UNIVERSIDAD LAICA “ELOY ALFARO” DE MANABÍ

EXTENSIÓN EN EL CARMEN

CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

Creada Ley No. 10 – Registro Oficial 313 de noviembre 13 de 1985

PROYECTO INTEGRADOR

PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

SISTEMA INFORMÁTICO CON PLN PARA LA GESTIÓN DE GUIONES AUDIOVISUALES EN SOFTMEDIA ULEAM EL CARMEN

VALENCIA HURTADO JERSON PATRICIO

AUTOR

ING. JEFFERSON OMAR ARCA ZAVALA MG.

TUTOR

EL CARMEN, 20 AGOSTO 2026

TRIBUNAL DE SUSTENTACIÓN

Uleam

Extensión El Carmen

TRIBUNAL DE SUSTENTACIÓN

Título del trabajo de titulación:

Sistema Informático con PLN para la Gestión de Guiones Audiovisuales en Softmedia Uleam El Carmen.

Modalidad:

Proyecto Integrador

Autor:

Valencia Hurtado Jerson Patricio

Tutor:

Ing. Arca Zavala Jefferson Omar, Mgtr

Tribunal de Sustentación

- Presidente A. S. Minaya Macias Renelmo Wladimir, Mgtr.
- Miembro Ing. Pozo Hernández Clara Guadalupe, Mgtr.
- Miembro Ing. Angel Wilson Villareal Cobeña, Mgtr

Fecha de sustentación:

20 agosto 2026

DECLARACIÓN DE AUTORÍA

UNIVERSIDAD LAICA "ELOY ALFARO" DE MANABÍ
EXTENSIÓN EN EL CARMEN

DECLARACIÓN DE AUTORÍA

La responsabilidad del contenido de este Trabajo de titulación, cuyo tema es:

Sistema Informático con PLN para la Gestión de Guiones Audiovisuales en Softmedia Uleam el Carmen, corresponde exclusivamente a: **Valencia Hurtado Jerson Patricio** con **C.I. 1351260748** y los derechos patrimoniales de la misma corresponden a la Universidad Laica "Eloy Alfaro" de Manabí.

Valencia Hurtado Jerson Patricio

C.I. 1351260748

DEDICATORIA

Dedico este trabajo de titulación, en primer lugar, a Dios, por darme la fortaleza, la sabiduría y la perseverancia necesarias para llegar hasta este momento. Por acompañarme durante estos años de estudio y permitirme superar cada dificultad que se presentó en el camino.

A mis padres, **Liliana Hurtado y Patricio Valencia**, por su apoyo incondicional durante todos estos años de estudios. Gracias por estar presentes en cada etapa de mi formación, por sus consejos, por sus sacrificios y por confiar en mí incluso cuando las cosas no fueron fáciles. Este logro también es de ustedes, porque detrás de cada meta alcanzada existe el esfuerzo y el respaldo de quienes siempre estuvieron a mi lado.

A mis mejores amigos, porque no hace falta mencionar sus nombres para que sepan lo importantes que han sido para mí. Gracias por estar en los buenos momentos y, sobre todo, por acompañarme en aquellos días en los que las cosas se hicieron más difíciles. Por las conversaciones, las risas, los consejos y por hacer que este camino universitario fuera mucho más llevadero.

A mis familiares, por sus palabras de ánimo, su cariño y por estar presentes de una u otra manera durante este proceso. Cada muestra de apoyo fue importante para seguir adelante y no perder de vista el objetivo. Finalmente, me dedico también estas palabras a mí mismo, por no rendirme, por continuar incluso en los momentos de cansancio y por haber llegado hasta aquí después de tantos años de esfuerzo, aprendizaje y sacrificios. Este trabajo representa el cierre de una etapa importante de mi vida y, al mismo tiempo, el inicio de nuevos retos y oportunidades.

Este logro es el resultado de muchos esfuerzos, experiencias y personas que dejaron una huella en mi camino. A todos ellos, mi más sincero agradecimiento.

Jerson Patricio Valencia Hurtado

AGRADECIMIENTO

Agradezco a todas las personas que, de una u otra manera, aportaron para que pudiera llegar hasta este momento. A mi familia y a mis mejores amigos, quienes saben lo que significó este camino y estuvieron presentes cuando las cosas no fueron fáciles.

También agradezco a quienes me brindaron una palabra de apoyo, un consejo o simplemente estuvieron ahí en los momentos en que más lo necesitaba. Cada experiencia, dificultad y aprendizaje fue parte de este proceso.

Llegar hasta aquí no fue sencillo, pero cada obstáculo me enseñó a seguir adelante. Este logro es una muestra de que, con perseverancia y sin miedo a intentarlo, las metas que alguna vez parecieron lejanas pueden convertirse en realidad.

A todos los que fueron parte de este camino, gracias.

Jerson Patricio Valencia Hurtado

"Caminante, no hay camino, se hace camino al andar"

Antonio Machado

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES

ÍNDICE DE TABLAS

ÍNDICE DE ANEXOS

RESUMEN

El Club SoftMedia de la ULEAM Extensión El Carmen enfrentaba retrasos significativos en la producción de material audiovisual institucional debido a la redacción manual de guiones, un proceso que consumía entre 4 y 8 horas por video y generaba sobrecarga en el equipo voluntario. Ante esta problemática, el presente proyecto integrador tuvo como objetivo desarrollar un módulo informático basado en Procesamiento de Lenguaje Natural (PLN), integrable al sistema web del club, que automatizara la generación de guiones estructurados en introducción, desarrollo y cierre. La investigación se desarrolló bajo un enfoque metodológico mixto de tipo concurrente, combinando entrevistas, observación directa y una encuesta aplicada a los 10 integrantes del club, lo que permitió diagnosticar el problema y sustentar la propuesta tecnológica. El módulo se construyó mediante la metodología ágil Scrum, organizada en cuatro Sprints, e implementó un modelo de lenguaje GPT-2 en español, ajustado (fine-tuned) con 120 ejemplos reales de guiones del club, expuesto como una API REST desarrollada en Python con FastAPI. Los resultados obtenidos en las pruebas de caja negra, caja blanca y en la prueba piloto evidenciaron una reducción superior al 99% en el tiempo de redacción, con un promedio de 40 segundos por guion generado, una tasa de éxito del 99% en las peticiones, una disponibilidad del 99.5% durante 72 horas continuas, una aceptación del 80% en la coherencia narrativa de los textos y una satisfacción del 90% entre los usuarios del club. Estos hallazgos confirman el cumplimiento del objetivo general planteado y demuestran que la incorporación de PLN constituye una solución viable, replicable y de bajo costo para optimizar la comunicación institucional en entornos universitarios con recursos limitados.

ABSTRACT

The SoftMedia Club at ULEAM's El Carmen Extension faced significant delays in producing institutional audiovisual content due to the manual drafting of scripts, a process that took between 4 and 8 hours per video and overloaded the volunteer team. In response to this problem, the present integrative project aimed to develop a software module based on Natural Language Processing (NLP), integrable into the club's web system, capable of automating the generation of structured scripts divided into introduction, development, and closing sections. The research followed a mixed-methods, concurrent approach, combining interviews, direct observation, and a survey applied to the club's 10 members, which supported the diagnosis of the problem and the technological proposal. The module was built using the Scrum agile methodology across four Sprints, and implemented a Spanish-language GPT-2 model fine-tuned with 120 real scripts produced by the club, exposed as a REST API developed in Python with FastAPI. Results from black-box testing, white-box testing, and the pilot trial showed a reduction of more than 99% in drafting time, with an average of 28 seconds per generated script, a 99% request success rate, 99.5% availability over 72 continuous hours of testing, 80% acceptance of narrative coherence, and 90% user satisfaction among club members. These findings confirm that the general objective was achieved and demonstrate that incorporating NLP is a viable, replicable, and low-cost solution for optimizing institutional communication in university settings with limited resources.

CAPÍTULO I

1 INTRODUCCIÓN

1.1 Introducción

El presente proyecto se desarrolla en la ULEAM Extensión El Carmen, específicamente en el Club SoftMedia, encargado de la producción y difusión de contenido audiovisual de la extensión, dando cobertura a diferentes actividades académicas, culturales y sociales que se llevan a cabo en la institución. Dicho contenido es difundido en diversas redes sociales para lograr mayor alcance y repercusión.

Actualmente, la redacción de guiones para los contenidos audiovisuales se realiza de manera manual, lo que conlleva una pérdida considerable de tiempo por parte del equipo que conforma SoftMedia. Esta situación genera ineficiencia, debido a que el club cumple funciones similares a un departamento de comunicación, realizando coberturas de distintos eventos socioculturales con herramientas inadecuadas para un trabajo de reportaje y planificación narrativa.

Ante esta situación, se plantea la implementación de un sistema basado en técnicas de Procesamiento de Lenguaje Natural (PLN) para la escritura automática de guiones, optimizando así el tiempo de producción por parte de los integrantes de SoftMedia y mejorando así la calidad de los diferentes contenidos audiovisuales.

1.2 Presentación del tema

Sistema informático con PLN para la gestión de guiones audiovisuales en SoftMedia Uleam El Carmen.

1.3 Ubicación y contextualización de la problemática

El presente proyecto se desarrolla en la Uleam Extensión El Carmen en las inmediaciones del Club SoftMedia, dicho Club es el encargado del contenido audiovisual de la extensión difundiendo las diferentes actividades académicas, culturales y sociales que se llevan acabó en la institución, cuyo contenido es difundido en las diferentes redes sociales para su repercusión.

Actualmente la redacción de los guiones para los diferentes contenidos audiovisuales se lo lleva a cabo de manera manual, lo que conlleva una gran pérdida de tiempo en redacción por parte del equipo que conforma SoftMedia, lo que genera una ineficiencia, ya que el Club cumple la función de un departamento de comunicación informal, realizando las coberturas de

diferentes eventos socio culturales que se lleven acabó en las actividades de la institución con las herramientas inadecuadas para este trabajo de reportaje, Ante esta situación de faltas de herramientas se plantea la implementación de un sistema basado en técnicas de Procesamiento de Lenguaje Natural (PLN) para la escritura de guiones de manera automática optimizando asi el tiempo de producción por parte de los integrantes de SoftMedia, mejorando asi la calidad de la producción de los diferentes contenidos audiovisuales.

El Club SoftMedia integrado principal mente por estudiantes voluntarios de las carreras de Ingeniería en Tecnologías de la Información y la carrera de Ingeniería en Software cuyos integrantes son los responsables de la producción del contenido audiovisual de los diferentes actividades académicas, siendo asi los responsables de la difusión de varios videos al mes, cuya cifra de videos producidos se ve afectada por el retraso en la escritura de guiones y as su vez la publicaicon en las diferentes redes sociales, adema en un contexto regional como Manabí donde la Uleam Extención El Carmen brinda sus servicios a las diferentes localidades del cantón, las cuales no todas cuentan con el acceso a los recurso digitales, siendo asi SoftMedia el principal promotor del contenido audiovisual pero las limitaciones de escritura manual de estos guiones limitando asi su cobertura, lo que resalta la urgencia de implementar diferentes herramientas tecnológicas para la escritura manual.

Pese a que el Club SoftMedia pudiera interpretarse como un caso singular, su situación se inscribe en una tendencia más amplia: la brecha digital que enfrentan las instituciones de educación superior ubicadas fuera de los grandes centros urbanos de la región. La integración de TIC en América Latina, en efecto, ha sido dispareja, siendo las extensiones universitarias y sus extenciones las que sufren mayores limitaciones en infraestructura, capacitación humana y recursos económicos para una adopción digital continuad.

La brecha entre sedes matrices y extensiones es una realidad en Ecuador, el Plan Nacional de Educación Digital 2023-2027 apuesta por el fortalecimiento de competencias digitales en la educación superior y por la adopción progresiva de tecnologías emergentes, como la inteligencia artificial y el procesamiento de lenguaje natural, para reducir esas diferencias

Ofrecer una solución replicable para otras extensiones universitarias con contextos similares y contribuir a reducir la brecha digital regional es el propósito que articula la propuesta de automatización de guiones con PLN en SoftMedia. Si bien la transformación digital suele asociarse a grandes inversiones en infraestructura, el caso de la Uleam Extensión

El Carmen muestra que lo que realmente hace falta son soluciones tecnológicas contextualizadas y sostenibles. Esta iniciativa cobra un valor especial precisamente allí, donde el acceso a recursos digitales es limitado y la producción audiovisual debe sostenerse sin la infraestructura de las sedes matrices.

1.4 Planteamiento del problema

1.4.1 Problematización

En el Club SoftMedia de la Uleam extensión El Carmen, existe un flujo constante de producción de videos para cubrir eventos académicos, culturales y sociales, en el cual participan estudiantes voluntarios pero la escritura manual de guiones es lenta e ineficiente, lo que conlleva a retrasos en la entrega de contenidos y sobrecarga en el equipo de comunicación, así mismo como la disminución en la frecuencia de publicaciones audiovisuales, esto torna un ambiente de incertidumbre en la difusión institucional ya que no se cuenta con herramientas automatizadas que ayuden a este punto clave para la universidad.

Estas limitaciones aparte de afectar en tiempos de coberturas de actividades académicas desarrolladas en la institución, también limita las capacidades de SoftMedia para redactar guiones innovadores para el contenido audiovisual que capten la atención del público universitario.

A todo esto se suma la presión constante entre los integrantes de SoftMedia para entregar material audiovisual que cumpla con las diferentes expectativas en torno a la calidad del material, atrapando al equipo en un bucle de trabajo repetitivo de trabajo lo que priva al equipo de creatividad y una falta de compromiso que se verá a largo plazo.

1.4.2 Génesis del problema

La problemática surgio con la apertura del Club SofteMedia, cuando el material audiovisual era poco frecuente y los guiones eran improvisados. Con el aumento significativo de eventos académicos la demanda de material audiovisual aumento, generando una sobre carga de trabajo a los integrantes del Club, la falta de recursos y el poco conocimiento de los integrantes en material audiovisual genero un flujo de trabajo ineficiente principalmente por la necesidad de priorizar otras actividades académicas.

En principios SoftMedia tenía un papel fundamental en las coberturas para eventos básicos, pero su rápido crecimiento a nivel institucional aumenta la demanda de producción en un 70%, aumentando la necesidad de implementar en SoftMedia herramientas tecnológicas

para la gestión del contenido, ya que la gestión manual de dicho contenido afecta de manera negativa en la eficacia de producción por parte de SoftMedia.

Con el tiempo, la falta de actualización en procesos también afectó la motivación del equipo, ya que los voluntarios enfrentaban críticas por retrasos sin contar con herramientas para mejorar, lo que profundizó la desconexión entre esfuerzo y resultados.

1.4.3 Estado actual del problema

En la actualidad, aproximadamente el 60% del tiempo se dedica a escribir guiones a mano, lo que genera retrasos significativos en la producción de cada video. Con un equipo reducido (entre 5 y 7 redactores voluntarios), no se cuenta con herramientas de automatización, lo que resulta en una disminución de publicaciones mensuales.

Con el crecimiento constante de la demanda de contenido audiovisual en SoftMedia y la falta de organización, sumado a las actividades académicas que deben cumplir los integrantes del club, se genera una repercusión significativa en la lentitud de producción. Esto también produce inconsistencias en las narrativas (guiones), haciéndolas más repetitivas, lo que afecta la cobertura de eventos de importancia institucional y puede derivar en contenido de menor calidad. Como consecuencia, se obtiene una baja repercusión del contenido publicado, dejando a SoftMedia con menor capacidad de innovación.

