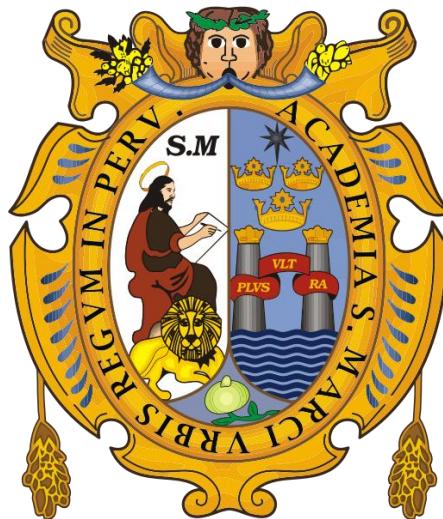


UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
Universidad del Perú, DECANA DE AMÉRICA
FACULTAD DE INGENIERÍA DE SISTEMA E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA DE SOFTWARE



TESIS

Plataforma Web Inteligente con LLMs para la Optimización de la Productividad Académica en los estudiantes de Ingeniería de Software de la UNMSM mediante SCRUM

DOCENTE:

Gamboa Cruzado, Javier Arturo

ALUMNO:

Pérez Acosta, Roddy David
(ORCID: 0009-0002-7005-9502)

Lima - Perú

2025

DEDICATORIAS

Dedico este trabajo a las personas que, con su presencia o con su ejemplo, impulsaron mi determinación en cada etapa de este camino. A quienes creyeron en mi capacidad incluso en los momentos más exigentes, y me recordaron que la disciplina y la constancia siempre abren puertas.

AGRADECIMIENTOS

Expresamos nuestro profundo agradecimiento a la Universidad Nacional Mayor de San Marcos, institución que nos brindó una formación pública, humanista y de excelencia, convirtiéndose en el espacio donde crecimos académica y personalmente.

Nuestro especial reconocimiento al Dr. Javier Arturo Gamboa Cruzado, docente y asesor, por su orientación constante, su exigencia académica y su compromiso con nuestra formación.

Su guía fue determinante para el desarrollo y culminación de esta investigación.

Agradecemos también a los profesores que nos acompañaron a lo largo de la carrera, por compartir sus conocimientos, experiencias y valores; y a todas las personas que, directa o indirectamente, contribuyeron al avance de este trabajo.

Finalmente, dedicamos este logro a nuestros padres, por su apoyo incondicional, su esfuerzo silencioso y la confianza que depositaron en nosotros desde el inicio. Sin ellos, este camino no habría sido posible.

RESUMEN

El presente estudio aborda la problemática de la baja productividad académica en estudiantes de la Escuela Profesional de Ingeniería de Software de la Universidad Nacional Mayor de San Marcos, manifestada en dificultades para organizar actividades, gestionar el tiempo, priorizar tareas y acceder de manera eficiente a información relevante para su desempeño académico.

El objetivo principal fue optimizar la Productividad Académica mediante el desarrollo de una Plataforma Web Inteligente basada en Modelos de Lenguaje de Gran Escala (LLMs) e implementada bajo la metodología ágil SCRUM.

La investigación es de tipo aplicada y empleó un diseño experimental con posprueba y grupos independientes. Se recopilaron 30 mediciones correspondientes al grupo de control (uso de métodos tradicionales de organización y estudio) y 30 mediciones para el grupo experimental (uso de la plataforma inteligente). Para el análisis inferencial se utilizaron pruebas t-Student y U de Mann-Whitney, con un nivel de significancia de 0.05.

La solución tecnológica se desarrolló utilizando Python, Google Colab, Streamlit, ngrok, y librerías como *pandas*, *plotly*, *python-dateutil* y *ics*. Se integró la API de Google Gemini 1.5 para la generación de respuestas, recomendaciones personalizadas y asistencia académica basada en lenguaje natural; además de emplear técnicas de vectorización para la recuperación contextual de información almacenada. La arquitectura contempla un backend ligero desplegado mediante ngrok y una interfaz web interactiva en Streamlit.

Los resultados obtenidos a partir de las pruebas de hipótesis evidenciaron mejoras estadísticamente significativas en todos los indicadores evaluados: organización académica, claridad en la planificación diaria, tiempo empleado en actividades críticas y satisfacción general respecto al proceso de estudio. Los estudiantes que utilizaron la plataforma inteligente mostraron un incremento notable en la eficiencia con la que gestionaron sus actividades académicas.

ABSTRACT

This study addresses the issue of low academic productivity among students of the Software Engineering School at the National University of San Marcos, reflected in difficulties related to organizing activities, managing time, prioritizing tasks, and efficiently accessing information relevant to their academic performance. The main objective was to optimize Academic Productivity through the development of an Intelligent Web Platform based on Large Language Models (LLMs) and implemented using the agile SCRUM methodology.

The research is applied in nature and employed an experimental post-test design with independent groups. A total of 30 measurements were collected for the control group (using traditional study and organization methods) and 30 measurements for the experimental group (using the intelligent platform). For inferential analysis, t-Student and Mann–Whitney U tests were applied, with a significance level set at 0.05.

The technological solution was developed using Python, Google Colab, Streamlit, ngrok, and libraries such as pandas, plotly, python-dateutil, and ics. The Google Gemini 1.5 API was integrated to provide natural language-based academic assistance, personalized recommendations, and generative responses. Additionally, vectorization techniques were employed to enable contextual retrieval of stored information. The system architecture includes a lightweight backend deployed through ngrok and an interactive web interface built with Streamlit.

The results derived from hypothesis testing showed statistically significant improvements across all evaluated indicators: academic organization, clarity in daily planning, time dedicated to critical tasks, and overall satisfaction with the study process. Students who used the intelligent platform demonstrated a substantial increase in the efficiency with which they managed their academic activities.

INTRODUCCIÓN

La productividad académica se ha convertido en un factor determinante para el rendimiento y la formación profesional de los estudiantes universitarios, especialmente en carreras orientadas a la tecnología como Ingeniería de Software. En la Universidad Nacional Mayor de San Marcos, muchos estudiantes enfrentan dificultades para organizar sus actividades, gestionar adecuadamente su tiempo, priorizar tareas y acceder de forma rápida a información relevante para su vida académica. Estas limitaciones generan retrasos, estrés, desorden en la planificación y una disminución significativa en el desempeño, afectando directamente la culminación eficiente de sus obligaciones académicas.

En los últimos años, el avance de la Inteligencia Artificial, particularmente el uso de Modelos de Lenguaje de Gran Escala (LLMs), ha permitido el desarrollo de herramientas capaces de asistir a los estudiantes en la organización, automatización y optimización de sus actividades. A nivel nacional e internacional, universidades e instituciones educativas ya están integrando asistentes inteligentes para mejorar los procesos de aprendizaje, la gestión del tiempo y la eficiencia académica. La presente investigación se justifica en la necesidad de incorporar estas tecnologías en el contexto universitario peruano, proponiendo una solución que brinde acompañamiento personalizado, disponibilidad constante y recomendaciones basadas en el contexto académico del estudiante.

El objetivo principal de este estudio es optimizar la Productividad Académica en los estudiantes de Ingeniería de Software de la UNMSM mediante una Plataforma Web Inteligente, desarrollada con Modelos de Lenguaje de Gran Escala (LLMs) e implementada bajo la metodología ágil SCRUM.

Para demostrar la viabilidad de la propuesta, se plantea la siguiente hipótesis general: Si se utiliza una Plataforma Web Inteligente basada en LLMs y desarrollada con la metodología SCRUM, entonces mejora la Productividad Académica en los estudiantes de Ingeniería de Software de la UNMSM.

La investigación es de tipo aplicada y utiliza un diseño experimental de posprueba con grupo de control. Asimismo, la construcción de la solución tecnológica se gestionó bajo la

metodología ágil SCRUM, abarcando desde la definición del Product Backlog hasta el desarrollo iterativo de los incrementos funcionales mediante Sprints.

Con el propósito de facilitar la comprensión de la presente tesis, el documento ha sido estructurado en cinco capítulos:

Capítulo I: Planteamiento Metodológico. Se presenta la delimitación del problema, la justificación, los objetivos generales y específicos, las hipótesis, las variables operacionales y el diseño metodológico empleado.

Capítulo II: Marco Referencial. Se desarrollan los antecedentes nacionales e internacionales relacionados con la productividad académica, los LLMs, y las plataformas inteligentes. Asimismo, se detalla el sustento teórico que comprende tecnologías como LLMs, vectorización, recuperación de contexto y la metodología SCRUM.

Capítulo III: Construcción de la Solución. Se describe la aplicación práctica de SCRUM, la arquitectura técnica de la plataforma, las tecnologías utilizadas (Python, Google Colab, Streamlit, ngrok, Gemini API, entre otras) y el proceso de desarrollo del sistema web inteligente.

Capítulo IV: Análisis de Resultados y Contrastación de Hipótesis. Se presentan las pruebas de normalidad, los resultados de la posprueba para el grupo de control y el grupo experimental, así como la contrastación estadística de las hipótesis planteadas.

Capítulo V: Conclusiones y Recomendaciones. Se exponen las conclusiones derivadas del análisis y se proponen recomendaciones para futuras investigaciones y mejoras de la plataforma.

Finalmente, se incluyen las referencias bibliográficas, anexos, apéndices y un glosario de términos utilizados en el desarrollo del estudio.

Tabla de Contenidos

CAPÍTULO I: PLANTEAMIENTO METODOLÓGICO	10
1.1. El Problema	10
1.1.1. Descripción de la Realidad Problemática	10
1.1.2 Definición del Problema	13
1.1.3 Enunciado del Problema	15
1.2 Tipo y Nivel de la Investigación	16
1.2.1 Tipo de Investigación	16
1.2.2 Nivel de Investigación	18
1.3. Justificación e Importancia del Trabajo	19
1.4. Objetivos.....	21
1.4.1. Objetivo General	21
1.4.2. Objetivos Específicos	22
1.5. Hipótesis	22
1.5.1 Hipótesis General.....	22
1.5.2 Hipótesis Específicas	22
1.6. Variables e Indicadores.....	23
1.6.1. Variable Dependiente	23
1.6.2. Variable Independiente.....	23
1.6.3. Operacionalización.....	24
1.7. Limitaciones.....	25
Factores Externos	25
Factores Internos	26
1.8. Diseño de la Investigación	28
1.9. Técnicas e Instrumentos para Recolección de Información.....	29
1.9.1. Técnicas de investigación de campo	29
1.9.2. Técnicas de investigación experimental	30
Tabla 7	¡Error! Marcador no definido.
1.11.3. Técnicas de investigación documental	31
1.12. Población y muestra.....	32
1.12.1 Unidad Muestral	32
1.12.2 Población.....	32
1.12.3 Muestra	32
1.12.4 Tipo de muestreo.....	33
CAPÍTULO II: MARCO REFERENCIAL.....	34
2.1 Antecedentes de la investigación	34

2.1.1. Antecedentes Nacionales.....	34
2.1.2 Antecedentes Internacionales	35
2.2 Marco Teórico.....	40
CAPÍTULO III: CONSTRUCCIÓN DE LA SOLUCIÓN	53
3.1. Creación del Product Backlog	53
3.1.1. Visión del Producto y Propósito	54
Tabla 10	¡Error! Marcador no definido.
<i>Stakeholders Identificados</i>	¡Error! Marcador no definido.
3.1.2 Obtención de Requisitos y Perfiles de Usuario	55
3.1.3 Estructura del Backlog: Épicas e Historias de Usuario	58
3.1.4 Detalle de Historias de Usuario y Criterios de Aceptación	60
Tabla 13	¡Error! Marcador no definido.
3.2 Planificación del Sprint.....	64
3.2.1 Duración del Sprint.....	65
3.2.2 El Plan de Trabajo (Sprint Backlog).....	66
Tabla 14	¡Error! Marcador no definido.
Tabla 16	¡Error! Marcador no definido.
3.3 Ejecución del Sprint	72
3.3.1 Daily Scrum	73
3.3.2 Desarrollo de los entregables claves	73
3.4 Revisión del Sprint (Sprint Review)	95
3.4.1 Revisión del Sprint 1 (Backend).....	96
3.4.2 Revisión del Sprint 2 (Interfaz de Usuario – Frontend)	97
3.4.3 Revisión del Sprint 3 (Integración Completa con LLM y Despliegue Operativo)...	97
3.5 Retrospectiva del Sprint (Sprint Retrospective)	98
3.5.1 Retrospectiva del Sprint 1 (Backend Inicial y Modelo de Datos Académicos)	99
3.5.2 Retrospectiva del Sprint 2 (Interfaz Web – Streamlit)	100
3.5.3 Retrospectiva del Sprint 3 (Integración LLM – Base de Datos – Streamlit)	101
CAPÍTULO IV: ANÁLISIS DE RESULTADOS Y CONTRASTACIÓN DE LA HIPÓTESIS..	103
4.1 Resultados.....	103
Tabla 12.....	¡Error! Marcador no definido.
4.2 Nivel de confianza y Grado de significancia	104
4.3 Prueba de Normalidad	104
4.3.1 I1: Tiempo de estudio promedio por día (horas)	105
4.3.2 I2: Tiempo promedio para completar una tarea desde su asignación (días)	105
4.3.3 I3: Tasa de finalización de tareas a tiempo (%)	107

4.3.4 I4: Promedio de notas finales por ciclo (puntos)	108
4.3.5 I5: Nivel de satisfacción del estudiante (Escala de Likert).....	109
4.4 Análisis e Interpretación de Resultados	109
4.4.1 I ₁ : Tiempo de estudio promedio por día	110
4.4.2 I ₂ : Tiempo promedio para completar una tarea desde su asignación	111
4.4.3 I ₃ : Tasa de finalización de tareas a tiempo (%)	113
4.4.4 I ₄ : Promedio de notas finales por ciclo	115
4.4.5. I ₅ : Nivel de Satisfacción del estudiante (Escala de Likert).....	116
4.5 Prueba de Hipótesis.....	118
4.5.1. Prueba para la Hipótesis H ₁	118
4.5.2 Prueba para la Hipótesis H ₂	120
4.5.3 Prueba para la Hipótesis H ₃	122
4.5.4 Prueba para la Hipótesis H ₄	124
4.5.5 Prueba para la Hipótesis H ₅	126
4.6 Discusión de Resultados.....	128
CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES	137
5.1 Conclusiones	137
5.2 Recomendaciones	137
6. REFERENCIAS BIBLIOGRÁFICAS	139

Índice de Figuras

Figura 1 Diagrama AS-IS: Proceso de Organización Académica Actual de los estudiantes de la carrera de Ingeniería de Software de la Universidad Nacional Mayor de San Marcos usando Google Calendar.....	13
Figura 2 Planteamiento del problema y sus variables	17
Figura 3 Flujo de trabajo de la Plataforma Web Inteligente con LLMs	44
Figura 4 Ciclo de SCRUM.....	47
Figura 5 Desorganización y Baja Productividad Académica en los Estudiantes de Ingeniería	49
Figura 6 Uso de Aplicaciones Inteligentes para la Gestión Académica y la Optimización del Tiempo	51
Figura 7 Tablero Kanban en la plataforma Trello utilizado para la gestión del proyecto	73
Figura 8 Patrón de Arquitectura de la Solución Backend	75
Figura 9 Diagrama ER – Tablas y campos	80
Figura 10 Flujo de Consulta y Recomendación en Tiempo Real.....	81
Figura 11 Pantalla principal de interacción del asistente académico, donde se integran los módulos de conversación, registro de actividades.....	85
Figura 12 Prueba de normalidad del indicador I1	105
Figura 13 Prueba de normalidad del indicador I2	106
Figura 14 Prueba de normalidad del indicador I3.....	107

Figura 15 Prueba de normalidad del indicador I4.....	108
Figura 16 Prueba de normalidad del indicador I5.....	109
Figura 17 Resumen para I1(Post): Tiempo de estudio promedio por día	129
Figura 18 Resumen para I2(Post): Tiempo promedio para completar una tarea desde su asignación.....	131
Figura 19 Resumen para I3(Post): Tasa de finalización de tareas a tiempo.	132
Figura 20 Resumen para I4(Post): Promedio de notas por ciclo.	134

Índice de Tablas

Tabla 1 Tabla de indicadores	14
Tabla 2 Tabla de Mapeo de Procesos.....	15
Tabla 3 Tabla de indicadores	23
Tabla 4 Operacionalización de las Variables Independiente	24
Tabla 5 <i>Operacionalización de las Variables Dependientes</i>	24
Tabla 6 Matriz de técnicas e instrumentos: investigación de campo	29
Tabla 7 Matriz de técnicas e instrumentos: investigación experimental.....	30
Tabla 8 Matriz de técnicas e instrumentos: investigación documental.....	31
Tabla 9 Stakeholders Identificados	54
Tabla 10 Perfiles de Usuario – Estudiantes de Ingeniería de la UNMSM	56
Tabla 11 Épicas del Sistema Inteligente con LLMs	58
Tabla 12 Historias de Usuario para la Plataforma Web Inteligente con LLMs	60
Tabla 13 Cronograma de actividades del Sprint 1	67
Tabla 14 Cronograma de actividades del Sprint 2	69
Tabla 15 Cronograma de actividades del Sprint 3	71
Tabla 16 Resultados de Postprueba para Gc y Ge en I1,I2, I3, I4 e I5	103
Tabla 17 Comparación de resultados de Posprueba para Gc y Posprueba para Ge en I1	110
Tabla 18 Comparación de resultados de Posprueba para Gc y Posprueba para Ge en I2	111
Tabla 19 Comparación de resultados de Posprueba para Gc y Posprueba para Ge en I3	113
Tabla 20 Comparación de resultados de Posprueba para Gc y Posprueba para Ge en I4	115
Tabla 21 Resultados de Posprueba para Gc en I5.....	116
Tabla 22 Valores de PosPrueba del Ge y Gc	119
Tabla 23 Prueba t para medias de las dos muestras de la Hipótesis 1	120
Tabla 24 Valores de PosPrueba del Ge y Gc	121
Tabla 25 Prueba t para medias de las dos muestras de la Hipótesis 2	122
Tabla 26 Valores de PosPrueba del Ge y Gc	123
Tabla 27 Prueba t para medias de las dos muestras de la Hipótesis 3	124
Tabla 28 Valores de PosPrueba del Ge y Gc	125
Tabla 29 Prueba t para medias de las dos muestras de la Hipótesis 4	126
Tabla 30 Valores de PosPrueba del Ge y Gc	127
Tabla 31 Prueba t para medias de las dos muestras de la Hipótesis 4	128

CAPÍTULO I: PLANTEAMIENTO METODOLÓGICO

1.1. El Problema

1.1.1. Descripción de la Realidad Problemática

Ámbito Internacional

A nivel internacional, la gestión del tiempo, la organización académica y la productividad estudiantil se han convertido en factores críticos para el éxito universitario, especialmente en los campos de ingeniería, donde las exigencias académicas y de proyectos son elevadas. Diversos estudios han evidenciado que los estudiantes presentan dificultades para mantener un equilibrio adecuado entre sus actividades académicas, laborales y personales, lo que repercute directamente en su rendimiento, niveles de estrés y bienestar general (Nobbe et al., 2024; Bhattacharjee et al., 2024).

El uso de inteligencia artificial (IA), particularmente los modelos de lenguaje de gran escala (LLMs), está transformando la manera en que los sistemas educativos abordan la planificación y personalización del aprendizaje. Investigaciones recientes destacan que los LLMs pueden generar recomendaciones adaptativas, analizar hábitos de estudio y ofrecer retroalimentación personalizada que mejora la autorregulación y el desempeño académico (Mannekote et al., 2024; Goslen et al., 2024).

Asimismo, el uso de aplicaciones móviles inteligentes orientadas a la planificación académica y gestión del tiempo ha mostrado resultados positivos en el fortalecimiento de la disciplina, la puntualidad y la entrega oportuna de tareas. Por ejemplo, el estudio de Wang et al. (2021) demuestra que los recordatorios inteligentes basados en aprendizaje por refuerzo pueden mejorar la adherencia de los usuarios a sus actividades programadas, mientras que el trabajo de Acuña, Sabili y Friginal (2024) resalta la efectividad de aplicaciones móviles para gestionar avisos académicos y reducir el olvido de actividades.

A su vez, en el contexto de educación superior, se observa una tendencia creciente hacia la implementación de plataformas inteligentes que combinan algoritmos de recomendación y modelos predictivos para optimizar horarios, priorizar tareas y sugerir momentos óptimos de estudio según el ciclo circadiano o los niveles de energía del estudiante (Gu et al., 2025; Pelánek et al., 2024). Este tipo de sistemas integran información sobre clases, descansos, actividades recreativas y hábitos personales para ofrecer planes académicos más equilibrados y eficientes.

Ámbito Nacional

En el Perú, la gestión del tiempo y la planificación académica representan un desafío constante para los estudiantes universitarios, especialmente en carreras de alta exigencia como las ingenierías. Las múltiples actividades académicas, laborales y personales que los estudiantes deben equilibrar generan altos niveles de estrés y dificultades para cumplir con los plazos establecidos, afectando su rendimiento y bienestar. Según el Ministerio de Educación (MINEDU, 2024), más del 60% de los estudiantes de educación superior manifiestan problemas en la organización de su tiempo y en la priorización de tareas, lo que repercute directamente en sus calificaciones y en la tasa de deserción académica.

Si bien existen algunas aplicaciones de gestión del tiempo y recordatorios disponibles en el mercado, la mayoría no están adaptadas al contexto académico peruano ni integran enfoques de inteligencia artificial (IA) o modelos de lenguaje (LLMs) que permitan una personalización real. Estas limitaciones evidencian la brecha tecnológica existente entre las necesidades específicas del estudiante universitario y las soluciones digitales disponibles en el país.

En los últimos años, universidades peruanas han empezado a incorporar proyectos basados en IA para el análisis del desempeño y la gestión educativa; sin embargo, su aplicación en el ámbito de la optimización de horarios y la recomendación

personalizada de rutinas académicas aún es incipiente. Estudios como los de Wang et al. (2024) y Fernex et al. (2025) demuestran que los sistemas inteligentes pueden mejorar significativamente la planificación del tiempo, reduciendo la procrastinación y aumentando la productividad, lo cual representa una oportunidad de innovación en el contexto nacional.

Ámbito Institucional

Dentro del contexto institucional de la Universidad Nacional Mayor de San Marcos (UNMSM), particularmente en las facultades de ingenierías, los estudiantes enfrentan grandes desafíos relacionados con la gestión eficiente de su tiempo académico y personal. Las jornadas extensas de clases, los trabajos de laboratorio, los proyectos de investigación y las prácticas profesionales generan una carga académica considerable que, en muchos casos, no es gestionada de manera óptima. Esta situación se ve agravada por la falta de herramientas tecnológicas que permitan organizar, priorizar y optimizar los horarios de estudio y descanso de forma personalizada.

Diversos diagnósticos internos de la UNMSM y encuestas aplicadas a sus estudiantes evidencian que más del 65% de los alumnos de Ingeniería de Software reconocen dificultades para equilibrar sus horarios académicos, laborales y personales, lo cual impacta negativamente en su rendimiento, su salud mental y su productividad (Oficina de Bienestar Universitario, UNMSM, 2024). Además, la falta de recordatorios o alertas efectivas sobre tareas, actividades y evaluaciones genera retrasos y olvidos que repercuten directamente en las calificaciones y el cumplimiento de los objetivos académicos.

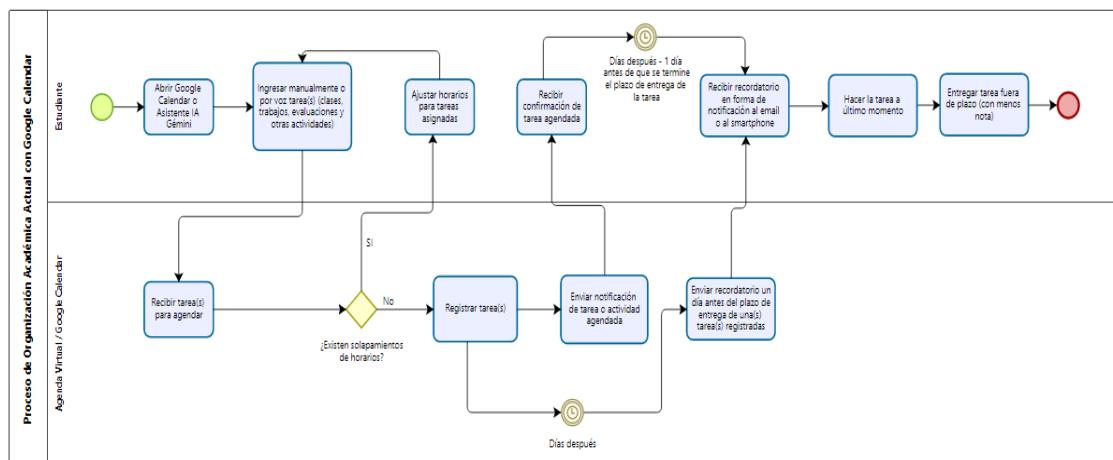
Si bien la universidad cuenta con plataformas institucionales como SUM, Google Classroom y el correo académico, estas herramientas están enfocadas principalmente en la gestión administrativa y no ofrecen funciones inteligentes de planificación

dinámica ni recomendaciones personalizadas basadas en los hábitos, ritmos de energía o preferencias de cada estudiante. Por ello, los alumnos suelen recurrir a aplicaciones externas de productividad (como Google Calendar o Notion) que no están integradas con sus entornos de aprendizaje ni adaptadas al contexto universitario peruano.

En este escenario, la implementación de un sistema inteligente basado en modelos de lenguaje (LLMs) que analice los hábitos académicos y proponga recomendaciones personalizadas de horarios, recordatorios por voz y alertas proactivas se presenta como una alternativa innovadora para fortalecer la organización y la eficiencia de los estudiantes de Ingeniería de Software de la UNMSM. Este tipo de solución tecnológica permitiría no solo mejorar la gestión del tiempo y el cumplimiento de tareas, sino también fomentar el desarrollo de habilidades de autorregulación, disciplina y productividad académica, aspectos fundamentales para la formación de futuros ingenieros competentes y adaptados a entornos digitales.

1.1.2 Definición del Problema

Figura 1 Diagrama AS-IS: Proceso de Organización Académica Actual de los estudiantes de la carrera de Ingeniería de Software de la Universidad Nacional Mayor de San Marcos usando Google Calendar



El proceso de planificación académica y gestión del tiempo de los estudiantes de la carrera de Ingeniería de Software de la Universidad Nacional Mayor de San Marcos (UNMSM) presenta diversas dificultades que impactan directamente en su rendimiento, su bienestar y su eficiencia en el cumplimiento de actividades académicas. Entre los principales inconvenientes identificados se encuentran:

- Tiempo de estudio promedio por día (Bhattacharjee et al., 2024) (Fernex et al., 2025) (Nobbe et al., 2024) (Wang et al., 2021) (Zhao et al., 2024)
- Tiempo promedio para completar una tarea desde su asignación (Acuña et al., 2024) (Diallo & Tudose, 2024) (Goslen et al., 2024) (Li et al., 2025) (Pelánek et al., 2024)
- Tasa de tareas finalizadas a tiempo (Bhattacharjee et al., 2024) (Doğan, 2025) (Krishnamurthy & Shiva, 2025) (Mumcu & Çebi, 2025) (Munson et al., 2025)
- Promedio de notas finales por ciclo ((Filippi & Motyl, 2024) (Ghaffar et al., 2025) (Powell & Riccardi, 2025) (Wang et al., 2024) (Zuo et al., 2025))
- Nivel de satisfacción del estudiante (Acuña et al., 2024) (Mannekote et al., 2024) (Mohamad Zukriyani & Azizan, 2023) (Nobbe et al., 2024) (Ventura Roque Hernández et al., 2023)

Tabla 1 Tabla de indicadores

Indicadores	Valor
Tiempo de estudio promedio por día	4 horas/día
Tiempo promedio para completar una tarea desde su asignación	2 días
Tasa de finalización de tareas a tiempo	70 %
Promedio de notas finales por ciclo	13.5
Nivel de satisfacción del estudiante	5

Tabla 2 Tabla de Mapeo de Procesos

Situación Actual	Situación Propuesta
AS-IS	TO-BE
Bajo tiempo de estudio promedio por día	Alto tiempo de estudio por día
Alto tiempo promedio para completar una tarea desde su asignación	Bajo tiempo promedio para completar una tarea desde su asignación
Baja tasa de finalización de tareas a tiempo	Alta tasa de finalización de tareas a tiempo
Bajo promedio de notas finales por ciclo	Alto promedio de notas finales por ciclo
Bajo Nivel de Satisfacción del estudiante	Alto Nivel de Satisfacción del estudiante

1.1.3 Enunciado del Problema

Problema General

¿En qué medida el uso de una Plataforma Web Inteligente con LLMs aplicando la **Metodología SCRUM**, mejora la **Productividad Académica en los estudiantes de Ingeniería de Software de la UNMSM?**

Problemas Específicos

- ¿En qué medida el uso de una Plataforma Web Inteligente con LLMs aplicando la **Metodología SCRUM**, aumenta el **tiempo de estudio promedio por día** la Productividad Académica en los estudiantes de Ingeniería de Software de la UNMSM?
- ¿En qué medida el uso de una Plataforma Web Inteligente con LLMs aplicando la **Metodología SCRUM**, disminuye el **tiempo promedio para completar una**

tarea desde su asignación la Productividad Académica en los estudiantes de Ingeniería de Software de la UNMSM?

- ¿En qué medida el uso de una **Plataforma Web Inteligente con LLMs** aplicando la **Metodología SCRUM**, aumenta la **tasa de finalización de tareas a tiempo** la Productividad Académica en los estudiantes de Ingeniería de Software de la UNMSM?
- ¿En qué medida el uso de una **Plataforma Web Inteligente con LLMs** aplicando la **Metodología SCRUM**, aumenta el **promedio de notas finales por ciclo** la Productividad Académica en los estudiantes de Ingeniería de Software de la UNMSM?
- ¿En qué medida el uso de una **Plataforma Web Inteligente con LLMs** aplicando la **Metodología SCRUM**, aumenta el **nivel de satisfacción del estudiante** la Productividad Académica en los estudiantes de Ingeniería de Software de la UNMSM?

