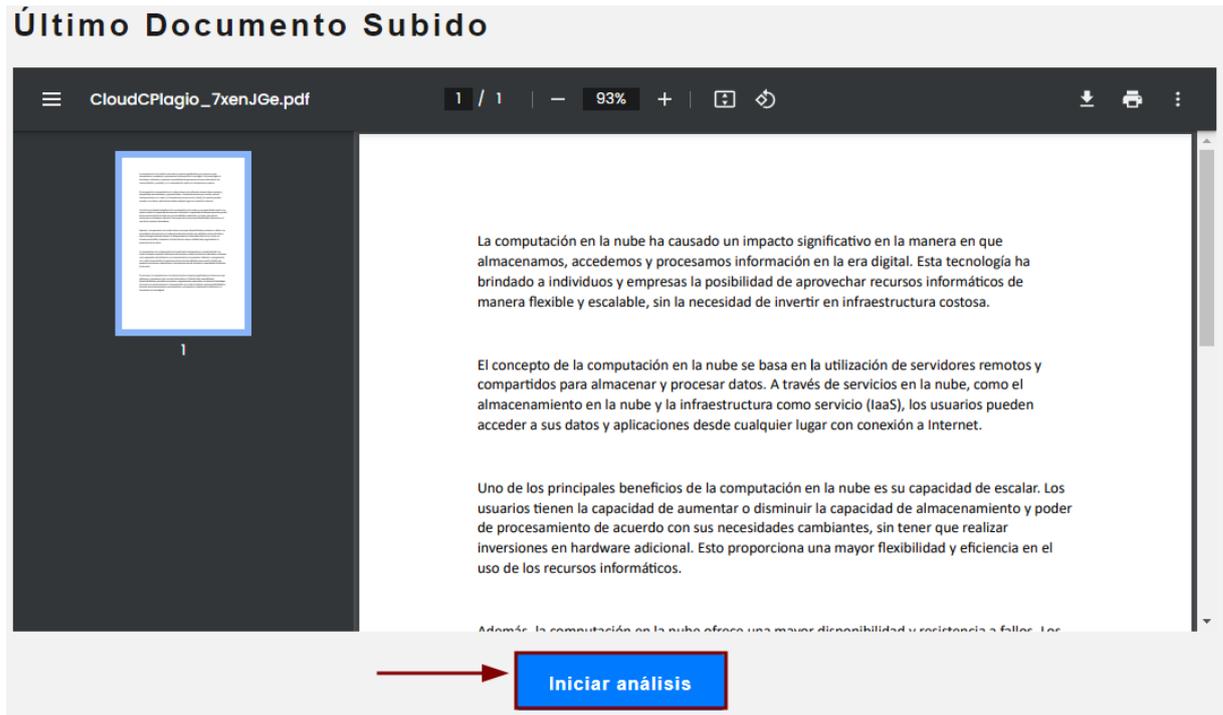


Imagen 19: Iniciar análisis.

- Seguidamente se presenta un mensaje, para informar que ha empezado el análisis, así mismo se le informa que el resultado le llegara a su correo, hacer clic en **[Regresar]**, para volver a la página principal.

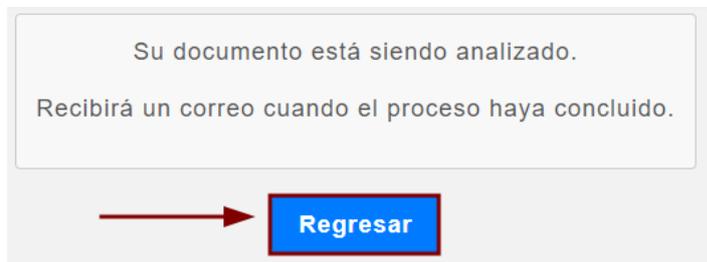


Imagen 20: Mensaje de aviso del análisis de plagio.

- Posteriormente revisamos el correo para validar si se generó el resultado de plagio.

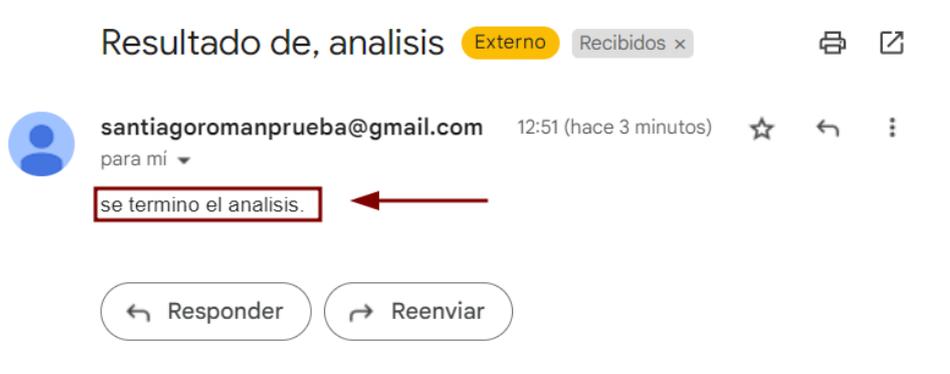


Imagen 21: Correo de verificación.

b. Ver resultados de plagio.

- Paso 1: Para ver el resultado de plagio hacer clic en **[Resultados]**, en la página principal del paso 5 del acceso al sistema.



Imagen 22: Resultado de plagio.

- Paso 2: A continuación, se presenta una lista de todos los resultados de plagio generados, para visualizar el resultado de plagio hacer clic en **[Revision]**.

Numero	Documento	Subido Por	Revision
1	Introducción_2_XeZokhk.pdf	jaimo jaimo	Revision Eliminar

Imagen 23: Ver resultado de plagio.

- Paso 3: Finalmente se presenta el resultado de plagio generado.



Imagen 24: Resultado de plagio.

c. **Compartir resultado de plagio.**

- Realiza todos los pasos de la sección ver resultado de plagio, dentro del paso 3 hacer clic en [Compartir].



Imagen 25: Compartir resultado de plagio.

- Seguidamente se presenta un formulario, ingresar los datos correspondientes y hacer clic en [Compartir].

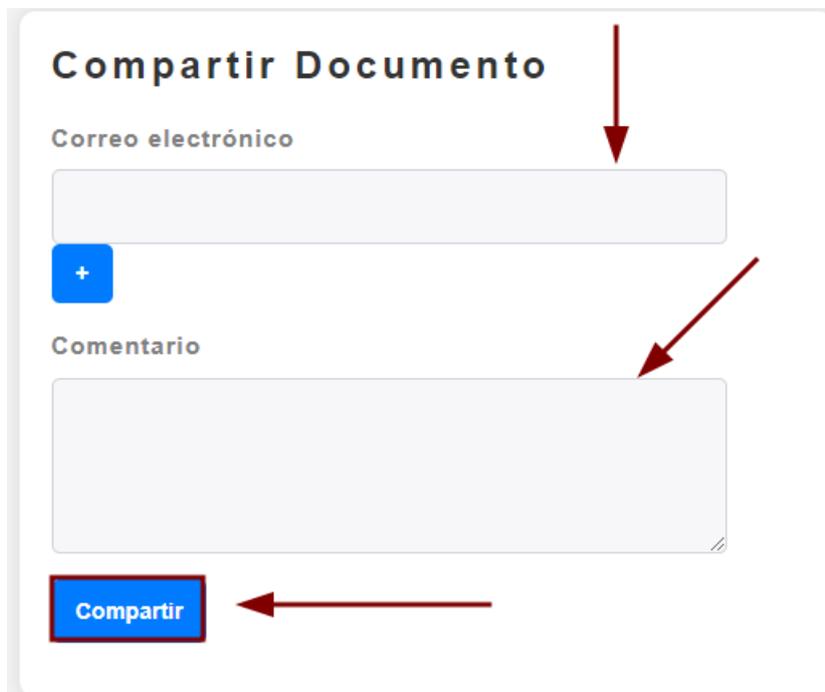


Imagen 26: Compartir resultado de plagio.

- Posteriormente verificar si el proceso se llevó a cabo con éxito y hacer clic sobre el documento compartido.

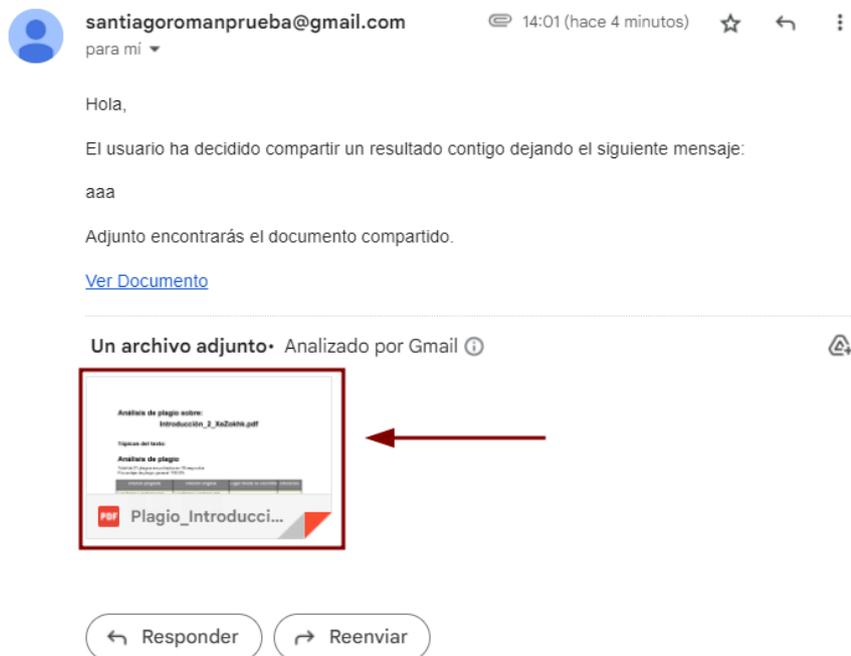


Imagen 27: Resultado compartido.

- Finalmente se puede visualizar el documento compartido.

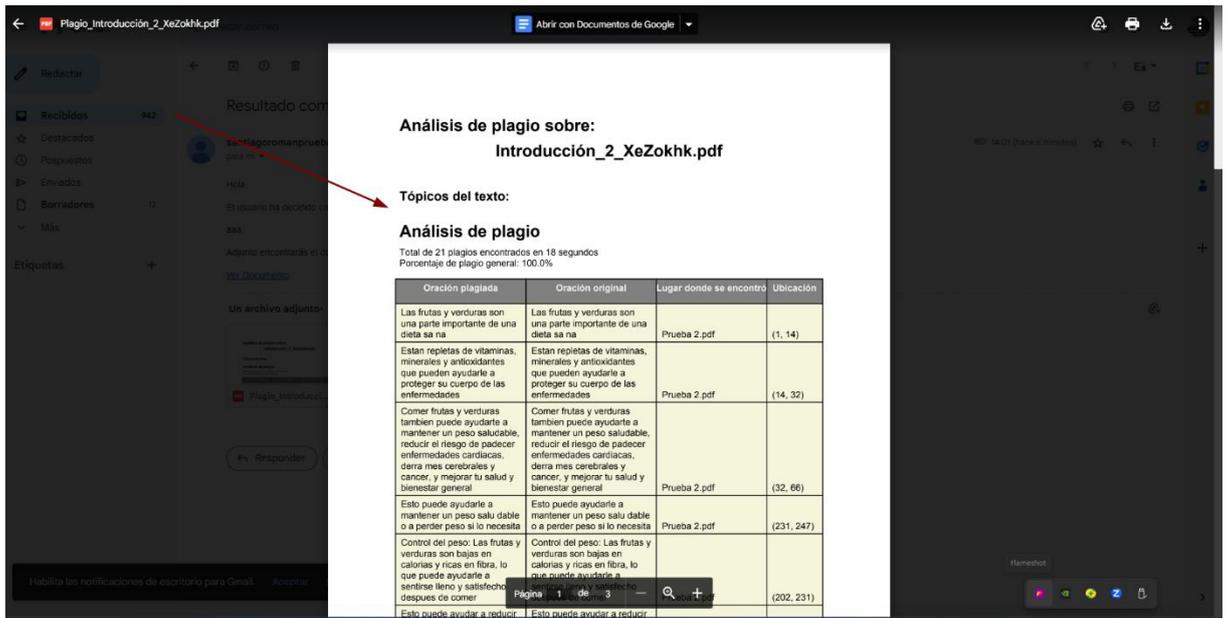


Imagen 28: Resultado compartido.

d. Editar resultado de plagio

- Realiza todos los pasos de la sección ver resultado de plagio, dentro del paso 3 hacer clic en [Editar Detalles].



Imagen 29: Editar detalle.

- Seguidamente se presenta una tabla detallando todos los párrafos plagiados, hacer clic en **[Eliminar]**, para reducir el resultado de plagio.