1.5 Diagrama causa – efecto del problema

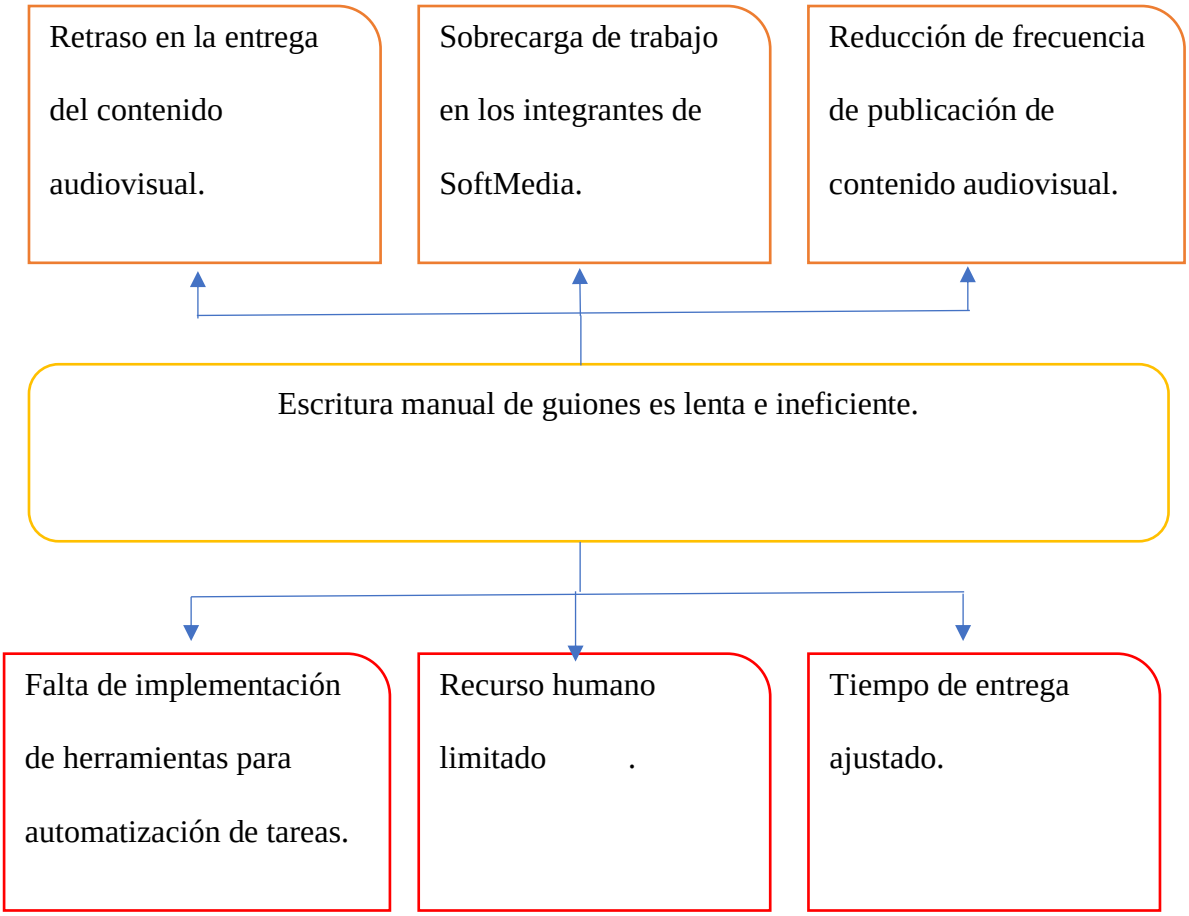


Ilustración 1 Diagrama causa-efecto del problema de investigación

1.6 Objetivos

1.6.1 Objetivo general

Desarrollar un módulo informático basado en Procesamiento de Lenguaje Natural (PLN), integrable al sistema web del Club SoftMedia de la Uleam El Carmen, que automatice la generación de guiones estructurados para los videos producidos por el club.

1.6.2 Objetivos específicos

- Diagnosticar la problemática actual en la producción de guiones audiovisuales del Club SoftMedia.
- Examinar y seleccionar técnicas de procesamiento de lenguaje natural (PLN) adecuadas para la generación automática de texto estructurado, revisando marcos teóricos, librerías y modelos disponibles que sean compatibles con contextos educativos de recursos limitados, con el propósito de fundamentar la propuesta tecnológica.

- Analizar el flujo de trabajo y redacción de guiones audiovisuales dentro del Club SoftMedia atreves de métodos de observación directa y encuestas internas a los miembros, con la finalidad de identificar los principales problemas en el trabajo diario dentro del Club, ayudando a establecer una base solidad para la aplicación de soluciones automatizadas.
- Diseñar y desplegar un módulo funcional integrando PLN para la escritura automatizada de guiones audiovisuales, integrado a la interfaz principal del sistema de SoftMedia atreves de API REST verificando su eficacia en los trabajos diarios de los miembros del Club SoftMedia.
- Evaluar la factibilidad del módulo en la optimización de los procesos de redacción de guiones audiovisuales en el Club SoftMedia atreves de pruebas piloto midiendo la optimización de tiempo de producción en el material audiovisual, mejorando la calidad narrativa en eventos académicos, con el fin de evaluar el impacto del módulo en el Club SoftMedia.

1.7 Justificación

Hoy en día la tecnología avanza de manera descontrolada, transformando áreas importantes siendo la educación y la comunicación áreas totalmente reformadas por la tecnología. El Club SoftMedia enfrenta su principal problemática por el incremento significativo de eventos académico. Cuya escritura manual de los guiones genera un retraso significativo en la eficacia del Club SoftMedia, teniendo una limitación considerable en la producción de contenido que pude llegar a ser de gran repercusión a nivel universitario, al resolver el problema de escritura manual de los guiones se vería una gran optimización en los tiempos de entrega del material, teniendo una mayor repercusión a nivel institucional.

Al implementar un sistema que ayude a la gestión del contenido de manera automática facilita las tareas de los integrantes de SoftMedia lo que ayudaría a cumplir con los tiempos de entrega, al llevar un registro más organizado y automatizado se fortalece una mayor partición estudiantil, lo que promueve también un aprendizaje más profundas para los estudiantes en herramientas que implemente Procesamiento de Lenguaje Natural (PLN).

Al implementar esta solución tecnológica, desarrollada en la Uleam Ext El Carmen, se crea una necesidad de replicable para otras extensiones universitarias, aumentando el potencial colaborativo interdisciplinarios elevando la producción de material audiovisual en los entornos académicos con presupuestos reducidos para la producción audiovisual, que a su vez ayuda a

reducir la gran brecha digital que mantienen aisladas a varias comunidades locales dentro del ámbito sociocultural y académico.

Desde una perspectiva teórica, conviene complementar la justificación de este proyecto integrador revisando la literatura sobre eficiencia administrativa e institucional, que ha señalado de forma recurrente que la falta de herramientas automatizadas en los flujos de trabajo es uno de los principales obstáculos para el desempeño organizacional, sobre todo en entornos con recursos limitados. Los modelos conceptuales de liderazgo orientados a la eficiencia, por su parte, sostienen que la incorporación de tecnología en los procesos operativos no solo optimiza tiempos, sino que también libera capacidad humana para tareas de mayor valor como la planificación estratégica y la creatividad.

Partiendo de que la historia del pensamiento administrativo revela que la automatización ha sido desde siempre una herramienta para crecer sin multiplicar el trabajo manual, el caso de SoftMedia encaja perfectamente: su demanda de producción audiovisual ha aumentado más rápido que su capacidad de redacción disponible.

Cabe agregar que diversos estudios sobre gestión documental en instituciones de educación superior han evidenciado que la digitalización de procesos de generación y manejo de contenidos incide directamente en la eficiencia operativa de las áreas de comunicación institucional, reduciendo plazos de entrega y mejorando la trazabilidad de la información producida. En esa dirección, una revisión sistemática reciente concluye que automatizar tareas repetitivas de redacción y organización documental es una de las estrategias más efectivas para instituciones que, como el Club SoftMedia, carecen de personal técnico especializado y de presupuestos holgados para adquirir software comercial.

Partiendo de una necesidad concreta del club, el proyecto no se agota en esa solución operativa: se enmarca en un campo más extenso eficiencia administrativa, automatización y gestión documental digital y ofrece, además, evidencia aplicada que demuestra la viabilidad de estas herramientas en entornos universitarios con escasos recursos.

1.8 Impactos esperados

1.8.1 Impacto tecnológico

Mejorar la optimización en los procesos automatizados dentro del Club SoftMedia, facilitando llevar un mejor control en los flujos de trabajos internos del Club, promoviendo la implementación de modelos de IA en el ámbito académico, buscando reducir el uso de Software costoso, abriendo paso a futuras implementaciones de proyectos tecnológicos con

recursos limitados, integrando el uso de herramientas de código abierto (open-source) aumentando el nivel tecnológico dentro de la Uleam Ext El Carmen.

1.8.2 Impacto social

Aumenta la difusión de eventos académicos desarrollados dentro del ámbito socio-educativo de la Uleam Ext El Carmen, promoviendo una mejor relación con la comunidad, reafirmando un sentido de pertenencia atreves de una comunicación más inclusiva, impulsando una mayor participación voluntaria de los estudiantes en eventos académicos.

1.8.3 Impacto ecológico

Al implantar un sistema automatizado en la redacción de guiones audiovisuales se busca reducir el uso de papel en borradores, aportando una mayor sostenibilidad ecológica disminuyendo significativamente la tala de árboles, optimizando de igual manera el uso de energía con la optimización de tareas, siguiendo la línea ecológica de la Uleam, contribuyendo con el entorno ecológico dentro de la institución al reducir significativamente el uso de material fisico (hojas, papel) para la redacción de guiones, evitando la generación de residuos en el Club SoftMedia.

1.8.4 Impacto académico

la implementación del módulo de PLN genera un impacto formativo relevante en los estudiantes de Ingeniería en Tecnologías de la Información e Ingeniería en Software que participan en el Club SoftMedia, al ofrecerles la posibilidad de aplicar conocimientos de procesamiento de lenguaje natural, ingeniería de software y metodologías ágiles en un proyecto real con usuarios finales definidos. Esa experiencia, sin duda, trasciende lo que se aprende en el aula.

El proyecto promueve la colaboración interdisciplinaria entre estudiantes de carreras vinculadas a la producción audiovisual y a la tecnología, con lo que se fortalecen competencias de trabajo en equipo que resultan valiosas para su futura inserción profesional. Esta articulación entre inteligencia artificial y formación universitaria, reconocida como una línea prioritaria para la modernización de la educación superior en la región, permite que los estudiantes se familiaricen con herramientas de IA aplicada desde etapas tempranas de su formación.

CAPÍTULO II

2 MARCO TEÓRICO DE LA INVESTIGACIÓN

2.1 Antecedentes históricos

2.1.1 Sistema informático

En las últimas décadas, los sistemas informáticos evolucionaron desde soluciones centradas en cálculo y almacenamiento local hacia plataformas conectadas (web y nube) que integran automatización, analítica e inteligencia artificial para apoyar procesos organizacionales. Esta transición incrementó la capacidad de gestionar información, estandarizar productos comunicacionales y acelerar tareas repetitivas, siempre que existan reglas claras de calidad y control humano. (Myers et al., 2023)

En el ámbitos académicos, la automatización del flujos de trabajos se asocian a sistemas informáticos compuestos por herramientas de IA, ayudando en la gestión automatizada de registros de datos, trazabilidad y acortando tareas repetitivas dentro del ámbito de trabajo de SoftMedia, en la escritura de guiones audiovisuales el sistema ayuda en la recolección y almacenamiento de datos del evento convirtiéndolas en borradores para el material final, mantenido una postura narrativa institucional y estructurado. (CENDITEL, 2022)

En el caso del Club SoftMedia, un sistema informático orientado a guiones implica: capturar información mínima del evento (título, objetivo, lugar, participantes), generar una versión preliminar del texto narrable y permitir edición colaborativa. Esta visión es coherente con enfoques modernos de ingeniería de software que priorizan utilidad real, iteración y validación temprana con usuarios. (Canosa Ferreiro, 2024c; Celi-Párraga et al., 2023)

2.1.2 Procesamiento de Lenguaje Natural (PLN)

El Procesamiento de Lenguaje Natural (PLN) es el campo de la inteligencia artificial que permite a las máquinas procesar lenguaje humano para tareas como extracción de información, clasificación, resumen, traducción, corrección y generación de texto. Su crecimiento reciente se debe en gran parte al aprendizaje profundo y al uso de arquitecturas tipo transformer, que mejoran el manejo de contexto y coherencia en múltiples tareas. (Cortez Vásquez, 2020)

El PLN ha pasado de enfoques con reglas rígidas a modelos neuronal modernos con capacidad de aprendizaje con el uso de grandes volúmenes de datos, lo que facilita resultados

de escrituras de una manera más fluida con una estructura gramaticales con tono institucionales, pero estos modelos tienes sesgos en sus resultados que obligan a los miembros a diseñar planes de revisión técnica y validación del producto final, en el ámbito de SoftMedia se aplica PLN como un apoyo secundario para borradores, no como una fuente principal de escritura de guiones audiovisuales. (Cortez Vásquez, 2020; Lassi, 2022)

En SoftMedia se emplea también el PLN como herramienta para la optimización en el tiempo de redacción manual, generando un borrador final de los guiones usando datos breves y concisos de eventos que se lleven a cabo. La optimización de tiempo se centraliza en acelerar la escritura del primer borrador con una estructura estandarizada, teniendo una revisión por parte de los miembros de SoftMedia para tener una mayor fidelidad de coherencia narrativa para el producto final de los videos institucionales. (Gómez-Rodríguez, 2025)

2.1.3 Guiones audiovisuales

La función de un guion audiovisual es trabajar como un material de apoyo para la planificación operativa y narrativa del producto final, ordena los datos proporcionados definiendo ritmos y secuencias de mensajes narrativos atreves de la coordinación planificada en el trabajo de los miembros de SoftMedia definiendo los que graban, redactan y editan el material audiovisual, un guion evita improvisaciones, redundancia de materiales ayudando a tener una comunicación más eficaz entre los miembros del Club. (Mesonero Izquierdo, 2024)

En el mundo digital al que pertenecemos, las plataformas digitales han revolucionado las narrativas y formatos del contenido digital, adoptándolos a estructuras breves lo que obliga a adaptar el contenido digital con narrativas claras desde los primeros minutos, por lo que la escritura de guiones no solo debe de ser de manera esporádica, sino mantener una estructura con criterios estructurales claros que ayudan a mantener una narrativa coherente de principio a fin. (Túñez-López et al., 2021)

En SoftMedia, la dificultad se incrementa por la rotación de voluntarios y la simultaneidad de tareas (cobertura, edición, publicación). Por eso, un sistema que ayude a producir borradores consistentes puede mejorar eficiencia sin eliminar el rol humano: el guion debe seguir siendo editado para asegurar precisión institucional y estilo propio del club. (Canosa Ferreiro, 2024c)

2.2 Antecedentes de investigaciones relacionadas al tema presentado

2.2.1 Automatización de contenidos educativos con PLN.

En el ámbito educativo los métodos como el PLN se aplica en automatización de tareas repetitivas, lo que lleva a una mejor optimización en el flujo de trabajo mejorando el rendimiento en la entrega del producto final, atendiendo así toda la demanda de actividades académicas de una manera profesional, estas aplicaciones integradas con PLN ofrecen salidas (texto) de manera estructurada y coherente gracias a la integración de modelos preentrenados y técnicas de control de formatos. (Cortez Vásquez, 2020)

La redacción automática no busca eliminar la revisión humana en los textos generados, sobre todo en la difusión académica e institucionales se recomienda una revisión correctiva de coherencia, exactitud y adecuación de la narrativa a públicos específicos antes de ser publicados, el enfoque de trabajo suele ser "generación + edición" donde el módulo de PLN ayuda a reducir la escritura manual del primer borrador del guion, y los miembros del Club aplican criterios editoriales. (Creswell & Creswell, 2023)

Para SoftMedia,este margen estructural debe ser esencial, ya que un guion institucional debe poseer una estructura recurrente: presentación del evento, puntos clave, participante (autoridades, estudiantes) y cierre. Esta estructura predeterminada facilita el uso de plantillas y criterios de salida, para generar borrados que luego se podrán ajustar al tipo de evento a difundir mediante contenido audiovisual. (Cortez Vásquez, 2020)

2.2.2 Generación de textos para comunicación digital.

Los modelos modernos de PLN son utilizado significativamente para la elaboración de borradores en el ámbito digital, los borradores generados no son el producto final para la elaboración del contenido digital, más bien son una base sólida para formular ideas y construir estructuras narrativas coherentes según el tipo de evento, este toma fuerza cuando se combinan restricciones (límite de longitud, puntos clave del evento, publico especifico) ya que su propósito es entregar contenido útil para un propósito específico. (Gómez-Rodríguez, 2025)

Una de las mayores ventajas de la comunicación digital radica en la rapidez con que se pueden producir y revisar distintas versiones lo que brinda a los equipos la capacidad de evaluar varias opciones y elegir la más conveniente. Aun así, este beneficio puede transformarse en un riesgo si se generan textos que, aunque parecen coherentes, carecen de precisión. Para evitarlo, se sugiere incorporar mecanismos de validación y garantizar la trazabilidad, señalando los datos del evento que sirvieron de base para el guion. (Lassi, 2022)

En el ámbito de SoftMedia, la generación asistida por el módulo PLN puede desempeñar un papel de estandarización, garantizando que cada guion incluya los campos esenciales y conserve la coherencia con el tono institucional. De esta manera el procesamiento de lenguaje natural incrementa la productividad sin sustituir la valoración humana sobre la pertinencia y el estilo narrativo. (Canosa Ferreiro, 2024c; Gómez-Rodríguez, 2025)

2.2.3 PLN en la producción de contenidos institucionales.

En el ámbito institucional, la producción de contenido digital tiene como características principales: claridad, consistencia y precisión. Los modelos entrenados de PLN sirven como guía para reducir errores frecuentes y a mantener una estructura uniforme en las redacciones de guiones, para el equipo de SofteMedia esto es esencial cuando hay una alta demanda de actividades que deben ser difundidas por redes sociales. (Cortez Vásquez, 2020)

En instituciones socio-académicas deben regir políticas en el uso de PLN como: un filtro de revisión humana, controlar los datos sensibles de eventos, para evitar la difusión de información errónea sobre todo cuando se trabaja con modelos generativos. (Hallinan et al., 2021)

La eficacia en la redacción es el punto clave para SoftMedia, al proporcionar un borrador con los datos esenciales de los eventos abre la posibilidad de editar y realizar correcciones colaborativas antes de tener el producto final. (Celi-Párraga et al., 2023).

