



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE INGENIERÍA
CARRERA DE COMPUTACIÓN**

TEMA:

Desarrollo de un prototipo funcional de un agente de IA Generativa especializado en la creación de borradores de contenido editorial universitario.

AUTOR:

**Pérez Mayorga, Jean Carlos
Morocho Macías, Julio César**

**Trabajo de integración curricular previo a la obtención del título de
INGENIERO EN CIENCIAS DE LA COMPUTACIÓN**

TUTOR:

Castañeda Clark, María Del Carmen

**Guayaquil, Ecuador
03 de marzo de 2026**



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE INGENIERÍA
CARRERA DE COMPUTACIÓN

CERTIFICACIÓN

Certificamos que el presente proyecto de trabajo de integración curricular fue realizado en su totalidad por el Sr. **Pérez Mayorga, Jean Carlos** y el Sr. **Morocho Macías, Julio César** como requerimiento para la obtención del título de **INGENIERO EN CIENCIAS DE LA COMPUTACIÓN**.

TUTOR

f. _____

Castañeda Clark, María Del Carmen

Guayaquil, a los 03 días del mes de marzo del año 2026



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE INGENIERÍA
CARRERA DE COMPUTACIÓN
DECLARACIÓN DE RESPONSABILIDAD**

**Nosotros, Pérez Mayorga, Jean Carlos
Morocho Macías, Julio César**

DECLARAMOS QUE:

El trabajo de integración curricular, **Desarrollo de un prototipo funcional de un agente de IA Generativa especializado en la creación de borradores de contenido editorial universitario**, Requerimiento para la obtención del título de **Ingeniero en Ciencias de la Computación** ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Proyecto de T.I. referido.

Guayaquil, a los 03 días del mes de marzo del año 2026

f. _____
Pérez Mayorga, Jean Carlos

f. _____
Morocho Macías, Julio César



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE INGENIERÍA
CARRERA DE COMPUTACIÓN

AUTORIZACIÓN

Nosotros, **Pérez Mayorga, Jean Carlos**
Morocho Macías, Julio César

Autorizo a la Universidad Católica de Santiago de Guayaquil a la **publicación** en la biblioteca de la institución del Proyecto de Trabajo de integración curricular “**Desarrollo de un prototipo funcional de un agente de IA Generativa especializado en la creación de borradores de contenido editorial universitario.**”, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, a los 03 días del mes de marzo del año 2026

LOS AUTORES:

f. _____
Pérez Mayorga, Jean Carlos

f. _____
Morocho Macías, Julio César



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE INGENIERÍA
CARRERA DE COMPUTACIÓN

REPORTE ANTIPLAGIO



CERTIFICADO DE ANÁLISIS
magister

Tesis_Jean_Carlos_Julio_Moroc
docx

4%
Textos
sospechosos

Nombre del documento: Tesis_Jean_Carlos_Julio_Morocdocx.docx
ID del documento: 84a076c1f5802b6f6f5e12f686ba1f9214841a
Tamaño del documento original: 4,5 MB

Depositante: María del Carmen Castañeda



CERTIFICADO DE ANÁLISIS
magister

Tesis_Jean_Carlos_Julio_Moroc
docx

Nombre del documento: Tesis_Jean_Carlos_Julio_Morocdocx.docx
ID del documento: 84a076c1f5802b6f6f5e12f686ba1f9214841a
Tamaño del documento original: 4,5 MB

Depositante: Mar
Fecha de depósito
Tipo de carga: Int
fecha de fin de an

Ubicación de las similitudes en el documento:

Fecha de elaboración: 11/02/2026



Firmado electrónicamente por:
MARIA DEL CARMEN
CASTANEDA CLARK

Maria del Carmen Castañeda Clark

Tutor del Proyecto de T.I.

Carrera de Ingeniería en Ciencias de la Computación

Ubicación de las similitudes en el documento:

Herramienta Recortes

Captura de pantalla copiada en el portapap
Guardado automáticamente en la carpeta d

AGRADECIMIENTO

Agradezco a Dios misericordioso y a la Virgen María, por ser mi guía en este camino estudiantil y por brindarme el apoyo, la paciencia y la dedicación para lograr cumplir el objetivo planteado en mi vida.

A Esthela Mayorga, mi madre, el pilar fundamental para que todo tenga sentido en mi vida, la cual me impulsó a salir adelante ,gracias por darme consejos valiosos que los llevo guardados en mi corazón y que los recuerdo siempre cuando esté dispuesta a tomar alguna decisión difícil, por darme la vida y por brindarme todo el amor necesario para poder cultivar día a día el valor de la perseverancia y demostrarlo las veces que sea necesario para que no me falte nada junto con mis hermanos Daniel y Jennifer , afirmando que todo es posible con esfuerzo y disciplina durante mi trayectoria universitaria, para así poder luchar por mis sueños hasta conseguirlos.

A Carlos Pérez, mi padre, quien desde pequeña me enseñó a ser responsable en mis obligaciones, a ser puntual y a tener una excelente presencia en todo ámbito que se me presente, gracias por el apoyo económico que recibí para poder lograr cumplir esta meta en mi vida, siempre dispuesto a brindarme palabras alentadoras de superación para ser una mujer con principios y valores, es mi fortaleza para no decaer y seguir por el camino del bien, a pesar de ser un hombre de carácter ,me sirvió de mucho esas retadas, esas horas de espera hasta que salga de mi universidad después de culminar mis clases, valoré su esfuerzo y responsabilidad, por esa razón ahora puedo lograr tomar decisiones correctas, es mi ejemplo de lucha y optimismo para que todo sea un éxito total.

Agradezco a toda mi familia por brindarme su confianza y por ofrecerme incondicionalmente su apoyo en los buenos y malos momentos.

Jean Carlos Perez

Agradecimiento

Expreso mi profundo agradecimiento a Dios por guiar mis pasos y permitirme alcanzar este objetivo académico.

A mis padres, por su apoyo constante y por ser el pilar fundamental en mi formación personal y profesional.

A mi esposa e hijo, por su comprensión durante los momentos de dedicación y esfuerzo que este proyecto demandó.

A mi tutora la Ing. María del Carmen Castañeda Clark, por su orientación, conocimientos y acompañamiento continuo en el desarrollo de esta investigación.

A los docentes de la carrera de Ingeniería en Sistemas, por compartir sus conocimientos y contribuir de manera significativa a mi formación académica.

Julio Cesar Morocho Macias

DEDICATORIA

Dedico este presente trabajo a Dios , por darme la sabiduría de seguir adelante, sin perder la fe y dedicación de culminar con éxito un objetivo más que tengo en mi vida.

A mis padres: Esthela Mayorga y Carlos Pérez, por el esfuerzo, apoyo incondicional, consideración, dedicación, perseverancia, para llegar a cumplir este gran paso en ser una profesional de calidad.

A mi hermano Anderson Daniel Pérez Mayorga, quien fue mi ejemplo de vida para poder continuar con mis estudios y por brindarme las mejores experiencias a lo largo de su desarrollo profesional, las cuales me motivaron a sobresalir día a día.

A mi hermana Jennifer Perez quien me demostraba su apoyo y sus consejos me han ayudado a mejorar como persona y en mi vida estudiantil.

Esto va una dedicatoria a mi abuelita para una persona especia quien no podrá compartir este logro pero se que desde arriba estar orgullo de mi

A mi persona especial, por demostrarme su apoyo, cariño y pasión durante esta etapa.

Jean Carlos Perez

Dedicatoria

Dedico el presente trabajo de titulación, previo a la obtención del título de Ingeniero en Sistemas, en primer lugar, a Dios, por brindarme la vida, la sabiduría y la fortaleza necesarias para culminar esta importante etapa de mi formación profesional.

A mis padres, por su amor incondicional, esfuerzo y sacrificio a lo largo de mi vida. Gracias por inculcarme valores, principios y el deseo constante de superación. Este logro también les pertenece.

A mi hijo, Sebastián, quien es mi mayor motivación y la razón que impulsa cada uno de mis esfuerzos. Su presencia en mi vida representa la inspiración diaria para seguir creciendo personal y profesionalmente.

A mi esposa, Verónica, por su amor, comprensión y apoyo incondicional durante todo este proceso académico. Su paciencia, confianza y palabras de aliento fueron fundamentales para culminar con éxito esta meta.

Julio Cesar Morocho Macias



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE INGENIERÍA
CARRERA DE COMPUTACIÓN**

TRIBUNAL DE SUSTENTACIÓN

f. _____

**ING. ANA CAMACHO CORONEL, MGS
DIRECTORA DE CARRERA**

f. _____

**ING. JOSE ERAZO AYON, MGS
DOCENTE DE LA CARRERA**

f. _____

**ING. ROBERTO GARCIA SANCHEZ, MGS
OPONENTE**

ÍNDICE

INDICE DE TABLAS	XIII
ÍNDICE DE FIGURAS	XIV
RESUMEN	XV
ABSTRACT	XVI
INTRODUCCIÓN	2
CAPÍTULO I	3
PLANTEAMIENTO DEL PROBLEMA.....	3
1.1 Formulación del problema	3
1.2 Justificación e importancia	3
1.3 Delimitación	4
1.4 Objetivos	5
1.4.1 Objetivo General	5
1.4.2 Objetivos Específicos	5
1.5 Alcances del problema	5
1.6 Limitaciones:	5
CAPÍTULO II	7
MARCO TEÓRICO.....	7
2.1 Inteligencia Artificial Generativa (IAGen)	7
2.1.1 Modelos de Lenguaje de Gran Escala (LLMs).....	8
2.1.2 Prompt Engineering	9
2.2 IA en Procesos Editoriales	10
2.2.1 Agentes de IA Especializados	11
2.2.2 Interacción Humano-IA.....	12
2.2.3 Integración de APIs de IA.....	13
2.2.4 Sistemas Web Monolíticos	14
2.2.5 Aplicaciones de IAGen en Educación Superior	14
2.2.6 Calidad del contenido generado por IA	15
2.3 Ética y regulación del uso de IA en educación	16
2.3.1 Procesos Editoriales Académicos	16
2.3.2 JavaScript	17
2.3.3 Producto Mínimo Viable (MVP)	18
2.4 React	18
2.4.1 Chakra UI.....	19
2.4.2 Microsoft Azure y su herramienta Foundry	19
2.4.3 Modelo GPT-4. 1.....	20

CAPÍTULO III	21
METODOLOGÍA DE LA INVESTIGACIÓN	21
3.1 Tipo de investigación.....	21
3.2 Población y muestra.....	21
3.2.1 Selección de la muestra	21
3.3 Técnicas e instrumentos para obtención de información	21
3.4 Análisis de resultados de las entrevistas	22
3.4.1 Diseño de la encuesta.....	24
3.4.2 Análisis de los resultados de la encuesta	24
Composición de la muestra	25
Uso de la IA generativa para escribir.....	25
Herramientas utilizadas	26
Tipos de documentos que redactan	26
Aspectos importantes más valorados	27
Nivel de confianza en borradores generados por IA	28
Riesgos percibidos	28
Funciones deseadas para el prototipo.....	29
Utilidad percibida del agente de IA	29
CAPÍTULO IV	31
4.1 Diseño de la solución	31
4.2. Desarrollo del prototipo	33
4.2.1 Modelo GPT-4. 1.....	34
4.2.2 Comparación de herramientas	36
4.3 Implementación detallada de los módulos	39
4.3.1 Módulo de entrada (ModuleInput).....	39
4.3.2. Motor de Prompt Engineering.	41
4.3.3. Cliente de API para conexión con el modelo	43
4.3.4. Módulo de salida (Module Output).....	46
4.3.5. Análisis comparativo del prototipo	48
CONCLUSIONES.....	53
RECOMENDACIONES	54
REFERENCIAS BIBLIOGRAFICAS	55
ANEXOS	58
Anexo 1 Modelo del Prototipo.....	58

INDICE DE TABLAS

Tabla 1	Población y muestra.....	21
Tabla 2	React Vs Angular.....	36
Tabla 3	Interfaz de usuario de Chakra Vs CSS puro	37
Tabla 4	JavaScript frente a Python	37
Tabla 5	Ingeniería rápida avanzada versus ingeniería rápida básica	38
Tabla 6	Microsoft Azure Vs DigitalOcean	38
Tabla 7	Resumen de Diferenciadores Técnicos	51
Tabla 8	Tabla por Token.....	52

ÍNDICE DE FIGURAS

Figura 1	Representación conceptual de la Inteligencia Artificial Generativa.....	8
Figura 2	Proceso de Prompt Engineering en la interacción humano–modelo.....	10
Figura 3	Aplicación de la inteligencia artificial en los procesos editoriales.	11
Figura 4	Interacción entre el ser humano y la inteligencia artificial.	13
Figura 5	Procesos editoriales académicos apoyados por inteligencia artificial.....	17
Figura 6	<i>Gráfico estadístico de muestra por carrera</i>	25
Figura 7	<i>¿Han utilizado IA para redactar textos académicos?</i>	26
Figura 8	<i>Herramientas de IA más utilizadas (top10)</i>	26
Figura 9	<i>Tipos de documentos redactados con mayor frecuencia</i>	27
Figura 10	<i>Aspectos más valorados en un borrador generado por IA</i>	27
Figura 11	<i>Nivel de confianza en un borrador generado por IA (Media=3.08)</i>	28
Figura 12	Riesgos percibidos en el uso de la IA para redactar	29
Figura 13	<i>Funciones deseadas para el prototipo</i>	29
Figura 14	<i>Utilidad percibida del agente de IA (Media=3.78)</i>	30
Figura 15	Arquitectura del prototipo	32
Figura 16	<i>Costo aproximado por millón de tokens del modelo gpt-4.1</i>	36
Figura 17	Módulo de Entrada (ModuleInput) en arquitectura de sistema.	39
Figura 18	Estados manejados en el formulario de ingreso.....	40
Figura 19	<i>Función que extrae texto del archivo</i>	41
Figura 20	Cliente de API para conexión con modelo de inteligencia artificial.	43
Figura 21	Implementación del cliente para conexión con el modelo en Microsoft Foundry.	45
Figura 22	Módulo de Salida (ModuleOutput) en arquitectura de sistema.	46
Figura 23	<i>Módulo de salida. Función de exportación a formato docx</i>	48
Figura 24	Estructura del prototipo.....	58
Figura 25	Llenado los parámetros para generar	58
Figura 26	Se genera la información que se requiere.....	59

RESUMEN

El proyecto propone el desarrollo de un prototipo funcional de un agente de IA Generativa especializado en la creación de borradores de contenido editorial universitario. A diferencia de herramientas generales como los distintos chatbox que existen en internet, tales como ChatGPT, Claude, Deepseek, entre otras, este agente estará contextualizado para las necesidades comunicacionales de unidades académicas.

La solución busca automatizar la construcción de prompts especializados, asegurando coherencia en tono, estilo institucional y eficiencia en la generación de contenido.

