

UNIVERSIDAD MARIANO GÁLVEZ DE GUATEMALA
FACULTAD DE INGENIERÍA EN SISTEMAS DE INFORMACIÓN

**SISTEMA WEB DETECTOR DE PLAGIO DE INVESTIGACIONES ACADÉMICAS DE
NIVEL SUPERIOR PARA LA FACULTAD DE INGENIERÍA EN SISTEMAS EN UNA
UNIVERSIDAD GUATEMALTECA**



Lourdes Adriana Pérez Barillas

Guatemala, noviembre del 2024

UNIVERSIDAD MARIANO GÁLVEZ DE GUATEMALA
FACULTAD DE INGENIERÍA EN SISTEMAS DE INFORMACIÓN

**SISTEMA WEB DETECTOR DE PLAGIO DE INVESTIGACIONES ACADÉMICAS
DE NIVEL SUPERIOR PARA LA FACULTAD DE INGENIERÍA EN SISTEMAS EN
UNA UNIVERSIDAD GUATEMALTECA**



Tesis presentada por
LOURDES ADRIANA PÉREZ BARILLAS
Previo a optar al grado académico de
LICENCIADA
Y al título profesional de
INGENIERA EN SISTEMAS DE INFORMACIÓN

Guatemala, noviembre del 2024

**AUTORIDADES DE LA FACULTAD
Y TRIBUNAL QUE PRACTICÓ EL EXAMEN DEL TRABAJO DE
GRADUACIÓN**

DECANO DE LA FACULTAD:	Ing. Jorge Alberto Arias Tobar
SECRETARIO DE LA FACULTAD:	Ing. Hugo Adalberto Hernández Santizo

TRIBUNAL EXAMINADOR

PRESIDENTE:	Ing. Tribunal examinador
SECRETARIO:	Ing. Secretario.
VOCAL:	Ing. Vocal

AUTORIZACIÓN PARA LA IMPRESIÓN DEL TRABAJO DE GRADUACIÓN

REGLAMENTO DE TRABAJO DE GRADUACIÓN

Artículo 8°. RESPONSABILIDAD

Solamente el autor es responsable de los conceptos expresados en el trabajo de tesis. Su aprobación en manera alguna implica responsabilidad para la Universidad.

índice

Capítulo 1 – Anteproyecto de investigación.....	1
1.1 Antecedentes	1
1.2 Justificación	3
1.3 Planteamiento del Problema	4
1.4 Objetivos	5
1.4.1 Objetivo General.....	6
1.4.2 Objetivos Específicos.....	6
1.5 Viabilidad.....	6
1.5.1 Viabilidad de mercado	6
1.5.2 Viabilidad técnica/tecnológica.....	7
1.5.2 Viabilidad de Soporte	8
1.5.3 Viabilidad administrativa.....	8
1.5.4 Viabilidad económica	9
1.6 Alcance	10
1.6.1 Geográfico.....	10
1.6.2 Tecnológico o Técnica.....	10
1.7 Pregunta de Investigación	10
1.7.1 Pregunta general.....	11
1.7.2 Preguntas específicas	11
1.8 Hipótesis	11
1.9 Variables	12
1.9.1 Variables independiente.....	12
1.9.2 Variable dependiente	13

1.10 Indicadores	13
1.11 Supuestos	14
1.12 Métodos de investigación	15
1.12.1 Generalidades.....	15
1.12.2 Diseño de la investigación	15
1.12.3 Población y Muestra	16
1.12.4 Instrumentos de la investigación.....	16
1.12.5 Metodología RUP	16
1.13 Planificación de capítulos	18
1.14 Estimación de recursos	19
2. Capítulo 2 – Marco Teórico.....	22
3.1 Dinámica de Estudiante y Profesor.....	33
3.1.1 Captura de Roles y Gestión de Sesión	33
3.2 Funciones para Estudiantes.....	33
3.3 Funciones para Profesores	34
3.4 Interacción entre Roles	35
3.5 Flujo de inscripción de estudiantes	35
3.6 Accesibilidad y seguridad.....	35
3.6.1 Autenticación de Usuarios	35
3.6.2 Control de Acceso.....	35
3.7 Algoritmo de Similitud de Coseno	36
3.7.1 Preprocesamiento de los Documentos	36
3.8 Base de datos.....	37
Capítulo 4. Desarrollo.....	39
4.1 Desarrollo del frontend	39

4.2 Desarrollo del backend	41
4.2.1 Conexión de Django a SQL Server.....	41
4.2.2 Método POST para enviar datos a las vistas.....	42
4.2.3 Método cosine_similarity.....	42
Capítulo 5. Pruebas	44
5.1 Pruebas unitarias	44
5.1.2 Carga correcta de los archivos	44
5.1.2 Validación de formularios para crear estudiantes	45
5.1.3 Analisis de similitud de documentos	45
5.1.4 Casos de prueba	46
Capítulo 6. Pruebas de certificación	47
6.1 Pruebas de funcionalidad	47
6.2 Pruebas de seguridad.....	48
6.3 Pruebas de carga	48
6.4 Procedimiento de las pruebas.....	49
6.4.1 Casos de prueba	49
Capítulo 7. Implementación.....	51
7.1 Implementación del VPS	51
7.1.1 Importar el proyecto a la VPS con Git.....	52
7.2 Implementación de la BDD desde Azure.....	52
7.3 Librerías necesarias para el funcionamiento del proyecto	53
7.4 Implementación de Open AI en el proyecto	54
7.4.1 Algoritmo de Similitud de Coseno con Open AI.....	55
Capítulo 8. Mantenimiento	57
8.1 Mantenimiento preventivo de la VPS	57

8.1.1 Mantenimiento preventivo a la Base de datos	57
8.1.2 Mantenimiento preventivo de los recursos de OpenAI.....	59
8.2 Mantenimiento correctivo	59
8.2.1 Implementación del patrón de diseño Circuit Breaker	60
8.2.2 Implementación de validador de códigos identificadores de estudiantes y profesores.....	60
8.3 Recomendaciones de seguridad	61
Capítulo 9 – Conclusiones	62
9.1 Análisis de similitudes de Tesis	62
9.2 Propuesta de un modelo de pago	62
9.3 Accesibilidad para Estudiantes No Inscritos.....	63
9.4 Implementación de IA y Vectorización	63
9.5 Conclusión final	63
Capítulo 10 – Recomendaciones.....	65
10.1 Expansión del Sistema a Tesis Culminadas.....	65
10.1.1 Posibles beneficios.....	65
10.2 Digitalización de Tesis Antiguas	65
10.2.1 Posibles acciones	66
10.3 Implementación de RPA (Automatización Robótica de Procesos) para la Carga Masiva de Documentos.....	66
10.4 Fomentar la Participación de los Estudiantes en la Subida de Documentos	66

Índice de Figuras

Figura 1 <i>Ubicación actual de una de las sedes de la Universidad de estudio</i>	10
--	----

Figura 2 <i>Medición de la variable dependiente con respecto a las variables independientes</i>	13
Figura 3 <i>Diagrama de casos de uso general del sistema de administración de anteproyectos de tesis</i>	17
Figura 4 <i>Cronograma preliminar de actividades</i>	18
Figura 5 <i>Ejemplo de Estructura de Desglose de Trabajo (EDT) del proyecto</i>	19
Figura 6 <i>Estructura de Desglose de Trabajo (EDT) del proyecto</i>	20
Figura 7 <i>Diagrama Entidad Relación para analisis de anteproyectos de tesis en una universidad</i>	38
Figura 8 <i>Plantilla HTML base para todas las plantillas del proyecto</i>	40
Figura 9 <i>Bloque de contenido utilizando el base.html</i>	41
Figura 10	41
Figura 11 <i>Método POST para obtener datos de los formularios de las plantillas</i>	42
Figura 12 <i>Uso del método de cálculo de similitud de coseno</i>	43
Figura 13 <i>Error detectado en el cálculo de similitud de tesis</i>	48

Índice de Tablas

Tabla 1 Viabilidad económica	9
Tabla 2 Actividades preliminares de la investigación	17
Tabla 3 Planificación de costes del proyecto con detalle de los recursos a utilizar	20
Tabla 4 Matriz de roles y responsabilidades según el modelo de diagrama RACI	20

Introducción

La presente investigación tiene como objetivos el mejorar el proceso de revisión de tesis aprobadas en la fase de anteproyecto para los docentes y estudiantes de Proyecto de Graduación en una universidad guatemalteca; Facilitar la detección de proyectos duplicados; Automatizar el proceso de revisión de tesis; Implementar una base de datos para los estudiantes en donde visualicen los títulos de los proyectos conforme a su búsqueda para disminuir la duplicidad o el plagio.

A través de la pregunta de investigación "¿Cómo puede optimizarse el proceso de consulta de anteproyectos de tesis en la Facultad de Ingeniería en Sistemas de Información de una universidad guatemalteca para gestionar los proyectos aprobados y reducir la probabilidad de plagio o duplicidad?", se busca perfeccionar dicho proceso con el fin de reducir la posibilidad de plagio o duplicidad, mejorar la eficiencia en la revisión de tesis con recursos limitados, agilizar la detección de temas duplicados y proporcionar a los estudiantes de la facultad información sobre los temas en desarrollo.

La justificación indica que la detección de plagio sin herramientas tecnológicas puede ser una tarea laboriosa. Aunque algunas instituciones usan herramientas como Turnitin, estas solo detectan similitudes en bases de datos específicas. Por eso, tanto docentes como estudiantes necesitan una base de datos universitaria que permita consultar proyectos de tesis previos. Este proyecto de tesis propone desarrollar un software exclusivo para una universidad que detecte plagio, ayudando a los estudiantes a evitar duplicar investigaciones y a culminar con éxito sus tesis de Ingeniería al contar con ejemplos de trabajos anteriores.

El desarrollo de proyectos de tesis es una de las fases más críticas en la formación académica de los estudiantes universitarios. En el contexto educativo actual, garantizar la originalidad y la calidad en los trabajos de investigación es fundamental no solo para promover el aprendizaje autónomo, sino también para prevenir el plagio y la falta de ética académica. En este sentido, las herramientas tecnológicas desempeñan un papel crucial, y el uso de sistemas automáticos para la comparación de documentos ha ganado relevancia en diversas instituciones educativas. Este trabajo se enfoca en el diseño y desarrollo de una plataforma para el análisis de similitud de anteproyectos de tesis, con la finalidad de facilitar la evaluación de la originalidad de los proyectos preliminares y, en un futuro, de las tesis culminadas.

La implementación de un sistema de análisis de similitud en el ámbito académico no solo permite identificar coincidencias entre textos, sino que también sirve como una herramienta de prevención, proporcionando retroalimentación a los estudiantes en las primeras etapas de sus investigaciones. Sin embargo, a pesar de sus beneficios, el proceso de gestión y comparación de los documentos presenta una serie de desafíos, especialmente cuando se trata de integrar tesis de años anteriores, las cuales se encuentran en formato físico y no han sido digitalizadas. Este desafío de digitalización limita la capacidad del sistema para acceder y procesar trabajos académicos antiguos, afectando la calidad y la amplitud de las comparaciones.

A lo largo del desarrollo de este proyecto, se ha explorado cómo superar esta barrera mediante la propuesta de soluciones como la automatización de la carga de documentos a través de RPA (Automatización Robótica de Procesos). La automatización no solo facilita la integración masiva de documentos, sino que también optimiza la gestión de grandes volúmenes de información, reduciendo el margen de error humano y aumentando la eficiencia del sistema. Además, se ha considerado la expansión del sistema más allá de los anteproyectos, con la intención de incluir también las tesis culminadas en futuras implementaciones, lo que permitirá ofrecer una solución más completa para la comparación y validación de trabajos académicos en diferentes etapas del proceso de redacción.

El presente trabajo propone, por lo tanto, un modelo de sistema de análisis de similitud de tesis que permita a las instituciones académicas gestionar y evaluar la calidad de los trabajos de manera efectiva. Además, se aborda la importancia de la digitalización de tesis históricas, la optimización del proceso de revisión mediante tecnologías avanzadas, y la necesidad de crear una plataforma accesible tanto para los estudiantes como para los profesores. A través de la implementación de estas tecnologías y estrategias, se busca no solo mejorar la calidad de los proyectos de tesis, sino también contribuir a la construcción de un entorno académico más transparente y eficiente.

Capítulo 1 – Anteproyecto de investigación

1.1 Antecedentes

El origen de la automatización se encuentra en el siglo XVIII de la Revolución Industrial, que vio cómo la mecanización reemplaza a la mano de obra. La automatización es un valioso asistente, sirviendo no solo para ahorrar esfuerzos; también se ha incautado como un medio de protección contra los errores. El propósito de la herramienta automatizada que controla el proceso de revisión de la tesis es reducir el esfuerzo humano para completar la tarea de detectar una violación de derechos, lo que ayuda a evitarla.

En su investigación titulada "Similitud en tesis de pregrado de medicina publicadas en repositorios de Universidades de Trujillo", Loayza Salvatierra (2019) se propuso determinar la frecuencia de similitud en las tesis de pregrado de medicina de las universidades de la ciudad de Trujillo. Según los resultados obtenidos, se evaluaron un total de 292 tesis de la Universidad Nacional de Trujillo (UNT) y 263 de la Universidad Privada Antenor Orrego (UPAO). La similitud promedio encontrada en las tesis de la UNT fue del 32.59%, mientras que en las de la UPAO fue del 32.49%. Además, se observó una desviación estándar de la similitud de 16.17% en la UNT y 13.63% en la UPAO.

Respecto a la metodología, todas las tesis fueron recolectadas y analizadas utilizando el software Plagiarism Checker X, y los datos se ingresaron en una hoja de cálculo Excel junto con información detallada sobre el autor, asesor, tipo de estudio, universidad de origen, año de sustentación y enlace al repositorio correspondiente. Las conclusiones de la investigación resaltan la alta frecuencia de similitud encontrada en las tesis de pregrado de medicina de las universidades de Trujillo, señalando un problema significativo en cuanto a la originalidad de los trabajos académicos en estas instituciones.

Piñero Pérez, et al., (2019), en el artículo titulado "Desarrollo de un Repositorio de Datos para Investigaciones en Gestión de Proyectos" de la Universidad de las Ciencias Informáticas de Cuba, se estableció como objetivo la creación de un repositorio especializado en gestión de proyectos. Este repositorio consta de 18 bases de datos destinadas a almacenar tesis de doctorado, tesis de maestría, artículos y otros tipos de investigaciones relevantes en el campo. El enfoque de

la investigación radica en proporcionar una herramienta que facilite el acceso al conocimiento y la experimentación con nuevos algoritmos de estructura de datos. Con este fin, se diseñó un modelo específico para la construcción de repositorios de datos destinados al desarrollo de investigaciones en este ámbito.

Las conclusiones principales del estudio resaltan la eficiencia en el acceso a la información y la facilitación de la experimentación en la gestión de proyectos. El repositorio mejora significativamente la centralización y recuperación de información relevante, lo que permite a los investigadores encontrar y utilizar datos específicos de manera más eficiente. Asimismo, el repositorio proporciona un entorno adecuado para la validación y prueba de nuevos algoritmos, facilitando la innovación en la estructura de datos.

Castro Rodríguez (2020), en su artículo "El plagio académico desde la perspectiva de la ética de la publicación científica", de la Universidad Católica del Perú, el cual tuvo como objetivo abordar los conceptos fundamentales, las causas, los factores relacionados y las consecuencias del plagio desde el punto de vista ético de la publicación científica. Propone un enfoque que orienta al investigador a llevar a cabo su estudio con integridad, asegurando que se cumplan los estándares de excelencia científica y se genere confianza en su desarrollo. Se llevó a cabo una revisión exhaustiva con el fin de identificar las causas del plagio.

En su estudio, Castro Rodríguez realizó una revisión exhaustiva de la literatura existente para identificar las causas del plagio académico. Entre las conclusiones, se destaca la necesidad de fomentar una cultura de integridad académica desde las etapas iniciales de la formación del investigador. Se identificaron factores relacionados con el plagio, como la presión por publicar, la falta de conocimiento sobre lo que constituye plagio, y la insuficiente formación ética en las instituciones académicas.

Luis (2022) en el artículo "Causas del plagio académico en estudiantes universitarios de educación: percepción docente de una universidad dominicana", realizado en la Universidad Pedagógica Experimental Libertador de República Dominicana, se tuvo como objetivo identificar, a través de la visión de los docentes, las principales razones detrás del plagio entre los estudiantes universitarios. El estudio se centró en analizar casos de plagio en diversos trabajos académicos, como tesis, trabajos de fin de grado y disertaciones. Se llevó a cabo una exhaustiva revisión

bibliográfica para comprender las causas del plagio académico, así como su impacto y las recomendaciones para prevenirlo.

Las conclusiones del estudio destacan que la principal causa del plagio académico radica en la falta de comprensión sobre lo que constituye plagio y las normas de citación adecuadas. Además, se identificó que la presión por obtener buenas calificaciones y la falta de tiempo debido a múltiples compromisos académicos y personales son factores que contribuyen significativamente al plagio.

González Lemus (2019), en su tesis titulada "Sistema para la automatización del proceso de trabajo de graduación para los estudiantes de la escuela de Ingeniería Mecánica Industrial, Facultad de Ingeniería, Universidad de San Carlos de Guatemala", el objetivo principal fue desarrollar una aplicación web que permitiera la gestión eficiente de la información relacionada con proyectos de graduación. Se enfocó en supervisar y registrar las diferentes aprobaciones requeridas durante el proceso. Para lograrlo, se emplearon diagramas entidad-relación y tablas para definir los requisitos necesarios.

Entre los hallazgos más importantes, se menciona que la implementación de un sistema automatizado reduce significativamente el tiempo y el esfuerzo requeridos por los estudiantes y docentes para completar las aprobaciones necesarias. Además, el sistema proporciona una mayor transparencia y claridad en los procedimientos, lo que ayuda a evitar errores y retrasos.

1.2 Justificación

En la actualidad, la detección de plagio es una tarea que, en ausencia de herramientas tecnológicas, puede resultar muy laboriosa y consumir mucho tiempo. Algunas instituciones utilizan herramientas como Turnitin para detectar similitudes en bases de datos de proyectos académicos. Sin embargo, estas herramientas se limitan a las bases de datos específicas que se les indiquen. Por esta razón, cuando se inicia un proyecto de tesis en un curso académico, tanto docentes como estudiantes necesitan una base de datos que permita consultar, a nivel universitario, los proyectos ya existentes. Este proyecto de tesis propone el desarrollo de un software de detección de Este software permitirá a los estudiantes orientarse con otros anteproyectos, evitando la duplicación de investigaciones y asegurando que las nuevas tesis no tengan un alto porcentaje de similitud con trabajos anteriores. A largo plazo, se espera optimizar la gestión de los

anteproyectos de tesis para que tanto profesores como estudiantes puedan alinear eficazmente los objetivos de la investigación y responder adecuadamente a la hipótesis planteada.

1.3 Planteamiento del Problema

En el ámbito universitario, la adecuada gestión de los anteproyectos de tesis resulta esencial para fomentar la investigación innovadora y evitar la duplicación de esfuerzos. Esta organización garantiza que cada estudiante aborde temas originales y relevantes, respetando los tiempos de vigencia definidos para cada proyecto. Según el Repositorio Académico de la Universidad de Chile, cada documento depositado representa el esfuerzo intelectual de sus autores y está protegido bajo políticas de propiedad intelectual establecidas por los mismos autores. Esto significa que los derechos morales y de autor son inalienables y pertenecen al creador de la obra desde el momento de su creación, sin transferencia automática al repositorio ni pérdida de control sobre el contenido depositado ("Los derechos de autor pertenecen al autor de la obra por el mero hecho de su creación" (Repositorio Académico, Universidad de Chile, s.f.)).