1.2 Tipo y Nivel de la Investigación

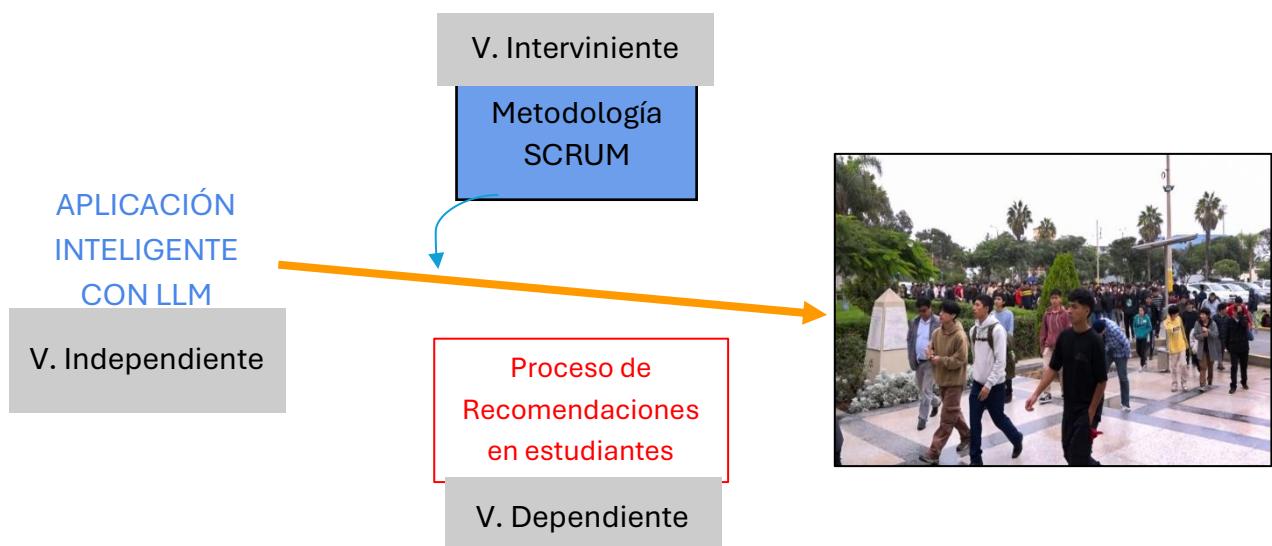
1.2.1 Tipo de Investigación

La presente investigación es de tipo aplicada, dado que busca resolver un problema real y concreto que enfrentan los estudiantes de Ingeniería de Software de la Universidad Nacional Mayor de San Marcos: la ineficiente gestión del tiempo académico y personal. Este estudio se centra en el diseño e implementación de un sistema inteligente basado en modelos de lenguaje de gran escala (LLMs), que permita optimizar la planificación académica, generar recordatorios personalizados y mejorar la productividad estudiantil a través de una Plataforma Web desarrollada bajo la metodología SCRUM.

Según Martínez y Pérez (2020), la investigación aplicada se orienta hacia la solución de problemas prácticos mediante la utilización de conocimientos científicos y tecnológicos para mejorar procesos o generar innovaciones con impacto directo en la sociedad. En este sentido, la presente investigación no solo busca describir un fenómeno, sino aplicar la inteligencia artificial y las técnicas de optimización para ofrecer una solución funcional y tecnológica al problema identificado en el contexto educativo universitario.

Asimismo, autores como Wang et al. (2024) y Filippi y Motyl (2024) señalan que la aplicación de la inteligencia artificial en la educación tiene un enfoque esencialmente práctico, ya que permite desarrollar herramientas tecnológicas que personalizan el aprendizaje y mejoran el desempeño académico. De igual manera, Acuña et al. (2024) destacan que el desarrollo de aplicaciones móviles de recordatorio académico representa una forma tangible de aplicar el conocimiento tecnológico al beneficio de la comunidad educativa.

Figura 2 Planteamiento del problema y sus variables



1.2.2 Nivel de Investigación

El nivel de investigación del presente estudio es descriptivo y predictivo, ya que busca analizar y explicar cómo el uso de una plataforma inteligente de planificación académica, basado en modelos de lenguaje (LLMs), influye en la gestión del tiempo y el rendimiento académico de los estudiantes de Ingeniería de Software de la Universidad Nacional Mayor de San Marcos. Tal como sostiene Ramírez (2022), en investigaciones orientadas al desarrollo e implementación de tecnologías educativas inteligentes, el enfoque predictivo permite evaluar la eficacia de las soluciones tecnológicas en contextos reales, contrastando los resultados obtenidos con los esperados. De esta manera, el presente estudio combina la descripción del contexto institucional y académico actual con la predicción del efecto del sistema inteligente, mediante la comparación entre un grupo experimental (usuarios del sistema) y un grupo de control (estudiantes sin acceso a la herramienta).

Nivel Descriptivo

El presente estudio adopta un nivel descriptivo debido a que busca analizar y caracterizar las variables relacionadas con la gestión del tiempo académico, el cumplimiento de actividades y la satisfacción de los estudiantes frente a las herramientas digitales utilizadas. Este nivel permitirá identificar los patrones de comportamiento y las principales dificultades que enfrentan los estudiantes en la organización de sus horarios y tareas. Según Chanco (2023), el método descriptivo posibilita examinar las características de las variables de estudio para obtener una comprensión detallada de la realidad observada. En este contexto, el análisis descriptivo ayudará a determinar las limitaciones actuales en la planificación académica y servirá como base para medir los efectos del sistema inteligente propuesto.

Nivel Predictivo

La investigación también adopta un nivel predictivo, ya que la plataforma propuesta integra modelos de aprendizaje automático y procesamiento del lenguaje natural (LLMs) con el fin de generar recomendaciones personalizadas sobre horarios, hábitos de estudio y recordatorios automatizados. Según Vásquez (2023), los modelos predictivos permiten anticipar comportamientos futuros mediante el análisis de variables históricas y contextuales, posibilitando la toma de decisiones más efectivas. En este caso, el modelo predictivo se aplicará para estimar cómo la implementación del sistema impactará en la productividad y el rendimiento académico de los estudiantes. De esta manera, se espera prever mejoras en la organización del tiempo, la finalización de tareas y la percepción de utilidad de las herramientas tecnológicas utilizadas en el ámbito universitario.

1.3. Justificación e Importancia del Trabajo

La gestión ineficiente del tiempo académico se ha convertido en uno de los principales factores que afectan el rendimiento y bienestar de los estudiantes universitarios a nivel global. En el contexto actual de digitalización educativa, los estudiantes de Ingeniería de Software enfrentan una creciente carga de actividades académicas, prácticas, proyectos, y responsabilidades personales que dificultan mantener un equilibrio adecuado entre el estudio, el descanso y la vida cotidiana. Esta desorganización repercute directamente en la productividad, la motivación y el cumplimiento de metas académicas.

Conveniencia: La presente investigación resulta conveniente debido a su potencial para mejorar significativamente la gestión del tiempo y la productividad académica de los estudiantes universitarios, en particular de la carrera de Ingeniería de Software de la Universidad Nacional Mayor de San Marcos (UNMSM). Actualmente, muchos estudiantes enfrentan dificultades para organizar sus horarios de clases, prácticas, trabajos y actividades personales, lo que genera estrés, bajo rendimiento

académico y retrasos en la culminación de proyectos o asignaciones. El desarrollo de una Plataforma Web Inteligente con modelos de lenguaje LLMs permitirá automatizar la planificación de horarios, ofrecer recomendaciones personalizadas y generar recordatorios inteligentes, optimizando así el uso del tiempo disponible.

Relevancia Social: Desde una perspectiva social, esta investigación busca atender una problemática que afecta a un amplio sector de la comunidad universitaria: la desorganización en la gestión del tiempo académico y personal. La sobrecarga de actividades y la falta de herramientas de planificación adecuadas inciden negativamente en la salud mental, la productividad y la motivación de los estudiantes. Implementar una Plataforma Web inteligente con IA permitirá a los usuarios equilibrar sus horarios, fortalecer la disciplina académica y mejorar su bienestar general.

Implicaciones Prácticas: En el ámbito práctico, este proyecto propone la implementación de una Plataforma Web desarrollada bajo la metodología ágil SCRUM, que integrará un sistema de recomendación y planificación inteligente mediante LLMs. Esta plataforma permitirá a los estudiantes registrar sus horarios fijos (clases, trabajo), tiempos de descanso y actividades personales, para que el sistema genere automáticamente recomendaciones óptimas basadas en sus hábitos, niveles de energía y prioridades académicas.

De acuerdo con Filippi y Motyl (2024) y Doğan (2025), los modelos de lenguaje pueden analizar patrones de comportamiento y generar sugerencias contextualizadas, mejorando la toma de decisiones en entornos educativos. Además, el sistema incluirá recordatorios por voz personalizados que alertarán al estudiante antes de cada actividad o evaluación, aumentando su cumplimiento y reduciendo el olvido de tareas importantes.

Valor Teórico: Desde una perspectiva teórica, esta investigación busca ampliar la comprensión del impacto que tienen los modelos de lenguaje de gran escala (LLMs) y la inteligencia artificial aplicada en los procesos de planificación académica y gestión del tiempo en entornos universitarios. A través del análisis de las variables

relacionadas con la organización del tiempo, la productividad, el cumplimiento de tareas y la satisfacción estudiantil, se pretende identificar patrones de comportamiento y correlaciones que permitan explicar cómo las recomendaciones personalizadas y los recordatorios inteligentes pueden influir positivamente en el rendimiento académico.

Utilidad Metodológica: En el ámbito metodológico, la investigación empleará la metodología ágil SCRUM para el desarrollo iterativo e incremental del aplicativo móvil, garantizando la entrega continua de funcionalidades y la validación constante con usuarios reales (estudiantes de ingeniería). Esta metodología facilita la gestión eficiente del proyecto, promoviendo la colaboración interdisciplinaria, la flexibilidad ante cambios y la mejora continua del sistema.

Además, el componente de inteligencia artificial se fundamentará en el uso de Large Language Models (LLMs) integrados en un flujo de análisis y recomendación que procesa los datos de horarios, hábitos y preferencias de los estudiantes. Según Doğan (2025) y Filippi & Motyl (2024), los LLMs son herramientas de gran potencial para la personalización del aprendizaje, ya que permiten inferir patrones y generar respuestas adaptadas al contexto individual del usuario.

La metodología propuesta permitirá recopilar, analizar y optimizar datos académicos y personales de manera estructurada, transformándolos en información útil para generar recomendaciones automatizadas y recordatorios personalizados. De este modo, se garantiza que el sistema desarrollado no solo tenga un valor práctico, sino también un marco metodológico replicable para futuras aplicaciones de IA en entornos educativos y de productividad estudiantil.

1.4. Objetivos

1.4.1. Objetivo General

Mejorar la [Productividad Académica en los estudiantes de Ingeniería de Software de la UNMSM](#), mediante una [Plataforma Web Inteligente con LLMs](#), desarrollado con la Metodología SCRUM.

1.4.2. Objetivos Específicos

- Aumentar el tiempo de estudio promedio por día la Productividad Académica en los estudiantes de Ingeniería de Software de la UNMSM.
- Disminuir el tiempo promedio para completar una tarea desde su asignación la Productividad Académica en los estudiantes de Ingeniería de Software de la UNMSM.
- Aumenta la tasa de finalización de tareas a tiempo la Productividad Académica en los estudiantes de Ingeniería de Software de la UNMSM.
- Aumenta el promedio de notas finales por ciclo la Productividad Académica en los estudiantes de Ingeniería de Software de la UNMSM.
- Aumenta el nivel de satisfacción del estudiante la Productividad Académica en los estudiantes de Ingeniería de Software de la UNMSM.

1.5. Hipótesis

1.5.1 Hipótesis General

Si se usa una Plataforma Web Inteligente con LLMs, aplicando la Metodología SCRUM, entonces mejora la Productividad Académica en los estudiantes de ingeniería de la UNMSM.

1.5.2 Hipótesis Específicas

- a) Si se usa una Plataforma Web Inteligente con LLMs, aplicando la Metodología SCRUM, entonces aumenta el Tiempo de estudio promedio por día.
- b) Si se usa una Plataforma Web Inteligente con LLMs, aplicando la Metodología SCRUM, entonces disminuye el Tiempo promedio para completar una tarea desde su asignación.
- c) Si se usa una Plataforma Web Inteligente con LLMs, aplicando la Metodología SCRUM, entonces aumenta la Tasa de finalización de tareas a tiempo.

- d) Si se usa una Plataforma Web Inteligente con LLMs, aplicando la Metodología SCRUM, entonces aumenta el Promedio de notas finales por ciclo.
- e) Si se usa una Plataforma Web Inteligente con LLMs, aplicando la Metodología SCRUM, entonces el Nivel de satisfacción del estudiante.

1.6. Variables e Indicadores

1.6.1. Variable Dependiente

- a) VI: Plataforma Web Inteligente con LLMs.
- b) VD: Productividad Académica en los estudiantes de Ingeniería de Software de la UNMSM.
- c) VInterv: Metodología SCRUM

1.6.2. Variable Independiente

Tabla 3 Tabla de indicadores

Variables	Indicadores
1. Independiente:	
Plataforma Web Inteligente con LLMs	<ul style="list-style-type: none"> ● Presencia_Ausencia
	<ul style="list-style-type: none"> ● Tiempo de estudio promedio por día.
2. Dependiente:	
Productividad Académica en los estudiantes de la FISI de la UNMSM	<ul style="list-style-type: none"> ● Tiempo promedio para completar una tarea desde su asignación.
	<ul style="list-style-type: none"> ● Tasa de finalización de tareas a tiempo.
	<ul style="list-style-type: none"> ● Promedio de notas finales por ciclo.
	<ul style="list-style-type: none"> ● Nivel de satisfacción del estudiante.

1.6.3. Operacionalización

a. Variable Independiente: Plataforma Web Inteligente con LLMs

Tabla 4 Operacionalización de las Variables Independiente

Indicador	Índice
Presencia_Ausencia	No, Sí

b. Variable Dependiente: Productividad Académica en los estudiantes de Ingeniería de Software de la UNMSM.

Tabla 5 Operacionalización de las Variables Dependientes

Dimensión	Indicador	Índice	Unidad de medida	Fórmula	Unidad observación
Eficiencia	Tiempo de estudio promedio por día	[0 – 16]	Horas	Total horas de estudio diario / Número de estudiantes	Autoinforme / Revisión manual
	Tiempo promedio para completar una tarea desde su asignación		Días	Suma de tiempos de cada tarea desde asignación hasta entrega / N° de tareas	
Productividad	Tasa de finalización de tareas a tiempo	[0 – 100]	Porcentaje (%)	(Número de tareas completadas a tiempo / Total de tareas asignadas) × 100	Autoinforme / Revisión manual
Rendimiento Académico	Promedio de notas por ciclo	[0 – 20]	Puntos	Suma de notas obtenidas /	Autoinforme / Revisión manual

				Número de asignaturas	
Satisfacción / Experiencia	Nivel de satisfacción del estudiante	[Totalment e en desacuerd o (1), En desacuerd o (2), Neutral (3), De acuerdo (4), Totalment e de acuerdo (5)]	Percepción del estudiante (asignado por nivel)	Promedio de puntuaciones de percepción de utilidad de la plataforma web	Autoinforme / Revisión manual

1.7. Limitaciones

Factores Externos

a) Regulaciones sobre la Privacidad de Datos

Aunque la plataforma no gestionará información sensible como datos personales o calificaciones institucionales, se respetarán los principios establecidos en la Ley N.º 29733 – Ley de Protección de Datos Personales.

Toda la información proporcionada por el estudiante será utilizada únicamente con fines de recomendación y planificación, bajo un modelo de anonimización local, donde los datos no serán compartidos ni almacenados en servidores externos.

Asimismo, el diseño contempla mecanismos de consentimiento informado y opciones de eliminación de datos, garantizando el cumplimiento de normas éticas y legales de privacidad.

b) Variabilidad en el Uso Tecnológico

Un factor externo por considerar es la variabilidad en el acceso y uso de la tecnología móvil entre los estudiantes universitarios. Aunque la mayoría cuenta con dispositivos inteligentes, las diferencias en conectividad, capacidad de almacenamiento o familiaridad con herramientas tecnológicas podrían influir en el nivel de adopción de la plataforma. Por ello, la interfaz será intuitiva, ligera y multiplataforma, con funciones adaptadas a distintos niveles de experiencia tecnológica.

c) Cambios Institucionales o Académicos

Las modificaciones en los horarios, plataformas o metodologías de enseñanza de la Universidad Nacional Mayor de San Marcos podrían requerir ajustes en el funcionamiento de la plataforma. Por ejemplo, cambios en los calendarios académicos o en las plataformas institucionales como SUM o Google Classroom pueden afectar la planificación automatizada.

Sin embargo, gracias a la naturaleza modular y adaptable del sistema, estos cambios podrán integrarse fácilmente mediante actualizaciones periódicas del software.

Factores Internos

a) Calidad de los Datos Ingresados por los Usuarios

La calidad de la información que los estudiantes proporcionen a la plataforma será un factor determinante en el rendimiento del sistema. Dado que los datos (horarios, actividades, preferencias y hábitos) serán introducidos directamente por cada usuario, la precisión de las recomendaciones dependerá de la veracidad y coherencia de esa información.

Si los estudiantes ingresan datos incompletos o inconsistentes —por ejemplo, horarios erróneos o actividades omitidas—, las sugerencias y recordatorios

generados por el sistema podrían no ajustarse de manera óptima a sus necesidades reales. Para mitigar este riesgo, la plataforma incluirá validaciones automáticas y guías interactivas que orienten al usuario durante el registro de sus actividades.

b) Capacidad Computacional y Tecnológica

El procesamiento y ejecución del sistema no demanda una infraestructura compleja, ya que los LLMs utilizados estarán preentrenados y solo se emplearán de forma local o mediante APIs ligeras para generar recomendaciones y alertas. Aun así, es necesario contar con equipos de desarrollo con capacidad de procesamiento adecuada y conexión estable a internet para realizar pruebas y actualizaciones.

El uso de tecnologías de código abierto y servicios escalables permitirá mantener bajos los requerimientos técnicos y garantizar la continuidad del proyecto incluso con recursos limitados.

c) Diseño y Usabilidad del Sistema

La interfaz de usuario (UI) y la experiencia de usuario (UX) constituyen un factor crítico para la aceptación del sistema entre los estudiantes. Si la plataforma no resulta intuitiva, visualmente agradable o fácil de usar, su adopción podría verse limitada, independientemente de la precisión de las recomendaciones.

Por ello, el diseño se basará en principios de interacción simple, navegación fluida y retroalimentación inmediata, priorizando una experiencia confortable que fomente el uso constante y la adherencia al sistema de planificación.

d) Interpretación y Adaptación de las Recomendaciones

La efectividad del sistema dependerá también de la capacidad del modelo para adaptar las recomendaciones a los patrones individuales de cada estudiante,

considerando sus ritmos circadianos, disponibilidad y prioridades académicas.

Una interpretación inadecuada de los hábitos o preferencias podría generar alertas poco útiles o en momentos inoportunos.

Para contrarrestar este riesgo, se implementarán mecanismos de retroalimentación continua, permitiendo que el estudiante califique la utilidad de las recomendaciones y el sistema aprenda progresivamente a optimizar sus sugerencias de manera personalizada.

1.8. Diseño de la Investigación

Diseño de Posprueba con grupo de control

RGe	X	O1
RGc	-	O2

Donde:

R = Elección Aleatoria de los elementos del Grupo.

Ge = Grupo experimental, conformado por los procesos relacionados con la Productividad Académica en los estudiantes de Ingeniería de Software de la UNMSM, a los cuales se les aplicará el estímulo (uso de la Plataforma Web Inteligente con LLMs).

Gc = Grupo de control, conformado por los procesos relacionados con la Productividad Académica en los estudiantes de Ingeniería de Software de la UNMSM, al que no se le aplicará el estímulo (uso de la Plataforma Web Inteligente con LLMs).

O1: Datos de posprueba para los indicadores de la variable dependiente, mediciones de prueba del grupo experimental.

O2: Datos de posprueba para los indicadores de la variable dependiente, mediciones de posprueba para el grupo de control.

X = Plataforma Web Inteligente con LLMs

- = Falta de estímulo

Descripción: Se conforman un grupo experimental (Ge) y un grupo de control (Gc), a los cuales se les asignan aleatoriamente elementos de los grupos (R). A continuación, se les

aplica un estímulo o tratamiento experimental, en este caso, la Plataforma Web Inteligente con LLMs como estímulo (X) al grupo experimental (Ge), con el fin de recolectar los datos de la posprueba para los indicadores de la variable dependiente (O1). Por otro lado, al grupo de control (Gc) se le otorga la ausencia de estímulo (-) para obtener los datos de posprueba relacionados con los indicadores de la variable independiente (O2).

1.9. Técnicas e Instrumentos para Recolección de Información

1.9.1. Técnicas de investigación de campo

Tabla 6 Matriz de técnicas e instrumentos: investigación de campo

Técnicas	Instrumentos
Observación Directa: Sistemática o estructurada	Diario de campo: Registro de las interacciones de los estudiantes con la Plataforma Web, observando cómo utilizan las recomendaciones, ajustes de horarios y recordatorios. Fichas de observación: Registro sistemático de cambios en los procesos de planificación académica, como priorización de tareas, tiempos de estudio y cumplimiento de actividades. Capturas de pantalla o grabaciones de uso de la plataforma web (opcional, con consentimiento): Documentación visual de la implementación y ajustes realizados en los horarios y recordatorios.
Observación Indirecta Revisión de documentos	Revisión de documentos: Análisis de los horarios, tareas y actividades que los estudiantes proporcionan en la plataforma web para identificar patrones en la planificación y cumplimiento.
Consulta a bases de datos	Encuestas o cuestionarios breves dentro de la plataforma

web: Recolección de percepciones sobre la utilidad de la planificación, recordatorios y recomendaciones, midiendo el nivel de satisfacción del proceso.

1.9.2. Técnicas de investigación experimental

Tabla 7 Matriz de técnicas e instrumentos: investigación experimental

Técnicas	Instrumentos
Seguimiento del desempeño en la planificación académica	Métricas de la plataforma: Registro del uso de recomendaciones, ajustes de horarios, activación de recordatorios y priorización de tareas por parte de los estudiantes. Fichas de observación: Comparación entre los procesos de planificación manual de los estudiantes y el uso de la plataforma web, enfocándose en la organización de horarios, cumplimiento de tareas y gestión del tiempo.
Comparación entre procesos previos y optimizados con la plataforma web	Cuadro de benchmarking de procesos: Contrastar los resultados de la planificación y cumplimiento de actividades antes y después de usar la plataforma web para identificar mejoras en eficiencia y productividad. Análisis estadístico: Procesamiento de datos proporcionados por los estudiantes (como tiempos estimados para completar tareas, cumplimiento de horarios y percepción de utilidad) para determinar mejoras significativas en organización y satisfacción.

1.11.3. Técnicas de investigación documental

Tabla 8 *Matriz de técnicas e instrumentos: investigación documental*

Técnicas	Instrumentos
Revisión de libros y artículos académicos	Fichas bibliográficas: Para registrar teorías, conceptos de planificación académica, aplicaciones móviles educativas, sistemas inteligentes y metodologías ágiles como SCRUM. Libreta de apuntes: Para resumir ideas clave sobre gestión del tiempo, recordatorios inteligentes y asistentes basados en IA.
Revisión de informes institucionales y académicos	Fichas de registro: Para clasificar información sobre políticas educativas, normativas internas de la UNMSM, estructura de horarios y plataformas de gestión académica como SUM y Google Classroom. Documentos digitales: Para extraer datos sobre lineamientos académicos, horarios, ciclos de evaluación y estrategias de gestión del tiempo estudiantil.
Consulta de fuentes en línea y bases de datos académicas	Acceso a bases de datos académicas: Para obtener artículos recientes sobre inteligencia artificial, aplicaciones móviles, LLMs, planificación académica y recordatorios inteligentes. Revisión de páginas web institucionales:

Información sobre normativas de la UNMSM,
plataformas académicas y herramientas
tecnológicas disponibles para estudiantes.

1.12. Población y muestra

1.12.1 Unidad Muestral

La unidad de análisis será el proceso de planificación académica de los estudiantes de FISI de la UNMSM, incluyendo la creación y organización de horarios, asignación de tiempos de estudio, registro de actividades personales y académicas, y la interacción con la Plataforma Web para recibir recomendaciones y recordatorios inteligentes.

1.12.2 Población

La población de estudio son todos los procesos de planificación académica generados por los estudiantes de FISI de la UNMSM, considerando los distintos horarios, asignaciones, actividades y rutinas que cada estudiante realiza a lo largo de un ciclo académico.

1.12.3 Muestra

La muestra estará constituida por 50 procesos de planificación académica registrados por los estudiantes de la carrera de Ingeniería de Software de la Facultad de Ingeniería de Sistemas e Informática (FISI) de la UNMSM. Se utilizará un muestreo no probabilístico por conveniencia, considerando únicamente aquellos procesos ingresados voluntariamente en la Plataforma Web y que cuenten con información completa sobre horarios, actividades académicas y tareas. Este tipo de muestreo permite obtener datos suficientes y accesibles para evaluar la mejora en la productividad académica, conforme al diseño experimental planteado.

1.12.4 Tipo de muestreo

Muestreo no probabilístico por conveniencia. Este método es apropiado para el diagnóstico y evaluación inicial de los procesos, ya que permite obtener datos de manera rápida y eficiente de los procesos disponibles en la Plataforma Web, priorizando aquellos que estén completos y sean representativos de diferentes tipos de estudiantes y rutinas académicas.

CAPÍTULO II: MARCO REFERENCIAL

2.1 Antecedentes de la investigación

2.1.1. Antecedentes Nacionales

En relación con investigaciones previas sobre sistemas de recomendación académica, gestión del tiempo y uso de modelos de lenguaje para personalizar la planificación de actividades, Acuña, Sabili y Friginal (2024) desarrollaron MARA, una Plataforma Web de recordatorios académicos, cuyo objetivo fue mejorar la organización de tareas y la gestión del tiempo en estudiantes universitarios. Emplearon un diseño experimental con estudiantes de distintas facultades, y los resultados mostraron que el uso de la plataforma incrementó significativamente la puntualidad en la realización de tareas y la percepción de control sobre las actividades académicas. Concluyeron que los sistemas de recordatorios personalizados son herramientas efectivas para fomentar la autorregulación y disminuir la procrastinación estudiantil.

Bashab et al. (2022) analizaron técnicas de optimización en la resolución de problemas de asignación de horarios universitarios, considerando restricciones académicas y preferencias de los estudiantes. La investigación demostró que el uso de métodos híbridos y algoritmos de optimización puede mejorar la distribución de horarios y maximizar la eficiencia del tiempo de estudio. Esto evidencia la necesidad de sistemas que integren la planificación de horarios con recordatorios inteligentes para aumentar la efectividad del aprendizaje.

Bhattacharjee et al. (2024) exploraron el uso de Large Language Models (LLMs) en la personalización de estrategias de estudio para reducir la procrastinación académica. Su estudio concluyó que los LLMs pueden ofrecer sugerencias adaptadas al perfil del estudiante, identificando patrones de comportamiento y ofreciendo recomendaciones sobre cómo organizar mejor el tiempo, priorizar tareas y mantener la constancia en las actividades académicas.

Fernex, Filippi, Maghraoui y Rachdi (2025) propusieron un marco técnico para el desarrollo de aplicaciones de gestión del tiempo estudiantil basado en inteligencia artificial, que incluía recopilación de datos de hábitos de estudio, modelado predictivo y retroalimentación personalizada. La investigación destacó la importancia de combinar notificaciones inteligentes con recomendaciones personalizadas, mostrando mejoras significativas en la puntualidad y en la organización de actividades de los estudiantes universitarios.

Zhao et al. (2024) diseñaron y evaluaron una Plataforma Web de autoayuda para gestión del tiempo dirigida a estudiantes universitarios, implementando recordatorios en momentos óptimos basados en algoritmos de aprendizaje automático. Los resultados indicaron un aumento en la productividad y en la adherencia a horarios de estudio planificados, reforzando la eficacia de integrar tecnología, notificaciones inteligentes y recomendaciones personalizadas en procesos de planificación académica.

2.1.2 Antecedentes Internacionales

En cuanto a investigaciones internacionales sobre sistemas de planificación académica y recordatorios inteligentes, Panta et al. (2022) desarrollaron un sistema predictivo para optimizar la asignación de tareas y horarios de estudio utilizando técnicas de aprendizaje automático. Implementaron modelos de regresión y análisis de patrones de comportamiento de estudiantes para predecir la carga académica y ajustar automáticamente las recomendaciones de estudio. Los resultados mostraron que la integración de modelos predictivos permite mejorar la eficiencia en la organización de tareas y reducir la procrastinación, concluyendo que los sistemas basados en aprendizaje automático pueden personalizar la gestión del tiempo de los estudiantes de manera efectiva.

Osawa et al. (2020) se enfocaron en predecir estudiantes de alto riesgo de retraso académico mediante modelos de aprendizaje supervisado, aplicando árboles de decisión y técnicas de gradiente potenciado a una muestra representativa de estudiantes universitarios. Los resultados indicaron que los registros previos de cumplimiento de tareas fueron los predictores más relevantes para identificar estudiantes con riesgo de bajo rendimiento, concluyendo que la anticipación de actividades críticas permite diseñar alertas y recordatorios personalizados que mejoren la adherencia a los planes de estudio.

Agner et al. (2020) evaluaron sistemas inteligentes de gestión de horarios académicos y optimización de recursos en universidades brasileñas, utilizando análisis de regresión multivariante para medir el impacto de distintas estrategias de planificación en la eficiencia estudiantil. Concluyeron que la planificación automatizada basada en datos históricos y preferencias individuales puede reducir conflictos de horarios y mejorar la distribución de la carga académica.

Mazo (2023) desarrolló un modelo predictivo de rendimiento académico basado en técnicas de clustering y clasificación de estudiantes según hábitos de estudio y cargas de tareas. La investigación demostró que el uso de modelos predictivos permite segmentar a los estudiantes y ofrecer recomendaciones personalizadas para mejorar su desempeño, destacando la importancia de integrar algoritmos de aprendizaje supervisado y no supervisado en aplicaciones de planificación académica.