El prototipo consistirá en una aplicación web con una interfaz simple donde el usuario ingrese parámetros como tema, tono y público efectivo. El sistema procesará estos datos, generará instrucciones estructuradas y se integrará con la API de un modelo de IA generativa para producir borradores de posts o comunicados. Finalmente, se realizarán pruebas con usuarios para validar la calidad y utilidad del sistema.

Palabras Clave: Inteligencia artificial generativa, agente de IA, contenido editorial universitario, prototipo funcional.

ABSTRACT

This project proposes the development of a functional prototype of a Generative AI agent specialized in creating drafts of university editorial content. Unlike general tools such as the various chatbots available online, such as ChatGPT, Claude, and Deepseek, this agent will be contextualized to the communication needs of academic units.

The solution aims to automate the creation of specialized prompts, ensuring consistency in tone, institutional style, and efficiency in content generation.

The prototype will consist of a web application with a simple interface where the user enters parameters such as topic, tone, and target audience. The system will process this data, generate structured instructions, and integrate with the API of a generative AI model to produce drafts of posts or announcements. Finally, user testing will be conducted to validate the system's quality and usability.

Key words: Generative Artificial Intelligence, AI Agent, University Editorial Content, Functional Prototype,

INTRODUCCIÓN

Las unidades académicas universitarias, es decir, departamentos, facultades, centros de investigación, deben generar continuamente contenido escrito de calidad para sus boletines, sitios web y comunicaciones internas. Este proceso manual consume tiempo y recursos preciosos.

Aunque existen herramientas de inteligencia artificial de propósito general como ChatGPT, su uso directo para necesidades editoriales institucionales tiene limitaciones: falta de especialización, inconsistencia en el tono y el estilo, y la necesidad de repetir instrucciones de contexto para cada solicitud.

Este proyecto propone el desarrollo de un prototipo de agente de IA especializado que actúe como asistente de edición contextualizado.

Su propósito no es la generación propiamente dicha de texto (como lo hacen las herramientas mencionadas anteriormente), sino la rápida automatización y sistematización del proceso de ingeniería para un campo específico: la comunicación universitaria.

El sistema incorpora conocimiento de lo que constituye "buen contenido editorial universitario" en una interfaz parametrizada, lo que garantiza coherencia, ahorra tiempo y reduce la curva de aprendizaje asociada con el uso general de la IA.

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

1.1 Formulación del problema

La dificultad que enfrentan las universidades en la producción de borradores editoriales radica en la inversión de tiempo y recursos humanos. Surge entonces la necesidad de explorar cómo un agente de IA generativa puede apoyar este proceso. Por ello, el problema se formula en la siguiente pregunta: ¿De qué manera se puede diseñar y desarrollar un prototipo funcional de un agente de IA generativa que contribuya a la creación de borradores de contenido editorial universitario, asegurando estándares de calidad y pertinencia académica?

A continuación, se describe el flujo y los componentes principales del prototipo:

1. **Módulo de Entrada de Usuario:** Una interfaz web donde el editor ingresa los parámetros que definirán el post.
2. **Módulo de Procesamiento:** Este es el núcleo lógico del sistema. Tomará los parámetros del usuario y construirá una instrucción estructurada y detallada para el modelo de IA.
3. **Módulo de Integración con el Modelo de IA:** La instrucción estructurada se enviará a la API del modelo de IA generativa. Este módulo se encargará de la comunicación técnica: autenticación, envío de la solicitud y recepción de la respuesta.
4. **Módulo de Salida y Presentación:** La respuesta del modelo de IA se presentará de forma clara en la interfaz web, en un área de texto donde el usuario pueda copiarlo, editarlo o generar una nueva versión.

1.2 Justificación e importancia

Las unidades académicas de una universidad requieren producir constantemente contenido editorial para boletines, comunicados internos, sitios web y redes institucionales. Este proceso, realizado de manera manual, demanda tiempo, coordinación y habilidades

especializadas en redacción, lo que lo convierte en una tarea operativa costosa y susceptible a inconsistencias en estilo y calidad.

Aunque existen herramientas de IA de propósito general, su uso directo presenta limitaciones: falta de especialización en el dominio universitario, necesidad de configurar instrucciones manualmente para cada solicitud y variabilidad en el tono institucional. Por ello, se vuelve necesario contar con un sistema que automatice la creación de instrucciones y garantice resultados coherentes y alineados con la identidad editorial de la universidad.

La propuesta de desarrollar un prototipo de agente de IA especializado se justifica porque optimiza el tiempo, reduce el esfuerzo editorial, y sistematiza la generación de contenido, permitiendo a los equipos académicos concentrarse en las actividades estratégicas y no en procesos repetitivos. Además, aporta un valor añadido sobre las herramientas genéricas al contextualizar la generación de texto en un entorno institucional específico.

1.3 Delimitación

Delimitación temática

Este proyecto se centra exclusivamente en contenidos editoriales de instituciones educativas, excluyendo otros formatos académicos como ensayos, artículo, científico o tesis.

La atención principal se dirige hacia la automatización del diseño de prompts, no hacia la creación de un modelo propio de inteligencia artificial

Delimitación tecnológica

Se empleará únicamente la API del modelo de IA generativa como el motor para generar texto

La implementación del sistema se realizará como una única aplicación web, sin utilizar arquitecturas distribuidas o móviles.

Se realizarán entrevistas y encuestas a editores académicos.

El tiempo estimado en que se desarrollará este proyecto es de 4 meses.

1.4 Objetivos

1.4.1 Objetivo General

- ❖ Diseñar y desarrollar un prototipo funcional de un agente de IA Generativa especializado en la creación de borradores de contenido editorial universitario, que ofrezca un valor agregado sobre las herramientas de IA de propósito general mediante la parametrización contextualizada del proceso de generación.

1.4.2 Objetivos Específicos

- ❖ Identificar las limitaciones del uso directo de chatbots de IA Generativa.
- ❖ Definir las necesidades de contenido de los borradores y los parámetros de entrada más relevantes para el agente IA, mediante el análisis de requerimientos.
- ❖ Implementar el prototipo integrando la API de la IA Generativa, con la lógica de prompt engineering especializado para la creación de los borradores de contenido editorial universitario
- ❖ Diseñar la arquitectura del sistema y una interfaz de usuario simple que permita a un usuario ingresar los parámetros y recibir el post generado.
- ❖ Realizar pruebas de funcionalidad y una evaluación cualitativa con un grupo de usuarios potenciales para validar la utilidad y calidad de los borradores generados.

1.5 Alcances del problema

- ❖ Desarrollo de una aplicación web monolítica y simple.
- ❖ Interfaz de usuario con campos de entrada para:
 - Tema o palabras claves principales.
 - Selección del tono en el que el post estará escrito.
 - Definición del público objetivo
- ❖ Generación de borradores de posts en formato de texto plano.
- ❖ Documentación técnica y del usuario del prototipo

1.6 Limitaciones:

Estas son las restricciones que pueden afectar el alcance:

Limitaciones tecnológicas

- Dependencia total de la API. Si el servicio falla, el sistema no genera contenido.
- No incluye entrenamientos de modelos personalizados.

Limitaciones técnicas del prototipo

- Solo produce texto plano – no genera formatos enriquecidos.
- El sistema no aprende del historial del editor, por lo que no personaliza progresivamente el estilo.

Limitación de contenido

- El agente puede generar borradores, pero no garantiza perfecciones editoriales, por lo que el usuario debe revisar y editar
- La calidad depende del modelo externo, no del prototipo.
- La calidad depende del modelo extremo, no del prototip .

Limitaciones de usabilidades

- Es un prototipo simple, por lo que si diseño no busca cubrir todas las necesidades reales de un departamento editorial profesional.

CAPÍTULO II

MARCO TEÓRICO

2.1 Inteligencia Artificial Generativa (IAGen)

La Inteligencia Artificial Generativa (IAGen) comprende modelos capaces de producir contenido nuevo —texto, imágenes o código— a partir de patrones aprendidos en grandes volúmenes de datos. Este tipo de IA ha crecido rápidamente desde 2020 gracias a arquitecturas profundas y escalables que permiten generar contenido coherente en múltiples dominios (Kambhampati, 2023). A diferencia de modelos tradicionales, la IAGen no solo clasifica o predice, sino que crea, lo que la convierte en una herramienta con alto potencial en tareas editoriales y educativas.

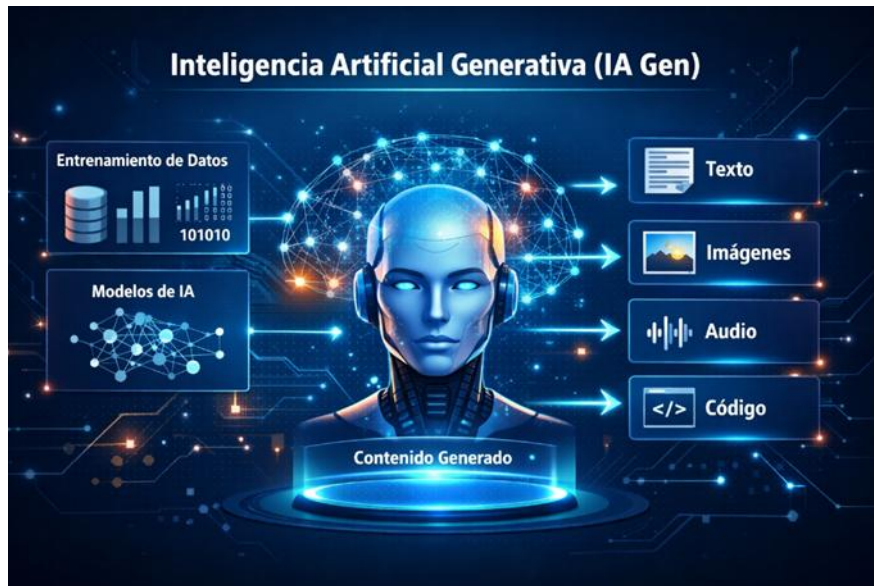
Los avances en deep learning y en entrenamientos masivos han impulsado la calidad de los modelos generativos. El uso de técnicas como Reinforcement Learning from Human Feedback (RLHF) y el ajuste fino ha permitido que los modelos sigan instrucciones con mayor precisión y coherencia contextual (Goodfellow et al., 2016). Esto ha facilitado su adopción en instituciones académicas y organizaciones que requieren automatizar la producción de texto.

En entornos editoriales universitarios, la IAGen permite acelerar la generación de borradores, adaptarse a diferentes audiencias y mantener consistencia comunicacional. Sin embargo, también se deben considerar riesgos como la generación de información inexacta o fabricada, por lo que la supervisión humana sigue siendo crítica (Xu et al., 2023). La literatura enfatiza que la combinación entre IA generativa y control editorial humano produce mejores resultados que cualquiera por separado.

Finalmente, organizaciones internacionales han recomendado lineamientos éticos para el uso responsable de IAGen en sectores sensibles como educación y comunicación institucional. La UNESCO (2021) señala la importancia de la transparencia, la revisión humana y el uso responsable para evitar sesgos o desinformación. Estas recomendaciones son fundamentales para integrar IA en procesos editoriales formales.

Figura 1

Representación conceptual de la Inteligencia Artificial Generativa.



Nota. Imagen generada mediante inteligencia artificial con ChatGPT (OpenAI, 2026).

2.1.1 Modelos de Lenguaje de Gran Escala (LLMs)

Los Modelos de Lenguaje de Gran Escala (LLMs) constituyen la base tecnológica de la mayoría de sistemas de IA generativa modernos. Estos modelos utilizan arquitecturas Transformer, las cuales se caracterizan por su capacidad de procesar secuencias de forma paralela y atender relaciones de largo alcance en el texto (Vaswani et al., 2017). Desde 2020, su escala y precisión han aumentado drásticamente.

La aparición de los llamados foundation models ha ampliado aún más el alcance de los LLMs. Estos modelos generalistas, entrenados con billones de palabras, pueden adaptarse a múltiples tareas mediante prompting o ajuste fino (Bommasani et al., 2021). Esto ha permitido sustituir tareas manuales repetitivas dentro de procesos académicos y administrativos. Sin embargo, su poder también exige controles de seguridad y supervisión humana.

En entornos editoriales, los LLMs permiten generar borradores iniciales coherentes, adaptar la información a público académico y replicar estilos institucionales cuando se usan prompts

bien diseñados. No obstante, su conocimiento está limitado por los datos del entrenamiento, lo que hace necesaria la verificación factual posterior (Mitrović et al., 2023). En comunicaciones académicas, este punto es especialmente relevante para mantener la credibilidad institucional.

Los últimos estudios advierten que los LLMs, aunque poderosos, pueden reproducir sesgos presentes en sus datos de entrenamiento y generar información incorrecta si no se contextualizan correctamente (Floridi & Cowls, 2019). Por ello, su integración en sistemas universitarios requiere políticas, criterios de evaluación y diseño cuidadoso de procesos.

2.1.2 Prompt Engineering

El prompt engineering consiste en diseñar instrucciones estructuradas para orientar a los modelos generativos hacia un resultado preciso. Desde 2021 se ha convertido en un área de investigación clave dado que un mismo modelo puede producir salidas radicalmente distintas dependiendo de cómo se formule el prompt (Liu et al., 2023). Para aplicaciones institucionales, esta práctica es esencial para obtener contenido confiable y consistente.

Dentro del ámbito editorial, el prompt engineering permite controlar el tono, la longitud, la formalidad, el estilo y otros parámetros relacionados con identidad y lineamientos institucionales. Esto facilita que los borradores generados por IA se ajusten a la comunicación universitaria (White et al., 2023). Un buen prompt puede reducir significativamente la necesidad de correcciones humanas posteriores.

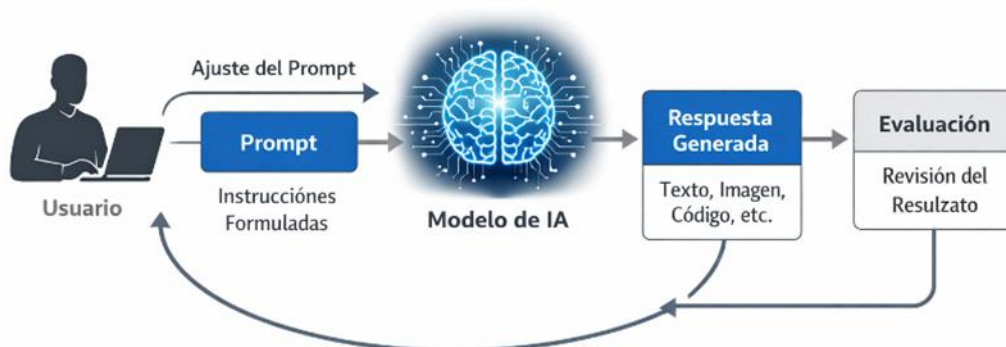
La literatura reciente identifica diferentes técnicas como few-shot prompting, chain-of-thought prompting y template prompting, que permiten mejorar la precisión y reducir la variabilidad del modelo (Liu et al., 2023). Estas técnicas son aplicables a tu proyecto, donde el objetivo es construir prompts automatizados a partir de parámetros del usuario.

