



UNIVERSIDAD PRIVADA DE TACNA

FACULTAD DE INGENIERIA

Escuela Profesional de Ingeniería de Sistemas

Propuesta del Proyecto
Desarrollo de un Sistema de Conversión y
Organización de Documentos Técnicos en Markdown
con Estructuración Automática y Control de
Versiones para los estudiantes en la facultad de
Ingeniería de Sistemas

Curso: Patrones de Software

Docente: Ing. Patrick Jose Cuadros Quiroga

Integrantes:

<i>Chambi Cori, Jerson Roni</i>	<i>(2021072619)</i>
<i>Flores Quispe, Jaime Elias</i>	<i>(2021070309)</i>
<i>Leyva Sardon, Elvis Ronald</i>	<i>(2021072614)</i>
<i>Chite Quispe, Brian Danilo</i>	<i>(2021070015)</i>

Tacna – Perú
2025

Proyecto

Desarrollo de un Sistema de Conversión y Organización de Documentos Técnicos en Markdown con Estructuración Automática y Control de Versiones para los estudiantes en la facultad de Ingeniería de Sistemas , UPT, 2025

Presentado por:

***Chambi Cori, Jerson Roni
Flores Quispe, Jaime Elias
Leyva Sardon, Elvis Ronald
Chite Quispe, Brian Danilo***

***(2021072619)
(2021070309)
(2021072614)
(2021070015)***

UPT

3/07/2025

CONTROL DE VERSIONES					
Versión	Hecha por	Revisada por	Aprobada por	Fecha	Motivo
1.0	MPV	ELV	ARV	10/10/2020	Versión Original

Tabla de contenido

Resumen Ejecutivo	3
I Propuesta narrativa	4
1. Planteamiento del Problema.....	4
2. Justificación del proyecto	4
3. Objetivo general	5
4. Beneficios	6
5. Alcance	7
6. Requerimientos del sistema	7
7. Restricciones	7
8. Supuestos.....	7
9. Resultados esperados	8
10. Metodología de implementación	8
11. Actores claves	9
12. Papel y responsabilidades del personal	10
13. Plan de monitoreo y evaluación	12
14. Cronograma del proyecto	15
15. Hitos de entregables	17
II Presupuesto	18

1. Planteamiento de aplicación del presupuesto	18
2. Presupuesto	18
3. Análisis de Factibilidad	18
4. Evaluación Financiera.....	18
 Anexo 01 – Requerimientos del Sistema <i>{nombre del sistema}</i>	 19

RESUMEN EJECUTIVO

Nombre del Proyecto propuesto:

Desarrollo de un Sistema de Conversión y Organización de Documentos Técnicos en Markdown con Estructuración Automática y Control de Versiones para los estudiantes en la facultad de Ingeniería de Sistemas, *Upt* , 2025

Propósito del Proyecto y Resultados esperados:

El propósito del proyecto es desarrollar un sistema que permita a los estudiantes de la Facultad de Ingeniería de Sistemas convertir, organizar y estructurar automáticamente documentos técnicos en formato Markdown, integrando funcionalidades de control de versiones para facilitar la gestión colaborativa y el seguimiento de cambios.

Los resultados esperados son:

- Automatizar la conversión de documentos técnicos a formato Markdown.
- Estructurar de manera ordenada los contenidos mediante plantillas predefinidas.
- Implementar un sistema de control de versiones para gestionar ediciones y colaboraciones.
- Facilitar la organización y consulta de documentos por parte de los estudiantes y docentes.
- Mejorar la eficiencia en la elaboración y revisión de informes académicos y técnicos.

Población Objetivo:

Los beneficiarios del proyecto son los estudiantes y docentes de la Facultad de Ingeniería de Sistemas, quienes podrán gestionar de manera más eficiente sus documentos técnicos, prácticas, informes académicos y proyectos de investigación, optimizando el tiempo y mejorando la organización del contenido.

Monto de Inversión (En Soles):

S/. 11,660

Duración del Proyecto (En Meses):

6 meses

I Propuesta narrativa

1. Planteamiento del Problema

En la Facultad de Ingeniería de Sistemas, la documentación técnica juega un papel crucial en la enseñanza y el desarrollo de proyectos académicos. Actualmente, los estudiantes y docentes utilizan múltiples formatos de documentos, como Word, PDF, HTML y TXT, lo que dificulta la estandarización y estructuración de la información. La falta de un formato unificado y herramientas eficientes para la conversión y organización de documentos genera problemas de accesibilidad, pérdida de versiones previas y dificultades en la gestión colaborativa.