Esta política destaca que cada obra académica debe ser gestionada con cuidado para proteger los derechos de sus autores y su valor intelectual. Por lo tanto, un sistema organizado de gestión de temas de tesis no solo permite a los estudiantes trabajar en temas relevantes y originales, sino que también respeta los derechos de propiedad intelectual, evitando el uso indebido de ideas previamente desarrolladas sin el debido reconocimiento.

En la universidad de estudio, actualmente la gestión de los temas de anteproyectos de tesis se lleva a cabo a nivel de cada sede, sin un sistema centralizado que permita un control general. Esto dificulta el seguimiento y la protección de los temas durante su vigencia de dos años, lo cual aumenta la posibilidad de que se aborden temas repetidos o fuera de los plazos permitidos. Como resultado, los estudiantes enfrentan dificultades para verificar si su tema propuesto ya ha sido abordado en otra sede o si aún se encuentra dentro del tiempo de vigencia establecido. Esta falta de transparencia en la gestión de los temas afecta la originalidad de los proyectos y limita las oportunidades para explorar nuevas áreas de investigación.

Además, la ausencia de un sistema que permita el acceso a ejemplos de anteproyectos de tesis aprobados complica el proceso de definición y delimitación de temas de investigación para los estudiantes. Cada estudiante enfrenta este proceso de manera aislada y, con frecuencia, realiza

múltiples revisiones y correcciones sin contar con una referencia clara sobre los estándares de calidad y delimitación temática. Una herramienta de este tipo no solo facilitaría la orientación de los estudiantes, sino que también promovería la originalidad y el aprovechamiento de nuevas oportunidades de investigación, al evitar la repetición de temas previamente trabajados.

La formación universitaria incluye el desarrollo de habilidades para identificar y analizar problemáticas relevantes tanto para la sociedad como para el campo de estudio del estudiante. Para lograr esto, es crucial formular preguntas que permitan enfocar y abordar problemas de manera específica y manejable (Universidad de Chile, s.f.).

El problema que esta investigación aborda es, entonces, la ausencia de un sistema centralizado que permita gestionar y proteger los temas de anteproyectos de tesis a nivel universitario. Un sistema de esta naturaleza garantizaría la consulta y organización adecuada de los temas durante su tiempo de vigencia, evitando duplicidades y promoviendo una investigación académica más eficiente y original.

1.4 Objetivos

La facultad de Ingeniería en Sistemas de Información contiene dentro del pensum de la carrera el curso de Proyecto de Graduación 1 y 2, con el objetivo de que los estudiantes reciban la educación que se necesita para desarrollar un proyecto de tesis, que sirve como base para la entrega final de la tesis para el título de Ingeniería. Esto causa que existan muchos proyectos de tesis manejados por la universidad, los cuales se administran por medio de una carta de autorización para desarrollar el tema. Actualmente no existe un proceso unificado para administrar las tesis de la facultad en la universidad, pero se administra de distinta manera en cada sede, por lo que las cartas de autorización pueden autorizar temas similares al mismo tiempo. Esto involucra que los docentes del curso apoyen a los estudiantes a validar cada tema consultando a la universidad para garantizar su autenticidad.

Para resolver el problema se planteó un objetivo general compuesto por diferentes objetivos específicos para desarrollar un sistema que implemente nuevos procesos que hagan que la administración de proyectos de tesis sea más sencillo de manejar.

1.4.1 Objetivo General

Apoyar el proceso de administración de anteproyectos de tesis mediante la automatización de la revisión y calificación, para que los docentes puedan detectar indicios de duplicidad o plagio. Además, brindar acceso a los estudiantes para conocer los temas vigentes.

1.4.2 Objetivos Específicos

- Facilitar la detección de proyectos duplicados o plagiados.
- Automatizar el proceso de revisión de tesis.
- Implementar una herramienta para los estudiantes que permita la visualización de los títulos y descripciones de los proyectos solo si los documentos son públicos, ayudando a reducir la duplicidad o el plagio conforme a la búsqueda realizada.

1.5 Viabilidad

1.5.1 Viabilidad de mercado

Para entender la viabilidad de mercado se necesita especificar el quién es el interés principal en el proyecto, es decir el mercado objetivo. Los usuarios principales del sistema detector de plagio son los estudiantes y los docentes que imparten el curso de Proyecto de Graduación, los cuales brindan el conocimiento necesario para realizar la investigación de tesis. El mercado objetivo entonces es la universidad guatemalteca y los usuarios en el curso de Proyecto de Graduación (docentes profesionales y estudiantes).

En la actualidad existen herramientas en el mercado que ayudan a garantizar la autenticidad de los proyectos académicos muy populares como Turnitin, Grammarly, Plag, etc. Los precios de estas herramientas para universidades pueden variar, sin embargo el dato exacto sobre el precio institucional es brindado al ponerse en contacto con la organización. Existen planes personales que varían entre los \$20.

Los potenciales usuarios, como los estudiantes del curso de Proyecto de Graduación deben presentar ideas al comenzar el semestre primero, consultando tesis de la universidad y de otras universidades para darse ideas sobre problemas a resolver en Guatemala. Las estadísticas sobre la prevalencia del plagio en anteproyectos de tesis en una sede muestran que existen proyectos con

similitud, por lo que solicitan a los estudiantes realizar el cambio de tema. Los estudiantes no tienen conocimiento de los temas que están en proceso en la universidad de estudio.

El propósito del sistema para detección de plagio del presente proyecto es apoyar a los estudiantes para que, en su lluvia de ideas, exista el conocimiento de temas propuestos con anterioridad. Esta herramienta facilita a las instituciones a organizar de mejor manera los temas de proyectos de tesis para los usuarios como docentes y estudiantes. Algunas universidades cuentan con herramientas similares y algunas aún carecen de un método de administración de temas de tesis como anteproyecto.

1.5.2 Viabilidad técnica/tecnológica

Para el desarrollo del proyecto, se emplea la técnica de vectorización de textos para realizar análisis de similitud. Este enfoque convierte los textos en representaciones numéricas que pueden ser procesadas por algoritmos especializados en análisis de datos. Dado que las máquinas no interpretan el lenguaje como los humanos, la vectorización transforma palabras y frases en vectores (listas o matrices de números) que capturan sus características distintivas. Esto es esencial en el Procesamiento de Lenguaje Natural (NLP), facilitando tareas de análisis de similitud entre textos.

El proyecto se hospeda en un Servidor Privado Virtual (VPS), que almacena los documentos a analizar. Además, la información de usuarios y los detalles de las tesis se gestionan mediante una base de datos SQL, la cual está alojada en un servidor separado para garantizar una estructura de almacenamiento segura y organizada.

El proyecto implementa el algoritmo de similitud de coseno para comparar los componentes clave de un proyecto de tesis, una vez que estos se han convertido en vectores. Este algoritmo evalúa la similitud entre dos vectores en un espacio multidimensional midiendo el "ángulo" entre ellos en lugar de la distancia. En el contexto de análisis de texto, permite comparar documentos al representarlos como vectores de palabras, lo cual es útil para identificar el grado de similitud con otros anteproyectos en función de su contenido principal.

Para asegurar un rendimiento óptimo en el sistema de detección de plagio, se han estimado los requerimientos de hardware necesarios. La infraestructura debe permitir a la institución almacenar grandes volúmenes de documentos y brindar acceso eficiente a estudiantes y profesores.

Los requisitos mínimos incluyen un procesador de 8 núcleos (vCPU), 16 GB de RAM y 1 TB de almacenamiento SSD. No obstante, para alcanzar un rendimiento superior, la institución puede optar por especificaciones mayores según sus necesidades. El objetivo del proyecto es ofrecer una gestión ágil de los anteproyectos de tesis y un tiempo de respuesta óptimo para los usuarios.

1.5.2 Viabilidad de Soporte

Para asegurar el funcionamiento ininterrumpido de la metodología propuesta y fomentar su crecimiento en la implementación, el proyecto incluye un plan integral de soporte y actualizaciones. Dado que nuestra metodología hace uso del procesamiento de lenguaje natural (PLN), se espera que los avances en este campo sean de naturaleza exponencial.

Con el objetivo de mantener al día nuestro software, se planea ofrecer una amplia gama de documentación y recursos de soporte para los usuarios. Esto incluirá manuales detallados, guías de uso y una base de conocimientos que aborde las preguntas frecuentes y ofrezca soluciones a los problemas comunes.

1.5.3 Viabilidad administrativa

El objetivo de proteger los derechos de autor es asegurar la seguridad jurídica para los autores, titulares de derechos conexos y patrimoniales, y sus herederos. Además, busca dar la publicidad adecuada a las obras, actos y documentos mediante su inscripción cuando los titulares lo soliciten (Registro de la Propiedad Intelectual de Guatemala, 2022).

Se puede decir entonces que la función de los derechos de autor es garantizar la seguridad jurídica para los autores y para las obras. Mediante la inscripción de una obra, no sólo se brindan los datos obtenidos por el autor, sino también facilita el acceso a ella para investigaciones hereditarias.

Autor

Es la persona física quien realiza la creación intelectual. Solo las personas naturales pueden ser autoras de una obra; sin embargo, el Estado, las entidades de derecho público y las personas jurídicas pueden ser titulares de los derechos establecidos para los autores, en los casos especificados en la misma (Art. 5 de la Ley de Derecho de Autor y Derechos Conexos de Guatemala, 2000).

Se puede decir entonces que un autor es aquel que realiza una obra, y puede ser una persona natural, jurídica, una entidad de derecho público o el Estado el titular de una creación intelectual.

1.5.4 Viabilidad económica

Para garantizar la viabilidad económica del proyecto, se ha realizado una estimación detallada de los recursos materiales y humanos, así como del tiempo necesario y los costos asociados. En la siguiente tabla, se desglosan los elementos clave que constituyen el presupuesto para el desarrollo del sitio web, desde su creación hasta su implementación. Todos los gastos contemplados son para el desarrollo desde cero de un sitio web robusto y funcional, considerando los costos anuales y mensuales involucrados.

Tabla 1

Viabilidad económica

Concepto	Detalles	Costo unitario	Cantidad	Costo total
Programador	Pago por hora	Q150.00	960 horas	Q144,000.00
Certificado	Mensual	Q152.00	12 meses	Q1,824.00
Servidor Virtual Privado		Q18,240.00	1 año	Q18,240.00
Base de datos		Q9,120.00	1 año	Q9,120.00
Servicios de APIs		Q912.00	1 año	Q912.00
Dominio		Q960.00	1 año	Q960.00
Total anual				Q176,056.00

Se ha evaluado el plan de hosting utilizado inicialmente para el proyecto y, tras revisar las recomendaciones, se decidió utilizar una suscripción en OVH Cloud. Este plan ofrece 16 GB de RAM, 8 núcleos de CPU y 1 TB GB de almacenamiento, asegurando un rendimiento óptimo para manejar el procesamiento intensivo que requiere el sistema de detección de plagio. Además, proporciona ancho de banda ilimitado y copias de seguridad diarias, lo cual garantiza estabilidad y protección para el proyecto.

Debido a los elevados costos del proyecto derivados de los recursos necesarios, la institución podría considerar la opción de cobrar a los estudiantes por el uso del servicio, según sus necesidades y presupuesto.

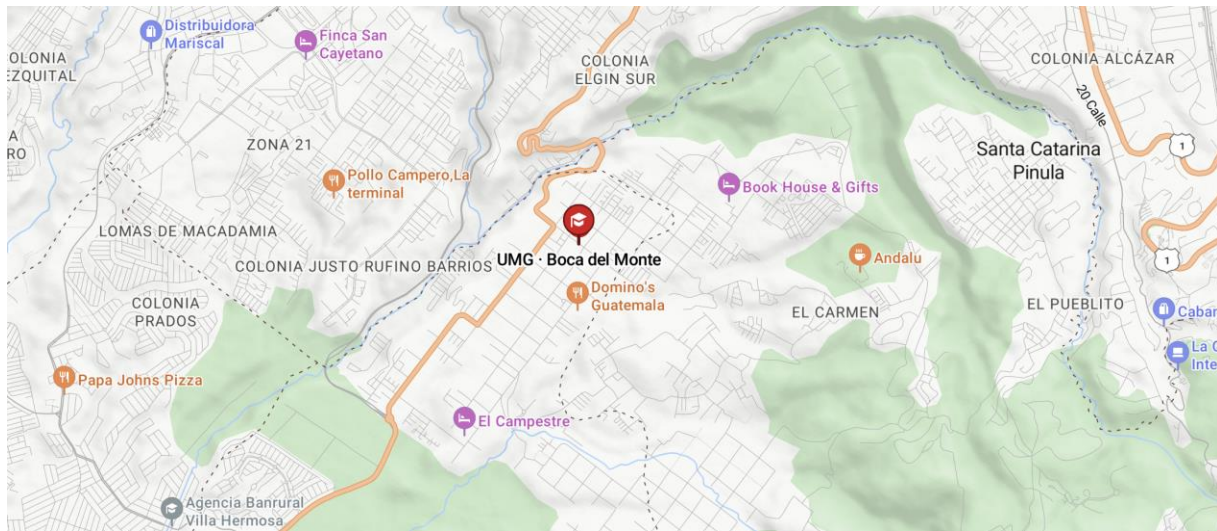
1.6 Alcance

1.6.1 Geográfico

La investigación se llevará a cabo en la sede de una Universidad ubicada en Boca del Monte, del municipio de Villa Canales en el departamento de Guatemala. En la Figura 1 se visualiza el mapa de la ubicación exacta.

Figura 1

Ubicación actual de una de las sedes de la Universidad de estudio



1.6.2 Tecnológico o Técnica

El proyecto se centra en el uso del lenguaje de programación Python y la base de datos sobre SQL para administrar los datos de los usuarios y los documentos que serán analizados.

El análisis de textos con Procesamiento del Lenguaje Natural (PNL) implementado por la librería PDFVectorizer y el algoritmo de similitud de coseno integrado a la misma librería para mejorar la eficiencia en el cálculo de similitudes

1.7 Pregunta de Investigación

El sistema pretende solucionar un problema de administración de proyectos de tesis y detección de similitudes de proyectos. La pregunta de investigación se basa en el objetivo general,

el cual se dividió en preguntas específicas para responder específicamente cómo se solucionará cada objetivo específico.

1.7.1 Pregunta general

¿Cómo se puede mejorar el método de consulta de los anteproyectos de tesis para la facultad de Ingeniería en Sistemas de Información en una universidad guatemalteca para administrar los proyectos aprobados disminuyendo la probabilidad de plagio o duplicidad?

1.7.2 Preguntas específicas

- ¿Cómo se puede facilitar la detección de proyectos duplicados o plagiados?
- ¿Cuánto podemos automatizar el proceso de revisión de tesis?
- ¿Implementar una base de datos para los estudiantes en la que se pueda visualizar únicamente los títulos de los proyectos, ayuda a disminuir el plagio o la duplicidad?

1.8 Hipótesis

La hipótesis sugiere resultados encontrados en la web con respecto a la herramienta a nivel internacional Turnitin, la cual ha observado que, con el tiempo, las instituciones pueden ver una disminución de entre el 17.4% y el 66.8% en trabajos con más del 50% de contenido no original (Turnitin, s.f.).

La eficiencia en la revisión de tesis y automatización de la misma radicará en encontrar errores comunes y párrafos En un análisis realizado por Grammarly (s.f.), mostró que los usuarios redujeron 158.3 horas por año y un 69% de mejora en calidad de los escritos.

La implementación de un sistema de detección de plagio mejorará significativamente el proceso de administración de proyectos de tesis, al disminuir la probabilidad de plagio o duplicación de contenido en un rango estimado del 30% al 50%. Este sistema permitirá identificar de manera más precisa y eficiente contenido no original, reduciendo aproximadamente un 50% de los errores comunes. Además, optimizará el tiempo y los recursos utilizados en la revisión de tesis, al facilitar la comparación de similitudes entre los proyectos de forma automatizada. Esto proporcionará a los estudiantes retroalimentación oportuna sobre la originalidad de sus trabajos en curso.

1.9 Variables

Las variables de estudio además de ser la manera en que se espera visualizar resultados, también amplían el propósito de la investigación. En este estudio se evalúan variables que identifican la importancia del conocimiento de investigaciones en proceso y la revisión de similitudes entre todas ellas.

1.9.1 Variables independiente

La administración de proyectos de tesis en general implica factores que caracterizan la forma en la que funciona para saber cómo mejorarlo. Para identificar cómo se implementan los objetivos, se obtienen las siguientes variables independientes:

- a) Temas duplicados
- b) Porcentaje de temas duplicados o plagiados en la revisión de tesis
- c) Búsqueda de temas presentados

Para definir el concepto de temas duplicados, el Comité Internacional de Editores de Revistas Médicas (ICMJE) utiliza el título "Publicación Superpuesta". El ICMJE define la publicación duplicada como "la publicación de un artículo que se superpone sustancialmente con uno ya publicado, sin referencia clara y visible a la publicación previa". La medición de los temas duplicados nos permite saber la magnitud de la necesidad de controlar los temas en un repositorio en línea. Se medirá el porcentaje de temas duplicados al inicio del estudio en diferentes sedes.

La revisión de tesis se define por la autora como el proceso sistemático y crítico mediante el cual se examina y evalúa un documento académico extenso, el cual ha sido elaborado por un estudiante como requisito para la obtención de un título universitario. La medición de la variable de revisión de un proyecto de tesis se realizará con el objetivo de reducir el tiempo de lectura y análisis. Se planea medir el tiempo invertido en la revisión de proyectos de tesis para automatizar el proceso haciéndolo más productivo.

La variable independiente de búsqueda de temas presentados se define por el autor como el proceso de investigar y analizar la existencia previa de trabajos o investigaciones relacionadas con un tema específico, la cual se medirá con un porcentaje de satisfacción para identificar si la búsqueda de trabajos es efectiva y relevante.

Según (Sampieri Hernández, 2014) se les llaman variables independientes a aquellos elementos, como tratamientos, estímulos, influencias o intervenciones, que son manipulados en experimentos. Esto nos sirve para observar los efectos resultantes el cual en este caso define como variable dependiente.

1.9.2 Variable dependiente

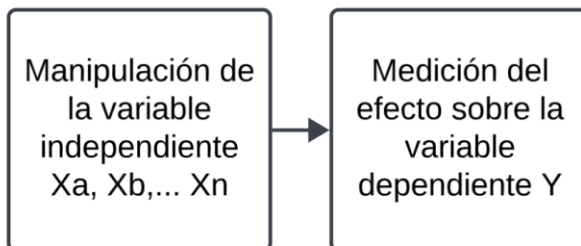
Para mejorar el proceso de revisión de tesis aprobadas en la fase de anteproyecto, se medirá la eficiencia en el proceso de revisión de tesis como variable dependiente, la cual es definida por el autor como el aumento de productividad y disminución de recursos en el proceso de detección de tesis duplicadas y revisión de tesis.

La variable dependiente se medirá con respecto al porcentaje de temas duplicados detectados, el tiempo invertido en la revisión de los trabajos y en la satisfacción de los estudiantes y docentes con respecto al proceso actual y al proceso que se quiere implementar.

Según (Sampieri Hernández, 2014) la variable dependiente no se altera deliberadamente, sino que se observa y registra para entender cómo la modificación de la variable independiente influye en ella. Esta dinámica se representa en la Figura 2.

Figura 2

Medición de la variable dependiente con respecto a las variables independientes



1.10 Indicadores

Los indicadores brindarán información proveniente de la universidad con datos obtenidos de cuestionarios brindados a la Universidad. Los indicadores son los siguientes:

- Identificación de patrones de duplicación

- Índice de similitud entre temas duplicados y publicaciones previas
- Tiempo de respuesta en cuanto al proceso de esperar a recibir una tesis para revisión.
- Tiempo promedio de revisión.
- Tasa de error en la revisión.
- Eficiencia de la búsqueda siendo esta rápida o muy lenta.