Finalmente, el prompt engineering no elimina la necesidad de supervisión humana. Investigaciones muestran que los prompts pueden amplificar sesgos o producir resultados bien redactados pero incorrectos si no se diseñan con restricciones claras (Xu et al., 2023).

Por ello, un agente de IA especializado debe combinar prompts eficientes con filtros y verificación humana.

Figura 2

Proceso de Prompt Engineering en la interacción humano–modelo



Nota. Elaboración propia

2.2 IA en Procesos Editoriales

La IA ha transformado significativamente el sector editorial desde 2020. Publicaciones recientes muestran que las herramientas de IA se utilizan para generar borradores, corregir estilo, clasificar documentos y asistir en la planificación editorial (Smith, 2022). Dentro de instituciones educativas, donde existe una demanda constante de comunicados y boletines, estas herramientas aportan velocidad y estandarización.

Los sistemas de IA permiten a los equipos de comunicación concentrarse en tareas de mayor valor, como supervisión, diseño estratégico y validación. Esto es especialmente útil en universidades donde las unidades académicas producen contenido comunicacional recurrente (Scoble & Israel, 2023). La automatización de borradores puede reducir tiempos operativos en más del 50 %, según reportes recientes.

Sin embargo, la literatura también menciona retos: verificar la veracidad del contenido generado, evitar errores estilísticos y garantizar que el texto final refleje la identidad

institucional. Por eso, la IA es vista como un asistente y no un reemplazo del editor humano (UNESCO, 2023). Este equilibrio es clave para evitar riesgos comunicacionales.

Además, estudios actuales recomiendan integrar IA dentro de flujos formales, con fases de revisión, criterios de calidad y políticas claras. Esto coincide con el diseño de tu prototipo, que propone un agente especializado como una herramienta auxiliar para producción editorial universitaria.

Figura 3

Aplicación de la inteligencia artificial en los procesos editoriales.



Nota. Elaboración propia.

2.2.1 Agentes de IA Especializados

Los agentes de IA especializados son sistemas diseñados para operar dentro de un dominio específico, integrando reglas, datos contextuales y flujos de trabajo particulares. A diferencia de los modelos generalistas, estos agentes reducen la variabilidad y proporcionan respuestas consistentes y alineadas a un propósito particular (Wooldridge, 2021). Desde 2021 se han vuelto populares en sectores como salud, educación y comunicación institucional.

El valor principal de un agente especializado radica en su capacidad de encapsular conocimiento del dominio: glosarios, estilos, parámetros, restricciones y requisitos institucionales. Estos elementos permiten generar contenido que sigue reglas claras, lo que

es esencial en contextos editoriales universitarios (Russell & Norvig, 2021). Esta capacidad es precisamente lo que busca tu proyecto.

Los agentes también permiten incorporar mecanismos de verificación, auditoría y control, garantizando que el contenido cumpla estándares éticos y de calidad. A nivel técnico, pueden funcionar como una capa intermedia entre el usuario y el modelo generativo, estructurando los datos y validando la salida (Bommasani et al., 2021).

Finalmente, estudios recientes muestran que los agentes especializados facilitan la adopción de IA en organizaciones, ya que ofrecen más seguridad, coherencia y trazabilidad que los modelos generalistas usados directamente. Esto los convierte en una opción preferida para tareas sensibles como la comunicación institucional.

2.2.2 Interacción Humano–IA

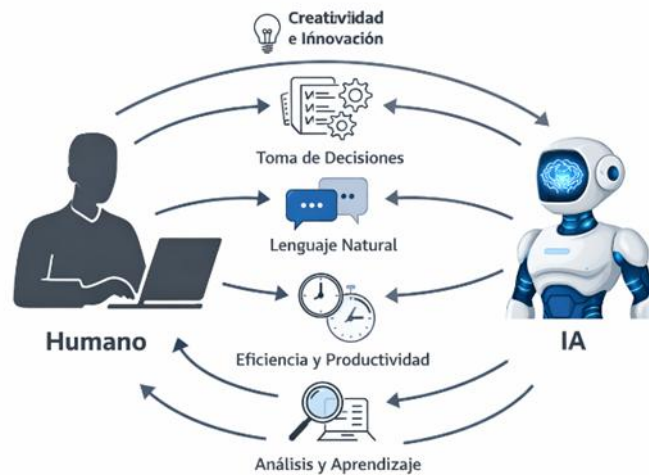
La interacción entre humanos y sistemas de IA ha evolucionado hacia un modelo colaborativo donde el usuario supervisa, corrige y valida contenido generado. Shneiderman (2020) propone un enfoque de “IA centrada en el humano”, donde la tecnología complementa el trabajo experto en lugar de reemplazarlo. Este enfoque es fundamental en entornos académicos y editoriales.

Las investigaciones contemporáneas destacan la importancia del diseño de interfaces que permitan editar, revisar y comprender el contenido generado por IA (Amershi et al., 2019). Una buena interacción mejora la confianza del usuario y facilita la adopción de nuevas herramientas tecnológicas. Esto es esencial en instituciones donde los flujos editoriales son rigurosos.

En el ámbito universitario, la interacción humana–IA permite que la IA genere borradores rápidos, mientras los editores ajustan contenido, verifican información y aseguran fidelidad institucional. Esta colaboración aumenta eficiencia sin comprometer la calidad académica (UNESCO, 2023).

Estudios recientes también recomiendan incluir mecanismos de transparencia, como mostrar cómo el modelo llega a ciertas conclusiones o permitir revisar versiones anteriores (Shneiderman, 2020). Estas prácticas fortalecen la aceptación institucional y la responsabilidad editorial.

Figura 4
Interacción entre el ser humano y la inteligencia artificial.



Nota.Elaboración propia.

2.2.3 Integración de APIs de IA

Las APIs de IA, como Azure OpenAI o Copilot, permiten conectar aplicaciones personalizadas con modelos generativos avanzados sin necesidad de entrenar modelos propios. Desde 2021 su uso ha aumentado debido a mejoras en seguridad, gobernanza y escalabilidad (Microsoft, 2024). Estas APIs son clave en proyectos que requieren IA sin infraestructura compleja.

Una API permite enviar instrucciones (prompts), recibir salidas generadas y procesar muchos parámetros desde una interfaz única. La integración requiere manejar autenticación, límites, errores y protección de datos. O'Reilly (2023) señala que el diseño modular es importante para mantener la flexibilidad y evitar dependencias rígidas.

En el contexto de tu prototipo, la API actúa como motor de generación mientras tu sistema controla los parámetros y estructura los prompts. Según Microsoft (2024), este enfoque reduce

la complejidad técnica, especialmente para prototipos o soluciones institucionales en etapas tempranas.

Además, las APIs modernas incluyen configuraciones para controlar privacidad, lo cual es esencial en universidades que manejan datos sensibles. La literatura aconseja filtrar información antes de enviarla a la API para evitar riesgos de exposición (UNESCO, 2023).

2.2.4 Sistemas Web Monolíticos

Los sistemas monolíticos reúnen en un solo proyecto el backend, el frontend y la lógica de negocio. Esta arquitectura es recomendada para prototipos, proyectos pequeños o etapas iniciales dado que permite desarrollar y probar más rápido (Fowler, 2020). En tu proyecto, un monolito es adecuado porque simplifica la integración con la API y el módulo de prompts.

Aunque los monolitos pueden escalar menos eficazmente que los microservicios, la literatura sugiere que para aplicaciones con pocos módulos y una función principal definida —como tu prototipo— la opción monolítica reduce costos y complejidad técnica (Richards, 2015). Además, su mantenimiento inicial es más simple.

La clave está en organizar correctamente las capas internas del monolito, como recomienda Fowler (2020): separar presentación, servicios, lógica de negocio y adaptadores externos. Esto facilita migrar a microservicios si la aplicación crece en el futuro.

Finalmente, estudios recientes indican que, en instituciones educativas, donde los recursos técnicos pueden ser limitados, la arquitectura monolítica ofrece una implementación más viable para proyectos de corta duración o pilotos tecnológicos (O'Reilly, 2023).

2.2.5 Aplicaciones de IAGen en Educación Superior

Desde 2020, las universidades han adoptado IA generativa para apoyar redacción académica, tutorías, retroalimentación automática, comunicación institucional y análisis de documentos (UNESCO, 2023). Esta tecnología se ha convertido en un apoyo importante para docentes y administrativos.

Investigaciones muestran que la IAGen mejora la personalización educativa, genera material de apoyo y permite automatizar la creación de resúmenes, guías y explicaciones (Zawacki-Richter et al., 2019). En comunicación institucional, también facilita la producción de boletines y comunicados coherentes entre unidades académicas.

No obstante, la literatura destaca riesgos: pérdida de habilidades, errores factuales, dependencia tecnológica y afectaciones a la integridad académica. Por esto, UNESCO (2023) recomienda implementar marcos éticos y capacitación en alfabetización digital para docentes y estudiantes.

En términos operativos, la IAGen puede mejorar la eficiencia sin reemplazar la revisión humana. Los estudios recomiendan un enfoque híbrido donde la IA produzca el borrador y el editor aplique criterios académicos (Scoble & Israel, 2023).

2.2.6 Calidad del contenido generado por IA

La calidad del texto generado por IA se evalúa mediante criterios como coherencia, precisión factual, claridad, ausencia de sesgos y adecuación al contexto. Desde 2020 ha crecido el interés por estudiar las “alucinaciones” o errores fabricados producidos por LLMs (Mitrović et al., 2023). Este fenómeno es crítico, especialmente en educación y comunicación institucional.

Xu et al. (2023) señalan que es imprescindible combinar evaluación automática con evaluación humana, ya que muchos errores de IA no se detectan mediante algoritmos tradicionales. Esto respalda la necesidad de supervisión editorial.

La calidad también depende de diseños adecuados de prompts y del uso de plantillas estructuradas para garantizar consistencia estilística y terminológica (White et al., 2023). En contextos universitarios, donde el tono institucional es clave, estas plantillas son cruciales.

Finalmente, la literatura recomienda mantener registros y auditoría del contenido generado para mejorar continuamente los prompts, detectar fallos recurrentes y asegurar responsabilidad institucional (UNESCO, 2021).

2.3 Ética y regulación del uso de IA en educación

La ética en IA educativa se centra en transparencia, privacidad, equidad y responsabilidad. UNESCO (2021) establece directrices globales para el uso de IA en educación, enfatizando la necesidad de supervisión humana y protección de datos.

Floridi & Cowls (2019) proponen un marco unificado de principios éticos que incluye beneficencia, justicia y autonomía. Estos principios son aplicables cuando una universidad adopta IA en sus procesos editoriales.

Además, la literatura recomienda lineamientos internos como declarar el uso de IA en documentos oficiales, evitar compartir datos sensibles en prompts y establecer políticas claras para estudiantes y docentes (UNESCO, 2023).

La regulación emergente y las prácticas institucionales coinciden en que la IA debe apoyar — y no reemplazar— el juicio profesional humano en tareas académicas y comunicacionales.

2.3.1 Procesos Editoriales Académicos

Los procesos editoriales académicos incluyen planificación, redacción, revisión, corrección, aprobación y difusión. Estas actividades requieren precisión, consistencia y apego a normas institucionales (Day & Gastel, 2017). En universidades, estos flujos son críticos en la comunicación institucional.

La automatización mediante IA puede asistir en etapas iniciales como creación de borradores, resúmenes y reformulación de contenido. Sin embargo, la literatura insiste en que la revisión humana es indispensable para garantizar integridad académica (Greetham, 2020).

Herramientas de IA pueden reforzar la eficiencia, pero no sustituyen la responsabilidad editorial en verificaciones, citación y estilo, especialmente en entornos académicos (Smith, 2022). El rol del editor es interpretar, contextualizar y validar.

Finalmente, buenas prácticas editoriales recomiendan integrar IA dentro de procesos regulados, con fases de revisión humana, criterios de calidad y documentación del flujo, como enfatizan las recomendaciones recientes de la industria (Scoble & Israel, 2023).

Figura 5

Procesos editoriales académicos apoyados por inteligencia artificial.



Nota. Elaboración propia.

2.3.2 JavaScript

Un lenguaje de programación de alto nivel, interpretado y ampliamente utilizado en el desarrollo web moderno para dotar a las páginas de comportamiento dinámico e interacción con el usuario. Originalmente fue creado en 1995 por Brendan Eich, desarrollándose en un periodo muy breve y convirtiéndose rápidamente en una de las tecnologías fundamentales de la World Wide Web. JavaScript permite que las páginas no solo presenten contenido estático, sino que respondan a acciones del usuario, actualicen el contenido sin recargar la página y proporcionen una experiencia dinámica y rica en funcionalidades.

JavaScript se considera uno de los tres pilares del desarrollo web, junto con HTML y CSS. Mientras que HTML estructura el contenido de un sitio web y CSS define su presentación visual, JavaScript habilita la lógica de programación que permite interactividad y comportamiento automatizado en tiempo real.

Desde sus inicios como un simple lenguaje de scripting para navegadores, JavaScript ha evolucionado para superar las limitaciones del lado del cliente. Con la introducción de entornos como Node.js, ha ampliado su uso hacia el lado del servidor, permitiendo que el mismo lenguaje sea empleado para desarrollar tanto el frontend como backend de aplicaciones completas. Esta expansión ha consolidado su posición como una tecnología versátil y omnipresente en soluciones web modernas.

2.3.3 Producto Mínimo Viable (MVP)

El Producto Mínimo Viable (MVP) es una estrategia empleada en el desarrollo de software que consiste en crear la versión más básica de un producto, incluyendo solamente las funciones esenciales que permitan a usuarios reales probarlo y ofrecer sus comentarios tempranos. Este enfoque facilita la validación de suposiciones sobre el diseño, reduce riesgos y dirige las mejoras basándose en datos, evitando así malgastar tiempo y recursos en características innecesarias durante la etapa inicial del proyecto. La implementación del MVP en proyectos de innovación tecnológica es altamente recomendada dentro de metodologías ágiles y en áreas como la ingeniería de software y la gestión de proyectos, ya que está orientada a proporcionar un valor constante y a fomentar un aprendizaje continuo.

2.4 React

A diferencia de un framework completo, React se ocupa de la capa visual (UI) de una aplicación, permitiendo a los programadores construir interfaces modulares combinando componentes reutilizables. Esta estructura basada en componentes facilita mantener el código organizado, simplifica su mantenimiento y promueve la reutilización, particularmente en aplicaciones que gestionan estados complejos o que requieren interfaces interactivas. React utiliza un DOM virtual que optimiza el rendimiento al actualizar solo aquellas partes de la interfaz que se necesitan, aumentando así la eficacia y la satisfacción del usuario en aplicaciones de una sola página (SPA). Actualmente, se considera una de las bibliotecas más populares para el

desarrollo del frontend, con el respaldo de una comunidad activa y el apoyo de Meta Platforms, siendo ampliamente utilizada en aplicaciones web modernas. Se han realizado estudios recientes sobre React que han explorado mejoras en la eficiencia del renderizado y técnicas avanzadas de rendimiento, destacando cómo las estructuras modulares y la hidratación adaptativa pueden optimizar aplicaciones web altamente interactivas construidas con React.