El principal problema radica en la falta de un sistema eficiente que permita convertir, organizar y gestionar documentos técnicos de manera automatizada y estructurada. Sin una herramienta que facilite la conversión a Markdown y la gestión de versiones, los estudiantes enfrentan dificultades para mantener una documentación clara y accesible en el tiempo.

Asimismo, la interoperabilidad con plataformas de almacenamiento en la nube y repositorios como GitHub es limitada, lo que dificulta la colaboración y el versionado de documentos en proyectos académicos. A esto se suman los riesgos de incompatibilidad con ciertos dispositivos y navegadores, la necesidad de seguridad en el manejo de la información y la posible resistencia al cambio por parte de los usuarios.

Este proyecto busca resolver estas problemáticas mediante el desarrollo de un Sistema de Conversión y Organización de Documentos Técnicos en Markdown con Estructuración Automática y Control de Versiones, proporcionando una solución integral para mejorar el flujo de trabajo académico y la accesibilidad a la documentación técnica dentro de la facultad.

2. Justificación del proyecto

En el contexto académico de la Facultad de Ingeniería de Sistemas, los estudiantes y docentes generan una gran cantidad de documentos técnicos que requieren una adecuada organización, formato estandarizado y facilidad de edición colaborativa. Sin embargo, actualmente se enfrentan a dificultades como la falta de herramientas para estructurar automáticamente los contenidos, el uso de formatos inconsistentes y la limitada capacidad para llevar un control de versiones.

El uso de Markdown como formato base permite una redacción técnica clara, ligera y fácilmente compatible con múltiples plataformas. Al implementar un sistema que

automatice la conversión y estructuración de documentos en Markdown, y que además integre control de versiones, se mejora significativamente la productividad, la calidad de los trabajos académicos y la trazabilidad de los cambios realizados en ellos.

Este proyecto no solo optimiza el flujo de trabajo de los estudiantes, sino que también promueve buenas prácticas en la documentación técnica, colaboración y gestión de versiones, habilidades clave en el ámbito profesional de la ingeniería de software.

3. Objetivo general

Diseñar e implementar un sistema web que facilite la conversión automática de documentos a formato Markdown, mejorando la organización, estructuración y accesibilidad de la documentación técnica en entornos académicos.

4. Beneficios

El desarrollo del sistema de conversión y organización de documentos técnicos en Markdown con estructuración automática y control de versiones generará los siguientes beneficios:

- Estandarización de documentos: Facilita la creación de documentos técnicos con una estructura clara y uniforme.
- Ahorro de tiempo: Automatiza tareas repetitivas como el formato y organización del contenido.
- Mejora en la colaboración: El control de versiones permite que varios estudiantes trabajen en conjunto sin pérdida de información.
- Trazabilidad de cambios: Permite visualizar y recuperar versiones anteriores de un documento con facilidad.
- Fomento de buenas prácticas: Introduce a los estudiantes al uso de herramientas profesionales de documentación y control de versiones.
- Accesibilidad y portabilidad: Markdown es un formato ligero y compatible con diversas plataformas y herramientas.
- Apoyo al proceso educativo: Docentes podrán revisar de forma más clara y organizada los informes y proyectos de los estudiantes.

5. Alcance

El alcance del proyecto incluye el desarrollo de un sistema web que permita:

- Convertir automáticamente documentos en formatos Word (.docx), PDF, HTML y TXT a Markdown (.md), asegurando la preservación de títulos, listas y otros formatos.
- Generar archivos de navegación automática (_Sidebar.md, _Footer.md) basados en la estructura del documento original.
- Ofrecer una visualización previa del documento convertido para garantizar la fidelidad del formato.
- Incorporar una funcionalidad de mejora de documentos con IA.
- Permitir la autenticación de usuarios para gestionar versiones previas de los documentos.
- Facilitar la descarga de los archivos convertidos en Markdown listos para su uso en plataformas de documentación.