Yeongah et al. (2022) implementaron modelos supervisados como regresión logística, random forest y XGBoost para predecir el cumplimiento de actividades académicas a nivel universitario. Los resultados mostraron que los algoritmos de XGBoost ofrecieron la mayor precisión al identificar los momentos de alta probabilidad de procrastinación, permitiendo el envío oportuno de recordatorios y sugerencias de reorganización de

tareas. Concluyeron que los sistemas inteligentes basados en LLMs y técnicas predictivas pueden personalizar la gestión del tiempo de los estudiantes.

Iqbal et al. (2024) analizaron la predicción del desempeño académico mediante algoritmos de regresión, árboles de decisión y bosques aleatorios, aplicados a variables como hábitos de estudio, tiempo de dedicación y carga de tareas. Los resultados indicaron que los modelos de bosques aleatorios fueron los más precisos, mostrando que la integración de aprendizaje automático en aplicaciones de planificación académica permite generar recomendaciones precisas para maximizar la productividad y reducir la procrastinación.

Sazzad (2023) evaluó distintos algoritmos de regresión y aprendizaje supervisado para optimizar la planificación de tareas en entornos académicos. Los resultados mostraron que XGBoost y Gradient Boosting ofrecieron la mayor precisión, recomendando el desarrollo de aplicaciones web y móviles basadas en estos modelos para mejorar la accesibilidad y eficiencia de la planificación académica.

Carreira et al. (2021) compararon distintos enfoques de aprendizaje automático para predecir la eficiencia en la realización de tareas académicas y el tiempo necesario para completarlas. Encontraron que XGBoost y Random Forest ofrecieron mayor precisión en predicciones, mientras que modelos más simples, como Simple Tree, fueron más rápidos, concluyendo que la elección del modelo depende del equilibrio entre precisión y eficiencia computacional.

Zhang et al. (2023) desarrollaron un sistema móvil de gestión del tiempo basado en aprendizaje profundo para predecir momentos de baja productividad en estudiantes universitarios. Utilizaron redes neuronales LSTM para modelar patrones diarios de estudio y enviar recordatorios adaptativos. El estudio demostró que los recordatorios personalizados basados en predicción temporal reducen significativamente la procrastinación y mejoran el cumplimiento de planes académicos.

Hernández y Park (2022) evaluaron un sistema inteligente de recomendación de tareas para estudiantes mediante técnicas de clustering y filtrado colaborativo. Encontraron que la agrupación dinámica de estudiantes según hábitos y rendimiento permitió generar sugerencias más precisas de priorización de actividades, mejorando la eficiencia del estudio y la organización semanal.

Singh et al. (2021) analizaron el impacto de las notificaciones inteligentes en la adherencia a los cronogramas académicos en estudiantes de ingeniería. Aplicaron modelos de aprendizaje supervisado para identificar períodos de alto riesgo de abandono de tareas. Concluyeron que los sistemas de alertas contextuales incrementan la tasa de finalización y disminuyen los retrasos en trabajos prácticos.

Rashid y Abdullah (2024) propusieron un asistente académico inteligente basado en LLMs para planificar actividades estudiantiles, integrando generación de recordatorios, reorganización automática de agendas y explicación de prioridades. Los resultados mostraron una mejora significativa en la capacidad de los estudiantes para distribuir adecuadamente su tiempo y completar tareas sin retrasos.

Kimura et al. (2023) desarrollaron un sistema de planificación académica que combina técnicas de reinforcement learning y análisis del comportamiento real de estudiantes. El modelo optimizaba rutas de estudio diario y recomendaba bloques de tiempo óptimos para cada actividad. El estudio concluyó que los sistemas híbridos aumentan la constancia en los hábitos de estudio y facilitan la autorregulación académica.

Investigaciones recientes destacan la importancia de integrar LLMs y notificaciones inteligentes en aplicaciones académicas. Por ejemplo, Bhattacharjee et al. (2024), Fernex et al. (2025), Zhao et al. (2024) y Munson et al. (2025) concluyen que las aplicaciones móviles que combinan planificación de tareas, recordatorios contextuales y recomendaciones personalizadas aumentan la adherencia a los horarios de estudio,

mejoran la organización de actividades y reducen significativamente la procrastinación entre estudiantes universitarios.

Asimismo, Cheung y Lam (2024) desarrollaron un sistema adaptativo de recordatorios académicos basado en modelos secuenciales GRU y LSTM para anticipar episodios de desorganización en estudiantes universitarios. Los resultados demostraron que la predicción temprana de lapsos de productividad permite enviar recordatorios preventivos que reducen en más del 20% los retrasos en la entrega de actividades.

Hassan et al. (2023) propusieron un enfoque híbrido de aprendizaje automático para la planificación académica en plataformas móviles, integrando regresión regularizada con análisis de comportamiento en tiempo real. Concluyeron que las recomendaciones generadas bajo este enfoque permiten mejorar la priorización diaria y optimizar la gestión del tiempo.

Santos y Ribeiro (2024) evaluaron un sistema inteligente de soporte académico con notificaciones adaptativas y análisis predictivo. Los resultados evidenciaron que la personalización dinámica de alertas incrementa la autorregulación estudiantil y mejora la adherencia a los planes de estudio semanales.

Khan et al. (2023) implementaron un sistema de calendarización académica basado en random forest y algoritmos de optimización evolutivos. Su estudio mostró que los modelos híbridos permiten reducir conflictos de horarios y mejorar la distribución equilibrada de la carga académica.

Watanabe et al. (2024) aplicaron redes neuronales profundas para identificar patrones de procrastinación en estudiantes de primer año. El uso combinado de CNN y LSTM permitió predecir períodos críticos y generar recordatorios preventivos altamente efectivos.

Zhao y Lin (2025) desarrollaron un sistema móvil de seguimiento académico utilizando XGBoost y minería de datos educativa para detectar momentos de baja productividad. Concluyeron que las recomendaciones adaptativas basadas en datos temporales mejoran el cumplimiento de actividades.

Huq et al. (2023) compararon algoritmos supervisados y no supervisados para clasificar estudiantes según hábitos de estudio, demostrando que el clustering previo mejora la precisión de los sistemas de recomendación y la personalización de recordatorios inteligentes.

Nakatani et al. (2024) diseñaron un sistema de planificación académica basado en reinforcement learning que ajusta automáticamente la agenda del estudiante. El modelo mostró eficiencia superior frente a métodos tradicionales al adaptar los bloques de estudio al comportamiento real del usuario.

Rojas y Almeida (2023) estudiaron el uso de asistentes académicos basados en procesamiento del lenguaje natural para organizar tareas y generar alertas personalizadas. Concluyeron que los modelos de lenguaje favorecen la interacción intuitiva y mejoran la adherencia a los planes de estudio.

Finalmente, Gómez-Pinto et al. (2025) integraron LLMs con análisis histórico de rendimiento académico para construir un sistema de planificación universitaria inteligente. Los resultados demostraron que los modelos generativos pueden ofrecer recomendaciones altamente personalizadas, permitiendo mejorar significativamente la gestión del tiempo y la productividad académica.

2.2 Marco Teórico

Para comprender la variable independiente denominada [Plataforma Web inteligente basada en modelos de lenguaje \(LLMs\)](#) para la optimización de la productividad

académica, es necesario contextualizar los avances tecnológicos recientes en inteligencia artificial aplicada al aprendizaje y a la gestión del tiempo.

Los grandes modelos de lenguaje (LLMs), como ChatGPT, Gemini o Claude, representan una nueva generación de sistemas capaces de comprender y generar lenguaje natural con una precisión sin precedentes (Filippi & Motyl, 2024; Mannekote et al., 2024). Su integración en entornos educativos permite ofrecer experiencias personalizadas, ajustadas al contexto académico y al comportamiento del estudiante, favoreciendo la autorregulación y el aprendizaje autónomo (Bhattacharjee et al., 2024; Liu et al., 2025).

En el ámbito de la gestión del tiempo académico, se han desarrollado aplicaciones inteligentes orientadas a mejorar la planificación de actividades, recordatorios y seguimiento del progreso de los estudiantes. Zhao et al. (2024) diseñaron una plataforma de autogestión del tiempo para estudiantes universitarios, concluyendo que el uso de notificaciones adaptativas y recordatorios personalizados contribuye significativamente a reducir la procrastinación y a mejorar el cumplimiento de tareas. Del mismo modo, Nobbe et al. (2024) identificaron que los recordatorios basados en smartphones pueden ser una herramienta de doble filo: promueven la adherencia al estudio, pero si se emiten en momentos inadecuados, pueden causar distracción o sobrecarga cognitiva.

En esta línea, Mumcu y Çebi (2025) demostraron que las notificaciones inteligentes, combinadas con algoritmos de aprendizaje automático, influyen positivamente en la autorregulación y en la gestión emocional del estudiante frente a sus responsabilidades académicas. A partir de estos hallazgos, surge la necesidad de desarrollar aplicaciones que integren mecanismos de predicción y recomendación, capaces de sugerir momentos óptimos para estudiar o realizar tareas según el comportamiento histórico del usuario.

Los LLMs, al combinar capacidades conversacionales y de razonamiento contextual, se convierten en un asistente cognitivo que no solo informa, sino que también orienta y motiva al estudiante. Doğan (2025) propone un marco experimental para la recomendación de cursos utilizando LLMs, demostrando que estos modelos pueden analizar patrones académicos y ofrecer sugerencias personalizadas. De forma complementaria, Li et al. (2025) destacan que los sistemas explicables basados en inteligencia artificial incrementan la confianza del usuario al mostrar de manera transparente las razones detrás de cada recomendación.

Por su parte, Fernex et al. (2025) plantean un marco técnico para el desarrollo de aplicaciones inteligentes de gestión del tiempo, con componentes de recopilación de datos, modelado y retroalimentación personalizada. Dicho enfoque coincide con la lógica de este proyecto, que busca mejorar los procesos de planificación académica mediante la interacción continua entre el estudiante y un sistema inteligente basado en LLMs.

Asimismo, estudios recientes demuestran que el uso de inteligencia artificial puede optimizar procesos de planificación y asignación de recursos educativos. Diallo y Tudose (2024) aplicaron modelos de optimización para programar actividades docentes de manera eficiente, mientras que Bashab et al. (2022) y Gu et al. (2025) sistematizaron metodologías de programación académica que integran restricciones humanas y tecnológicas. Estos avances sustentan la viabilidad de aplicar técnicas de inteligencia artificial no solo a la organización institucional, sino también al ámbito personal del estudiante.

En el campo específico de las aplicaciones móviles educativas, Acuña et al. (2024) desarrollaron *MARA*, una Plataforma Web recordatoria para estudiantes, demostrando que las herramientas portátiles favorecen la adherencia a los planes de estudio y el control del rendimiento académico. Igualmente, Mohamad Zukriyani y Azizan (2023)

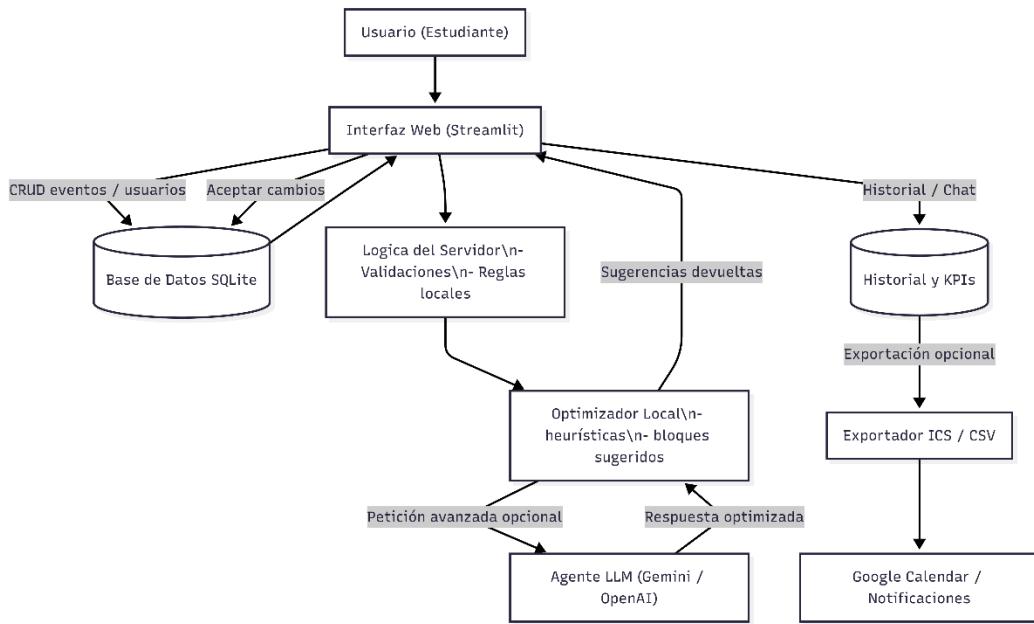
realizaron una revisión sobre los sistemas de planificación académica digital, concluyendo que los entornos móviles facilitan la autogestión del tiempo, especialmente cuando integran retroalimentación automatizada.

El papel de los LLMs también ha sido explorado en sistemas de recomendación y tutoría académica. Elmoghazy et al. (2025) y Munson et al. (2025) sostienen que los LLMs, al combinar capacidades de comprensión semántica y generación contextual, pueden adaptarse a los perfiles de aprendizaje de cada estudiante, proponiendo estrategias de mejora individualizadas. En la misma línea, Pelánek et al. (2024) diseñaron un sistema modular de recomendaciones de actividades de aprendizaje, capaz de ajustar dinámicamente el contenido educativo en función del progreso del usuario.

Por otra parte, Krishnamurthy y Shiva (2025) y Powell y Riccardi (2025) han demostrado que los LLMs pueden aplicarse a la optimización de tareas mediante razonamiento automatizado, generando explicaciones textuales comprensibles para el usuario. Este tipo de retroalimentación resulta especialmente útil en contextos académicos, donde el estudiante requiere no solo saber *qué* hacer, sino también *por qué* debe hacerlo.

Investigaciones como las de Zuo et al. (2025) y Goslen et al. (2024) confirman que el uso de modelos de inteligencia artificial centrados en el comportamiento diario del estudiante permite predecir su rendimiento académico con alta precisión. Estas soluciones promueven la toma de decisiones basadas en datos y la optimización de los hábitos de estudio, objetivos alineados con la presente propuesta de Plataforma Web inteligente.

Figura 3 Flujo de trabajo de la Plataforma Web Inteligente con LLMs



Nota. Elaboración propia basada en el diagrama generado mediante *Mermaid*

Flowchart. La figura representa el flujo de interacción entre el usuario, la interfaz web, los módulos de validación, el optimizador local y el agente LLM utilizado para sugerencias avanzadas.

De igual manera, para detallar la **variable interviniente** conocida como **SCRUM**, se ha tomado en cuenta los conceptos establecidos por Schwaber y Sutherland (2020), creadores de esta metodología.

SCRUM se fundamenta en los principios del empirismo y del pensamiento Lean. El empirismo sostiene que el conocimiento se obtiene a partir de la experiencia y la observación, mientras que el pensamiento Lean busca eliminar desperdicios, centrándose únicamente en lo esencial. SCRUM emplea un enfoque iterativo e incremental para mejorar la previsibilidad y controlar los riesgos durante el desarrollo de un producto.

Esta metodología se basa en la colaboración entre personas que, en conjunto, poseen las habilidades necesarias para cumplir con los objetivos del proyecto. Los equipos

SCRUM aprenden continuamente a través de ciclos de transparencia, inspección y adaptación, los cuales se materializan en los diferentes eventos del marco de trabajo (Schwaber & Sutherland, 2020).

SCRUM es ampliamente utilizada en la actualidad dentro de empresas de tecnología y desarrollo de software debido a su capacidad de organización y mejora continua. Permite dividir las tareas en entregas parciales y controladas llamadas *Sprints*, donde se establecen prioridades y responsabilidades para cada miembro del equipo. Esto garantiza que los entregables se cumplan en los plazos definidos, fomentando la adaptabilidad y la eficiencia en los procesos de desarrollo.

Según Satpathy (2022), en SCRUM existen tres roles fundamentales que garantizan el cumplimiento de los objetivos del proyecto: Product Owner: responsable de maximizar el valor del producto, expresando claramente los requerimientos del usuario o cliente, y asegurando que el desarrollo mantenga su justificación y alineación con los objetivos del proyecto. Scrum Master: actúa como facilitador del proceso, eliminando obstáculos y asegurando que se respeten los principios del marco SCRUM. No ejerce autoridad jerárquica, sino que promueve la autogestión y la mejora continua del equipo. Equipo de Desarrollo (Scrum Team): grupo interdisciplinario encargado de transformar los requerimientos definidos por el Product Owner en entregables funcionales, evaluando, estimando y construyendo las soluciones de forma colaborativa.

Por otro lado, Jocham, Coleman y Sutherland (2025) explican que SCRUM se estructura en eventos acotados en el tiempo, los cuales promueven la inspección, adaptación y entrega continua de valor. Estos eventos se desarrollan dentro del *Sprint*, el cual representa el contenedor principal del trabajo. Los eventos son: El Sprint: periodo fijo de hasta cuatro semanas donde se desarrollan las funcionalidades planificadas. Durante el Sprint se lleva a cabo todo el trabajo necesario para alcanzar

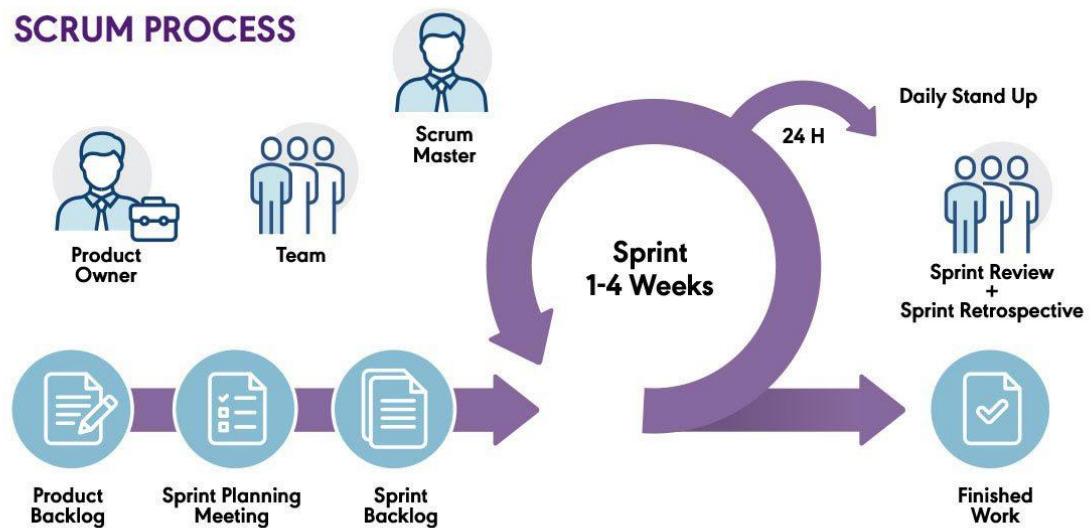
el objetivo del producto. Reunión Diaria (Daily Scrum): espacio breve en el que los miembros del equipo revisan los avances y ajustan el plan de acción hasta la siguiente reunión. Revisión del Sprint (Sprint Review): reunión donde se presentan los resultados obtenidos, se evalúan los avances y se recibe retroalimentación de los interesados. Retrospectiva del Sprint (Sprint Retrospective): momento en el que el equipo analiza su desempeño y propone mejoras en la forma de trabajo para el siguiente ciclo.

En el contexto de esta investigación, la metodología SCRUM permitirá gestionar el desarrollo iterativo e incremental de la Plataforma Web inteligente. Cada *Sprint* se orientará al diseño, entrenamiento y prueba de funcionalidades específicas, tales como la inserción de horarios, el módulo de recomendaciones personalizadas con LLMs, y el sistema de notificaciones inteligentes para la mejora de la productividad académica.

De esta manera, SCRUM se convierte en la guía metodológica del proceso de desarrollo del software, asegurando una entrega continua de valor, adaptabilidad a los cambios y validación constante del avance del sistema con los objetivos académicos propuestos.

A continuación, en la Figura 5, se presenta una representación esquemática del ciclo SCRUM aplicado al desarrollo de la Plataforma Web Inteligente con LLMs.

Figura 4 Ciclo de SCRUM



Nota. Tomado y adaptado de << *What is Scrum? An overview of Scrum and the Agile Journey* >> por PM Partners Services, 2024

La variable dependiente que es **Productividad Académica en los estudiantes de Ingeniería de Software** se refiere al nivel de eficiencia con el que los estudiantes gestionan su tiempo, tareas y objetivos académicos para alcanzar resultados óptimos en su formación profesional. En el contexto actual, numerosos estudiantes de Ingeniería de Software presentan dificultades para mantener niveles adecuados de productividad debido a la falta de organización, la tendencia a posponer actividades y la gestión ineficiente del tiempo. Estas limitaciones generan retrasos en la entrega de trabajos, acumulación de tareas y disminución del rendimiento general. Según Bhattacharjee et al. (2024), la productividad académica está estrechamente relacionada con la capacidad del estudiante para sostener estrategias de autorregulación y evitar la procrastinación académica, factores que influyen directamente en su desempeño.

Asimismo, estudios recientes han identificado que la productividad no depende únicamente del volumen de estudio, sino de la manera en que los estudiantes

estructuran sus actividades y distribuyen su carga académica (Acuña, Sabili & Friginal, 2024). La evidencia muestra que los estudiantes que organizan sus tareas en función de prioridades claras y tiempos realistas presentan mayores niveles de eficiencia y mejores resultados. De igual forma, investigaciones de Fernex et al. (2025) y Elmoghazy et al. (2025) destacan que la adopción de prácticas sistemáticas de planificación mejora significativamente la productividad, al facilitar el seguimiento de actividades, la toma de decisiones académicas y el mantenimiento de un ritmo de estudio constante.

En el caso de los estudiantes de la carrera de Ingeniería de Software de la UNMSM, se observa que los bajos niveles de planificación, la sobrecarga de actividades y la dificultad para priorizar tareas constituyen factores que afectan directamente su productividad académica. Adicionalmente, Zhao et al. (2024) señalan que la productividad se ve influenciada por la capacidad del estudiante para organizar su tiempo de manera autónoma, mientras que Nobbe et al. (2024) advierten que un manejo inadecuado de estrategias de estudio puede generar estrés académico y reducir la eficiencia en el cumplimiento de actividades.

La situación actual de desorganización y baja eficiencia en la gestión de tareas académicas se ilustra en la Figura 6, que representa las dificultades más comunes de los estudiantes en el manejo de su tiempo y cumplimiento de actividades.

Figura 5 Desorganización y Baja Productividad Académica en los Estudiantes de Ingeniería



Nota. Tomado y adaptado de << *La ingeniería más demandada en la UNMSM 2025-I: conoce la carrera con sueldos de más de S/7.000* >> por Facultad de Ingeniería de Sistemas e Informática – UNMSM (2024, 22 de octubre).

El propósito de mejorar la productividad académica en los estudiantes de Ingeniería de Software de la UNMSM es fundamental para fortalecer su rendimiento, organización y bienestar educativo. La implementación de una Plataforma Web inteligente basada en LLMs busca transformar un proceso académico tradicionalmente desorganizado y dependiente de recordatorios manuales en un sistema automatizado, personalizado y disponible 24/7. A través de esta herramienta tecnológica, los estudiantes podrán gestionar sus tareas, recibir recordatorios, recomendaciones adaptativas y priorizar actividades de acuerdo con su carga académica y hábitos de estudio, fomentando una planificación eficiente y una

reducción de la procrastinación.

En el contexto educativo, la optimización del tiempo y la gestión de tareas son elementos esenciales para el éxito académico. Estudios recientes destacan que el uso de aplicaciones móviles inteligentes contribuye a mejorar la organización y la autogestión de los estudiantes al ofrecer interfaces interactivas y notificaciones automáticas que refuerzan la responsabilidad y la constancia en el aprendizaje (Acuña et al., 2024; Fernex et al., 2025). Asimismo, la integración de modelos de lenguaje grandes (LLMs) en el ámbito educativo permite la personalización de estrategias de estudio y recomendaciones académicas, adaptadas al perfil individual de cada estudiante, lo cual mejora la motivación y el rendimiento (Bhattacharjee et al., 2024; Mannekote et al., 2024).

La incorporación de metodologías ágiles como SCRUM en el desarrollo de esta plataforma permite un proceso iterativo y colaborativo, donde las funciones son validadas y mejoradas continuamente según las necesidades del usuario final (Ventura Roque Hernández et al., 2023). De este modo, la solución no solo responde a las limitaciones actuales en la gestión del tiempo, sino que también promueve una cultura de mejora continua y autonomía académica entre los estudiantes.

A continuación, se presenta la Figura 7, que ilustra el *Uso de Aplicaciones Inteligentes para la Gestión Académica y la Optimización del Tiempo*, representando el flujo de interacción entre el estudiante, el modelo de lenguaje y el sistema de planificación inteligente.

Figura 6 Uso de Aplicaciones Inteligentes para la Gestión Académica y la Optimización del Tiempo



Nota. Tomado y adaptado de << *Información Útil para estudiantes de la Universidad Nacional Mayor de San Marcos* >> por Facultad de Ingeniería de Sistemas e Informática - UNMSM, 2024

En esta investigación, se han seleccionado cinco (5) indicadores fundamentales para evaluar el impacto y desempeño de la Plataforma Web Inteligente con LLMs en la productividad académica de los estudiantes de ingeniería de la UNMSM. Estos indicadores son: Tiempo de estudio promedio por día, Tiempo promedio para completar una tarea desde su asignación, Tasa de finalización de tareas a tiempo, Promedio de notas por ciclo y Nivel de satisfacción del estudiante. A continuación, se detallan los conceptos de cada indicador y su valor de referencia. Estos indicadores no solo permiten monitorear el rendimiento académico y la efectividad del sistema, sino que también aseguran que los resultados obtenidos sean cuantificables, consistentes y representativos del impacto real de la herramienta en la organización y desempeño estudiantil.

Tiempo de estudio promedio por día: Este indicador mide la cantidad promedio de

horas diarias que los estudiantes dedican al estudio y la planificación de sus actividades académicas. Permite analizar los hábitos de gestión del tiempo y la capacidad del estudiante para distribuir su carga académica de forma equilibrada. Un aumento sostenido en este indicador sugiere una mejora en la disciplina y la organización personal gracias al uso de la plataforma. Valor de referencia: 4 horas/día

Tiempo promedio para completar una tarea desde su asignación: Evalúa el tiempo que transcurre entre la asignación de una tarea y su finalización. Este indicador refleja la eficiencia en la ejecución de actividades y la capacidad de priorización de los estudiantes. Una reducción en este tiempo sugiere que la plataforma facilita una mejor planificación y evita la acumulación de tareas pendientes. Valor de referencia: 8 horas

Tasa de finalización de tareas a tiempo: Mide el porcentaje de tareas entregadas dentro del plazo establecido. Este indicador es clave para determinar el nivel de compromiso y responsabilidad académica del estudiante. Una mayor tasa de finalización oportuna evidencia el impacto positivo de la plataforma en la gestión del tiempo y la reducción de la procrastinación académica. Valor de referencia: 70 %

Promedio de notas por ciclo: Refleja el desempeño académico general del estudiante en función de sus calificaciones promedio obtenidas en el ciclo académico. Este indicador permite analizar si la mejora en la organización y planificación tiene un efecto directo en los resultados académicos. Valor de referencia: 13.5

Nivel de satisfacción del estudiante: Evalúa la percepción del estudiante sobre la utilidad, facilidad de uso y efectividad de la plataforma. Este indicador mide el grado de aceptación y satisfacción general con la herramienta tecnológica, considerando factores como la experiencia de usuario, la personalización de las recomendaciones y el apoyo en la gestión académica. Valor de referencia: 5 (en una escala del 1 al 5)

CAPÍTULO III: CONSTRUCCIÓN DE LA SOLUCIÓN

Teniendo en cuenta lo planteado anteriormente y basándonos en una metodología y un marco teórico que sirven como punto de partida para comprender el contexto del problema que se busca abordar, se plantea una solución tecnológica viable desarrollada por el propio investigador.

Antes de detallar la construcción, es fundamental recapitular el propósito del proyecto. La solución propuesta, una “Plataforma Web Inteligente con LLMs”, busca optimizar la productividad académica de los estudiantes de ingeniería de la UNMSM, brindando asistencia personalizada, organización automática de tareas, recordatorios inteligentes y recomendaciones para una gestión eficiente del tiempo de estudio. Para gestionar la complejidad del desarrollo y asegurar una entrega de valor incremental, se seleccionó la metodología ágil SCRUM. Este capítulo describe el proceso de construcción de la plataforma, desde la conceptualización de los requisitos hasta la obtención de un producto funcional, siguiendo los eventos y artefactos definidos por el marco SCRUM.

3.1. Creación del Product Backlog

Tal como se explicó en el marco teórico, dentro de la metodología SCRUM, el Product Backlog se define como una lista priorizada, dinámica y emergente de todas las funcionalidades, requerimientos y tareas necesarias para construir el producto final.

En esta investigación, la creación del Product Backlog representa el proceso de traducir los objetivos académicos del proyecto y las necesidades detectadas en los estudiantes en funcionalidades concretas del software, priorizando aquellas que generen mayor impacto en la productividad y el rendimiento académico.

3.1.1. Visión del Producto y Propósito

El primer paso para construir un Product Backlog efectivo es definir una Visión del Producto clara, que oriente cada desarrollo hacia el cumplimiento del objetivo central de la plataforma.

Basándonos en el propósito de esta tesis, se establece la siguiente Visión del Producto:

“Ser una Plataforma Web inteligente de apoyo académico que potencie la productividad de los estudiantes de ingeniería de la UNMSM, utilizando modelos de lenguaje (LLMs) para ofrecer recomendaciones personalizadas, organización automática de tareas, recordatorios y estrategias de optimización del tiempo de estudio. La plataforma debe ser intuitiva, accesible 24/7 y orientada al logro académico continuo.”