2.4.1 Chakra UI

Chakra UI es una biblioteca que proporciona componentes de interfaz de usuario para aplicaciones desarrolladas con React. Ofrece elementos visuales listos para usar, como botones, formularios, menús y componentes de navegación, lo que simplifica la creación de interfaces accesibles, consistentes y adaptables sin tener que diseñar estilos desde cero. Bibliotecas como Chakra UI aceleran significativamente el proceso de desarrollo al ofrecer elementos personalizables y optimizados, lo que permite a los desarrolladores enfocarse en la lógica de la aplicación en lugar de construir manualmente cada componente visual. Además, Chakra UI prioriza la accesibilidad y la coherencia visual, aspectos clave para mejorar la experiencia del usuario en aplicaciones web complejas.

2.4.2 Microsoft Azure y su herramienta Foundry

Es una plataforma en la nube que proporciona infraestructura, herramientas y servicios gestionados para desarrollar, implementar y escalar aplicaciones inteligentes. Entre sus diversas ofertas, Azure AI Foundry es una solución diseñada para facilitar el desarrollo de aplicaciones de inteligencia artificial generativa, permitiendo la conexión con distintos modelos de lenguaje avanzados. Ofrece la capacidad de ajuste y un entorno integrado para manejar agentes y flujos de trabajo de IA. Azure Foundry permite a los programadores controlar el acceso a modelos como GPT-4. 1 y otros que se encuentran en el catálogo de IA generativa de OpenAI, garantizando gobernanza, seguridad y escalabilidad en su uso en

entornos prácticos.

Foundry se distingue por su propósito de facilitar la creación de agentes de IA, integrando herramientas tanto para el entrenamiento como para la implementación de modelos, así como también ofreciendo apoyo a flujos de trabajo que incluyen métricas de evaluación y su incorporación en ambientes de producción.

2.4.3 Modelo GPT-4. 1

Los modelos de lenguaje de gran tamaño, como GPT-4. 1, son sistemas de inteligencia artificial generativa que utilizan la arquitectura Transformer, lo que les permite generar texto coherente y contextual con un alto nivel de calidad textual a partir de instrucciones o prompts. GPT (Generative Pre-trained Transformer) hace referencia a un tipo de modelo previamente entrenado con grandes cantidades de datos y optimizado para seguir instrucciones y crear contenido en distintos dominios. Aunque la documentación sobre GPT-4. 1 específicamente puede ser limitada en la literatura académica formal debido a su reciente aparición, la plataforma Azure AI Foundry lo incluye entre los modelos avanzados que se pueden integrar en aplicaciones de IA generativa, destacando su relevancia tecnológica en soluciones actuales de creación de contenido.

Modelos como GPT-4. 1 son frecuentemente utilizados para generar texto, comprender y responder a instrucciones complejas, desempeñándose como componentes fundamentales en agentes de IA que necesitan producir contenidos coherentes y adaptados a contextos específicos. Su implementación en aplicaciones reales permite automatizar tareas de redacción, elaboración de borradores y asistencia inteligente en actividades lingüísticas, siempre bajo la supervisión humana para garantizar calidad y precisión.

CAPÍTULO III

METODOLOGÍA DE LA INVESTIGACIÓN

3.1 Tipo de investigación

Para el presente proyecto de investigación se combina métodos cualitativos (entrevistas) y cuantitativos (encuestas), debido a que se busca desarrollar un prototipo funcional.

Con un diseño exploratorio-descriptivo, ya que se indaga necesidades y luego se describe cómo el prototipo responde a ellas.

3.2 Población y muestra

Población objetivo: estudiantes y docentes universitarios de la Facultad de Ingeniería

Tabla 1
Población y muestra

Tipo	Número
Docentes	50
Estudiantes	495
Total	545

Nota. Tomado de los reportes de Matriculados y Reporte de Docentes por facultad de Ingeniería la UCSG.

3.2.1 Selección de la muestra

Para la entrevista se consideró entrevistar al Director de Publicaciones que también es Docente de la Facultad de ingeniería, y encuestar a diversos docentes de la Facultad. En cuanto a los estudiantes se eligió a estudiantes de los últimos ciclos de la Facultad ya que han tenido una mayor experiencia redactando ensayos y en la utilización de la IA

Para una población de 545 personas, con un nivel de confianza de 95% y un margen de error del 5%, la muestra es de aproximadamente 226 encuestados.

3.3 Técnicas e instrumentos para obtención de información

Para la recolección de información que se utilizará en el desarrollo del proyecto, la entrevista cuenta con preguntas abiertas, que se realizarán de forma presencial. Las encuestas tienen preguntas cerradas, realizadas en Google Forms para una mejor distribución y recepción de respuestas. Se encuentran en los Anexos.

3.4 Análisis de resultados de las entrevistas

Uno de los entrevistados fue un **docente-investigador** cuya labor se centra en procesos institucionales y mejora continua. Su perfil resultó relevante por la frecuencia con la que redacta textos editoriales (reseñas, notas institucionales, perfiles académicos), lo que le otorga una perspectiva práctica sobre la utilidad de la herramienta.

- **Carga de trabajo y tono institucional:** el docente invierte entre 1 y 2 horas en cada borrador, destacando la dificultad de sintetizar sin perder contenido y la importancia de mantener claridad y tono institucional.
- **Condiciones de implementación:** recomendó capacitación, lineamientos éticos y acompañamiento institucional para garantizar un uso responsable.

Los hallazgos de la entrevista se utilizaron para **ajustar el diseño del prototipo**, especialmente en:

- La necesidad de personalización por área académica.
- La importancia de mantener un tono institucional claro.
- La inclusión de mecanismos que promuevan la validación humana como parte del flujo de trabajo.

En la siguiente entrevista se pueden resaltar los siguientes puntos:

1. Contexto del docente

- Docente en programación y Director de Publicaciones.

- Alta frecuencia en la elaboración de textos editoriales y divulgativos.
- Reconoce que la redacción implica investigación y organización, siendo una tarea repetitiva y demandante de tiempo.

2. Experiencia y dificultades

- Desafíos: evitar repeticiones, lograr equilibrio entre extensión y concreción, transmitir conceptos con claridad.
- Importancia del tono institucional y la claridad, adaptando el estilo según el público (docentes, estudiantes, general).
- Ha usado herramientas digitales/IA como apoyo inicial, con resultados buenos pero que requieren ajustes.
- Considera útil la IA para reducir carga de trabajo en publicaciones y difusión.
- Requerimiento: temática y extensión para generar prompts más precisos.

3. Aspectos éticos y pedagógicos

- Preocupación por la pérdida de autenticidad e identidad humana en textos generados por IA.
- Necesidad de validación humana antes de publicar (supervisión, lectura, aprobación).
- Riesgos: veracidad, autoría y propiedad intelectual; peligro de textos genéricos si se delega totalmente a la IA.
- Impacto en estudiantes: puede ser riesgo si inhibe creatividad, pero también oportunidad de aprendizaje si se usa como apoyo para perfeccionar la redacción.

4. Cierre

- Recomienda el prototipo para facultad/departamento, especialmente en páginas de difusión y comunicación institucional.
- Condiciones para éxito: pruebas, exposiciones y parametrizaciones que aseguren autenticidad y eviten textos genéricos.

Se puede concluir en estas entrevistas que la IA es una herramienta muy importante para generar contenidos editoriales y así poder reducir cargas de trabajo, siempre que se adapten parámetros como tema, público objetivo, tono y extensión, sin embargo, debe existir supervisión humana para garantizar la autenticidad, creatividad, originalidad y tono institucional, y así evitar que los textos se conviertan en genéricos.

3.4.1 Diseño de la encuesta

El diseño de la encuesta se centro hacia las variables centrales del estudio. De los encuestado 17 fueron docentes y 210 fueron estudiantes. A continuación, se detallan las preguntas.

¿Ha utilizado alguna herramienta de IA para apoyar la redacción o revisión de textos académicos?

¿Qué tipos de documentos cree que son más adecuados para que los redacte la IA?

¿Qué nivel de confianza tendría en un borrador generado por IA antes de la revisión humana?

¿Qué riesgos percibe en el uso de IA para redactar textos académicos?

¿Qué funciones considera más útiles en el prototipo?

En una escala del 1 al 5, ¿qué tan útil considera que sería un agente de IA generativa para apoyar la labor editorial universitaria o trabajo académico?

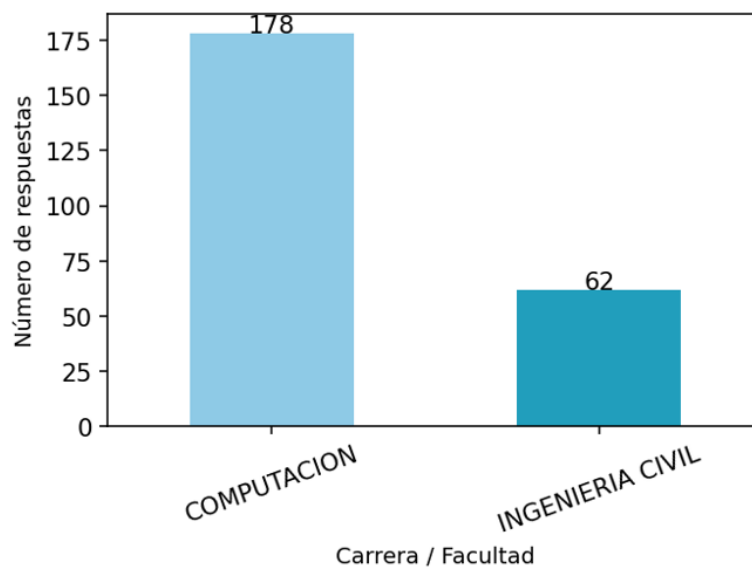
3.4.2 Análisis de los resultados de la encuesta

Composición de la muestra

De un total de 240 participantes, donde la mayor parte son de la carrera de Computación (n=178) y de Ingeniería Civil (n=62), como podemos observar en la figura la distribución por facultad.

Figura 6

Gráfico estadístico de muestra por carrera

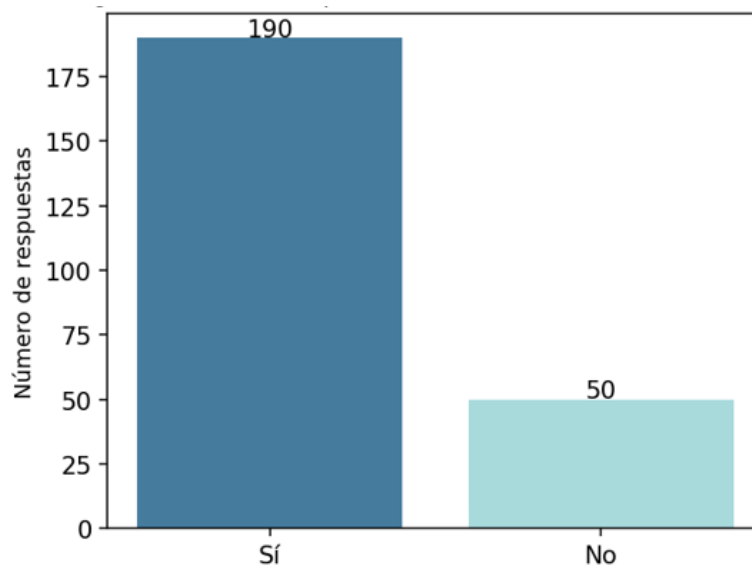


Uso de la IA generativa para escribir

De acuerdo a la pregunta **¿Has utilizado alguna herramienta de IA para redactar textos académicos?**, el **79%** (190/240) respondió **Sí**, y el **21%** (50/240) **No**. Esta pregunta mide la adopción y madurez de uso, se muestra que la mayoría ya usa IA, por esto el prototipo debe mejorar la calidad y ética, ya que se demuestra la creciente de IA generativa en el proceso de redacción, lo que se puede percibir como una herramienta útil, aunque deja también un poco de dudas sobre la ética.

Figura 7

¿Han utilizado IA para redactar textos académicos?

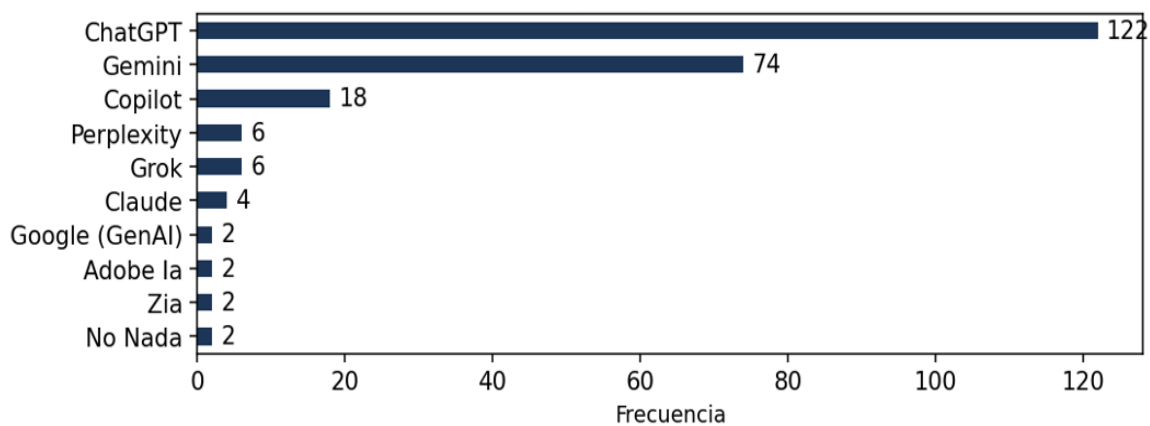


Herramientas utilizadas

Para identificar las herramientas más utilizadas y poder planificar el lenguaje de interacción como prompt, formatos, se consultó a los que, si han usado IA, ¿qué herramienta utiliza? Y las respuestas muestran un top 5:

Figura 8

Herramientas de IA más utilizadas (top10)



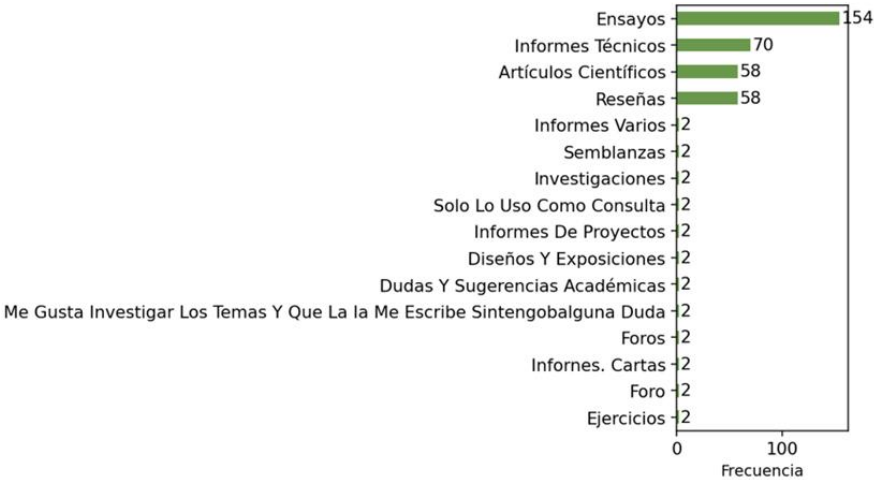
Tipos de documentos que redactan

Se realizó la pregunta **¿Qué tipo de documentos académicos redactas con mayor frecuencia?**, donde los tipos de documentos que redactas con mayor frecuencia son los

ensayos, informes técnicos, artículos científicos, y reseñas como muestra la figura, con este perfil se orienta a un diseño del prototipo hacia plantillas guiadas y asistencias específicas.