TABLA DE PLAGIOS

Porcentaje de plagio: 18.75%

num oraciones del documento: 16

ORACIÓN PLAGIADA	ORACIÓN ORIGINAL	PORCENTAJE DE PLAGIO	LUGAR DONDE SE ENCONTRÓ	UBICACIÓN	ACCIONES
Ademas, la computacion en la nube ha permitido el surgimiento de servicios de software como servicio (SaaS), que proporcionan acceso a aplicaciones y herramientas de alta calidad sin necesidad de instalarlas localmente	Ademas, la computacion en la nube ha permitido el surgimiento de servicios de software como servicio (SaaS), que proporcionan acceso a aplicaciones y herramientas de alta calidad sin necesidad de instalarlas localmente	100,00%	CloudCPlagio.pdf	(280, 317)	ELIMINAR ↑
Su flexibilidad, escalabilidad y disponibilidad han permitido a personas y organizaciones aprovechar al maximo la tecnologia sin incurrir en costos excesivos	Su flexibilidad, escalabilidad y disponibilidad han permitido a personas y organizaciones aprovechar al maximo la tecnologia sin incurrir en costos excesivos	100,00%	CloudCPlagio.pdf	(342, 364)	ELIMINAR

Imagen 30: Reducir el porcentaje de plagio.

- Finalmente hacer clic en **[Confirmar cambios]** para efectuar la acción, caso contrario hacer clic en **[Descartar cambios]** para cancelar la acción.

TABLA DE PLAGIOS

Porcentaje de plagio: 6.25%

num oraciones del documento: 16

ORACIÓN PLAGIADA	ORACIÓN ORIGINAL	PORCENTAJE DE PLAGIO	LUGAR DONDE SE ENCONTRÓ	UBICACIÓN	ACCIONES
En resumen, la computacion en la nube ha tenido un impacto significativo en la forma en que utilizamos y accedemos a los recursos informaticos	En resumen, la computacion en la nube ha tenido un impacto significativo en la forma en que utilizamos y accedemos a los recursos informaticos	100,00%	CloudCPlagio.pdf	(317, 342)	ELIMINAR

CONFIRMAR CAMBIOS **DESCARTAR CAMBIOS**

Imagen 31: Reducir el porcentaje de plagio.

e. Eliminar resultado de plagio

- En el paso 2 de la sección ver resultado de plagio, hacer clic en **[Eliminar]**.

Numero	Documento	Subido Por	Revision	
1	Introducción_2_XeZokhk.pdf	jaimo jaimo	Revision	Eliminar

Imagen 32: Eliminar resultado de plagio.

- A continuación, se presenta un mensaje de confirmación para eliminar el resultado de plagio, hacer clic en **[Eliminar Resultado]** para ejecutar la acción eliminar, caso contrario hacer clic en **[Cancelar]** para rechazar la acción.

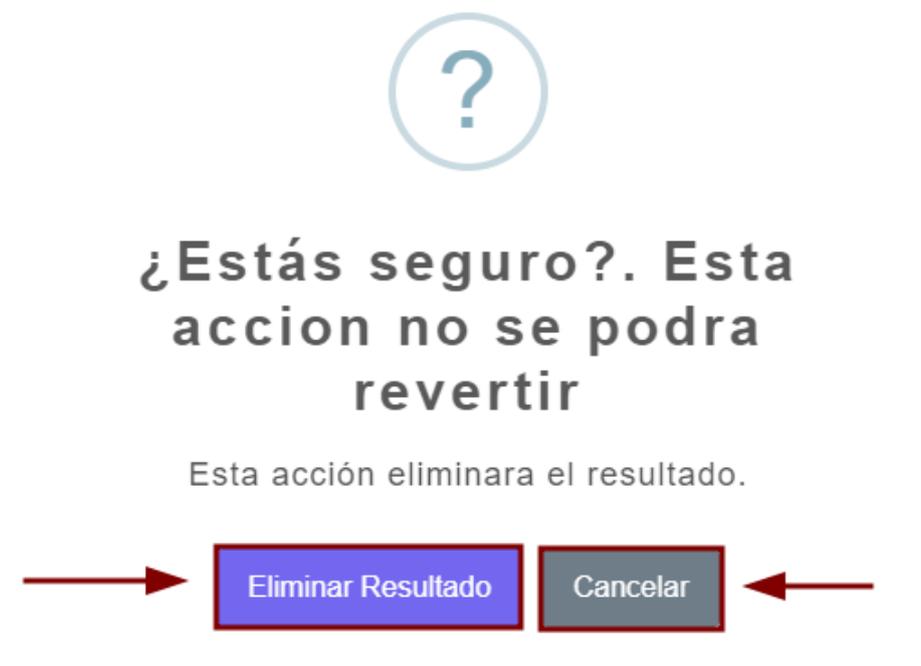


Imagen 33: Eliminar resultado de plagio.

6. Firmas de Responsabilidad

Acción	Funcionario	Firma
Elaborado por:	Jaime Oswaldo Paqui Medina Tesista	
	Santiago Alexander Román Silva Tesista	
Revisado por:	Oscar Miguel Cumbicus Pineda Cliente del Software	
Aprobado por:	Oscar Miguel Cumbicus Pineda Cliente del Software	

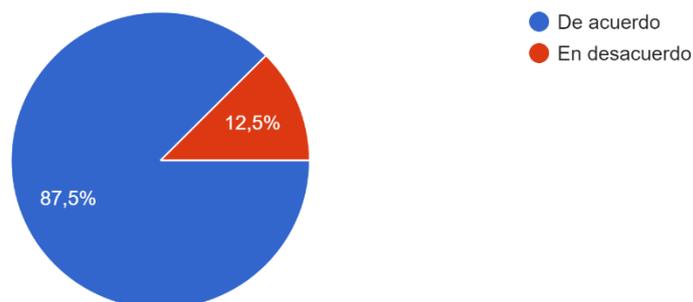
Anexo 12. Evaluación de la experiencia de usuario en el uso del sistema **Antiplagio**.

Facilidad de uso (FU)

Pregunta 1:

¿Cree usted que el proceso de registro o inicio de sesión en el sistema Antiplagio es sencillo y sin complicaciones?

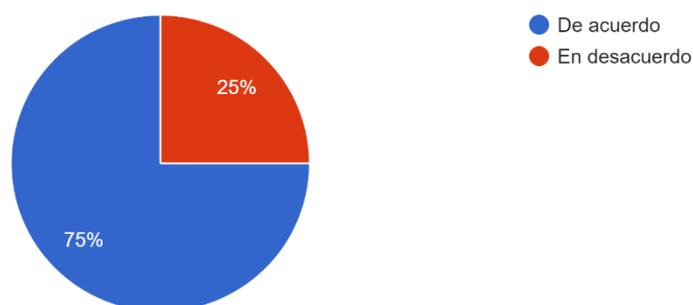
8 respuestas



Pregunta 2:

¿ Cree usted que es sencillo cargar sus documentos en el sistema Antiplagio para realizar el análisis de plagio?

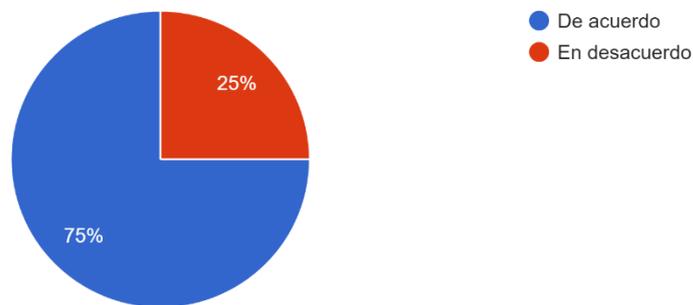
8 respuestas



Pregunta 3:

¿Le resulta sencillo navegar por las diferentes secciones dentro del sistema Antiplagio?

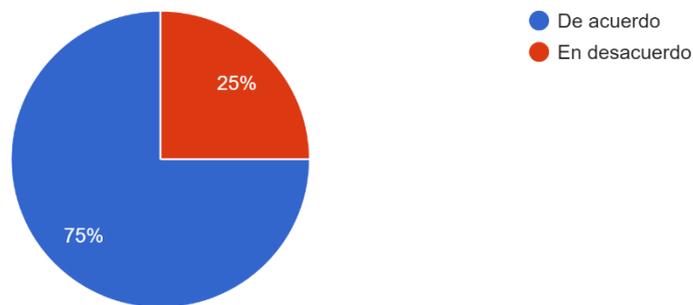
8 respuestas



Pregunta 4:

¿Encuentras que la interfaz del sistema Antiplagio es fácil de entender y utilizar?

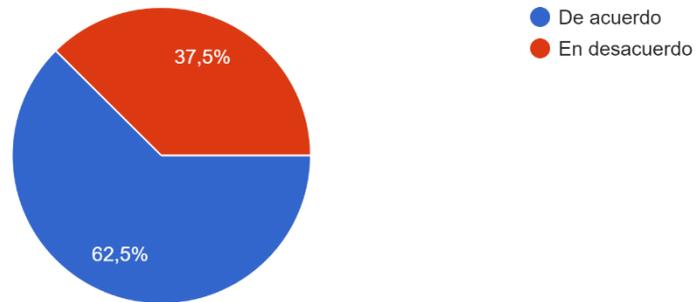
8 respuestas



Pregunta 5:

¿ Cree usted que el sistema Antiplagio le permite realizar el análisis de plagio con pocos pasos y sin complicaciones innecesarias?

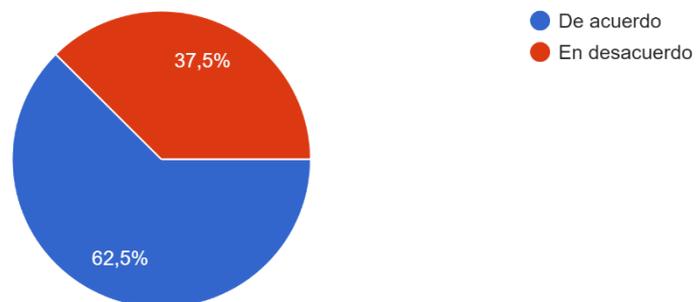
8 respuestas



Pregunta 6:

¿Le resulta cómodo interpretar los resultados de plagio generados por el sistema Antiplagio?

8 respuestas



Utilidad Percibida (PU)

Pregunta 1:

¿ Cree usted que el sistema Antiplagio le será útil para identificar posibles casos de plagio en los trabajos académicos?

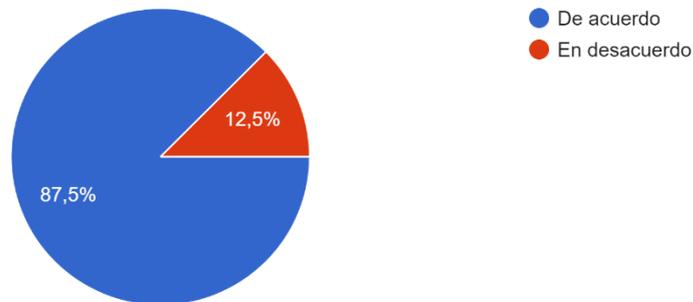
8 respuestas



Pregunta 2:

¿Cree usted que el sistema Antiplagio te proporcionará información útil y detallada sobre las fuentes o citas que podrían estar relacionadas con el plagio?

8 respuestas



Pregunta 3:

¿ Cree usted que el sistema Antiplagio es una herramienta accesible y adecuada para estudiantes y profesores en la Carrera de Computación?

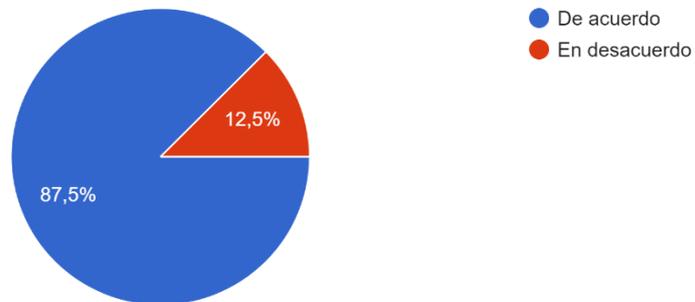
8 respuestas



Pregunta 4:

¿Considera que el acceso gratuito al sistema Antiplagio es una ventaja significativa en comparación con otras herramientas de detección de plagio de pago?

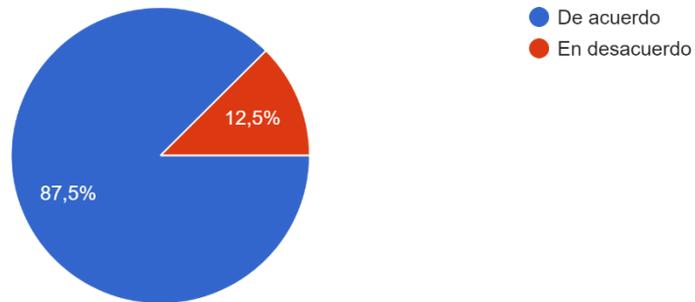
8 respuestas



Pregunta 5:

¿ Cree usted que el sistema Antiplagio ofrece un análisis de resultados claro y comprensible para ayudarte a mejorar la calidad de tus trabajos académicos?