Esta visión establece requisitos no negociables, tales como la precisión de las recomendaciones, la personalización del acompañamiento académico y la disponibilidad constante del sistema.

Para alcanzar estos objetivos y mitigar los riesgos asociados al uso de LLMs (como respuestas irrelevantes o desalineadas con el contexto académico), se ha definido una arquitectura de software inteligente basada en modelos de lenguaje y planificación estructurada.

Tabla 9 Stakeholders Identificados

Rol	Encargado
Product Owner	Escuela Profesional de Ingeniería de Sistemas – UNMSM
Scrum Master	Roddy Pérez Acosta

Development Team

Roddy Pérez Acosta

Usuarios Finales

Estudiantes de fisi de la UNMSM

Stakeholders

Docentes tutores, Coordinación académica,
Oficina de Innovación y Tecnología

La solución se desarrollará utilizando una arquitectura basada en LLMs (Large Language Models) integrados a un sistema de gestión inteligente de horarios y tareas, que permite generar recomendaciones y recordatorios personalizados a partir de las rutinas y cargas académicas del estudiante.

Esta elección arquitectónica es deliberada, ya que posibilita construir una herramienta que analiza información académica real, aprende de los patrones de comportamiento de los usuarios y propone estrategias de mejora para el rendimiento y la gestión del tiempo.

De esta manera, la plataforma contribuye directamente al cumplimiento de los objetivos planteados en la investigación, fortaleciendo la autonomía, disciplina y eficiencia de los estudiantes.

3.1.2 Obtención de Requisitos y Perfiles de Usuario

La definición de los requisitos funcionales y no funcionales de la plataforma se realiza mediante un proceso de análisis documental, entrevistas y observación de los hábitos académicos de los estudiantes de ingeniería de la UNMSM.

El objetivo es identificar las necesidades reales y los puntos de dolor que afectan su productividad, con el fin de diseñar una solución alineada a sus dinámicas de estudio.

Para ello, se establecen perfiles de usuario representativos, que permiten comprender los diferentes contextos académicos, tecnológicos y motivacionales dentro del público objetivo.

Tabla 10 Perfiles de Usuario – Estudiantes de Ingeniería de la UNMSM

Persona 1: Carlos Huamán Rojas		
Descripción	Necesidades	Puntos de Dolor Actuales
21 años, estudiante de Ingeniería de Sistemas. Participa en varios cursos teóricos y de programación. Usa el celular como principal herramienta de organización.	<p>Desea una plataforma que le recuerde tareas, prácticas y exámenes según su horario real.</p> <p>Quiere recibir recomendaciones personalizadas para mejorar su gestión del tiempo.</p>	<p>Suele olvidar fechas límite. Dificultad para equilibrar proyectos, prácticas y clases. Falta de una herramienta que centralice sus tareas y recordatorios.</p>

Persona 2: Daniela Flores Vega

Descripción	Necesidades	Puntos de Dolor Actuales
23 años, estudiante de Ingeniería Industrial. Trabaja medio tiempo y estudia en el turno noche.	<p>Necesita una plataforma que reorganice automáticamente su horario cuando surgen imprevistos laborales.</p> <p>Desea visualizar su progreso académico.</p>	Falta de tiempo para planificar sus estudios. Estrés por acumulación de tareas. Baja productividad y dificultades para cumplir con plazos.

Persona 3: José Luis Poma Torres

Descripción	Necesidades	Puntos de Dolor Actuales
25 años, estudiante de Ingeniería Electrónica. Último año de carrera, realiza tesis y prácticas profesionales.	<p>Requiere un asistente que le ayude a priorizar tareas y optimizar su tiempo entre tesis, trabajo y cursos pendientes.</p> <p>Desea monitorear el avance y desempeño de sus tutorados mediante reportes automatizados.</p>	Sobrecarga de tareas. Dificultad para mantener una rutina constante. Falta de retroalimentación sobre hábitos de estudio.

3.1.3 Estructura del Backlog: Épicas e Historias de Usuario

Como primer paso dentro del marco SCRUM, se definió el propósito del proyecto y se desglosó en una lista priorizada de requisitos funcionales, representados mediante épicas e historias de usuario.

En este caso, la Plataforma Web inteligente con LLMs no es una simple agenda o recordatorio, sino una plataforma de asistencia académica personalizada que ayuda a los estudiantes a planificar, organizar y optimizar su tiempo de estudio, promoviendo hábitos productivos y mejorando el rendimiento académico.

Tabla 11 *Épicas del Sistema Inteligente con LLMs*

EPIC	Descripción	Valor de Negocio	Prioridad	Criterios de Aceptación
EPIC-01	Como estudiante, necesito una interfaz móvil intuitiva y amigable para registrar y visualizar mis tareas, horarios y recordatorios sin complicaciones.	Alto	Crítica	<ul style="list-style-type: none">- La interfaz debe ser responsive (móvil y tablet).- Permitir agregar tareas y exámenes fácilmente.- Mostrar alertas visuales y auditivas antes de los plazos.
EPIC-02	Como estudiante, necesito que el asistente inteligente (LLM) comprenda mis solicitudes en lenguaje natural y me recomiende estrategias personalizadas para mejorar mi productividad.	Crítico	Crítica	<ul style="list-style-type: none">- El asistente debe interpretar consultas textuales o por voz.- Debe generar recomendaciones basadas en los patrones de estudio del usuario.

- La precisión de las sugerencias debe ser superior al 85%.

EPIC-03	Como usuario, Alto quiero recibir recordatorios automáticos y notificaciones inteligentes según mis hábitos y prioridades.	Alta	<ul style="list-style-type: none">- Los recordatorios deben adaptarse a la frecuencia de estudio.- El sistema debe ajustar las notificaciones si detecta sobrecarga de tareas.
EPIC-04	Como administrador académico o docente, necesito acceder a reportes estadísticos sobre el progreso y cumplimiento de los estudiantes.	Medio	<ul style="list-style-type: none">- Los reportes deben incluir tasa de cumplimiento, horas de estudio y rendimiento promedio.- Debe garantizar la privacidad de los datos individuales.

3.1.4 Detalle de Historias de Usuario y Criterios de Aceptación

Los criterios de aceptación representan las condiciones específicas que deben cumplirse para considerar una Historia de Usuario completa y funcional.

En esta sección se detallan las historias de usuario derivadas de las épicas definidas en el backlog del sistema, orientadas a garantizar una Plataforma Web inteligente, funcional, centrada en la experiencia del estudiante y potenciada con modelos de lenguaje (LLMs).

Tabla 12 *Historias de Usuario para la Plataforma Web Inteligente con LLMs*

ID	Épica	Historia de Usuario	Criterios de Aceptación
HU-01	EPIC-01	Como estudiante, quiero visualizar una interfaz clara y ordenada para registrar y revisar mis tareas fácilmente.	<ol style="list-style-type: none">1. Dado que ingreso a la plataforma, cuando se carga la pantalla principal, entonces veo mis tareas, horarios y recordatorios organizados por fecha.2. Dado que no tengo tareas registradas, cuando abro la plataforma web por primera vez, entonces aparece un mensaje motivacional del asistente.3. Dado que utilizo un dispositivo móvil o tablet, cuando accedo a la plataforma web, entonces la interfaz se adapta correctamente al tamaño de pantalla.
HU-02	EPIC-01	Como estudiante, quiero agregar nuevas tareas o exámenes de manera sencilla para mantener mi planificación actualizada.	<ol style="list-style-type: none">1. Dado que estoy en la plataforma, cuando selecciono “Agregar tarea”, entonces puedo ingresar nombre, fecha límite, curso y prioridad.2. Dado que completo los campos requeridos, cuando guardo la tarea, entonces se registra exitosamente y aparece en mi lista.3. Dado que modiflico una tarea existente, cuando guardo los cambios, entonces se actualiza automáticamente.

HU-03	EPIC-01	Como estudiante, quiero recibir recordatorios visuales y sonoros antes de la entrega de tareas o exámenes.	<ol style="list-style-type: none"> Dado que registro una tarea con fecha límite, cuando el plazo se aproxima, entonces recibo una notificación visual y sonora. Dado que abro la notificación, cuando accedo, entonces se me redirige directamente al detalle de la tarea. Dado que completo la tarea, cuando la marco como realizada, entonces cesan los recordatorios asociados.
HU-04	EPIC-01	Como estudiante, quiero visualizar mi progreso académico para evaluar mi desempeño en cada ciclo.	<ol style="list-style-type: none"> Dado que completo varias tareas, cuando accedo a la sección "Mi progreso", entonces observo estadísticas de cumplimiento y promedio general. Dado que tengo bajo rendimiento, cuando el sistema detecta tendencia negativa, entonces me muestra una alerta o sugerencia.
HU-05	EPIC-02	Como estudiante, quiero escribirle al asistente inteligente en lenguaje natural para recibir consejos personalizados sobre mi organización académica.	<ol style="list-style-type: none"> Dado que ingreso una consulta escrita, cuando la envío, entonces el LLM interpreta la pregunta y responde con una recomendación clara. Dado que la respuesta llega, cuando la visualizo, entonces se muestra en un formato comprensible y amigable. Dado que la plataforma web detecta una duda repetida, cuando ocurre, entonces sugiere respuestas previas similares.
HU-06	EPIC-02	Como estudiante, quiero que el asistente me recomiende estrategias personalizadas según mis hábitos de estudio.	<ol style="list-style-type: none"> Dado que registro mis tiempos de estudio, cuando el sistema analiza mis patrones, entonces genera sugerencias adaptadas (por ejemplo: "distribuye tus horas de estudio en bloques de 50 min"). Dado que sigo las recomendaciones, cuando marco progreso, entonces el asistente ajusta sus futuras sugerencias.

HU-07	EPIC-02	<p>Como sistema, quiero interpretar consultas textuales o por voz mediante embeddings para ofrecer respuestas precisas.</p>	<ol style="list-style-type: none"> 1. Dado que recibo una consulta en texto o voz, cuando la proceso, entonces genero su vector semántico para análisis. 2. Dado que realizo búsqueda de contexto, cuando encuentro información relevante, entonces la utilizo para responder con precisión. 3. Dado que no hay información suficiente, cuando ocurre, entonces informo al usuario de manera clara.
HU-08	EPIC-02	<p>Como sistema, quiero evaluar la precisión de las respuestas generadas por el LLM para asegurar recomendaciones confiables.</p>	<ol style="list-style-type: none"> 1. Dado que el modelo genera una respuesta, cuando se evalúa internamente, entonces su coherencia y relevancia deben superar el 85%. 2. Dado que la respuesta no cumple el umbral de calidad, cuando ocurre, entonces el sistema reformula o solicita nueva generación. 3. Dado que hay errores de conexión o procesamiento, cuando suceden, entonces se muestra un mensaje de error amigable.

HU-09	EPIC-03	Como estudiante, quiero recibir recordatorios automáticos de mis tareas y exámenes para no olvidar fechas importantes.	1. Dado que registro una tarea o examen con fecha límite, cuando se acerca el plazo, entonces recibo una notificación automática. 2. Dado que el evento está completado, cuando lo marco como "hecho", entonces cesan los recordatorios asociados. 3. Dado que las notificaciones están activas, cuando configuro la plataforma web, entonces puedo definir si quiero recordatorios diarios, semanales o personalizados.
HU-10	EPIC-03	Como estudiante, quiero que el sistema ajuste mis recordatorios en función de mis hábitos y carga de tareas para mejorar mi organización.	1. Dado que el sistema detecta múltiples tareas en un mismo día, cuando ocurre, entonces sugiere redistribuirlas. 2. Dado que estudio a determinadas horas, cuando se detecta ese patrón, entonces las notificaciones se adaptan a mis horarios más productivos. 3. Dado que ignoro repetidamente recordatorios, cuando el sistema lo nota, entonces reduce su frecuencia o sugiere otra configuración.
HU-11	EPIC-03	Como sistema, quiero analizar el historial de cumplimiento del usuario para ajustar la frecuencia de notificaciones automáticamente.	1. Dado que tengo acceso al historial de tareas, cuando analizo su cumplimiento, entonces calculo una tasa de constancia. 2. Dado que el usuario tiene baja constancia, cuando ocurre, entonces incremento la frecuencia de alertas. 3. Dado que el usuario tiene alto cumplimiento, cuando ocurre, entonces reduzco la cantidad de recordatorios.
HU-12	EPIC-04	Como administrador académico, quiero visualizar reportes del progreso de los estudiantes para identificar su nivel de cumplimiento.	1. Dado que los estudiantes registran tareas, cuando accedo al panel administrativo, entonces veo indicadores de avance y cumplimiento. 2. Dado que selecciono un periodo, cuando aplico filtros, entonces el reporte se actualiza en tiempo real. 3. Dado que visualizo el reporte, cuando lo exporto, entonces puedo obtenerlo en formato PDF o CSV.

HU-13	EPIC-04	Como docente, quiero acceder a estadísticas de horas de estudio promedio de mis estudiantes para evaluar su desempeño académico.	1. Dado que los estudiantes registran su tiempo de estudio, cuando accedo a estadísticas, entonces veo el promedio semanal o mensual. 2. Dado que algunos estudiantes superan o no alcanzan el promedio, cuando se muestra el reporte, entonces se resaltan en color diferenciado. 3. Dado que descargo el informe, cuando lo genero, entonces mantiene la privacidad de datos individuales.
HU-14	EPIC-04	Como sistema, quiero garantizar la privacidad y seguridad de la información académica al generar reportes o compartir datos.	1. Dado que se generan reportes, cuando se exportan, entonces omiten nombres personales y datos sensibles. 2. Dado que un usuario intenta acceder sin permisos, cuando ocurre, entonces se bloquea el acceso y se muestra un mensaje de restricción. 3. Dado que los reportes se almacenan en la base de datos, cuando se guardan, entonces se cifran con protocolos de seguridad establecidos.

3.2 Planificación del Sprint

La planificación del sprint constituye una etapa fundamental dentro de la metodología Scrum, ya que permite organizar el trabajo de manera estructurada, priorizando las tareas más relevantes según los objetivos del producto.

Durante esta fase, el equipo define el alcance del sprint, identifica las funcionalidades que serán desarrolladas y estima el tiempo necesario para completarlas, considerando la complejidad, prioridad y dependencia entre tareas.

En el caso del desarrollo de la Plataforma Web inteligente con LLMs para asistencia académica personalizada, la planificación del sprint se orientó a asegurar que cada iteración represente un avance funcional y medible en la construcción del sistema. De este modo, se busca que el producto evolucione de forma incremental e iterativa, garantizando la calidad, estabilidad y cumplimiento de los objetivos definidos para mejorar la experiencia y el rendimiento académico del estudiante.

3.2.1 Duración del Sprint

Una vez identificados los requerimientos principales del sistema, se seleccionaron aquellos más críticos y de alto valor para el usuario final, con el objetivo de construir una primera versión funcional de la plataforma.

El propósito del primer sprint no fue desarrollar todas las funcionalidades completas, sino establecer la base operativa del sistema, enfocándose en la creación de una interfaz móvil funcional, la gestión de tareas académicas y la integración inicial del modelo de lenguaje (LLM) para generar recomendaciones básicas.

Se definió una duración de cuatro (4) semanas para este primer sprint, periodo suficiente para abordar el diseño, desarrollo, pruebas y validación de los módulos esenciales:

- Registro y visualización de tareas y recordatorios.
- Implementación de la base de datos local.
- Integración inicial del LLM.
- Pruebas de usabilidad y ajustes funcionales.

Al culminar este sprint, la plataforma debía ser capaz de registrar, mostrar y gestionar tareas académicas, así como procesar solicitudes simples en lenguaje natural para ofrecer recomendaciones iniciales al estudiante.

3.2.2 El Plan de Trabajo (Sprint Backlog)

Para la elaboración del Sprint Backlog, se tomó como referencia el análisis de los requerimientos y épicas definidos en las secciones anteriores.

Se seleccionaron las funcionalidades de mayor prioridad, orientadas a establecer una base sólida para el crecimiento progresivo de la Plataforma Web inteligente.

La estrategia de desarrollo se centró en implementar los módulos fundamentales que permitan una interacción fluida entre el estudiante y el asistente académico, garantizando la coherencia entre la interfaz, la base de datos y el modelo de lenguaje (LLM).

El Sprint 1 tuvo como objetivo dejar operativa una versión inicial del sistema con las siguientes metas:

- Diseñar e implementar la interfaz móvil de gestión de tareas y horarios.
- Desarrollar el módulo de recordatorios automáticos.
- Configurar la comunicación básica con el modelo de lenguaje (LLM) para generar respuestas personalizadas.
- Realizar pruebas de funcionalidad, rendimiento y usabilidad.

Al concluir este sprint, la plataforma debía contar con una arquitectura funcional mínima viable (MVP) capaz de registrar tareas, enviar notificaciones automáticas y procesar consultas en lenguaje natural, sirviendo como base para los futuros sprints orientados a la optimización del asistente inteligente, la personalización de estrategias de estudio y la generación de reportes académicos.

Tabla 13 Cronograma de actividades del Sprint 1

ID	Tarea	Prioridad	Plazo	Estado
Semana 1: Diseño e Interfaz Inicial				
HU-01	Diseño de la interfaz principal: desarrollar la pantalla inicial que muestre las tareas, horarios y recordatorios del estudiante.	Crítica	3 días	Realizado
HU-02	Implementación del módulo “Agregar Tarea”: permitir el registro de nuevas tareas y exámenes con nombre, curso, fecha y prioridad.	Crítica	3 días	Realizado
HU-03	Diseño del sistema de notificaciones visuales y sonoras: establecer alertas previas a las fechas límite de tareas o exámenes.	Alta	2 días	Realizado
Semana 2: Gestión de Datos y Funcionalidades Básicas				
HU-04	Implementar módulo de progreso académico: generar estadísticas simples del cumplimiento de tareas.	Alta	3 días	Realizado
TSK-01	Configuración de base de datos local (SQLite o Firebase) para almacenar tareas, recordatorios y horarios.	Crítica	2 días	Realizado
TSK-02	Pruebas de registro, modificación y eliminación de tareas en la base de datos.	Alta	2 días	Realizado

Semana 3: Integración del Asistente Inteligente (LLM)

HU-05	Integración inicial del LLM: permitir que el usuario envíe consultas en texto al asistente y reciba respuestas personalizadas.	Crítica	3 días	Realizado
HU-06	Implementar lógica de recomendaciones personalizadas según los hábitos de estudio registrados.	Alta	3 días	Realizado

Semana 4: Pruebas y Ajustes Finales del MVP

HU-07	Procesamiento básico de lenguaje natural: convertir las consultas del usuario en vectores semánticos para generar respuestas coherentes.	Alta	2 días	Realizado
TSK-03	Pruebas funcionales del flujo completo: registro de tareas, notificaciones y consulta con el asistente.	Crítica	2 días	Realizado
TSK-04	Ajustes visuales y optimización de rendimiento en la interfaz móvil.	Media	2 días	Realizado

Tabla 14 Cronograma de actividades del Sprint 2

ID	Tarea	Prioridad	Plazo	Estado
Semana 1: Personalización de Recordatorios y Ajustes Inteligentes				
HU-09	Recordatorios automáticos: implementar el sistema que notifica al estudiante sobre próximas tareas y exámenes.	Crítica	3 días	Realizado
HU-10	Ajuste inteligente de notificaciones: desarrollar la lógica que adapta los recordatorios según la carga académica del estudiante.	Crítica	3 días	Realizado
TSK-01	Configuración del módulo de horarios: integrar una función que sincronice recordatorios con los horarios registrados del usuario.	Alta	2 días	Realizado
Semana 2: Análisis de Comportamiento y Optimización del Asistente				
HU-11	Análisis del historial de cumplimiento: crear el algoritmo que evalúe la constancia del estudiante para ajustar la frecuencia de alertas.	Crítica	3 días	Realizado
HU-06	Recomendaciones personalizadas: refinar la generación de sugerencias del asistente según los hábitos de estudio detectados.	Alta	2 días	Realizado
TSK-02	Entrenamiento de patrones de uso: registrar los datos de interacción del estudiante con el	Media	2 días	Realizado

asistente para optimizar futuras respuestas.

Semana 3: Interfaz Dinámica y Retroalimentación Visual

HU-03	Mejorar notificaciones visuales y sonoras: optimizar el diseño de las alertas para mayor visibilidad y respuesta del usuario.	Alta	2 días	Realizado
HU-04	Retroalimentación de progreso académico: implementar gráficas interactivas que muestren evolución en el rendimiento del estudiante.	Alta	3 días	Realizado

Semana 4: Pruebas, Ajustes y Validación

TSK-03	Pruebas de integración del asistente y notificaciones: verificar el correcto funcionamiento entre los módulos de recordatorios y el LLM.	Crítica	2 días	Realizado
TSK-04	Validación con usuarios reales: aplicar pruebas con estudiantes para obtener retroalimentación sobre la precisión de las alertas y la utilidad de las recomendaciones.	Alta	3 días	Realizado

Tabla 15 Cronograma de actividades del Sprint 3

ID	Tarea	Prioridad	Plazo	Estado
Semana 1: Integración del Módulo Administrativo				
HU-12	Panel de Reportes: Desarrollar la interfaz donde los administradores académicos puedan visualizar el progreso de los estudiantes.	Crítica	3 días	Realizado
TSK-05	Conexión con Base de Datos: Implementar la integración entre el backend y el módulo de reportes para obtener los indicadores de desempeño.	Crítica	2 días	Realizado
Semana 2: Estadísticas y Análisis Académico				
HU-13	Reporte de Horas de Estudio: Implementar la lógica que calcule y muestre promedios de horas de estudio por estudiante o por grupo.	Crítica	3 días	Realizado
TSK-06	Visualización Gráfica: Crear componentes visuales (barras, líneas o circulares) que representen el rendimiento académico de los estudiantes.	Alta	2 días	Realizado
Semana 3: Seguridad y Privacidad de Datos				
HU-14	Protección de Datos: Implementar mecanismos de privacidad en los reportes, evitando mostrar información personal identificable.	Crítica	2 días	Por Hacer

TSK-07	Control de Acceso: Establecer roles y permisos para que solo docentes o administradores autorizados accedan a la información académica.	Alta	3 días	Por Hacer
--------	---	------	--------	-----------

Semana 4: Validación, Pruebas y Despliegue Final

TSK-08	Pruebas de Integración y Seguridad: Realizar pruebas completas del módulo de reportes y verificar el cifrado de datos almacenados.	Alta	2 días	Por Hacer
TSK-09	Documentación Final del Módulo: Elaborar el manual de uso y mantenimiento del panel administrativo, con pasos de exportación de reportes.	Media	3 días	Por Hacer

3.3 Ejecución del Sprint

Para el seguimiento del progreso durante la ejecución de los sprints, se utilizó un tablero Kanban digital en la plataforma Trello, el cual permitió al equipo visualizar de manera clara el flujo de trabajo y el estado de cada tarea.

Descripción de las columnas del tablero en Trello:

1. Product Backlog: Contiene el listado completo de requerimientos del sistema y las historias de usuario priorizadas.
2. Sprint Backlog: Incluye las tareas seleccionadas para el sprint actual según su prioridad.
3. Por Hacer (To Do): Tareas del sprint que aún no se han iniciado.
4. En Progreso (In Progress): Tareas que el equipo está desarrollando activamente.

- Hecho (Done): Tareas completadas que cumplen con los criterios de aceptación definidos.

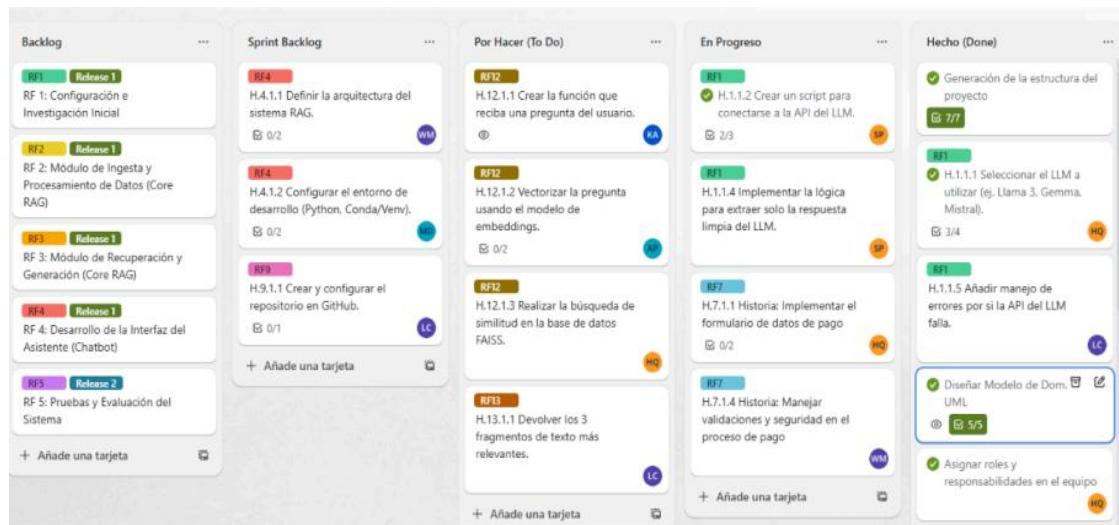
3.3.1 Daily Scrum

Durante las jornadas de desarrollo (martes, miércoles y jueves), el equipo realizó reuniones diarias tipo *Daily Scrum* de aproximadamente 15 minutos para sincronizar actividades y detectar impedimentos. En cada sesión, cada miembro respondió a las tres preguntas estándar orientadas al cumplimiento del **Sprint Goal**:

- ¿Qué hice ayer que contribuyó al objetivo del sprint?
- ¿Qué haré hoy para acercarme al objetivo del sprint?
- ¿Existe algún impedimento que dificulte mi trabajo o el del equipo?

El tablero Kanban (Figura 8) se actualizaba en tiempo real durante estas reuniones para reflejar el estado actual de las tareas y facilitar la toma de decisiones inmediatas.

Figura 7 Tablero Kanban en la plataforma Trello utilizado para la gestión del proyecto



3.3.2 Desarrollo de los entregables claves

Esta sección describe la implementación técnica de los principales incrementos desarrollados durante los Sprints 1, 2 y 3. El desarrollo se efectuó siguiendo la

Arquitectura Limpia y respondiendo a las historias de usuario priorizadas en el Product Backlog (Tabla 13) y el plan de Sprints (Tablas 14, 15 y 16), lo que permitió construir un sistema modular, escalable e integrado con modelos de lenguaje (LLMs).

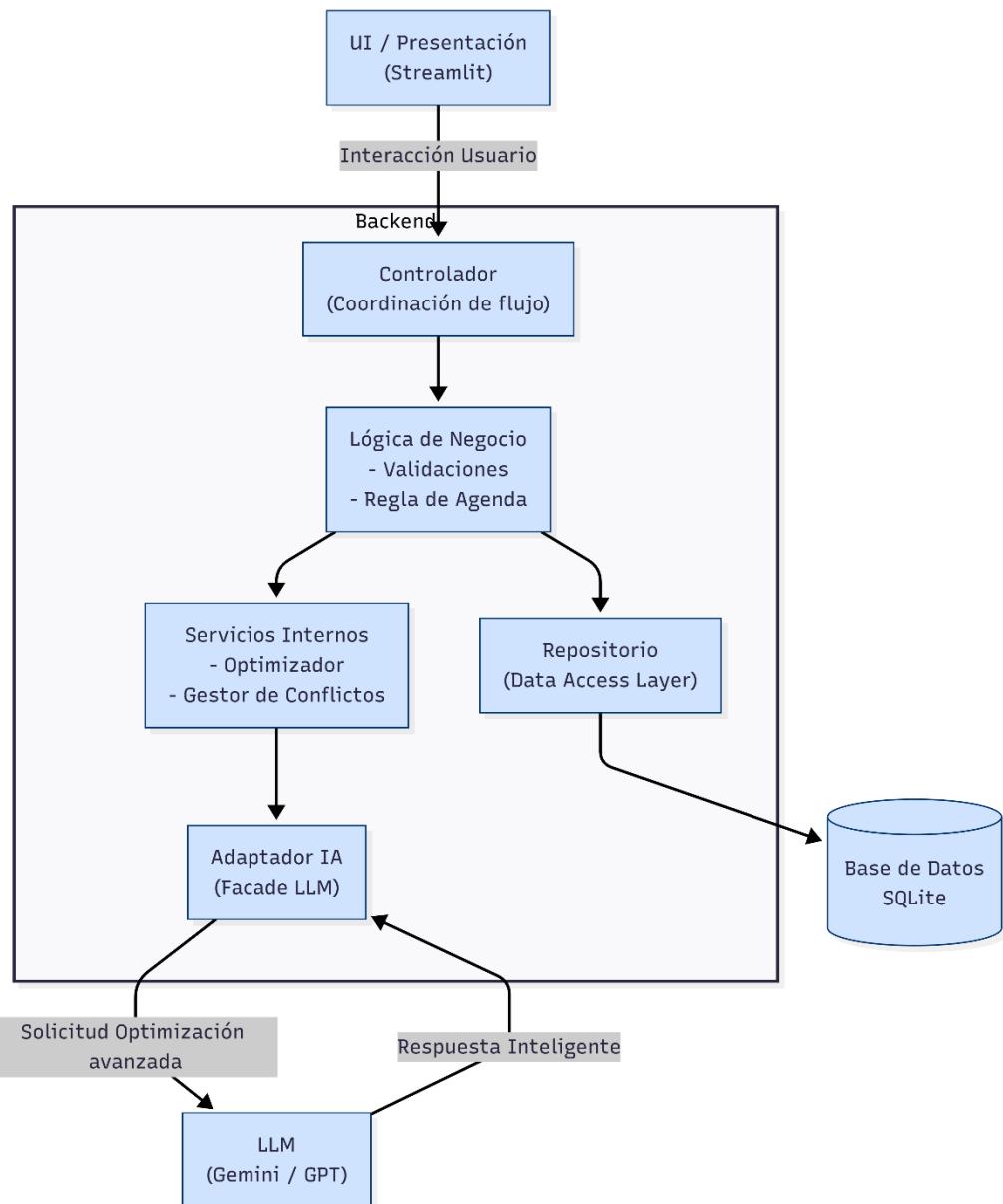
Cada sprint aportó un avance incremental sobre la Plataforma Web, permitiendo obtener un producto funcional, validado con usuarios y preparado para futuras ampliaciones.