1.11 Supuestos

La similitud de proyectos de investigación en estudiantes difiere con respecto a la institución en la que la misma se lleva a cabo, lo cual dice que puede disminuirse con ciertas prácticas que realizan los establecimientos con menor porcentaje de similitud presentada los trabajos de investigación de los estudiantes. Según (Loayza Salvatierra, 2019) la frecuencia de similitud de las tesis de pregrado de medicina del repositorio de la Universidad Nacional de Trujillo fue de 32.59% con 292 tesis, a diferencia de la Universidad Privada Antenor Orrego sede Trujillo el cual fue de 32.49% con 263.

Para tener un mejor control de los trabajos estudiantiles, hoy en día se utilizan plataformas virtuales en donde se verifica que el trabajo sea auténtico, sin embargo, cuando una tesis está en la fase de anteproyecto, no todas las instituciones universitarias cuentan con una herramienta que verifique un tema para validar si existe alguno en proceso y evitar que esta investigación se duplique. Según (Piñero Pérez, et al., 2019) la creación de un repositorio de tesis garantiza la pertenencia y el acceso a la base de datos hace más fácil para los autores el poder enterarse de otros temas en proceso.

Cuando hablamos del plagio estudiantil, existen casos en los que es muy difícil poder detectarlo cuando existen textos literales introducidos al trabajo sin una cita, pero en el caso de ser duplicados, puede ser detectado cuando existe un control. Según (Castro Rodriguez, 2020) el plagio se comprende como un fenómeno complejo y multidimensional que compromete los estándares éticos de las publicaciones científicas. Sus orígenes abarcan motivaciones personales, la pereza académica, la facilidad y la comodidad de acceso a material para elaborar un trabajo.

En casos de plagio, desde la percepción del docente, puede representar dificultades el poder detectar plagio en investigaciones de estudiantes. Según (Luis, 2022) el plagio entre estudiantes es una práctica ampliamente difundida que requiere atención para mitigar sus efectos. Este estudio

busca fundamentar esta afirmación, centrándose en la dimensión del control docente para la prevención del plagio. Se destaca que muchos docentes no detectan estos problemas en los documentos de los estudiantes, lo que constituye una variable significativa en este contexto.

Según (González Lemus, 2019) el sistema de automatización de proyectos de graduación optimiza la elaboración de trabajos al brindar un mayor control sobre los estudiantes y sus temas a todo el personal involucrado. Además, proporciona a los estudiantes la ventaja de acceder a los temas en desarrollo, lo que previene la pérdida de tiempo en la planificación de propuestas.

1.12 Métodos de investigación

1.12.1 Generalidades

Se estableció un enfoque de investigación cuantitativo con un alcance correlacional, puesto que se necesita verificar cómo se puede automatizar la gestión de documentos y cuánto tiempo se reduce.

1.12.2 Diseño de la investigación

Dado el objetivo del estudio que será mejorar el proceso de gestión y control de tesis aprobadas en la fase de anteproyecto para los docentes de Proyecto de Graduación en una universidad guatemalteca, se recurrirá a un diseño experimental que se aplicará de manera transversal, considerando que el tema analiza más de una sede y se necesita aplicar las variables para ver un antes y un después, se procederá a realizar una investigación de tipo correlacional y poder encontrar la relación entre las variables de estudio.

El uso de la técnica de estudio será experimental, debido a que se manipulan variables para obtener un resultado favorecedor, en este caso para la automatización de revisión de tesis y cómo el sistema de gestión de anteproyectos disminuye la probabilidad de duplicidad y plagio.

En un estudio experimental se examinan las relaciones directas entre las variables de interés, sin la influencia de otras variables, lo que permite determinar relaciones causales con mayor exactitud. Por ejemplo, en un estudio sobre el aprendizaje, se podrían ajustar el estilo de liderazgo del profesor, el método de enseñanza y otros factores (Hernández Sampieri et al., 2014).

Por otra parte, las investigaciones transversales son aquellas que con los datos recolectados se estudiará en un solo momento (Liu, 2008 y Tucker, 2004). Su objetivo es examinar variables y entender su influencia y relación en un momento específico, similar a capturar una fotografía de un evento. Por ejemplo, evaluar las percepciones y actitudes de mujeres jóvenes (de 18 a 25 años) que han experimentado abuso sexual durante el último mes en una ciudad latinoamericana (Sampieri Hernández, 2014).

1.12.3 Población y Muestra

Los docentes que imparten la cátedra de Proyecto de Graduación 1 y 2 en una universidad guatemalteca, los cuales están distribuidos en las sedes de la Facultad de Ingeniería en Sistemas. También forma parte de la población los estudiantes de la facultad.

De igual manera, los estudiantes inscritos en los cursos de Proyecto de Graduación 1 y 2 para el uso de la plataforma.

1.12.4 Instrumentos de la investigación

Dentro de la recolección de datos sobre la metodología que se utiliza actualmente, se utilizarán los siguientes instrumentos de investigación, los cuales se encuentran en el apéndice:

- Cuestionarios
- Encuestas

1.12.5 Metodología RUP

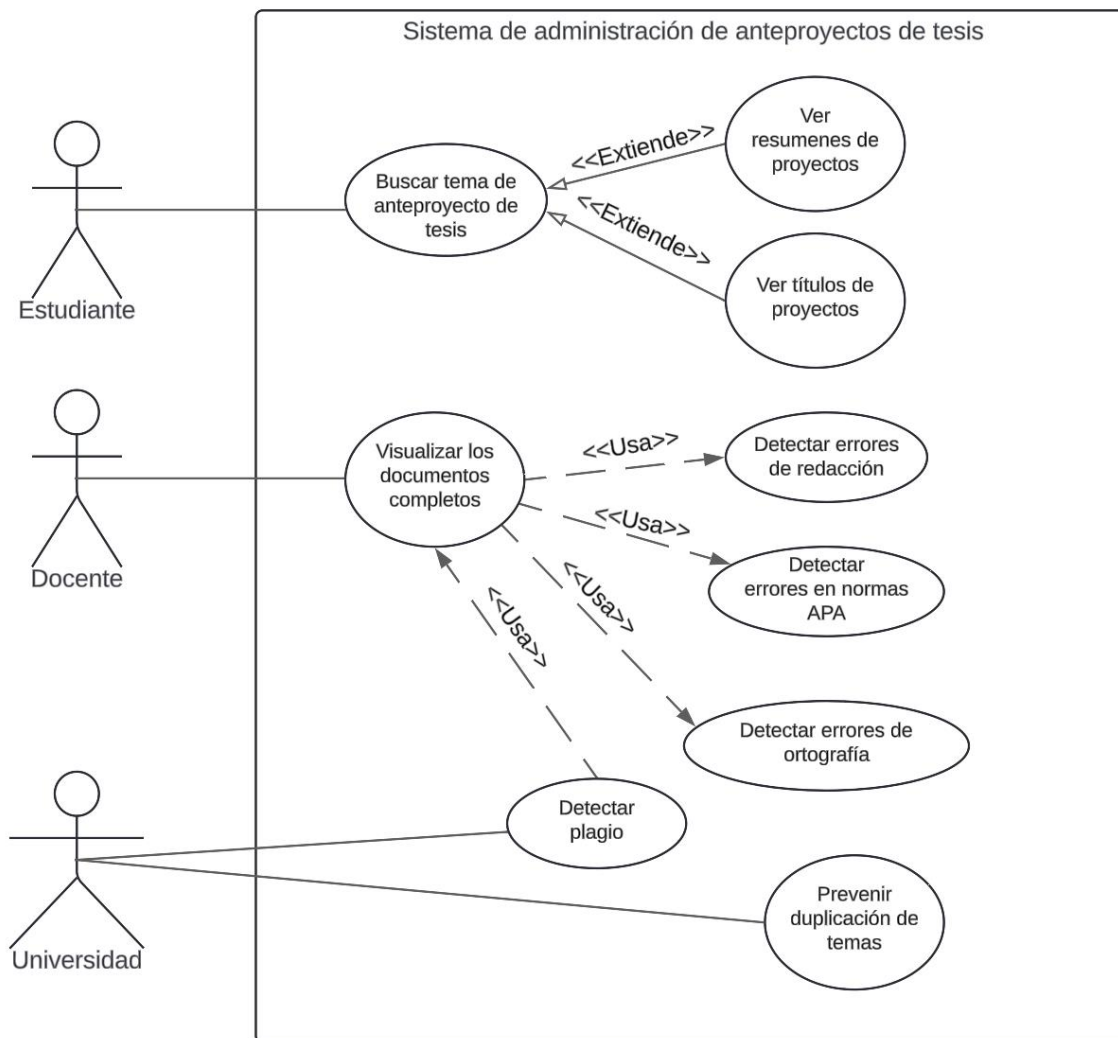
Alexandra (2023) explica que la metodología RUP es un proceso sistemático y organizado que asigna roles, responsabilidades y tareas a los integrantes del equipo. Su propósito es asegurar la calidad del producto mientras se cumple con el cronograma y el presupuesto establecidos, para satisfacer las necesidades del cliente. La metodología RUP abarca más de 30 perfiles, más de 20 actividades y más de 70 artefactos.

Para explicar la funcionalidad del sistema y sus actores, se utiliza el diagrama de casos de uso. Un diagrama de casos de uso ilustra la interacción entre los actores y los casos de uso del sistema. Este diagrama representa la funcionalidad del sistema en términos de su interacción externa (Grau & Segura, s. f.).

El sistema académico para la gestión de anteproyectos de tesis lleva 2 personajes principales a los que va dirigido. En la Figura 3 se puede ver a dos actores y la manera en que interactúan con el sistema.

Figura 3

Diagrama de casos de uso general del sistema de administración de anteproyectos de tesis



En la figura se pueden ver 3 actores, en los cuales en cada uno se define su funcionalidad. Cabe destacar que existen ocasiones en las que se usa la palabra “Usa” y “Extiende”, las cuales son las relaciones de casos de uso.

- Usa: Cuando un caso hace uso de otro.

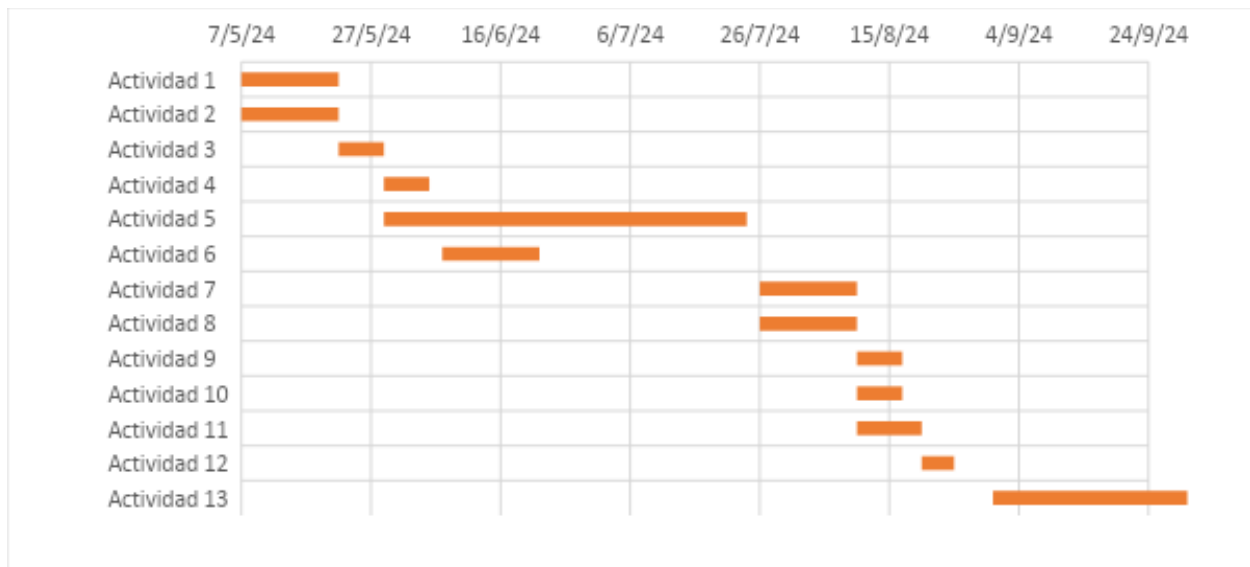
- Extiende: Cuando un caso de uso añade funcionalidades adicionales a otro caso de uso (Grau & Segura, s. f.).

1.13 Planificación de capítulos

La investigación tiene algunos puntos que será de utilidad explicar en una división de capítulos. Para entender un poco más sobre las actividades a realizar en la investigación, el pronóstico del cronograma se indica en la Figura 4.

Figura 4

Cronograma preliminar de actividades



Las actividades de la Figura 4 explican su descripción en la Tabla 2. De esta manera se brinda la información de los capítulos y de las actividades de la investigación.

Tabla 2

Actividades preliminares de la investigación

Actividad	Descripción
Actividad 1	Capítulo II: Marco Teórico
Actividad 2	Análisis del desarrollo
Actividad 3	Capítulo III: Marco Administrativo
Actividad 4	Capítulo IV: Análisis del desarrollo y diseño del sistema
Actividad 5	Desarrollo
Actividad 6	Capítulo V: Desarrollo
Actividad 7	Pruebas del sistema y desarrollo de manual de usuario

Actividad 8	Capítulo VI: Pruebas del sistema
Actividad 9	Índice tentativo
Actividad 10	Bibliografía
Actividad 11	Conclusiones y recomendaciones
Actividad 12	Anexos
Actividad 13	Defensa de tesis

1.14 Estimación de recursos

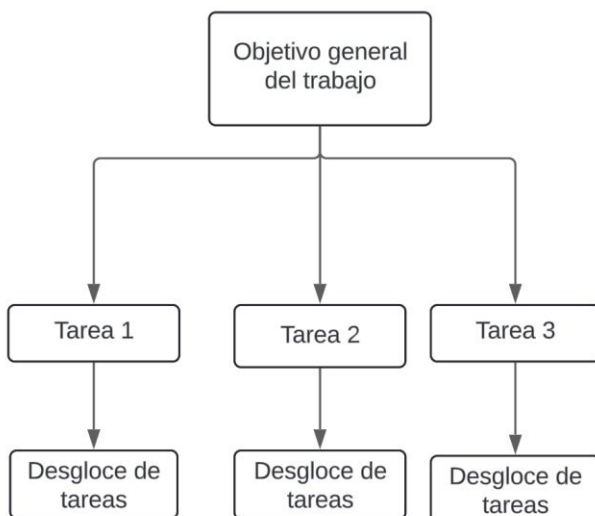
En esta fase de planificación de un proyecto, se especifican los trabajos específicos para cada tarea y se identifican los recursos necesarios para llevarlos a cabo (Raya, 2013, p. 105). Para la planificación de recursos se utilizará un diagrama RACI.

El diagrama RACI es una matriz de asignación de responsabilidades que emplea las categorías de responsable, encargado, consultar e informar (*Responsible, Accountable, Consult, Inform*) para determinar la participación de los interesados en las actividades del proyecto (PMI, 2023).

Para la definición de las actividades del trabajo se utilizará la herramienta EDT, la cual, según Puentes et al. (2022), representa el trabajo a realizar de los entregables o resultados tangibles (Puentes et al., 2022). En la Figura 5 se visualiza cómo se estructurará el EDT.

Figura 5

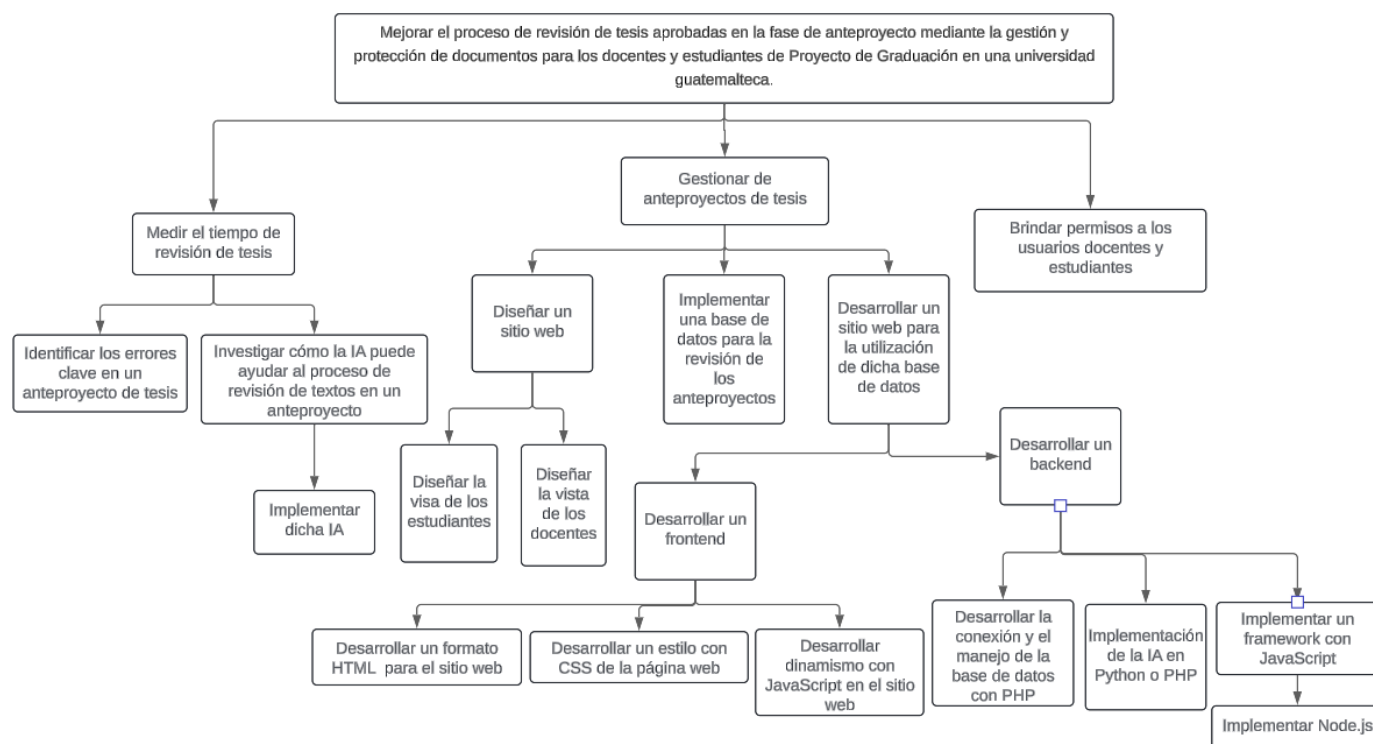
Ejemplo de Estructura de Desglose de Trabajo (EDT) del proyecto



En la Figura 6 se puede ver cómo se desglosan las tareas del objetivo, y de algunas tareas, surgen tareas hijas. De esta manera ya están identificadas las actividades EDT del proyecto.

Figura 6

Estructura de Desglose de Trabajo (EDT) del proyecto



Para la asignación de responsabilidades, se utilizó un diagrama RACI, tal como se indicó anteriormente. Este método fue ejemplificado por Hernández et al. (2014, p. 258). Además, se realizó la asignación de recursos materiales en términos de gastos y tiempo estimado, tal como fue ejemplificado por Raya et al. (2013, p. 108).

En la Tabla 3 se muestran los recursos materiales estimados, el tiempo aproximado de trabajo requerido y el costo económico. En Guatemala, existen cursos en línea con un costo aproximado de Q100.00 y libros digitales que pueden costar alrededor de \$10.00. Asimismo, los libros físicos en librerías guatemaltecas pueden tener un costo aproximado de \$200.00. Se estimó un presupuesto de Q500.00 para materiales necesarios que requieren refuerzo para el proyecto, siendo este un gasto único. Además, el costo del hosting y dominio web se estimó en aproximadamente Q300.00 al año utilizando la herramienta HostGator.