8 respuestas



Pregunta 6:

¿ Cree usted que el sistema Antiplagio es una herramienta valiosa para la carrera de computación?

8 respuestas



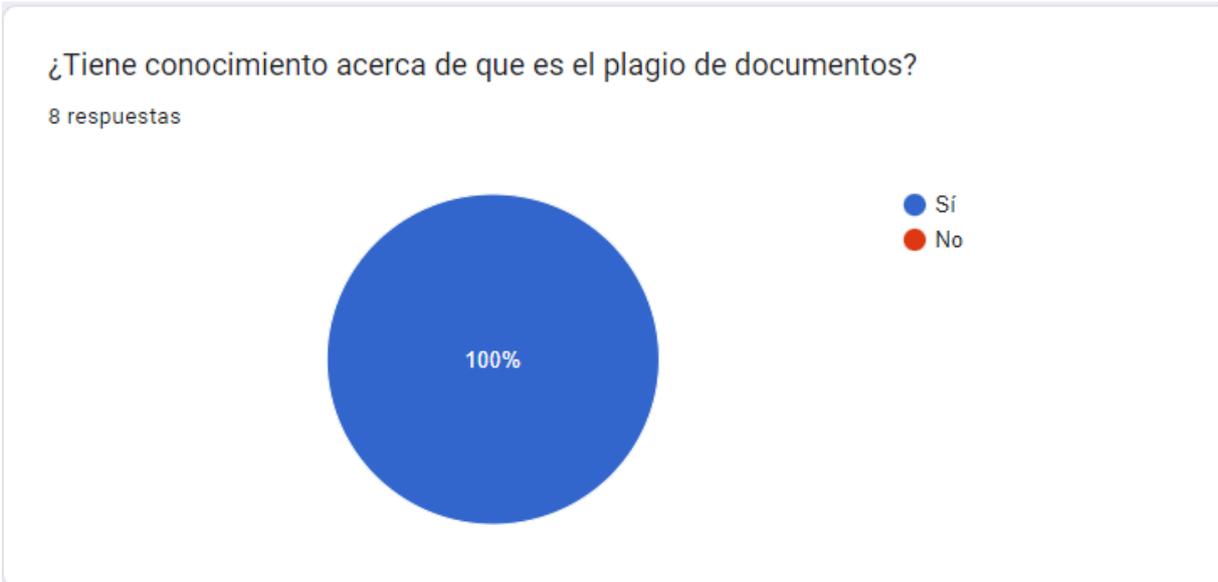
Anexo 13. Formulario con las respectivas preguntas, para realizar la encuesta correspondiente.

Usuarios que han respondido

Correo electrónico

- pfordonez@unl.edu.ec
- wpchamba@unl.edu.ec
- jose.o.guaman@unl.edu.ec
- genoveva.suing@unl.edu.ec
- andres.navas@unl.edu.ec
- maria.ruilova.s@unl.edu.ec
- fjalvarez@unl.edu.ec
- mario.a.zambrano@unl.edu.ec

Pregunta 1:

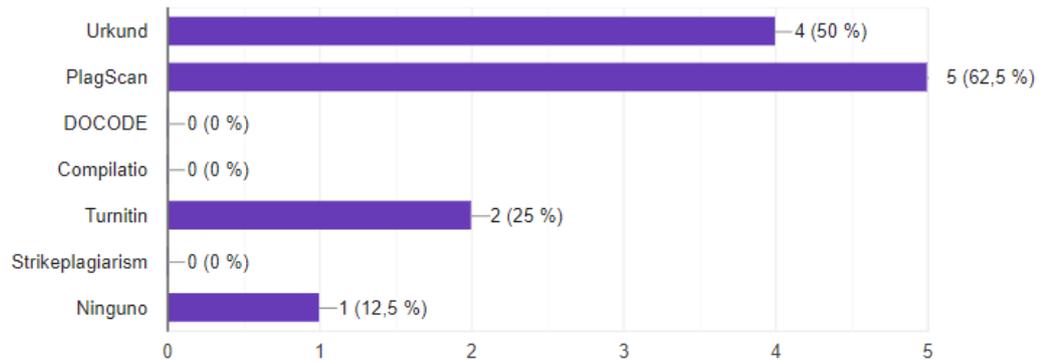


Pregunta 2:

¿Cuál de las siguientes herramientas para el control del plagio conoce

 Copiar

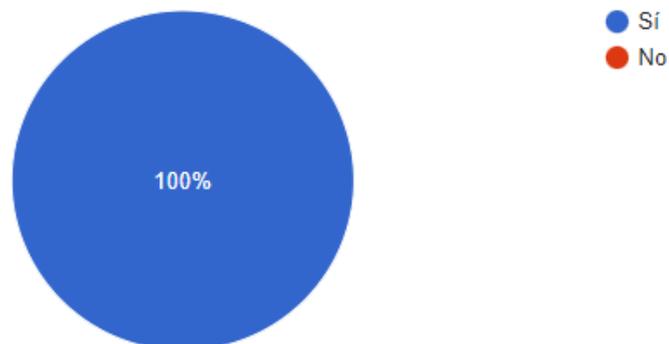
8 respuestas



Pregunta 3:

¿Cree que detectar plagio en un trabajo, es beneficioso para el progreso de la carrera?

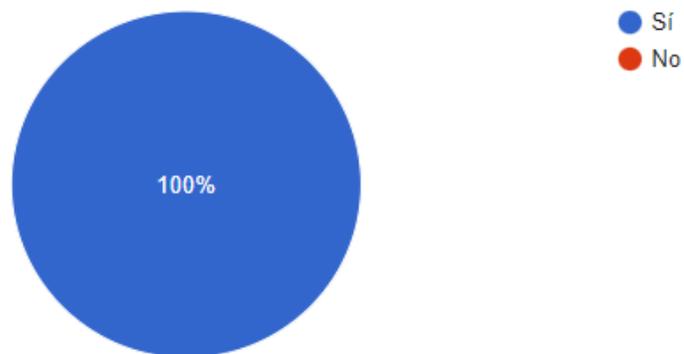
8 respuestas



Pregunta 4:

¿Le gustaría a usted que se cree un software anti-plagio de la carrera, que sea gratuito?

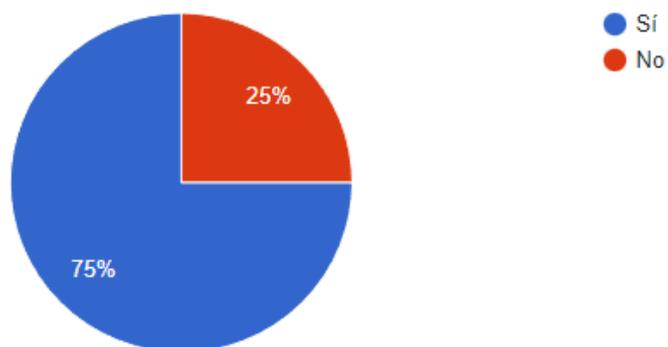
8 respuestas



Pregunta 5:

¿Piensa que sería bueno que la carrera utilice un software anti-plagio propio de la carrera?

8 respuestas



Anexo 14. Documento de Arquitectura de software con el modelo 4+1.

Arquitectura de software 4+1

Proyecto: Prototipo de Software anti plagio para la carrera de Ingeniería en Computación de la Universidad Nacional de Loja, utilizando redes neuronales.

Versión: 1.0

Fecha: 05/10/2023

Ficha del documento

Versión	Fecha de revisión	Responsables	Descripción de modificación
1.0	05/10/2023	Jaime Oswaldo Paqui Medina y Santiago Alexander Román Silva.	N/A

Índice de Contenidos

Ficha del documento	251
Índice de Contenidos.....	252
Índice de Figuras.....	254
Índice de Tablas.....	255
1. Introducción.....	256
1.1. Propósito	256
1.2. Alcance	256
1.3. Definiciones, acrónimos y abreviaturas	256
1.4. Personal involucrado	256
1.5. Resumen.....	257
2. Representación Arquitectónica.....	257
2.1. Escenarios	257
2.2. Vista Lógica.....	258
2.3. Vista de Procesos.....	258
2.4. Vista de Desarrollo	258
2.5. Vista Física	258
3. Metas y limitaciones arquitectónicas.....	259
3.1. Del lado del servicio:	259
3.2. Del lado del cliente:.....	259
3.3. Seguridad:.....	259
3.4. Persistencia:	259
3.5. Confiabilidad / Disponibilidad:	259
3.6. Performance:.....	260
3.7. Portabilidad y Reutilización:.....	260
3.8. Herramientas de Desarrollo:	260
4. Vistas	260
4.1. Vista de Escenarios.....	260
4.1.1. Casos de uso	260
4.1.2. Descripción de cada caso de uso.....	263
4.2. Vista Lógica.....	268
4.2.1. Diagrama de Clases.....	268
4.2.2. Diagrama de Secuencia	269
4.3. Vista de Procesos	270
4.3.1. Diagrama de Actividades	272

7. Índice de Figuras

Figura 1. Rol Usuario	261
Figura 2. Rol Administrador	261
Figura 3. Docente.....	262
Figura 4. Estudiante	263
Figura 5. Diagrama de clases general	268
Figura 6. Diagrama de secuencia Registrarse.....	269
Figura 7. Diagrama de secuencia Inicio de Sesión.....	270
Figura 8. Diagrama de Secuencia Analizar plagio.....	271
Figura 8 Diagrama de actividades Registrarse.....	272
Figura 9. Diagrama de actividades Iniciar Sesión.....	273
Figura 10. Diagrama de actividades Analizar plagio.....	274
Figura 11. Diagrama de paquetes	275
Figura 12. Diagrama de despliegue.....	276

8. Índice de Tablas

Tabla 1. Definiciones, acrónimos y abreviaturas.....	256
Tabla 2. Personal involucrado Estudiante de la CIS.	256
Tabla 3. Personal involucrado Estudiante de la CIS.	257
Tabla 4. Personal involucrado Docente de la CIS.....	257
Tabla 5. Requisito funcional Registrar usuarios.....	263
Tabla 6. Requisito funcional Iniciar sesión.....	264
Tabla 7. Requisito funcional Validar registro de Usuarios.....	264
Tabla 8. Requisito funcional Ver lista de usuarios.	264
Tabla 9. Requisito funcional Cambiar rol de cuentas de usuario.	264
Tabla 10. Requisito funcional Editar usuarios.....	265
Tabla 11. Requisito funcional Eliminar usuarios.	265
Tabla 12. Requisito funcional Analizar plagio por el Estudiante.....	265
Tabla 13. Requisito funcional Analizar plagio por el Docente.	266
Tabla 14. Requisito funcional Compartir resultados de análisis de plagio.....	266
Tabla 15. Requisito funcional Ver listado de documentos analizados por el usuario.	266
Tabla 16. Requisito funcional Ver resultados de análisis de plagio.....	266
Tabla 17. Requisito funcional Editar resultados de análisis de plagio.....	267
Tabla 18. Requisito funcional Eliminar resultados de análisis de plagio.	267
Tabla 19. Requisito funcional Descargar resultados de análisis de plagio.....	267

1. Introducción

Este documento presenta la Arquitectura de Software diseñada para el Sistema Antiplagio, un prototipo de software destinado a la detección de plagio en documentos TIC de la Carrera de Computación de la UNL. La estructura de esta arquitectura se encuentra condensada en diversas vistas y componentes, los cuales detallan cada funcionalidad del software, siguiendo el enfoque del modelo 4 + 1.

1.1. Propósito

Este documento ofrece una descripción integral de la arquitectura del sistema, empleando diversas perspectivas arquitectónicas para representar distintos aspectos del mismo. Su objetivo es registrar y comunicar las decisiones arquitectónicas significativas adoptadas en el desarrollo del sistema. Así mismo se detalla la estructura del sistema mediante la presentación de cinco vistas distintas, siguiendo el modelo de 4 + 1. Se aborda tanto el comportamiento estático como dinámico del sistema en este documento, proporcionando todos los diagramas necesarios junto con sus correspondientes descripciones.

La aplicación del modelo de vista 4 + 1 posibilita una representación precisa del software, facilitando que diversas partes interesadas encuentren de manera efectiva la información necesaria en el presente documento arquitectónico.

1.2. Alcance

El presente documento abarca todos los elementos estáticos y dinámicos del sistema. Al adoptar el modelo de vista 4 + 1 como marco de referencia, engloba las diversas vistas del sistema, proporcionando así un documento integral.