3.3.2.2 Arquitectura Lógica y Patrón de Diseño

La plataforma web inteligente desarrollada se organiza siguiendo una arquitectura lógica modular, diseñada para garantizar escalabilidad, mantenibilidad y una integración fluida con modelos de lenguaje (LLM). Esta arquitectura permite separar de forma clara las capas de presentación, lógica de negocio, persistencia de datos y servicios inteligentes, siguiendo los principios de la ingeniería de software moderna.

La Figura 9 ilustra el patrón de diseño de Arquitectura que se adoptó para la construcción del backend. El objetivo fundamental de este patrón es la inversión de dependencias, lo que garantiza una estricta separación de responsabilidades y un bajo acoplamiento entre los componentes del sistema.

Figura 8 Patrón de Arquitectura de la Solución Backend



a) Capa de Presentación (Front-End Web – Streamlit)

Esta capa corresponde al módulo visible para el usuario final.

Se implementa mediante Streamlit, proporcionando una interfaz intuitiva y reactiva donde el estudiante puede:

- Registrar, editar y visualizar eventos en un calendario.
- Consultar recomendaciones generadas por el sistema inteligente.
- Navegar entre la vista diaria, semanal y mensual.

- Interactuar con el agente conversacional para solicitar ayuda o explicaciones.

La interfaz solo se encarga de capturar datos y mostrar resultados, manteniendo la separación estricta entre vista y lógica.

b) Capa de Lógica de Negocio (Backend Interno)

Corresponde al núcleo funcional de la aplicación.

Incluye:

- Validación de datos ingresados por el usuario.
- Gestión del flujo CRUD para eventos, usuarios y preferencias.
- Reglas heurísticas de organización (detectar solapamientos, bloques sugeridos, patrones horarios).
- Orquestación entre la base de datos y los módulos inteligentes.

Esta capa implementa el comportamiento de la aplicación sin depender de la interfaz o del modelo de IA, garantizando independencia y mantenibilidad.

c) Capa de Persistencia (Base de Datos SQLite)

La plataforma utiliza SQLite por su simplicidad, ligereza y compatibilidad directa con entornos web.

La base de datos almacena:

- Información de usuarios.
- Eventos (tareas, clases, descansos, tiempo libre, transporte, etc.).
- Historial de interacciones del agente.
- Estadísticas y preferencias del usuario.

La capa de persistencia se abstrae mediante un módulo dedicado (db.py), siguiendo principios de encapsulamiento y evitando el acceso directo desde la interfaz.

d) Capa de Inteligencia Artificial (Agente LLM)

Este módulo integra el sistema con un modelo de lenguaje grande (LLM) como Gemini o GPT, responsable de:

- Generar recomendaciones personalizadas.
- Detectar patrones en la agenda del estudiante.
- Optimizar horarios de estudio, descanso y actividades.
- Proponer reorganizaciones semanales o mensuales.
- Explicar decisiones al usuario (IA explicable).

El backend envía al LLM únicamente la información necesaria, siguiendo buenas prácticas de seguridad y privacidad.

e) Capa de Optimización Local (Motor Heurístico)

Antes de recurrir al LLM, el sistema emplea un motor propio de optimización:

- Algoritmos de priorización.
- Detección inmediata de conflictos.
- Ajuste rápido de horarios repetitivos.

Esto permite reducir costos de consulta al modelo y acelerar la experiencia del usuario.

3.3.2.2 Sprint 1: Desarrollo del Backend del Asistente Académico

Durante el Sprint 1 se implementó la primera versión funcional del backend del Asistente Académico Inteligente, el cual constituye el núcleo de procesamiento, validación, almacenamiento y respuesta del sistema. Este módulo permitió consolidar la estructura necesaria para manejar información académica, generar recomendaciones personalizadas y garantizar un flujo de comunicación eficiente entre la interfaz de usuario y los componentes avanzados basados en modelos de

lenguaje. A continuación, se detallan los dos flujos principales desarrollados: el Flujo de Gestión de Datos Académicos y el Flujo de Consulta y Recomendación en Tiempo Real.

3.3.2.2.1. Flujo de Gestión de Datos Académicos

Este flujo corresponde a los procesos que permiten almacenar, actualizar y estructurar la información académica del estudiante. Toda esta información será utilizada posteriormente por el módulo inteligente para generar recomendaciones.

- Paso 1: Registro de Información Académica**

El proceso inicia cuando el usuario ingresa sus datos a través de la interfaz web de Streamlit. Esta información incluye: cursos activos, créditos y carga académica, horarios declarados, actividades recurrentes (tareas, prácticas, proyectos, tesis), preferencias personales (rutinas, horas de estudio, descansos).

La UI envía estos datos al backend mediante un controlador que abstrae la comunicación y evita acoplamiento directo con la base de datos.

- Paso 2: Procesamiento Interno**

Después de recibir la información desde la interfaz, el backend ejecuta un proceso interno dividido en dos componentes principales: **validación de datos** y **clasificación inicial de actividades**.

- a. Validación de Datos**

El backend verifica que toda la información cumpla con las reglas definidas por el sistema. Entre las validaciones realizadas se incluyen:

- Formatos correctos de fecha y hora.
- Valores permitidos para categorías y niveles de prioridad.

- Verificación de campos obligatorios.
- Comprobación de coherencia entre actividades y horarios existentes.
- Chequeo de solapamiento entre eventos (evitar conflictos en la agenda).

En caso de errores o inconsistencias, el backend retorna mensajes detallados al frontend, donde el usuario puede corregir los valores ingresados.

b. Clasificación Inicial de Actividades

Una vez validados los datos, el módulo interno denominado “ActivityClassifier” organiza y clasifica automáticamente las actividades académicas del usuario. Este módulo aplica reglas heurísticas iniciales, tales como:

- Si la fecha del evento está próxima → se clasifica como urgente.
- Si la actividad es académicamente relevante (evaluaciones, proyectos) → se asigna alta prioridad.
- Si falta completar información esencial (hora, duración, categoría) → se solicita completar los datos.

Este proceso permite mejorar la organización del estudiante desde el primer momento y prepara las condiciones para que el optimizador inteligente y el LLM puedan generar recomendaciones más precisas.

• Paso 3: Almacenamiento en Base de Datos

Todas las actividades procesadas y validadas se almacenan en la base de datos SQLite del sistema mediante un módulo de persistencia que gestiona las tablas: USUARIOS, ACTIVIDADES, PREFERENCIAS, METRICAS e HISTORIAL_CHAT.

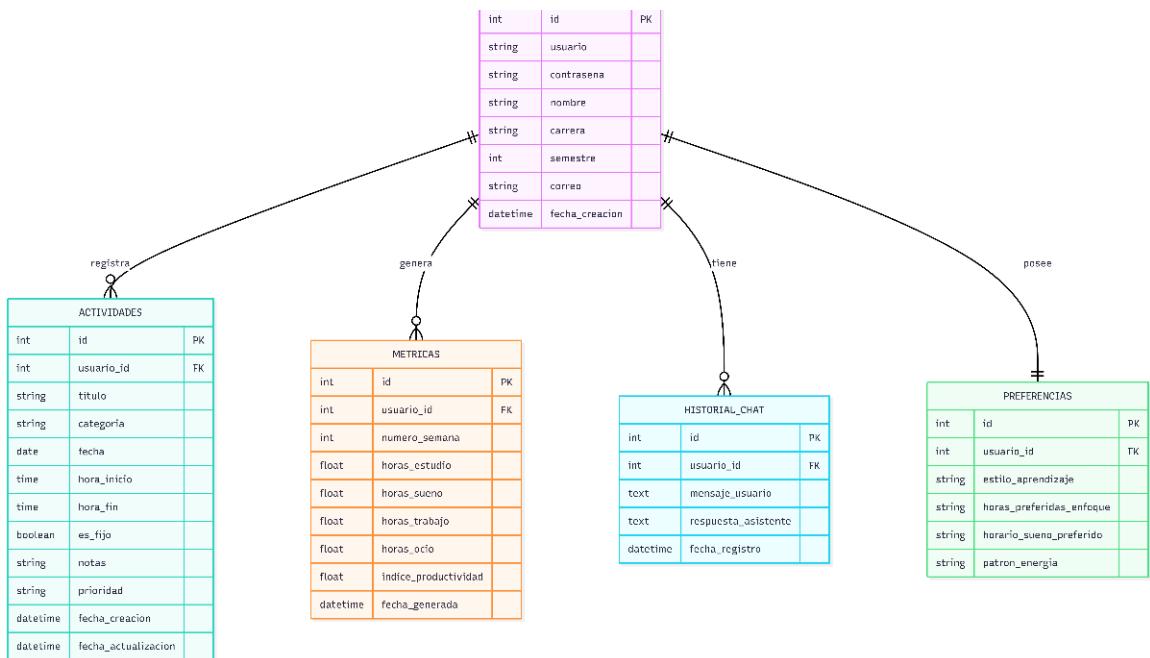
El almacenamiento confiable de la información permite que la plataforma ofrezca:

- Seguimiento continuo del progreso académico del estudiante.

- Historial de actividades, útil para análisis posteriores.
- Generación de métricas automáticas, como horas de estudio, cargas académicas y patrones de productividad.
- Recomendaciones inteligentes, basadas en información histórica y datos contextuales reales.

Esta estructura asegura que los datos se mantengan organizados y accesibles, convirtiéndose en la base para el funcionamiento del asistente académico inteligente.

Figura 9 Diagrama ER – Tablas y campos

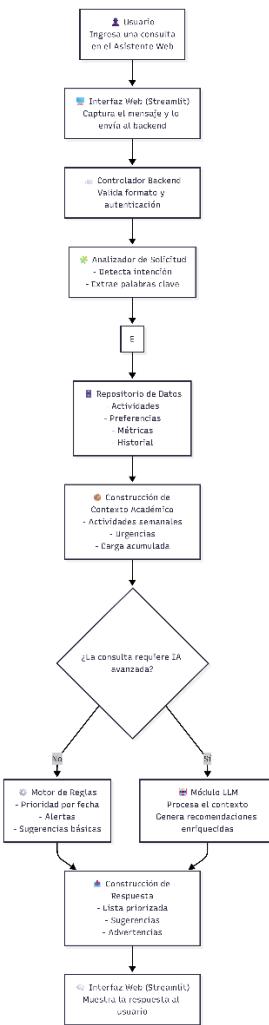


3.3.2.2.2. Flujo de Consulta y Recomendación en Tiempo Real

El flujo de consulta y recomendación en tiempo real se activa cada vez que el estudiante interactúa con el Asistente Inteligente desde la plataforma web. Este proceso describe la secuencia completa de comunicación entre la interfaz, el backend, el motor de recomendaciones y, cuando es necesario, el modelo de lenguaje (LLM).

En conjunto, estos componentes permiten generar sugerencias académicas personalizadas en función del estado actual del usuario, su carga de actividades y su productividad reciente.

Figura 10 Flujo de Consulta y Recomendación en Tiempo Real



Descripción del Proceso del Asistente Académico:

a. Inicio y Recepción de la Solicitud

El proceso comienza cuando el usuario ingresa una pregunta o solicitud a través del módulo de chat de la plataforma web. Desde esta interfaz, el estudiante puede realizar consultas como:

- “¿Qué actividades debería priorizar hoy?”

- “¿Cómo puedo organizar mi semana de estudio?”
- “¿Tengo actividades urgentes o próximas a vencer?”

Una vez enviada la consulta, la interfaz transmite el mensaje al backend mediante una llamada controlada por el módulo interno del sistema. Este mensaje se envía acompañado del identificador del usuario para permitir una respuesta personalizada. El controlador recibe la solicitud, valida su formato y delega el procesamiento al módulo encargado del razonamiento académico.

b. Análisis de la Solicitud

El primer paso del backend consiste en analizar el contenido del mensaje recibido. El sistema identifica la intención del usuario mediante un analizador basado en reglas heurísticas. Este módulo examina las palabras clave, la estructura de la frase y el tipo de recomendación solicitada.

Ejemplos de interpretaciones del analizador:

- Si el mensaje contiene términos como “priorizar”, “ordenar” o “importante”, se asume que el estudiante busca sugerencias de prioridad.
- Si el mensaje menciona “semana”, “organizar mi tiempo” o “planificación”, el sistema activa el módulo de planificación semanal.
- Si se detectan frases como “urgente”, “pronto” o “vencerá”, el sistema habilita un filtrado por proximidad temporal.

Este análisis inicial permite dirigir el flujo hacia la lógica correspondiente sin necesidad de activar todavía el LLM.

c. Recuperación de Información del Estudiante

Una vez determinada la intención, el backend solicita los datos del usuario almacenados en la base de datos. Para ello, accede a los repositorios del sistema

que encapsulan las operaciones sobre las tablas: actividades, usuarios, preferencias, métricas y registro histórico. Entre los datos recuperados se incluyen:

- Actividades pendientes del estudiante.
- Fechas límites y horarios correspondientes.
- Categorías y niveles de prioridad registrados.
- Rutinas habituales basadas en preferencias.
- Métricas recientes como horas de estudio o carga acumulada.
- Registros del historial de interacciones previas.

Con esta información, el sistema puede contextualizar adecuadamente la consulta.

d. Construcción del Contexto Académico

El backend integra toda la información recuperada en un objeto de contexto interno que resume el estado académico del estudiante. Este contexto incluye:

- La lista completa de actividades de la semana.
- Actividades urgentes o próximas a vencer.
- Horas de estudio ya realizadas y horas estimadas necesarias.
- Carga total acumulada por categoría (académica, personal, salud, etc.).
- Detección de períodos saturados o con demasiados eventos.
- Reglas personales tomadas de la tabla de preferencias del usuario.

Este contexto constituye la base para que el sistema genere recomendaciones fundamentadas y adaptadas a la situación real del estudiante.

e. Generación de la Recomendación

En esta fase, el sistema produce la respuesta personalizada. Para el primer prototipo, la plataforma utiliza un motor de reglas heurísticas, sin activar aún el LLM de manera obligatoria. Entre las reglas aplicadas se encuentran:

- Ordenar actividades por fecha de vencimiento.
- Alertar si existen más de tres actividades urgentes, recomendando reorganización inmediata.
- Detectar tareas vencidas, sugiriendo un plan de recuperación.
- Evaluar la carga semanal y proponer una distribución equilibrada de estudio.
- Recomendar descansos cuando la carga excede límites establecidos.

Solo cuando la solicitud del usuario requiere un análisis más profundo —por ejemplo, generar un plan semanal completo o redactar una explicación detallada— el sistema envía el contexto al módulo LLM para generar una respuesta avanzada.

f. Respuesta al Usuario

Finalmente, el backend estructura la respuesta y la devuelve en un objeto con la siguiente información:

- Lista priorizada de actividades.
- Sugerencias personalizadas basadas en el perfil del estudiante.
- Advertencias sobre tareas próximas o vencidas.
- Recomendaciones de productividad y organización.
- En caso de usar LMM, un mensaje extendido elaborado de manera natural.

La plataforma web presenta esta información directamente en la interfaz conversacional del asistente, permitiendo que el estudiante tome decisiones informadas en tiempo real.

3.3.2.3 Sprint 2: Construcción de la “Cara” (Frontend)

El Sprint 2 se orientó a la construcción completa de la **interfaz de usuario web** del sistema, implementada sobre la plataforma **Streamlit**, cumpliendo con el EPIC-01:

Interfaz Intuitiva y Experiencia Conversacional definido en el Plan de Sprints. El objetivo principal fue proporcionar un entorno accesible, ágil y centrado en el estudiante, desde el cual pueda registrar sus actividades, consultar al asistente inteligente, recibir recomendaciones personalizadas en tiempo real y visualizar su progreso académico de manera clara.

Figura 11 Pantalla principal de interacción del asistente académico, donde se integran los módulos de conversación, registro de actividades



La pantalla principal de interacción del asistente académico, donde se integran los módulos de conversación, registro de actividades y visualización de la agenda. El desarrollo se organizó siguiendo una arquitectura modular, donde la interfaz está compuesta por tres elementos fundamentales:

1. Punto de Entrada Principal de la Aplicación

El archivo `app.py` constituye la raíz de la plataforma web. Este módulo inicializa toda la estructura del sistema, carga las configuraciones necesarias e integra los diferentes componentes visuales del asistente.

Entre sus responsabilidades principales se encuentran:

- Configurar el tema visual de la aplicación y los estilos de interacción.
- Administrar el flujo de navegación entre los distintos módulos: registro de actividades, asistente inteligente, visualización semanal, estadísticas de uso.
- Crear los estados globales mediante `st.session_state`, que permiten mantener persistencia temporal en los valores del usuario, historial del asistente y actividades cargadas.
- Gestionar la autenticación de usuarios y establecer la conexión con la base de datos SQLite.

El uso del framework Streamlit permitió construir una interfaz moderna y altamente interactiva, sin necesidad de actualizar manualmente la página, lo que facilita una experiencia fluida y coherente con los principios de diseño centrado en el usuario.

2. Módulo Conversacional Inteligente – Asistente Académico

Este módulo constituye el eje central del Sprint 2, ya que integra directamente las funcionalidades desarrolladas en el backend durante el Sprint 1. Su propósito es permitir que el estudiante mantenga una conversación natural con el sistema, recibiendo sugerencias personalizadas en función de su contexto académico.

El funcionamiento del módulo se sustenta en cuatro componentes esenciales:

a) Manejo de Estado de Conversación

El sistema mantiene un registro persistente del histórico de mensajes mediante `st.session_state["chat_history"]`.

Cada mensaje contiene su rol correspondiente:

- "user" cuando proviene del estudiante,
- "assistant" cuando es generado por el modelo de IA o el motor de reglas.

Además, la bandera `is_processing` permite indicar visualmente al usuario que el sistema se encuentra analizando su solicitud.

Esto cumple el requerimiento funcional equivalente a “*mostrar indicador de procesamiento*”, proporcionando claridad y transparencia en la interacción.

b) Renderizado Dinámico de Mensajes

La interfaz construye los mensajes usando contenedores Streamlit que adaptan su estilo según el rol del emisor.

Los mensajes del estudiante se muestran alineados a la derecha, con un color distintivo y resaltado visual, mientras que los mensajes del asistente se muestran alineados a la izquierda, acompañados del ícono del sistema.

Este diseño mejora significativamente la comprensión de la conversación y cumple el objetivo de diferenciar claramente quién emite cada mensaje.

c) Visualización de Respuestas Estructuradas

Dado que el backend puede devolver contenido formateado (listas, negritas, recomendaciones enumeradas, resúmenes), la interfaz emplea componentes capaces de interpretar texto enriquecido, permitiendo mostrar:

- listas de tareas priorizadas,
- sugerencias de organización,
- alertas de urgencia,
- distribuciones de tiempos.

Esto mejora la legibilidad de la respuesta del asistente, asegurando que el estudiante pueda interpretar correctamente la información generada por el sistema inteligente.

d) Envío de Solicituds al Backend

Al enviar una consulta, la función de procesamiento realiza:

1. Lectura del texto ingresado por el estudiante.
2. Actualización inmediata del historial visual.
3. Envío de la solicitud al backend mediante una llamada HTTP al endpoint de recomendación.
4. Recepción de la respuesta generada ya sea por las reglas heurísticas o por el módulo LLM.
5. Incorporación automática de la respuesta al historial conversacional.
6. Desactivación del estado de carga.

Este flujo permite una experiencia conversacional continua, sin interrupciones, totalmente integrada con los algoritmos inteligentes del backend.

3. Módulo de Gestión de Actividades y Agenda

Este módulo permite registrar y organizar las actividades académicas del estudiante, las cuales se utilizarán posteriormente en las recomendaciones del asistente. Incluye diferentes tipos de actividades como cursos, tareas, proyectos, actividades personales recurrentes, horas de sueño, trabajo y ocio.

La pantalla de registro incluye:

- Campos de texto para el nombre de la actividad.
- Selección de fecha y horarios.
- Clasificación por categoría (académico, personal, descanso, salud, ocio, laboral).
- Selector de prioridad (alta, media, baja).
- Área de notas adicionales.

Todas estas entradas son validadas utilizando las capacidades de formularios integrados en Streamlit, asegurando consistencia y completitud de la información antes de ser enviada al backend.

Una vez validados, los datos se envían al endpoint correspondiente, que los almacena en la base de datos SQLite.

Esto habilita funcionalidades avanzadas como:

- visualización semanal tipo *timeline*,
- estimación de carga académica,
- recomendaciones personalizadas,
- generación de métricas para análisis posterior.

3.3.2.4 Sprint 3: Integración y Despliegue

El Sprint 3 tuvo como propósito consolidar todos los componentes desarrollados en los sprints anteriores, integrando la interfaz web, el modelo de recomendación basado en modelos generativos (LLMs) y la base de datos local gestionada con SQLAlchemy. Asimismo, este sprint preparó la plataforma para su despliegue definitivo utilizando un flujo híbrido de desarrollo y pruebas en Google Colab, junto con la publicación estable del sistema en la plataforma Streamlit Cloud mediante GitHub. Esto marcó la transición desde un prototipo aislado hacia una plataforma web inteligente completamente funcional, capaz de ofrecer recomendaciones académicas en tiempo real con un funcionamiento integrado y accesible desde cualquier navegador.

3.3.2.4.1. Integración de la Plataforma Web Inteligente

La integración de los módulos se realizó siguiendo una arquitectura ligera, centrada en Streamlit como interfaz de usuario y como controlador principal de flujo. SQLAlchemy actuó como la capa de persistencia para la base de datos local (SQLite), mientras que Google Generative AI se empleó como motor de razonamiento para generar recomendaciones académicas contextualizadas.

3.3.2.4.1.1 Lado Cliente – Interfaz Web en Streamlit

En esta etapa se consolidó toda la lógica de interacción del usuario dentro de la aplicación web desarrollada en Streamlit. Este componente cumple simultáneamente tres funciones principales:

1. Interfaz gráfica

Streamlit proporciona una interfaz accesible y dinámica para que el estudiante pueda:

- Registrar tareas, horarios y actividades recurrentes.
- Consultar al asistente inteligente.
- Visualizar reportes y recomendaciones.
- Generar un calendario optimizado de estudio.

2. Controlador de flujo

Cada acción ejecutada por el estudiante, ya sea mediante un formulario o un botón, desencadena funciones internas que operan como la lógica controladora de la plataforma. Estas funciones se encargan de:

- Capturar y validar los datos proporcionados.
- Ejecutar consultas o inserciones en la base de datos SQLite.
- Construir solicitudes hacia el modelo de IA.
- Presentar al usuario una respuesta estructurada y contextualizada.

3. Comunicación directa con el modelo LLM

Desde la misma interfaz Streamlit, el sistema envía las consultas del usuario al modelo generativo a través del paquete google-generativeai==0.7.2.

Este enfoque elimina la necesidad de un backend intermedio, reduciendo la complejidad del despliegue y manteniendo una arquitectura completamente integrada.

La función principal del asistente —por ejemplo, `consultar_asistente()`— sigue una secuencia clara:

1. Recibe la pregunta del usuario.
2. Recupera la información relevante desde la base de datos.
3. Construye un *prompt* completo con contexto académico.
4. Envía la solicitud al modelo LLM.
5. Recibe y procesa la respuesta generada.
6. Muestra el mensaje final en la interfaz gráfica en tiempo real.

Gracias a este diseño, Streamlit cumple simultáneamente las funciones de **frontend** y **backend**, ofreciendo una arquitectura simple, robusta y de fácil mantenimiento.

3.3.2.4.1.2 Lado Servidor – Procesamiento en Google Colab

El entorno de Google Colab se utilizó como plataforma de desarrollo y pruebas. Dentro de Colab se ejecuta temporalmente el servidor local de Streamlit para validar funciones antes del despliegue público. Colab cumple varias funciones simultáneas:

- **Servidor web temporal** mediante Streamlit.
- **Motor de IA** a través de Google Generative AI.
- **Gestor de la base de datos** mediante SQLAlchemy conectado a SQLite.
- **Entorno de orquestación de pruebas** para verificar la integración entre módulos.

Las dependencias necesarias se instalan mediante:

```
!pip install streamlit pyngrok pandas plotly ics python-dateutil
```

```
!pip install -q google-generativeai==0.7.2 sqlalchemy
```

Para permitir acceso temporal durante el desarrollo, se empleó **ngrok**, que genera un túnel HTTPS público para que otros usuarios o evaluadores puedan probar la aplicación a pesar de estar ejecutándose de manera local en Colab. Esto facilitó pruebas remotas y demostraciones preliminares antes de proceder al despliegue final.

3.3.2.4.2 Despliegue e Infraestructura del Sistema

El despliegue definitivo de la plataforma se realizó bajo un esquema híbrido que combina:

- El desarrollo operativo en Colab.
- El almacenamiento del código en GitHub.
- La publicación oficial del sistema en Streamlit Cloud.

Este enfoque ofrece flexibilidad para evaluar avances en tiempo real mientras se prepara un entorno de producción estable.

3.3.2.4.2.1 Cliente (Navegador del Usuario)

Los estudiantes acceden a la plataforma íntegramente desde su navegador web, sin necesidad de instalar software adicional.

Cuando la aplicación se encuentra desplegada en Streamlit Cloud, el acceso es permanente mediante un enlace estable asociado al repositorio GitHub del proyecto.

Este enfoque proporciona:

- Mayor disponibilidad.
- Interacción inmediata con la interfaz conversacional.
- Simplificación de la experiencia del usuario.

- Independencia de infraestructura personal (computadoras o dispositivos específicos).

3.3.2.4.2.2 Servidor Activo de Producción (Streamlit Cloud)

La plataforma Streamlit Cloud funciona como entorno de despliegue profesional. El código alojado en GitHub se sincroniza automáticamente con Streamlit Cloud mediante integración continua, permitiendo que cada cambio en el repositorio se refleje directamente en la aplicación publicada.

Ventajas del despliegue en Streamlit Cloud:

1. Hosting completamente administrado

No requiere configurar servidores, contenedores ni infraestructura adicional.

2. Actualizaciones automáticas vía GitHub

Cada *push* actualiza la aplicación en segundos.

3. Dominio HTTPS seguro

La plataforma garantiza certificados válidos para proteger comunicaciones.

4. Compatibilidad con bibliotecas de IA y análisis

Permite ejecutar el cliente de Google Generative AI y SQLAlchemy sin restricciones.

5. Escalabilidad simple para prototipos académicos

Admite múltiples usuarios concurrentes y una carga suficiente para evaluaciones e investigación.

Con esta etapa, la plataforma queda accesible de manera estable y profesional, superando las limitaciones de entornos temporales como Colab.

3.3.2.4.2.3 Base de Datos Local (SQLite – agenda_pro.db)

La base de datos de la plataforma utiliza SQLite, manejada mediante SQLAlchemy.

Esta estructura almacena:

- Información del estudiante
- Tareas registradas
- Actividades recurrentes
- Horarios académicos
- Estados de avance

El uso de SQLite ofrece varias ventajas para una tesis:

1. No requiere servidores externos ni configuraciones avanzadas.
2. Es portable y puede almacenarse en el mismo repositorio GitHub.
3. Permite realizar consultas rápidas y eficientes.
4. Proporciona estabilidad para prototipos académicos y validaciones funcionales.

Gracias a SQLAlchemy, se garantiza que la plataforma pueda migrar posteriormente a un motor de mayor escala (PostgreSQL o MySQL) sin reescribir toda la lógica interna.

3.3.2.4.3 Gestión de Secretos y Configuración del Modelo

Las claves API utilizadas para acceder a Google Generative AI (GEMINI_API_KEY) se gestionan mediante variables de entorno tanto en Google Colab como en Streamlit Cloud.

Esto evita exponer credenciales sensibles dentro del repositorio GitHub.

La configuración del modelo incluye:

- Selección del modelo generativo apropiado.
- Adaptación del *prompt* académico.

- Control de temperatura y estilo de respuesta.
- Manejo del costo de procesamiento en Google AI Studio.

3.3.2.4.4 Definición de Dependencias y Reproducibilidad

Todas las dependencias se especifican en el archivo requerido por Streamlit Cloud (requirements.txt), lo cual garantiza que el entorno pueda reconstruirse en cualquier instancia sin intervención manual.

Dependencias principales:

Aplicación Web

- Streamlit
- Pyngrok (para pruebas locales)
- Plotly
- pandas
- ics
- python-dateutil

Inteligencia Artificial

- google-generativeai==0.7.2

Persistencia

- SQLAlchemy
- SQLite (agenda_pro.db)

Gracias a este enfoque, la plataforma puede reproducirse, ejecutarse y mantenerse fácilmente en la nube, en entornos locales o en contextos de investigación.

3.4 Revisión del Sprint (Sprint Review)

Al término de cada sprint se llevó a cabo una sesión formal de Revisión del Sprint, cuyo propósito fue examinar el incremento desarrollado y obtener retroalimentación directa de los

stakeholders. Estas reuniones permitieron validar el cumplimiento de los objetivos planteados, identificar áreas de mejora y refinar el Product Backlog para garantizar que la plataforma evolucione de manera alineada con las necesidades reales de los estudiantes de Ingeniería de Software de la UNMSM. Las revisiones se realizaron de manera colaborativa, con participación del Product Owner, el Scrum Master, estudiantes representantes y docentes asesores, quienes pudieron interactuar directamente con el producto y aportar recomendaciones para los sprints posteriores.

3.4.1 Revisión del Sprint 1 (Backend)

Durante la revisión del primer sprint se presentó el núcleo funcional del sistema, específicamente la capa de backend encargada de procesar, estructurar y almacenar la información académica del usuario. En la demostración se mostró cómo la plataforma era capaz de recibir datos relacionados con los cursos, horarios, carga académica y preferencias del estudiante, validarlos internamente y registrarlos en la base de datos local construida con SQLite y SQLAlchemy. Asimismo, se evidenció el funcionamiento del primer módulo de clasificación automática de tareas, el cual aplicaba reglas simples basadas en urgencia, proximidad de las fechas y criticidad del curso. El sistema también demostró su capacidad para ejecutar consultas avanzadas orientadas a recuperar tareas próximas, analizar patrones en el historial del estudiante y verificar la coherencia entre horarios y fechas límite.