2.2.4 Herramientas de PLN para guiones educativos

Las bibliotecas y ecosistemas de PLN actuales simplifican tokenización, análisis y generación y además facilitan el accesoo a modelos preentrenados. Así, se obtienen prototipos operativos omitiendo el entrenamiento propio, lo cual se ajusta bien a los tiempos limitados en proyectos académicos con tiempo limitado. (Gómez-Rodríguez, 2025)

Para que las herramientas de PLN sean útiles en la práctica no basta con el modelo: se requiere un flujo integral que cuente con la entrada de datos, la generación por secciones, los filtros de longitud y una interfaz que permita editar, especialmente cuando el módulo de PLN se usa en eventos académicos con redacciones coherentes. (Celi-Párraga et al., 2023)

Para SoftMedia este prototipo busca simplicidad en: formulario corto, generación por bloque y la capacidad de ajustar los tonos de redacción, facilitando el trabajo de los miembros del club en el uso de coberturas rápidas. (Canosa Ferreiro, 2024b)

2.2.5 IA generativa en audiovisual universitario.

Si bien la IA generativa asiste en la ideación, los borradores narrativos y las reescrituras de guiones, su rendimiento en producción audiovisual depende de que la salida se guíe hacia una meta comunicacional clara, evitando así planteamientos genéricos o desconectados de la imagen institucional. (Myers et al., 2023)

Para que un video universitario represente con precisión eventos y actores institucionales, la calidad del resultado no solo exige buenos datos de entrada, sino también controles sistemáticos (plantillas, revisiones, listas de verificación) que mantengan la coherencia y fidelidad del mensaje. (Cortez Vásquez, 2020; Lassi, 2022)

Reducir tiempos de producción y mantener una frecuencia estable de publicaciones sin perder consistencia en las redes sociales es posible en SoftMedia gracias a la integración de IA generativa como "asistente de guion", que genera un texto base y permite iteraciones rápidas bajo supervisión humana. (Canosa Ferreiro, 2024c)

2.3 Definiciones conceptuales (contexto teórico)

2.3.1 Variable independiente: Procesamiento de Lenguaje Natural (PLN)

2.3.1.1 Fundamentos del PLN

En su enfoque actual, el PLN aprende representaciones del lenguaje desde grandes cantidades de datos mejorando la generalización, pero exige evaluar calidad y sesgos por parte de los integrantes de SoftMedia. Esta disciplina que combina lingüística y computación permite modelar el lenguaje como datos procesables para reconocer patrones, extraer información y generar guiones. (Cortez Vásquez, 2020)

Para que un sistema de redacción asistida genere borradores con coherencia razonable, resulta decisivo guiarlo con formato y restricciones, aprovechando así los modelos de atención, que han mejorado la comprensión y generación al capturar dependencias largas en el texto. (Gómez-Rodríguez, 2025)

Para este proyecto el fundamento aplicado es: ingresar datos del evento y obtener un guion preliminar estructurado. Para que esto sea útil institucionalmente, se requiere un proceso que combine generación con validación, evitando publicar texto no verificado o con tono inadecuado. (Cortez Vásquez, 2020; Lassi, 2022)

2.3.1.2 Técnicas básicas del PLN.

Las técnicas básicas incluyen preprocesamiento (normalización, limpieza), segmentación de texto, y procedimientos que permiten representar el lenguaje para que un modelo lo procese. Hoy en día hay el preprocesamiento sigue siendo clave para la estabilidad y consistencia de sistemas con formularios y guiones seccionados, pese a que los modelos actuales admiten variaciones sin mayor afectación. (Universidad de Minnesota, 2023)

En guiones institucionales, el control de forma (estructura) resulta tan decisivo como el contenido, pues el texto debe ser narrable y editable. Para ello, estas técnicas permiten acotar longitud, suprimir repeticiones, normalizar nombres y optimizar la legibilidad. (Gómez-Rodríguez, 2025)

2.3.1.2.1 Tokenización

La función principal de la Tokenización es dividir el texto en palabras para que el modelo tenga una mayor taza de interpretación del texto entregado, en los sistemas modernos la tokenización ayuda a un mejor manejo de vocabulario amplios. (Cortez Vásquez, 2020)

Pese a que pueda parecer un detalle técnico, la tokenización es una pieza fundamental en la redacción automatizada de ya que ayuda a mantener la coherencia narrativa y, además, ayuda a evaluar que el texto cumpla con la longitud esperada para narraciones breves. (Cortez Vásquez, 2020; Gómez-Rodríguez, 2025)

2.3.1.2.2 Lematización

Reduciendo las palabras a su raíz, la lematización achica las variantes morfológicas y da más solidez a la terminología. Puede que no siempre sea requerida por los modelos modernos de PLN, pero en contextos institucionales, para estandarizar vocabulario o igualar términos afines, su utilidad es muy frecuente. (Cortez Vásquez, 2020)

La lematización ayuda a evitar repeticiones innecesarias y mantener uniformidad en palabras/términos claves en la escritura de guiones, facilitando la edición final mantenido una mejor consistencia en el texto final. (REUE, 2024)

2.3.1.2.3 Análisis morfosintáctico

Detectar problemas de concordancia oraciones mal construidas o segmentos confusos es posible mediante el análisis morfosintáctico, que examina la estructura gramatical y las relaciones entre palabras. Con ello se eleva la calidad del borrador antes de la revisión humana final. (Torres-Hostos, 2020)

Integrar en SoftMedia una verificación morfosintáctica automática (por ejemplo, para localizar frases largas o repeticiones) permite aligerar el trabajo de corrección manual y agilizar el flujo productivo cuando los tiempos de entrega son sumamente cortos. (Celi-Párraga et al., 2023)

2.3.1.3 Modelos de lenguaje

La evolución reciente de los modelos de lenguaje apunta a arquitecturas que capturan contexto y generan textos más coherentes, aunque sin controles no garantizan exactitud factual. Su funcionamiento básico consiste en estimar probabilidades de secuencias o producir texto a partir de patrones aprendidos. (Gómez-Rodríguez, 2025)

Estas herramientas deben trabajar bajo restricciones de generación como: estructuras predeterminadas, datos obligatorios y limites en su extensión narrativa. Sin estas condiciones, el texto puede desviarse del evento o introducir afirmaciones no verificadas. Por ello, el diseño del sistema (reglas y validación) es parte esencial de la solución. (Lassi, 2022)

2.3.1.3.1 N-gramas y modelos estadísticos

Los n-gramas modelan secuencias cortas y son útiles como baselines o para plantillas repetitivas, pero su capacidad de mantener coherencia global es limitada. Aun así, pueden aportar en tareas específicas como sugerir frases recurrentes o cierres institucionales estandarizados. (Cortez Vásquez, 2020)

En un sistema de guiones, los modelos estadísticos pueden complementar enfoques modernos como "filtros" para detectar repeticiones o incoherencias locales. Sin embargo, para producir un borrador narrativo completo suele ser preferible un enfoque neuronal controlado por plantillas. (Torres-Hostos, 2020)

2.3.1.3.2 Redes neuronales (BERT, GPT)

Modelos como BERT se asocian a comprensión y representación contextual, mientras que modelos tipo GPT se orientan a generación. Para redactar guiones por secciones sin desviarse del mensaje resulta útil que modelos basados en transformers capturen relaciones contextuales amplias, mejorando así la coherencia del texto. (Gómez-Rodríguez, 2025)

Para SoftMedia, la generación guiada (plantilla que obligue a incorporar datos del evento y limite longitud) resulta el camino adecuado, complementado con evaluación humana y controles de calidad ante la posibilidad de que los modelos arrojen texto fluido pero inexacto. (Megahed et al., 2024)

2.3.1.3.3 Arquitectura Transformer y modelos generativos

La arquitectura Transformer constituye uno de los avances más importantes dentro del campo del Procesamiento del Lenguaje Natural, debido a su capacidad para procesar grandes cantidades de texto y comprender relaciones contextuales de largo alcance mediante mecanismos de atención (attention mechanism). A diferencia de enfoques tradicionales basados en reglas o modelos estadísticos, los Transformers permiten analizar simultáneamente todos los elementos de una secuencia textual, mejorando significativamente la coherencia y calidad de los resultados generados. (Cortez Vásquez, 2020)

La generación automática de texto, el resumen, la redacción asistida y el contenido especializado son áreas donde los modelos Transformer como GPT demuestran alto rendimiento, gracias a un preentrenamiento masivo y a la posibilidad de ajuste fino por transfer learning hacia dominios concretos.

Los modelos generativos al transformar información estructurada proporcionada por el usuario en borradores organizados, facilitan la elaboración de guiones y reducen notablemente los tiempos de planificación en contextos de producción audiovisual.

2.3.1.3.4 Fine-Tuning y Aprendizaje por Transferencia

Partiendo de un modelo que ya ha visto grandes volúmenes de información, el transfer learning toma ese aprendizaje previo y lo reorienta hacia tareas particulares. Pero con conjuntos de datos mucho más reducidos. Así se aprovecha lo que el modelo ya sabe, sin necesidad de reentrenarlo todo desde el principio.

Partiendo de la escasez de datos masivos en entornos académicos o institucionales, el Transfer Learning resulta una alternativa práctica mediante el Fine-Tuning, que especializa modelos ya entrenados para dominios específicos y con ello eleva la calidad de lo generado, adaptándolo a lo que realmente se necesita. (Gómez-Rodríguez, 2025)

En el contexto de guiones audiovisuales, el Fine-Tuning hace algo bastante práctico: ajusta el modelo a formatos redaccionales concretos a las estructuras que maneja la institución y a estilos comunicativos particulares, lo que se traduce en resultados más certeros y aprovechables.

2.3.1.4 Herramientas de desarrollo

El desarrollo de un sistema con PLN no es solo cuestión del modelo: requiere herramientas para integrarlo, controlar lo que entra, generar por partes y evaluar lo que sale.

Ademas, la arquitectura tiene que incluir mantenibilidad, pruebas y trazabilidad, porque el sistema se usará en producción final y no deben existir error. (Kaur et al., 2023)

Aprovechando modelos ya entrenados y armando un pipeline con pasos claros: formulario, generación, revisión y exportación del guion, se obtiene una estrategia efectiva en proyectos académicos que cuentan con un límite de tiempo ajustado. Y con eso, el esfuerzo inicial se reduce, y se puede insistir sobre mejoras a medida que llega la retroalimentación. (Gómez-Rodríguez, 2025)

2.3.1.4.1 Librerías Python (spaCy, NLTK).

spaCy es una librería enfocada en el procesamiento de texto de forma eficiente, adecuada para construir pipelines de PLN (tokenización, análisis lingüístico y extracción de información) con buen desempeño en aplicaciones reales. Su uso resulta pertinente para desarrollar módulos de escritura asistida o análisis de texto de entrada, como en el caso de generación de borradores de guiones a partir de datos del evento. (Cortez Vásquez, 2020)

Por su parte, NLTK es una librería orientada a trabajar con lenguaje humano en Python, ofreciendo una colección amplia de recursos y algoritmos útiles en tareas base de PLN (análisis, procesamiento y prototipado). Al ser herramientas de código abierto, facilitan el desarrollo en contextos con recursos limitados; además, pueden integrarse y optimizarse dentro de flujos de trabajo de machine learning mediante prácticas de diseño y mantenimiento de sistemas (por ejemplo, validación, pruebas y despliegue incremental). (Celi-Párraga et al., 2023)

2.3.1.4.2 Plataformas (HuggingFace)

Hugging Face es un ecosistema/plataforma que ofrece modelos modernos preentrenados, datasets y espacios para implementar aplicaciones, destacando por facilitar el acceso a arquitecturas transformer y componentes reutilizables (tokenizadores, pipelines y utilidades de inferencia). Esto permite desarrollar prototipos sin necesidad de entrenar modelos desde cero, lo cual es especialmente útil para proyectos académicos con tiempos acotados y equipos pequeños. (COEV, 2024)

En un contexto como SoftMedia, esta accesibilidad favorece la democratización del desarrollo (probar, ajustar y evaluar modelos) siempre que se mantenga control sobre el formato de salida y revisión humana del contenido. Además, al trabajar con modelos generativos, es necesario considerar aspectos de uso responsable (calidad, sesgos, transparencia y validación) antes de aplicar el texto en comunicación institucional. (Lassi, 2022)

2.3.1.5 Casos de uso en comunicación institucional

El PLN se ha integrado rápidamente en instituciones para estandarizar textos, acelerar borradores y mantener la consistencia. Lo que resulta muy práctico cuando hay que generar muchos mensajes con estructura parecida, como convocatorias o reseñas breves de actividades. (Torres-Hostos, 2020)

Al tratarse de comunicación pública, se necesita gobernanza: revisión humana, control de sesgos y técnicas para documentar cómo se generó el texto. Al implementar técnicas somo estas, reducen el riesgo de errores y dan más confianza al proceso. (Lassi, 2022)

2.3.1.5.1 Automatización de contenido textual

Gracias a la automatización se puede generar borradores rápidos y con una estructura fija. En la práctica, se implementan modelos, plantillas y validación, para que el resultado tenga coherencia y sirva en un flujo de trabajo real. (Celi-Párraga et al., 2023)

Dentro del Club SoftMedia la generación automatizada de borradores de giones reduce el tiempo de la realización de materiales audiovisuales, gracias a las plantillas pre definidas a utilizar agilizando el tiempo de revisión y la narrativa espontanea. (Canosa Ferreiro, 2024c)

2.3.1.5.2 Adaptación de lenguaje al público objetivo

Ajustando tono, extensión y vocabulario al público al cual está dirigido el contenido audiovisual, la adaptación de lenguaje cumple su función con la ayuda de modelos entrenados guiando la salida con ejemplos, instrucciones y limitaciones de formato, con lo que se gana en consistencia del estilo institucional. (Gómez-Rodríguez, 2025)

Aunque el público universitario de SoftMedia pide claridad, dinamismo y que se reduzca la utilización de términos técnicos que estorben, la adaptación del lenguaje no termina ahí. Requiere además una revisión editorial que asegure la identidad institucional y prevenga sesgos o tonos que desentonen. (Pérez González, 2021)

2.3.1.6 Aplicaciones en educación

Es importante señalar que, en el ámbito educativo, el PLN se utiliza para apoyar la redacción y organización de contenidos, con especial atención a la productividad y la accesibilidad de materiales. Cuando se tiene una estructura clare base, con criterios que permitan evaluar la calidad de lo generado, es cuando estas herramientas rinden mucho mejor (Rodríguez Fernández, 2021)

Estructurando información, definiendo plantillas y evaluando coherencia narrativa, el PLN fortalece el aprendizaje práctico en SoftMedia. Y en un contexto de rotación de voluntarios, el sistema estabiliza la producción al dar un punto de partida que no cambia. (Creswell & Creswell, 2023)

2.3.1.6.1 PLN en generación de contenidos educativos

La generación de contenido debe evaluarse por coherencia, cobertura de puntos clave, corrección lingüística y adecuación al propósito. En la generación de borradores de guiones de debe tener en cuenta si el borrador es narrable y se ajusta al tiempo del producto final. (Gómez-Rodríguez, 2025)

Aunque el equipo de SoftMedia se queda con la revisión final de los borradores, el verdadero aporte del PLN está en abaratar el primer borrador. Y así se consigue un equilibrio práctico: eficiencia, por un lado, control editorial por el otro. (Cortez Vásquez, 2020)