Tabla 3*Planificación de costes del proyecto con detalle de los recursos a utilizar*

Actividad EDT	Recurso	Duración estimada	Importe
Medir el tiempo de revisión de tesis	Internet, cursos y libros para investigar sobre AI	3 semanas	Q.500.00
Gestionar anteproyectos de tesis incluyendo programación, desarrollo de IA y una bases de datos	Software de hosting para una página web y su base de datos	4 semanas	Q.300.00 cada/año
Brindar permisos a los usuarios docentes y estudiantes	Internet, cursos y libros para implementar seguridad para los usuarios	3 semanas	Q.500.00

Para la asignación de responsabilidades del anteproyecto de tesis de la Tabla 4, se considerará la evaluación continua por parte de los usuarios y del docente del curso de Proyecto de Graduación. Tanto los docentes como los estudiantes utilizarán el sistema; sin embargo, el docente encargado de la revisión de tesis proporcionó los requerimientos del sistema y será responsable de validar su funcionamiento, así como de aprobar o desaprobado las funciones.

Tabla 4*Matriz de roles y responsabilidades según el modelo de diagrama RACI*

Actividad	Desarrollador	Docente del curso dueño del producto	Estudiantes	Docentes de PG1	Universidad
Medir el tiempo de revisión de tesis	R	C	I	A	A
Gestionar anteproyectos de tesis incluyendo programación, desarrollo de IA y una bases de datos	R	C	I	I	A
Brindar permisos a los usuarios docentes y estudiantes	R	C	I	C	A

2. Capítulo 2 – Marco Teórico

En el contexto actual de la educación universitaria para la facultad de Ingeniería en Sistemas, la gestión eficiente de los anteproyectos de tesis y la garantía de originalidad se han convertido en pilares fundamentales para asegurar la calidad académica y la integridad intelectual. El creciente acceso a recursos digitales y la necesidad de combatir el plagio de manera efectiva, surge la necesidad de desarrollar herramientas que no solo faciliten la gestión de proyectos, sino que también integren tecnologías avanzadas como la inteligencia artificial para fortalecer los procesos de revisión y detección de duplicidad.

Este marco teórico explora diversos aspectos, desde fundamentos de la seguridad informática en aplicaciones web hasta el diseño robusto de bases de datos y la aplicación de algoritmos avanzados para la detección precisa de plagio. Además se analizan las últimas tendencias en el uso de inteligencia artificial en la educación, destacando cómo la optimización de procesos en revisión de anteproyectos de tesis, mejorando así la retroalimentación académica y garantizar la originalidad del trabajo académico.

Al profundizar en estos temas, este estudio no solo busca ofrecer una visión comprehensiva del estado del arte en gestión de anteproyectos de tesis, sino también sentar las bases teóricas y metodológicas para el desarrollo de una plataforma que aborde la administración de temas de tesis en el ámbito académico.

2. 1 Objetivos del Marco Teórico

1. **Explorar los fundamentos de la seguridad informática en aplicaciones web:** Analizar los principios y fundamentos de autenticación, autorización y encriptación, así como las amenazas comunes como la inyección SQL y el Cross-Site Scripting (XSS).
2. **Examinar estrategias para el diseño y gestión eficiente de base de datos:** Evaluar técnicas de modelado de datos, sistemas de gestión de bases de datos y optimización de consultas para el rendimiento y la escalabilidad.
3. **Investigar métodos y herramientas avanzadas de detección de plagio:** Revisar algoritmos de comparación textual, análisis de n-gramas y herramientas como Turnitin y Grammarly para la detección precisa de plagio.
4. **Analizar aplicaciones emergentes de inteligencia artificial en la educación:** Explorar cómo la IA puede potenciar la revisión automática de textos académicos, ofrecer

recomendaciones personalizadas y mejorar la retroalimentación de los docentes a los estudiantes.

2.2 Definición y conceptos básicos

Para comenzar a analizar el tema del proyecto, es importante conocer los conceptos en los que éste se basa, tales como una página web, la gestión de proyectos y la Inteligencia Artificial.

2.2.1 Definición de una página web

Una página web es un documento electrónico accesible en internet que presenta información de manera visual y organizada. Las páginas web pueden tener diversos fines, como brindar información, promocionar productos o servicios, compartir contenido multimedia, interactuar con usuarios y facilitar la comunicación y colaboración en línea. La creación de una página web puede variar en complejidad: desde opciones sencillas, como los sistemas de gestión de contenido (CMS) que proporcionan plantillas predefinidas y requieren poco conocimiento técnico (GoDaddy, 2023).

2.2.2 Tipos de páginas web

Páginas web dinámicas: Una página web dinámica es un conjunto de páginas cuyo contenido cambia según la ubicación de los visitantes, acciones pasadas realizadas en el sitio, zonas horarias y más. Además de HTML, CSS y JavaScript, un sitio web dinámico usa un lenguaje de scripting del lado del servidor como PHP o Python. Esto activa la conexión con la base de datos para permitir funciones interactivas y cambios de contenido (Mora, 2023).

Páginas web estáticas: Un sitio estático consta de páginas web que siempre se ven iguales cada vez que los visitantes las acceden. Se puede crear usando HTML, CSS y JavaScript. Aunque son más rápidos y fáciles de construir que los sitios dinámicos, ofrecen una funcionalidad más limitada. A pesar de estas limitaciones, los sitios estáticos son muy populares para usos como portafolios y currículos (Infante, 2023).

2.2.2 Gestión de proyectos

La gestión de proyectos consiste en coordinar procesos, herramientas, miembros del equipo y habilidades para entregar proyectos que cumplan con los objetivos y requisitos establecidos. Además, empodera al equipo para completar proyectos al alinearlos con objetivos claros,

aumentando la transparencia y visibilidad, facilitando la comunicación y definiendo el alcance del proyecto (Atlassian, s. f.).

Ámbito académico: La gestión de proyectos puede ser una herramienta valiosa tanto para educadores como para estudiantes al planificar, organizar y llevar a cabo proyectos de manera efectiva y eficiente. Con el uso adecuado de conocimientos, habilidades, herramientas y técnicas específicas, es posible alcanzar metas y objetivos dentro de un alcance, presupuesto y calendario definidos.

En el ámbito educativo, al diseñar un curso, es fundamental establecer objetivos de aprendizaje claros, crear calendarios, identificar recursos y evaluar los resultados obtenidos. Los proyectos grupales y las tareas de investigación ofrecen oportunidades para el trabajo colaborativo, donde es esencial establecer objetivos, asignar responsabilidades, definir plazos y monitorear el progreso.

Además, la gestión de proyectos puede aplicarse para la planificación y ejecución de eventos educativos como excursiones, talleres y conferencias. Esto requiere la elaboración de calendarios, coordinación logística y el trabajo con briefings detallados (*¿Cómo debe utilizarse la gestión de proyectos en la formación y la educación?*, s. f.).

2.2.3 Inteligencia artificial

La inteligencia artificial (IA) engloba un conjunto de tecnologías que permiten a las computadoras llevar a cabo diversas funciones avanzadas, como la capacidad de percepción visual, comprensión y traducción del lenguaje tanto hablado como escrito, análisis de datos, recomendaciones, entre otras. Es fundamental en la innovación computacional moderna al generar valor tanto para individuos como para organizaciones. Un ejemplo de esto es el reconocimiento óptico de caracteres (OCR), donde la IA se utiliza para extraer texto y datos de imágenes y documentos, convirtiendo contenido no estructurado en datos organizados que pueden ser utilizados por empresas, proporcionando información estadística valiosa (*¿Qué es la inteligencia artificial o IA?*, s. f.).

2.3 Estado del Arte en Herramientas para la Administración de Anteproyectos de Tesis

El crecimiento de estas soluciones se debe, en parte, a la creciente accesibilidad de los recursos digitales y a la necesidad de abordar adecuadamente la cuestión del plagio. En opinión del lector, la mejor legislación no puede detener al 100% a los estudiantes que deciden copiar y pegar todo su trabajo de investigación en línea. Como resultado, la tecnología ha permitido prácticas más eficientes y de alta calidad para monitorear y mejorar los resultados y la integridad del proceso de la tesis. Este apartado esbozará el estado actual de estas resoluciones basadas en la evidencia.

2.3.1 Turnitin

Turnitin es una herramienta utilizada en entornos académicos para la detección de similitudes y plagio en trabajos escritos. Funciona mediante un sistema de comparación textual que analiza documentos para identificar coincidencias con contenido existente en su base de datos, así como con recursos en línea y otros documentos previamente cargados. Los resultados se presentan en informes detallados que destacan áreas específicas donde se encontraron similitudes, proporcionando a educadores y estudiantes una herramienta para promover la integridad académica y mejorar la calidad del trabajo escrito.

Funcionalidades de Turniting: Esta herramienta ofrece funcionalidades que pueden ser muy útiles para que los docentes puedan integrarla en sus clases, como lo son:

- **Borradores y múltiples revisiones:** Permite revisar trabajos de investigación como monografías o tesis con evaluaciones continuas de cada sección entregada, facilitando la prevención del plagio y proporcionando retroalimentación durante el proceso de elaboración.
- **Entregas finales:** Garantiza la protección de los derechos de autor de los estudiantes y previene la colusión, asegurando la originalidad de los trabajos entregados.
- **Evaluación por pares o co-evaluación:** Habilita a los estudiantes para leer, revisar y evaluar documentos enviados por sus compañeros de manera anónima o atribuida, fomentando la colaboración y el aprendizaje entre pares.

- ***Grade anything* o Evaluación de archivos sin texto:** Permite la evaluación de entregas que incluyen presentaciones, fotografías, dibujos, gráficos, entre otros formatos, ampliando las posibilidades de evaluación más allá del texto escrito (Turnitin, 2019).

2.3.2 Plag

Plag es una plataforma en línea diseñada para la detección y prevención del plagio, asegurando la autenticidad y originalidad de los contenidos escritos. Utilizando algoritmos avanzados y bases de datos extensas, la plataforma escanea textos en busca de similitudes con fuentes en línea y materiales publicados. Ofrece un conjunto completo de herramientas para eliminar el plagio y revisar la gramática, mejorando así la calidad y precisión de la escritura en general. Es útil para estudiantes, profesores, escritores y empresas al mitigar el riesgo de enfrentar complicaciones legales asociadas con el plagio. En resumen, Plag es una herramienta indispensable para preservar la integridad académica y profesional de los trabajos.

Protección de datos y documentos en Plag: En Plag, se prioriza la protección de los datos y documentos personales de los usuarios como un principio fundamental. La política asegura la exclusividad del usuario sobre sus documentos y datos personales, prohibiendo estrictamente cualquier forma de uso para copiar o distribuir los documentos cargados. Además, los documentos no se incluyen en una base de datos comparativa. Tanto los datos como el contenido de los documentos están protegidos por medidas legales sólidas. El acceso a esta información está restringido exclusivamente a los usuarios y empleados autorizados para proporcionar soporte al cliente (Plag, s.f.).

2.3.3 SafeAssign

SafeAssign es una herramienta que compara tareas enviadas de estudiantes con un conjunto de trabajos académicos para identificar secciones que sean similares en otros trabajos. Esta es una herramienta que evalúa la originalidad de las entregas de tareas y ayudar a los estudiantes a utilizar correctamente las citas y poder hacer uso de otras fuentes en sus tareas.

Las bases de datos que utiliza SafeAssign se basa en un algoritmo único de coincidencia de texto que puede identificar copias exactas o inexactas del material de origen. Esta herramienta compara textos desde varias bases de datos:

- **Base de datos de referencia global:** más de 15 millones de artículos que estudiantes de varias instituciones clientes de Blackboard han enviado para ayudar a evitar copia entre instituciones.
- **Archivos de documentos institucionales:** artículos enviados a SafeAssign por personas de la institución del estudiante.
- **Internet:** SafeAssign busca en una amplia gama de fuentes en Internet mediante un servicio de búsqueda interno para encontrar similitudes de texto.
- **Base de datos de revistas ProQuest ABI/Inform:** Más de 3000 títulos de publicaciones, 4,5 millones de documentos y más de 200 categorías temáticas desde la década de 1970 (*SafeAssign*, s. f.).

2.4 Seguridad informática en aplicaciones web

La seguridad informática en aplicaciones web es un aspecto muy importante en el desarrollo y en la implementación de sistemas que gestionan proyectos académicos. Actualmente existe un aumento de plataformas digitales para la presentación y revisión de trabajos estudiantiles; Sin embargo, es esencial garantizar que estas aplicaciones sean seguras y estén protegidas contra diversas amenazas. La protección de datos personales y sobre todo mantener la integridad en el sistema educativo.

2.4.1 Principios básicos de seguridad informática

La ciberseguridad, seguridad informática o seguridad de la información se define como un conjunto de políticas, procedimientos y medidas para proteger la información. Los principios de la seguridad informática se utilizan para garantizar la misma.

1. **Integridad de la información:** Se refiere a que la información que se encuentra almacenada no se ha manipulado por terceros de manera malintencionada. Esto quiere decir que la información no será manipulada por personas que no están autorizadas.
2. **Disponibilidad de la información:** Se refiere a que la información debe estar disponible en todo momento a las personas autorizadas para acceder a ella, de igual manera que ésta pueda recuperarse en caso de un incidente de seguridad que pueda corromperla. Esto quiere decir que la información esté disponible cuando sea necesario.

3. **Autenticidad:** Se refiere a garantizar la legitimidad de la información de una organización o autor de ésta. Se debe ser capaz de comprobar que el usuario o la persona que autoriza y firma sea el autor original, evitando así que un hacker consiga suplantar la identidad de un autor.
4. **Confidencialidad de la información:** Conocida también como la privacidad. Esto hace referencia a que la información solo debe ser conocida por las personas que necesitan conocerla y que han sido autorizadas para (UNIR, s. f.).

2.4.2 Amenazas Comunes

La seguridad web es fundamental para el desarrollo y mantenimiento de sitios y aplicaciones web. El artículo de Comillas Ciberseguridad (2024) explora algunas vulnerabilidades comunes, explicando su funcionamiento y cómo pueden ser explotadas.

Inyecciones SQL

La inyección SQL es un tipo de ciberataque en el que se intenta introducir código malicioso en un sitio web para vulnerar su sistema de seguridad. Este ataque corrompe la estructura del código con el objetivo de encontrar datos confidenciales e información sensible de la víctima (NordVPN, 2023).

Si los desarrolladores web no son meticulosos al crear un sitio, podrían dejar una abertura para que alguien provoque efectos inesperados en la base de datos correspondiente. Las inyecciones SQL ocurren cuando el hacker introduce o inyecta código SQL malicioso en el sitio web. Este tipo de malware, conocido como carga útil, consigue que la consulta se realice en la base de datos como si se tratase de una legítima (Avast, s. f.).

En este tipo de amenaza, el atacante utiliza campos de entrada de datos, como en muchas ocasiones se encuentran en formularios o URLs. Este problema puede surgir principalmente por una validación escasa en las entradas del usuario.

Por ejemplo, en el inicio de sesión de un sitio web que solicita un nombre de usuario y una contraseña, un atacante puede introducir un nombre de usuario junto con una parte de una consulta SQL y un comando. Esto puede hacer que el servidor ejecute la consulta como si el atacante fuera un usuario válido, permitiendo el acceso sin necesidad de una contraseña (Comillas Ciberseguridad, 2024).

Existen algunos métodos comunes de inyección SQL que los hackers utilizan para introducirse en la base de datos de un sitio web. Estos métodos fueron descritos en el artículo de Avast (s. f.):

- **Inyección de SQL mediante la introducción de datos del usuario:** La forma más común y sencilla de perpetrar un ataque de inyección SQL es mediante la introducción de datos por parte del usuario. Muchos sitios web recopilan estas entradas y las transmiten al servidor. Por ejemplo, al hacer un pedido en línea y proporcionar una dirección, este dato se recopila. Un ejemplo sencillo en la vida real sería si un candidato a un empleo, en lugar de escribir su nombre "Juan González", escribiera "Contratar a Juan González". Al decirlo en voz alta en el departamento de RR.HH., podrían enviarle una oferta de empleo a Juan.
- **Inyección de SQL mediante variables de servidor:** Esto ocurre cuando se introduce un enlace de un sitio web en un navegador, el cual tiene una rápida secuencia de comunicación cuyo propósito es ofrecer el sitio al usuario. En este proceso de comunicación, el navegador solicita una lista de datos para que el sitio se renderice correctamente, los cuales son las variables de servidor. Este proceso consiste en introducir sigilosamente código sql en las solicitudes del navegador, las cuales se pueden inyectar en la base de datos del sitio web.
- **Inyección mediante la modificación de cookies:** Las cookies son archivos que residen en el navegador y hacen que los sitios web tengan accesibilidad a la información de los usuarios. Son útiles cuando los usuarios no recuerdan sus credenciales de acceso; Sin embargo, hay sitios que emplean cookies para seguir las actividades del usuario en más sitios. Los atacantes podrían manipular o envenenar las cookies de manera que cuando transmitan información al servidor del sitio, envíen consultas de código SQL a la base de datos.

Los ataques de inyección SQL se realizan de distintas maneras, pero las consecuencias son aún más impactantes, como pérdida de dinero de usuarios o incluso robo de identidad.

Cross-Site Scripting (XSS)

Es una vulnerabilidad de seguridad que aprovecha la falta de mecanismos de filtrado en los campos de entrada y permiten la inyección en páginas web vistas por el usuario en lenguajes script, como JavaScript. Los ataques de XSS más graves pueden incluir la divulgación de cookie de sesión del usuario, divulgación de archivos, instalación de programas de caballo de Troya, redirigir al usuario a otro sitio o página web (Cross-Site Scripting, s. f.).

Según (*Types of XSS / OWASP Foundation*, s. f.) los tipos y categorías de vulnerabilidades de cross-site scripting (XSS) son tres, identificados hasta 2005:

- **Reflected XSS (AKA Non-Persistent or Type I):** Ocurre cuando los datos introducidos por el usuario son inmediatamente devueltos con un mensaje de error, un resultado de búsqueda u otra respuesta que incluya los datos ingresados por el usuario total o parcialmente. Esto sucede sin que los datos devueltos sean seguros.
- **Stored XSS (AKA Non-Persistent or Type II):** Generalmente ocurre cuando los datos introducidos por el usuario se almacenan en el servidor objetivo, como en una base de datos, en un foro de mensajes, en registros de visitantes, campos de comentarios, etc. Un usuario puede recuperar los datos almacenados desde la aplicación web sin que estos sean seguros para renderizar desde el navegador.
- **DOM Based XSS (Type 0):** El DOM Based XSS es un tipo de ataque de cross-site scripting (XSS) que ocurre en el "Document Object Model" (DOM) del navegador del usuario. El DOM es una estructura jerárquica que representa el contenido de una página web y se puede manipular mediante scripts del lado del cliente, como JavaScript. En un ataque DOM Based XSS, el payload del ataque (es decir, el código malicioso) se ejecuta al modificar el entorno DOM del navegador del usuario víctima.

2.4.3 Buenas Prácticas y Estándares

Las buenas prácticas y estándares de seguridad ayudan a prevenir ataques y garantizar que las aplicaciones funcionen de manera confiable. Entre los estándares y prácticas están el OWASP Top Ten, HTTPS y Content Security Policy (CSP).

En esta sección se exploran prácticas y estándares, destacando su relevancia en plataformas de gestión de proyectos académicos y sistemas de detección de plagio.