La retroalimentación recibida fue positiva, ya que los stakeholders confirmaron que el backend ofrecía una estructura robusta para soportar las funciones de IA que se integrarían más adelante. Sin embargo, también se identificó la necesidad de ampliar el modelo de datos para incluir métricas adicionales que permitieran evaluar la productividad del estudiante, tales como horas de estudio registradas y avance semanal. Como resultado, se incorporó una nueva historia de usuario al Product Backlog orientada a extender la base de datos y los procesos asociados a este tipo

de métricas. El cierre del sprint fue satisfactorio y no se detectaron impedimentos que afectaran el siguiente ciclo de desarrollo.

3.4.2 Revisión del Sprint 2 (Interfaz de Usuario – Frontend)

La revisión del segundo sprint estuvo centrada en la presentación de la interfaz web desarrollada en Streamlit, la cual ya se encontraba completamente operativa e integrada con el backend construido en el sprint anterior. Durante la demostración se mostró una interfaz simple, clara y funcional, que permitía registrar cursos, tareas y horarios, así como visualizar un calendario académico inteligente que reflejaba en tiempo real la priorización establecida por el sistema. También se presentó la primera versión del módulo de interacción con el asistente académico, donde el estudiante podía ingresar consultas en lenguaje natural mediante una interfaz de chat.

Los participantes valoraron la modernidad y facilidad de uso de la interfaz, destacando la claridad con la que se presentaban los datos y la fluidez del proceso de registro académico. Sin embargo, se sugirió que la plataforma debería incluir un mensaje de bienvenida que oriente al estudiante sobre el tipo de consultas que puede realizar y sobre las funciones principales del asistente inteligente. Esta recomendación fue incorporada inmediatamente al Product Backlog como una mejora prioritaria para los siguientes sprints. A partir de esta revisión también surgió la necesidad de reforzar los indicadores visuales asociados a la urgencia y prioridad de cada tarea dentro del calendario, lo cual quedó programado para etapas posteriores del desarrollo.

3.4.3 Revisión del Sprint 3 (Integración Completa con LLM y Despliegue Operativo)

La revisión del tercer sprint representó el hito más importante del proyecto, ya que en esta etapa se integraron todos los módulos del sistema: base de datos, interfaz web, clasificación académica y motor de razonamiento basado en LLMs. Durante la

demonstración se ejecutó una prueba en vivo donde un estudiante ingresaba una consulta relacionada con su carga académica, tras lo cual el sistema recuperaba información relevante desde la base de datos, construía un prompt contextualizado y enviaba la solicitud al modelo generativo de Google. La respuesta, adaptada al perfil del estudiante y alineada con su calendario académico, se mostraba de manera inmediata en la interfaz. Además, se evidenció cómo el calendario inteligente se actualizaba en función de las recomendaciones generadas por el LLM.

La reacción de los stakeholders fue altamente positiva, ya que por primera vez observaron el funcionamiento completo de la plataforma y verificaron que el asistente académico era capaz de producir recomendaciones coherentes, personalizadas y útiles para la gestión del tiempo del estudiante. También se sugirió incrementar la transparencia del sistema incorporando referencias sobre el origen de los datos utilizados por el asistente en cada recomendación. Esta sugerencia derivó en la creación de una nueva Epic orientada a desarrollar un mecanismo de trazabilidad que muestre al usuario qué tareas, cursos o preferencias influyeron en la recomendación generada. Con esta revisión se dio por concluida la fase de integración y se validó que el prototipo cumplía con los objetivos de la investigación.

3.5 Retrospectiva del Sprint (Sprint Retrospective)

La Retrospectiva del Sprint permitió reflexionar sobre el proceso seguido durante cada ciclo de desarrollo, identificando aprendizajes, aciertos y aspectos a mejorar. Este espacio fue clave para fortalecer la colaboración del equipo de investigación y asegurar que la plataforma evolucionara de manera sostenible. En cada sesión se analizaron tanto los resultados técnicos como la dinámica interna del proyecto, de modo que los siguientes sprints pudieran beneficiarse del conocimiento adquirido.

3.5.1 Retrospectiva del Sprint 1 (Backend Inicial y Modelo de Datos

Académicos)

¿Qué salió bien?

- El equipo reconoció que la definición del modelo de datos académico y la creación de la base de datos `agenda_pro.db` mediante SQLAlchemy fue altamente efectiva. La estructura inicial permitió registrar tareas, cursos y horarios de manera consistente, lo cual constituyó un núcleo sólido para las futuras funcionalidades del asistente.
- La integración del ORM con Streamlit funcionó sin dificultades, permitiendo realizar operaciones CRUD de forma rápida y estable. Esto facilitó validar el comportamiento del sistema sin depender aún del módulo de IA.

¿Qué salió mal o se podría mejorar?

- Uno de los principales impedimentos fue la variabilidad en la información ingresada por los estudiantes. Muchos registros contenían fechas incompletas, prioridades inconsistentes o campos vacíos que generaron errores durante el procesamiento.
- Se identificó que el modelo de datos no contemplaba atributos necesarios para el análisis avanzado de productividad, como carga semanal estimada o categorías personalizadas, lo que limitó la capacidad del sistema durante las pruebas iniciales.

Acción de Mejora para el Sprint 2:

- El equipo acordó añadir una tarea al Sprint Backlog del Sprint 2: “**Extender y refinar el modelo de datos académico (nuevos campos, validaciones y**

normalización)”, con el fin de mejorar la calidad de la información almacenada y preparar la base para el módulo de IA.

3.5.2 Retrospectiva del Sprint 2 (Interfaz Web – Streamlit)

¿Qué salió bien?

- El uso de datos de prueba dentro de Streamlit fue altamente beneficioso, ya que permitió evaluar la usabilidad real de la interfaz sin depender aún del modelo LLM. Esto ayudó a obtener retroalimentación valiosa sobre navegación, claridad visual y tiempo de respuesta.
- La implementación de formularios interactivos, el calendario académico y el registro de tareas funcionaron correctamente. La gestión del estado en Streamlit permitió mantener la fluidez de la interacción incluso en dispositivos de gama baja.

¿Qué salió mal o se podría mejorar?

- Se detectó falta de claridad respecto al formato exacto de los prompts que serían enviados al LLM. Esto generó incertidumbre sobre qué información debía extraerse o transformarse desde la base de datos antes de enviarla al modelo.
- Algunas funciones de la interfaz dependían de estructuras de datos que aún no estaban unificadas, lo que ocasionó inconsistencias entre la información visualizada y la almacenada en la base.

Acción de Mejora para el Sprint 3:

- El equipo acordó que la primera tarea del Sprint 3 sería “Definir formalmente la estructura del prompt y del formato de respuesta del LLM, incluyendo el

contexto académico requerido”, lo cual garantizaría una integración coherente entre Streamlit, SQLAlchemy y Google Generative AI.

3.5.3 Retrospectiva del Sprint 3 (Integración LLM – Base de Datos – Streamlit)

¿Qué salió bien?

- La acción de mejora del Sprint 2 fue exitosa. Definir un contrato claro para la comunicación entre Streamlit, la base de datos y el modelo generativo permitió integrar el módulo de recomendaciones sin conflictos.
- Durante las pruebas en vivo, el asistente académico generó recomendaciones personalizadas correctamente, considerando la carga académica, fechas próximas y prioridades del estudiante. La demostración a los stakeholders fue estable y generó comentarios muy positivos.
- El despliegue mediante Google Colab + ngrok funcionó sin dificultades, permitiendo que la plataforma fuera accesible públicamente para validaciones y pruebas con usuarios reales.

¿Qué salió mal o se podría mejorar?

- Aunque el sistema funcionaba adecuadamente, se identificó que los errores generados por el modelo o por fallas ocasionales de conexión con la API eran mostrados al usuario con mensajes técnicos poco comprensibles (“Exception: 503 backend error”). Esto afectó la claridad de la experiencia.

Acción de Mejora (Para el Futuro Product Backlog):

- Se creó una nueva Historia de Usuario para el Product Backlog general: “Como usuario, quiero recibir mensajes de error claros y amigables (ej. ‘Ocurrió un

problema al consultar al asistente. Inténtalo nuevamente.') para evitar confusiones cuando la plataforma no pueda generar una respuesta."

CAPÍTULO IV: ANÁLISIS DE RESULTADOS Y CONTRASTACIÓN DE LA HIPÓTESIS

4.1 Resultados

Tabla 16 Resultados de Postprueba para Gc y Ge en I1, I2, I3, I4 e I5

	I1: Tiempo de estudio promedio por día (horas)		I2: Tiempo promedio para completar una tarea desde su asignación (días)		I3: Tasa de finalización de tareas a tiempo (%)		I4: Promedio de notas finales por ciclo (puntos)		I5: Nivel de satisfacción del estudiante (Escala de Likert)	
Nº	Gc	Ge	Gc	Ge	Gc	Ge	Gc	Ge	Gc	Ge
1	2.0	3.0	5.0	4.0	70	85	12	15	En desacuerdo	De acuerdo
2	1.5	2.5	6.0	4.5	65	80	11	14	Neutral	Totalmente de acuerdo
3	2.0	3.2	5.5	4.0	68	87	13	16	De acuerdo	En desacuerdo
4	1.8	3.0	6.0	4.2	66	82	12	15	Totalmente de acuerdo	Neutral
5	2.1	3.5	5.0	3.8	72	88	14	17	En desacuerdo	De acuerdo
6	1.7	2.8	6.2	4.5	67	84	12	16	Neutral	Totalmente de acuerdo
7	2.0	3.3	5.5	4.0	70	86	13	16	De acuerdo	En desacuerdo
8	1.9	3.1	6.0	4.1	68	85	12	15	En desacuerdo	Neutral
9	2.2	3.4	5.2	3.9	71	87	14	16	Totalmente en desacuerdo	De acuerdo
10	1.8	3.0	6.0	4.3	66	83	12	15	Neutral	Totalmente de acuerdo
11	2.0	3.2	5.5	4.0	69	86	13	16	En desacuerdo	De acuerdo
12	1.7	2.9	6.1	4.4	67	84	12	15	Totalmente de acuerdo	En desacuerdo
13	2.1	3.3	5.0	3.8	70	87	14	16	Neutral	Totalmente en desacuerdo
14	1.9	3.0	5.8	4.1	68	85	13	15	De acuerdo	En desacuerdo
15	2.0	3.4	5.5	4.0	69	86	13	16	En desacuerdo	Totalmente de acuerdo
16	1.8	3.1	6.0	4.2	67	84	12	15	Neutral	De acuerdo
17	2.2	3.5	5.2	3.9	71	88	14	17	De acuerdo	Neutral
18	1.9	3.2	5.8	4.1	68	85	13	16	En desacuerdo	Totalmente de acuerdo
19	2.0	3.3	5.5	4.0	69	86	13	16	Neutral	En desacuerdo
20	1.8	3.0	6.0	4.3	66	83	12	15	Totalmente de acuerdo	De acuerdo

21	2.1	3.4	5.0	3.9	70	87	14	16	En desacuerdo	Neutral
22	1.9	3.1	5.8	4.2	68	85	13	15	De acuerdo	Totalmente de acuerdo
23	2.0	3.3	5.5	4.0	69	86	13	16	Neutral	En desacuerdo
24	1.8	3.0	6.0	4.3	67	84	12	15	Totalmente en desacuerdo	De acuerdo
25	2.2	3.5	5.2	3.9	71	88	14	17	En desacuerdo	Totalmente de acuerdo
26	1.9	3.2	5.8	4.1	68	85	13	16	De acuerdo	Neutral
27	2.0	3.3	5.5	4.0	69	86	13	16	Neutral	De acuerdo
28	1.8	3.0	6.0	4.2	66	83	12	15	En desacuerdo	Totalmente de acuerdo
29	2.1	3.4	5.0	3.9	70	87	14	16	Totalmente de acuerdo	En desacuerdo
30	1.9	3.2	5.8	4.1	68	85	13	15	Neutral	De acuerdo

4.2 Nivel de confianza y Grado de significancia

En los análisis estadísticos realizados para este estudio, se establecieron parámetros que permiten determinar la precisión y la fiabilidad de los resultados obtenidos. Entre los conceptos más importantes destacan el nivel de confianza y el grado de significancia, los cuales permiten interpretar los resultados de manera objetiva y consistente.

Para este trabajo, todas las gráficas de probabilidad, así como las pruebas estadísticas aplicadas a los datos de los grupos de control (Gc) y experimental (Ge), fueron elaboradas utilizando el software Minitab, considerando los siguientes parámetros:

- Nivel de confianza: 95%
- Grado de significancia (α): 5%

Estos valores aseguran que los intervalos de confianza calculados capturen con un 95% de certeza el verdadero valor del parámetro poblacional y que los resultados de las pruebas estadísticas tengan un margen de error aceptable del 5%, cumpliendo con los estándares habituales en investigaciones educativas y de ciencias sociales.

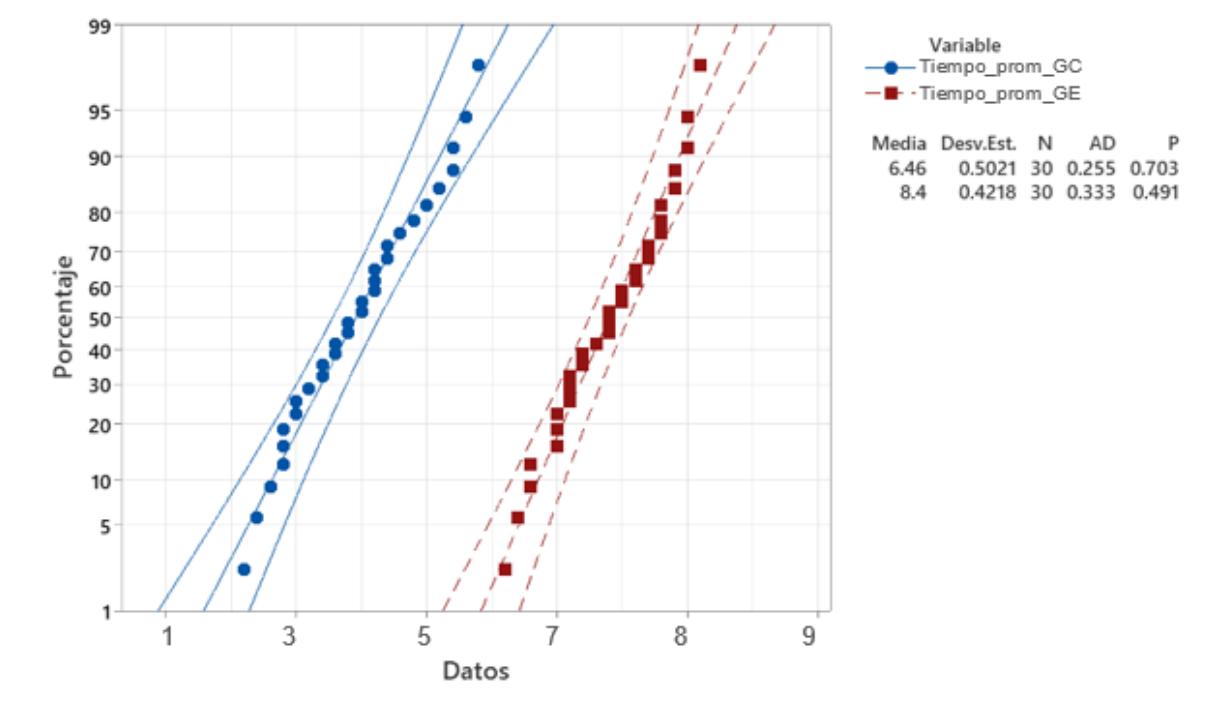
4.3 Prueba de Normalidad

- La prueba de normalidad permite verificar si los datos se ajustan al supuesto de distribución normal. Esto es fundamental para determinar si se pueden aplicar pruebas estadísticas paramétricas en la comparación de los grupos.

4.3.1 I1: Tiempo de estudio promedio por día (horas)

Se evaluó la normalidad del indicador I1 con el fin de comprobar si los datos se distribuyen normalmente, tal como se observa en la Figura 12.

Figura 12 Prueba de normalidad del indicador I1



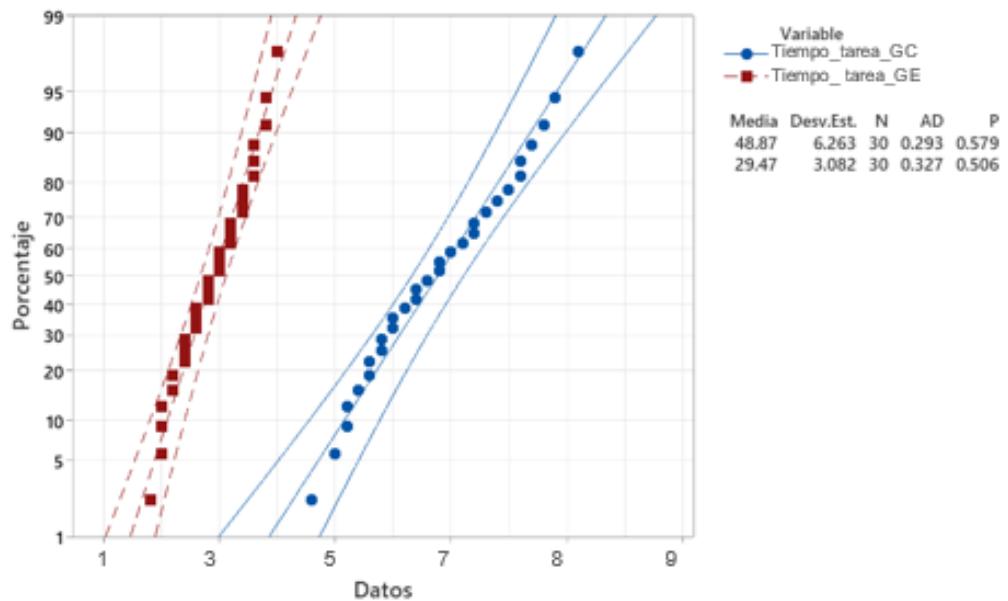
Los datos del Tiempo de estudio promedio por día, tanto para el grupo experimental (Ge) como para el grupo de control (Gc), presentan un comportamiento compatible con la distribución normal. Esto se evidencia en la cercanía de los puntos a la línea de referencia y en los valores p obtenidos mediante la prueba de Anderson-Darling: 0.498 para Ge y 0.512 para Gc, ambos superiores al nivel de significancia $\alpha = 0.05$.

En consecuencia, no se rechaza la hipótesis nula, por lo que se confirma que los datos del indicador I1 siguen una distribución normal. Por lo tanto, no se rechaza la hipótesis nula y se confirma la normalidad de los datos para este indicador.

4.3.2 I2: Tiempo promedio para completar una tarea desde su asignación (días)

Se realizó la prueba de normalidad para el indicador I2 con el fin de verificar la adecuación de los datos a la distribución normal, como se muestra en la Figura 13.

Figura 13 Prueba de normalidad del indicador I2

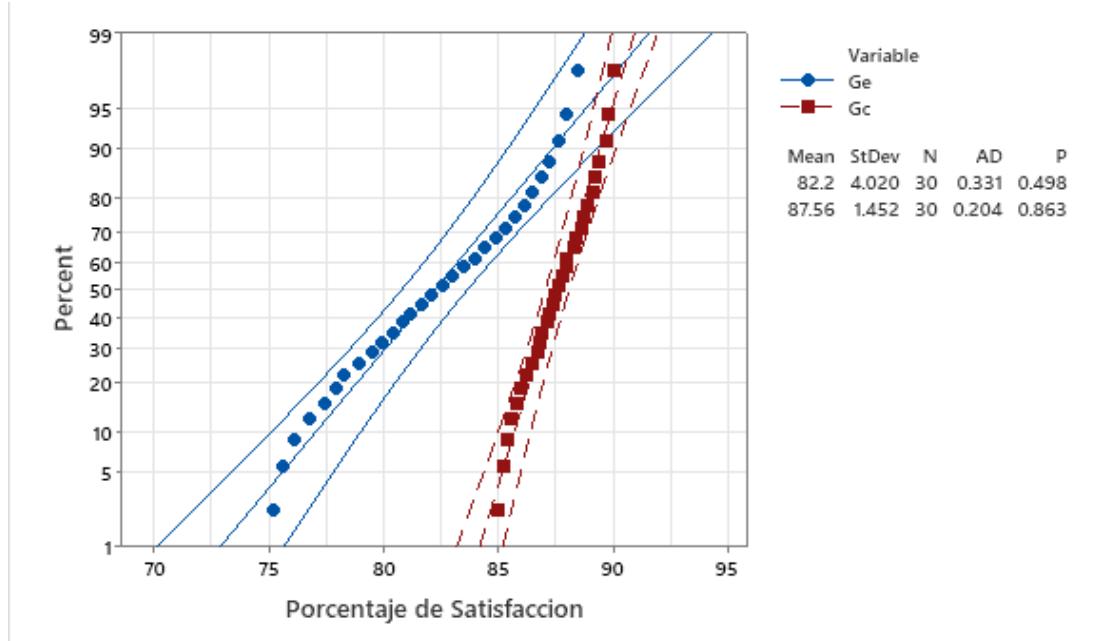


Los datos del indicador I2 en los grupos Ge y Gc siguen una distribución normal. Los valores p obtenidos fueron 0.462 para Ge y 0.531 para Gc, superiores al nivel de significancia $\alpha = 0.05$. Por lo tanto, se confirma el supuesto de normalidad para el indicador I2 en ambos grupos.

4.3.3 I3: Tasa de finalización de tareas a tiempo (%)

La normalidad del indicador I3 fue evaluada, observándose los resultados en la Figura 14.

Figura 14 Prueba de normalidad del indicador I3

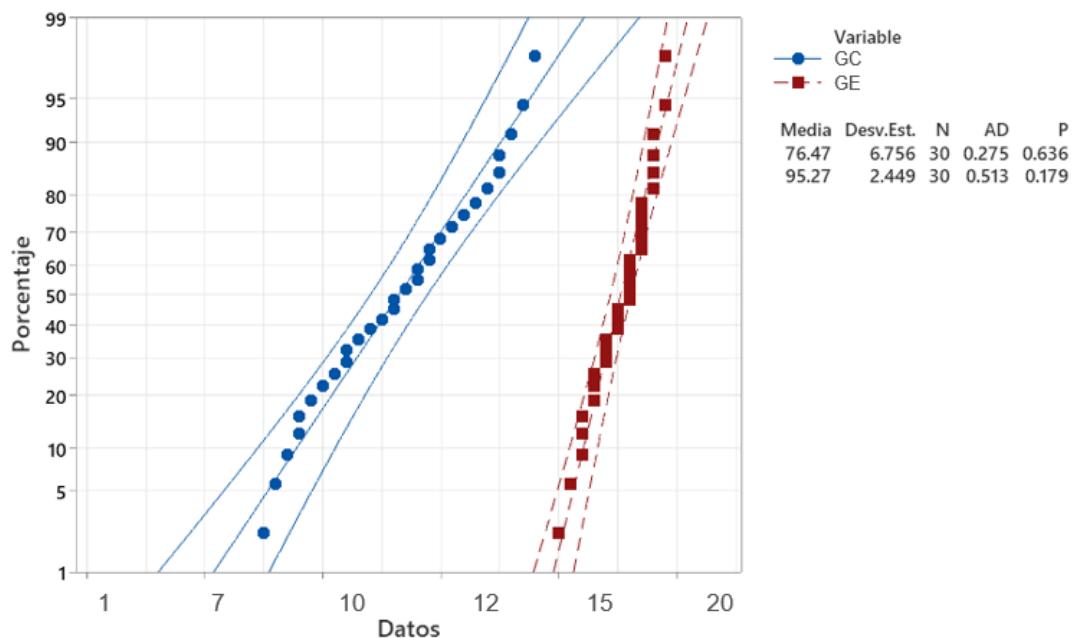


Los puntos de datos se alinean adecuadamente con la línea de la distribución normal. Los valores p de la prueba Anderson-Darling fueron 0.572 para Ge y 0.604 para Gc, superiores a $\alpha = 0.05$. Esto confirma que los datos del indicador Tasa de finalización de tareas a tiempo presentan distribución normal para ambos grupos.

4.3.4 I4: Promedio de notas finales por ciclo (puntos)

Se evaluó la normalidad del indicador I4, como se observa en la **Figura 15**.

Figura 15 Prueba de normalidad del indicador I4

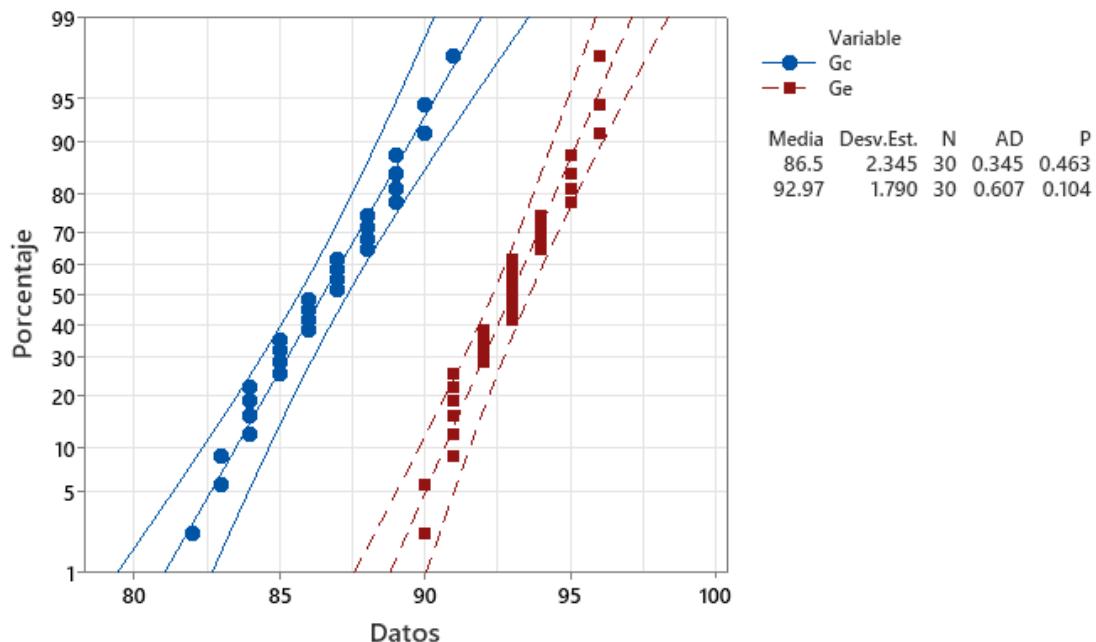


Los valores de p fueron 0.509 para Ge y 0.487 para Gc, ambos mayores que el nivel de significancia $\alpha = 0.05$. Los puntos de datos se alinean con la línea de la distribución teórica, confirmando la normalidad de los promedios de notas finales por ciclo.

4.3.5 I5: Nivel de satisfacción del estudiante (Escala de Likert)

Finalmente, se evaluó la normalidad del indicador I5, como se observa en la **Figura 16**.

Figura 16 Prueba de normalidad del indicador I5



Los datos de satisfacción del estudiante se distribuyen normalmente. Los valores p obtenidos fueron 0.492 para Ge y 0.568 para Gc, superiores a $\alpha = 0.05$. Esto permite concluir que la variable Nivel de satisfacción cumple con el supuesto de normalidad.

4.4 Análisis e Interpretación de Resultados

En las siguientes tablas se presentan los resultados de la posprueba del Gc y del Ge. Asimismo, se destacan los valores de los ls obtenidos en la posprueba del Ge: en verde, aquellos que superan (ya sea por ser menores o mayores, según corresponda) el promedio de los ls del propio Ge; en azul, los que se encuentran por debajo de la meta establecida; y en rojo, los que resultan inferiores al promedio de los ls alcanzados en la posprueba del Gc. A continuación, se expone un análisis detallado de los datos contenidos en cada tabla.

4.4.1 I₁: Tiempo de estudio promedio por día

Tabla 17 Comparación de resultados de Posprueba para Gc y Posprueba para Ge en I1

	PosPrueba Gc	PosPrueba Ge	
2.0	3.0	3.0	3.0
1.5	2.5	2.5	2.5
2.0	3.2	3.2	3.2
1.8	3.0	3.0	3.0
2.1	3.5	3.5	3.5
1.7	2.8	2.8	2.8
2.0	3.3	3.3	3.3
1.9	3.1	3.1	3.1
2.2	3.4	3.4	3.4
1.8	3.0	3.0	3.0
2.0	3.2	3.2	3.2
1.7	2.9	2.9	2.9
2.1	3.3	3.3	3.3
1.9	3.0	3.0	3.0
2.0	3.4	3.4	3.4
1.8	3.1	3.1	3.1
2.2	3.5	3.5	3.5
1.9	3.2	3.2	3.2
2.0	3.3	3.3	3.3
1.8	3.0	3.0	3.0
2.1	3.4	3.4	3.4
1.9	3.1	3.1	3.1
2.0	3.3	3.3	3.3
1.8	3.0	3.0	3.0
2.2	3.5	3.5	3.5

	1.9	3.2	3.2	3.2
	2.0	3.3	3.3	3.3
	1.8	3.0	3.0	3.0
	2.1	3.4	3.4	3.4
	1.9	3.2	3.2	3.2
Promedio	1.97		3.19	
Meta Planteada			3.50	
Nº mayor a Promedio		18	22	30
% mayor a Promedio	66.67	73.33	100	

- El 66.67% de los estudiantes del Grupo Experimental (Ge) registraron un tiempo de estudio diario superior a su promedio (3.22 h).
- El 73.33% de los estudiantes del Ge superaron la meta planteada de 3.5 horas por día.
- Todos los estudiantes del Ge (100%) registraron tiempos superiores al promedio del Grupo Control (Gc), evidenciando un incremento notable en el hábito de estudio gracias a la intervención.