2.3.1.6.2 Casos en universidades

Reducir el riesgo y aprender con feedback real es el punto principal de avanzar mediante prototipos incrementales en la adopción del uso del PLN en las entidades universitarias. El trayecto: validar utilidad básica, ajustar calidad y, finalmente integrar al flujo operativo. (Celi-Párraga et al., 2023)

En SoftMedia, la implementación práctica arranca con un generador de guion bastante sencillo, y a partir de ahí se van dando pasos sucesivos: se refinan las plantillas, se acota la extensión, se añaden verificaciones lingüísticas y se adapta el estilo a los eventos que el club realmente cubre. (Canosa Ferreiro, 2024c; Celi-Párraga et al., 2023)

2.3.1.7 Desafíos y limitaciones del PLN

Por más que los modelos generativos produzcan textos fluidos y convincentes, los desafíos principales están en el sesgo de los datos, la escasa supervisión del contenido y la dependencia de la entrada del usuario. Y es que esa fluidez puede ocultar información ficticia, por lo que la validación y la revisión antes de la publicación se vuelven requisitos obligados. (Megahed et al., 2024)

Además, la adopción técnica exige capacitación mínima y un diseño centrado en el usuario. Un sistema útil debe ser fácil de operar para voluntarios y debe orientar a ingresar datos correctos para mejorar calidad del guion. (Celi-Párraga et al., 2023)

2.3.1.7.1 Problemas con datos sesgados

El tono de la narrativa pude ser gran portador de sesgos de datos, En contenidos informativos institucionales es un gran problema estos tipos de sesgos porque pude llegar a afectar la imagen pública de la institución. (Kaur et al., 2023)

Para tratar de solucionar estos problemas, se debe trabajar con documentación de datos, verificar salidas de texto, y aplicar un filtro de evaluación humano por parte de los miembros del Club, lo que reducirá el riego de publicaciones con información errónea. (Cortez Vásquez, 2020)

2.3.1.7.2 Limitaciones en la capacitación técnica

Partiendo de que la falta de capacitación puede traer mal uso en la práctica, entradas de datos incompletas, expectativas elevadas y la falta de revisión final, para una buena práctica de equipos voluntarios es clara: acompañar la herramienta con guías simples y listas de verificación para revisar el guion previo a la grabación. (Celi-Párraga et al., 2023)

Menos fricción y más valor práctico se consigue cuando un enfoque ágil guía la adopción del sistema, con entrenamiento basado en ejemplos reales, mejoras iterativas y ajustes del flujo guiados por retroalimentación. (Alaimo, 2021)

2.3.1.8 Tendencias futuras en PLN

Aunque las tendencias futuras de la evolución del PLN prometen asistentes de producción más completos, gracias a la integración multimodal, mejores controles de seguridad y herramientas de evaluación de sesgos y transparencia, también traen consigo una exigencia paralela: más gobernanza y validación. (Lassi, 2022)

La calidad y coherencia de los borradores de guiones seguirá dependiendo de la revisión humana y del control de datos de entrada, incluso si SoftMedia avanza hacia paquetes de producción que incluyan guion, versión corta para redes, lista de puntos clave y sugerencias de rótulos. (Mesonero Izquierdo, 2024)

2.3.1.8.1 Integración con multimodalidad (texto + video)

La multimodalidad apunta a vincular el lenguaje con lo visual y lo auditivo, y cobra sentido en guiones porque la narración tiene que coordinarse con escenas y elementos gráficos, aunque el proyecto actual trabaja solo con texto, diseñar el guion por bloques ya permite asociar cada parte a tomas y rótulos específicos. (Gómez-Rodríguez, 2025)

A futuro, el sistema podría sugerir descripciones de escena o puntos de apoyo visual, pero esto debe controlarse para mantener coherencia con el evento real. Por ello, se requiere un enfoque de plantillas y validación para no inventar contenido. (Cortez Vásquez, 2020)

2.3.1.8.2 Ética y sostenibilidad en PLN

La ética en PLN implica transparencia, control de sesgos, respeto por datos personales y revisión humana, especialmente en comunicación institucional. Además de una documentación de la generación del texto y quien verifico su fiabilidad. (Lassi, 2022)

Producir más contenido sin aumentar la carga humana es asociar sostenibilidad con eficiencia operativa. En SoftMedia, eso se traduce en menos tiempo de redacción manual y repetitiva, menos reprocesos manteniendo la calidad para que corregir borradores defectuosos no se convierta en una fuente adicional de trabajo. (Celi-Párraga et al., 2023)

2.3.2 Variable dependiente: Guiones audiovisuales

2.3.2.1 Estructura de un guion institucional

Priorizar la información del evento sosteniendo un hilo lógico y cerrar con una idea facilita la estructura clásica de: introducción, desarrollo y cierre. Con eso, el mensaje se vuelve comprensible y narrable, justo lo que piden los videos institucionales breves sin ser tan densos para los usuarios. (Túñez-López et al., 2021)

Estas estructuras clásicas en entornos universitarios son claves para una operación estandarizad, facilitando la colaboración mutua entre los integrantes sin perder la coherencia narrativa. Esto es útil cuando se trabaja con voluntarios y plazos ajustados. (Canosa Ferreiro, 2024c)

2.3.2.1.1 Introducción

La introducción capta atención y contextualiza el evento rápidamente. En video para redes, se recomienda que sea directa, breve y alineada al objetivo comunicacional (informar, invitar, destacar un logro). (Mesonero Izquierdo, 2024)

Un sistema con PLN puede generar introducciones consistentes si recibe campos mínimos (evento, lugar, objetivo), teniendo en cuenta un filtro de revisión por los integrantes de SoftMedia verificando el tono de narración y verificación de datos delicados, como nombres antes de publicar el producto final. (Cortez Vásquez, 2020)

2.3.2.1.2 Desarrollo

Exponiendo los puntos clave: qué ocurrió, quiénes participaron y su relevancia, el desarrollo cumple su función. Pero a su vez tiene que sostener la coherencia, evitar reiteraciones y dosificar el ritmo narrativo para que encaje con la duración del video. (Túñez-López et al., 2021)

Partiendo de una lista de ideas y transformándolas en párrafos conectados por transiciones, el desarrollo con PLN toma forma. Pero para evitar la dispersión, lo correcto es acotar la longitud y asegurarse de que cada frase repose sobre datos concretos del evento. (Lassi, 2022)

2.3.2.1.3 Cierre

El cierre además de incluir agradecimientos también tiene la función de reafirmar la idea central del evento, teniendo una longitud corta sin modificar o incluir datos que confundan al consumidor del material audiovisual (usuarios/estudiante). (Túñez-López et al., 2021)

Estandarizando los cierres mediante plantillas institucionales se logra con la implementación del sistema recortando tiempos de redacción. Eso sí, el equipo debe verificar después que el cierre sea pertinente para el evento, no siempre se necesita cerrar con invitación o convocatoria, y eso hay que tenerlo en cuenta. (Celi-Párraga et al., 2023)

2.3.2.2 Componentes narrativos, técnicos y visuales

Aunque cada tipo de componente narrativo, técnico y visual tenga su propia lógica, el guion los integra todos: mensaje, tono, conectores, secuencia, tiempos, transiciones, rótulos, apoyos y recursos gráficos. Y esa integración, justamente, es la que reduce la improvisación y hace más fluida la coordinación entre el trabajo colaborativo de los miembros de SoftMedia. (Mesonero Izquierdo, 2024)

Que la salida del sistema sea modular (en bloques) y fácil de editar es clave para integrarse con grabación y edición. Aunque un sistema PLN solo produzca texto, un guion bien segmentado facilita convertirlo en escenas y decisiones técnicas. (Celi-Párraga et al., 2023)

2.3.2.2.1 Componentes narrativos

Lo que hace que un guion mantenga la consistencia y coherencia narratova son sus componentes narrativos que lo conforman: idea central, orden de hechos y tono, en instituciones académicas en la producción audiovisual se prioriza claridad del mensaje, y precisión del mismo para evitar ambigüedad. (Mesonero Izquierdo, 2024)

Vale la pena señalar que el PLN puede ayudar a mejorar conectores y fluidez, pero siempre que esté guiado por estructura y datos. Y la revisión humana siendo necesaria para asegurar que el texto refleje fielmente lo ocurrido en el evento. (Cortez Vásquez, 2020)

2.3.2.2.2 Componentes técnicos

Por más que parezcan detalles menores, planificar tomas y organizar escenas reduce repeticiones al grabar. Y un guion bien definido evita grabar sin rumbo, con la consecuencia de perder tiempo en edición porque falta material aprovechable. (Canosa Ferreiro, 2024c)

Optimizar la coordinación del equipo, sobre todo cuando los integrantes cumplen múltiples roles, es lo que permite un borrador bien estructurado al asignar responsabilidades claras: quién narra qué, qué se debe mostrar y qué rótulos se requieren. (Celi-Párraga et al., 2023)

2.3.2.2.3 Componentes visuales

Los componentes visuales ayudan a mejorar la claridad del mensaje y capta la atención de los usuarios que ven el material audiovisual, apoyándose con imágenes, planos de apoyo. Con un guion bien estructurado se facilita decidir que imágenes o planos mostrar evitando una redundancia de material visual. (Mesonero Izquierdo, 2024)

Evitar sobrecargar el video con texto y validar nombres y cargos para prevenir errores institucionales son tareas que exigen control editorial que deben encargarse los miembros de SoftMedia, aunque el PLN pueda apoyar redactando rótulos breves o frases destacables. (Lassi, 2022)

2.3.2.3 Proceso de creación, redacción y revisión colaborativa

Investigación previa de información preliminar para una correcta redacción de los borradores acompañada de una revisión colectiva forman parte de una estructura típica de un guion, teniendo en cuanta la coherencia narrativa adaptada al ámbito institucional. (Morales Castillo, 2019)

Aunque el PLN acelere el borrador inicial la revisión no se delega: la colaboración sigue ahí para afinar el tono, corregir datos y darle ritmo a la narración. Y en equipos con voluntarios como lo son los integrantes de SoftMedia, tener un borrador base para trabajar reduce la fricción y mejora la eficiencia del trabajo colaborativo. (Cortez Vásquez, 2020)

2.3.2.3.1 Investigación previa

La recolección de datos esenciales como: propósito, lugar, fecha y participantes previo a los eventos forman parte de una correcta investigación previa, sin esta práctica la generación de guiones pierde precisión para una coherencia narrativa. (Lassi, 2022)

Definir campos obligatorios y validaciones para evitar entradas ambiguas es una recomendación clave, porque en un sistema con PLN esta fase se traduce directamente en campos de formulario: entre más claros y completos, mejor será el borrador. (Celi-Párraga et al., 2023)

2.3.2.3.2 Redacción

La redacción recopila los datos previamente investigados para darle una estructura narrativa coherente, para videos institucionales breve es factible usar frases o párrafos de ideas claras para una locución y producción más fluida. (Túñez-López et al., 2021)

Adaptar estos borradores al estilo narrativo de SoftMedia y revisar precisión antes de grabar es mucho más factible implementando PLN ya que automatiza la redacción inicial, pero siempre entregando un texto editable y segmentado. (Gómez-Rodríguez, 2025)

2.3.2.3.3 Revisión colaborativa

Una buena revisión colaborativa interna de los miembros de SoftMedia, ayuda a erradicar errores y redundancia de la narrativa, ayudando así a mejorar los cortes finales del mensaje para mantener un tiempo de duración controlado del material audiovisual. (Mesonero Izquierdo, 2024)

Obteniendo el borrador por secciones, el PLN acelera esta fase y permite que la revisión se distribuya en bloques. Pero no basta con eso: hace falta una lista de verificación básica con nombres correctos, coherencia, tono, longitud y un llamado a la acción que cierre bien. (Cortez Vásquez, 2020)

2.3.2.4 Tipos de guiones: Literario, técnico, redes sociales y plataformas digitales

A pesar que cada tipo de guion responde a un propósito distinto: literario para narrar, técnico para producir, de redes para ser breve y dinámico, en el ámbito universitario suele preferirse una fórmula mixta: que sea narrable, pero que también funcione en la práctica para grabar y editar. (Mesonero Izquierdo, 2024; Túñez-López et al., 2021)

Para la producción de SoftMedia la estructura de guiones predeterminad es para redes sociales instituciónes eles breves, por ello el modelo PLN debe generar guiones bases para ese

tipo de contenido de difusión rápida, además manteniendo la coherencia narrativa institucional. (Gómez-Rodríguez, 2025)

2.3.2.4.1 Guion literario

Los guiones literarios son caracterizados por tener una mayor prioridad en: mensaje, orden narrativo y tono, mantenido una continuidad adecuada y una comunicación clara. (Mesonero Izquierdo, 2024)

Este tipo de guion es asociado con PLN ya que el sistema produce texto narrativo por secciones. Luego, el equipo ajusta estilo, precisión y conectores según el evento. (Cortez Vásquez, 2020)

2.3.2.4.2 Guion técnico

El guion técnico constituye una herramienta fundamental dentro de la producción audiovisual, ya que especifica de manera detallada cómo deberá ejecutarse visualmente cada escena. A diferencia del guion literario, incorpora elementos relacionados con la planificación de producción, incluyendo tipos de plano, movimientos de cámara, duración estimada de tomas, secuencia de escenas y recursos visuales necesarios para la grabación. (Mesonero Izquierdo, 2024)

La utilización de inteligencia artificial en la generación de guiones técnicos permite reducir considerablemente los tiempos de planificación audiovisual, proporcionando recomendaciones estructuradas que sirven como apoyo para los equipos de producción. Este enfoque resulta especialmente útil en entornos universitarios donde los recursos humanos y el tiempo disponible suelen ser limitados.

2.3.2.4.3 Guiones para redes sociales

Los guiones para redes requieren impacto rápido, idea central clara y cierre directo. Estos se complementan con mínima información de los eventos institucionales, evitando saturación de información. (Túñez-López et al., 2021)

Los sistemas PLN pueden generar este tipo de versiones cortas controlando longitud y estructuras para los borradores, permitiendo una correcta daptación del material para las diferentes redes sociales sin la necesidad de una nueva reestructuración para cada red social. (Gómez-Rodríguez, 2025)

2.3.2.5 Importancia del guion: Coherencia narrativa y optimización de recursos

Sosteniendo el hilo del mensaje y reduciendo la improvisación, el guion ayuda a mejorar la calidad final del producto. Y de paso, recorta las correcciones tardías las cuales son las que mas tiempo de producción consumen. (Mesonero Izquierdo, 2024)

Coordinando roles y mantener una identidad comunicacional estable es posible con un guion con bases consistentes en el trabajo de SoftMedia, donde la importancia se intensifica por la carga operativa: coberturas frecuentes, equipo voluntario y tiempos ajustados. (Canosa Ferreiro, 2024c)

2.3.2.5.1 Coherencia narrativa

Una continuidad secuencial lógica compuesta por: inicio, desarrollo y cierre, ayuda a llevar una coherencia narrativa limpia sin contradicciones. Esto hace que el video sea entendible y narrable con fluidez. (Mesonero Izquierdo, 2024)

En PLN, la coherencia mejora cuando hay estructura fija y conectores guiados. La revisión humana sigue siendo necesaria para ajustar transiciones y eliminar redundancias, especialmente en eventos con información limitada o dispersa. (Cortez Vásquez, 2020)

2.3.2.5.2 Optimización de recursos

La optimización de recursos se refiere a usar mejor el tiempo del equipo, reducir reprocesos y mejorar coordinación entre redacción, grabación y edición, todo esto se logra cuando se trabaja con un guion con bases solidad, evitando tener una redundancia de material visual para la producción final. (Mesonero Izquierdo, 2024)

Distribuir tareas entre redacción, grabación y edición, es lo que permite el guion en producción audiovisual universitaria, donde los recursos críticos suelen ser el tiempo disponible, la coordinación entre roles y la consistencia del mensaje institucional. (Canosa Ferreiro, 2024c)

Arrancando con un borrador inmediato que el PLN genera desde el inicio, el equipo no pierde demasiado tiempo en la primera versión. Y usando plantillas, el borrador sale con formato consistente y por secciones, lo que agiliza la revisión colaborativa y, de paso, achica las diferencias de criterio entre integrantes sobre el contenido del video. (Gómez-Rodríguez, 2025)