OWASP Top Ten: El Proyecto Abierto de Seguridad de Aplicaciones Web (OWASP) publica una lista de las diez vulnerabilidades de seguridad más críticas en aplicaciones web. Este recurso es muy útil para desarrolladores que desean aprender sobre las amenazas más comunes y cómo mitigarlas. La implementación de las recomendaciones del OWASP Top Ten 2021 puede ayudar a proteger las aplicaciones contra los ataques más frecuentes y peligrosos y se mueven de lugar conforme un estudio de ataques.

- **A01:2021 - Control de Acceso Roto (Broken Access Control):** Estos fallos ocurren cuando no se gestionan correctamente los permisos y accesos de usuario. Esto permite que personas restringidas accedan a la información, la manipulen o realicen acciones que están prohibidas.
- **A02:2021 - Fallos Criptográficos (Cryptographic Failures):** Ocurre cuando las implementaciones criptográficas en una aplicación son vulnerables, lo que puede llevar a la publicación de datos sensibles y comprometer la integridad del sistema.
- **A03:2021- Inyección:** Esta categoría abarca vulnerabilidades en los cuales los datos no confiables son enviados a un intérprete como parte de comando o consulta. Esto puede llevar a la ejecución de código no autorizado.
- **A04:2021- Diseño inseguro:** Esta categoría tiene las vulnerabilidades derivadas de decisiones de diseño inseguras o inadecuadas en sistemas y aplicaciones web, tales como errores de arquitectura, diseño de software, los cuales pueden ser explotados por los atacantes.
- **A05:2021- Configuración incorrecta de la seguridad:** Esta categoría aborda las vulnerabilidades resultantes de configuraciones incorrectas o inseguras en sistemas y aplicaciones web. Este tipo de errores puede exponer información sensible de los usuarios del sistema.
- **A06:2021- Componentes vulnerables y desfasados:** Identifica riesgos relacionados con el uso de componentes de software obsoletos, desactualizados o vulnerables en aplicaciones web; Estos pueden ser bibliotecas, módulos, *plugins* o *frameworks* que no han sido actualizados con las últimas correcciones de seguridad.
- **A07:2021-Fallas en la identificación y autenticación:** Identifica los riesgos de seguridad derivados de prácticas deficientes en la identificación y autenticación de usuarios dentro de las aplicaciones web. Las debilidades en este ámbito pueden permitir a los atacantes eludir los mecanismos de seguridad diseñados para proteger los datos y funcionalidades sensibles de la aplicación.
- **A08:2021-Fallas de la integridad de los programas informáticos y de los datos:** Aborda los riesgos de seguridad que surgen cuando no se mantienen la integridad y la confiabilidad del software y los datos en una aplicación web. Las vulnerabilidades en este ámbito pueden permitir a los atacantes modificar o corromper datos críticos, manipular el funcionamiento normal de la aplicación y comprometer la integridad de los sistemas en general.

- **A09:2021-Fallas en el registro y la supervisión de la seguridad:** Subraya la necesidad crítica de implementar y mantener registros de seguridad robustos y un monitoreo efectivo en las aplicaciones web para detectar y responder rápidamente a las amenazas de seguridad.
- **A10:2021-Falsificación de las solicitudes del lado del servidor:** Destaca la importancia de proteger adecuadamente las solicitudes realizadas desde el servidor para evitar manipulaciones maliciosas que podrían comprometer la seguridad y la integridad de los sistemas web (OWASP Top 10: 2021, s. f.).

Capítulo 3 – Análisis y diseño

3.1 Dinámica de Estudiante y Profesor

El sistema está diseñado para diferenciar y gestionar las interacciones de los usuarios según su rol (estudiante o profesor). Esta distinción se logra mediante un proceso de autenticación y autorización, que captura los roles de los usuarios al momento de iniciar sesión y adapta la experiencia del usuario en función de su tipo.

3.1.1 Captura de Roles y Gestión de Sesión

- **Inicio de sesión:** Al ingresar al sistema, el usuario deberá autenticarse con su correo electrónico y contraseña. El sistema capturará automáticamente el rol del usuario a través de su perfil registrado en la base de datos.
- **Identificación de Roles:** El sistema identificará el rol del usuario y redirigirá a una interfaz personalizada:
 - **Estudiante:** Tendrá acceso a funcionalidades como la carga de documentos, consulta de similitud y acceso a estadísticas básicas de su rendimiento.
 - **Profesor:** Tendrá acceso a las herramientas administrativas, como la creación de sedes y cursos, la gestión de estudiantes y la visualización de estadísticas de similitud de los documentos entregados por los estudiantes.

3.2 Funciones para Estudiantes

Una de las funcionalidades clave de la plataforma es permitir a los estudiantes cargar y revisar sus trabajos académicos, tales como tesis y anteproyectos de tesis, con el objetivo de garantizar la originalidad de sus contenidos antes de entregarlos oficialmente. A través de este proceso, los estudiantes podrán subir un único archivo por tarea y someterlo a un análisis detallado de similitud, utilizando una herramienta avanzada de detección de plagio.

Una vez que el estudiante haya cargado su documento, el sistema ejecutará un análisis exhaustivo comparando el trabajo con una amplia base de datos de textos académicos y recursos en línea. El resultado de este análisis será un informe de similitud, que proporcionará un porcentaje que refleja el grado de coincidencia del documento con otras fuentes existentes. Además, el informe incluirá los fragmentos específicos del texto que presentan similitudes, permitiendo al estudiante conocer con detalle las secciones que podrían necesitar revisión o reescritura.

Por otro lado, tanto los estudiantes como los profesores tendrán la opción de buscar documentos previos en la plataforma y realizar un análisis de similitud con otros trabajos almacenados. Esta funcionalidad les permitirá comparar documentos y obtener una visión más clara de la originalidad y el nivel de coincidencia en los trabajos presentados. Sin embargo, el acceso a los resultados y los informes estará restringido según el rol del usuario, garantizando la privacidad y el control de la información.

3.3 Funciones para Profesores

La plataforma ofrece una serie de herramientas específicas para los profesores, facilitando la gestión académica de los cursos, la evaluación de tareas y el análisis de similitud de los trabajos entregados. A través de estas funcionalidades, los profesores podrán crear y administrar sedes y cursos, gestionar la inscripción de estudiantes, asignar tareas y analizar la originalidad de los documentos entregados.

En primer lugar, los profesores podrán crear sedes dentro de la plataforma, que actúan como espacios académicos virtuales donde se agrupan los cursos. Cada sede podrá contener varios cursos específicos, cada uno con un código único que los estudiantes utilizarán para inscribirse. Esta flexibilidad permitirá una organización eficaz de las clases y garantizará que los estudiantes puedan acceder únicamente a los cursos correspondientes.

Los profesores también podrán agregar estudiantes a sus cursos de manera manual o permitirles inscribirse autónomamente ingresando su código de alumno. El sistema verificará que el código ingresado sea válido, y si el estudiante es aceptado por el profesor, podrá ser inscrito en el curso y comenzar a interactuar con el contenido, así como entregar sus tareas.

En cuanto a la evaluación académica, los profesores tendrán la capacidad de asignar tareas a los estudiantes, revisar los documentos entregados, y proporcionar retroalimentación y calificaciones. Además, podrán acceder a un panel de estadísticas de similitud, que les permitirá observar el porcentaje de similitud de los trabajos entregados por los estudiantes. Este análisis será fundamental para evaluar la originalidad de los trabajos y detectar posibles casos de plagio, contribuyendo a mantener la integridad académica dentro del curso.

3.4 Interacción entre Roles

El sistema de inscripción de cursos está diseñado para facilitar el acceso al contenido educativo, asegurando que solo los estudiantes validados por el profesor puedan participar. Para inscribirse, el estudiante debe ingresar su código de alumno, el cual será verificado por el sistema para confirmar su identidad y su aceptación en el curso. Una vez completada esta validación, el estudiante podrá acceder al material del curso, entregar tareas y utilizar herramientas para evaluar la similitud de su trabajo con otros documentos.

Además, tanto estudiantes como profesores tienen la capacidad de explorar documentos previos y realizar análisis de similitud. Sin embargo, se establece una distinción importante: los profesores tienen acceso completo a las estadísticas de similitud de todos los trabajos entregados en su curso, lo que les permite realizar un seguimiento detallado de la originalidad y el cumplimiento de los estudiantes. Esta función asegura un proceso transparente y equitativo en la evaluación del trabajo académico.

3.5 Flujo de inscripción de estudiantes

El flujo de inscripción y gestión de documentos en el sistema: el proceso comienza cuando el profesor crea una sede y asigna un curso con un código único. El estudiante, por su parte, ingresa al sistema y busca el curso utilizando su código de alumno. El sistema valida que el código corresponda a un estudiante registrado en ese curso, y si la validación es exitosa, el profesor podrá aprobar la inscripción. Una vez aprobado, el estudiante podrá comenzar a cargar sus documentos, los cuales serán analizados automáticamente por el sistema.

3.6 Accesibilidad y seguridad

3.6.1 Autenticación de Usuarios

Solo los usuarios autenticados podrán realizar las acciones relacionadas con su rol. El sistema garantizará que no haya acceso no autorizado a los documentos o funciones.

3.6.2 Control de Acceso

Se implementarán controles de acceso a la información, de modo que un profesor solo pueda ver los documentos y estadísticas de sus propios estudiantes, y un estudiante solo podrá ver sus propios documentos y resultados de similitud.

3.7 Algoritmo de Similitud de Coseno

El propósito del proyecto es utilizar la similitud de coseno como una medida para comparar la similitud entre documentos textuales. Esta técnica es ampliamente utilizada en el procesamiento de lenguaje natural (NLP) y recuperación de información debido a su eficiencia y precisión para evaluar la relación entre dos vectores en un espacio multidimensional. La similitud de coseno calcula el ángulo entre dos vectores, proporcionando un valor entre 0 y 1, donde 0 indica que los documentos son completamente disímiles y 1 indica que son idénticos.

Para realizar este cálculo, se emplearán las siguientes herramientas y librerías:

- **sklearn:** Esta librería de Python ofrece diversas herramientas para el procesamiento de datos y machine learning, entre ellas un módulo para la transformación de texto en vectores numéricos y el cálculo de similitudes.
- **pdfvectorizer:** Esta librería está diseñada específicamente para convertir archivos PDF en vectores de características numéricas, permitiendo el análisis de contenido de documentos en formato PDF.

3.7.1 Preprocesamiento de los Documentos

El primer paso en el cálculo de similitud consiste en convertir los documentos de texto en vectores numéricos que puedan ser procesados por los algoritmos. En este proyecto, se empleará una herramienta de vectorización que permitirá extraer el contenido relevante de los documentos PDF, específicamente los elementos clave como la pregunta, el objetivo, la hipótesis y la justificación.

El sistema requerirá que los usuarios ingresen estos elementos manualmente, los cuales serán validados contra el contenido del documento para asegurar su precisión y coherencia. Una vez validada esta información, se realizará un análisis de similitud de estos cuatro factores (pregunta, objetivo, hipótesis y justificación) con todas las tesis almacenadas en la base de datos. El objetivo es calcular el porcentaje de similitud respecto al tema de investigación, garantizando así que no se dupliquen temas previamente trabajados.

Para estos análisis, se aplicarán técnicas como la tokenización y la vectorización mediante el método TF-IDF (Term Frequency-Inverse Document Frequency), lo que permitirá representar

tanto los datos ingresados como el contenido de los documentos en un formato adecuado para el cálculo de similitudes.

3.8 Base de datos

La base de datos diseñada para este proyecto incluye tres tablas principales que permiten organizar y gestionar los datos de manera eficiente. La base de datos tiene más tablas sin embargo estas 3 son el núcleo del funcionamiento del proyecto.

Tabla de Historial de Análisis

Esta tabla registra cada análisis realizado, manteniendo un historial completo de los resultados obtenidos para cada documento. Aunque el documento de tesis pueda ser actualizado y reemplazado, los registros previos del historial de análisis permanecen almacenados, asegurando la trazabilidad y continuidad de la información.

Tabla de Similitud de Factores

En esta tabla se almacenan los cuatro factores clave analizados (pregunta, objetivo, hipótesis y justificación) junto con el porcentaje de similitud calculado. Esta información es crucial para evaluar si el tema propuesto tiene similitudes significativas con otras tesis almacenadas en la base de datos, con el propósito de evitar duplicidades.

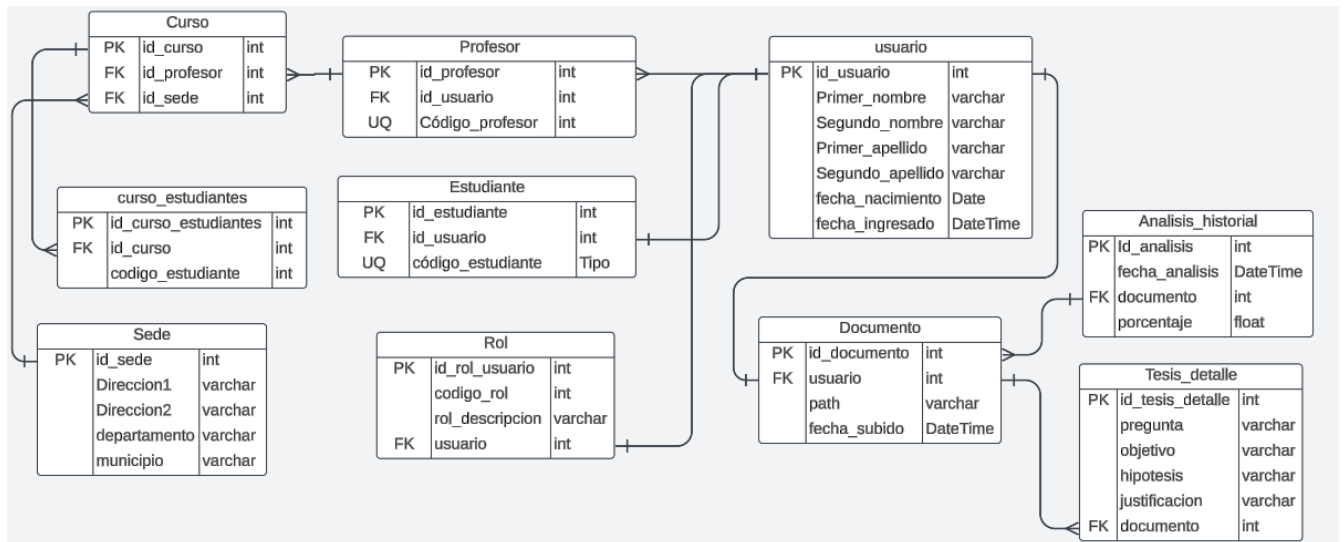
Tabla de Gestión de Documentos

Esta tabla contiene el nombre de la tesis, un identificador único (ID) y la ruta de almacenamiento (path) del documento en la VPS. Los documentos de tesis se guardan directamente en el servidor para garantizar su acceso seguro y centralizado.

Para una mejor comprensión de cómo funciona la base de datos, en la siguiente figura se puede visualizar el diagrama entidad relación.

Figura 7

Diagrama Entidad Relación para analisis de anteproyectos de tesis en una universidad



En la figura se presenta el modelo entidad-relación diseñado para la gestión y análisis de información en el sistema propuesto. Este modelo incluye entidades clave como Usuario, que almacena la información personal de los usuarios; Estudiante y Profesor, que identifican los roles específicos dentro del sistema; Documento, que registra los datos y la ubicación de los archivos subidos; y Análisis_Historial, donde se guarda un registro detallado de los análisis realizados, asegurando la trazabilidad incluso si los documentos son actualizados. Además, se encuentra la tabla Tesis_Detalle, que almacena información relacionada con los factores clave del análisis de similitud (pregunta, objetivo, hipótesis y justificación). Este modelo está estructurado para garantizar la integridad y organización de los datos, facilitando tanto su almacenamiento como su recuperación.

Capítulo 4. Desarrollo

Python es un buen lenguaje de programación si se busca facilidad para integración de tecnologías. Django es un framework de desarrollo web de código abierto escrito en Python. Está diseñado para facilitar la creación de aplicaciones web rápidas, seguras y escalables. Django sigue el principio "No repitas tu trabajo" (*Don't Repeat Yourself*, DRY) y utiliza el patrón Modelo-Vista-Controlador (MVC), aunque lo adapta a un esquema conocido como Modelo-Vista-Plantilla o *Template* en inglés (MVT).

Es importante resaltar que los fragmentos de código son ejemplos de cómo se implementó en el proyecto, ya que se pretende que se entienda el funcionamiento, pero mantener los derechos del autor.

Estos son los tres elementos que es fundamental para entender cómo se desarrolla el proyecto de similitud de anteproyectos de tesis.

Modelo: Es la representación de los datos y las reglas de negocio. En Django, los modelos se definen en clases dentro del archivo `models.py`, donde se especifican los campos y las relaciones de la base de datos.

Vista: La vista es responsable de procesar la lógica de la aplicación y devolver una respuesta, generalmente en forma de un objeto HTTP. En Django, las vistas se definen como funciones o clases dentro del archivo `views.py`.

Template: El template es responsable de la presentación de los datos, es decir, cómo se estructuran y muestran en el navegador. Django utiliza un sistema de plantillas basado en el lenguaje de plantillas Django (DTL), que permite integrar código Python en HTML.

4.1 Desarrollo del frontend

En el desarrollo de la aplicación web, uno de los retos fue la implementación de un diseño consistente en todas las páginas utilizando un template base. Para facilitar la reutilización y mejorar la mantenibilidad del código, se diseñó un archivo `base.html` que contiene la estructura común de todas las páginas, incluyendo el panel izquierdo de navegación. Este enfoque permite que las demás páginas hereden de `base.html`, lo que reduce la duplicación de código y mejora la escalabilidad del proyecto.

En la siguiente figura se puede visualizar como funciona el fragmento del archivo base.html:

Figura 8

Plantilla HTML base para todas las plantillas del proyecto

```
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>SDP</title>
7      <!-- Vinculación a Bootstrap -->
8      <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">
9  </head>
10 <body>
11     <div class="container-fluid">
12         <div class="row">
13             <!-- Panel izquierdo -->
14             <div class="col-md-3">
15                 <div class="sidebar">
16                     <ul class="nav flex-column">
17                         <li class="nav-item"><a class="nav-link" href="{% url 'home' %}">Menú principal</a></li>
18                         <li class="nav-item"><a class="nav-link" href="{% url 'formulario' %}">Buscar documentos</a></li>
19                         <li class="nav-item"><a class="nav-link" href="{% url 'about' %}">Configuración</a></li>
20                         <li class="nav-item"><a class="nav-link" href="{% url 'signout' %}">Salir</a></li>
21                     </ul>
22                 </div>
23             </div>
24
25             <!-- Contenido principal -->
26             <div class="col-md-9">
27                 {% block content %}
28                 <!-- Este bloque se sobrescribirá en las páginas hijas -->
29                 {% endblock %}
30             </div>
31         </div>
32     </div>
33
34     <!-- Scripts de Bootstrap -->
35     <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
36     <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.1/dist/umd/popper.min.js"></script>
37     <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
38 </body>
39 </html>
```

En este archivo base, hemos definido la estructura general de la página, que incluye un panel izquierdo con una barra de navegación y un área principal donde se inyecta el contenido específico de cada vista mediante el bloque `{% block content %}`.