6. Requerimientos del sistema

El sistema permitirá a los usuarios registrarse (RF01) y autenticarse mediante inicio de sesión (RF02) para acceder a sus funcionalidades. Podrán subir documentos en formatos como PDF o DOCX (RF03), los cuales serán convertidos automáticamente a Markdown (RF04) y descargables posteriormente (RF05). Además, podrán visualizar todos sus documentos (RF06), mantener un historial de versiones por cada uno (RF07) y asegurar que solo se acepten archivos con formato válido (RF08). El sistema incluirá la opción de previsualizar el archivo Markdown generado (RF09) y aplicar mejoras automáticas en estilo, ortografía y consistencia mediante inteligencia artificial (RF10). También se podrá analizar la similitud entre dos documentos mediante la similitud del coseno (RF11). Los archivos Markdown podrán ser publicados directamente como README.md en el repositorio GitHub del usuario (RF12) o en la sección Wiki del mismo, utilizando autenticación por token (RF13). Finalmente, el sistema permitirá publicar múltiples archivos en la Wiki y generar automáticamente una página de inicio con un índice de enlaces (RF14).

7. Restricciones

Para el desarrollo e implementación del sistema, se deben considerar las siguientes restricciones:

- Compatibilidad con formatos de entrada: Solo se admitirán documentos en formatos Word (.docx), PDF, HTML y TXT para la conversión a Markdown.
- Capacidad de almacenamiento y procesamiento: La plataforma tendrá limitaciones en el tamaño máximo de los archivos para garantizar un rendimiento óptimo.
- Acceso controlado: Solo usuarios autenticados podrán utilizar funciones avanzadas como la gestión de versiones y almacenamiento de documentos en la nube.
- Dependencia de herramientas externas: Algunas funcionalidades, como la implementación de inteligencia artificial, depende de API de terceros.

8. Supuestos

Para el correcto funcionamiento del sistema, se consideran las siguientes suposiciones y dependencias:

- Disponibilidad de conexión a internet: La plataforma requiere acceso a la web para el procesamiento y almacenamiento de documentos.
- Uso de estándares abiertos: Se asume que los formatos de entrada son compatibles con herramientas de conversión a Markdown.
- Acceso a repositorios GitHub: La integración con GitHub dependerá de los permisos de acceso establecidos por la plataforma.
- Infraestructura de servidores: El sistema debe alojarse en una plataforma confiable que garantice disponibilidad y seguridad de los datos.

9. Resultados esperados

Se espera como resultado la implementación de un sistema funcional que permita a los estudiantes y docentes de la Facultad de Ingeniería de Sistemas convertir documentos técnicos a formato Markdown de manera automática, garantizando una estructura clara, ordenada y estandarizada. Asimismo, el sistema permitirá gestionar versiones de documentos, mejorar automáticamente la redacción mediante inteligencia artificial, y facilitar la publicación de archivos en plataformas como GitHub. También se busca mejorar la colaboración académica, optimizar el tiempo dedicado a la elaboración de

informes técnicos y fomentar el uso de herramientas modernas de documentación y control de cambios. Finalmente, se espera que el sistema contribuya a elevar la calidad y profesionalismo de los trabajos entregados en el entorno académico.

10. Metodología de implementación

Para el desarrollo del sistema se empleará la metodología ágil SCRUM, que permite una gestión iterativa e incremental del proyecto, promoviendo la colaboración constante con los usuarios y la entrega continua de funcionalidades. El proceso se dividirá en sprints de duración fija, cada uno con objetivos específicos y entregables funcionales.

Inicialmente, se realizará una fase de levantamiento de requerimientos para definir claramente las funcionalidades del sistema y priorizarlas. Posteriormente, se desarrollará una prototipación rápida de la interfaz y flujo de uso, seguida de la implementación del backend y los módulos clave como la conversión a Markdown, control de versiones y publicación en GitHub.

Se utilizarán tecnologías como Node.js o Python para el backend, Markdown parsers y herramientas de conversión de documentos, así como APIs de GitHub para la integración con repositorios. En paralelo, se aplicarán pruebas unitarias y de integración en cada sprint para asegurar la calidad del producto. Finalmente, se realizará una fase de validación con estudiantes y docentes, recogiendo su retroalimentación para ajustes finales antes del despliegue definitivo del sistema.