Aunque la optimización suene atractiva para que sea real y no derive en "más trabajo corrigiendo", hacen falta controles/filtros: campos obligatorios, validación de datos y una

aprobación mínima antes de grabar o publicar. Porque eso reduce errores: nombres, fechas, lugares, que de salir mal, cuestan reputación y obligan a retrabajar. (Lassi, 2022)

2.3.2.6 Retos en la redacción de guiones en entornos académicos

Aunque los retos comunes pueden ser: falta de tiempo, rotación de integrantes, estilos diversos y falta de estandarización, aunque parezcan manejables por separado, en equipos voluntarios se combinan y generan inconsistencias, además de una mayor carga de corrección. (Mesonero Izquierdo, 2024)

El PLN puede mitigar parte del problema al estandarizar el borrador inicial, pero luego el control editorial sigue siendo un reto que no desaparece. Y por eso, el diseño del flujo qué genera el sistema, qué revisa el equipo y con qué criterios termina siendo determinante. (Mesonero Izquierdo, 2024)

2.3.2.6.1 Falta de personal capacitado

Con la rotación constante de los participantes del Club, la curva de aprendizaje se eleva de manera descontrolada, especialmente sin el uso de plantillas cada nuevo integrante del club escribe ideas diferentes a los demás, lo que que genera una perdida en la consistencia de la información perdiendo con ella la identidad institucional. (Canosa Ferreiro, 2024c)

Si bien el sistema PLN no lo resuelve todo, al ofrecer estructura fija de guion, guía de campos y salida consistente, puede operar como estándar. Y esa base, precisamente, acelera el aprendizaje del formato para los nuevos integrantes y da más estabilidad a la producción. (Cortez Vásquez, 2020)

2.3.2.6.2 Limitaciones de tiempo para producción

Por más que los plazos ajustados sean una realidad, su efecto es empujar a improvisar o a recortar revisiones, y eso afecta a la coherencia narrativa, estilo y precisión, lo que termina impactando la calidad del video y la percepción institucional. (Túñez-López et al., 2021)

Por ende, la optimización de tiempo en la redacción del primer borrador con el módulo PLN permite invertir ese tiempo ganado en la verificación y menoración del guion. Sin embargo, la rapidez debe acompañarse de revisión mínima obligatoria para evitar publicar borradores no validados. (Gómez-Rodríguez, 2025)

2.3.2.7 Gestión de historial y versionado de guiones

La gestión de historial y versionado constituye un mecanismo de trazabilidad que permite registrar todas las modificaciones realizadas sobre un documento durante su ciclo de

vida. En sistemas colaborativos, esta funcionalidad facilita el control de cambios, la recuperación de versiones anteriores y la identificación de los usuarios responsables de cada acción realizada. (Hallinan et al., 2021)

En el contexto de la generación automática de guiones audiovisuales, el versionado permite almacenar múltiples versiones de un mismo guion, registrar fechas de creación, modificación y eliminación lógica, así como mantener un historial completo de las operaciones realizadas por los usuarios autenticados del sistema. (Celi-Párraga et al., 2023)

La implementación de mecanismos de auditoría y trazabilidad contribuye a incrementar la confiabilidad, seguridad y control administrativo de la información generada por la plataforma.

2.3.3 Metodología de desarrollo Scrum

Ajustar requisitos según el uso y las necesidades observadas se logra con la metodologia Scrum porque su lógica de iteraciones cortas, entregas funcionales y retroalimentación continua encaja perfectamente cuando el producto debe validarse con usuarios reales. (Canosa Ferreiro, 2024a)

Si bien el proyecto puede parecer complejo, Scrum prioriza lo fundamental: un módulo mínimo que capture datos del evento y genere un borrador usable y después se agregan mejoras. También se puede evaluar el sistema por componentes: calidad del guion, facilidad de edición, aceptación del equipo, e irlo ajustando con lo que va dejando la experiencia real del Club SoftMedia.. (Canosa Ferreiro, 2024c; Celi-Párraga et al., 2023)

2.3.3.1 Roles en Scrum

Si bien los roles de Scrum pueden ajustarse en entornos académicos, la estructura base sigue siendo la misma: Product Owner (prioriza valor), Scrum Master (facilita y elimina impedimentos) y Developers (construyen el incremento). Y esa claridad, justamente, es la que previene el desorden y los retrasos. (Alaimo, 2021)

Representando las necesidades de SoftMedia en cuanto a estructura y calidad del guion (el Product Owner), velando por el proceso (el Scrum Master) e implementando y probando el prototipo (el equipo de desarrollo), la asignación de roles queda clara. Y así, el foco y la mejora continua acompañan cada Sprint. (Canosa Ferreiro, 2024b)

2.3.3.2 Artefactos y eventos en Scrum

Visibilidad del trabajo y priorización por valor: eso es lo que permiten los artefactos principales de Scrum: el Product Backlog, el Sprint Backlog y el Incremento. Esto ayuda a gestionar alcance y evitar que el proyecto se "dispare" en funcionalidades no esenciales. (Canosa Ferreiro, 2024a)

Los eventos (Planificación, Daily, Review y Retrospective) aseguran seguimiento, validación con usuarios y mejora del proceso. En este proyecto, la Sprint Review es clave para revisar si el guion generado es coherente, editable y útil para producción real. (Canosa Ferreiro, 2024b)

2.4 Conclusión del marco teórico

El recorrido teórico desarrollado en este capítulo permite establecer, en primer lugar, que el Procesamiento de Lenguaje Natural constituye una tecnología suficientemente madura para apoyar tareas de generación textual estructurada, pero su aplicación en un contexto institucional como el del Club SoftMedia exige límites claros: el modelo debe funcionar como un asistente de redacción y no como un sustituto de la revisión humana, ya que los antecedentes revisados coinciden en señalar la persistencia de sesgos, imprecisiones y errores de coherencia en los textos generados automáticamente. Esta conclusión justifica directamente el diseño de la propuesta desarrollada en el Capítulo IV, donde el módulo de PLN se plantea como un componente que produce borradores editables, y no contenido final listo para publicación.

En segundo lugar, el análisis de los antecedentes sobre guiones y narrativa audiovisual confirma que la calidad de un guion institucional depende menos de la originalidad estilística y más de la existencia de una estructura predecible: introducción, desarrollo y cierre que facilite tanto la generación automatizada como la edición posterior por parte de los miembros del club. Esta estructura recurrente, identificada de forma consistente en la literatura consultada, es la que finalmente se adopta como base del formulario de entrada y del formato de salida del módulo propuesto.

Finalmente, el marco teórico correspondiente a la metodología de desarrollo de software confirma la pertinencia de Scrum para un proyecto de estas características: un equipo reducido, con validaciones frecuentes de usuarios reales (los voluntarios de SoftMedia) y la necesidad de ajustar el alcance conforme avanza la comprensión del problema.

CAPÍTULO III

3 MARCO INVESTIGATIVO (DISEÑO METODOLÓGICO)

3.1 Introducción

El presente capítulo describe el diseño metodológico utilizado para diagnosticar la problemática del Club SoftMedia de la ULEAM Extensión El Carmen y, a partir de ello, sustentar la propuesta de un sistema informático con Procesamiento de Lenguaje Natural (PLN) orientado a automatizar la redacción de guiones audiovisuales. En consecuencia, se detallan el tipo y enfoque de investigación, los métodos empleados, las fuentes de información, la población y muestra, así como las técnicas e instrumentos de recolección y análisis de datos, con el fin de garantizar coherencia entre los objetivos del estudio, los procedimientos aplicados y los resultados obtenidos. (Creswell & Creswell, 2023; Flick, 2022).

3.2 Tipos de Investigación

3.2.1 Investigación Aplicada

La investigación aplicada se orienta a resolver necesidades concretas de un contexto real mediante el desarrollo de una solución utilizable. Cabe subrayar que el resultado esperado de este proyecto es un sistema informático basado en PLN que contribuya a reducir el tiempo de redacción manual de guiones, aliviando así la sobrecarga de los voluntarios del Club SoftMedia y, al mismo tiempo, fortaleciendo la frecuencia y consistencia de las publicaciones institucionales. (Arias, 2019)

3.2.2 Investigación exploratoria y descriptiva

Por más que el estudio pudiera reducirse a un simple listado de características, su carácter descriptivo lo lleva más lejos: describe roles, etapas, tiempos aproximados, número de revisiones y percepciones de los integrantes sobre automatización. Y esa riqueza, sumada a su enfoque exploratorio para comprender un problema operativo en un grupo reducido, es lo que le confiere profundidad. (Hernández Sampieri & Mendoza Torres, 2023)

3.3 Enfoque de la Investigación

Apoyándose en entrevistas semiestructuradas y observación directa para el componente cualitativo, la investigación adopta un enfoque mixto con predominio cualitativo y complemento cuantitativo. Así se logra una comprensión profunda de percepciones,

dificultades y expectativas de los integrantes del Club SoftMedia frente al proceso de redacción.

el componente cuantitativo recurre a una encuesta estructurada aplicada a toda la población (n = 10), cuyos resultados expresados en frecuencias y porcentajes cuantifican la magnitud del problema en tres dimensiones: tiempos de preparación, número de correcciones y nivel de aceptación de la automatización.

Integrando ambos componentes en una misma etapa de diagnóstico, se busca contrastar y complementar los hallazgos cualitativos con evidencia cuantitativa; esta estrategia corresponde a un diseño mixto de tipo concurrente. (Hernández Sampieri & Mendoza Torres, 2023)

3.3.1 Método Inductivo

La necesidad de un mecanismo de apoyo que estructure y agilice la generación de guiones audiovisuales es la conclusión general que se infiere, mediante el método inductivo, a partir de observaciones particulares del Club SoftMedia: retrasos, repetición de correcciones, dependencia de improvisación y dificultad para recordar detalles. (Hernández Sampieri & Mendoza Torres, 2023)

3.3.2 Método analítico-sintético

Se aplicó el método analítico–sintético al descomponer el proceso actual de creación de guiones en fases (solicitud, recolección de datos, borrador, revisiones y entrega) para identificar cuellos de botella. Sintetizando posteriormente un flujo optimizado con apoyo de PLN, se logró reducir pasos y tiempo total, y con ello se buscó proponer un proceso operativo más eficiente y viable para el club. (Celi-Párraga et al., 2023)

Tabla 1 Comparación del flujo de trabajo actual y propuesto para la generación de guiones audiovisuales

Fase actual (proceso manual)	**Tiempo promedio**	**Fase nueva (con sistema PLN)**	**Tiempo promedio**
1. Solicitud del video	5-10 min	1. Solicitud del video	5-10 min
2. Recolección de información	30-60 min	2. Ingreso de datos clave	3-5 min
3. Redacción del borrador inicial	3-6 horas	3.Generación automática	< 40 segundos

4. Primera revisión interna	30-60 min		
5. Segunda revisión colectiva	45-90 min	4. Revisión mínima y aprobación	5-10 min
6. Ajustes finales y aprobación	30-60 min		
7. Entrega al equipo de grabación	5-10 min		
Total	**4-8 horas**	**Total**	**< 15 minutos**

3.4 Fuentes de Información

3.4.1 Fuentes primarias

Se trabajó con información obtenida directamente del contexto de SoftMedia, utilizando técnicas cualitativas y cuantitativas para captar tanto percepciones como datos estructurados del proceso.

3.4.1.1 Entrevistas semiestructuradas

La entrevista semiestructurada permite obtener información profunda manteniendo una guía de preguntas flexible que se adapta al contexto y a respuestas emergentes. En este proyecto, se aplicaron entrevistas a los integrantes del Club SoftMedia para identificar: fases reales del proceso, dificultades, niveles de carga, y expectativas ante una herramienta de automatización. (Flick, 2022)

3.4.1.2 Observación Directa

Cabe señalar que la observación directa consiste en registrar de forma sistemática dinámicas y comportamientos en el contexto natural. Y en este estudio, justamente, permitió confirmar los tiempos aproximados invertidos en correcciones, la interacción entre miembros durante la revisión y los puntos donde la carga de trabajo tiende a acumularse, mediante una lista de verificación diseñada específicamente para registrar estos aspectos durante las sesiones de trabajo del Club SoftMedia (ver instrumento completo en el Anexo F) (Denzin & Lincoln, 2019)

3.4.1.3 Encuesta estructurada

Por más que una encuesta pueda parecer un instrumento sencillo, su aplicación estructurada a los integrantes del club permitió obtener datos comparables y sintetizables del proceso. Conviene precisar que el cuestionario, estructurado con preguntas cerradas, recogió

información sobre roles, tiempos de preparación, número de correcciones, niveles de frustración y aceptación de una herramienta automática. Esta técnica se justifica, justamente, porque permite medir tendencias internas del grupo y traducir percepciones en porcentajes que resultan útiles para el diagnóstico. (Palella & Martins, 2019)

3.4.2 Fuentes secundarias

Proporcionando información elaborada por otros autores, las fuentes secundarias ofrecen un sustento teórico, contextual y comparativo que respalda el desarrollo del estudio.

Las fuentes secundarias permitieron sustentar el marco metodológico y comparar el caso del club con prácticas de investigación y desarrollo de software. Se consultaron textos de metodología de investigación y de ingeniería de software, además de documentación interna de SoftMedia (actas, estadísticas de producción o reportes disponibles) para contextualizar el diagnóstico. (Hernández Sampieri & Mendoza Torres, 2023)

3.5 Operacionalización de Variables

Con base en la variable independiente (Procesamiento de Lenguaje Natural) y la variable dependiente (guiones audiovisuales) definidas en el Capítulo II, se presenta a continuación su operacionalización, vinculando cada variable con sus indicadores y con la técnica e instrumento que permite medirla.

Tabla 2 Operación de variables independientes y dependientes

Variable	Definición operacional	Indicadores	Técnica e ítem
Independiente: Procesamiento de Lenguaje Natural (PLN)	Capacidad del sistema para generar de forma automática un guion estructurado a partir de los datos de un evento	Tiempo de generación (segundos); tasa de éxito en la separación de secciones (introducción/desarrollo/cierre); tasa de error de generación	Encuesta (preguntas 9-12); pruebas de caja negra (CP-01 a CP-04); logs técnicos del módulo
Dependiente: Guiones audiovisuales	Calidad y utilidad percibida del guion generado para la producción audiovisual del club	Coherencia narrativa percibida (escala Likert); número de correcciones manuales tras la generación; nivel de aceptación del equipo	Encuesta (preguntas 4-8); entrevista semiestructurada; observación directa

3.6 Estrategia operacional para la recolección de datos

3.6.1 Población

La población del estudio estuvo conformada por el universo total de 10 miembros activos del Club SoftMedia de la ULEAM Extensión El Carmen, integrada por estudiantes voluntarios (redactores/editores) y coordinadores del club, quienes participan directamente en la planificación, redacción, grabación y edición de contenidos institucionales. (Rodríguez Gómez et al., 2019)

3.6.2 Segmentación

Para fines descriptivos, la población puede segmentarse por función principal dentro del flujo de producción (por ejemplo: redacción/locución, grabación, edición, coordinación). Esta segmentación permite interpretar diferencias de carga y percepción sobre la automatización del guion. (Flick, 2022)

Se segmentó en redactores (80 %) y coordinadores (20 %).

3.6.3 Técnica de muestreo

Dado que la población es reducida (N = 10), se trabajó con muestra censal (censo), es decir, se incluyó a todos los miembros activos disponibles. Por ende, no se aplicó ninguna técnica de muestreo probabilístico y mucho menos se el uso de tamaño de una muestra. (Denzin & Lincoln, 2019)

3.6.4 Tamaño de la muestra

La muestra corresponde al total de la población: n = 10 integrantes del Club SoftMedia.