Cada una de las páginas específicas de la aplicación hereda del archivo base.html y sobrescribe el bloque content para mostrar su propio contenido. Por ejemplo, una página que muestra un formulario podría tener el siguiente código:

Figura 9

Bloque de contenido utilizando el base.html

```
1 {% extends 'base.html' %}
2
3 {% block content %}
4 <h1>Formulario de Entrada</h1>
5 <form method="POST" action="{% url 'procesar_formulario' %}">
6     {% csrf_token %}
7     <div class="form-group">
8         <label for="campo1">Campo 1</label>
9         <input type="text" class="form-control" id="campo1" name="campo1">
10    </div>
11    <div class="form-group">
12        <label for="campo2">Campo 2</label>
13        <input type="text" class="form-control" id="campo2" name="campo2">
14    </div>
15    <button type="submit" class="btn btn-primary">Enviar</button>
16 </form>
17 {% endblock %}
```

4.2 Desarrollo del backend

En el desarrollo del backend del Sistema Web Detector de Plagio, como se menciona al inicio de este capítulo, se desarrolló con el framework Django junto con Python como lenguaje de programación principal. La estructura del backend se diseñó para manejar la carga y almacenamiento de documentos académicos y para realizar las solicitudes HTTP para el correcto funcionamiento de la interfaz de usuario.

4.2.1 Conexión de Django a SQL Server

Luego de configurar el servidor en Azure, el siguiente paso fue integrarlo a la aplicación. El paquete utilizado fue django-mssql-backend. Este paquete sirve para conectarse con SQL Server y provee soporte para Microsoft Server en proyectos desarrollados con Django. Este paquete fue esencial para facilitar la integración. En la figura 15 se muestra la configuración necesaria en el archivo settings.py que se crea en el entorno virtual

Figura 10

Configuración de la Base de Datos para Django

```

76
77 # Database
78 # https://docs.djangoproject.com/en/5.1/ref/settings/#databases
79
80 DATABASES = {
81     'default': {
82         'ENGINE': 'mssql',
83         'NAME': 'SDP_BDD',
84         'USER': '[REDACTED]',
85         'PASSWORD': '[REDACTED]',
86         'HOST': 'sistemadet[REDACTED]',
87         'PORT': '',
88         'OPTIONS': {
89             'driver': 'ODBC Driver 18 for SQL Server'
90         }
91     }
92 }
93

```

4.2.2 Método POST para enviar datos a las vistas

El método POST se utiliza para enviar datos desde los formularios a las vistas en Django. A continuación se muestra cómo se procesa un formulario en Django:

Figura 11

Método POST para obtener datos de los formularios de las plantillas

```

1 # views.py
2 from django.shortcuts import render, redirect
3 from django.http import HttpResponse
4
5 def procesar_formulario(request):
6     if request.method == 'POST':
7         campo1 = request.POST.get('campo1')
8         campo2 = request.POST.get('campo2')
9         # Realizar alguna operación con los datos recibidos
10        return HttpResponse(f'Datos recibidos: {campo1} y {campo2}')
11    return render(request, 'formulario.html')

```

4.2.3 Método cosine_similarity

En cuanto al cálculo de la similitud de coseno, se utilizó la librería scikit-learn para medir la similitud entre dos documentos. A continuación, se muestra un ejemplo del código en Python para realizar este procedimiento, así como la vectorización y conversión a un valor flotante redondeado a dos decimales:

Figura 12

Uso del método de cálculo de similitud de coseno

```
1  from sklearn.feature_extraction.text import TfidfVectorizer
2  from sklearn.metrics.pairwise import cosine_similarity
3
4  # Documentos de ejemplo
5  documento1 = "El rápido zorro marrón salta sobre el perro perezoso"
6  documento2 = "Un zorro veloz salta sobre un perro dormido"
7
8  # Vectorización de los documentos utilizando TF-IDF
9  vectorizer = TfidfVectorizer()
10 tfidf_matrix = vectorizer.fit_transform([documento1, documento2])
11
12 # Cálculo de la similitud de coseno
13 similitud = cosine_similarity(tfidf_matrix[0:1], tfidf_matrix[1:2])
14
15 # Conversión a float redondeado a 2 decimales
16 similitud_redondeada = round(similitud[0][0], 2)
17
18 print(f"La similitud de coseno entre los documentos es: {similitud_redondeada}")
```

En la figura se presenta un ejemplo de implementación para calcular la similitud entre documentos utilizando `TfidfVectorizer` de la librería `scikit-learn`. Este modelo transforma los documentos en vectores de características mediante la técnica TF-IDF (Term Frequency-Inverse Document Frequency) y, posteriormente, calcula la similitud de coseno entre los vectores usando la función `cosine_similarity`. Finalmente, el resultado obtenido es redondeado a dos decimales para facilitar su interpretación.

Capítulo 5. Pruebas

Durante el desarrollo del sistema de detección de plagio, surgieron varios errores que afectaron tanto el rendimiento de las vistas como las solicitudes HTTP. Uno de los primeros problemas fue el incorrecto renderizado de las vistas debido a una mala configuración de los templates. Las variables de contexto no se cargaban correctamente en las plantillas, lo que resultaba en páginas sin la información adecuada. Este error se solucionó asegurando que las vistas generaran las variables correctamente y que los nombres coincidieran entre las vistas y las plantillas.

Otro inconveniente relevante fue la gestión de las solicitudes POST, particularmente al validar formularios para la carga de archivos PDF. En varias ocasiones, los datos enviados no cumplían con las validaciones necesarias, lo que provocaba errores de tipo 400 (bad request). Este problema se resolvió ajustando las validaciones y mejorando la gestión de errores en las vistas. Además, trabajar con múltiples bases de datos presentó un desafío adicional al intentar asignar modelos específicos a bases de datos concretas. La solución fue unificar la configuración de la base de datos para garantizar un correcto funcionamiento del sistema.

5.1 Pruebas unitarias

5.1.2 Carga correcta de los archivos

Durante la implementación del sistema para la correcta subida de documentos, se desarrollaron formularios que permiten cargar la información en dos tablas de la base de datos. Estos formularios almacenaban de manera adecuada los datos requeridos, con excepción de un campo, específicamente el campo "sede", que estaba almacenando incorrectamente el nombre del autor en lugar de la sede correspondiente. Este error fue identificado y corregido, asegurando que los datos fueran almacenados correctamente en los campos correspondientes, evitando confusiones en el manejo de la información.

Adicionalmente, se evaluó cómo el sistema gestionaba la subida de la pregunta de investigación en los documentos de tesis. Se detectó un problema en la identificación automática de la pregunta de investigación, ya que la IA no lograba reconocerla en todos los proyectos, lo que requería intervención humana para su verificación. Para mejorar esta situación, se realizó una corrección generalizando los títulos y afinando el algoritmo para que la IA pudiera identificar de

manera más precisa si un texto era efectivamente una pregunta de investigación o si se trataba de otro contenido identificado erróneamente. Esto permitió aumentar la precisión de la IA, aunque sigue siendo necesario un nivel de supervisión humana.

5.1.2 Validación de formularios para crear estudiantes

En el proceso de validación de formularios, se implementó una verificación crucial relacionada con los estudiantes. Para que un usuario pueda crear una cuenta de estudiante, primero debe estar registrado en la base de datos como tal. Inicialmente, surgió un problema al crear un usuario con perfil de profesor, ya que este usuario tenía acceso a todos los permisos, incluyendo aquellos asignados a los estudiantes, lo que generaba conflictos en la administración de roles y permisos.

Este problema fue solucionado mediante una gestión adecuada de los permisos para cada tipo de usuario, asegurando que los profesores y estudiantes tuvieran accesos diferenciados según su rol. Ahora, el sistema permite la creación de cuentas de estudiante de manera correcta, respetando las restricciones establecidas, y también se ha implementado la funcionalidad para crear y administrar una lista de estudiantes, lo que facilita la gestión y organización de los mismos dentro del sistema.

5.1.3 Analisis de similitud de documentos

Durante las pruebas de análisis de similitud de documentos, se implementó la API de OpenAI, como se mencionó previamente en el capítulo 4. Esta integración permitía realizar comparaciones avanzadas entre documentos para identificar similitudes. Sin embargo, durante estas pruebas se detectó un problema en el proceso de identificación de la pregunta de investigación y los objetivos. El código estaba extrayendo información incorrecta, ya que incluía texto de apartados subsecuentes, lo que resultaba en una segmentación incorrecta de los datos clave.

Este error afectaba directamente la precisión del análisis de similitud, ya que la información incorrecta almacenada distorsionaba los resultados. Para solucionar este problema, se ajustó el código, afinando el proceso de extracción para asegurar que solo se capturara la pregunta de investigación y los objetivos reales, excluyendo cualquier información no relevante. Esto mejoró

la precisión del análisis de similitud, permitiendo obtener resultados más consistentes y útiles para la detección de plagio.

5.1.4 Casos de prueba

Los casos de prueba realizados incluyeron el uso de documentos repetidos para verificar tanto el proceso de subida de archivos como el análisis de similitud. En total, se realizaron 34 pruebas relacionadas con la carga de documentos, lo que permitió validar el correcto funcionamiento de esta funcionalidad en diversos escenarios, incluyendo la validación de datos y la detección de duplicados. Cada prueba sirvió para asegurar que el sistema gestionaba adecuadamente la información y que los documentos se almacenaban correctamente para el posterior análisis de similitud.

Adicionalmente, se llevaron a cabo 6 pruebas específicas para verificar la efectividad del sistema de inicio de sesión. Estas pruebas incluyeron 3 casos para la creación de cuentas de profesor, 2 para cuentas de estudiante y 1 para otro tipo de usuario. Cada prueba se enfocó en asegurar que el proceso de autenticación funcionara correctamente, permitiendo a los diferentes tipos de usuarios acceder y utilizar el sistema de acuerdo a los permisos asignados. Estas pruebas contribuyeron a garantizar la seguridad y diferenciación de roles dentro de la plataforma.

Capítulo 6. Pruebas de certificación

El propósito de las pruebas es validar que el sistema funciona correctamente en los aspectos críticos relacionados con la seguridad y la detección de plagio. Específicamente, las pruebas tienen como objetivo asegurar que las funcionalidades de autenticación de usuarios y autorización de accesos estén operando de manera adecuada, garantizando que solo los usuarios autorizados puedan acceder a determinadas secciones o documentos del sistema. Además, se busca verificar que los documentos cargados estén debidamente protegidos, tanto desde el punto de vista del acceso controlado como de la preservación de la integridad de los mismos, evitando manipulaciones o accesos no autorizados. De esta manera, las pruebas aseguran que el sistema ofrece un entorno seguro y confiable para la gestión de documentos y la detección eficiente de plagio.

6.1 Pruebas de funcionalidad

Durante las pruebas de funcionalidad, se identificaron dos problemas relacionados con la seguridad del sistema. En primer lugar, se detectó que las contraseñas de los usuarios no estaban encriptadas, lo que facilita el acceso no autorizado en caso de que las credenciales sean interceptadas. En segundo lugar, se comprobó que la gestión de permisos no estaba implementada correctamente, ya que cualquier usuario, independientemente de su rol o nivel de autorización, podía acceder y visualizar todos los documentos almacenados en el sistema. Esto representa una vulnerabilidad significativa, ya que el sistema no está restringiendo el acceso a los archivos de acuerdo con los permisos asignados, comprometiendo la privacidad y la integridad de los datos. Estos problemas deben corregirse para garantizar la seguridad y el correcto funcionamiento del sistema.

Durante las pruebas de funcionalidad del sistema de detección de plagio, se detectó un error significativo en el algoritmo de similitud. El problema radicaba en que, debido a un fallo de programación, el algoritmo comparaba las preguntas de investigación y objetivo de una misma tesis en lugar de comparar entre dos documentos diferentes. Como resultado, la similitud obtenida era extremadamente alta en todos los casos, lo que generaba falsos positivos y afectaba la precisión del análisis. Esta falla distorsionaba los resultados de la detección de plagio, ya que no reflejaba de manera correcta las comparaciones entre tesis distintas. Para corregir este problema se

analizaron los textos que obtuvo la IA y se detectó la línea problema, la cual se muestra en la figura 19.

Figura 13

Error detectado en el cálculo de similitud de tesis

```
for doc in todos_los_documentos:
    similitud_pregunta = calcular_similitud(documento.pregunta_investigacion, doc.pregunta_investigacion)
    similitud_resumen = calcular_similitud(documento.resumen, doc.resumen)
    similitud_total = (similitud_pregunta + similitud_resumen) / 2 # Podrías hacer un promedio ponderado

    similitudes.append({
        'documento': doc,
        'similitud': similitud_total
    })
```

Nota: En la segunda línea de código, se puede ver como en lugar de analizar dos preguntas diferentes, o dos textos diferentes, se estaba analizando el mismo documento.

6.2 Pruebas de seguridad

Durante las pruebas de inyección SQL realizadas en el sistema, se observó un comportamiento correcto al procesar entradas maliciosas en las URLs. En particular, el intento de inyección (32 OR 20) fue tratado como una URL mal formada y resultó en un error 404, lo que indica que el sistema no ejecutó ningún código SQL malicioso y simplemente rechazó la solicitud. Este resultado es positivo, ya que demuestra que Django está manejando adecuadamente los intentos de inyección en los endpoints definidos. Sin embargo, es recomendable realizar pruebas adicionales en otros puntos de entrada, como formularios o campos de búsqueda que interactúan directamente con la base de datos, para asegurar que también estén protegidos contra inyecciones SQL y que el sistema esté validando y escapando correctamente todas las entradas de usuario. Esto garantizará una mayor robustez en la seguridad del sistema.

6.3 Pruebas de carga

Durante las pruebas de carga del sistema, se evaluó cómo responde bajo un número elevado de solicitudes concurrentes, tanto en la creación de usuarios como en la subida y análisis de documentos PDF. Se observó que, en condiciones normales, la creación de usuarios puede tardar varios segundos, mientras que el proceso de subir un documento PDF y ejecutar el análisis de similitud puede extenderse a varios minutos, especialmente cuando el sistema maneja múltiples solicitudes de manera simultánea. Estas pruebas permitieron identificar los tiempos de respuesta

bajo carga, y si bien el sistema logra completar las solicitudes, los tiempos prolongados en ciertos procesos sugieren la necesidad de optimizar tanto la gestión de recursos como la eficiencia de los algoritmos utilizados para asegurar que el rendimiento no se degrade significativamente con un mayor volumen de usuarios y documentos.

6.4 Procedimiento de las pruebas

Para llevar a cabo las pruebas de carga y seguridad del sistema, se siguió un procedimiento estructurado que incluyó varias fases de análisis y la herramienta postman. A continuación, se detalla el proceso seguido.

Se utilizó Postman para realizar pruebas manuales de las API del sistema, incluyendo la autenticación de usuarios y la carga de documentos, simulando diferentes casos de uso.

Se configuró un entorno controlado para simular diferentes condiciones de uso, asegurando que las pruebas reflejaran situaciones reales de operación. Esto incluyó la configuración de un servidor local y el uso de bases de datos que replicaran el tamaño y estructura de las utilizadas en producción. El entorno también contaba con datos de prueba para verificar la correcta autenticación, subida de documentos y análisis de similitud.

6.4.1 Casos de prueba

- Se evaluó el rendimiento del sistema cuando varios usuarios subían documentos PDF de manera simultánea. Aquí se midió el tiempo necesario para completar la subida de un documento y activar el proceso de análisis de similitud. Se realizaron pruebas con 10 documentos diferentes al mismo tiempo.
- Se evaluó la detección de Similitud de prueba, la cual analizó la eficiencia del algoritmo de similitud bajo condiciones de alta carga, simulando múltiples documentos siendo analizados en paralelo. El objetivo era identificar si el tiempo de respuesta para realizar el análisis se incrementaba de manera significativa bajo una alta demanda y la diferencia de porcentajes que brindó el sistema.
- Se llevaron a cabo pruebas específicas para detectar vulnerabilidades de inyección SQL mediante la realización de consultas sencillas en la URL del sistema. Estas pruebas consistieron en intentar introducir consultas SQL maliciosas, como ' OR '1'=1, directamente

en los parámetros de la URL para evaluar si el sistema estaba validando correctamente las entradas y protegiendo las consultas a la base de datos. Se verificó que el sistema no ejecutara las consultas maliciosas y que las URLs generaran errores controlados, en lugar de permitir el acceso no autorizado o la manipulación de datos.

Cada uno de estos casos se monitorearon en los logs para detectar errores o alertas que pudieran indicar problemas de rendimiento o fallos de seguridad. Tras la ejecución de las pruebas, se recopilaron los resultados para cada caso de prueba. Se identificaron áreas de mejora en los tiempos de respuesta, especialmente en la creación de usuarios y en el análisis de documentos PDF bajo alta carga.

Capítulo 7. Implementación

7.1 Implementación del VPS

Para implementar el sistema de detección de plagio, se ha optado por utilizar una VPS en OVH Cloud, lo que permite garantizar un entorno controlado y escalable. El proceso de implementación comienza con la selección de un plan de VPS adecuado a las necesidades del proyecto, basándose en los requisitos mínimos y recomendados previamente definidos. Se seleccionó un sistema operativo compatible con el backend desarrollado en Django, como Ubuntu o Rocky Linux, y se configuró el acceso mediante SSH, creando un entorno seguro y controlado para la administración del servidor. También se aseguró la actualización del sistema y la creación de un usuario adicional con privilegios administrativos para evitar el uso directo del usuario root por motivos de seguridad.

Una vez establecida la conexión al servidor, se procedió a instalar las herramientas necesarias para el correcto funcionamiento del proyecto. Esto incluyó la instalación de Python y sus paquetes asociados mediante pip, con especial énfasis en la creación de un entorno virtual aislado para la gestión de dependencias. También se configuró la base de datos, optando por PostgreSQL o MySQL, según las necesidades del proyecto, y se ajustaron los parámetros necesarios en el archivo de configuración de Django para conectar la aplicación con la base de datos en el servidor.

Para el despliegue en producción de la aplicación Django, se utilizó Gunicorn como servidor WSGI, el cual permite manejar de manera eficiente las solicitudes de los usuarios. La integración con NGINX fue esencial para servir las solicitudes HTTP y los archivos estáticos de la aplicación, configurando ambos servicios para funcionar conjuntamente mediante un socket Unix. Se realizaron ajustes en la configuración de NGINX para redirigir adecuadamente las solicitudes al servidor Gunicorn y gestionar de forma óptima los recursos estáticos y media que requiere el sistema.

Finalmente, se estableció un sistema de seguridad adicional configurando un firewall mediante UFW, para garantizar que solo los puertos necesarios estuvieran abiertos al tráfico, como el puerto 80 para HTTP y el 443 para HTTPS. Además, se implementó un certificado SSL utilizando Let's Encrypt, lo que permitió garantizar comunicaciones seguras entre los usuarios y el

servidor. De esta manera, el sistema quedó listo para su funcionamiento, brindando una solución robusta y escalable capaz de manejar el procesamiento de documentos y la interacción con los usuarios de manera eficiente y segura.

7.1.1 Importar el proyecto a la VPS con Git

Para desplegar el proyecto Django en una VPS de OVH Cloud desde GitHub, primero se accede al servidor mediante SSH. Una vez dentro, es esencial actualizar el sistema e instalar las dependencias necesarias, como Python, pip, Git y virtualenv, lo cual garantiza un entorno adecuado para el desarrollo y la ejecución de la aplicación. Posteriormente, se clona el repositorio del proyecto desde GitHub utilizando Git, y se configura un entorno virtual para gestionar las dependencias del proyecto de manera aislada, instalando las librerías necesarias desde el archivo requirements.txt.

Con el entorno preparado, se configuran las variables de entorno que el proyecto necesita, como claves secretas o URLs de la base de datos. Luego, se ejecutan las migraciones de Django para configurar las tablas en la base de datos y se recopilan los archivos estáticos del proyecto. A continuación, se instala y configura Gunicorn para ejecutar la aplicación Django y se utiliza Nginx como servidor web para manejar las solicitudes HTTP y servir el sitio web. Se crea un archivo de configuración en Nginx para conectar Gunicorn con la aplicación, y se establece Gunicorn como un servicio del sistema para garantizar que la aplicación se ejecute en segundo plano de manera estable.