11. Actores claves

Los actores claves del proyecto son:

- Estudiantes de la Facultad de Ingeniería de Sistemas: Son los principales usuarios del sistema. Utilizarán la herramienta para convertir, estructurar y gestionar sus documentos técnicos en formato Markdown, así como para colaborar y publicar contenidos académicos.
- Docentes y asesores académicos: Usarán el sistema para revisar informes, proyectos y prácticas con una estructura estandarizada, lo que facilitará la evaluación y retroalimentación del contenido presentado por los estudiantes.

- **Equipo de desarrollo:** Responsables del análisis, diseño, implementación, pruebas y mantenimiento del sistema. Incluye programadores, diseñadores de interfaz y especialistas en integración con herramientas como GitHub.
- **Área de soporte tecnológico de la facultad:** Encargada de garantizar la infraestructura, disponibilidad y soporte técnico del sistema una vez desplegado.
- **Coordinación académica:** Supervisará el cumplimiento de los objetivos del proyecto y su alineación con las necesidades educativas de la facultad.

12. Papel y responsabilidades del personal

Para asegurar una implementación efectiva del proyecto, se asignarán los siguientes roles y responsabilidades al personal involucrado:

- **Jefe de Proyecto:** Encargado de la planificación, coordinación general y supervisión del desarrollo del sistema. Asegura que se cumplan los plazos, el alcance y los objetivos definidos.
- **Analista de Requerimientos:** Responsable de recopilar, documentar y validar las necesidades de los usuarios, transformándolas en requerimientos funcionales y no funcionales del sistema.
- **Desarrollador Backend:** Encargado de programar la lógica del sistema, implementar los módulos de conversión de documentos, control de versiones y conexión con APIs externas como GitHub.
- **Desarrollador Frontend:** Diseña e implementa la interfaz gráfica del sistema, asegurando una experiencia de usuario intuitiva y accesible para los estudiantes y docentes.
- **Especialista en Inteligencia Artificial:** Desarrolla el componente encargado de mejorar los archivos generados (estilo, ortografía, estructura), aplicando técnicas de procesamiento de lenguaje natural.
- **Tester o QA (Aseguramiento de Calidad):** Se encarga de diseñar y ejecutar pruebas para verificar que el sistema cumple con los requisitos y no presenta errores funcionales o de seguridad.
- **Docentes Asesores:** Proveen orientación académica, validan el enfoque pedagógico del sistema y participan en las pruebas piloto con estudiantes.

- **Soporte Técnico:** Se encarga del mantenimiento del sistema, asistencia a los usuarios y resolución de incidencias una vez implementado.

13. Plan de monitoreo y evaluación

El plan de monitoreo y evaluación tiene como objetivo asegurar el cumplimiento de los objetivos del proyecto, identificar posibles desviaciones y garantizar la calidad del sistema desarrollado. Para ello, se establecerán los siguientes mecanismos:

- **Indicadores de avance:** Se definirá un conjunto de indicadores cuantificables, como número de funcionalidades completadas por sprint, cumplimiento del cronograma, porcentaje de cobertura de pruebas y nivel de satisfacción de los usuarios.
- **Reuniones de seguimiento:** Se realizarán reuniones semanales del equipo de desarrollo (Scrum daily/weekly) y reuniones quincenales con docentes asesores para revisar el progreso, detectar obstáculos y tomar decisiones correctivas.
- **Evaluación de calidad funcional:** A través de pruebas unitarias, de integración y de usuario, se verificará que el sistema cumple con los requerimientos funcionales y no funcionales establecidos.
- **Retroalimentación de usuarios:** Se llevará a cabo una fase piloto con estudiantes y docentes seleccionados, quienes evaluarán la utilidad, usabilidad y desempeño del sistema mediante encuestas y entrevistas.
- **Documentación de resultados:** Todos los avances, problemas identificados, soluciones aplicadas y recomendaciones se documentarán en informes técnicos para asegurar la trazabilidad del proceso y facilitar futuras mejoras.
- **Informe final de evaluación:** Al término del proyecto, se elaborará un informe integral que incluya el análisis de resultados obtenidos, el cumplimiento de objetivos y propuestas para su mantenimiento o ampliación.