3.6.5 Análisis de herramientas de recolección

3.6.5.1 Entrevista – Guía de entrevista

Diseñando una guía semiestructurada, se buscó recabar información sobre el proceso actual, dificultades, criterios de calidad del guion y expectativas de automatización. Y eso, en la práctica, también dejó espacio para ahondar en aspectos emergentes que surgían en la conversación. (Flick, 2022)

3.6.5.2 Observación

Se utilizó una lista de verificación para registrar, en sesiones de trabajo del club, tiempos aproximados dedicados a redacción, número de correcciones, dificultades recurrentes y coordinación entre miembros, con el propósito de contrastar lo declarado en entrevistas y

encuesta con lo observado en práctica, mediante el instrumento detallado en el Anexo F (Palella & Martins, 2019)

3.6.5.3 Encuesta

El cuestionario se elaboró con preguntas cerradas (opción múltiple) para identificar patrones internos del club: tiempos, etapas con mayor consumo de esfuerzo, cantidad de correcciones, frustraciones y nivel de aceptación de una herramienta automática basada en PLN. (Palella & Martins, 2019)

3.6.5.4 Plan de recolección de datos

Tabla 3 Plan de recolección de datos según instrumentos, fechas y responsables

Actividad	Instrumento	Fecha	Responsable
Aplicación de entrevistas	Guía de 8 preguntas	15-30 oct 2025	Investigador
Observación participante	Lista de verificación	20 oct - 10 nov 2025	Investigador

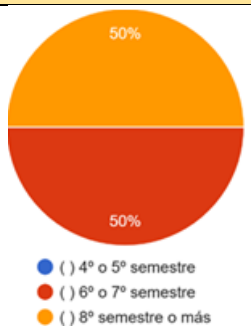
3.7 Análisis y Presentación de Resultados

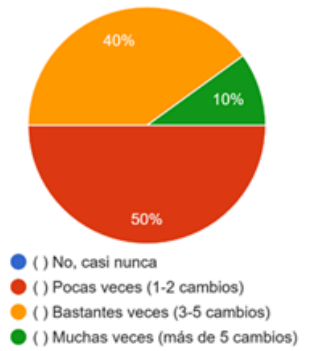
3.7.1 Tabulación y análisis de datos

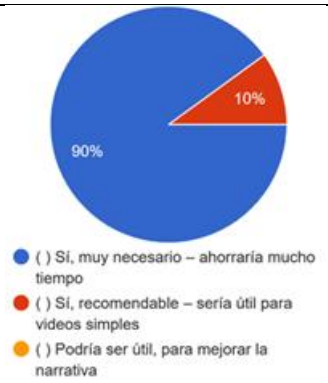
3.7.1.1 Encuesta aplicada a estudiantes del club SoftMedia de la Uleam El Carmen en el sector de la planta central.

Tabla 4 Resultados de la encuesta aplicada a los integrantes del Club SoftMedia

Pregunta	Resultado	Análisis
1. ¿Cuál es tu rol actual en el Club SoftMedia?		La mayoría de los integrantes se enfoca en roles de apoyo y edición de videos (80%), lo que indica que la preparación de narración es una tarea compartida pero no especializada.

Pregunta	Resultado	Análisis
2. ¿En qué semestre estás?		El grupo se divide en semestres avanzados, reflejando experiencia moderada que agrava la sobrecarga en la preparación de videos
3. ¿Cuánto tiempo llevas participando en la grabación o edición de videos del club?		El 50% lleva menos de 6 meses, explicando la falta de estandarización en los procesos, lo que conlleva un retrasa más significativo a la hora de edición o planificación de un video
4. Cuando llega la solicitud de un video, ¿cómo preparan normalmente lo que se va a decir o narrar?		El 40% de los integrantes depende de improvisación o textos completos, confirmando la ausencia de un proceso eficiente, mientras que un 50% depende de alguien que narre el guion del video, mientras que un 10% hace una narración directa.
5. ¿Cuánto tiempo aproximado te toma a ti preparar esas notas o el texto que se usa para narrar?		El 50% invierte más de 1 hora en anotaciones la para la narración, representando una sobrecarga significativa, mientras que un 30% le toma menos de una hora en la preparación narrativa y un 20% le toma entre 4 a 6 horas preparar una narrativa para el video.

Pregunta	Resultado	Análisis
6. ¿Qué parte de preparar el contenido del video (lo que se dice) es la que más te cuesta o donde se pierde más tiempo?		El 50% de los participantes reporta dificultades para alcanzar naturalidad en los textos, mientras que el otro 50% señala problemas con el recuerdo de detalles. Ambas cifras, en su conjunto, evidencian áreas concretas donde la automatización podría tener un papel relevante.
7. ¿Suelen hacer varios cambios o correcciones a lo que se va a narrar antes del video definitivo?		El 50% de los casos requiere tres o más modificaciones, lo que genera un retraso considerable en el desarrollo del proceso.
8. ¿Qué es lo que más te frustra o molesta cuando preparan lo que se va a decir en los videos?		El 80% de los participantes manifiesta frustración por el tiempo perdido o por olvidos en la narrativa, lo que evidencia de forma clara la necesidad de automatización.
9. Si pudieras cambiar una sola cosa de cómo preparamos el contenido de los videos, sería:		El 90% prioriza un sistema automático, confirmando la demanda principal.

Pregunta	Resultado	Análisis
10. ¿Qué tipo de ayuda o herramienta te gustaría tener para preparar más rápido lo que se narra en los videos?		El 90% de los encuestados se inclina por soluciones basadas en IA para mejorar las narrativas audiovisuales
11. ¿Has usado alguna vez herramientas como ChatGPT o similares para escribir textos o notas?		La aceptación del sistema se ve facilitada por un dato elocuente: el 90% ya tiene experiencia positiva con IA.
12. ¿Qué ventajas ves en usar una herramienta automática que genere sola el texto para narrar los videos del club?		el 90% subraya el ahorro de tiempo al implementar una herramienta que agiliza la creación de guiones audiovisuales, entonces queda evidenciado que esa es la ventaja percibida como más relevante.
13. ¿Crees que sería recomendable o necesario que el club tenga un sistema automático que genere el texto para narrar los videos (solo llenando un formulario corto con los datos del evento)?		El 100% ve necesario o recomendable el sistema o lo consideran muy necesario, ya que facilitaría bastante en la edición de videos

3.7.2 Presentación y descripción de los resultados obtenidos

En vista de los resultados de la encuesta se aprecia claramente una postura positiva por parte de los encuestados respecto al sistema automatizado para la generación de textos narrativos en los videos institucionales, obteniendo como resultados importantes el uso predominante de improvisación o notas rápidas (40%) y la inversión de más de 1 hora en preparación (50%), lo que representa una oportunidad adecuada para implementar este módulo web con inteligencia artificial que sea capaz de generar textos coherentes a partir de datos simples del evento, permitiendo así sostener procesos más eficientes para el Club SoftMedia y los videos que se producen en la extensión El Carmen. Así mismo, más del 50% de los encuestados destacan dificultades en la naturalidad de la narración o recuerdo de detalles, dando como punto de vista la necesidad de herramientas que estructuren el contenido automáticamente. En la pregunta seis los encuestados en su mayoría señalan problemas en recordar detalles (50%) o hacer que la narración suene más natural (50%), así mismo en la pregunta ocho con más del 80% consideran frustrantes la pérdida de tiempo y olvidos, por otra parte, la pregunta nueve confirma la demanda de automatización con el 90% prefiriendo un sistema que genere el texto. La pregunta diez refleja que en su mayoría de los encuestados prefieren un sistema automático (60%) o IA (30%), y un porcentaje menor opta por organización de notas, aunque en la pregunta once se aprecia un 90% de experiencia positiva con IA, con ello también se resalta en la pregunta doce el impacto positivo en ahorro de tiempo (90%), evidenciando que la amplia mayoría de los 10 encuestados (entre el 80% y el 100%, según la pregunta) se muestra a favor de la automatización.

Analizando y diseñando la solución tecnológica en etapas posteriores, se identificó que las necesidades del Club SoftMedia no eran solo narrativas. Y entonces se sumaron funcionalidades para la estructuración técnica: organización de escenas, secuencias y apoyos, sin que eso desviara el foco del objetivo principal optimizar la creación audiovisual con IA.

3.7.3 Informe final del análisis de los datos

Partiendo de la ausencia de un proceso estructurado para preparar narraciones en los videos del Club SoftMedia, se identifica una oportunidad clara: implementar un módulo web con IA que, con datos simples del usuario, genere borradores estructurados para contenidos audiovisuales. Eso, en la práctica, facilitaría la construcción narrativa y la organización preliminar de elementos de producción, y con ello se sostendrían procesos más eficientes para el club y sus integrantes.

Cabe recalcar que este tipo de módulos web con inteligencia artificial parten de un modelo entrenado para la generación de textos coherentes en español, como los modelos de PLN en bibliotecas como Hugging Face o spaCy, capaces de estructurar, interpretar y generar narraciones basadas en entradas simples como títulos, fechas y puntos clave del evento, haciendo automático el proceso. El uso de la encuesta reflejó la utilidad, confianza y necesidad del sistema en el club, con niveles de aceptación alta y un visto positivo por medio de la población encuestada.

CAPÍTULO IV

4 MARCO PROPOSITIVO (ELABORACIÓN DE LA PROPUESTA)

4.1 Introducción

En el presente capítulo se desarrolla de manera integral la propuesta tecnológica orientada a resolver la problemática identificada en los capítulos previos: la redacción manual y lenta de guiones audiovisuales en el Club SoftMedia de la ULEAM Extensión El Carmen, lo que genera retrasos en la producción de contenido, sobrecarga laboral en los estudiantes voluntarios y una disminución en la frecuencia y calidad de las publicaciones institucionales.

Partiendo de la necesidad de resolver esta situación, se propone construir un sistema con PLN que genere guiones estructurados a partir de datos simples que ingresen los miembros del club. El sistema, eso sí, no solo agilizará la redacción del texto narrativo, sino que también habilitará la edición, el almacenamiento, la exportación y la evaluación continua, articulando un flujo más eficiente y colaborativo.

A lo largo de este capítulo se describen con detalle los recursos humanos, tecnológicos y económicos necesarios para la implementación del proyecto, así como los requisitos funcionales y no funcionales derivados del diagnóstico realizado (Capítulo III). También se definen los roles de los usuarios, se presentan los diagramas UML que modelan la arquitectura del software y se especifican las interfaces gráficas diseñadas bajo principios de usabilidad y experiencia de usuario (UX/UI).

Vale la pena precisar que la metodología seleccionada para planificar el desarrollo del marco de trabajo es Scrum, con sus iteraciones cortas los Sprints, que facilitan la entrega de incrementos funcionales y la recogida de retroalimentación continua de los usuarios del Club SoftMedia. La razón, por supuesto, es clara: el proyecto demanda validación temprana con los voluntarios y flexibilidad para ajustar prioridades conforme a la experiencia real de uso.

El propósito que da sentido a la diversidad tecnológica del ecosistema es, justamente, adaptar el sistema al lenguaje y al estilo institucional. En la práctica, esa decisión se materializa en un conjunto de herramientas: Python con FastAPI para el backend y la API; React con Vite y TailwindCSS para el frontend; PostgreSQL para la base de datos; Docker para el despliegue; y, en el ámbito del PLN, Transformers de Hugging Face y PyTorch en su variante CPU, con un modelo GPT-2 en español afinado localmente. Todo ello, respaldado por herramientas abiertas y ampliamente soportadas.

Por más que las pruebas en entornos reales sean impredecibles, se ha diseñado un plan que combina caja negra y caja blanca con usuarios del Club SoftMedia para medir tiempos, calidad narrativa y satisfacción. Los resultados, que se presentarán en el Capítulo V, permitirán validar los objetivos del Capítulo I y comprobar si la automatización con PLN realmente optimiza la producción audiovisual en universidades con recursos acotados.

4.2 Descripción de la propuesta

La propuesta central de este proyecto integrador consiste en **diseñar, desarrollar e implementar un sistema informático web basado en Procesamiento de Lenguaje Natural (PLN)** que permita a los miembros del Club SoftMedia de la ULEAM Extensión El Carmen generar de manera semiautomática guiones audiovisuales para videos institucionales, académicos, culturales y sociales.

Entrenado con guiones anteriores del club, el modelo de lenguaje que sustenta el sistema recibe la información clave de un evento a través de un formulario sencillo y produce un borrador con tres secciones definidas: introducción (enganche y presentación), desarrollo (detalles, entrevistas o hitos) y cierre (llamada a la acción o reflexión). Luego, el redactor puede editar, corregir, almacenar, exportar a PDF o texto plano y evaluar la calidad, lo que alimenta el ciclo de retroalimentación del sistema.

Cabe precisar que el presente trabajo de titulación corresponde al diseño, desarrollo, pruebas y despliegue del módulo de PLN (backend especializado en la generación de guiones) como componente autocontenido e integrable mediante API REST. El frontend web y el backend principal de orquestación (autenticación, historial, editor y exportación) forman parte del sistema mayor del Club SoftMedia en el que se integra este módulo, y fueron desarrollados por otros integrantes del equipo de SoftMedia, tal como se detalla en la sección de Roles y Responsabilidades de este capítulo; dicho desarrollo queda fuera del alcance individual evaluado en este documento.

El desarrollo del módulo de PLN se llevó a cabo bajo la metodología ágil Scrum, descrita conceptualmente en el apartado 2.3.3 del Capítulo II y aplicada en cuatro sprints iterativos con revisiones periódicas junto al Product Owner y validaciones tempranas con los miembros del Club SoftMedia, tal como se detalla en los apartados de Planificación del Sprint y Backlog del Producto de este mismo capítulo.

4.3 Determinación de recursos

4.3.1 Humanos

Los recursos humanos estarán involucrados de manera directa e indirecta, ya que son participes directos de este proyecto.

Tabla 5 Talento humano involucrado en el desarrollo del módulo

Personal	Función
Investigador / Desarrollador (Jerson Valencia)	Desarrollo e implementación del sistema (backend, frontend, modelo PLN), entrenamiento del modelo, pruebas y despliegue.
Miembros del Club SoftMedia (10 estudiantes voluntarios)	Pruebas piloto, validación de guiones generados, retroalimentación para mejora del modelo.
Tutor de titulación (Ing. Jefferson Arca Zavala)	Supervisión técnica, validación de requisitos y resultados, acompañamiento metodológico.

4.3.2 Tecnológicos

El uso de recursos tecnológicos es primordial para cumplir a cabalidad con este proyecto por el cual son necesarios recursos tanto de software como hardware con sus especificaciones para el equilibrio entre la capacidad óptima y la parte económica.

Tabla 6 Especificaciones de hardware requeridas para el desarrollo e implementación del módulo PLN

Hardware	Especificaciones
Computadora de desarrollo	Procesador Intel Core i5/i7 o AMD Ryzen 5/7, 16 GB RAM, 256 GB SSD, GPU NVIDIA (recomendada para entrenamiento del modelo PLN).
Servidor de despliegue (opcional)	VPS con 4 vCPU, 8 GB RAM, 80 GB SSD (ej. DigitalOcean, AWS EC2) o servidor local de la ULEAM.
Dispositivos de prueba	Disipativos de los miembros de SoftMedia para realizar las pruebas respectivas.

Tabla 7 Especificaciones de Software requeridas para el desarrollo e implementación del módulo PLN

Software	**Especificaciones**
IDE	• Visual Studio Code
Framework	• React (con Vite) • TypeScript • TailwindCSS
Python 3.10+	Lenguaje principal para backend y modelo PLN.
SQLAlchemy	ORM para conexión y manejo de base de datos.
PostgreSQL	Base de datos relacional para almacenar usuarios, guiones, eventos y evaluaciones.
Transformers (Hugging Face), PyTorch	Procesamiento de lenguaje natural: carga del modelo GPT-2 afinado y generación de texto.
TailwindCSS	Estilizado de interfaces de usuario.
Docker / Docker Compose	Contenerización y orquestación de servicios (backend, frontend, BD).
Git / GitHub	Control de versiones y respaldo del código.
Google Colab (opcional)	Entrenamiento del modelo PLN en GPU gratuita.

Aunque la disponibilidad de herramientas sea un factor a considerar, la selección de los recursos tecnológicos de la Tabla 3 se fundamenta también en la literatura sobre ingeniería de software y automatización inteligente. Python, con su ecosistema consolidado de librerías de PLN como Transformers y PyTorch, permite construir prototipos de IA de forma ágil en contextos académicos de recursos acotados, manteniendo al mismo tiempo la calidad y la mantenibilidad del código.

Si bien la contenerización con Docker y Docker Compose podría considerarse una práctica estándar, su adopción en este proyecto se alinea con los principios de automatización inteligente descritos en la literatura sobre integración de sistemas. Empaquetar los servicios como unidades autocontenidas, según esa visión, reduce riesgos por entornos dispares entre

desarrollo y producción y, al mismo tiempo, facilita el despliegue en infraestructuras con pocos recursos, como la que opera la Uleam Extensión El Carmen.

Apoyándose en criterios como su carácter de código abierto, su madurez y su compatibilidad nativa con SQLAlchemy, PostgreSQL fue la opción elegida entre las bases de datos relacionales. Y esa elección no es menor: la integridad referencial de los datos generados por el sistema (usuarios, guiones, eventos) es un aspecto crítico cuando se trabaja con información sensible de la institución.