En cuanto al comportamiento estático del sistema, el presente documento examina los diagramas de clase, los diagramas de paquetes, entre otros. Por otro lado, se abordan los aspectos dinámicos del sistema mediante los diagramas de casos de uso y la utilización de diagramas de secuencia del sistema.

1.3. Definiciones, acrónimos y abreviaturas

Tabla 19. Tabla 1. Definiciones, acrónimos y abreviaturas

Nombre	Descripción
RN	Redes Neuronales
TIC	Trabajo de Integración Curricular
UNL	Universidad Nacional de Loja
CIS/C	Carrera de Ingeniería en Sistemas/Computación
ERS	Especificación de Requisitos Software
RF	Requerimiento Funcional
RNF	Requerimiento No Funcional

1.4. Personal involucrado

Tabla 20. Tabla 2. Personal involucrado Estudiante de la CIS.

Nombre	Jaime Osawaldo Paqui Medina
--------	-----------------------------

Rol	Analista y Desarrollador de Software
Categoría Profesional	Estudiante de la CIS
Responsabilidad	Análisis de información y diseño.
Información de contacto	jaimapaqui@unl.edu.ec

Tabla 21. Tabla 3. Personal involucrado Estudiante de la CIS.

Nombre	Santiago Alexander Roman Silva
Rol	Analista y Desarrollador de Software
Categoría Profesional	Estudiante de la CIS
Responsabilidad	Programador del módulo de software
Información de contacto	santiago.roman@unl.edu.ec

Tabla 22. Tabla 4. Personal involucrado Docente de la CIS

Nombre	Oscar Miguel Cumbicus Pineda.
Rol	Director del Trabajo de Titulación
Categoría Profesional	Docente de la de la CIS/C
Responsabilidad	Supervisar y asesorar en el desarrollo del Trabajo de Titulación
Información de contacto	oscar.cumbicus@unl.edu.ec

1.5. Resumen

El documento proporcionará un análisis minucioso de la arquitectura del software destinado a la detección de plagio en proyectos TIC de la carrera de Computación de la Universidad Nacional del Loja. Además, se abordarán secciones relacionadas con la representación arquitectónica del proyecto, que incluirá tanto la representación arquitectónica y sus limitaciones, así como las implementaciones de casos de uso. Las secciones subsiguientes detallarán de manera específica las cuatro vistas principales del sistema: Lógica, Procesos, Desarrollo e Implementación.

2. Representación Arquitectónica

Dentro de esta sección se proporciona una descripción detallada de la arquitectura mediante el empleo de las vistas establecidas en el modelo "4 + 1". Estas son empleadas para documentar el Prototipo de software diseñado para la identificación de plagio en proyectos de TIC de la carrera de Computación de la Universidad Nacional del Loja.

2.1. Escenarios

Audiencia:

Todas las partes interesadas del sistema, incluidos los usuarios finales.

Área:

Esta vista se centra en los casos de uso del sistema y está destinada a proporcionar una visión clara de cómo los actores interactúan con el software para lograr sus objetivos y las funcionalidades que el sistema debe ofrecer a los usuarios.

Esta vista detalla los casos de uso que representan funciones esenciales y centrales del sistema

2.2. Vista Lógica

Audiencia

Arquitectos de Software, Desarrolladores, Analistas de Sistemas.

Área:

Esta vista describe la estructura interna del sistema y se enfoca en las entidades, clases, y sus relaciones, permitiendo comprender cómo se organiza y representa la información en el software.

2.3. Vista de Procesos

Audiencia:

Arquitectos de Software, Desarrolladores, Analistas de Sistemas, Ingenieros de Pruebas.

Área:

Esta vista representa el comportamiento dinámico del sistema, mostrando cómo los diferentes componentes interactúan entre sí y responden a eventos. Es útil para entender el flujo de control y la lógica de procesamiento dentro del software.

2.4. Vista de Desarrollo

Audiencia:

Desarrolladores, Gerentes de Desarrollo, Gerentes de Calidad.

Área

Esta vista se enfoca en la estructura del software desde una perspectiva de código fuente. Muestra cómo se organizan los módulos, bibliotecas y componentes del sistema, lo que es fundamental para el desarrollo, mantenimiento y reutilización del código.

2.5. Vista Física

Audiencia:

Administradores de Sistemas, Ingenieros de Infraestructura, Arquitectos de Infraestructura.

Área:

Esta vista representa la infraestructura física y el despliegue del sistema, mostrando cómo se distribuyen y conectan los componentes del software en el entorno de ejecución. Es relevante para la configuración, escalabilidad y disponibilidad del sistema en producción.

3. Metas y limitaciones arquitectónicas

3.1. Del lado del servicio:

Meta: Diseñar la arquitectura del servidor de manera que pueda adaptarse a un crecimiento futuro en el número de usuarios y la carga de trabajo sin comprometer el rendimiento.

Limitación: Se debe considerar la capacidad de escalamiento horizontal y vertical, utilizando técnicas como la adición de más servidores o la distribución de carga para garantizar que el sistema pueda manejar una mayor demanda de manera eficiente.

Meta: Implementar mecanismos para detectar y recuperarse de posibles fallos en el servidor, minimizando el impacto en el funcionamiento del sistema.

Limitación: La arquitectura del servidor debe incluir estrategias de recuperación y respaldo, como la replicación de datos y la planificación de puntos de recuperación, para asegurar que el sistema pueda mantener su operatividad en caso de errores o interrupciones.

3.2. Del lado del cliente:

Meta: Diseñar una interfaz de usuario que ofrezca una experiencia fluida y receptiva, independientemente del dispositivo o plataforma utilizado por el usuario.

Limitación: La arquitectura del cliente debe considerar la optimización del rendimiento y la eficiencia en el procesamiento de la interfaz de usuario para garantizar tiempos de respuesta rápidos y una navegación sin problemas.

Meta: Minimizar el consumo de recursos en el cliente, como memoria y energía, para garantizar un rendimiento óptimo y una experiencia de usuario satisfactoria.

Limitación: La arquitectura del cliente debe optimizar el uso de recursos del dispositivo, evitando operaciones innecesarias y priorizando la eficiencia en el procesamiento y almacenamiento de datos.

3.3. Seguridad:

Meta: Garantizar la seguridad y confidencialidad de los datos sensibles del sistema, así como protegerlo contra accesos no autorizados y ataques externos.

Limitación: El sistema debe cumplir con los estándares y mejores prácticas de seguridad establecidos por la industria para proteger la integridad de los datos y la privacidad de los usuarios.

3.4. Persistencia:

Meta: Asegurar la persistencia de los datos relevantes del sistema para que la información no se pierda después de la finalización de las operaciones o en caso de fallos.

Limitación: La solución de persistencia debe ser eficiente y escalable, garantizando una recuperación rápida y confiable de los datos almacenados.

3.5. Confiabilidad / Disponibilidad:

Meta: Garantizar la disponibilidad continua del sistema y minimizar el tiempo de inactividad para evitar interrupciones en el servicio.

Limitación: La arquitectura debe incluir mecanismos de redundancia y tolerancia a fallos para mantener la disponibilidad incluso en situaciones de alto tráfico o incidencias inesperadas.

3.6. Performance:

Meta: Lograr un alto rendimiento y tiempos de respuesta rápidos para satisfacer las necesidades de los usuarios y permitir un uso eficiente del sistema.

Limitación: La arquitectura debe optimizar el uso de recursos, minimizar cuellos de botella y evitar problemas de escalabilidad que puedan afectar negativamente el rendimiento.

3.7. Portabilidad y Reutilización:

Meta: Diseñar el sistema de forma modular y flexible para facilitar su portabilidad a diferentes plataformas y permitir la reutilización de componentes en futuros proyectos.

Limitación: La arquitectura debe adherirse a estándares y tecnologías ampliamente adoptadas para maximizar la compatibilidad y portabilidad del software.

3.8. Herramientas de Desarrollo:

Meta: Proporcionar herramientas de desarrollo eficientes y efectivas que permitan a los desarrolladores trabajar de manera productiva y colaborativa en el proyecto.

Limitación: Las herramientas seleccionadas deben ser adecuadas para el equipo de desarrollo y estar bien integradas con el flujo de trabajo, facilitando el desarrollo, depuración y pruebas del software.