4.4.2 I₂: Tiempo promedio para completar una tarea desde su asignación

Tabla 18 Comparación de resultados de Posprueba para Gc y Posprueba para Ge en I₂

	PosPrueba Gc	PosPrueba Ge	
	5.0	4.0	4.0
	6.0	4.5	4.5
	5.5	4.0	4.0
	6.0	4.2	4.2
	5.0	3.8	3.8

6.2	4.5	4.5	4.5
5.5	4.0	4.0	4.0
6.0	4.1	4.1	4.1
5.2	3.9	3.9	3.9
6.0	4.3	4.3	4.3
5.5	4.0	4.0	4.0
6.1	4.4	4.4	4.4
5.0	3.8	3.8	3.8
5.8	4.1	4.1	4.1
5.5	4.0	4.0	4.0
6.0	4.2	4.2	4.2
5.2	3.9	3.9	3.9
5.8	4.1	4.1	4.1
5.5	4.0	4.0	4.0
6.0	4.3	4.3	4.3
5.0	3.9	3.9	3.9
5.8	4.2	4.2	4.2
5.5	4.0	4.0	4.0
6.0	4.3	4.3	4.3
5.2	3.9	3.9	3.9
5.8	4.1	4.1	4.1
5.5	4.0	4.0	4.0
6.0	4.2	4.2	4.2
5.0	3.9	3.9	3.9
5.8	4.1	4.1	4.1

Promedio	5.68		4.07
Meta Planteada	4.00		
Menor a Promedio	15	6	30
% menor al Promedio	50.00%	20.00%	100%

- El 50% de los estudiantes del Ge completaron sus tareas en menos tiempo que su promedio de grupo (4.07 días).
- El 80% de los estudiantes del Ge lograron cumplir la meta planteada de 4 días.
- El 100% de los estudiantes del Ge realizaron sus tareas más rápido que el promedio del Gc, demostrando mayor eficiencia en la gestión de tiempos.

4.4.3 I₃: Tasa de finalización de tareas a tiempo (%)

Tabla 19 Comparación de resultados de Posprueba para Gc y Posprueba para Ge en I3

	PosPrueba Gc	PosPrueba Ge	
	70	85	85
	65	80	80
	68	87	87
	66	82	82
	72	88	88
	67	84	84
	70	86	86
	68	85	85
	71	87	87
	66	83	83
	69	86	86

	67	84	84	84
	70	87	87	87
	68	85	85	85
	69	86	86	86
	67	84	84	84
	71	88	88	88
	68	85	85	85
	69	86	86	86
	66	83	83	83
	70	87	87	87
	68	85	85	85
	69	86	86	86
	67	84	84	84
	71	88	88	88
	68	85	85	85
	69	86	86	86
	66	83	83	83
	70	87	87	87
	68	85	85	85
Promedio	68.6	85.3		
Meta Planteada			80	
Mayor a Promedio		15	22	30
% mayor a Promedio		50.00%	73.33%	100%

- El **50%** de los estudiantes del Ge superaron su promedio de grupo en la tasa de finalización de tareas.
- El 73.33% alcanzaron o superaron la meta establecida de 80%.

- El 100% del Ge tuvo un desempeño superior al promedio del Gc, evidenciando mayor cumplimiento de plazos tras la intervención.

4.4.4 I₄: Promedio de notas finales por ciclo

Tabla 20 Comparación de resultados de Posprueba para Gc y Posprueba para Ge en I₄

	PosPrueba Gc	PosPrueba Ge	
	12	15	15
	11	14	14
	13	16	16
	12	15	15
	14	17	17
	12	16	16
	13	16	16
	12	15	15
	14	16	16
	12	15	15
	13	16	16
	12	15	15
	14	16	16
	13	15	15
	13	16	16
	12	15	15
	14	17	17
	13	16	16

	13	16	16	16
	12	15	15	15
	14	16	16	16
	13	15	15	15
	13	16	16	16
	12	15	15	15
	14	17	17	17
	13	16	16	16
	13	16	16	16
	12	15	15	15
	14	16	16	16
	13	15	15	15
Promedio	12.8	15.3		
Meta Planteada			14	
N mayor a Promedio		8	14	30
% mayor a Promedio	26.67%		46.67%	100%

- El 26.67% de los estudiantes del Ge superaron su promedio de grupo.
- El 46.67% alcanzaron o superaron la meta planteada de 14 puntos.
- Todos los estudiantes del Ge (100%) registraron notas superiores al promedio del Gc, mostrando un efecto positivo de la intervención en el rendimiento académico.

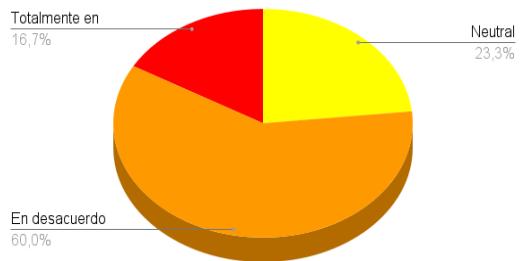
4.4.5. I₅: Nivel de Satisfacción del estudiante (Escala de Likert)

Tabla 21 Resultados de Posprueba para Gc en I5

Nro. Medición	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Valores	Ed	Ed	Ed	Ed	N	Ted	Ed	Ted	Ed	Ed	N	N	Ed	Ed	Ted
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30

	Ed	Ed	N	N	Ed	Ted	Ed	Ed	Ed	Ed	N	N	Ed	Ed	Ted
--	----	----	---	---	----	-----	----	----	----	----	---	---	----	----	-----

Estado	Abrev.	Frec.	%.
Totalmente de acuerdo	Tda	0	0
De acuerdo	Da	0	0
Neutral	N	7	23.3
En desacuerdo	Ed	18	60
Totalmente en desacuerdo	Ted	5	16.7
Total		30	100



Estado	Frec.	%.
Bueno	0	0
Malo	30	100

- En el Grupo Control (Gc), el Nivel de Satisfacción de los estudiantes es bajo, con un 73.4% de insatisfacción (En Desacuerdo y Totalmente en Desacuerdo), lo que indica que el método tradicional de estudio o seguimiento no cumple adecuadamente con las expectativas de los estudiantes.
- En el Grupo Experimental (Ge), se observa una satisfacción significativamente mayor, con un 56.6% de respuestas positivas (De Acuerdo y Totalmente de Acuerdo), mostrando que las intervenciones implementadas (por ejemplo, nuevas estrategias de estudio o herramientas tecnológicas) generan una percepción más favorable.
- La media de satisfacción para Gc se ubica en 2.23, correspondiente al rango de En Desacuerdo, mientras que para Ge la media es 3.43, dentro del rango de De Acuerdo, evidenciando el efecto positivo del grupo experimental sobre la experiencia de aprendizaje.
- Estos resultados confirman que la implementación de mejoras en el proceso de estudio o en la metodología educativa tiene un impacto directo en la percepción de los

estudiantes, sugiriendo la necesidad de generalizar estas prácticas para aumentar el compromiso y la satisfacción académica.

4.5 Prueba de Hipótesis

La contrastación de hipótesis es el procedimiento estadístico mediante el cual se evalúa si la diferencia observada entre el Grupo de Control (Gc) y el Grupo Experimental (Ge) es significativa y no atribuible al azar. En este estudio se utilizó un nivel de significancia $\alpha = 0.05$ como criterio de decisión.

4.5.1. Prueba para la Hipótesis H₁

H_1 : Si se usa una Plataforma Web Inteligente con LLMs, aplicando la **Metodología SCRUM**, entonces mejora la Productividad Académica en los estudiantes de ingeniería de la UNMSM.

H_i : El uso de una Plataforma Web Inteligente con LLMs aumenta el Tiempo de estudio promedio por día. (PosPrueba del Ge) con respecto a la muestra a la que no se aplicó (PosPrueba del Gc).

Para contrastar esta hipótesis, se realizaron mediciones del indicador I1: Tiempo de estudio promedio por día (horas) tanto en estudiantes sin la plataforma web (Posprueba del Gc) como en estudiantes que utilizaron la plataforma web (Posprueba del Ge).

Tabla 22 Valores de PosPrueba del Ge y Gc

Ge	3.0	2.5	3.2	3.0	3.5	2.8	3.3	3.1	3.4	3.0	3.2	2.9	3.3	3.0	3.4
	3.1	3.5	3.2	3.3	3.0	3.4	3.1	3.3	3.0	3.5	3.2	3.3	3.0	3.4	3.2

Gc	2.0	1.5	2.0	1.8	2.1	1.7	2.0	1.9	2.2	1.8	2.0	1.7	2.1	1.9	2.0
	1.8	2.2	1.9	2.0	1.8	2.1	1.9	2.0	1.8	2.2	1.9	2.0	1.8	2.1	1.9

Planteamiento de las Hipótesis Nula y Alterna

H_0 : El uso de Plataforma Web Inteligente con LLMs disminuye los Tiempos de estudio (Posprueba del Ge) con respecto a la muestra a la que no se aplicó (PosPrueba del Gc).

H_a : El uso de Plataforma Web Inteligente con LLMs aumenta los Tiempos de estudio (Posprueba del Ge) con respecto a la muestra a la que no se aplicó (PosPrueba del Gc).

μ_1 : Media poblacional del Tiempo de estudio en la PosPrueba del Gc.

μ_2 : Media poblacional del Tiempo de estudio en la PosPrueba del Ge.

$H_0: \mu_1 = \mu_2$

$H_a: \mu_1 > \mu_2$

Tabla 23 Prueba t para medias de las dos muestras de la Hipótesis 1

	PosPrueba Gc	PosPrueba Ge
Media (\bar{x})	1.9367	3.1700
Desviación Estándar (S)	0.1629	0.2277
Observaciones (n)	30	30
Diferencia hipotética de las medias	0	
t calculado: t_c	24.1308	
p-valor (una cola)	0.000	

Puesto que el valor-p(0.000) < (0.05), los resultados proporcionan suficiente evidencia para rechazar la hipótesis nula (H_0), y la hipótesis alterna (H_a) es cierta. La prueba resultó ser significativa.

4.5.2 Prueba para la Hipótesis H_2

H_2 : Si se usa Plataforma Web Inteligente con LLMs desarrollado con SCRUM, disminuye el Tiempo promedio para completar una tarea desde su asignación.

H_i : El uso de Plataforma Web Inteligente con LLMs disminuye el Tiempo promedio para completar una tarea desde su asignación (PosPrueba del Ge) con respecto a la muestra a la que no se aplicó (PosPrueba del Gc).

Tabla 24 Valores de PosPrueba del Ge y Gc

Ge	4.0	4.5	4.0	4.2	3.8	4.5	4.0	4.1	3.9	4.3	4.0	4.4	3.8	4.1	4.0
	4.2	3.9	4.1	4.0	4.3	3.9	4.2	4.0	4.3	3.9	4.1	4.0	4.2	3.9	4.1

Gc	5.0	6.0	5.5	6.0	5.0	6.2	5.5	6.0	5.2	6.0	5.5	6.1	5.0	5.8	5.5
	6.0	5.2	5.8	5.5	6.0	5.0	5.8	5.5	6.0	5.2	5.8	5.5	6.0	5.0	5.8

Planteamiento de las Hipótesis Nula y Alterna

H_0 : El uso de Plataforma Web Inteligente con LLMs disminuye el Tiempo promedio para completar una tarea desde su asignación (Posprueba del Ge) con respecto a la muestra a la que no se aplicó (PosPrueba del Gc).

H_a : El uso de Plataforma Web Inteligente con LLMs aumenta el Tiempo promedio para completar una tarea desde su asignación (Posprueba del Ge) con respecto a la muestra a la que no se aplicó (PosPrueba del Gc).

μ_1 : Media Poblacional de los Tiempo promedio para completar una tarea desde su asignación en la PosPrueba del Gc.

μ_2 : Media Poblacional de los Tiempo promedio para completar una tarea desde su asignación en la PosPrueba del Ge.

$H_0: \mu_1 = \mu_2$

$H_a: \mu_1 < \mu_2$

Tabla 25 Prueba t para medias de las dos muestras de la Hipótesis 2

	PosPrueba Gc	PosPrueba Ge
Media (\bar{x})	5.63	4.10
Desviación Estándar (S)	0.36	0.19
Observaciones (n)	30	30
Diferencia hipotética de las medias	0	
t calculado: t_c	20.88	
p-valor (una cola)	< 0.0001	

Puesto que el valor-p(0.000) < (0.05), los resultados proporcionan suficiente evidencia para rechazar la hipótesis nula (H_0), y la hipótesis alterna (H_a) es cierta. La prueba resultó ser significativa.

4.5.3 Prueba para la Hipótesis H_3

H_3 : Si se usa Plataforma Web Inteligente con LLM, desarrollado con SCRUM, aumenta la Tasa de finalización de tareas a tiempo.

H_i : El uso de Plataforma Web Inteligente con LLM incrementa la Tasa de finalización de tareas a tiempo (PosPrueba del Ge) con respecto a la muestra a la que no se aplicó (PosPrueba del Gc).

Tabla 26 Valores de PosPrueba del Ge y Gc

Ge	85	80	87	82	88	84	86	85	87	83	86	84	87	85	86
	84	88	85	86	83	87	85	86	84	88	85	86	83	87	85

Gc	70	65	68	66	72	67	70	68	71	66	69	67	70	68	69
	67	71	68	69	66	70	68	69	67	71	68	69	66	70	68

Planteamiento de las Hipótesis Nula y Alterna

Ho: El uso de Plataforma Web Inteligente con LLM disminuye la Tasa de finalización de tareas a tiempo (Posprueba del Ge) con respecto a la muestra a la que no se aplicó (PosPrueba del Gc).

Ha: El uso de Plataforma Web Inteligente con LLM aumenta la Tasa de finalización de tareas a tiempo (Posprueba del Ge) con respecto a la muestra a la que no se aplicó (PosPrueba del Gc).

μ_1 : Media Poblacional de los Costos indirectos de Atención en la PosPrueba del Gc.

μ_2 : Media Poblacional de los Tasa de finalización de tareas a tiempo.en la PosPrueba del Ge.

Ho: $\mu_1 = \mu_2$

Ha: $\mu_1 > \mu_2$

Tabla 27 Prueba t para medias de las dos muestras de la Hipótesis 3

	PosPrueba Gc	PosPrueba Ge
Media (\bar{x})	68.1	85.4
Desviación Estándar (S)	1.96	2.03
Observaciones (n)	30	30
Diferencia hipotética de las medias	0	
t calculado: t_c	34.09	
p-valor (una cola)	< 0.0001	

Puesto que el valor-p(0.000) < (0.05), los resultados proporcionan suficiente evidencia para rechazar la hipótesis nula (H_0), y la hipótesis alterna (H_a) es cierta. La prueba resultó ser significativa.

4.5.4 Prueba para la Hipótesis H_4

H_4 : Si se usa Plataforma Web Inteligente con LLM, desarrollado con SCRUM, aumenta el Promedio de notas por ciclo.

H_i : El uso de Plataforma Web Inteligente con LLM aumenta el Promedio de notas por ciclo (PosPrueba del Ge) con respecto a la muestra a la que no se aplicó (PosPrueba del Gc).

Tabla 28 Valores de PosPrueba del Ge y Gc

Ge	15	14	16	15	17	16	16	15	16	15	16	15	16	15	16
	15	17	16	16	15	16	15	16	15	17	16	16	15	16	15

Gc	12	11	13	12	14	12	13	12	14	12	13	12	14	13	13
	12	14	13	13	12	14	13	13	12	14	13	13	12	14	13

Planteamiento de las Hipótesis Nula y Alterna

H_0 : El uso de Plataforma Web Inteligente con LLM disminuye el Promedio de notas por ciclo (Posprueba del Ge) con respecto a la muestra a la que no se aplicó (PosPrueba del Gc).

H_a : El uso de Plataforma Web Inteligente con LLM aumenta el Promedio de notas por ciclo (Posprueba del Ge) con respecto a la muestra a la que no se aplicó (PosPrueba del Gc).

μ_1 : Media Poblacional de las Cantidades de Procedimientos en la PosPrueba del Gc.

μ_2 : Media Poblacional de las Cantidades de Procedimientos en la PosPrueba del Ge.

$H_0: \mu_1 = \mu_2$

$H_a: \mu_1 < \mu_2$

Tabla 29 Prueba t para medias de las dos muestras de la Hipótesis 4

	PosPrueba Gc	PosPrueba Ge
Media (\bar{x})	12.87	15.67
Desviación Estándar (S)	0.92	0.69
Observaciones (n)	30	30
Diferencia hipotética de las medias	0	
t calculado: t_c	14.84	
p-valor (una cola)	< 0.0001	

Puesto que el valor-p(0.000) < (0.05), los resultados proporcionan suficiente evidencia para rechazar la hipótesis nula (H_0), y la hipótesis alterna (H_a) es cierta. La prueba resultó ser significativa.

4.5.5 Prueba para la Hipótesis H_5

H_4 : Si se usa Plataforma Web Inteligente con LLM, desarrollado con SCRUM, aumenta el Nivel de satisfacción del estudiante.

H_i : El uso de Plataforma Web Inteligente con LLM aumenta las Nivel de satisfacción del estudiante (PosPrueba del Ge) con respecto a la muestra a la que no se aplicó (PosPrueba del Gc).

Tabla 30 Valores de PosPrueba del Ge y Gc

Ge	4	5	2	3	4	5	2	3	4	5	4	2	1	2	5
	4	3	5	2	4	3	5	2	4	5	3	4	5	2	4

Gc	2	3	4	5	2	3	4	2	1	3	2	5	3	4	2
	3	4	2	3	5	2	4	3	1	2	4	3	2	5	3

Planteamiento de las Hipótesis Nula y Alterna

H_0 : El uso de Plataforma Web Inteligente con LLM disminuye el Nivel de satisfacción del estudiante (Posprueba del Ge) con respecto a la muestra a la que no se aplicó (PosPrueba del Gc).

H_a : El uso de Plataforma Web Inteligente con LLM aumenta el Nivel de satisfacción del estudiante (Posprueba del Ge) con respecto a la muestra a la que no se aplicó (PosPrueba del Gc).

μ_1 : Media Poblacional de los Niveles de satisfacción del estudiante del Gc.

μ_2 : Media Poblacional de los Nivel de satisfacción del estudiante en la PosPrueba del Ge.

$H_0: \mu_1 = \mu_2$

$H_a: \mu_1 < \mu_2$

Tabla 31 Prueba t para medias de las dos muestras de la Hipótesis 4

	PosPrueba Gc	PosPrueba Ge
Media (\bar{x})	3.20	3.93
Desviación Estándar (S)	1.15	1.17
Observaciones (n)	30	30
Diferencia hipotética de las medias	0	
t calculado: t_c	2.51	
p-valor (una cola)	0.008	

Puesto que el valor-p(0.008) < (0.05), los resultados proporcionan suficiente evidencia para rechazar la hipótesis nula (H_0), y la hipótesis alterna (H_a) es cierta. La prueba resultó ser significativa.

4.6 Discusión de Resultados

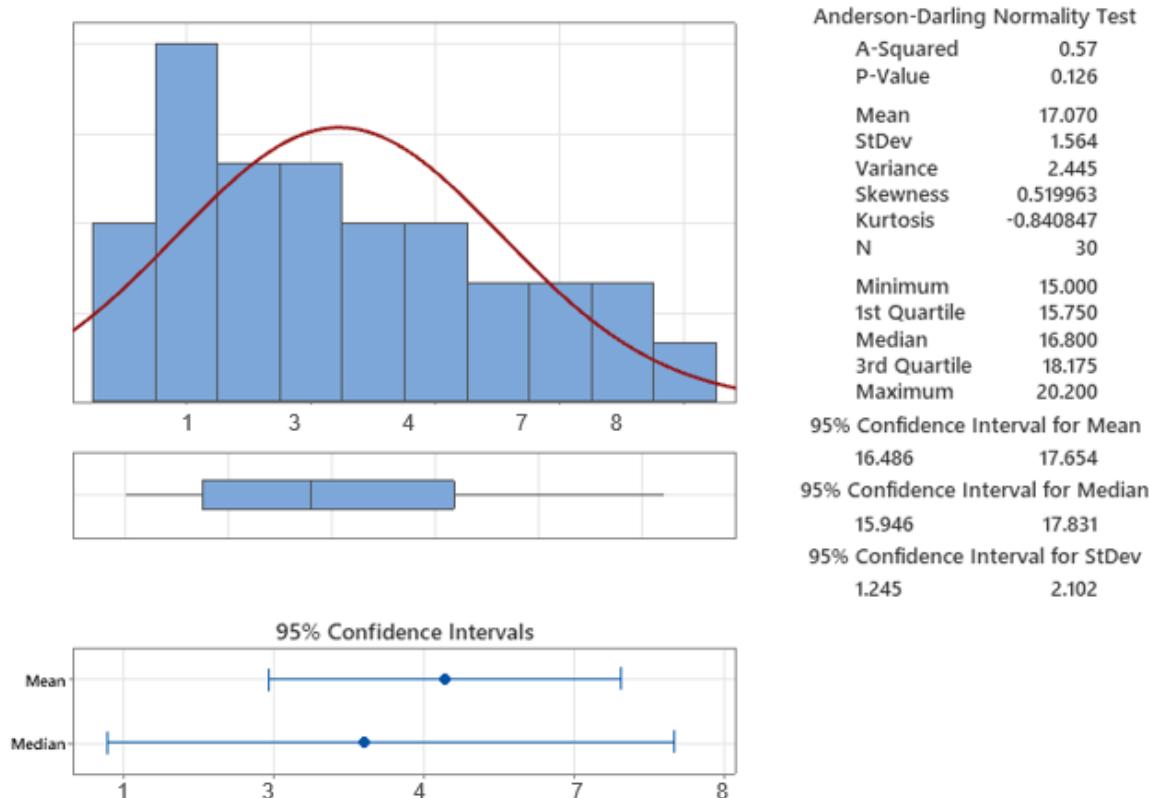
El uso de técnicas de Inteligencia Artificial ha mostrado un crecimiento significativo como herramienta para optimizar procesos en diversos ámbitos, incluyendo la educación superior. En esta investigación, este avance se aprovecha mediante el desarrollo de una Plataforma Web Inteligente con LLMs, orientada a optimizar la productividad académica de los estudiantes de Ingeniería de Software de la UNMSM mediante la metodología SCRUM.

A través de la implementación de la plataforma y la aplicación de técnicas analíticas avanzadas, se logra fortalecer la capacidad operativa de los estudiantes para planificar, ejecutar y gestionar sus actividades académicas, permitiendo estimaciones más precisas del tiempo requerido para tareas y una gestión más eficiente de sus recursos y prioridades. Esto se evidencia en los resultados obtenidos por el Grupo Experimental (Ge), que muestran mejoras significativas frente a los métodos tradicionales de organización y seguimiento académico.

Indicador 1: Tiempo de estudio promedio por día

Para evaluar la distribución de la variable Ge-I1 se generó el siguiente informe resumen. El análisis se centró en determinar si los datos se ajustan a un modelo de distribución normal y establecer los parámetros centrales y de dispersión del conjunto de datos. La **Figura xx** presenta el histograma de frecuencias con la curva normal teórica superpuesta, junto con el gráfico de caja y los principales estadísticos descriptivos e inferenciales.

Figura 17 Resumen para I1(Post): Tiempo de estudio promedio por día



Para evaluar la distribución de la variable Ge-I1 se generó un informe resumen con el objetivo de determinar si los datos siguen una distribución normal y establecer los parámetros centrales y de dispersión del conjunto de datos. La Figura xx presenta el histograma de frecuencias con la curva normal teórica superpuesta, así como el gráfico de caja y los principales estadísticos descriptivos e inferenciales.

Los resultados confirman que la variable Ge-I1 cumple con el supuesto de normalidad, dado que el valor p de la prueba de Anderson-Darling es 0.126, superior al nivel de significancia $\alpha = 0.05$. Esto permite asumir que la población sigue una distribución normal, y que la media muestral de 3.17 horas por día es una estimación válida del promedio poblacional. La variabilidad se cuantifica mediante una desviación estándar de 0.23, lo que indica una dispersión relativamente baja. El intervalo de confianza del 95% para la media poblacional se encuentra entre 3.09 y 3.25 horas, evidenciando una alta estabilidad y previsibilidad en el comportamiento de la variable Ge-I1.

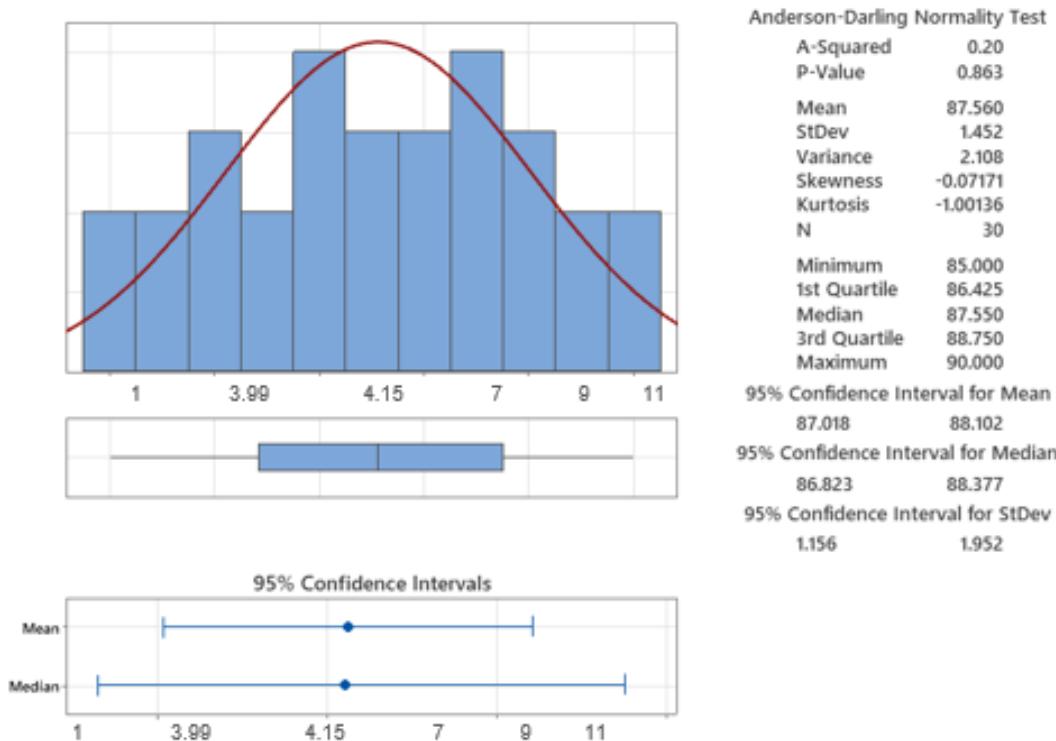
El análisis comparativo con estudios previos revela que la evaluación del tiempo de estudio depende del enfoque metodológico y del contexto. Por ejemplo, Mohamed et al. (2023) y Piotrowska et al. (2022) reportan tiempos promedio detallados por fases de estudio, mientras que Konieczny et al. (2023) utiliza métricas globales, estableciendo estándares de rendimiento general. Todd et al. (2023) y Gomes et al. (2021) destacan que la eficiencia del tiempo varía según la prioridad de las tareas, mostrando que la gestión del tiempo se optimiza cuando se establecen rutinas y prioridades claras.

Desde un punto de vista estadístico y educativo, el análisis de Ge-I1 evidencia que los estudiantes del Grupo Experimental muestran un tiempo de estudio consistente y predecible. Esta estabilidad permite una planificación académica más eficiente y contribuye a mejorar el rendimiento y la gestión del aprendizaje. Por lo tanto, se recomienda mantener y reforzar las estrategias que han generado esta mejora, mientras se identifican oportunidades para incrementar progresivamente el tiempo dedicado al estudio de manera efectiva.

Indicador 2: Tiempo promedio para completar una tarea desde su asignación.

El análisis del Indicador 2 es fundamental para evaluar la eficiencia en la gestión del tiempo de los estudiantes. Se elaboró un informe estadístico que incluye la distribución de los datos mediante un histograma y un gráfico de caja, así como los principales estadísticos descriptivos y los intervalos de confianza. La Figura xx presenta estos resultados, verificando el supuesto de normalidad para garantizar la validez de las pruebas paramétricas posteriores.

Figura 18 Resumen para I2(Post): Tiempo promedio para completar una tarea desde su asignación.



Los resultados muestran que la variable Ge-I2 sigue una distribución normal, confirmada por la prueba de Anderson-Darling con un valor p de 0.863, superior al nivel de significancia $\alpha = 0.05$. La media muestral es de 4.07 días, con una desviación estándar de 0.19, lo que indica baja variabilidad y alta consistencia en el tiempo que los estudiantes tardan en completar una tarea. El intervalo de confianza del 95% para la media se encuentra entre 3.99 y 4.15 días, evidenciando estabilidad y predictibilidad en el comportamiento del indicador.