Empleando herramientas gratuitas o de bajo costo Google Colab para entrenar el modelo, y Git con GitHub para el control de versiones, el proyecto se mantiene fiel al enfoque de sostenibilidad tecnológica del Capítulo I. Así, otros clubes o extensiones universitarias pueden replicarlo sin gastar en licencias, y eso, precisamente, encaja con el impacto tecnológico que se esperaba desde el inicio.

4.3.3 Económicos (presupuesto)

El recurso económico es esencial para la implantación exitosa del proyecto por ello se buscó en el mercado los equipos necesarios más accesibles y de capacidades óptimas para el funcionamiento correcto del sistema, en la siguiente tabla se detalla al respecto.

Tabla 8 Presupuesto estimado para la implementación del módulo

Concepto	Cantidad	Característica	C/U	Subtotal
Computadora	1	Para la visualizar el registro automatizado	400.00$	$400.00
Servidor VPS (6 meses)	1	4 vCPU, 8 GB RAM	$40.00/mes	$240.00
Internet	6 meses	Conexión mínima 20 Mbps	$25.00/mes	$150.00
Total				$790.00

Desarrollo de la propuesta mediante metodología Scrum

La metodología Scrum fue seleccionada debido a su capacidad para adaptarse a cambios progresivos en los requerimientos funcionales del sistema, permitiendo incorporar mejoras identificadas durante cada iteración del desarrollo. Este enfoque facilitó la evolución del

módulo de generación de guiones, permitiendo ajustar características funcionales conforme se obtenía retroalimentación de los usuarios del Club SoftMedia.

Conviene ampliar el fundamento teórico de la metodología Scrum utilizada en el desarrollo del módulo, dado que su elección determina en buena medida la organización del presente capítulo. Scrum es un marco de trabajo ágil basado en ciclos cortos e iterativos denominados Sprints, cuya duración fija permite entregar de manera incremental funcionalidades utilizables, revisadas y ajustadas conforme a la retroalimentación obtenida de los interesados en cada iteración.

A diferencia de los enfoques tradicionales de desarrollo en cascada, Scrum no exige contar con la totalidad de los requerimientos definidos desde el inicio del proyecto, lo cual resulta especialmente conveniente para un contexto como el de SoftMedia, en el que las necesidades reales de los miembros del club respecto de la estructura y el estilo de los guiones solo pudieron precisarse a medida que se generaban los primeros borradores del módulo y se recogía la opinión del equipo.

El marco de trabajo se sustenta en tres roles principales, Product Owner, Scrum Master y Development Team, tres artefactos, Product Backlog, Sprint Backlog e Incremento, y una serie de eventos formales, Sprint Planning, Daily Scrum, Sprint Review y Sprint Retrospective, que en conjunto garantizan la transparencia del proceso, la inspección continua de los avances y la adaptación oportuna del plan de trabajo ante cambios o hallazgos imprevistos. En los apartados siguientes de este capítulo se detallan los roles asumidos por cada participante del proyecto, así como los cuatro Sprints planificados, sus respectivos backlogs y los eventos de revisión llevados a cabo durante el desarrollo del módulo.

Cabe destacar que, si bien Scrum fue concebido originalmente para equipos de desarrollo de software de mayor tamaño, su adaptación a proyectos académicos individuales, como el presente trabajo de titulación, ha demostrado ser viable, siempre que se mantenga la disciplina de las revisiones periódicas y se documenten adecuadamente los incrementos logrados en cada iteración, tal como se evidencia en la planificación de Sprints descrita más adelante en este mismo capítulo.

4.3.4 Descripción del producto

4.3.4.1 Propósito del módulo

El módulo de PLN tiene como propósito principal automatizar la redacción de guiones audiovisuales para los videos producidos por el Club SoftMedia de la ULEAM Extensión El Carmen.

Generar un borrador estructurado en menos de 40 segundos, con edición mínima posterior, es lo que logra el módulo frente a las 4 a 8 horas que los estudiantes voluntarios dedican hoy a redactar manualmente cada video.

4.3.4.2 Funcionalidades clave del módulo

- A través de una petición POST al endpoint /generar-guion, el sistema recibe datos estructurados que comprenden título, fecha, lugar, lista de puntos clave, tipo de evento académico, cultural, deportivo o social y público objetivo.
- A partir de los campos ya validados por Pydantic, el módulo construye directamente el contexto textual del evento —título, tipo, lugar, fecha y público— y lo inserta en los prompts de generación. No hay, en este punto, una etapa adicional de preprocesamiento lingüístico.
- El sistema emplea el modelo datificate/gpt2-small-spanish un GPT-2 pequeño en español y lo afina con 120 ejemplos de guiones reales del club para realizar la generación automática del guion.
- Evitar etiquetas de sección y postprocesamiento de separación es el resultado de que el módulo realice tres llamadas independientes al modelo, una por bloque narrativo (introducción, desarrollo y cierre), cada una con su propio prompt específico..
- Entregando el resultado en JSON, se incluyen metadatos como el tiempo de generación (ms) y la versión del modelo, lo que permite trazabilidad y control.
- Guardar marca de tiempo, duración, longitud del prompt, longitud de la salida y versión del modelo, sin almacenar el contenido del guion por privacidad y para no saturar el almacenamiento, es lo que contempla el registro de logs técnicos anonimizados.
- Se ha incluido un endpoint de salud (/health) que permite al backend principal verificar en todo momento que el servicio se encuentra operativo.
- Pese a que se trata de un endpoint opcional (/reload-model), su carácter protegido y su capacidad para recargar el modelo en caliente sin detener el servicio, lo convierten en una herramienta valiosa para aplicar actualizaciones de forma continua.

Basándose en los principios de la arquitectura orientada a microservicios, el diseño de las funcionalidades descritas permite que cada componente se desarrolle, desplace y escale de forma independiente, comunicándose con el resto mediante contratos de interfaz bien definidos una API REST, en este caso. Y esa decisión, justamente, es especialmente adecuada para el módulo de PLN: el componente de generación de guiones puede evolucionar (por ejemplo, con un modelo más avanzado o reentrenado con más datos) sin que el backend principal o el frontend, desarrollados por el resto del equipo de SoftMedia, tengan que modificarse..

Exponiendo el módulo mediante el endpoint /generar-guion y respetando las convenciones REST (verbos HTTP, códigos de estado y JSON), el diseño favorece la interoperabilidad con cualquier cliente que quiera conectarse ahora o después. Y eso, en el fondo, es lo que buscan los sistemas actuales: utilidad real, iteración y validación temprana con los usuarios.

Por más que los endpoints de salud y recarga de modelo puedan parecer detalles técnicos secundarios, su inclusión responde a buenas prácticas de observabilidad y mantenibilidad que la literatura de automatización inteligente recomienda para todo servicio autocontenido: exponer mecanismos que permitan verificar disponibilidad y aplicar actualizaciones sin interrupciones prolongada. En SoftMedia, esos mecanismos son especialmente relevantes, porque la continuidad del flujo de generación de guiones es crítica para que los tiempos de cobertura de los eventos no se vean afectados.

4.3.4.3 Usuarios del módulo

Tabla 9 Roles y responsabilidades de los usuarios del módulo

Actor	Descripción	Interacción con el módulo
Backend principal	Aplicación central desarrollada por otros miembros del equipo.	Enviando peticiones POST al endpoint /generar-guion, se incluyen los datos del evento como parte de la solicitud.
Desarrollador del módulo (Jerson Valencia)	Responsable del entrenamiento y despliegue del protipo.	Entrena el modelo, revisa logs, recarga el modelo, monitoriza rendimiento.

Miembros del Club SoftMedia	Usuarios finales que consumen el sistema completo.	No interactúan directamente con el módulo, sino a través del frontend y backend.
Administrador del sistema	Puede recargar el modelo o verificar el estado.	Usa el endpoint /reload-model (con autenticación).

4.3.4.4 Condiciones de éxito del módulo

- Tiempo de generación inferior a 40 segundos en el entorno de producción (VPS con 4 vCPU, 8 GB RAM).
- Tasa de éxito del 99% (peticiones que terminan con código 2xx).
- Coherencia narrativa aceptable para el 80% de los guiones generados, medida mediante encuesta a los 10 miembros del club.
- Integración sin fricciones con el backend principal: el contrato de la API (OpenAPI) debe cumplirse al 100%.
- Consumo de recursos limitado: el contenedor no debe exceder los 4 GB de RAM ni utilizar más del 70% de una CPU durante la generación.

4.3.5 Historias de usuario del módulo (desde la perspectiva del consumidor y el desarrollador)

4.3.5.1 HU-PLN-01: Generación exitosa de guion

Tabla 10 Especificación de la historia de usuario: generación exitosa de un guion

Campo	**Detalle**
Id de historia	HU-PLN-01
Título	Generación básica de guion
Prioridad	Alta
Usuario	Backend principal del sistema
Riesgo de desarrollo	Alta
Tamaño de la tarea	Grande
Como:	Backend principal del sistema.
Quiero:	Enviar un JSON con los datos de un evento (título, fecha, lugar, puntos clave, tipo y público) y recibir un guion estructurado en introducción, desarrollo y cierre.

Para:	Mostrar el guion generado al usuario final (miembro de SoftMedia) sin necesidad de redacción manual.

4.3.5.1.1 Criterios de aceptación

1. El endpoint debe retornar un código HTTP 200 cuando la generación sea exitosa.
2. El cuerpo de la respuesta debe contener los campos:

- introduccion
- desarrollo
- cierre

3. Ningún campo puede permanecer vacío.
4. El tiempo de generación del borrador debe rondar los 40 segundos.
5. El texto generado debe redactarse en español y, además, guardar coherencia con el tipo de evento especificado en la entrada.
6. El sistema debe implementar mecanismos de control de repetición, con el fin de evitar frases redundantes o repetitivas en el texto generado.
7. El contenido generado debe ofrecer una estructura lógica y, a la vez, resultar legible para quien lo vaya a utilizar.
8. Detectando la ausencia de campos obligatorios en el JSON, el sistema debe devolver un mensaje de error descriptivo que oriente sobre qué falta o está mal formado.

4.3.5.1.2 Dependencias

- Modelo de PLN preentrenado y fine-tuned.
- Servidor con GPU/CPU disponible.
- Framework backend para exponer el endpoint.
- Librerías de inferencia y procesamiento de lenguaje natural.

4.3.5.2 HU-PLN-02: Manejo de errores de entrada

Tabla 11 Especificación de la historia de usuario: manejo de errores de entrada

Campo	Detalle
Id de historia	HU-PLN-02
Título	Validación de entrada y errores claros
Prioridad	Alta

Usuario	Backend principal
Riesgo de desarrollo	Media
Tamaño de la tarea	Mediana
Campo	Detalle
Como:	Backend principal.
Quiero:	Validar de forma rigurosa el JSON de entrada, el módulo debe responder con códigos de error estándar (400, 422) y mensajes descriptivos cuando los campos estén ausentes o mal formados.
Para:	Evitar que datos erróneos lleguen al modelo (lo que podría producir salidas incoherentes) y poder informar adecuadamente al frontend.

4.3.5.2.1 Criterios de aceptación

1. Si falta el campo título, el sistema debe devolver:

   { "detail": "El campo 'titulo' es requerido"}

   con código HTTP 422.

2. Si el campo fecha no presenta un formato válido por ejemplo, "2026-15-06", el sistema debe responder con el código de error HTTP 422.
3. Si puntos_clave no es un arreglo de cadenas o está vacío, el sistema debe responder con el código HTTP 422.
4. Si tipo_evento no pertenece a los valores permitidos:
   - académico
   - cultural
   - deportivo
   - social

   el sistema debe devolver código HTTP 422.

5. Los campos adicionales enviados en el JSON deben ser ignorados y no deben generar error.
6. Los mensajes de error deben ser claros, descriptivos y comprensibles para el frontend.

7. El sistema debe validar tipos de datos antes de enviar la información al modelo de PLN.

4.3.5.2.2 Dependencias

- Uso de Pydantic para validación de esquemas.
- Framework backend con soporte para validaciones automáticas.
- Middleware de manejo de errores HTTP.

4.3.5.3 HU-PLN-04: Almacenamiento y consulta de historial de guiones (responsabilidad del backend principal)

Tabla 12 Especificación de la historia de usuario: almacenamiento y consulta del historial de guiones

Campo	**Detalle**
Id de historia	HU-PLN-04
Título	Historial de guiones generados
Prioridad	Alta
Usuario	Miembro del Club SoftMedia
Riesgo de desarrollo	Media (para el equipo de backend)
Tamaño de la tarea	Grande (backend principal)
Como:	Miembro del Club SoftMedia.
Quiero:	Que después de generar un guion, este se guarde automáticamente en el sistema y pueda consultar todos los guiones que he generado anteriormente, con opción de verlos, editarlos y exportarlos nuevamente (a PDF o texto).
Para:	No perder el trabajo realizado, poder retomar guiones previos para eventos similares, y reutilizar ideas sin tener que volver a generar desde cero.

4.3.5.3.1 Criterios de aceptación

1. Al generar un guion, el backend principal debe almacenarlo automáticamente en la base de datos PostgreSQL asociado al id_usuario autenticado.

2. El usuario debe poder visualizar una lista paginada de sus guiones generados anteriormente.
3. Cada elemento del historial debe mostrar:
    - título del evento,
    - fecha de creación,
    - vista previa de la introducción.
4. Al seleccionar un guion del historial, el sistema debe mostrar el contenido completo:
    - introduccion
    - desarrollo
    - cierre
5. El usuario debe poder editar el contenido del guion y guardar los cambios realizados
6. El historial debe persistir entre sesiones y estar disponible desde cualquier dispositivo después de iniciar sesión.
7. El usuario debe poder exportar nuevamente un guion en formato:
    - PDF
    - TXT
8. Si el usuario no posee guiones guardados, el sistema debe mostrar un mensaje informativo indicando que no existen registros.
9. El tiempo de carga del historial no debe superar los 5 segundos para listas paginadas estándar.

4.3.5.3.2 Dependencias

- Se emplea PostgreSQL como sistema de base de datos, con una tabla específica destinada al almacenamiento de los guiones generados.
- El backend principal es desarrollado por otro miembro del equipo de SoftMedia.
- El frontend está destinado tanto a la visualización como a la edición de los guiones generados.
- El sistema incorpora librerías específicas para la exportación de guiones en formatos PDF y TXT.
- Sistema de autenticación de usuarios.

Aunque las historias de usuario puedan parecer simples, su redacción responde a un fundamento metodológico claro: son una técnica de especificación de requerimientos de los

marcos ágiles, que describen una funcionalidad desde la perspectiva del usuario final, siguiendo la estructura "Como [rol], quiero [funcionalidad], para [beneficio]". .

Que cada historia de usuario incluya una tabla de detalle, criterios de aceptación numerados y una sección de dependencias técnicas explícitas es el resultado de haber diseñado HU-PLN-01 a HU-PLN-04 conforme a los criterios INVEST (Alaimo, 2021). Esos criterios —Independiente, Negociable, Valiosa, Estimable, Small y Testeable— son, precisamente, los que recomienda la literatura para que las historias resulten útiles en la planificación de los Sprints.

Si bien la implementación de la historia HU-PLN-04 recae en el backend principal, que desarrolla otro miembro de SoftMedia, su presencia en este documento es imprescindible: permite definir el contrato de integración entre ese backend y el módulo de PLN, y así distinguir el alcance individual de este trabajo del que corresponde a los componentes colaborativos del equipo.