4. Vistas

4.1. Vista de Escenarios

4.1.1. Casos de uso

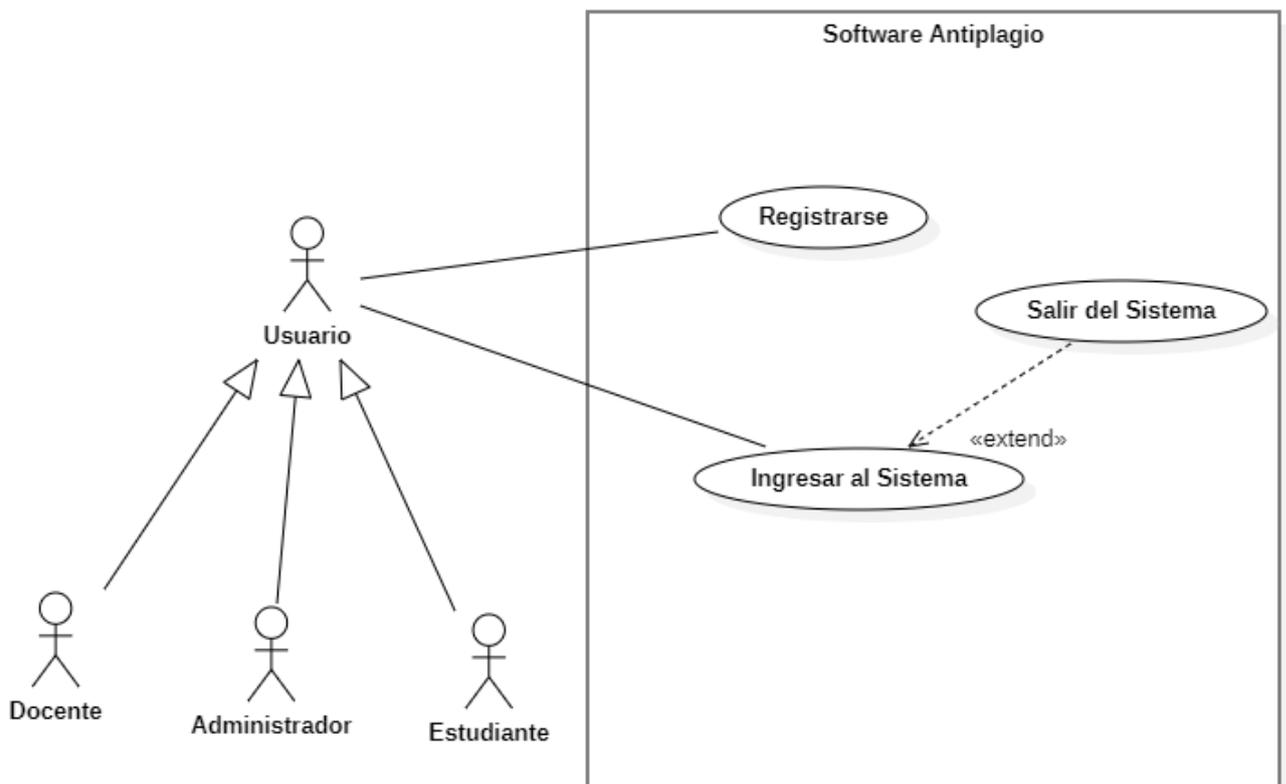


Figura 1. Rol Usuario

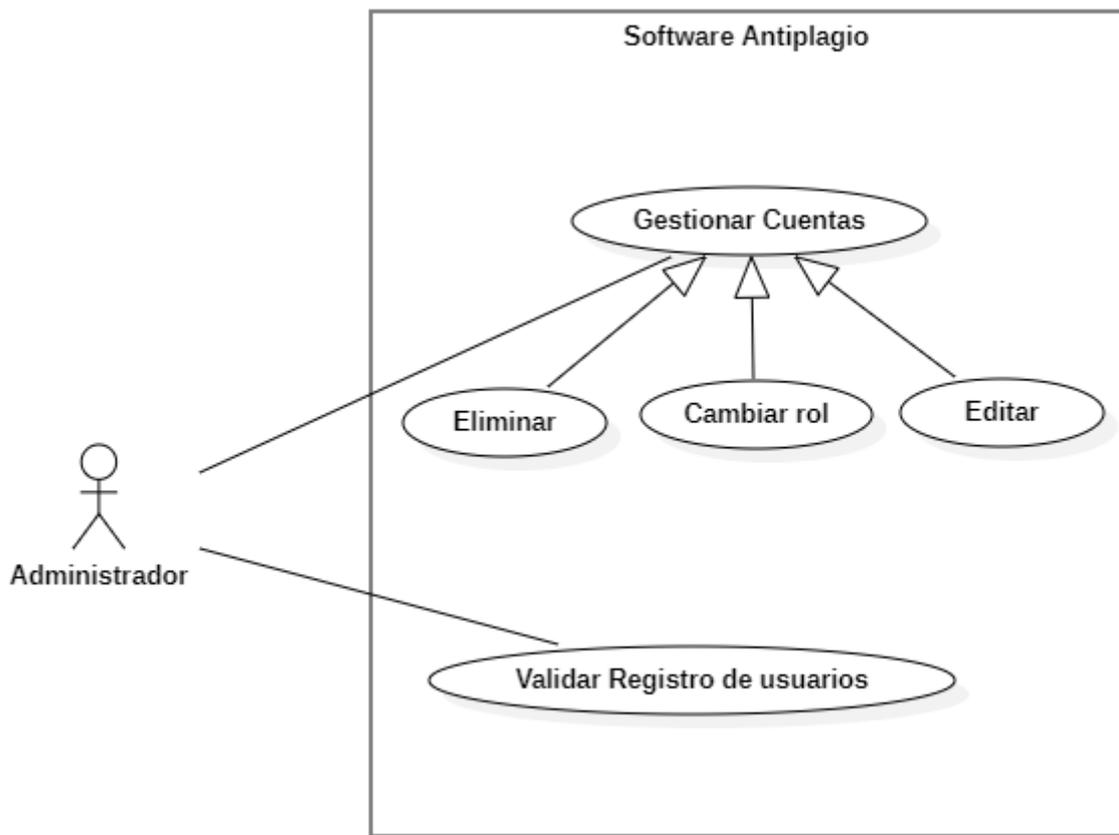


Figura 2. Rol Administrador

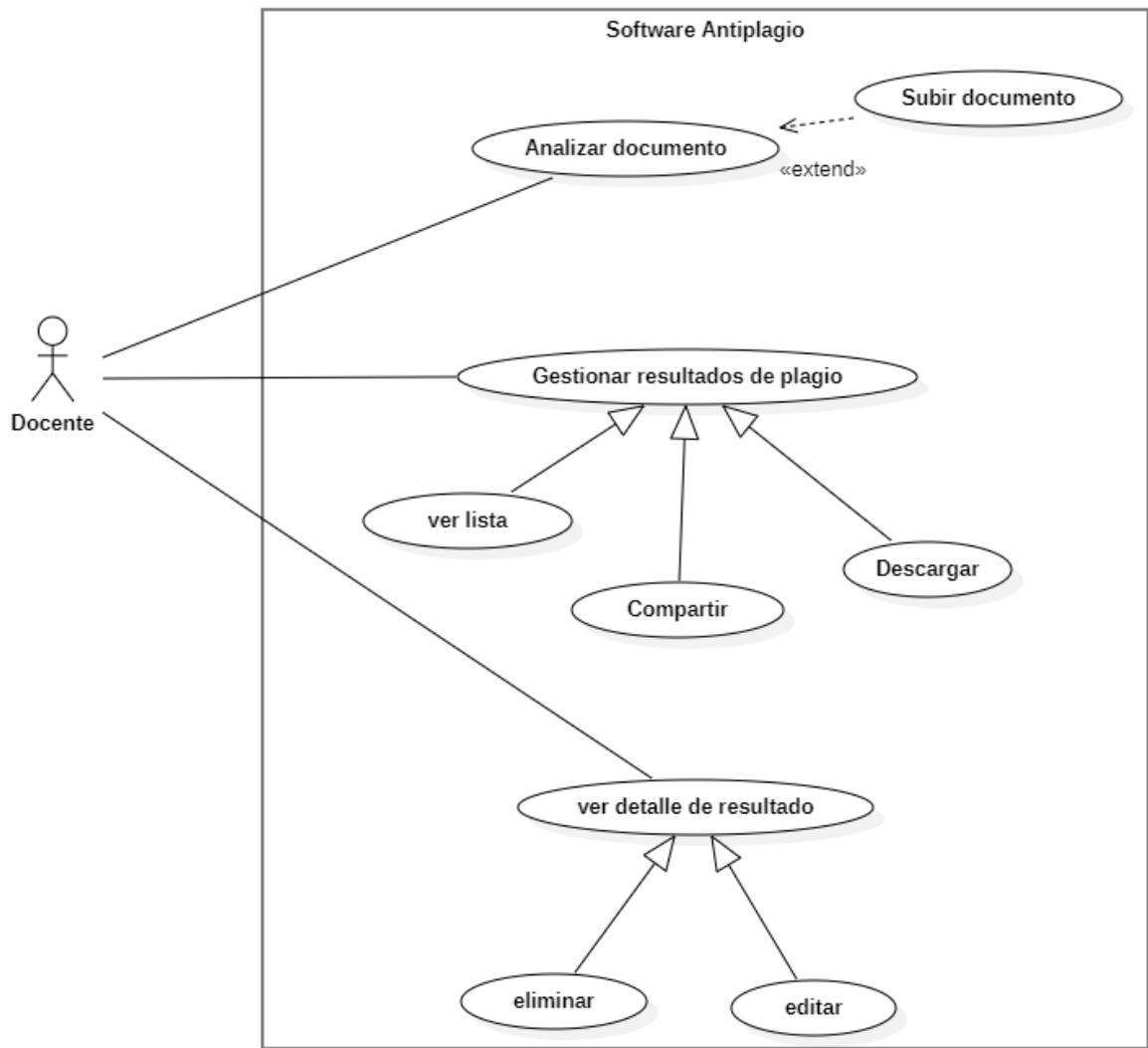


Figura 3. Docente

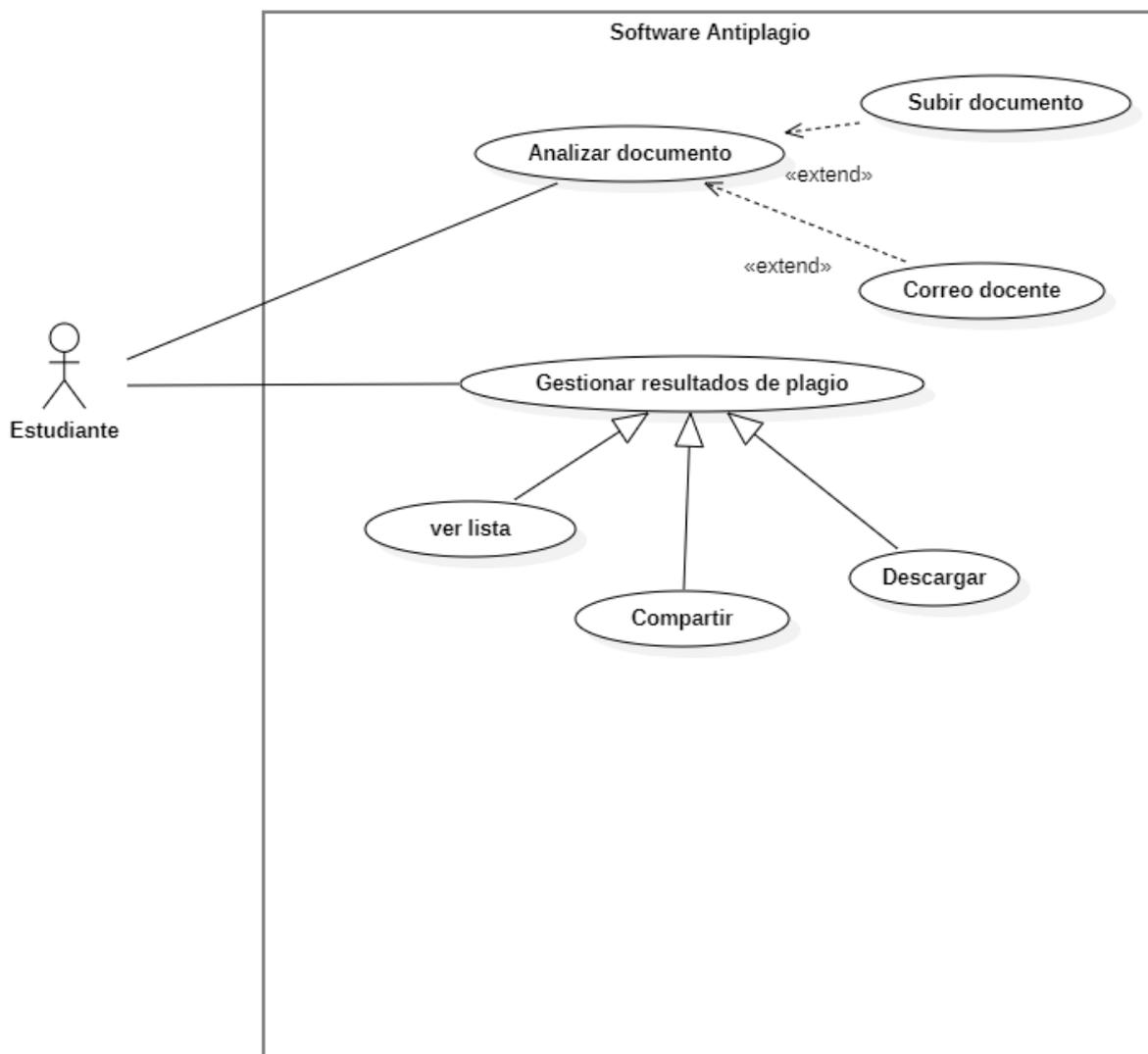


Figura 4. Estudiante

4.1.2. Descripción de cada caso de uso

Tabla 23. **Tabla 5.** Requisito funcional Registrar usuarios.

Identificación del requerimiento:	RF01
Nombre del Requerimiento:	Registrar usuarios
Descripción del requerimiento:	El sistema debe permitir al usuario poder registrarse, para ellos usuario deberá de disponer de los siguientes datos (nombre, apellido, nombre de usuario, correo y contraseña).
Dependencias:	S/N
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03 • RNF04 • RNF05 • RNF08
Prioridad del requerimiento:	Alta

Tabla 24. Tabla 6. Requisito funcional Iniciar sesión.

Identificación del requerimiento:	RF02
Nombre del Requerimiento:	Iniciar sesión.
Descripción del requerimiento:	El sistema debe permitir a los usuarios iniciar sesión, con sus credenciales (nombre de usuario y contraseña).
Dependencias:	<ul style="list-style-type: none"> • S/N
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF02 • RNF03 • RNF04 • RNF05
Prioridad del requerimiento:	Alta

Tabla 25. Tabla 7. Requisito funcional Validar registro de Usuarios

Identificación del requerimiento:	RF03
Nombre del Requerimiento:	Validar registro de usuarios.
Descripción del requerimiento:	El sistema debe permitir al administrador validar o rechazar el registro de un usuario.
Dependencias:	<ul style="list-style-type: none"> • S/N
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03 • RNF04 • RNF05 • RNF08
Prioridad del requerimiento:	Alta

Tabla 26. Tabla 8. Requisito funcional Ver lista de usuarios.

Identificación del requerimiento:	RF04
Nombre del Requerimiento:	Ver lista de usuarios.
Descripción del requerimiento:	El sistema debe permitir al administrador visualizar a todos los usuarios registrados.
Dependencias:	<ul style="list-style-type: none"> • S/N
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03 • RNF04 • RNF05 • RNF08
Prioridad del requerimiento:	Alta

Tabla 27. Tabla 9. Requisito funcional Cambiar rol de cuentas de usuario.

Identificación del requerimiento:	RF05
Nombre del Requerimiento:	Cambiar rol de cuentas de usuario.
Descripción del requerimiento:	El sistema debe permitir al administrador modificar el rol (Admin, Docente o Estudiante) de la cuenta de los usuarios registrados.
Dependencias:	<ul style="list-style-type: none"> • S/N
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01

	<ul style="list-style-type: none"> • RNF02 • RNF03 • RNF04 • RNF05 • RNF08
Prioridad del requerimiento:	Alta

Tabla 28. Tabla 10. Requisito funcional Editar usuarios.

Identificación del requerimiento:	RF06
Nombre del Requerimiento:	Editar usuarios.
Descripción del requerimiento:	El sistema debe permitir al administrador modificar los datos de los usuarios, tales como su nombres, apellidos y correo electrónico.
Dependencias:	<ul style="list-style-type: none"> • S/N
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03 • RNF04 • RNF05 • RNF08
Prioridad del requerimiento:	Alta

Tabla 29. Tabla 11. Requisito funcional Eliminar usuarios.

Identificación del requerimiento:	RF07
Nombre del Requerimiento:	Eliminar usuarios.
Descripción del requerimiento:	El sistema debe permitir al administrador eliminar a los usuarios registrados.
Dependencias:	<ul style="list-style-type: none"> • S/N
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03 • RNF04 • RNF05 • RNF08
Prioridad del requerimiento:	Alta

Tabla 30. Tabla 12. Requisito funcional Analizar plagio por el Estudiante.