Finalmente, se asegura el sitio con un certificado SSL utilizando Let's Encrypt, lo que habilita el protocolo HTTPS para garantizar comunicaciones seguras entre los usuarios y el servidor. Este proceso asegura que el proyecto esté completamente desplegado, accesible y seguro, funcionando de manera eficiente en la VPS de OVH Cloud.

7.2 Implementación de la BDD desde Azure

Para la implementación de la base de datos del sistema de detección de plagio, se optó por utilizar el servicio Azure SQL Database, lo cual permite contar con una infraestructura escalable y confiable en la nube. El proceso comenzó con la creación de un recurso de base de datos único desde el portal de Azure, donde se seleccionó la opción de Single Database. Este tipo de base de datos relacional es ideal para aplicaciones con necesidades de alta disponibilidad y escalabilidad.

Dentro de la configuración inicial, se asignó un grupo de recursos ya existente, y se definieron tanto el nombre de la base de datos como las credenciales del servidor SQL asociado, lo que permitió gestionar de manera centralizada los recursos del proyecto.

En la configuración del servidor SQL, se estableció un plan de servicio adecuado para las necesidades del sistema, seleccionando un Service Tier estándar, que ofrece un balance entre rendimiento y costo, garantizando que la base de datos pueda manejar de manera eficiente las consultas relacionadas con los documentos y usuarios del sistema. Asimismo, se eligió una región geográfica cercana para reducir la latencia en el acceso a los datos y optimizar la experiencia de los usuarios finales.

Una parte crucial de la implementación fue la configuración de las reglas de firewall para controlar el acceso a la base de datos. Para proteger la integridad de los datos, se configuró el acceso únicamente desde direcciones IP autorizadas, asegurando que solo los equipos autorizados pudieran conectarse al servidor SQL. Esta medida de seguridad es esencial en entornos de producción para mitigar riesgos asociados a accesos no deseados.

Finalmente, se obtuvieron las credenciales de conexión proporcionadas por Azure, que fueron integradas en la aplicación Django para interactuar directamente con la base de datos. A través de un cliente SQL, como Azure Data Studio, se ejecutaron las migraciones necesarias para crear las tablas y relaciones del sistema, lo cual garantizó que la estructura de datos estuviera lista para almacenar los documentos y gestionar las solicitudes de los usuarios. Este proceso aseguró una configuración segura y óptima para la base de datos en la nube, permitiendo su integración eficiente en el sistema de detección de plagio.

7.3 Librerías necesarias para el funcionamiento del proyecto

Las librerías utilizadas en el proyecto son fundamentales para garantizar el correcto funcionamiento y la escalabilidad del sistema. Django es la piedra angular del backend, ya que permite desarrollar de manera rápida aplicaciones web robustas y escalables, y se combina con `djangorestframework` para la creación de APIs eficientes, lo que facilita la integración de servicios externos y el manejo de solicitudes REST. Por otro lado, el uso de `django-mssql-backend` es crucial para la integración con bases de datos Microsoft SQL Server, lo que proporciona una base de datos

relacional potente y confiable, permitiendo que el sistema maneje grandes volúmenes de información de manera eficiente.

Adicionalmente, se utilizan varias librerías para mejorar la funcionalidad y rendimiento del sistema. scikit-learn y scipy son esenciales para el procesamiento y análisis de datos, habilitando funcionalidades avanzadas como la similitud entre documentos y otras operaciones de machine learning. Librerías como requests y httpx permiten que el sistema realice solicitudes HTTP hacia APIs externas, asegurando una comunicación fluida. Por su parte, fuzzywuzzy se utiliza para la comparación de cadenas de texto, lo que es especialmente útil para el análisis de similitud entre documentos. Finalmente, herramientas como virtualenv ayudan a aislar las dependencias del proyecto, garantizando que las versiones de las librerías no generen conflictos en el sistema.

7.3.1 Lista de Librerías Relevantes:

1. Django
2. django-mssql-backend
3. djangorestframework
4. scikit-learn
5. scipy
6. requests
7. httpx
8. fuzzywuzzy
9. virtualenv

7.4 Implementación de Open AI en el proyecto

Para integrar la API de OpenAI en el sistema de detección de plagio, primero se debe registrar una cuenta en la plataforma de OpenAI y obtener una clave API, que permite autenticar las solicitudes desde el sistema hacia la API. Con esta clave, se configura el entorno de desarrollo agregándola a las variables de entorno del proyecto, garantizando la seguridad de la clave. Luego, se instala la librería oficial de OpenAI, lo que facilita la interacción entre el sistema y la API mediante solicitudes HTTP.

Una vez configurada la clave y la librería, se implementa una función que utiliza el modelo GPT-4 o GPT-3.5 para comparar textos. Esta función envía los fragmentos de documentos a la API, solicitando un análisis de similitud. La API devuelve una respuesta en formato JSON que contiene la información detallada sobre cuán similares son los textos, lo que puede ser procesado y presentado al usuario de manera comprensible.

Para gestionar errores, es importante implementar manejadores que capturen posibles fallas en la comunicación con la API, como límites en las solicitudes o problemas de red. También se puede optimizar el uso de la API enviando solo los fragmentos más relevantes de los documentos, lo que ayudará a reducir costos y mejorar la eficiencia en el procesamiento de textos largos.

Finalmente, si el sistema está desarrollado en Django, se puede implementar una vista que permita a los usuarios cargar documentos y comparar su similitud mediante una interfaz web. Esta integración de OpenAI en el sistema de detección de plagio proporciona una solución avanzada y eficiente para realizar análisis automáticos de similitud entre documentos, mejorando la capacidad del sistema para detectar casos de plagio.

7.4.1 Algoritmo de Similitud de Coseno con Open AI

El algoritmo de similitud del coseno es una herramienta comúnmente utilizada para medir la similitud entre dos vectores, especialmente en tareas de procesamiento del lenguaje natural. Al aplicarlo con un modelo de embeddings, como los que ofrece OpenAI, es posible comparar textos de manera eficiente. Los embeddings convierten palabras o frases en vectores en un espacio de alta dimensionalidad, donde términos similares están más cerca entre sí. Esto permite analizar documentos de manera profunda, capturando las relaciones semánticas entre los términos.

Una vez que se generan los embeddings de dos textos, la similitud del coseno mide el ángulo entre esos vectores. Si el ángulo es pequeño, los textos son similares, mientras que un ángulo cercano a 90 grados indica que no hay similitud. La fórmula matemática toma el producto punto de los dos vectores, dividiéndolo por el producto de las magnitudes de los mismos. El valor resultante varía entre -1 y 1, donde un valor más cercano a 1 indica una mayor similitud.

Para comparar dos textos, primero se obtienen sus embeddings mediante la API de OpenAI y luego se aplica la fórmula de similitud del coseno. Esto proporciona un valor numérico que permite evaluar cuán similares son los documentos o fragmentos de texto. Este enfoque es

especialmente útil en sistemas de detección de plagio, ya que permite comparar documentos no solo en función de coincidencias exactas, sino también por su significado o contexto subyacente.

En definitiva, la combinación de embeddings y similitud del coseno ofrece una forma robusta de analizar similitudes textuales, y su uso en aplicaciones como la detección de plagio facilita una evaluación más precisa, identificando tanto coincidencias literales como similitudes semánticas entre textos.

Capítulo 8. Mantenimiento

8.1 Mantenimiento preventivo de la VPS

El mantenimiento del almacenamiento y de los pagos en la VPS es esencial para asegurar la continuidad operativa del sistema y evitar problemas que podrían afectar su rendimiento o disponibilidad. En cuanto al almacenamiento, si el disco del servidor se llena, el sistema puede experimentar lentitud, errores o incluso una caída total. Esto es particularmente relevante en el proyecto de detección de plagio, donde se manejan grandes volúmenes de documentos y datos que se almacenan y procesan constantemente. Mantener el almacenamiento bajo control permite que la bases de datos funcione sin interrupciones y que las tareas críticas, como la comparación de documentos y el acceso a los archivos, se realicen de manera eficiente.

En primer lugar, es importante llevar un control continuo sobre el uso del disco, ya que un uso elevado puede afectar el rendimiento de la base de datos y el procesamiento de documentos. El sistema debe tener configuradas alertas que notifiquen al administrador cuando el espacio en disco disponible caiga por debajo de un umbral crítico, generalmente alrededor del 80% de capacidad. Herramientas como Netdata o Nagios permiten monitorizar el uso de disco en tiempo real, ayudando a prever cuándo es necesario aumentar el almacenamiento o realizar tareas de limpieza, como eliminar archivos temporales o archivos de log innecesarios.

Además del almacenamiento, otro aspecto crítico es mantener al día los pagos anuales del servicio de VPS en OVH Cloud. Es esencial tener un calendario o recordatorios automáticos para asegurar que el servicio se renueve a tiempo y evitar posibles interrupciones que puedan dejar el sistema fuera de servicio. Configurar pagos automáticos puede ser una opción conveniente, pero es igualmente importante verificar de forma periódica el estado de la cuenta y el servicio para asegurarse de que no haya incidencias en la facturación. Mantener la VPS operativa y sin interrupciones garantiza que el sistema siga siendo accesible y funcional para los usuarios y el equipo de administración.

8.1.1 Mantenimiento preventivo a la Base de datos

El mantenimiento preventivo de la base de datos es una tarea crucial para asegurar que el sistema funcione de manera óptima y que los datos se mantengan seguros e íntegros. En un sistema como el de detección de plagio, donde se almacenan y procesan grandes volúmenes de datos, el

mantenimiento regular de la base de datos es necesario para prevenir problemas de rendimiento, pérdida de datos o corrupción de los mismos. Este mantenimiento preventivo implica varias acciones que se realizan periódicamente para mantener la estabilidad y eficiencia del sistema.

Una de las principales tareas es la optimización de consultas y el uso de índices. A medida que la base de datos crece, las consultas que antes eran rápidas pueden volverse lentas debido al aumento de registros. Implementar índices adecuados en las columnas más consultadas ayuda a mejorar el tiempo de respuesta del sistema. Además, es importante realizar análisis de fragmentación de tablas, ya que la fragmentación excesiva puede ralentizar las consultas.

Otra tarea importante es realizar copias de seguridad periódicas (backups). Esto asegura que, en caso de un fallo del servidor o pérdida de datos, se pueda restaurar la base de datos a un estado funcional reciente. Los backups pueden programarse diaria o semanalmente, dependiendo del volumen de datos que maneje el sistema. Además, es recomendable verificar periódicamente la integridad de los backups para asegurarse de que los archivos sean utilizables en caso de emergencia.

Por último, el monitoreo del uso de espacio en la base de datos es esencial para prevenir problemas de almacenamiento. A medida que la base de datos crece, puede llegar a consumir el espacio asignado, lo que afecta al rendimiento y puede incluso hacer que el sistema se detenga. Implementar alertas automáticas que notifiquen cuando el espacio de almacenamiento está cerca de agotarse es una buena práctica para adelantarse a problemas graves. De este modo, se pueden planificar ampliaciones de almacenamiento o realizar tareas de limpieza de datos innecesarios.

Otra tarea crucial en el mantenimiento preventivo de la base de datos es la verificación de los recursos instalados y la planificación de actualizaciones (upgrades) para asegurar que el sistema continúe funcionando de manera eficiente a medida que crece. A medida que aumenta la cantidad de documentos y usuarios en el sistema de detección de plagio, es posible que los recursos inicialmente asignados a la base de datos, como la capacidad de procesamiento, la memoria o el almacenamiento, ya no sean suficientes para manejar la carga. Es fundamental revisar periódicamente el uso de los recursos en Azure, como el CPU y el IOPS de la base de datos, para identificar cuellos de botella o posibles sobrecargas. Si los recursos actuales se acercan a sus límites, será necesario realizar un upgrade a un plan superior para asegurar que la base de datos continúe respondiendo eficientemente bajo una mayor demanda.

Además, es importante mantener un control riguroso sobre los pagos anuales de la base de datos en Azure. Como la base de datos es una parte esencial del sistema, la interrupción de su servicio por falta de pago podría causar una pérdida temporal o permanente de acceso a los datos. Por lo tanto, es recomendable establecer recordatorios automáticos o configurar pagos automáticos para evitar cualquier interrupción en el servicio. También es útil revisar periódicamente los costos del servicio en Azure y evaluar si los recursos que se están pagando son adecuados o si se pueden optimizar. Esto no solo asegura que el sistema funcione de manera fluida, sino que también ayuda a gestionar los costos de manera eficiente a largo plazo.

8.1.2 Mantenimiento preventivo de los recursos de OpenAI

Una tarea crucial en el mantenimiento preventivo del sistema es la verificación constante del uso de recursos de la API de OpenAI, ya que es una pieza clave para el análisis de similitud de documentos en el proyecto de detección de plagio. Es necesario monitorear el consumo de la API, especialmente el número de solicitudes realizadas y las respuestas recibidas, para asegurarse de que el sistema siga funcionando sin interrupciones. Dado que OpenAI establece límites en la cantidad de consultas y respuestas que pueden ser procesadas dentro de un periodo determinado, es vital revisar el estado de uso mediante el panel de control de OpenAI, que proporciona estadísticas sobre el consumo de API.

Si el proyecto está alcanzando frecuentemente los límites de uso, será necesario ajustar el plan de suscripción para incrementar la cantidad de respuestas permitidas. Esto garantiza que el sistema pueda manejar la demanda creciente sin que se presenten retrasos o errores en la detección de similitudes. Además, esta verificación ayuda a prever el crecimiento futuro del sistema y a planificar con antelación la necesidad de más recursos, evitando interrupciones inesperadas en el servicio. Realizar estas revisiones de manera regular es una práctica esencial para mantener la eficiencia y la continuidad del sistema, y asegurar que siempre se disponga de la capacidad necesaria para cubrir las necesidades del proyecto.

8.2 Mantenimiento correctivo

En el mantenimiento correctivo del sistema, se considera fundamental que en futuros desarrollos de soluciones similares se implementen mejoras clave para optimizar y robustecer la propuesta original.

8.2.1 Implementación del patrón de diseño Circuit Breaker

El mantenimiento correctivo en este proyecto tiene como objetivo optimizar la carga de los documentos de tesis para mejorar la eficiencia del sistema. Actualmente, el proceso de carga de archivos puede ralentizarse dependiendo del tamaño del documento y las condiciones del servidor, lo que afecta la experiencia del usuario y el rendimiento general. Para resolver esto, es fundamental optimizar este proceso y asegurar que, sin importar posibles fallas al cargar un documento, el sistema siga funcionando de manera estable y no afecte a otros módulos o funcionalidades.

Un patrón de diseño que permite que la aplicación continúe operando a pesar de este tipo de fallas es el Patrón de *Circuit Breaker*. Este patrón actúa como un interruptor que "abre" o "corta" las conexiones cuando detecta fallas recurrentes en un servicio, como la carga de documentos, previniendo que el sistema entero se vea afectado. Mientras el proceso de carga se recupera, el resto de la aplicación sigue funcionando sin interrupciones. Esto asegura que, aunque la carga de un archivo específico falle temporalmente, los usuarios puedan seguir interactuando con otras partes del sistema, como el análisis de similitud o la gestión de cuentas, sin experimentar problemas.

Este enfoque no solo mejora la robustez del sistema, sino que también permite una mejor gestión de errores, contribuyendo a un proceso de mantenimiento correctivo más eficiente y minimizando el impacto en el usuario final.

8.2.2 Implementación de validador de códigos identificadores de estudiantes y profesores

Un mantenimiento esencial es la validación de los códigos identificadores con la institución, asegurando que los usuarios, especialmente los estudiantes y profesores, estén correctamente registrados y autenticados. Esto implica validar tanto los códigos de carnet como los dominios de correos institucionales para garantizar que solo usuarios autorizados accedan al sistema. La falta de esta validación podría permitir que personas no pertenecientes a la institución accedan o carguen documentos, comprometiendo la integridad del sistema.

Esta medida no solo refuerza la seguridad, sino que también mejora la fiabilidad de los datos, ya que permite asegurar que cada cuenta esté vinculada a un miembro legítimo de la

institución. Implementar este tipo de validaciones como parte del mantenimiento correctivo fortalece el control de acceso y evita posibles riesgos relacionados con usuarios no autorizados, lo que contribuye a un entorno más seguro y controlado para el uso del sistema.

8.3 Recomendaciones de seguridad

Incluir recomendaciones de seguridad en la tesis es fundamental para garantizar la protección y el correcto funcionamiento del sistema en el entorno real. La seguridad es un aspecto clave en cualquier desarrollo de software, especialmente cuando el sistema maneja información sensible, como documentos académicos, datos de usuarios y registros institucionales. Estas recomendaciones permiten prever y mitigar posibles vulnerabilidades, asegurando que el acceso a la plataforma sea controlado y los datos almacenados estén debidamente protegidos.

Además, las recomendaciones de seguridad son necesarias para asegurar la continuidad y robustez del sistema ante posibles amenazas. Implementar medidas como la autenticación de dos factores (2FA) y la autenticación multifactor (MFA) para superusuarios refuerza las defensas del sistema contra accesos no autorizados y evita potenciales robos de información. Estas acciones son esenciales para mantener la integridad de los datos y garantizar que el sistema pueda operar de manera segura y eficiente a largo plazo.

Otra medida de seguridad esencial es la encriptación de los documentos al momento de ser subidos al sistema. Este proceso asegura que, incluso si un tercero no autorizado logra acceder a los archivos almacenados, no pueda leer ni extraer información sensible. La encriptación protege no solo el contenido del documento, sino también los metadatos asociados, que podrían contener información valiosa como el autor, la fecha de creación, o las ubicaciones de almacenamiento.

Además, es importante que los documentos permanezcan encriptados durante todo su ciclo de vida en el sistema, y que únicamente se descifren temporalmente cuando sea necesario leerlos o procesarlos. Esto evitaría posibles vulnerabilidades, como el robo de información en caso de que los archivos sean interceptados en tránsito o durante su almacenamiento. Implementar este enfoque contribuiría significativamente a mejorar la seguridad y protección de los datos manejados por el sistema.

Capítulo 9 – Conclusiones

9.1 Análisis de similitudes de Tesis

El objetivo principal de este proyecto fue analizar y comparar las tesis presentadas en una misma sede educativa, con el fin de determinar el porcentaje de similitud entre ellas. Tras el análisis de diversas tesis, se ha observado que el rango de similitud entre los documentos varía entre un 20% y un 45%. Este rango es significativamente bajo, lo que indica que, en términos generales, las tesis presentadas por los estudiantes son lo suficientemente originales y no presentan altos niveles de plagio.

De acuerdo con la hipótesis planteada al inicio del proyecto, se esperaba que la similitud entre los documentos no superara el 50%. Los resultados obtenidos confirman que este objetivo fue alcanzado, ya que ninguno de los documentos analizados mostró un porcentaje de similitud superior a este umbral. Este hallazgo es importante, ya que demuestra que la mayoría de los estudiantes siguen buenas prácticas de redacción y citación, minimizando el riesgo de plagio en sus proyectos de tesis.

9.2 Propuesta de un modelo de pago

Uno de los desafíos identificados durante el desarrollo del proyecto fue la necesidad de realizar una inversión significativa en infraestructura y servicios para mantener el sistema en funcionamiento, especialmente debido a los costos asociados con el análisis de grandes volúmenes de documentos. A raíz de este factor, se propone implementar un modelo de pago para financiar el proyecto. Este modelo podría ser diseñado de dos maneras: por institución o por estudiante inscrito al curso.

Modelo por institución: En este caso, la institución educativa podría abonar una tarifa anual para permitir el acceso ilimitado al sistema de análisis de tesis, tanto para estudiantes actuales como para futuras promociones.