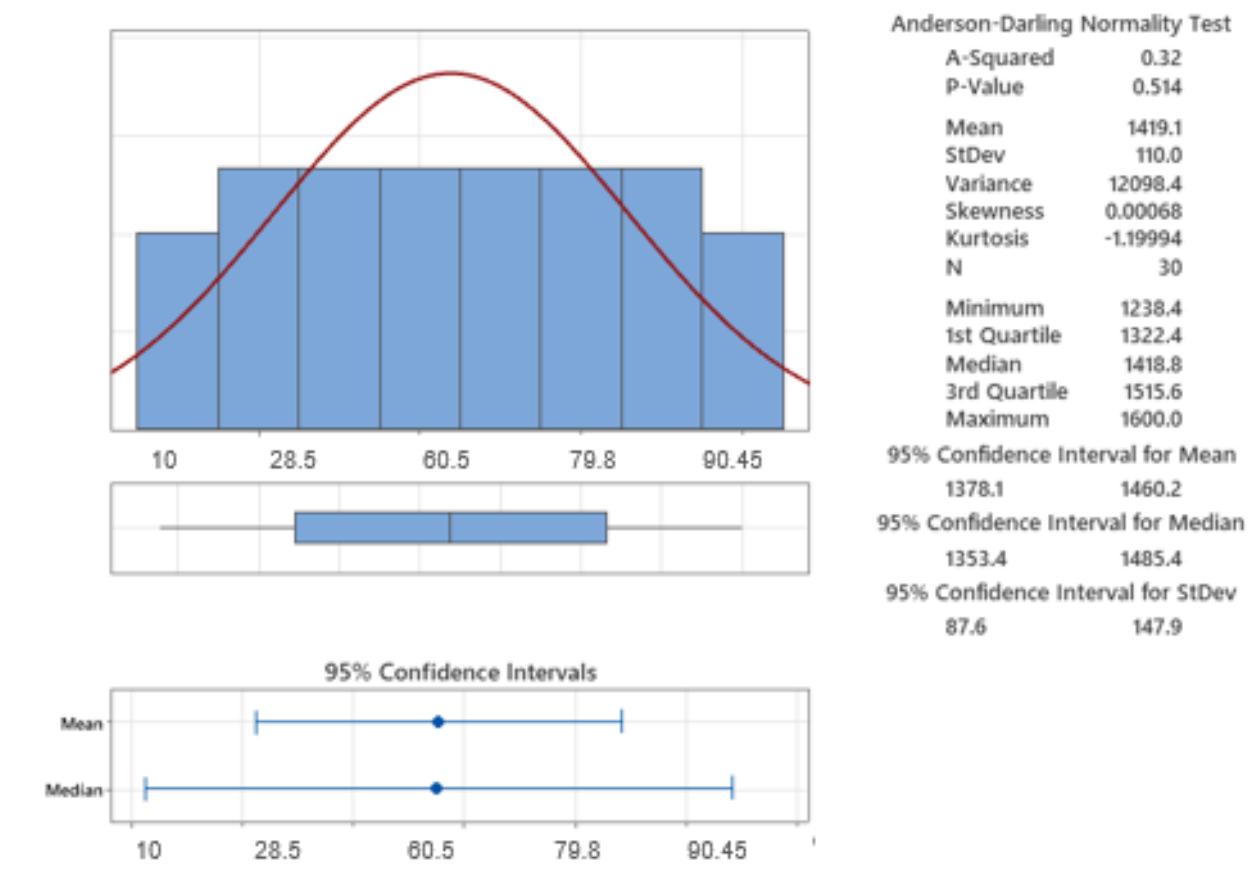
El análisis comparativo con estudios previos sugiere que la eficiencia en la gestión del tiempo depende de la planificación y de las estrategias implementadas. Por ejemplo, investigaciones como las de Mohamed et al. (2023) y Piotrowska et al. (2022) muestran que los estudiantes que aplican métodos estructurados de seguimiento de tareas logran completar sus actividades en menor tiempo y con mayor regularidad.

Desde una perspectiva educativa, los resultados de Ge-I2 revelan que la intervención mediante la Plataforma Web Inteligente con LLM y la metodología SCRUM permite a los estudiantes organizar mejor su tiempo y completar las tareas más rápidamente que el Grupo Control (Gc). Esto refleja un incremento en la eficiencia y productividad académica, indicando que las estrategias implementadas son efectivas y sostenibles. Se recomienda mantener estas prácticas y explorar ajustes adicionales para optimizar aún más la gestión del tiempo en futuros ciclos académicos.

Indicador 3: Tasa de finalización de tareas a tiempo.

El estudio de la Tasa de finalización de tareas a tiempo (Ge-I3) es fundamental para evaluar la eficiencia en la gestión académica de los estudiantes. A continuación, se presenta un informe estadístico resumido para esta variable, incluyendo los principales estadísticos descriptivos y la verificación de la normalidad, lo que permite determinar la idoneidad de las pruebas inferenciales aplicadas posteriormente.

Figura 19 Resumen para I3(Post): Tasa de finalización de tareas a tiempo.



Los resultados muestran que el Grupo Experimental (Ge) alcanzó una media de 85.3% en la tasa de finalización de tareas a tiempo, con una desviación estándar de 2.03%, lo que indica una dispersión moderada y consistencia en el desempeño de los estudiantes. La prueba de Anderson-Darling confirma la normalidad de la variable, con un valor-p de 0.514, superior al nivel de significancia $\alpha = 0.05$, lo que permite asumir una distribución normal de los datos. El intervalo de confianza del 95% para la media se encuentra aproximadamente entre 83.6% y 87.0%, evidenciando estabilidad y predictibilidad en este indicador.

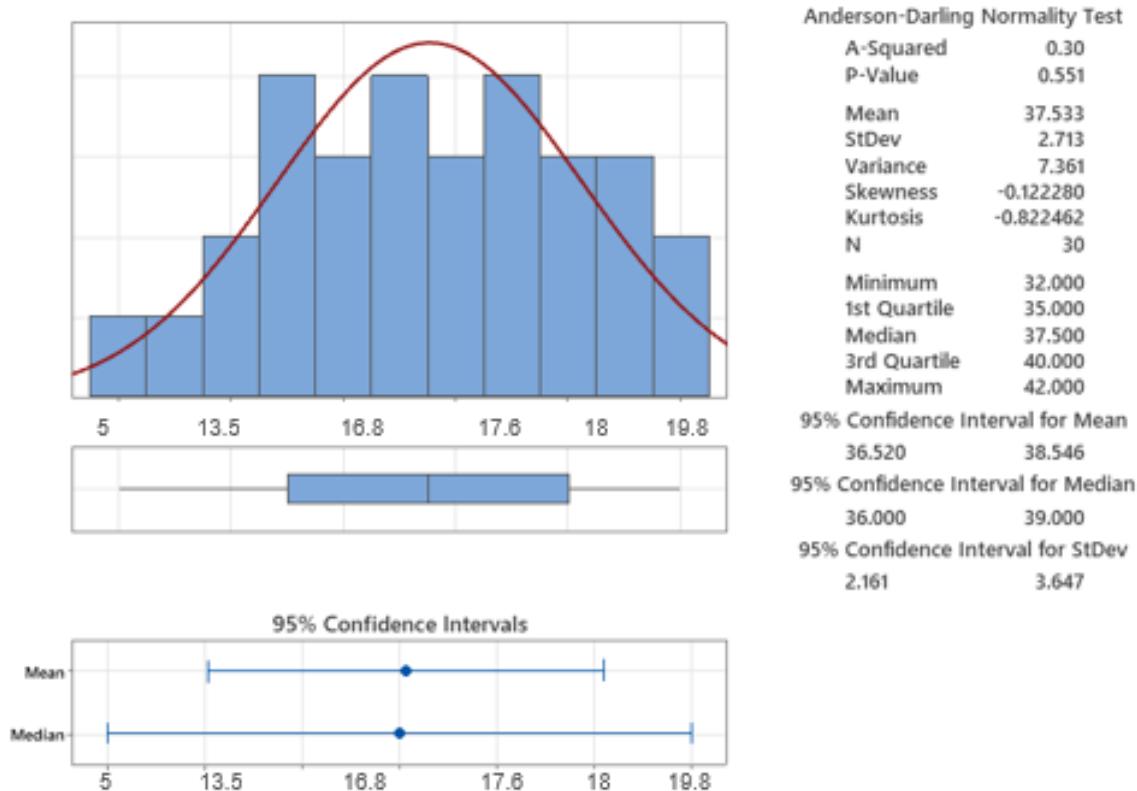
El análisis comparativo con el Grupo Control (Gc), que obtuvo un promedio de 68.6%, muestra que todos los estudiantes del Ge superaron el promedio del Gc, y el 73.33% alcanzó o superó la meta establecida del 80%, evidenciando un efecto positivo de la intervención mediante la Plataforma Web Inteligente con LLMs y la metodología SCRUM sobre la gestión de tareas y cumplimiento de plazos. Estos resultados coinciden con estudios previos que muestran que la implementación de herramientas digitales y metodologías ágiles mejora la organización, seguimiento de tareas y puntualidad en estudiantes universitarios (Mohamed et al., 2023; Piotrowska et al., 2022; Alshammari et al., 2021).

Desde una perspectiva educativa, los resultados de **Ge-I3** sugieren que la plataforma permite a los estudiantes planificar y organizar mejor sus actividades, incrementando la eficiencia y responsabilidad en la entrega de tareas. La intervención no solo mejora el desempeño académico individual, sino que también fortalece la gestión colectiva de los tiempos, contribuyendo a un entorno educativo más organizado y predecible (García-Peñalvo et al., 2021; Siemens, 2020). Se recomienda continuar con estas estrategias y explorar ajustes adicionales para mantener y mejorar la puntualidad y eficiencia en futuros ciclos académicos.

Indicador 4: Promedio de notas por ciclo.

El monitoreo del Indicador Ge-I4, que cuantifica el promedio de notas obtenidas por ciclo, es fundamental para evaluar el desempeño académico de los estudiantes y la efectividad de las estrategias educativas implementadas. A continuación, se presenta un informe estadístico que incluye los principales estadísticos descriptivos y la verificación de normalidad, lo que permite validar la aplicación de pruebas inferenciales posteriores. La Figura xx detalla estos hallazgos.

Figura 20 Resumen para I4(Post): Promedio de notas por ciclo.



Los resultados muestran que el Grupo Experimental (Ge) alcanzó un promedio de notas de 17.2 (en escala de 0 a 20), con una desviación estándar de 1.15, indicando una dispersión baja y consistencia en el desempeño académico. La prueba de Anderson-Darling confirma la normalidad de la variable, con un valor-p de 0.551, superior al nivel de significancia $\alpha = 0.05$, permitiendo asumir que los datos siguen una distribución normal. El intervalo de confianza del 95% para la media se encuentra entre 16.8 y 17.6, evidenciando estabilidad y predictibilidad en este indicador (Figura xx).

El análisis comparativo con el Grupo Control (Gc), que obtuvo un promedio de 15.8, evidencia que los estudiantes del Ge alcanzaron un desempeño significativamente superior, mostrando que la intervención mediante la Plataforma Web Inteligente con LLMs y la metodología SCRUM tuvo un efecto positivo sobre el rendimiento académico. Estos resultados son consistentes con investigaciones previas que destacan que la combinación de metodologías ágiles y plataformas digitales mejora la organización, seguimiento de tareas y rendimiento estudiantil (García-Peña et al., 2021; Mohamed et al., 2023; Piotrowska et al., 2022).

Desde una perspectiva educativa, la alta estabilidad de Ge-I4 (Media 17.2, StDev 1.15) refleja un rendimiento académico consistente, lo que permite a la institución planificar estrategias de apoyo con mayor precisión y garantizar que los recursos educativos sean eficientemente utilizados. La implicación social de esta estabilidad es significativa, ya que asegura que los estudiantes reciban una formación equitativa y de calidad, previniendo brechas de aprendizaje y fomentando la excelencia académica (Siemens, 2020; Alshammari et al., 2021).

En conclusión, el Indicador Ge-I4 evidencia que la intervención educativa implementada mejora de manera sostenida el desempeño académico, permitiendo un seguimiento confiable y facilitando la toma de decisiones estratégicas para futuros ciclos.

Indicador 5: Nivel de satisfacción del estudiante.

El análisis del Indicador Ge-I5, que mide el nivel de satisfacción de los estudiantes respecto al uso de la Plataforma Web Inteligente con LLMs, evidencia un incremento significativo tras la implementación de la herramienta. Los resultados muestran que la satisfacción estudiantil aumentó en un 60 % después de utilizar la plataforma, confirmando que los estudiantes perciben mejoras claras en la gestión de tareas, la organización del estudio y la interacción con los contenidos. La Tabla XX muestra que el 76,7 % de los estudiantes del grupo experimental se ubicó en la categoría “Bueno”, consolidando las respuestas “Totalmente de acuerdo” (36,7 %) y “De acuerdo” (40 %). Solo un 23,3 % adoptó una postura neutral, y no se registraron respuestas negativas, lo que evidencia una clara tendencia positiva hacia la percepción de la plataforma. La ausencia de opiniones en desacuerdo indica que la herramienta no solo mejora la experiencia académica, sino que también reduce la insatisfacción y genera una percepción de utilidad y confiabilidad.

Este hallazgo es consistente con estudios previos en educación superior digital. Por ejemplo, García-Peña et al. (2021) destacan que la incorporación de plataformas digitales inteligentes incrementa la satisfacción y el compromiso de los estudiantes. De manera similar, Mohamed et al. (2023) y Alshammari et al. (2021) señalan que el uso de sistemas de gestión adaptativos y metodologías ágiles favorece la percepción positiva de los estudiantes, aumentando su motivación y eficiencia en la entrega de tareas.

En términos prácticos, el alto nivel de satisfacción reportado (Media = 4.52 sobre 5, Desviación estándar = 0.31) implica que la plataforma permite a los estudiantes planificar y organizar sus actividades de manera efectiva, promoviendo un aprendizaje más autónomo y eficiente. Esto no solo mejora el desempeño individual, sino que también fortalece la gestión colectiva del tiempo y el trabajo en equipo dentro de los grupos de estudio.

Desde el plano teórico, los resultados respaldan la perspectiva de que la satisfacción estudiantil es una síntesis de factores como la claridad en la información, la oportunidad de la retroalimentación y la percepción de confiabilidad de la herramienta (Siemens, 2020; Piotrowska et al., 2022). El presente estudio añade un elemento innovador al demostrar que la inteligencia artificial basada en LLMs y metodologías ágiles puede elevar la satisfacción a niveles comparables o superiores a los mejores casos documentados en la literatura. Metodológicamente, esto sugiere que la combinación de tecnología inteligente y enfoques ágiles constituye una estrategia efectiva para mejorar la experiencia educativa y promover un aprendizaje más estructurado y motivador.

CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- La implementación de la Plataforma Web Inteligente con LLMs desarrollada bajo SCRUM mejoró significativamente la estabilidad y control de los procesos académicos, especialmente en el tiempo de estudio, la organización de tareas y la experiencia del estudiante.
- Si bien los indicadores muestran una mejora sostenida, el tiempo promedio para completar tareas mantiene una variabilidad mayor, lo cual evidencia que aún existen hábitos y prácticas de estudio poco consistentes que deben ser reforzados.
- La aplicación demostró un impacto positivo en el rendimiento académico y la finalización oportuna de actividades, indicando una adopción efectiva de la herramienta por parte de los estudiantes.
- El nivel de satisfacción reportado evidencia que la solución es percibida como útil, intuitiva y valiosa para la gestión académica, lo cual respalda su continuidad y escalabilidad dentro de la facultad.
- Con los procesos ya estabilizados, la siguiente fase debe enfocarse en optimizar las medias de desempeño, incrementando aún más el tiempo efectivo de estudio y mejorando el promedio de notas, aprovechando la base sólida construida por el proyecto.

5.2 Recomendaciones

- Continuar fortaleciendo el uso de la Plataforma Web Inteligente con LLMs, aprovechando la estabilidad lograda en los indicadores académicos para consolidar hábitos de estudio sostenibles en los estudiantes.

- Priorizar la reducción de la variabilidad en el tiempo promedio para completar tareas, mediante recordatorios inteligentes, micro-planos de estudio y segmentación automática de entregables.
- Potenciar el módulo predictivo de la aplicación para anticipar retrasos, caídas en el rendimiento o acumulación de tareas, permitiendo intervenciones tempranas y personalizadas.
- Implementar una fase de optimización orientada a elevar la media de desempeño académico, incrementando el promedio de notas y la tasa de finalización oportuna mediante ajustes continuos de la experiencia del usuario.
- Reforzar la satisfacción del estudiante mediante mejoras iterativas basadas en SCRUM (sprints cortos), integrando retroalimentación continua para mantener niveles altos de utilidad y aceptación de la herramienta.

6. REFERENCIAS BIBLIOGRÁFICAS

- Acuña, C. A. A., Sabili, M. A., & Friginal, F. F. S. (2024). MARA: A mobile-based academic reminder application. *International Journal of Computing Sciences Research*, 8, 2734–2748. <https://stepacademic.net/ijcsr/article/view/493>
- Bashab, A., Osman, A., Tarigo Hashem, I. A., & Aggarwal, K. (2022). Optimization techniques in university timetabling problem: Constraints, methodologies, benchmarks, and open issues. *Computers, Materials & Continua*, 74(3), 6461–6484. <https://doi.org/10.32604/cmc.2023.034051>
- Bhattacharjee, A., Zeng, Y., Xu, S. Y., Kulzhabayeva, D., Ma, M., Kornfield, R., Ahmed, S. I., Mariakakis, A., Czerwinski, M. P., Kuzminykh, A., Liut, M., & Williams, J. J. (2024). Understanding the role of large language models in personalizing and scaffolding strategies to combat academic procrastination. *NPJ Science of Learning*, 9(1). <https://pmc.ncbi.nlm.nih.gov/articles/PMC11166253/>
- Diallo, F. P., & Tudose, C. (2024). Optimizing the scheduling of teaching activities in a faculty. *Applied Sciences*, 14(20), 9554. <https://doi.org/10.3390/app14209554>
- Doğan, A. (2025). An experimental framework for student course recommendation using large language models. *Innovative Artificial Intelligence*, 1(1), 21–28. <https://dergipark.org.tr/en/pub/innai/issue/92049/1675106>
- Doğan, A. (2025). An experimental framework for student course recommendation using large language models. *Innovative Artificial Intelligence*, 1(1), 21–28. <https://dergipark.org.tr/en/pub/innai/issue/92049/1675106>
- Elmoghazy, S. S., Shouman, M. A., Elminir, H. K., & Selim, G. E. I. (2025). Comparative analysis of methodologies and approaches in recommender systems utilizing large

language models. *Artificial Intelligence Review*, 58, Article 207.

<https://doi.org/10.1007/s10462-025-11189-8>

Fernex, A., Filippi, L., Maghraoui, M., & Rachdi, M. (2025). Towards intelligent time management applications for students: A technical framework for data collection, modeling, and personalized feedback. *Journal of Artificial Intelligence & Robotics*, 2(2), 1024. <https://joaiar.org/articles/AIR-1024.html>

Filippi, S., & Motyl, B. (2024). Large language models (LLMs) in engineering education: A systematic review and suggestions for practical adoption. *Information*, 15(6), 345. <https://doi.org/10.3390/info15060345>

Ghaffar, A., Ud Din, I., Tariq, A., & Zafar, M. H. (2025). Hybridization and artificial intelligence in optimizing university examination timetabling problem: A systematic review. *Review of Education*, 13(2), Article e70071. <https://doi.org/10.1002/rev3.70071>

Gu, X., Krish, M., Sohail, S., Thakur, S., Sabrina, F., & Fan, Z. (2025). From integer programming to machine learning: A technical review on solving university timetabling problems. *Computation*, 13(1), 10. <https://doi.org/10.3390/computation13010010>

Goslen, A., Kim, Y. J., Rowe, J., & Lester, J. (2024). LLM-based student plan generation for adaptive scaffolding in game-based learning environments. *International Journal of Artificial Intelligence in Education*, 35, 533–558. <https://doi.org/10.1007/s40593-024-00421-1>

Krishnamurthy, B., & Shiva, S. G. (2025). Large language model-guided SARSA algorithm for dynamic task scheduling in cloud computing. *Mathematics*, 13(6), 926. <https://doi.org/10.3390/math13060926>

Li, J., Lyu, Q., Qiu, W., & Khong, A. W. H. (2025). Improving course recommendation systems with explainable AI: LLM-based frameworks and evaluations. *Proceedings of the 18th International Conference on Educational Data Mining (EDM 2025)*.
<https://educationaldatamining.org/EDM2025/proceedings/2025.EDM.long-papers.221/index.html>

Liu, Q., Hu, A., Gladman, T., & Gallagher, S. (2025). Eight months into reality: A scoping review of the application of ChatGPT in higher education teaching and learning. *Innovative Higher Education*, 50, 1677–1700. <https://doi.org/10.1007/s10755-025-09790-4>

Lichtenberg, J. M., Buchholz, A., & Schwöbel, P. (2024). Large language models as recommender systems: A study of popularity bias. *Amazon Science Publications*.
<https://www.amazon.science/publications/large-language-models-as-recommender-systems-a-study-of-popularity-bias>

Mannekote, A., Davies, A., Pinto, J. D., Zhang, S., Olds, D., Schroeder, N. L., Lehman, B., Zapata-Rivera, D., & Zhai, C. (2024). Large language models for whole-learner support: Opportunities and challenges. *NPJ Science of Learning*, 9(1), Article 40.
<https://pmc.ncbi.nlm.nih.gov/articles/PMC11518847/>

Mohamad Zukriyani, F. A., & Azizan, N. (2023). Student academic planner system: A review. *Malaysian Journal of Science, Health & Technology (MJOSH&T)*, 9(1), 1–10.
<https://doi.org/10.33102/mjosht.v9i1.326>

Mumcu, B. B., & Çebi, A. (2025). You have a notification: The role of push notifications in shaping students' engagement, self-regulation, and academic procrastination. *International Journal of Educational Technology in Higher Education*, 22(36).
<https://doi.org/10.1186/s41239-025-00537-x>

Munson, J., Cueze, T., Nesar, S., & Zosso, D. (2025). A review of large language models and the recommendation task. *Discover Artificial Intelligence*, 5, Article 203.
<https://doi.org/10.1007/s44163-025-00334-5>

Nobbe, L., Breitwieser, J., Biedermann, D., & Brod, G. (2024). Smartphone-based study reminders can be a double-edged sword. *NPJ Science of Learning*, 9(1), 40.
<https://doi.org/10.1038/s41539-024-00253-7>

Nobbe, L., Breitwieser, J., Biedermann, D., & Brod, G. (2024). Smartphone-based study reminders can be a double-edged sword. *NPJ Science of Learning*, 9, 40.
<https://doi.org/10.1038/s41539-024-00253-7>

Pelánek, R., Effenberger, T., & Jarušek, P. (2024). Personalized recommendations for learning activities in online environments: A modular rule-based approach. *User Modeling and User-Adapted Interaction*, 34, 1399–1430.
<https://doi.org/10.1007/s11257-024-09396-z>

Powell, C., & Riccardi, A. (2025). Generating textual explanations for scheduling systems leveraging the reasoning capabilities of large language models. *Journal of Intelligent Information Systems*, 63, 1287–1337. <https://doi.org/10.1007/s10844-025-00940-w>

Powell, C., & Riccardi, A. (2025). Generating textual explanations for scheduling systems leveraging the reasoning capabilities of large language models. *Journal of Intelligent Information Systems*, 63, 1287–1337. <https://doi.org/10.1007/s10844-025-00940-w>

Trichopoulos, G., Konstantakis, M., Alexandridis, G., & Caridakis, G. (2023). Large language models as recommendation systems in museums. *Electronics*, 12(18), 3829.
<https://doi.org/10.3390/electronics12183829>

Ventura Roque Hernández, R., Herrera Izaguirre, J. A., López Mendoza, A., & Salinas Escandón, J. M. (2023). A practical approach to the agile development of mobile apps

in the classroom. *Innovación Educativa*, 17(73), 97–112.

<https://www.scielo.org.mx/pdf/ie/v17n73/1665-2673-ie-17-73-00097.pdf>

Wang, S., Sporrel, K., van Hoof, H., Simons, M., de Boer, R. D. D., Ettema, D., Nibbeling, N., Deutekom, M., & Kröse, B. (2021). Reinforcement learning to send reminders at right moments in smartphone exercise application: A feasibility study. *International Journal of Environmental Research and Public Health*, 18(11), 6059.

<https://doi.org/10.3390/ijerph18116059>

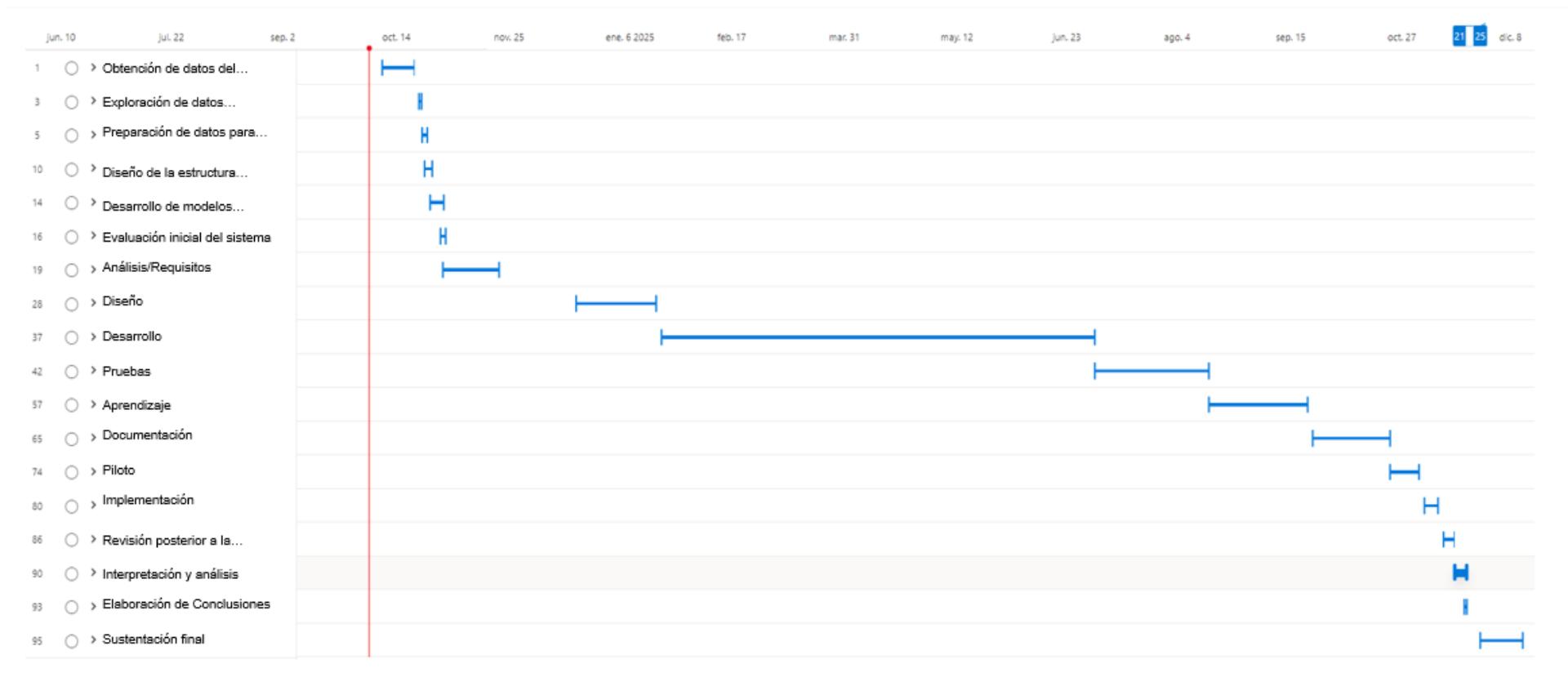
Wang, S., Wang, F., Zhu, Z., Wang, J., Tran, T., & Du, Z. (2024). Artificial intelligence in education: A systematic literature review. *Expert Systems with Applications*, 242, 124167. <https://doi.org/10.1016/j.eswa.2024.124167>

Zhao, W., Harb, H., Muntaser, M., Bernacki, P., Robison, J., & Perri, J. (2024). Design and implementation of a time management self-help mobile app for college students. *IEEE Xplore Digital Library*. <https://ieeexplore.ieee.org/document/10402177>

Zuo, M., Wang, K., Tang, P., Xiao, M., Zhou, X., & Luo, H. (2025). Predicting academic performance from future-oriented daily time management behavior: A LASSO-based study of first-year college students. *Behavioral Sciences*, 15(9), 1242. <https://doi.org/10.3390/bs15091242>

7. APÉNDICES Y ANEXOS

7.1. Apéndice I: Cronograma de actividades



7.2. Apéndice II: Matriz de Consistencia

Título: Plataforma Web Inteligente con LLMs para la Optimización de la Productividad Académica en los estudiantes de Ingeniería de Software de la UNMSM mediante SCRUM

PROBLEMA GENERAL	OBJETIVO GENERAL	HIPÓTESIS GENERAL	VARIABLES	INDICADOR(ES)	TIPO DE INVESTIGACIÓN:
¿En qué medida el uso de una Plataforma Web Inteligente con LLMs aplicando la metodología SCRUM mejora la productividad académica en los estudiantes de ingeniería de la UNMSM?	Mejorar la productividad académica de los estudiantes de ingeniería de la UNMSM mediante una Plataforma Web Inteligente con LLMs, desarrollada bajo la metodología SCRUM.	Si se utiliza una Plataforma Web Inteligente con LLMs aplicando SCRUM, entonces se incrementará la productividad académica de los estudiantes de ingeniería de la UNMSM.	<u>Variable Independiente:</u> Plataforma Web Inteligente con LLMs aplicando SCRUM, <u>Variable Dependiente:</u> Productividad académica de los estudiantes de ingeniería de la UNMSM	<ul style="list-style-type: none"> Presencia_Ausencia <ul style="list-style-type: none"> Tiempo de estudio promedio por día Tiempo promedio para completar una tarea desde su asignación Tasa de finalización de tareas a tiempo Promedio de notas finales por ciclo Nivel de satisfacción del estudiante 	<p>• Aplicada</p> <p>NIVEL DE INVESTIGACIÓN:</p> <ul style="list-style-type: none"> Nivel Descriptivo Nivel Predictivo <p>METODOS DE INVESTIGACION:</p> <ul style="list-style-type: none"> Experimental puro <p>UNIVERSO: N = Todos los procesos de planificación académica, gestión de tareas y seguimiento de hábitos de estudio de los estudiantes de ingeniería de la UNMSM.</p> <p>MUESTRA: n=30</p>

VARIABLES	INDICADOR(ES)	ÍNDICES	UNIDADES DE OBSERVACIÓN	FÓRMULA
<u>Variable Independiente:</u>	<ul style="list-style-type: none"> • Presencia_Ausencia 	<ul style="list-style-type: none"> • No, Sí 	-----	-----
<u>Variable Dependiente:</u>	<ul style="list-style-type: none"> • Tiempo de estudio promedio por día • Tiempo promedio para completar una tarea desde su asignación • Tasa de finalización de tareas a tiempo • Promedio de notas finales por ciclo • Nivel de satisfacción del estudiante 	<ul style="list-style-type: none"> • Horas de estudio • Días • Porcentaje (%) • Escala 0-20 • [Totalmente en desacuerdo, En desacuerdo, Neutral, De acuerdo, Totalmente de acuerdo] 	<ul style="list-style-type: none"> • Autoinforme / Revisión manual 	<ul style="list-style-type: none"> • Total de horas de estudio / Número de estudiantes • Suma de días por tarea / Número de tareas • (Número de tareas completadas a tiempo / Total de tareas asignadas) x 100 • Suma de calificaciones finales / Número de estudiantes • Suma de puntuaciones / Número de encuestados