Identificación del requerimiento:	RF08
Nombre del Requerimiento:	Analizar plagio por el Estudiante.
Descripción del requerimiento:	El sistema permitirá al estudiante, agregar documentos al sistema para su respectivo análisis, así mismo deberá disponer de un correo electrónico de un docente registrado en el sistema.
Dependencias:	<ul style="list-style-type: none"> • S/N
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03 • RNF04 • RNF05

	•
Prioridad del requerimiento:	Alta

Tabla 31. Tabla 13. Requisito funcional Analizar plagio por el Docente.

Identificación del requerimiento:	RF09
Nombre del Requerimiento:	Analizar plagio por el Docente.
Descripción del requerimiento:	El sistema permitirá al docente, agregar documentos al sistema para su respectivo análisis.
Dependencias:	<ul style="list-style-type: none"> • S/N
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03 • RNF04 • RNF05
Prioridad del requerimiento:	Alta

Tabla 32. Tabla 14. Requisito funcional Compartir resultados de análisis de plagio.

Identificación del requerimiento:	RF10
Nombre del Requerimiento:	Compartir resultados de análisis de plagio.
Descripción del requerimiento:	El sistema debe permitir a los usuarios compartir el análisis del documento plagiado, para ello deberán disponer
Dependencias:	<ul style="list-style-type: none"> • S/N
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02
Prioridad del requerimiento:	Alta

Tabla 33. Tabla 15. Requisito funcional Ver listado de documentos analizados por el usuario.

Identificación del requerimiento:	RF11
Nombre del Requerimiento:	Ver listado de documentos analizados por el usuario.
Descripción del requerimiento:	El usuario podrá visualizar todos los documentos que hayan sido analizados.
Dependencias:	<ul style="list-style-type: none"> • S/N
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF02 • RNF03 • RNF04 • RNF05
Prioridad del requerimiento:	Alta

Tabla 34. Tabla 16. Requisito funcional Ver resultados de análisis de plagio.

Identificación del requerimiento:	RF12
Nombre del Requerimiento:	Ver resultados de análisis de plagio.
Descripción del requerimiento:	El sistema debe mostrar el resultado de análisis de plagio.
Dependencias:	<ul style="list-style-type: none"> • S/N
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF02 • RNF03

	<ul style="list-style-type: none"> • RNF04 • RNF05
Prioridad del requerimiento:	Alta

Tabla 35. Tabla 17. Requisito funcional Editar resultados de análisis de plagio

Identificación del requerimiento:	RF13
Nombre del Requerimiento:	Editar resultados de análisis de plagio
Descripción del requerimiento:	El sistema debe permitir al docente editar el detalle del análisis de plagio.
Dependencias:	<ul style="list-style-type: none"> • S/N
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03 • RNF04 • RNF05
Prioridad del requerimiento:	Alta

Tabla 36. Tabla 18. Requisito funcional Eliminar resultados de análisis de plagio.

Identificación del requerimiento:	RF14
Nombre del Requerimiento:	Eliminar resultados de análisis de plagio.
Descripción del requerimiento:	El sistema debe permitir al usuario eliminar el resultado del análisis de plagio.
Dependencias:	<ul style="list-style-type: none"> • S/N
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03 • RNF04 • RNF05
Prioridad del requerimiento:	Alta

Tabla 37. Tabla 19. Requisito funcional Descargar resultados de análisis de plagio.

Identificación del requerimiento:	RF14
Nombre del Requerimiento:	Descargar resultados de análisis de plagio.
Descripción del requerimiento:	El sistema debe permitir al usuario descargar los resultados obtenidos.
Dependencias:	<ul style="list-style-type: none"> • S/N
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03 • RNF04 • RNF05
Prioridad del requerimiento:	Alta

4.2. Vista Lógica

4.2.1. Diagrama de Clases

El diagrama de clases es una representación gráfica que muestra la estructura del sistema y las relaciones entre las clases que lo componen. En este diagrama se puede observar cómo interactúan las distintas entidades y como se organiza la lógica del programa para detectar plagio entre documentos y textos.

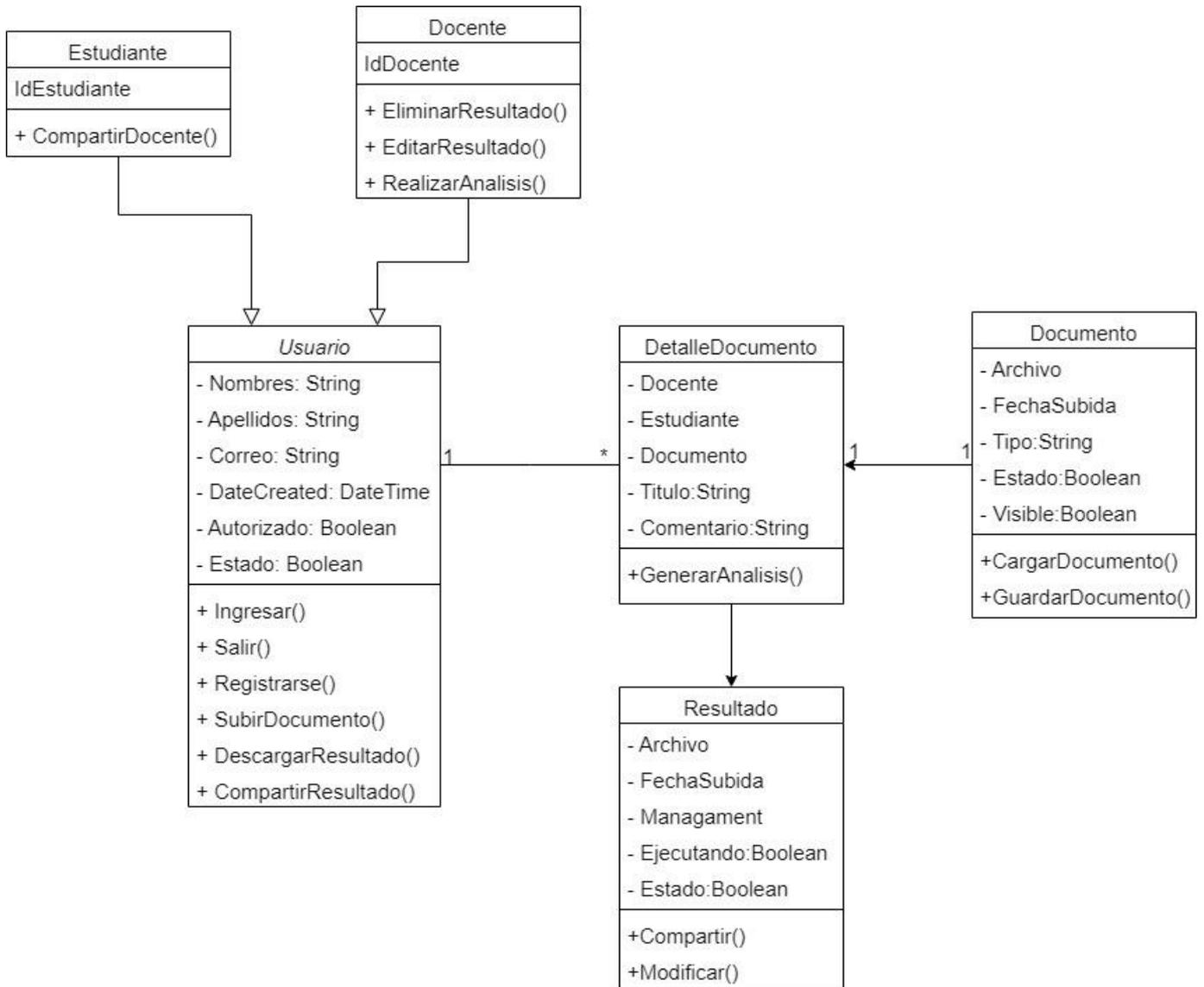


Figura 5. Diagrama de clases general

4.2.2. Diagrama de Secuencia

Estos diagramas ofrecen una representación gráfica que muestra la interacción y el flujo de mensajes entre las diversas clases del sistema, se centran en describir cómo las entidades se comunican a lo largo del tiempo para llevar a cabo las distintas actividades que realizara el usuario.

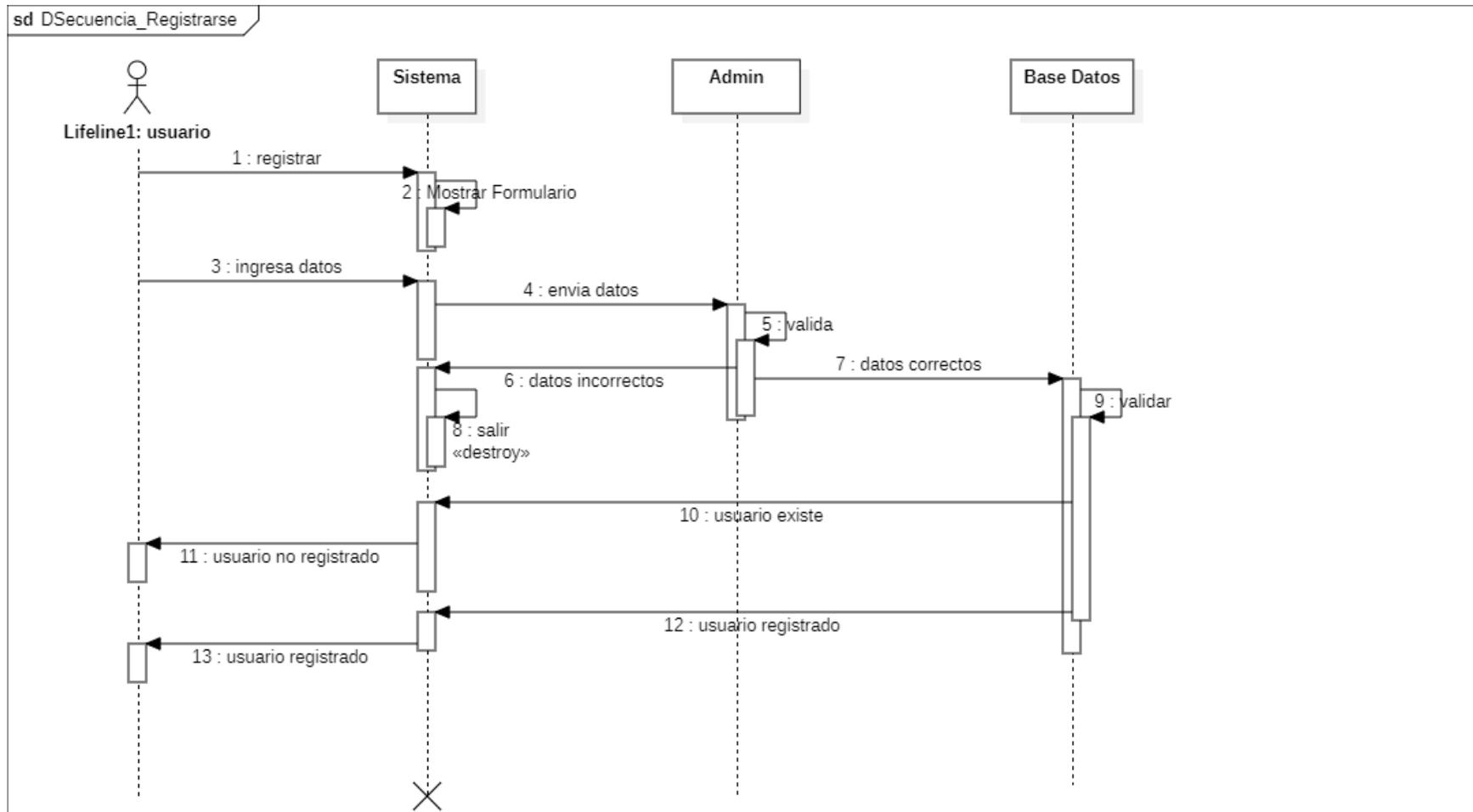


Figura 6. Diagrama de secuencia Registrarse.