Modelo por estudiante inscrito: Alternativamente, se podría implementar un pago individual por estudiante inscrito al curso de tesis, lo que permitiría que solo aquellos que realmente estén participando en el proceso de redacción de tesis paguen por el servicio.

Ambas opciones proporcionarían una fuente de financiamiento que permitiría mantener y expandir el sistema sin sobrecargar a los usuarios.

9.3 Accesibilidad para Estudiantes No Inscritos

Un aspecto interesante que surgió durante el desarrollo del sistema fue la capacidad de ofrecer acceso a los proyectos de tesis a estudiantes que no están inscritos en el curso de tesis o anteproyecto. Aunque los usuarios que no están inscritos no pueden someter sus propios trabajos para análisis, sí tienen la posibilidad de consultar proyectos de tesis de otros estudiantes. Esto podría ser útil para aquellos que aún no han comenzado su propio proyecto de tesis y desean obtener inspiración o revisar los detalles de otros anteproyectos antes de iniciar su propio trabajo. Esta función fomenta la transparencia y la accesibilidad al conocimiento generado por otros estudiantes.

9.4 Implementación de IA y Vectorización

Durante el desarrollo del proyecto, se exploró la posibilidad de utilizar inteligencia artificial para la vectorización de los documentos de tesis, con el fin de mejorar la precisión en la medición de similitud. Sin embargo, tras analizar diferentes enfoques y probar algunas implementaciones, se llegó a la conclusión de que el uso de librerías especializadas en Python, como TfidfVectorizer de scikit-learn, resulta ser mucho más eficiente y efectivo para este propósito.

La vectorización de documentos mediante la librería scikit-learn permite transformar los textos en representaciones numéricas de manera rápida y precisa, sin necesidad de implementar algoritmos complejos de inteligencia artificial. Esta solución no solo es más accesible desde el punto de vista técnico, sino que también ofrece un rendimiento mucho más alto en términos de tiempo de procesamiento y consumo de recursos. Por lo tanto, se decidió adoptar este enfoque, lo que contribuyó a mejorar la eficiencia del sistema y garantizar una mayor escalabilidad.

9.5 Conclusión final

El proyecto ha logrado alcanzar los objetivos establecidos, demostrando que es posible realizar un análisis de similitud de tesis entre documentos de una misma sede educativa sin que los porcentajes de similitud superen el 50%. La implementación de un sistema de pago, así como la

posibilidad de consultar proyectos por parte de estudiantes no inscritos, abre nuevas oportunidades para la sostenibilidad y la accesibilidad del sistema.

Además, la decisión de utilizar librerías estándar en Python para la vectorización y medición de similitud ha permitido que el sistema sea más eficiente y menos dependiente de soluciones complejas de inteligencia artificial. El uso de herramientas ya disponibles en el ecosistema Python ha sido una estrategia acertada para mantener el proyecto simple, eficiente y efectivo.

Por último, se recomienda seguir con el monitoreo del sistema y realizar ajustes en función de la retroalimentación de los usuarios y las instituciones, con el fin de optimizar el proceso y garantizar la sostenibilidad del proyecto a largo plazo.

Capítulo 10 – Recomendaciones

10.1 Expansión del Sistema a Tesis Culminadas

Actualmente, el sistema de análisis de similitud se aplica a anteproyectos de tesis, permitiendo la comparación de trabajos preliminares entre estudiantes. Sin embargo, para mejorar la utilidad y el alcance del sistema, sería beneficioso expandir esta herramienta a tesis culminadas. La posibilidad de comparar tesis terminadas permitiría no solo detectar similitudes entre proyectos preliminares, sino también evaluar el nivel de originalidad de las tesis definitivas antes de su presentación final.

Recomendación: Se recomienda que, una vez el sistema esté consolidado en la fase de anteproyectos, se amplíe su funcionalidad para incluir tesis culminadas. Esto podría implicar la creación de un flujo de trabajo adicional en el que los estudiantes suban sus tesis finales para ser comparadas con otros proyectos previamente depositados en el sistema.

10.1.1 Posibles beneficios

Mejora de la calidad académica: Al comparar tesis terminadas, se asegurará que los trabajos finales cumplan con los estándares de originalidad y evitarán el plagio de trabajos anteriores.

Evaluación más completa: El análisis de similitud de tesis finales brindaría a las instituciones una herramienta para validar la calidad y la autenticidad de los trabajos entregados por los estudiantes.

10.2 Digitalización de Tesis Antiguas

Una de las limitaciones del sistema actual es la falta de acceso a tesis antiguas, las cuales fueron entregadas en formato físico y no se encuentran disponibles en formato digital. Este reto es particularmente relevante cuando se considera la expansión del sistema a tesis culminadas, ya que muchas de estas tesis históricas aún no han sido digitalizadas.

Recomendación: Es fundamental que las instituciones educativas implementen un proceso de digitalización masiva de las tesis entregadas en años anteriores, tanto de anteproyectos como de tesis culminadas. Esta digitalización no solo permitiría que los documentos antiguos puedan ser

procesados en el sistema de similitud, sino que también contribuiría a la preservación digital de los trabajos académicos de los estudiantes.

10.2.1 Posibles acciones

Digitalización escalonada: Priorizar la digitalización de tesis más relevantes o las más consultadas, y progresivamente incorporar el resto de los trabajos.

Uso de servicios externos: Contratar empresas o servicios especializados en digitalización para asegurar que el proceso sea eficiente y cumpla con los estándares de calidad.

Uso de OCR: Implementar tecnología de **Reconocimiento Óptico de Caracteres (OCR)** para convertir documentos escaneados en texto editable que pueda ser procesado por el sistema de análisis de similitudes.

10.3 Implementación de RPA (Automatización Robótica de Procesos) para la Carga Masiva de Documentos

Una de las barreras más importantes al trabajar con grandes volúmenes de tesis es la necesidad de cargar los documentos al sistema de forma manual. Este proceso puede resultar muy tedioso, especialmente si el sistema debe gestionar tanto anteproyectos como tesis culminadas. Para facilitar y agilizar este proceso, se recomienda implementar RPA (Automatización Robótica de Procesos).

Recomendación: Implementar bots de RPA para automatizar la carga masiva de documentos al sistema. Estos bots pueden encargarse de leer y procesar documentos en diversos formatos (por ejemplo, PDFs o Word) y almacenarlos en la base de datos sin intervención humana. La automatización no solo ahorrará tiempo, sino que también evitará errores manuales en la carga de los documentos.

10.4 Fomentar la Participación de los Estudiantes en la Subida de Documentos

Una de las claves para garantizar el éxito del sistema es incentivar la participación activa de los estudiantes en la carga de sus anteproyectos y tesis al sistema. Actualmente, solo aquellos que están inscritos en el curso de anteproyecto de tesis tienen acceso a la plataforma, pero ampliar

la base de datos con más trabajos mejorará la calidad de las comparaciones y el análisis de similitud.

Recomendación: Promover la carga de documentos por parte de los estudiantes desde las primeras etapas de la redacción de sus tesis. Esto no solo facilitará el acceso a los proyectos, sino que también permitirá una mayor cobertura de la base de datos para la comparación de documentos.

Referencias

Castro Rodriguez, Y. (2020). El plagio académico desde la perspectiva de la ética de la publicación. *Revista Cubana de Información en Ciencias de la Salud*, 31(4). Universidad Católica del Perú. <https://www.medigraphic.com/pdfs/acimed/aci-2020/aci204o.pdf>

Congreso de la República Guatemalteca. (2000, 03 de noviembre). Ley de Derecho de Autory y Derechoss Conexos de Guatemala. Obtenido de https://mcd.gob.gt/wp-content/uploads/2013/07/ley_derechos_de_autor_conexos_01.pdf

Loayza Salvatierra, N. M. (2019). Similitud en tesis de pregrado de medicina publicadas en repositorios de Universidades de Trujillo. Universidad Nacional de Trujillo. Recuperado de <https://hdl.handle.net/20.500.14414/13441>

Luis, E. J. (2022). Causas del plagio académico en estudiantes universitarios de educación: percepción docente de una Universidad Dominicana. *Revista Educare*.

Piñero Pérez, P., Pérez Pupo, I., Rivero Hechavarría, C., Rojas Lusardo, C., González Sosa, R., & Torres López, S. (2019). Repositorio de datos para investigaciones en gestión de proyectos. Repositorio de datos para investigaciones en gestión de proyectos.

Turnitin. (s.f.). 4 opciones para configurar un ejercicio de Turnitin. Recuperado de <https://latam.turnitin.com/blog/4-opciones-para-configurar-un-ejercicio-de-turnitin>

SafeAssign. (s. f.). SafeAssign. Recuperado de <https://help.blackboard.com/es-es/Learn/Instructor/Ultra/Assignments/SafeAssign>

Sierra Martínez, G. (2022). UNAM desarrolla herramienta contra el plagio digital. Fundación UNAM. Recuperado de <https://shre.ink/DCyr>

REDIB. (s. f.). ¿Quiénes somos? Proyecto REDIB. Recuperado de <https://www.redib.org/quienes-somos-proyecto>

SIBDI crea repositorio digital para tesis. (s. f.). Universidad de Costa Rica. Recuperado de <https://www.ucr.ac.cr/noticias/2014/06/19/sibdi-crea-repositorio-digital-para-tesis.html>

González Lemus, E. J. (2019). *Sistema para la automatización del proceso de trabajo de graduación para los estudiantes de la escuela de Ingeniería Mecánica Industrial, Facultad de Ingeniería, Universidad de San Carlos de Guatemala* [Tesis de licenciatura, Universidad de San Carlos de Guatemala]. Repositorio Institucional USAC. <https://shre.ink/DCyI>

Plag. (2024). *Plag - Detector de plagio e IA*. Recuperado el 21 de junio de 2024, de <https://www.plag.es>

Universidad de El Salvador. (s.f.). *Repositorio Institucional de la Universidad de El Salvador*. Recuperado de <https://repositorio.ues.edu.sv/home>

Turnitin. (s. f.). *Efectividad global de Turnitin*. Recuperado el 24 de julio de 2024, de <https://www.turnitin.com/static/global-effectiveness/>

Repositorio Académico, Universidad de Chile. (s.f.). *Derechos de autor*. Universidad de Chile. Recuperado el 13 de noviembre de 2024, de https://repositorio.uchile.cl/page/derechos_autor

Universidad de Chile. (s.f.). *Cómo delimitar un tema de investigación*. Aprendizaje UChile. Recuperado el 13 de noviembre de 2024, de <https://aprendizaje.uchile.cl/recursos-para-leer-escribir-y-hablar-en-la-universidad/profundiza/profundiza-la-escritura/como-delimitar-un-tema-de-investigacion/>

Apéndice

La investigación sobre detección de similitudes y administración de anteproyectos en la facultad de Ingeniería en Sistemas, incluye entrevistas y cuestionarios realizados a la población de estudio.

Apéndice 1. Cuestionario sobre la experiencia del docente

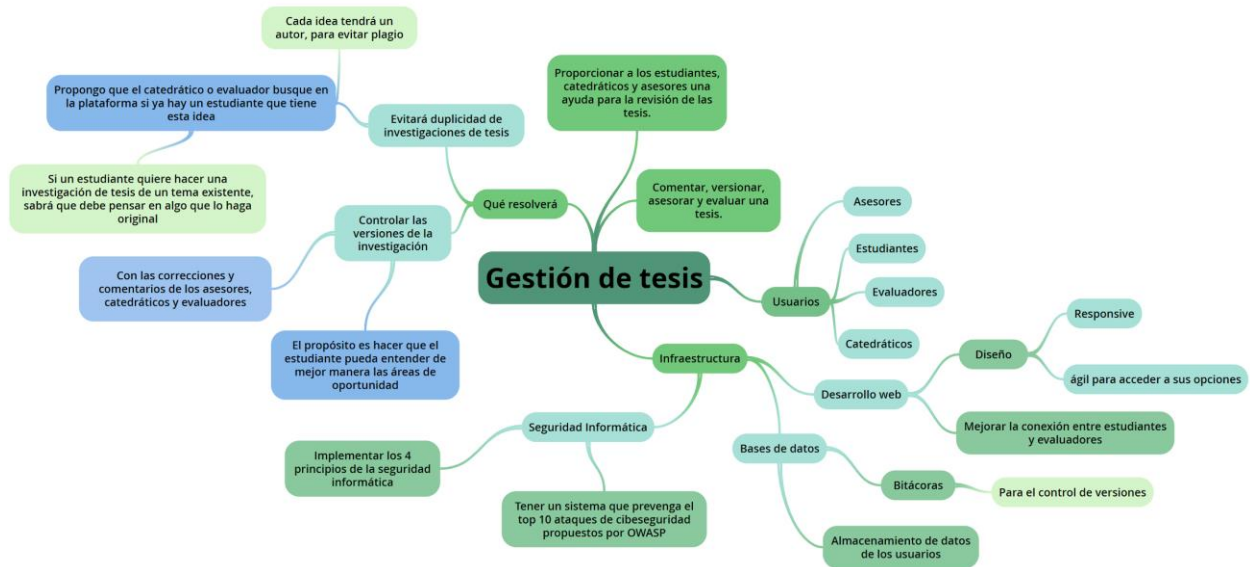
Este cuestionario fue realizado a una docentes de la facultad de Ingeniería en Sistemas en la sede de boca del monte, la cual imparte el curso de Proyecto de Graduación.

1. ¿Conoce algunos casos en los que un estudiante llega con una tesis que ya fue presentada con anterioridad? ¿Si es así, puede recordar aproximadamente cuantas veces?
2. Si la respuesta anterior fue afirmativa, ¿existieron problemas legales? Si llega a pasar, ¿Qué clase de problemas legales pueden ocurrir?
3. En la sede de Boca del Monte, además de usted, ¿hay alguien más que tenga acceso a las investigaciones de tesis de PG1 y PG2. (Catedráticos, secretaría, etc)?
4. En la sede de Boca del Monte, ¿Aproximadamente cuántos proyectos de tesis se culminan?

Apéndice 2. Matriz operacional de variables

Objetivo General	Variable dependiente	Variables independientes	Objetivo	Definición conceptual	Definición operacional	Dimensiones	Indicadores
Mejorar el proceso de revisión de tesis aprobadas en la fase de anteproyecto para los docentes y estudiantes de Proyecto de Graduación en una universidad guatemalteca.	Eficiencia en el proceso de revisión de tesis.	Temas duplicados	Facilitar la detección de proyectos duplicados.	El Comité Internacional de Editores de Revistas Médicas (ICMJE) utiliza el título "Publicación Superpuesta" para este término. El ICMJE define la publicación duplicada como "la publicación de un artículo que se superpone sustancialmente con uno ya publicado, sin referencia clara y visible a la publicación previa".	La medición de los temas duplicados nos permite saber la magnitud de la necesidad de controlar los temas en un repositorio en línea.	Porcentaje de temas duplicados al inicio del estudio en diferentes sedes.	Identificación de patrones de duplicación
							Índice de similitud entre temas duplicados y publicaciones previas
		Revisión de tesis	Automatizar el proceso de revisión de tesis.	La revisión de tesis es el proceso sistemático y crítico mediante el cual se examina y evalúa un documento académico extenso, el cual ha sido elaborado por un estudiante como requisito para la obtención de un título universitario.	La medición de la revisión de un proyecto de tesis se realizará de manera <u>más óptima</u> , para reducir el tiempo de lectura y análisis.	Tiempo invertido en la revisión de proyectos de tesis.	Tiempo de respuesta en cuanto al proceso de esperar a recibir una tesis para revisión.
							Tiempo promedio de revisión.
		Búsqueda de temas presentados	Implementar una base de datos para los estudiantes en donde visualicen los títulos de los proyectos conforme a su búsqueda para disminuir la duplicidad o el plagio.	Se refiere al proceso de investigar y analizar la existencia previa de trabajos o investigaciones relacionadas con un tema específico.	La medición de la satisfacción de los estudiantes y docentes en la búsqueda de temas de anteproyecto.	Porcentaje de satisfacción	Tasa de error en la revisión.
							Eficiencia de la búsqueda siendo esta rápida o muy lenta.
							Relevancia de las fuentes frente a la búsqueda de usuario

Apéndice 3. Mapa conceptual de proyecto de tesis



Apéndice 4. Visualización de resúmenes

Cuenta

Menú principal

Buscar documentos

Configuración de cuenta

Salir

Sistema Web Detector de Plagio o similitudes

Porcentaje de similitud: 39%

Abrir.pdf

Pregunta general

¿Cómo se puede mejorar el método de consulta de los anteproyectos de tesis para la facultad de Ingeniería en Sistemas de Información en una universidad guatemalteca para administrar los proyectos aprobados disminuyendo la probabilidad de plagio o duplicidad?

Objetivo general

Apoyar el proceso de administración de anteproyectos de tesis mediante la automatización de la revisión y calificación, para que los docentes puedan detectar indicios de duplicidad o plagio. Además, brindar acceso a los estudiantes para conocer los temas vigentes.

Hipótesis

La hipótesis sugiere resultados encontrados en la web con respecto a la herramienta a nivel internacional Turnitin, la cual ha observado que, con el tiempo, las instituciones pueden ver una disminución de entre el 17.4% y el 66.8% en trabajos con más del 50% de contenido no original (Turnitin, s.f.).

Apéndice 5. Visualización de la pantalla de búsqueda de documentos

Menú principal
Buscar documentos
Configuración de cuenta
Salir

Cuenta

Escribe lo que deseas buscar

Q

Nombre de tesis

Sistema Web detector de plagio

Autor

Lourdes Adriana Pérez Barillas

Sede

Boca del Monte

Nombre de tesis

Sistema Web detector de plagio

Autor

Lourdes Adriana Pérez Barillas

Sede

Boca del Monte

Nombre de tesis

Sistema Web detector de plagio

Autor

Lourdes Adriana Pérez Barillas

Apéndice 6. Reportes de los estudiantes de un curso

Menú principal
Buscar documentos
Configuración de cuenta
Salir

Cuenta

Tus estudiantes presentan estos porcentajes a nivel de curso de similitud.

Estudiante	Porcentaje
Lourdes	39%
Julio	35%
Pédro	20%
Juan	40%
Ania	56%

Tus estudiantes presentan estos porcentajes de similitud a nivel institución.

Estudiante	Porcentaje
Lourdes	39%
Julio	35%
Pédro	20%
Juan	40%
Ania	56%

Apéndice 7. Subir documento

Menú principal
Buscar documentos
Configuración de cuenta
Salir

Cuenta

Subir documento

Lourdes Adriana

Pérez Barillas

Examinar

Apéndice 8. Opciones disponibles para un profesor

Menú principal
Buscar documentos
Configuración de cuenta
Salir

Cuenta

Bienvenido, Saúl

Ver curso

Ver Sede

Reporte de análisis

Ver estudiantes

Apéndice 9. Opciones disponibles para un estudiante

Menú principal
Buscar documentos
Configuración de cuenta
Salir

Cuenta

Bienvenida, Lourdes

Ver curso

Subir documento

Reporte de análisis

Ver documento de tesis


Apéndice 10. Detalles de tesis

Menú principal
Buscar documentos
Configuración de cuenta
Salir

Cuenta

Casi listo...

Antes de guardar tu tesis, por favor tomate un tiempo para indicar los siguientes detalles sobre tu tesis. Esto ayuda a identificar la similitud con otros proyectos de tesis.



Pregunta general

Escribe la pregunta que tu proyecto desea responder a lo largo del desarrollo.

Objetivo general

Escribe el objetivo de tu tesis. Debes indicar cual es la meta del proyecto que deseas implementar.

Hipótesis

Escribe la posible respuesta a tu pregunta de tesis para indicar la hipótesis.

Justificación

Escribe la justificación de tu tesis. Justifica por qué es importante que tu proyecto se lleve a cabo.

Guardar