14. Cronograma del proyecto

Semana	Fechas	Fase / Actividades
Semana 1	20 – 26 marzo	Planificación y Análisis - Requisitos funcionales y no funcionales- Análisis del mercado y competencia- Definición de objetivos y alcance
Semana 2	27 marzo – 2 abril	Diseño - Arquitectura técnica (Flask + HTML/CSS)- Wireframes y diseño de interfaz de usuario- UX/UI

Semanas 3-5	3 – 23 abril	Desarrollo del Backend - Configuración del entorno- Lógica del servidor (Flask)- Rutas, modelos, controladores- Pruebas unitarias
Semanas 6-7	24 abril – 7 mayo	Desarrollo del Frontend - Maquetación con HTML/CSS- Integración con el backend- Validación de formularios
Semanas 8-9	8 – 21 mayo	Integración de Componentes - Conexión frontend-backend- Validación de flujos- Pruebas funcionales
Semana 10	22 – 28 mayo	Pruebas Internas y Ajustes - Corrección de bugs- Refactorización- Validación técnica
Semanas 11-12	29 mayo – 11 junio	Pruebas de Usuario (UAT) - Feedback de usuarios- Registro de mejoras- Validación funcional
Semanas 13-14	12 – 25 junio	Mejoras y Documentación - Implementación de mejoras- Documentación técnica y de usuario- Preparación para producción
Semanas 15-16	26 junio – 5 julio	Despliegue y Cierre - Pruebas finales- Despliegue de la aplicación- Presentación del proyecto- Evaluación final

Hitos de entregables

A lo largo del desarrollo del proyecto, se establecerán los siguientes hitos clave, cada uno asociado a un entregable que marca el avance significativo del sistema:

1. Hito 1 – Documento de Requerimientos Funcionales y No Funcionales

Entregable: Documento detallado que recoge las necesidades del sistema, los casos de uso y las especificaciones técnicas.

Fecha estimada: [semana 2]

2. Hito 2 – Diseño de Arquitectura del Sistema y Prototipo de Interfaz

Entregable: Diagramas de arquitectura (lógica, física, base de datos) y prototipo navegable de la interfaz de usuario.

Fecha estimada: [semana 4]

3. Hito 3 – Desarrollo de Módulo de Registro, Login y Subida de Documentos

Entregable: Módulos funcionales para autenticación y carga de archivos, con validación de formato.

Fecha estimada: [semana 6]

4. Hito 4 – Conversión Automática a Markdown y Previsualización

Entregable: Módulo funcional para transformar documentos a Markdown y mostrar una

vista previa.

Fecha estimada: [semana 8]

5. Hito 5 – Mejora Automática del Documento y Control de Versiones

Entregable: Integración del componente de inteligencia artificial para mejora del archivo y manejo de versiones.

Fecha estimada: [semana 10]

6. Hito 6 – Publicación en GitHub (README y Wiki)

Entregable: Funcionalidad para publicar archivos directamente en repositorios y wikis de GitHub usando autenticación segura.

Fecha estimada: [semana 12]

7. Hito 7 – Pruebas Piloto y Evaluación con Usuarios Finales

Entregable: Informe de resultados de pruebas con estudiantes y docentes, retroalimentación y ajustes finales.

Fecha estimada: [semana 13]

8. Hito 8 – Entrega Final del Sistema y Documentación Técnica

Entregable: Sistema completamente funcional, documentación técnica y manual de usuario.

Fecha estimada: [semana 14]

II Presupuesto

1. Planteamiento de aplicación del presupuesto

El presupuesto asignado al proyecto será aplicado estratégicamente en distintas áreas clave para asegurar el desarrollo e implementación exitosa del sistema. La distribución propuesta es la siguiente:

- Desarrollo del sistema (60%): Se destinará la mayor parte del presupuesto a la contratación de profesionales como programadores, analistas y especialistas en inteligencia artificial, responsables de la implementación de las funcionalidades principales como la conversión a Markdown, control de versiones y publicación en GitHub.
- Infraestructura tecnológica (15%): Este porcentaje cubrirá los costos de servidores en la nube, servicios de almacenamiento, licencias de software, herramientas de desarrollo y APIs necesarias para la integración con plataformas externas como GitHub o motores de conversión.

- Capacitación y documentación (10%): Incluye la elaboración de manuales de usuario, guías técnicas y la realización de talleres para capacitar a estudiantes y docentes en el uso del sistema.
- Pruebas y aseguramiento de la calidad (5%): Se invertirá en pruebas funcionales, de usabilidad y rendimiento, asegurando que el sistema cumpla con los estándares de calidad antes de su entrega final.
- Soporte técnico inicial (5%): Incluye la atención de incidencias técnicas durante los primeros meses tras la implementación, así como ajustes o mejoras menores de acuerdo a la retroalimentación recibida.
- Contingencias e imprevistos (5%): Se reservará una parte del presupuesto para cubrir cualquier gasto no previsto, ajustes técnicos o ampliaciones mínimas del alcance durante la ejecución del proyecto.