Figura 9

Tipos de documentos redactados con mayor frecuencia

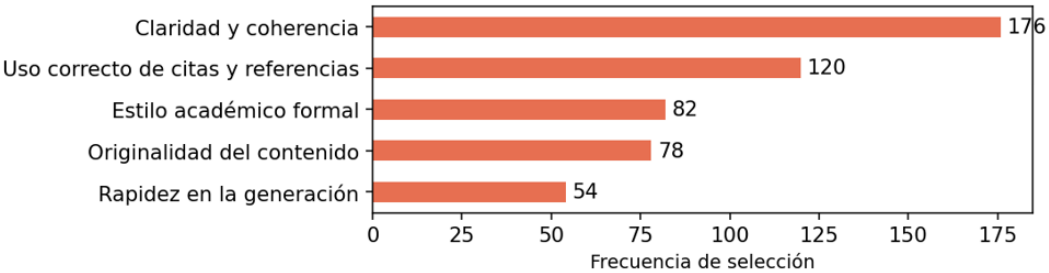


Aspectos importantes más valorados

Se realizó la pregunta **¿Qué aspectos son más importantes en un borrador generado por IA?**, para conocer lo que los usuarios más valoran en cuanto a los criterios de calidad de los borradores generados por la IA la **“Claridad y coherencia” (n=176)**, junto con el **uso correcto de citas y referencias**, el **estilo académico formal**, la **originalidad del contenido** y por último la rapidez de generación, se prioriza la claridad y el manejo de referencias lo que indica que el prototipo debe optimizar la organización, la coherencia y también ofrecer asistencia para realizar citas.

Figura 10

Aspectos más valorados en un borrador generado por IA

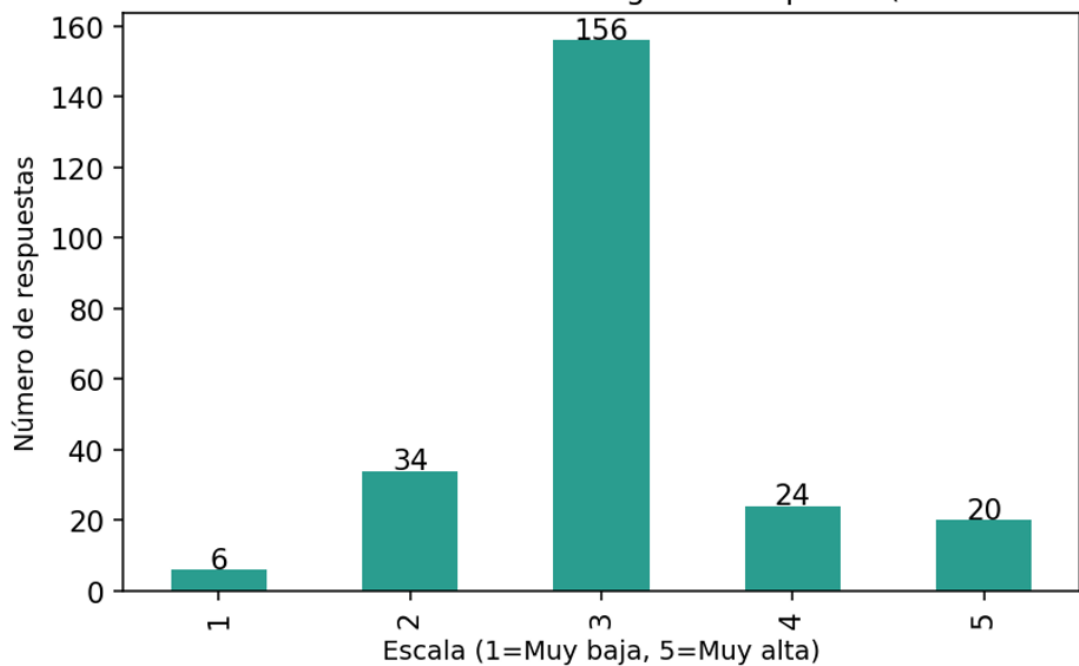


Nivel de confianza en borradores generados por IA

Con la pregunta ¿Qué nivel de confianza tendrías en un borrador generado por IA?, se puede observar que existe una confianza moderada con una media de 3.08, existe confianza, pero no se delega completamente el juicio académico a la IA, para lo que se demuestra la importancia y la necesidad de verificación humana en el flujo de trabajo.

Figura 11

Nivel de confianza en un borrador generado por IA (Media=3.08)

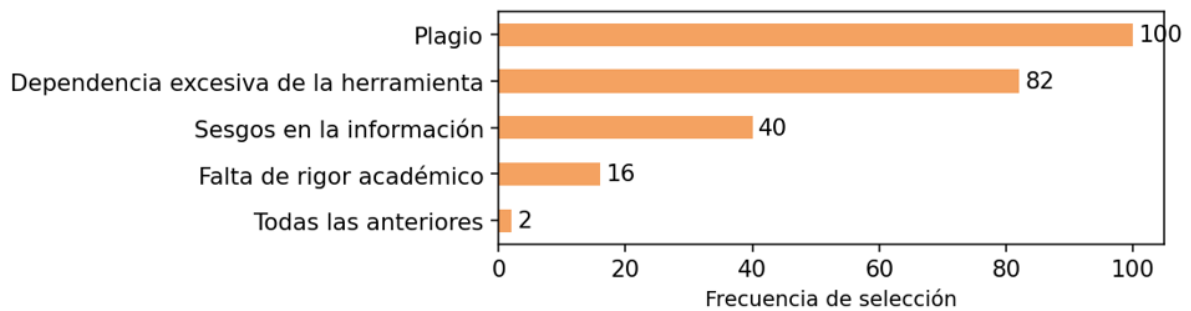


Riesgos percibidos

Tenemos la pregunta: ¿Qué riesgos percibes en el uso de IA para redactar?. Para identificar los riesgos que se deben mitigar con el diseño, se puede identificar que los más señalados fueron el **Plagio (n=100)**, la **Dependencia excesiva**, la **Falta de rigor académico** y los **Sesgos en la información** (Figura 12). Esto implica que se debe manejar la IA con responsabilidad.

Figura 12

Riesgos percibidos en el uso de la IA para redactar

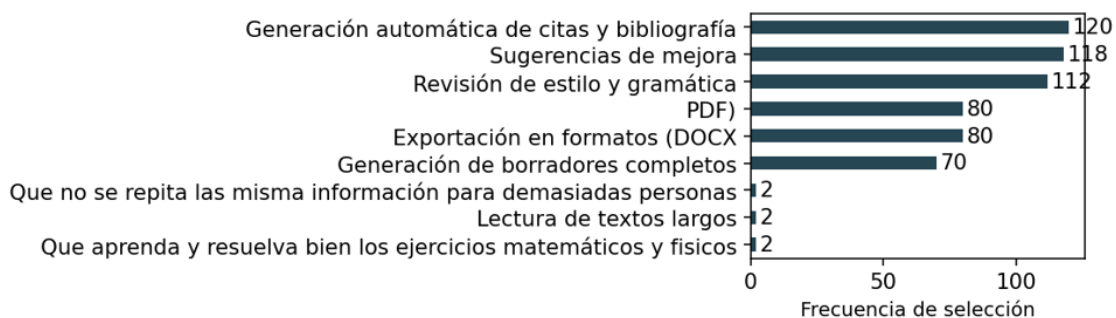


Funciones deseadas para el prototipo

Con esta pregunta muy importante para el desarrollo del prototipo **¿Qué funciones te gustaría que tenga el prototipo?**, las respuestas con más demanda son: **“Generación automática de citas y bibliografía” (n=120)**, seguida de **Borradores completos, Revisión de estilo y gramática, Sugerencias de mejora y Exportación a DOCX/PDF** (Figura 13). Este resultado será de mucha utilidad ya que servirá para organizar lo que se va a construir, dar prioridades, tener una guía y alinearse con las necesidades del usuario, un prototipo con foco en citas verificables, revisión de estilo y exportación.

Figura 13

Funciones deseadas para el prototipo



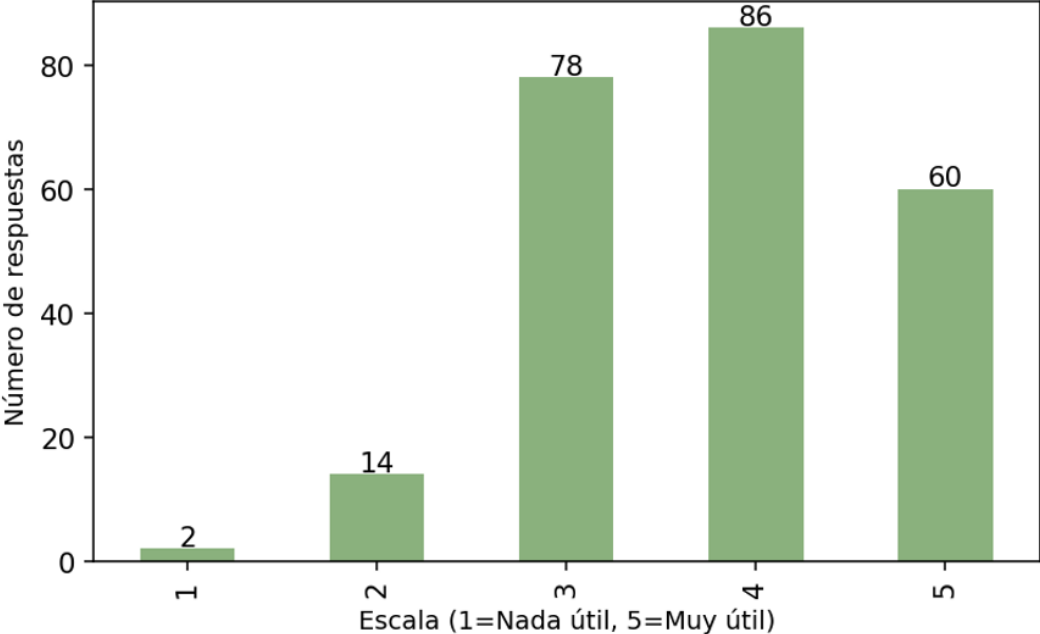
Utilidad percibida del agente de IA

Finalmente se tiene la siguiente pregunta, **En una escala 1–5, ¿qué tan útil sería un agente de IA para tu trabajo académico?**, esta pregunta permite medir la intención de uso e impacto percibido, que es importante para sustentar la viabilidad del prototipo. Se tiene una media de 3.78, la misma que se percibe como positiva, ya que supera el punto medio, con lo

cual se está anticipando los beneficios que el prototipo le puede aportar como un agente de IA enfocado en redacción académica.

Figura 14

Utilidad percibida del agente de IA (Media=3.78)



CAPÍTULO IV

PROPUESTA TECNOLÓGICA

4.1 Diseño de la solución

Se desarrolló un prototipo funcional de un agente de IA especializado en la redacción de contenidos editoriales universitarios. El proyecto se llevó a cabo utilizando una metodología de desarrollo iterativa e incremental donde se priorizó la creación de un producto funcional mínimo viable (MVP) y luego se perfeccionó en ciclos cortos y enfocados.

Técnicamente: se integró en módulos independientes que separaban la interfaz de usuario, que se desarrolló con React 18 y Chakra UI, de la lógica empresarial, que se desarrolló en JavaScript.

El núcleo del sistema reside en un motor de ingeniería sobre la marcha que transforma los parámetros del usuario en instrucciones de modelo de lenguaje detalladas y estructuradas.

La solución se integró con el ecosistema de Microsoft Azure mediante la herramienta Microsoft Foundry.

El proceso de desarrollo se desarrolló en tres etapas principales:

1) Definición de problemas y diseño de arquitectura modular: Se definieron las necesidades funcionales y se organizó la arquitectura principal del sistema.

2) Implementación central (sistema de avisos, interfaz e integración con IA): se crearon el generador de avisos, la interfaz de usuario y la conexión a la API de inteligencia artificial generativa.

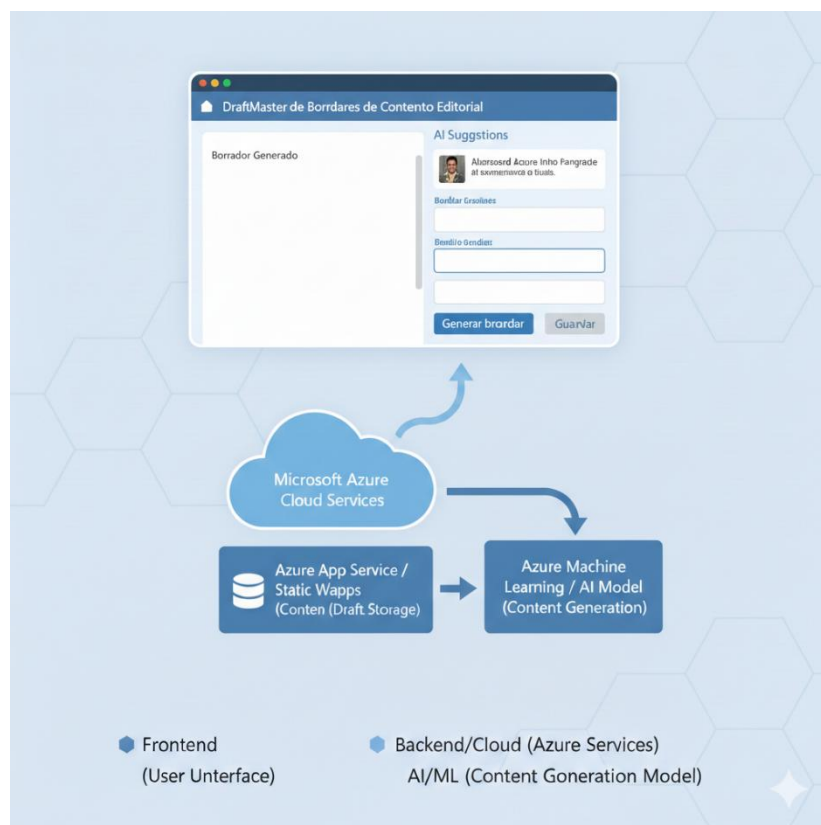
3) Mejoras en el diseño institucional: exportación a .docx (Word) y mejoras en la experiencia del usuario, como diseño visual en paralelo y desplazamiento automático, así como implementación de manejo simple de archivos de texto.