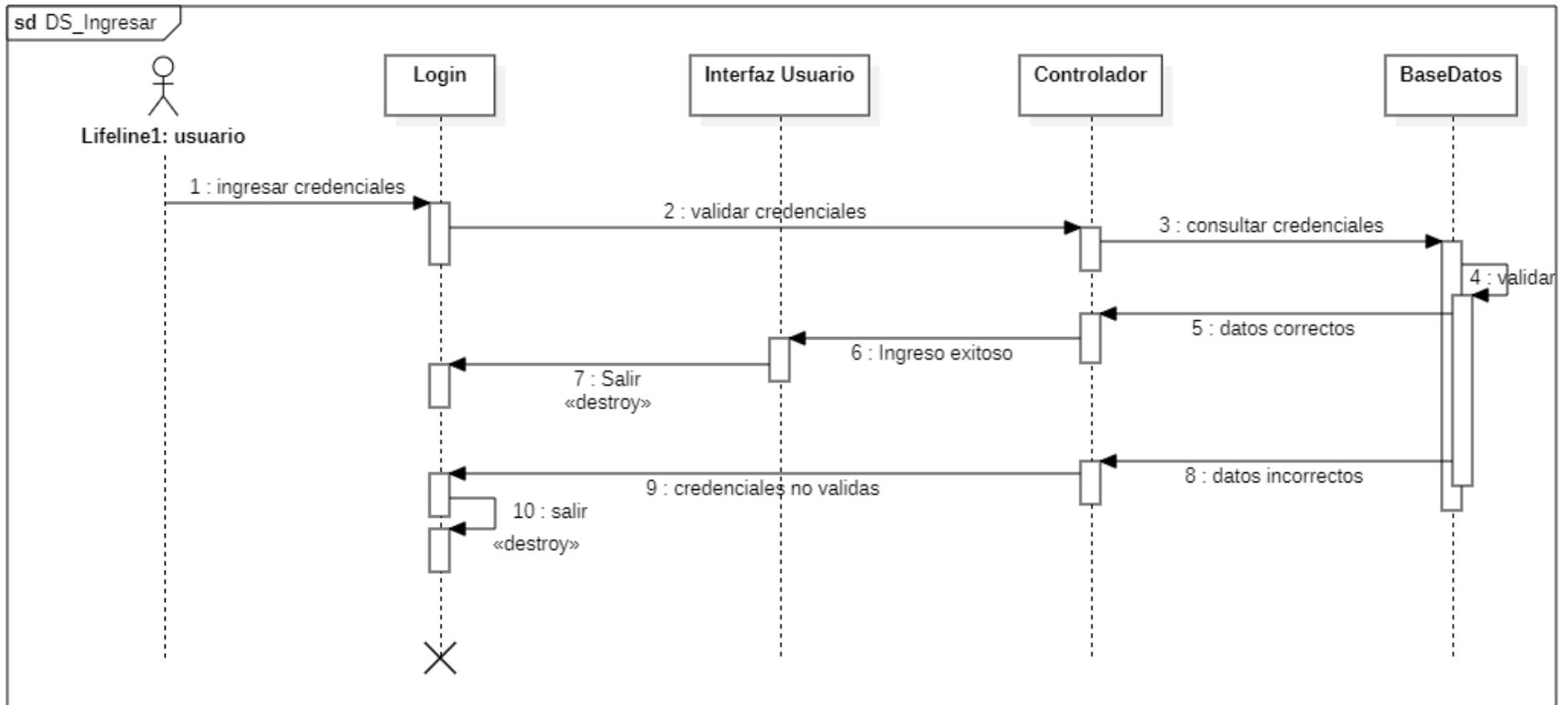


Figura 7. Diagrama de secuencia Inicio de Sesión.

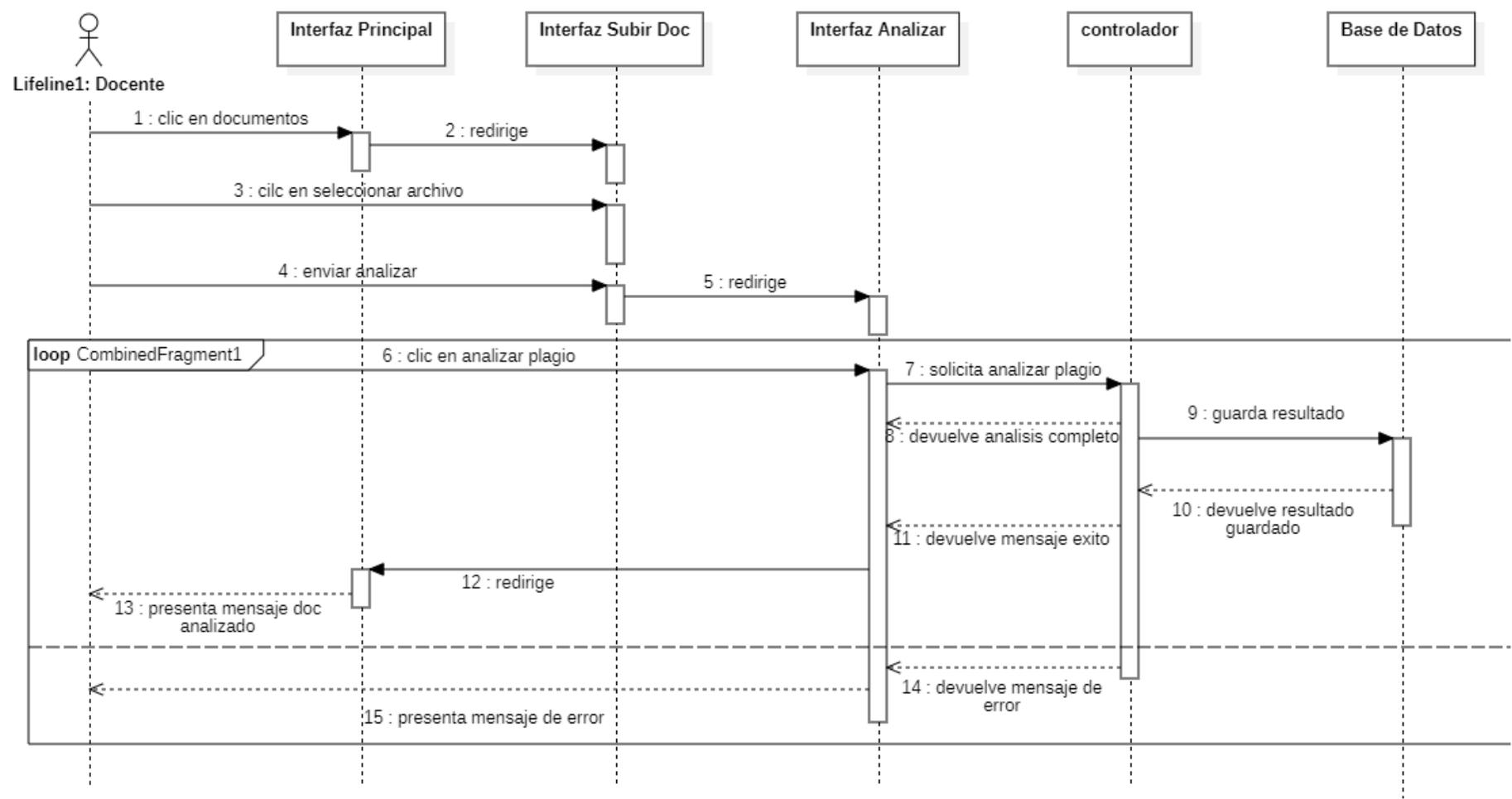


Figura 8. Diagrama de Secuencia Analizar plagio.

4.3. Vista de Procesos

4.3.1. Diagrama de Actividades

Los diagramas de actividades muestran el flujo de las diferentes acciones y decisiones que ocurren dentro de cada actividad del sistema.

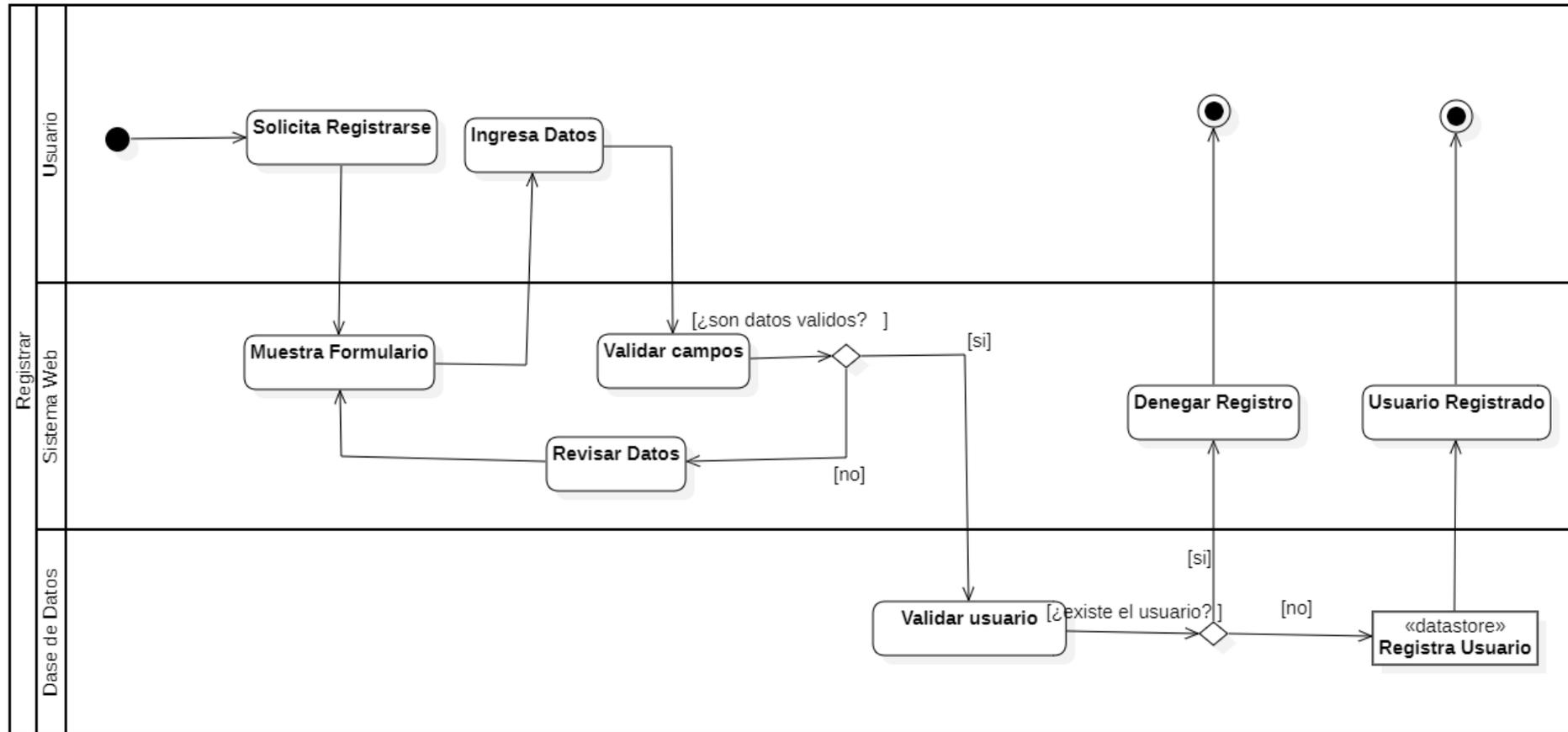


Figura 6 Diagrama de actividades Registrarse.

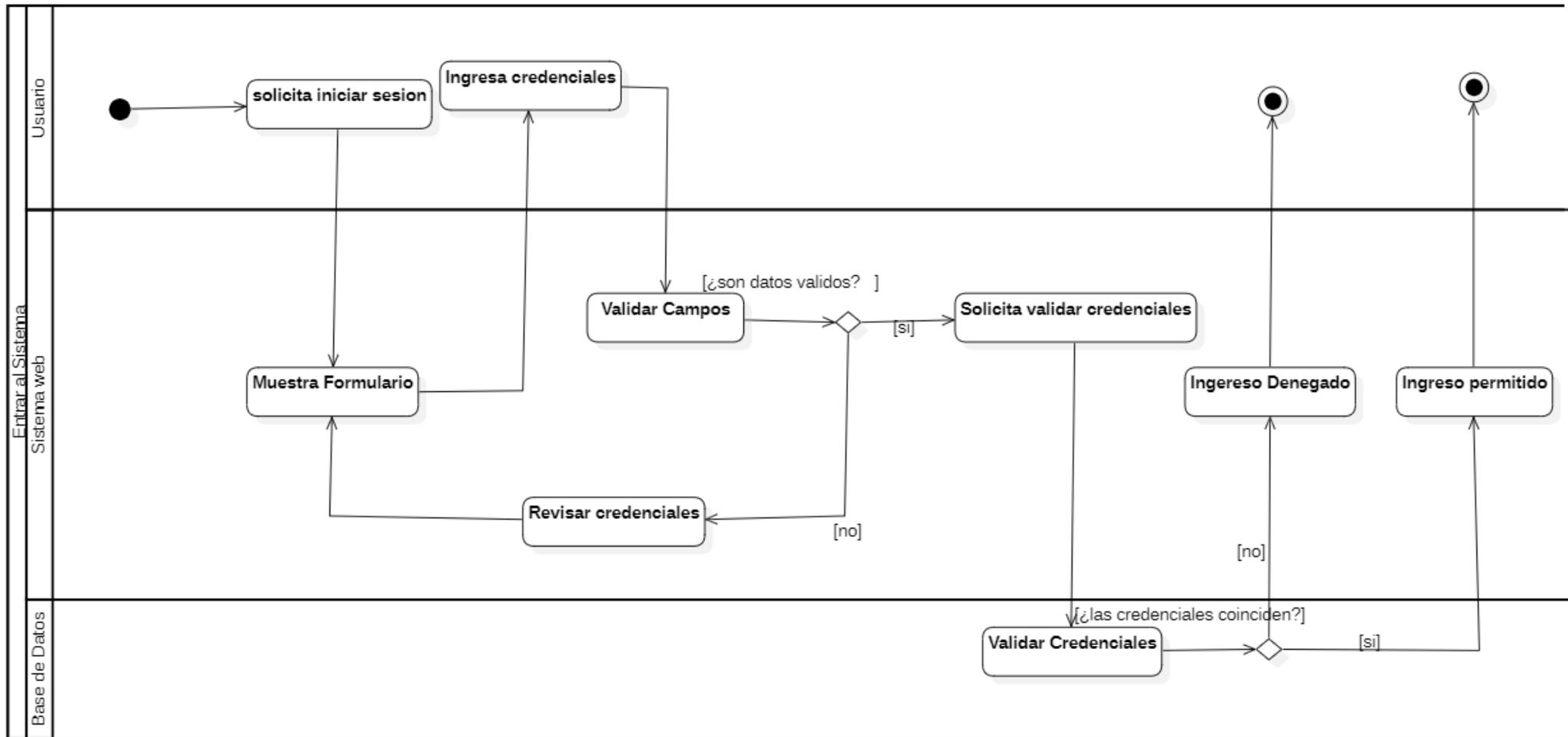


Figura 9. Diagrama de actividades Iniciar Sesión.