2. Presupuesto

3. Hablar de presupuestos manejables en el contexto del desarrollo del Sistema de Conversión y Organización de Documentos Técnicos en Markdown implica establecer una planificación financiera clara y eficiente que permita optimizar los recursos disponibles sin comprometer la calidad técnica ni el cumplimiento de los objetivos académicos del proyecto.

4.

5. Dado que este sistema está orientado a mejorar los procesos de documentación en entornos universitarios, se busca mantener un equilibrio entre funcionalidad, sostenibilidad y bajo costo operativo. Para ello, es necesario definir y proyectar los siguientes costos principales:

6.

7. Costos Generales

8.

CONCEPTO	CANTIDAD	COSTO UNITARIO	COSTO TOTAL
Material de oficina (cuadernos, lápices, papel, etc.)	-	-	S/. 100
Transporte (reuniones técnicas)	-	-	S/. 200
Total			S/. 300

9.

10. Costos operativos durante el desarrollo

11.

<i>Concepto</i>	<i>Cantidad</i>	<i>Costo mensual (S/)</i>	<i>Total (6 meses) (S/)</i>
<i>Servicios básicos (agua, luz, internet)</i>	<i>1</i>	<i>300</i>	<i>1800</i>
Total			1800

12.

13. Costos del ambiente

14.

Concepto	Costo estimado (S/)
Dominio web (.com o .org)	50
Hosting para plataforma	230
Total	280

15.

16.

17.

18.

19. Costos de infraestructura

20.

21.

Informe de Costos de Infraestructura - Proyecto

Este informe detalla los costos mensuales estimados de los recursos desplegados en la infraestructura del proyecto.

Total Estimado

Total mensual estimado: \$12.985 USD/mes

47 soles al mes.

Este reporte fue generado automáticamente el día: 2025-05-31

Recursos con Costo

Recurso	Componente	Cantidad	Unidad	Costo Mensual
azurerm_service_p...	Instance usage (B1)	730	hours	\$12.41
azurerm_mssql_dat...	Compute (serverless,	Variable	vCore-hours	Depende del uso
azurerm_mssql_dat...	GP_S_Gen5_2) Storage	5	GB	\$0.575

22.

23.

24. Costos de personal

25.

26. Costos de sueldo del equipo de desarrollo durante la duración del proyecto.

27.

28.

Rol	Cantidad	Salario mensual (S/) por 5 horas al día	Duración (meses)	Subtotal (S/)
Desarrollador Backend	1	600	6	3600
Desarrollador Frontend	1	600	6	3600
Analista de Requerimientos	1	600	3	1800
Total				9000

29. Costos totales del desarrollo del sistema

30.

<i>Categoría</i>	<i>Costo total (S/)</i>
<i>Costos generales</i>	<i>300</i>
<i>Costos operativos</i>	<i>1800</i>
<i>Costos del ambiente</i>	<i>280</i>
<i>Costos de infraestructura</i>	<i>280</i>
<i>Costos de personal</i>	<i>9000</i>
<i>Costo total del proyecto</i>	<i>11,660</i>

31. Análisis de Factibilidad

1.1 Justificación de la Inversión

5.1.1 Beneficios del Proyecto

El desarrollo de este sistema web aportará mejoras significativas en los procesos académicos relacionados con la gestión y organización de documentación técnica. Sus beneficios abarcan tanto aspectos cuantificables como cualitativos, impactando directamente en la eficiencia, accesibilidad y colaboración dentro de la comunidad académica.

Beneficios Tangibles

- **Reducción del tiempo de conversión manual de documentos:** Se estima una reducción del 40% en el tiempo requerido para convertir archivos a Markdown respecto a métodos tradicionales.
- **Disminución de errores de formato y pérdida de contenido:** Gracias a la estructuración automática, se espera una reducción del 30% en inconsistencias en documentos convertidos.
- **Mejora en la productividad académica:** Los estudiantes y docentes podrán centrarse en el contenido técnico, ahorrando tiempo en tareas de edición y organización de documentos.