Las decisiones clave de ingeniería incluyeron una separación estricta de la interfaz de usuario y la lógica para una mejor mantenibilidad, el uso de técnicas de ingeniería rápida especializadas para garantizar la calidad del borrador y la adopción de un diseño receptivo y accesible desde el principio.

El sistema resultante es funcional, rápido, utilizable y produce resultados listos para su integración en flujos de trabajo editoriales universitarios del mundo real, lo que demuestra el valor práctico de la IA generativa en el contexto de la comunicación institucional.

Figura 15

Arquitectura del prototipo



Nota. Generado con Gemini Nano Banana

4.2. Desarrollo del prototipo

Se eligieron tecnologías y herramientas para la creación del prototipo funcional, lo que garantizó una implementación rápida, flexibilidad, capacidad de escalamiento y una fácil integración con servicios de inteligencia artificial generativa.

Esta elección estratégica se basó en un análisis comparativo de varias opciones disponibles en el mercado, priorizando aquellas que ofrecen el equilibrio óptimo entre curva de aprendizaje, documentación disponible, comunidad de soporte y compatibilidad con objetivos específicos del proyecto.

El enfoque elegido fue un conjunto tecnológico moderno pero estable, evitando tecnologías demasiado experimentales que pudieran amenazar la estabilidad del prototipo, manteniendo al mismo tiempo la posibilidad de evolucionar a arquitecturas más complejas en etapas posteriores del proyecto.

Los principales componentes tecnológicos y herramientas utilizados en el proceso de creación de prototipos se describen a continuación: Se eligió React como la biblioteca de interfaz de usuario principal debido a su modelo de componentes reutilizable y su ecosistema maduro de herramientas adicionales. Se eligió Vite como herramienta de compilación en lugar de la aplicación tradicional Create React porque Vite ofrece un tiempo de desarrollo significativamente más rápido debido al uso de módulos ES nativos y al reemplazo casi instantáneo de Hot Module. Esta combinación nos permitió iterar rápidamente la interfaz de usuario y probar diferentes diseños y flujos de trabajo sin las penalizaciones de rendimiento de otras configuraciones.

Se utilizó JavaScript puro ES6 para implementar la lógica de negocios y el procesamiento de datos, organizados en módulos independientes de la capa de presentación.

Esta separación arquitectónica permitió desarrollar y probar la funcionalidad principal del sistema sin depender de una interfaz gráfica, facilitando tanto las pruebas unitarias como la posible reutilización de esta lógica en otras plataformas.

La integración con servicios externos de IA se implementó utilizando un modelo de cliente HTTP con un sólido manejo de errores, reintentos estratégicos y validación de respuestas para garantizar la confiabilidad del sistema incluso en condiciones de red subóptimas o disponibilidad fluctuante del modelo de lenguaje.

4.2.1 Modelo GPT-4. 1

Se eligió el sistema GPT-4.1 como recurso para el desarrollo de contenidos debido a su capacidad para comprender instrucciones complejas, mantener la coherencia del contexto y producir texto de alta calidad.

De las alternativas evaluadas, GPT-4.1 ofreció el mejor equilibrio entre razonamiento avanzado, conocimiento del dominio académico y costo unitario.

Un token que es significativamente mejor que versiones anteriores, como GPT-3.5 turbo, para tareas que requieren una comprensión profunda del contexto y la ejecución precisa de instrucciones complejas, manteniendo un costo menor que opciones más especializadas o de mayor tamaño.

Este sistema ayudó a crear borradores con un estilo formal, adecuado para diferentes públicos y adaptados a pautas definidas por el usuario.

Sin embargo, el proyecto no se centró en entrenar el modelo, sino en construir una capa intermedia de ingeniería instantánea que apuntara precisamente a sus respuestas a las publicaciones académicas. Este enfoque de "ingeniería rápida", en lugar de "entrenamiento de modelos", fue más eficiente y práctico para la creación de prototipos, ya que permitió obtener resultados de dominios específicos de alta calidad sin recopilar grandes conjuntos de datos de entrenamiento ni incurrir en el alto costo y la complejidad técnica de ajustar los modelos de lenguaje.

Las especificaciones técnicas del modelo implementado incluyeron: una ventana de contexto de 128.000 tokens que permitía incluir instrucciones detalladas y un contexto extenso sin truncamiento; un modo de implementación como un punto final de inferencia en tiempo real optimizado para baja latencia en aplicaciones interactivas; midió una latencia promedio de 2,3 segundos por solicitud, lo que proporcionó una experiencia de usuario receptiva; y un SLA de disponibilidad del 99,9 % garantizado por parte de Azure, que aseguró la confiabilidad operativa. Estas características técnicas, combinadas con las capacidades inherentes de GPT-4.1 para un razonamiento complejo, manteniendo coherentes los textos largos y adaptándose a diferentes estilos y tonos, lo convirtieron en una opción ideal para el propósito específico de generar contenido editorial de alta calidad.

Figura 16

Costo aproximado por millón de tokens del modelo gpt-4.1

Hechos rápidos OpenAI

gpt-4.1

Finalización de chat responses

Directo desde Azure ⓘ

Datos de entrenamiento actualizados por última vez
Mayo 2024

Precios
[Consultar directamente los precios de Azure](#)

Referencia >

0,83 Índice de calidad
Calidad de la inteligencia artificial

3,5 USD por 1 millón de tokens
[Coste estimado](#)

Id. de modelo
Hacer referencia a este id. de modelo al implementar el modelo en el código

`azureml://registries/azure-openai/models/gpt-4.1/versions/2025-04-14`

4.2.2 Comparación de herramientas

Tabla 2
React Vs Angular

React	Angular
Es más ligero y flexible	Es más pesado y estructurado
Curva de aprendizaje más sencilla	Más complejo (TypeScript obligatorio)
Mayor comunidad y demanda laboral	Comunidad fuerte, pero menor que React

React	Angular
Más adaptable a diferentes proyectos	Más rígido en su arquitectura

Tabla 3
Interfaz de usuario de Chakra Vs CSS puro

Chakra UI	Material UI
Un diseño más flexible y personalizable	Basado en el diseño de materiales (más difícil)
Sintaxis más sencilla	Configuraciones más complejas
Mejor experiencia para temas (modo ligeramente oscuro/claro)	Una adaptación más amplia pero más técnica
Más fácil	Es más pesado

Tabla 4
JavaScript frente a Python

JavaScript	Python
El idioma principal en el front-end de la web.	Se utiliza principalmente en backend y ciencia de datos.
Funciona en un navegador	No funciona directamente en el navegador.
Ideal para aplicaciones web interactivas	Ideal para análisis de datos e inteligencia artificial
Permite el desarrollo de pila completa (Node.js)	Más fuerte en backend y automatización

Tabla 5*Ingeniería rápida avanzada versus ingeniería rápida básica*

Básica	Avanzada
Preguntas directas	Instrucciones estructuradas
Resultados generales	Resultados más precisos
Sin más contexto adicional	Utilizar el contexto, el rol y las limitaciones
Menos control	Mayor control sobre la producción

Tabla 6*Microsoft Azure Vs DigitalOcean*

Microsoft Azure	DigitalOcean
Puede resultar difícil al principio, hay muchas opciones avanzadas	Muy fácil de usar, perfecto si estás empezando
Muy alto - perfecto para grandes empresas	Bueno, pero más limitado que Azure
Funciona muy bien con Microsoft y herramientas empresariales	No tan integrado con los sistemas empresariales
Muy amplio (IA, análisis, servidores, bases de datos, seguridad)	Más pequeño (gotita, bases simples, Kubernetes básico)

4.3 Implementación detallada de los módulos

4.3.1 Módulo de entrada (ModuleInput)

Figura 17

Módulo de Entrada (ModuleInput) en arquitectura de sistema.



Nota.Elaboración propia mediante herramienta de inteligencia artificial, 2026.

Este componente de React fue diseñado para que administremos estados en un formulario que fue validado en tiempo real con Chakra UI FormError.

El componente ModuleInput se estructuró como un formulario controlado que contiene siete parámetros editoriales esenciales: tipo de contenido, público objetivo, tono, tema principal, puntos clave a incluir, longitud aproximada y elementos de estilo a evitar.

Cada parámetro se implementó con un componente de formulario de la interfaz de usuario de Chakra que mejor se adapta a su esencia: una selección para configuraciones predefinidas, un área de texto para texto largo como tema principal y una entrada con un control de matriz dinámica para puntos clave y exclusiones.

Este marco permitió al usuario especificar intuitivamente requisitos editoriales complejos y guiarlos a través de un flujo lógico que refleja el proceso de planificación de contenido real en un entorno universitario.

El manejo del estado se implementó utilizando el gancho `useState` de React para cada campo, lo que permite un control detallado sobre la validación y el comportamiento de la interfaz.

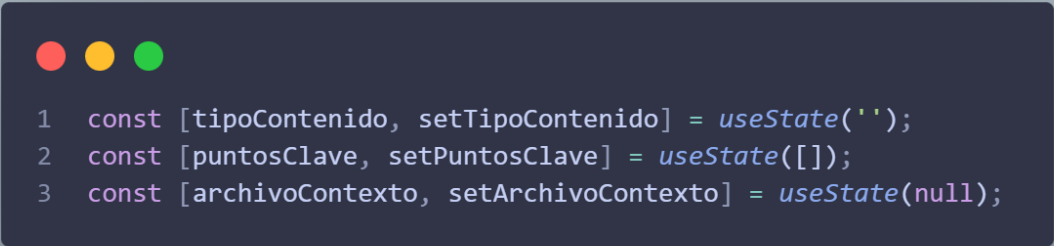
Para campos complejos como "puntos clave" y "excluir elementos", se implementó un sistema de matriz dinámica donde el usuario podía agregar, editar y eliminar elementos usando botones con íconos de Chakra UI, con validación en tiempo real que evita entradas duplicadas o vacías.

Las validaciones se integraron utilizando el marco `FormControl` de Chakra UI junto con `FormErrorMessage`, que mostraba mensajes contextuales cuando un campo requerido estaba vacío o contenía datos no válidos, proporcionando comentarios instantáneos sin la necesidad de enviar un formulario.

Este enfoque proactivo mejoró enormemente la experiencia del usuario al prevenir errores comunes antes de que ocurran y ayudar al editor a encontrar entradas válidas.

Figura 18

Estados manejados en el formulario de ingreso



```
1  const [tipoContenido, setTipoContenido] = useState('');
2  const [puntosClave, setPuntosClave] = useState([]);
3  const [archivoContexto, setArchivoContexto] = useState(null);
```

Además del formulario principal, se extendió el ModuleInput para incluir un sistema opcional de procesamiento de archivos que permitía a los usuarios adjuntar documentos de referencia. Este sistema utilizaba la API FileReader de JavaScript para extraer texto de archivos .txt, .md y .csv, aplicando validaciones de tamaño (máximo 50KB) y formato antes de procesar el contenido. El texto extraído se limpiaba automáticamente, se truncaba si excedía los 5,000 caracteres para respetar los límites de contexto del modelo, y se mostraba un preview al usuario para confirmación. Esta funcionalidad se integró perfectamente con el flujo existente, pasando el texto procesado como contexto adicional al motor de prompts cuando el usuario hacía clic en "Generar Borrador", sin requerir cambios en la estructura fundamental del componente o en la experiencia de usuario para quienes no necesitaban esta funcionalidad.

Figura 19

Función que extrae texto del archivo



```
1  const extractText = async (file) => {
2    return new Promise((resolve, reject) => {
3      const reader = new FileReader();
4      reader.onload = (e) => {
5        const text = cleanText(e.target.result);
6        if (isReadableText(text)) resolve(text);
7        else reject(new Error('Texto no legible'));
8      };
9      reader.readAsText(file, 'UTF-8');
10   });
11  };
```

4.3.2. Motor de Prompt Engineering.

Aquí se utilizó la siguiente estructura jerárquica de cinco niveles, que traduce los parámetros del usuario en instrucciones precisas para el modelo de lenguaje. Nivel 1: System Prompt define el rol básico del sistema como “un redactor especializado en comunicación institucional universitaria con más de 10 años de experiencia”, creando un contexto profesional que guía todo el proceso de generación.

Este nivel incluye requisitos de calidad específicos, orientación al público universitario y un tono institucional adecuado que establezca las bases sobre las que se construirá el proyecto. La selección cuidadosa de este rol inicial fue esencial para lograr resultados que sonaran auténticamente colegiados en lugar de genéricos o comerciales.

En el Nivel 2: Definición de tarea, los parámetros específicos del usuario se traducen en instrucciones específicas que determinan la elección de cada forma para cumplir con los requisitos del modelo.

Por ejemplo, si el usuario selecciona "Tono: institucional formal", este nivel agrega instrucciones detalladas para un lenguaje académico preciso, evita contracciones y mantiene una voz objetiva. Nivel 3: Las restricciones estructurales definen el formato exacto requerido para un borrador, con especificaciones que varían según el tipo de contenido: las comunicaciones internas utilizan una estructura formal con un título y una firma formales, mientras que las publicaciones de blog adoptan un formato más narrativo con una introducción atractiva y preguntas para el lector.

Esta diferenciación estructural asegura que cada tipo de contenido se ajuste a las convenciones esperadas en el entorno universitario. Nivel 4: la integración del

contexto maneja de manera inteligente la información adicional proporcionada por el usuario, especialmente el texto extraído de los archivos adjuntos.

Cuando se descubre contexto adicional, este nivel proporciona una guía clara sobre cómo integrarlo: priorizando datos específicos sobre información general, proporcionando números y fechas precisos y dejando claro qué elementos del contexto son más relevantes para el tema principal.

Finalmente, Nivel 5: Pautas de calidad establecen estándares de calidad editorial como claridad sobre la complejidad innecesaria, alineación con la identidad institucional, evitación de sesgos, lenguaje inclusivo y una instrucción crítica para no incluir marcadores de posición como “[datos aquí]” sino integrar de manera confiable información cuando faltan datos específicos.

4.3.3. Cliente de API para conexión con el modelo

Figura 20

Cliente de API para conexión con modelo de inteligencia artificial.



Nota. Elaboración propia mediante herramienta de inteligencia artificial, 2026.

El cliente API implementado en foundryClient.js tiene un diseño robusto que prioriza la confiabilidad y la experiencia del usuario incluso en condiciones adversas de la red. El patrón de disyuntor se introdujo para evitar fallas en cascada: después de 3 fallas consecutivas en la

comunicación con el servicio Foundry, el sistema cambia automáticamente a un modo alternativo que utiliza datos de simulación local (datos simulados). Este modelo no solo protege el sistema de la sobrecarga de un servicio externo que pueda experimentar problemas, sino que también garantiza una experiencia de usuario fluida al mostrar un borrador de muestra con un informe claro del uso de datos simulado mientras se restablece la conexión.