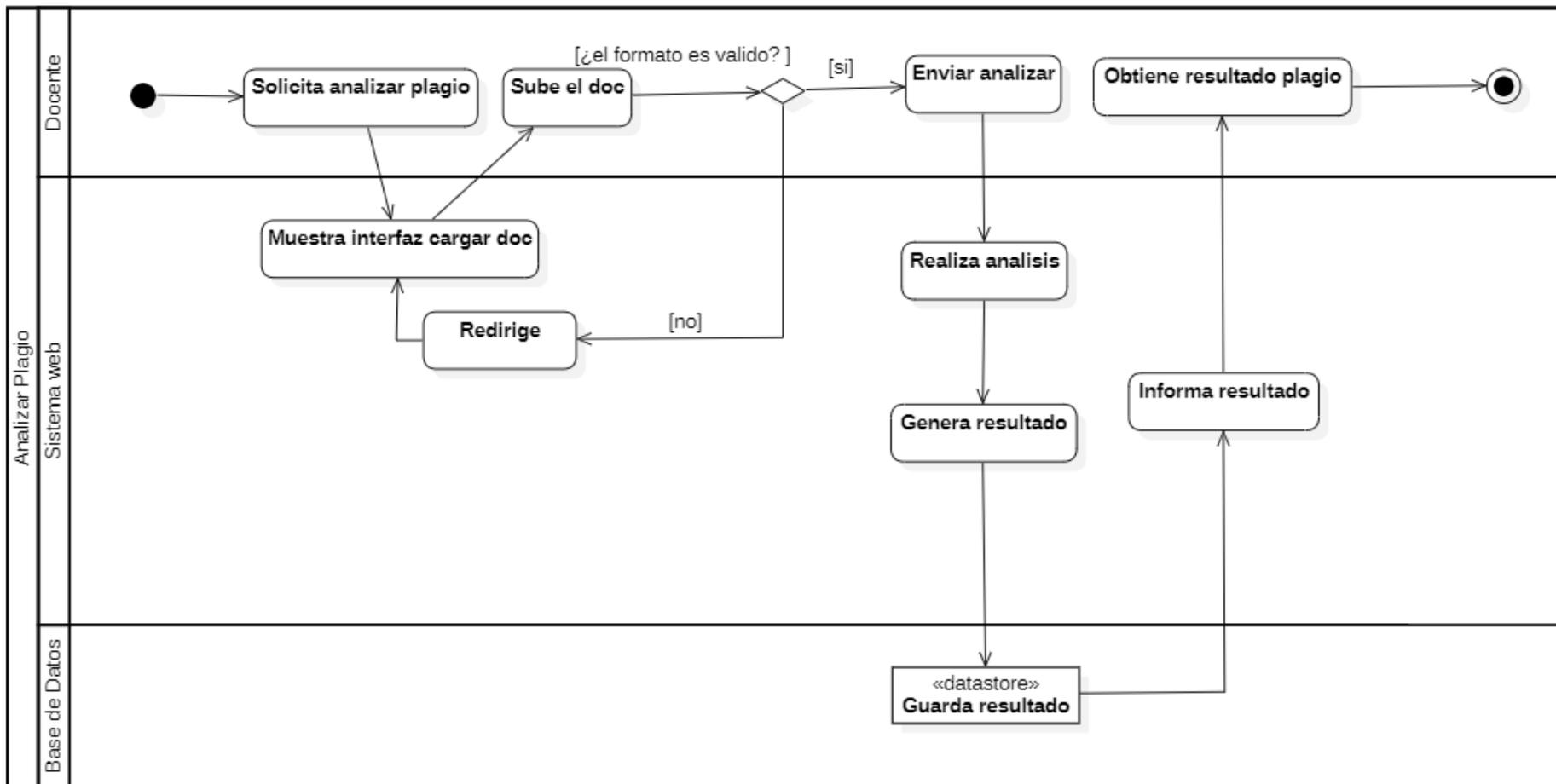


Figura 10. Diagrama de actividades Analizar plagio.

4.4. Vista de Desarrollo

4.4.1. Diagrama de paquetes

El diagrama de paquetes muestra como están agrupadas y organizadas los diferentes elementos (clases, interfaces) del sistema para la detección de plagio.

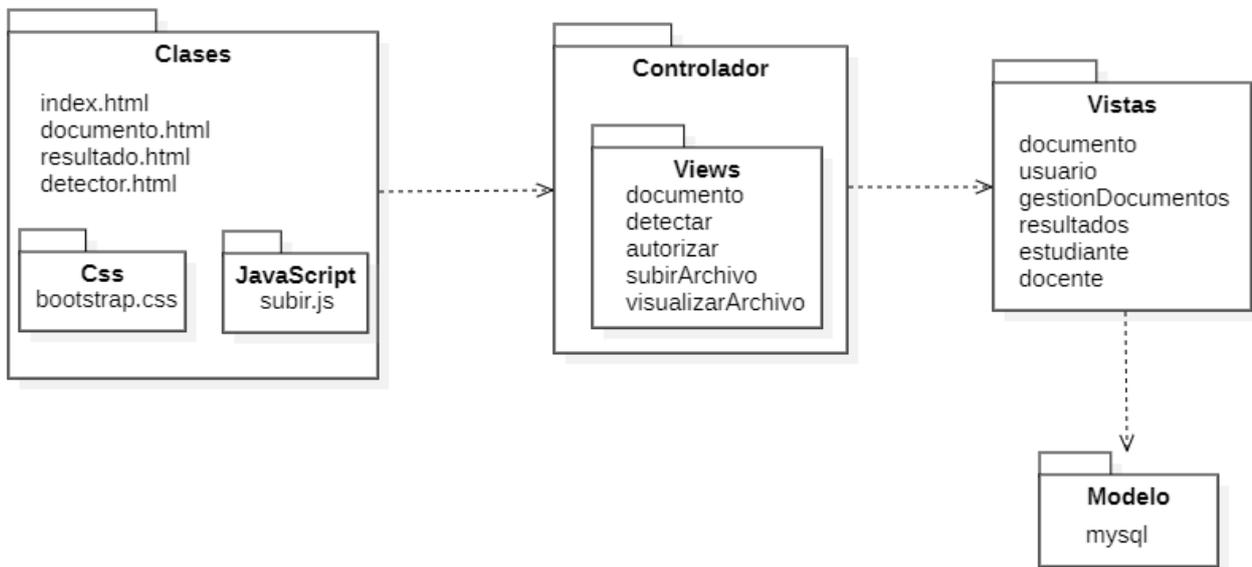


Figura 11. Diagrama de paquetes

4.5. Vista Física

4.5.1. Diagrama de despliegue

Representa la infraestructura física y el despliegue del sistema, mostrando cómo se distribuyen y conectan los componentes del software en el entorno de ejecución.

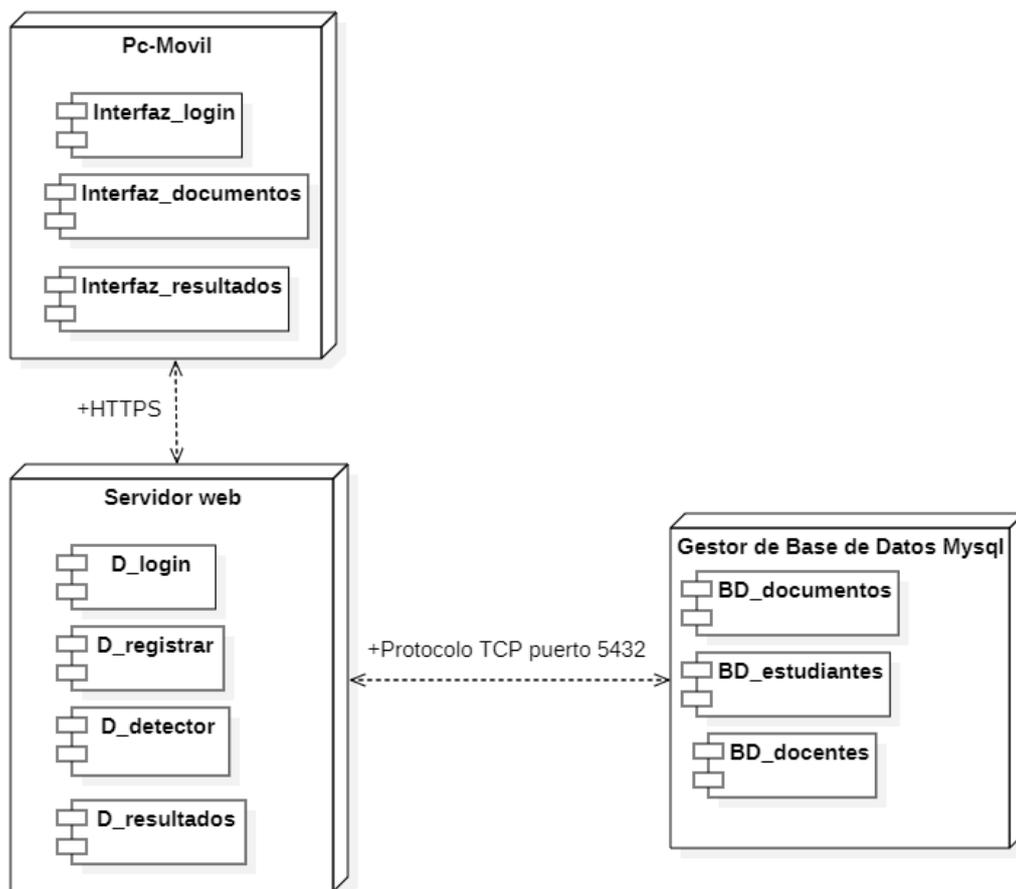


Figura 12. Diagrama de despliegue

5. Firmas de responsabilidad

Acción	Funcionario	Firmas
Elaborado por:	Jaime Oswaldo Paqui Medina Analista de Software	
	Santiago Alexander Román Silva Desarrollador de Software	
Revisado por:	Ing. Oscar Miguel Cumbicus Pineda, Mg.Sc Director del Trabajo de Integración Curricular	

Anexo 15. Certificación de traducción del resumen

Certificación de traducción

Loja, 06 de febrero del 2025

MSc.

Edi M. Herrera Ch.

LICENCIADO EN CIENCIAS DE LA EDUCACION MENCIÓN IDIOMA INGLÉS

Certifica. –

Haber traducido de español a inglés el resumen del trabajo de investigación titulado: **¿Cómo controlar el plagio mediante el desarrollo de un software Anti-plagio para la carrera de Computación en la Universidad Nacional de Loja (UNL), mediante redes neuronales?**, por los tutores estudiantes:

Jaime Oswaldo Paqui Medina. CI. 1900823236 y Santiago Alexander Román Silva. CI. 1106041989.

Es todo cuanto puedo certificar en honor la verdad, pudiendo la interesada hacer uso del presente documento cuando lo considere conveniente.



Edi M. Herrera Ch.

ENGLISH TEACHER

Nro. Reg. Senecyt: 1008-09-9272