- **Optimización del almacenamiento y gestión de versiones:** Se prevé una disminución del 25% en la pérdida de información gracias al control de versiones integrado.

Beneficios Intangibles

- **Mayor accesibilidad a la documentación:** Al utilizar Markdown, los documentos serán más ligeros, portables y compatibles con plataformas de publicación académica o repositorios en línea.
- **Fomento de la colaboración académica:** La integración con plataformas como GitHub permitirá un trabajo colaborativo más ágil y organizado entre estudiantes y docentes.
- **Mejora en la estandarización de la documentación:** El sistema contribuirá a uniformizar la forma en que se presentan los documentos técnicos, elevando el nivel académico general.
- **Impulso a la innovación educativa:** La adopción de herramientas modernas fomentará el uso de tecnologías emergentes en el entorno universitario, motivando a los estudiantes a digitalizar sus flujos de trabajo.

5.1.2 Criterios de Inversión

5.1.2.1 Relación Beneficio/Costo (B/C)

<i>Parámetro</i>	<i>Valor</i>
<i>Inversión inicial</i>	<i>S/. 11,660</i>
<i>Tasa de descuento</i>	<i>9%</i>

<i>Periodo</i>	<i>Ingreso (S/.)</i>	<i>Egreso (S/.)</i>	<i>Flujo efectivo (S/.)</i>
<i>0</i>	–	–	<i>– 11,660</i>
<i>1</i>	<i>8,000</i>	<i>1,800</i>	<i>6,200</i>

2	10,000	1,800	8,200
3	12,000	1,800	10,200

B/C
s/. 1.78

Se obtiene S/. 0.78 de utilidad neta por cada sol de egreso operativo

5.1.2.2 Valor Actual Neto (VAN)

Periodo	Ingreso (S/.)	Egreso (S/.)	Flujo efectivo (S/.)
0	–	–	– 11,660
1	8,000	1,800	6,200
2	10,000	1,800	8,200
3	12,000	1,800	10,200

VAN
s/. 8,806.26

VAN (Valor Actual Neto = s/. 8,806.26): Esto indica que el valor presente de los flujos de efectivo esperados es S/. 8,806.26. El VAN positivo sugiere que el proyecto genera más valor que el costo de la inversión inicial, lo que lo hace financieramente viable

<i>Periodo</i>	<i>Ingreso (S/.)</i>	<i>Egreso (S/.)</i>	<i>Flujo efectivo (S/.)</i>
0	–	–	– 11,380
1	8,000	1,800	6,200
2	10,000	1,800	8,200
3	12,000	1,800	10,200

5.1.2.3 Tasa Interna de Retorno (TIR)

TIR
57%

TIR (Tasa Interna de Retorno = 57%): La TIR es mucho mayor que la tasa de descuento (9%). Esto significa que el retorno esperado del proyecto es significativamente superior al costo del capital, lo que lo hace muy atractivo.

4. Evaluación Financiera

La evaluación financiera del proyecto permite analizar si los costos asociados a su desarrollo, implementación y mantenimiento están justificados frente a los beneficios académicos y operativos que generará.

Con un presupuesto estimado de S/ **11,660**, el sistema aportará beneficios significativos como la automatización del procesamiento de documentos técnicos, la mejora en la calidad y presentación de informes, la reducción del tiempo dedicado por docentes a la revisión de trabajos y el fomento del trabajo colaborativo y profesional en entornos reales como GitHub.

Además, el uso de tecnologías abiertas como Markdown y plataformas gratuitas o de bajo costo (GitHub, herramientas en la nube, APIs de IA) reduce considerablemente los costos operativos a largo plazo. Este enfoque hace que el sistema sea escalable, reutilizable y sostenible, permitiendo su aplicación en otras facultades o instituciones.

La relación costo-beneficio es favorable, dado que con una inversión relativamente baja se logra un impacto directo en la formación profesional de los estudiantes, mejorando su productividad, competencias técnicas y experiencia en documentación colaborativa.

En resumen, el proyecto es financieramente viable y genera un alto valor académico en proporción a su inversión.

Anexo 01 – Requerimientos del Sistema { Desarrollo de un Sistema de Conversión y Organización de Documentos Técnicos en Markdown con Estructuración Automática y Control de Versiones para los estudiantes en la facultad de Ingeniería de Sistemas }