El mecanismo de retroceso exponencial controla de forma inteligente los intentos de conexión, comenzando con un retraso de 1 segundo después del primer fallo, duplicándose a 2 segundos después del segundo intento fallido, luego 4 segundos y finalmente 8 segundos antes de disparar el disyuntor. Esta estrategia avanzada le da al servicio externo tiempo para recuperarse de problemas temporales sin abrumarlo con solicitudes continuas, al tiempo que reduce la latencia del usuario cuando los problemas son breves.

Cada reintento incluye un registro detallado que ayuda a diagnosticar si los problemas son de red, autenticación, límites de velocidad o disponibilidad del servicio, lo que le permite realizar ajustes de configuración específicos.

El sistema de monitoreo registra métricas de rendimiento detalladas para cada interacción con la API, incluida la latencia de respuesta, la cantidad de tokens utilizados en solicitudes y respuestas, códigos de estado HTTP y tiempo de reintento. Estos registros están estructurados en formato JSON para facilitar el análisis y se muestran en la consola del desarrollador en el modo de desarrollo, lo que proporciona información valiosa sobre la optimización.

Por ejemplo, después de que se descubrió que las indicaciones muy largas se correlacionaban con tiempos de respuesta elevados, se introdujo el truncamiento inteligente, que preserva la estructura esencial al tiempo que reduce los marcadores innecesarios.

Este monitoreo continuo nos permitió afinar la creación no solo de clientes API, sino también de avisos, creando un sistema que se optimiza de forma iterativa en función de datos de rendimiento reales.

Figura 21

Implementación del cliente para conexión con el modelo en Microsoft Foundry

```
1  export class FoundryClient {
2    constructor(endpoint, apiKey) {
3      this.endpoint = endpoint;
4      this.headers = {
5        'Content-Type': 'application/json',
6        'api-key': apiKey
7      };
8    }
9
10   async generateContent(prompt, options = {}) {
11     const payload = {
12       model: process.env.VITE_FOUNDRY_MODEL_NAME,
13       messages: [
14         { role: 'system', content: 'Eres un redactor universitario...' },
15         { role: 'user', content: prompt }
16       ],
17       temperature: options.temperature || 0.7,
18       max_tokens: options.max_tokens || 2000,
19       top_p: 0.9,
20       frequency_penalty: 0.1
21     };
22
23     const response = await fetch(this.endpoint, {
24       method: 'POST',
25       headers: this.headers,
26       body: JSON.stringify(payload)
27     });
28
29     return this.handleResponse(response);
30   }
31 }
```

4.3.4. Módulo de salida (Module Output)

Figura 22

Módulo de Salida (ModuleOutput) en arquitectura de sistema.



Nota. Elaboración propia mediante herramienta de inteligencia artificial, 2026.

ModuleOutput fue diseñado como un componente de React que gestiona la presentación profesional de borradores generados por IA, transformando una respuesta de texto sin formato en una interfaz interactiva con varias funciones fáciles de editar. El componente maneja cuatro estados principales: carga (mientras se genera el borrador), error (cuando ocurren problemas de conexión o procesamiento), vacío (antes de la primera generación) y exitoso (mostrando el borrador completo).

Cada modo tiene su propia representación visual utilizando los componentes de la interfaz de usuario de Chakra: un cortador animado con un mensaje contextual durante la carga, advertencias de error con la opción de volver a intentarlo, un mensaje de bienvenida cuando no hay contenido y un panel de herramientas de edición completo cuando el borrador está listo.

Esta clara diferenciación de modos guía al usuario a través del flujo de generación con comentarios inmediatos y comprensibles.

Las características principales de ModuleOutput incluyen un sistema de visualización responsivo con controles de tamaño de fuente (pequeña, normal, grande, extra grande), modo de visualización (vista previa formateada frente a texto sin formato) y funcionalidad de edición en vivo utilizando el área de texto incorporada. Los borradores se muestran con espaciado tipográfico optimizado para una lectura extendida, conservan los saltos de línea y la estructura identificada en el texto generado y aplican estilos personalizados a encabezados, listas y resaltados usando CSS-in-JS de Chakra UI.

Además, el componente calcula y muestra estadísticas en tiempo real: el número de palabras, caracteres y párrafos, proporcionando mediciones objetivas de la longitud y densidad del contenido generado, útiles para ajustar los parámetros de longitud en generaciones futuras.

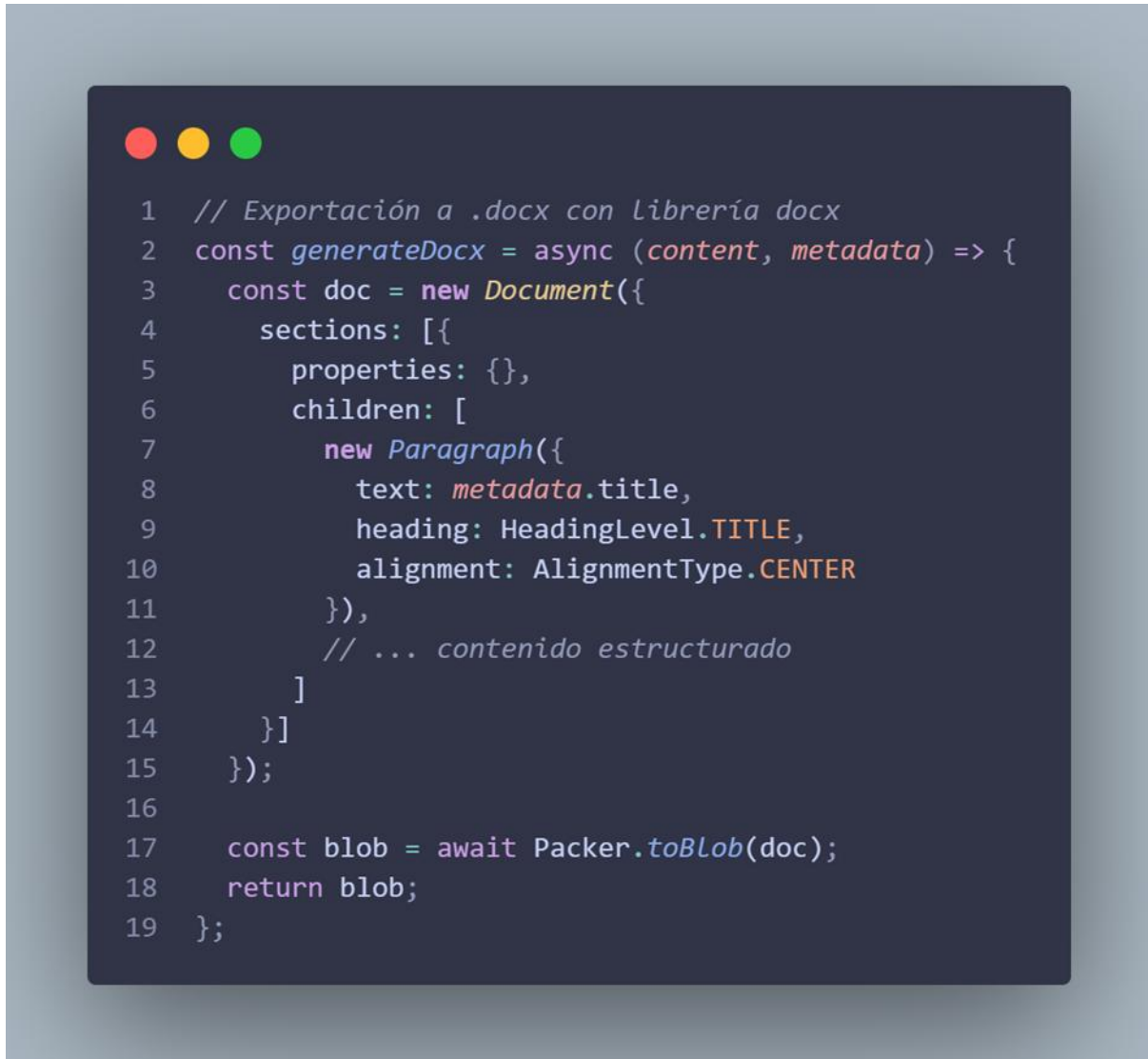
Para aumentar la usabilidad de su flujo de trabajo de edición, ModuleOutput incluye herramientas avanzadas de exportación y administración: botones para copiar al portapapeles (con confirmación visual), descargar como un archivo .txt con nombre según la fecha y el tipo de contenido, y exportar como un documento .docx con formato profesional utilizando la biblioteca docx. También implementa un sistema de historial local que guarda automáticamente los últimos 5 borradores generados (con marca de tiempo y vista previa), lo que le permite restaurar versiones anteriores sin reconstruir nuevamente.

El botón "Regenerar" le permite crear variantes del mismo borrador conservando los parámetros originales, mientras que el panel de depuración plegable (visible solo durante el desarrollo) muestra el mensaje exacto enviado al modelo, lo que

facilita el perfeccionamiento continuo del mensaje técnico en función de resultados del mundo real.

Figura 23

Módulo de salida. Función de exportación a formato docx



```
1 // Exportación a .docx con Librería docx
2 const generateDocx = async (content, metadata) => {
3   const doc = new Document({
4     sections: [{
5       properties: {},
6       children: [
7         new Paragraph({
8           text: metadata.title,
9           heading: HeadingLevel.TITLE,
10          alignment: AlignmentType.CENTER
11        }),
12        // ... contenido estructurado
13      ]
14    }]
15  });
16
17  const blob = await Packer.toBlob(doc);
18  return blob;
19 };
```

4.3.5. Análisis comparativo del prototipo

1. Especialización de Dominio (Domain-Specific Fine-Tuning)

Prompt Engineering: Este prototipo implementa prompts estructurados con más de 530 líneas de lógica específica para comunicación universitaria UCSG, mientras que ChatGPT/Claude genéricos dependen de prompts genéricos del usuario, y otros generadores usan templates predefinidos rígidos.

Contexto Institucional: El sistema tiene hardcodeada la identidad UCSG, contactos institucionales, terminología específica y restricciones de nombres propios. Las alternativas carecen de este contexto o requieren configuración manual.

Diferencia técnica clave: El sistema inyecta automáticamente contexto institucional embebido que garantiza coherencia de marca sin intervención del usuario.

2. Arquitectura de Prompt Engineering Multi-Capa

El sistema implementa una construcción dinámica de prompts con 7 capas diferenciadas:

- Capa A - Rol y expertise: Define 8+ años de experiencia contextualizada
- Capa B - Tarea específica: Parámetros dinámicos del formulario
- Capa C - Especificaciones: Tono adaptativo con 4 variantes
- Capa D - Información específica: Array dinámico de puntos clave
- Capa E - Elementos a evitar: Control negativo de estilo
- Capa F - Temporalidad automática: Sistema único de fechas contextuales (resuelve problema crítico de LLMs)
- Capa G - Estructuras por tipo: 5 templates estructurales diferentes

3. Procesamiento de Contexto Externo con Sanitización

Límite de tokens: 5000 caracteres (~1250 tokens) para no exceder contexto del modelo

Integración inteligente: Instrucciones específicas para contextualizar sin copiar literalmente

4. Sistema de Estructuras Diferenciadas por Tipo de Contenido

El sistema implementa 5 plantillas estructurales distintas:

- Comunicado interno: 7 secciones desde Encabezado hasta Firma UCSG

- Noticia institucional: Lead periodístico + pirámide invertida
- Entrada de blog: Título + Subtítulos + Pregunta al lector
- Resumen de evento: Info básica → Impacto → Próximos eventos
- Anuncio: Titular directo → Llamado a acción estructurado

5. Integración con Microsoft Foundry (API Propietaria)

Backend: Microsoft Foundry (Azure) vs OpenAI/Anthropic directo en alternativas

Manejo de respuestas: 5 patrones de parsing diferentes para máxima compatibilidad

Privacidad: Datos procesados en infraestructura Azure controlada, no en servidores externos

6. Exportación Profesional a DOCX

El sistema incluye generación de documentos Word estructurados con la librería docx:

- Formato nativo: .docx con estilos (vs copy-paste manual en ChatGPT)
- Estructura: Headings jerárquicos automáticos (vs texto plano)
- Metadatos: Incrustados en documento (vs sin metadatos)
- Uso inmediato: Listo para revisión editorial (vs requiere formateo)

7. Sistema de Identidad Visual Institucional

La interfaz está brandeada nativamente con colores institucionales UCSG:

8. Restricciones Inteligentes de Identidad

El sistema previene alucinaciones de nombres ficticios mediante instrucciones explícitas:

Instrucción embebida: "NO inventes ni uses nombres propios de personas (Decanos, Directores, Coordinadores). Usa siempre los cargos institucionales (ej. 'El Decanato', 'La Dirección de Carrera', 'El Rectorado')."

Tabla 7
Resumen de Diferenciadores Técnicos

Diferenciador	ChatGPT	Este Prototipo
Prompt especializado UCSG engineering	✗	✓
Control temporal automático	✗	✓
5 estructuras por tipo contenido	✗	✓
Exportación DOCX nativa	✗	✓
Branding institucional embebido	✗	✓
Restricciones de alucinación	✗	✓
Procesamiento archivos adjuntos	Manual	✓
Backend Azure/Foundry privado	✗	✓
Costo por uso	Por token	Controlado

Conclusión Técnica

1. Especialización de dominio: No es un wrapper de ChatGPT, sino un sistema con lógica de negocio embebida
2. Coherencia institucional garantizada: Contexto UCSG inyectado automáticamente
3. Control de calidad editorial: Estructuras predefinidas por tipo de contenido

4. Prevención de errores comunes: Sistema de temporalidad y restricción de nombres ficticios

5. Integración de flujo de trabajo: Exportación directa a formato profesional (.docx)

6. Privacidad y control: Backend en infraestructura Azure vs. APIs públicas

Tabla 8
Tabla por Token

Modelo	Precio Estándar por millón de tokens
GPT-4.1-14-04-2025 Global	Entrada: \$2 Entrada en caché: \$0.50 Salida: \$8
GPT-5 2025-08-07 Global	Entrada: \$1.25 Entrada en caché: \$0.13 Salida: \$10
GPT-5.1 Global	Entrada: \$1.25 Entrada en caché: \$0.13 Salida: \$10
GPT-5.2 Global	Entrada: \$1.75 Entrada en caché: \$0.18 Salida: \$14

CONCLUSIONES

Se puede concluir que los chatbots de IA generativa son herramientas de apoyo, que no sustituyen el criterio humano ni la validación que se le debe dar. Es de mucha utilidad y valor, pero se deben completar ciertos procesos de validación, bajo una supervisión crítica y responsable del ser humano.

El sistema logra su objetivo principal de reducir significativamente el tiempo requerido para la fase de borrador inicial, transformando parámetros editoriales estructurados en documentos coherentes y listos para revisión humana en cuestión de segundos. La arquitectura modular implementada ha demostrado ser robusta y sostenible, permitiendo iteraciones rápidas durante el desarrollo y estableciendo una base sólida para futuras expansiones.

La experiencia de desarrollo reveló que el éxito depende claramente de la calidad del prompt engineering. El modelo GPT-4.1 tendía a ignorar información contextual específica proporcionada por el usuario, prefiriendo datos genéricos de su entrenamiento. Este desafío se superó mediante una estructura de prompts jerárquica con instrucciones explícitas, posicionamiento estratégico del contexto y mecanismos de priorización claros.

El diseño propuesto es una solución ligera, las cuales luego pueden crecer hacia integraciones más avanzadas, por ejemplo publicaciones automáticas en redes sociales o llegar a personalizar por usuario. El sistema combina simplicidad inicial con potencial expansión, lo que asegura que el usuario pueda ingresar prompts rápidas y efectivas.

Finalmente, el proceso de pruebas de funcionalidad y la evaluación cualitativa con varios usuarios fueron muy importantes ya que validan la parte técnica y la experiencia real del uso del sistema, asegurando que la aplicación sea estable, ágil y confiable. Además, la evaluación permite verificar que los borradores generados sean percibidos como útiles, como el tipo de contenido, el tono, y público objetivo solicitado, es decir adecuados al contexto, para confirmar que el contenido aporta valor a los usuarios.

RECOMENDACIONES

Basado en la implementación y pruebas realizadas, se recomienda a las instituciones interesadas en adoptar esta tecnología comenzar con un enfoque similar de Producto Mínimo Viable, priorizando la integración con infraestructura cloud empresarial como Microsoft Azure por sus ventajas en seguridad, gobernanza de datos y compatibilidad con cuentas institucionales. El uso de Microsoft Foundry con el modelo GPT-4.1 demostró ser particularmente adecuado para el contexto universitario, ofreciendo un balance óptimo entre capacidades, costo y controles administrativos. Se sugiere establecer desde el inicio protocolos claros de revisión humana y marcas de "Generado con IA" en todos los borradores, no solo por transparencia ética sino también para mantener la responsabilidad editorial final en manos humanas, posicionando la tecnología como asistente más que como reemplazo.

Para futuras iteraciones del sistema, se identifican tres áreas prioritarias de desarrollo: implementar un sistema de fine-tuning con ejemplos reales de comunicación universitaria para especializar aún más las respuestas del modelo, añadir integración con sistemas de gestión de contenido (CMS) utilizados comúnmente en universidades para crear un flujo de trabajo completo desde la generación hasta la publicación, y desarrollar capacidades de análisis de sentimiento y ajuste de tono automático que permitan adaptar el contenido a diferentes audiencias y canales de comunicación. Estas mejoras transformarían el prototipo actual en una plataforma integral que no solo acelera la creación inicial de contenido, sino que también eleva su calidad estratégica mediante insights basados en datos.

REFERENCIAS BIBLIOGRAFICAS

- Aguilar, A. (2024, 23 de agosto). Inteligencia artificial en la educación superior: Aplicaciones y ejemplos [Fotografía]. Timely. <https://time.ly/es/blog/inteligencia-artificial-en-la-educacion-superior-aplicaciones-y-ejemplos/>
- Alonso, S., Kalinowski, M., Ferreira, B., Barbosa, S. D. J., & Lopes, H. (2023). A systematic mapping study and practitioner insights on the use of software engineering practices to develop MVPs. arXiv. <https://arxiv.org/abs/2305.08299>
- Amershi, S., Weld, D. S., Vorvoreanu, M., Fourney, A., Nushi, B., Collisson, P., & Horvitz, E. (2019). Guidelines for human–AI interaction. Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, 1–13.
- AsifaliAlimali Sayed, & Patel, M. O. S. (2023). The influence of JavaScript on website development and user experience. IJRASET Journal for Research in Applied Science and Engineering Technology. <https://doi.org/10.22214/ijraset.2023.56284>
- Bommasani, R., Hudson, D., Adeli, E., Altman, R., Arora, S., von Arx, S., & Liang, P. (2021). On the opportunities and risks of foundation models. Stanford Institute for Human-Centered Artificial Intelligence.
- Britannica Editors. (2025). JavaScript: History, applications, & facts. Encyclopædia Britannica. <https://www.britannica.com/technology/JavaScript>
- Day, R. A., & Gastel, B. (2017). How to write and publish a scientific paper (8th ed.). Cambridge University Press.
- Fernández, O. (2025, 17 de agosto). Microservicios [Fotografía]. Aprender Big Data. <https://aprenderbigdata.com/microservicios/>
- Floridi, L., & Cows, J. (2019). A unified framework of five principles for AI in society. Harvard Data Science Review, 1(1).
- Fowler, M. (2020). Monoliths versus microservices. ThoughtWorks.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.
- Greetham, B. (2020). Academic writing: Skills and strategies. Bloomsbury Publishing.
- Guerra, M. (2024, 20 de febrero). Principios éticos en la educación con inteligencia artificial [Fotografía]. Prevención Integral. <https://www.prevencionintegral.com/actualidad/noticias/2024/02/14/principios-eticos-educacion-con-inteligencia-artificial-ia>
- IA, N. (2024, 9 de abril). Exploring the leading open-source large language models of 2024: Innovations and applications [Fotografía]. Novita. <https://blogs.novita.ai/es/exploring-the-leading-open-source-large-language-models-of-2024-innovations-and-applications/>
- Iglesias, L. F. (2025, 26 de marzo). La inteligencia artificial generativa: Creando como seres humanos [Fotografía]. Semanario Universidad. <https://semanariouniversidad.com/suplementos/forja/la-inteligencia-artificial-generativa-creando-como-seres-humanos/>
- Inteligencia artificial. (2024, 14 de mayo). ¿Estamos listos para el próximo nivel en la interacción con la inteligencia artificial? [Fotografía]. Consultoría Uno.

<https://www.consultoria.uno/blog/estamos-listos-para-el-proximo-nivel-en-la-interaccion-con-la-inteligencia-artificial-asi-parece>

- Kambhampati, S. (2023). GenAI: The rise of generative artificial intelligence. *AI Magazine*, 44(3), 201–210.
- Kelly, A. (2024, abril). What is prompt engineering? An overview of prompt engineer [Fotografía]. Euphoria XR. <https://euphoriaxr.com/what-is-prompt-engineering-an-overview-of-prompt-engineer/>
- Kinsta. (2024). Bibliotecas de componentes React: Mejores herramientas UI como Chakra UI. Kinsta Blog. <https://kinsta.com/es/blog/bibliotecas-de-componentes-react/>
- Lee, J., Ahn, J., & Yi, K. (2025). React-tRace: A semantics for understanding React hooks. arXiv. <https://arxiv.org/abs/2507.05234>
- Liu, P., Yuan, W., Fu, J., Jiang, Z., & Ma, S. (2023). Pre-train, prompt, predict: A systematic survey of prompting methods. *ACM Computing Surveys*, 55(9).
- Microsoft. (2024). Azure OpenAI & Copilot API documentation. Microsoft Learn.
- Microsoft Corporation. (2025). Microsoft Foundry [Plataforma de IA generativa]. Microsoft Azure. <https://azure.microsoft.com/es-es/products/ai-foundry>
- Microsoft Corporation. (2026). GPT-4.1 – Azure AI Model Catalog. Microsoft Azure. <https://ai.azure.com/catalog/models/gpt-4.1>
- Mitrović, J., Glavaš, G., & Šnajder, J. (2023). Hallucinations in large language models: A survey. *AI Review*.
- Mozilla Developer Network (MDN). (2025). ¿Qué es JavaScript? https://developer.mozilla.org/es/docs/Learn_web_development/Core/Scripting/What_is_JavaScript
- OpenAI. (2026). *Imagen generada mediante inteligencia artificial sobre Inteligencia Artificial Generativa*. ChatGPT. <https://www.openai.com>
- O'Reilly Media. (2023). Designing APIs for AI-driven applications. O'Reilly Media.
- Puiggros, V. (2025). APIs en inteligencia artificial [Fotografía]. *Inteligencia Artificial*. <https://inteligenciartificial.me/wiki/apis-en-ia/>
- Richards, M. (2015). *Software architecture patterns*. O'Reilly Media.
- Russell, S., & Norvig, P. (2021). *Artificial intelligence: A modern approach* (4th ed.). Pearson.
- Scoble, R., & Israel, S. (2023). *The AI revolution in content creation*. Wiley.
- Shneiderman, B. (2020). Human-centered AI. *Interactions*, 27(4), 68–71.
- Smith, J. (2022). Artificial intelligence in editorial workflows. *Publishing Research Quarterly*, 38, 567–580.
- UNESCO. (2021). *Recommendation on the ethics of artificial intelligence*. UNESCO Publishing.
- UNESCO. (2023). *AI and education: Guidance for policy-makers*. UNESCO Publishing.

- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.
- White, J., Li, S., Williams, J., Kuehl, N., & Nichols, W. (2023). Prompt engineering: A guide to always getting the best from AI models. *Communications of the ACM*, 67(1), 56–64.
- Wooldridge, M. (2021). *An introduction to multi-agent systems* (2nd ed.). Wiley.
- Xu, W., Guo, Y., & Lin, C. (2023). Evaluating the quality of AI-generated text. *Journal of Artificial Intelligence Research*, 76, 133–159.

ANEXOS

Anexo 1 Modelo del Prototipo

Figura 24
Estructura del prototipo



Figura 25
Llenado los parámetros para generar

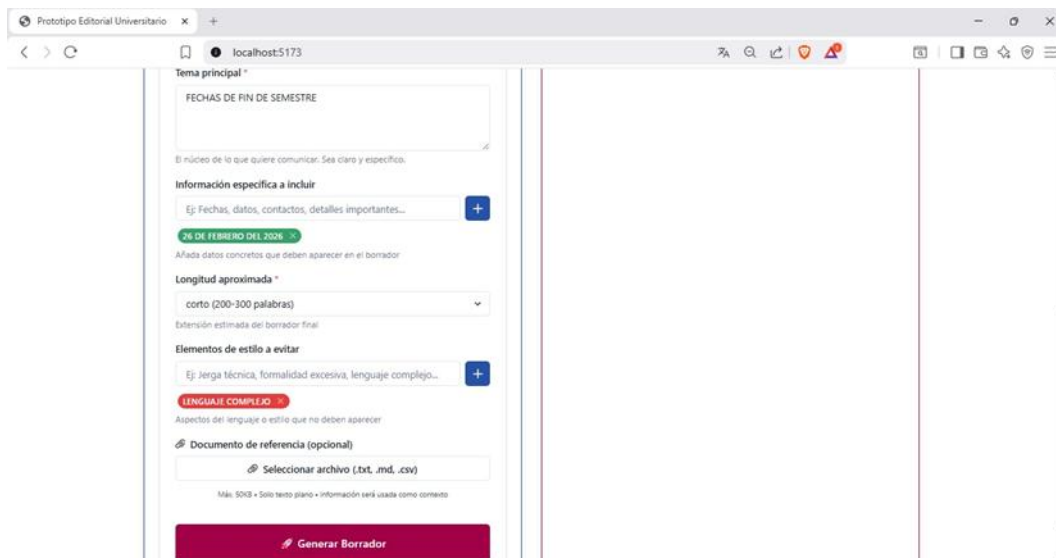


Figura 26
Se genera la información que se requiere



Para cumplir nuestro cuarto objetivo se realizó una entrevista

- 1. ¿Que le pareció al resultado de la información que buscaba?
- 2. ¿La aplicación es fácil de manejar es decir si es intuitiva?
- 3. ¿Recomendaría la herramienta a otros estudiantes o docentes ?
- 4. ¿Que parte le pareció más interesante del prototipo si se ajustó al contexto universitario?



DECLARACIÓN Y AUTORIZACIÓN

Nosotros, **Pérez Mayorga, Jean Carlos**, con C.C: # **0941265621** y **Morocho Macías, Julio César**, con C.C # 0705985513 autores del trabajo de integración curricular, **Desarrollo de un prototipo funcional de un agente de IA Generativa especializado en la creación de borradores de contenido editorial universitario**, requerido para la obtención del título de **Ingeniero en Ciencias de la Computación** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaramos tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizamos a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, 03 de marzo del 2026

f. _____
Pérez Mayorga, Jean Carlos
C.C: 0941265621

f. _____
Morocho Macías, Julio César
C.C: 0705985513



REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA

FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN

TEMA Y SUBTEMA:	Desarrollo de un prototipo funcional de un agente de IA Generativa especializado en la creación de borradores de contenido editorial universitario.		
AUTOR(ES)	Pérez Mayorga, Jean Carlos Morocho Macías, Julio César		
REVISOR(ES)/TUTOR(ES)	Castañeda Clark, María Del Carmen		
INSTITUCIÓN:	Universidad Católica de Santiago de Guayaquil		
FACULTAD:	Ingeniería		
CARRERA:	Computación		
TÍTULO OBTENIDO:	Ingeniero en Ciencias de la Computación		
FECHA DE PUBLICACIÓN:	03 de marzo del 2026	No. DE PÁGINAS:	59 p
ÁREAS TEMÁTICAS:	Inteligencia artificial, Educación superior, Gestión de la información, Desarrollo de software.		
PALABRAS CLAVES/ KEYWORDS:	<i>Inteligencia artificial generativa, agente de IA, contenido editorial universitario, prototipo funcional.</i>		
RESUMEN/ABSTRACT:	El proyecto propone el desarrollo de un prototipo funcional de un agente de IA Generativa especializado en la creación de borradores de contenido editorial universitario. A diferencia de herramientas generales como los distintos chatbox que existen en internet, tales como ChatGPT, Claude, Deepseek, entre otras, este agente estará contextualizado para las necesidades comunicacionales de unidades académicas. La solución busca automatizar la construcción de prompts especializados, asegurando coherencia en tono, estilo institucional y eficiencia en la generación de contenido. El prototipo consistirá en una aplicación web con una interfaz simple donde el usuario ingrese parámetros como tema, tono y público efectivo. El sistema procesará estos datos, generará instrucciones estructuradas y se integrará con la API de un modelo de IA generativa para producir borradores de posts o comunicados. Finalmente, se realizarán pruebas con usuarios para validar la calidad y utilidad del sistema.		
ADJUNTO PDF:	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO	
CONTACTO CON AUTOR/ES:	Teléfono: +593-986432389 +593-997892531	Correo: E-mail: jean.perez@cu.ucsg.edu.ec E-mail: julio.morocho@cu.ucsg.edu.ec	
CONTACTO CON LA INSTITUCIÓN (COORDINADOR DEL PROCESO UTE)::	Toala Quimí, Edison José Teléfono: +593-990-976776 E-mail: edison.toala@cu.ucsg.edu.ec		
SECCIÓN PARA USO DE BIBLIOTECA			
Nº. DE REGISTRO (en base a datos):			
Nº. DE CLASIFICACIÓN:			
DIRECCIÓN URL (tesis en la web):			