4.3.6 Diseño del Sistema / Descripción Técnica

4.3.6.1 Caso de uso: Módulo PLN

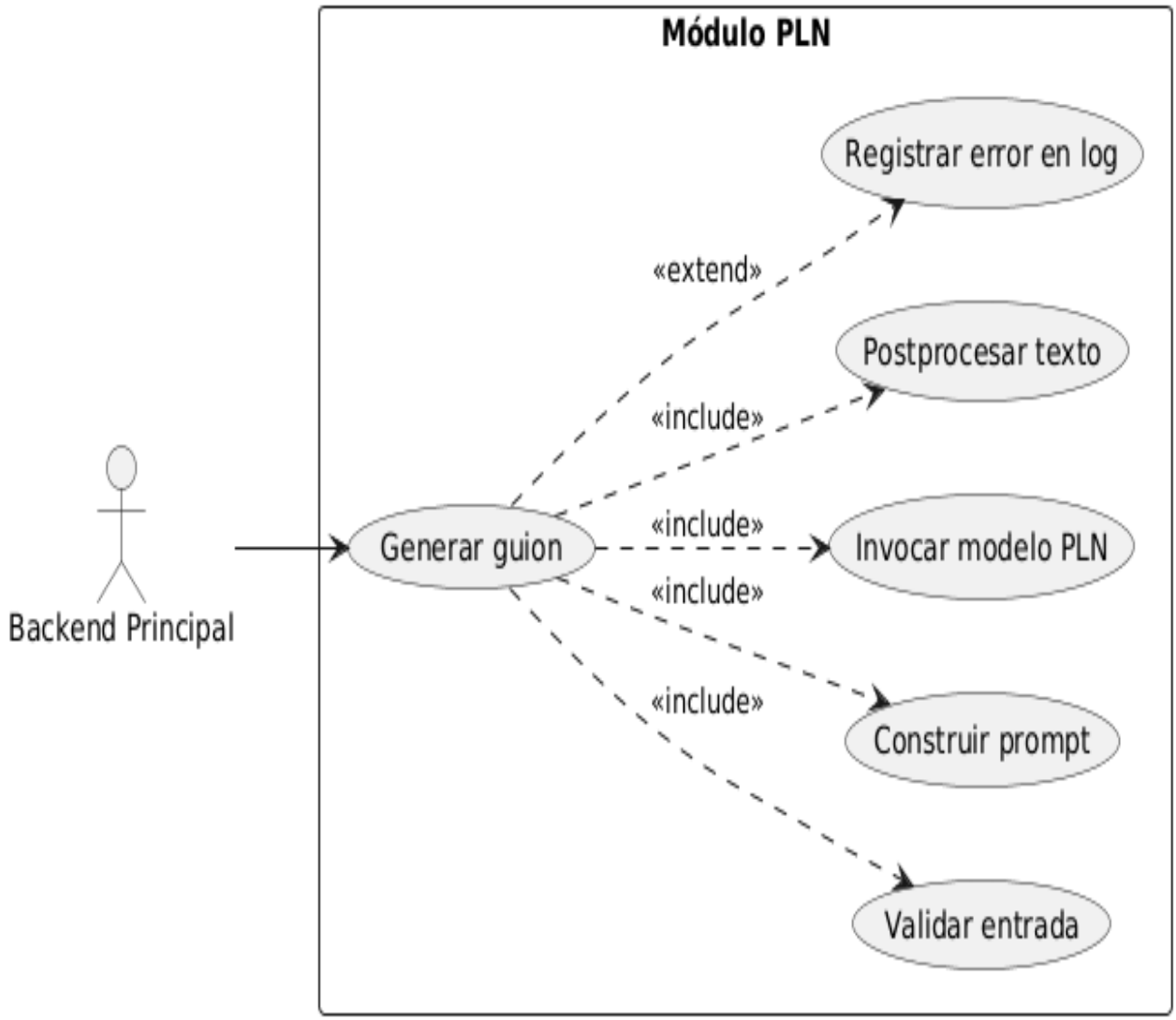


Ilustración 2: Diagrama caso de uso del módulo (PLN).

4.3.6.2 Caso de uso: Módulo PLN

Tabla 13 Ficha descriptiva del caso de uso "Generar guion" del módulo PLN

Campo	**Valor**
Nombre del caso de uso	Generar guion
Fecha	12/06/2026
Elaborado por	Valencia Hurtado Jerson Patricio
Actores	Backend principal (sistema)

Objetivo	Producir un borrador de guion audiovisual estructurado (introducción, desarrollo, cierre) a partir de datos de un evento.
Precondiciones	El módulo está en ejecución, el modelo de lenguaje está cargado en memoria y la API Key es válida.
Postcondiciones	Se genera un registro en el log técnico (sin el contenido del guion) y se devuelve el guion en formato JSON.
Medios para realizar la solicitud	API REST – HTTP POST a /generar-guion **Medios para realizar la solicitud**
Pasos	1. El backend envía un JSON con titulo, fecha, lugar, puntos_clave, tipo_evento, publico. 2. El módulo valida el esquema con Pydantic. 3. Construye un prompt estructurado con instrucciones y etiquetas. 4. Invoca el modelo GPT-2 fine-tuned con los parámetros configurados. 5. Postprocesa el texto para separar introducción, desarrollo y cierre. 6. Registra en logs/generations.log el tiempo de generación, longitudes y versión del modelo (sin el texto del guion). 7. Devuelve código HTTP 200 con el guion en JSON.
Situaciones excepcionales	- Esquema inválido (falta campo o tipo incorrecto) → 422 con mensaje.
Revisado por	Ing. Jefferson Omar Arca Zavala

4.3.6.3 Diagrama de base de datos

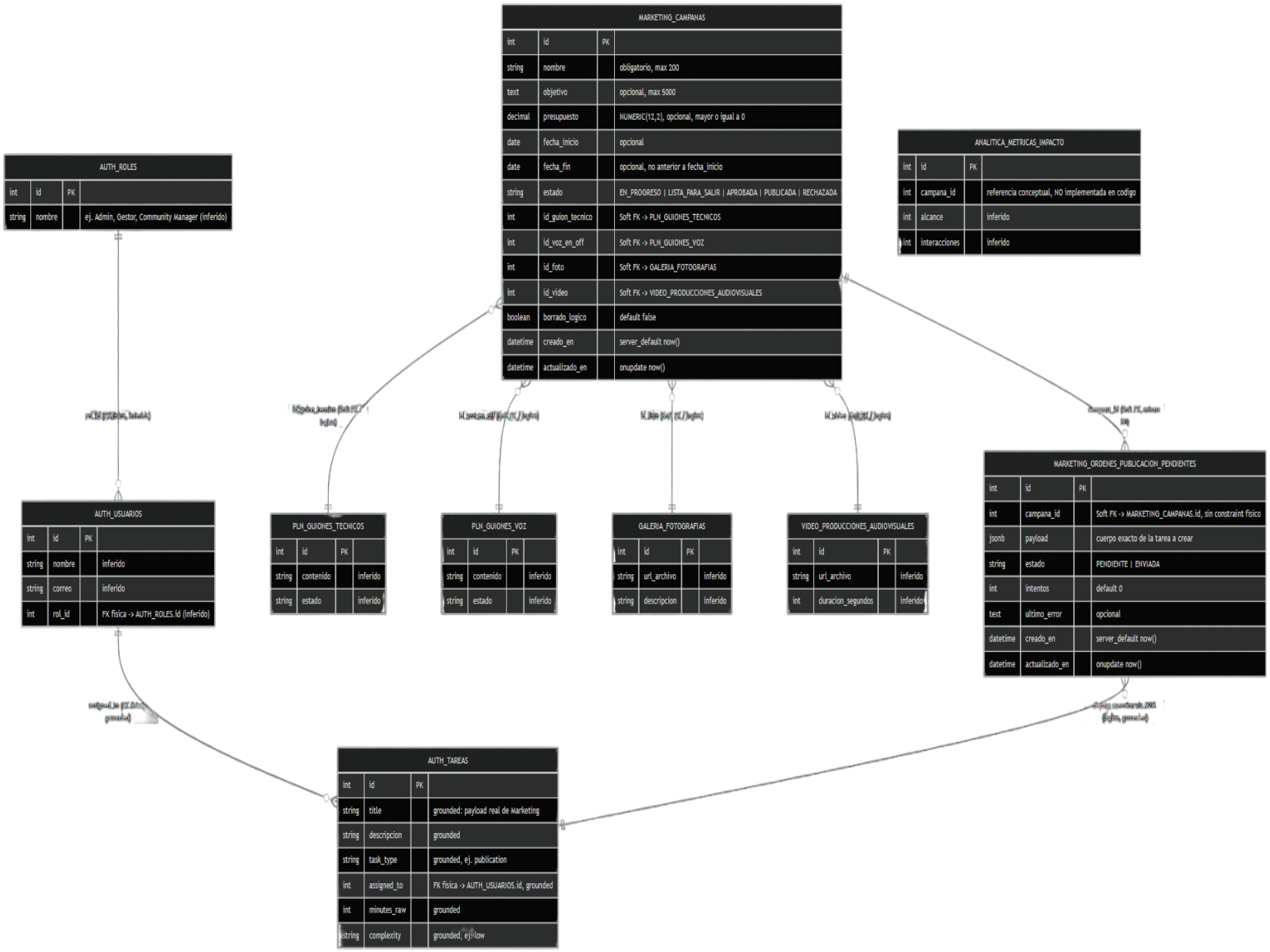


Ilustración 3 Diagrama de base de datos

4.3.6.4 Diagrama de Secuencia

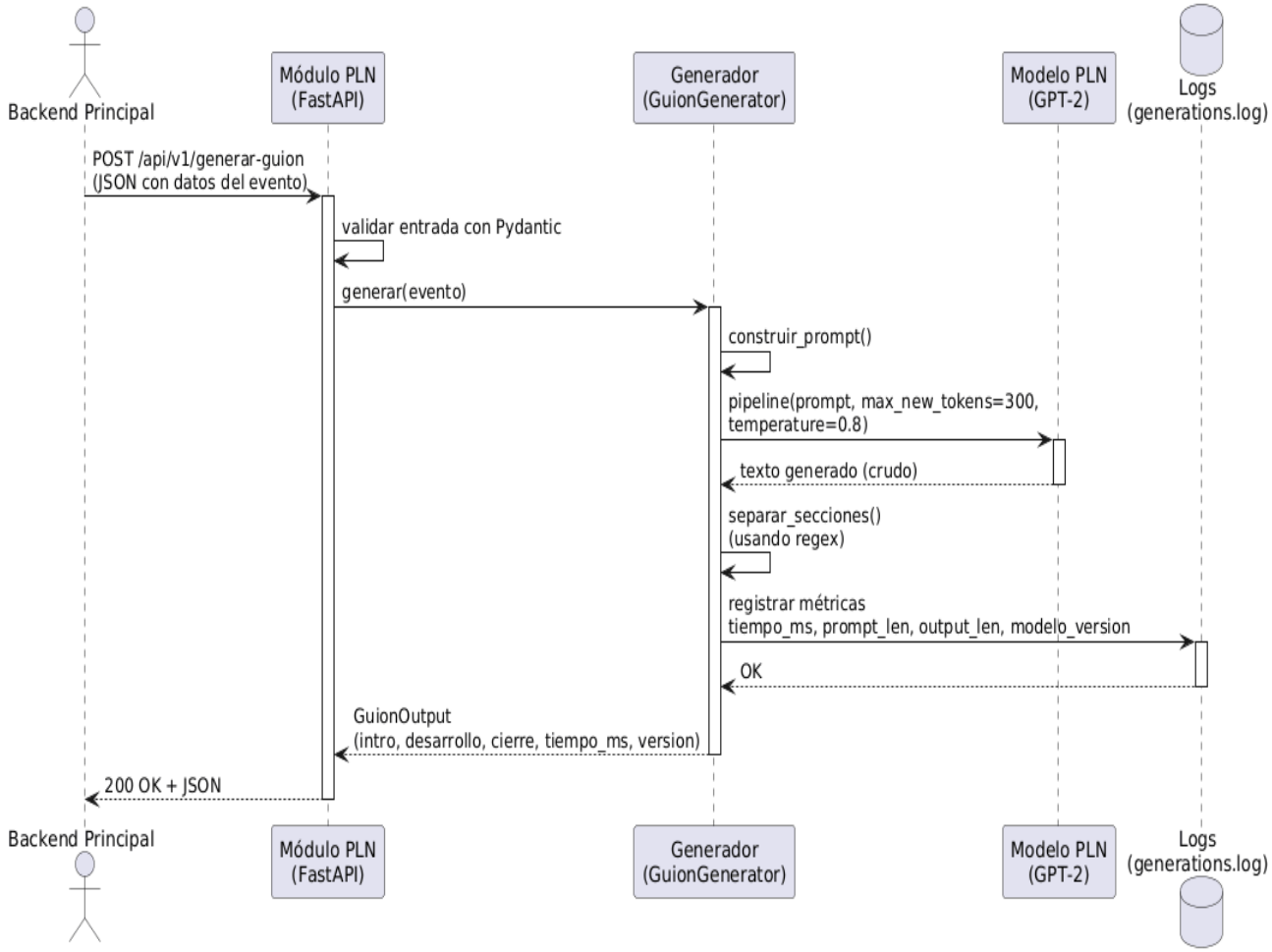


Ilustración 4: Diagrama de secuencia del proceso de generación exitosa de un guion

Descripción: El backend principal envía los datos del evento. El módulo valida, construye el prompt, llama al modelo GPT-2, postprocesa la salida, registra las métricas y finalmente devuelve el guion estructurado.

4.3.6.5 Diagrama de Clases

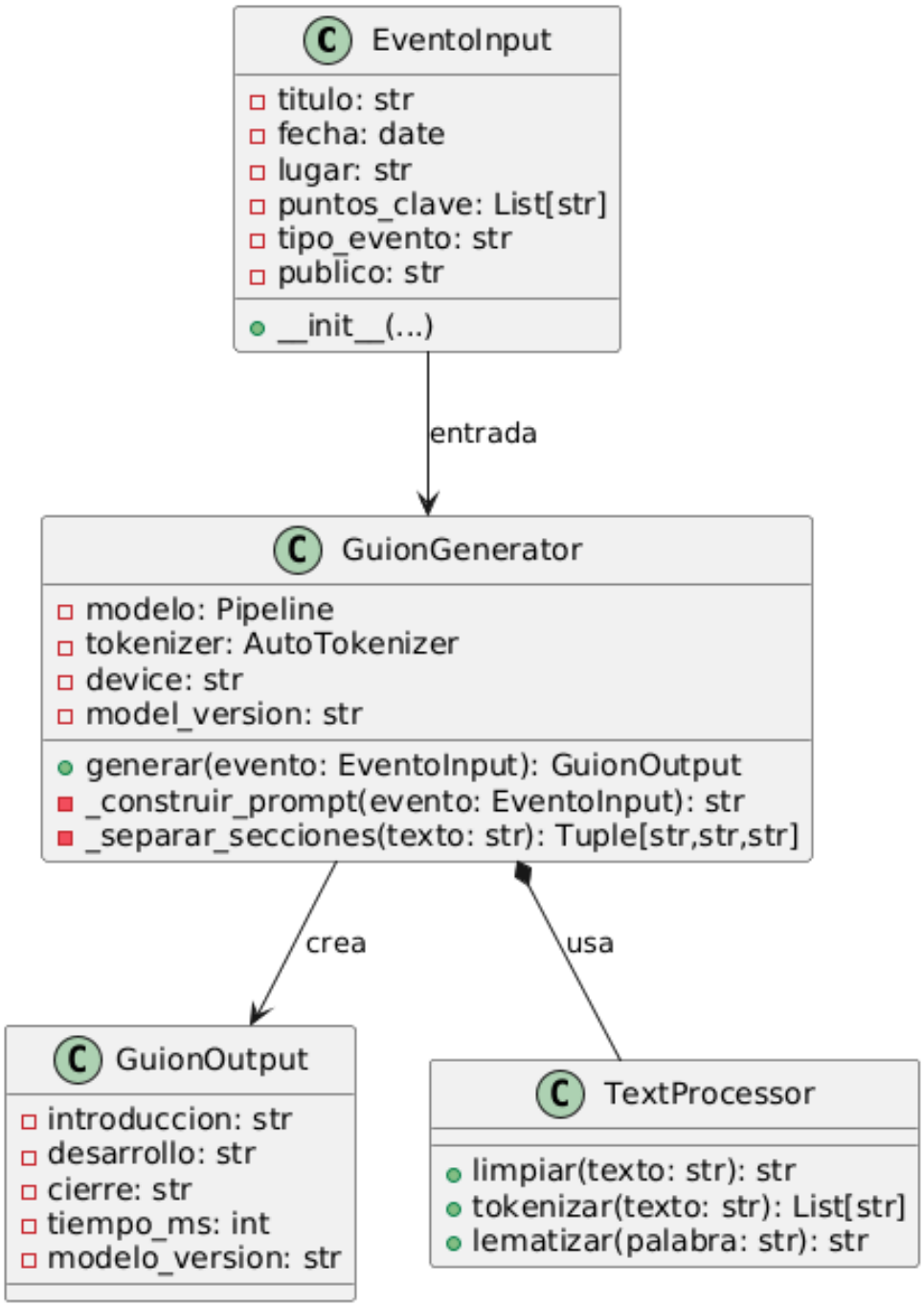


Ilustración 5 Diagrama de clases del modelo de dominio del módulo PLN

Descripción: Generar un GuionOutput a partir de un EventoInput mediante un TextProcessor auxiliar es la función de la clase GuionGenerator, cuyo núcleo exhibe atributos y métodos con sus tipos definidos.

4.3.6.6 Diagrama de Estado

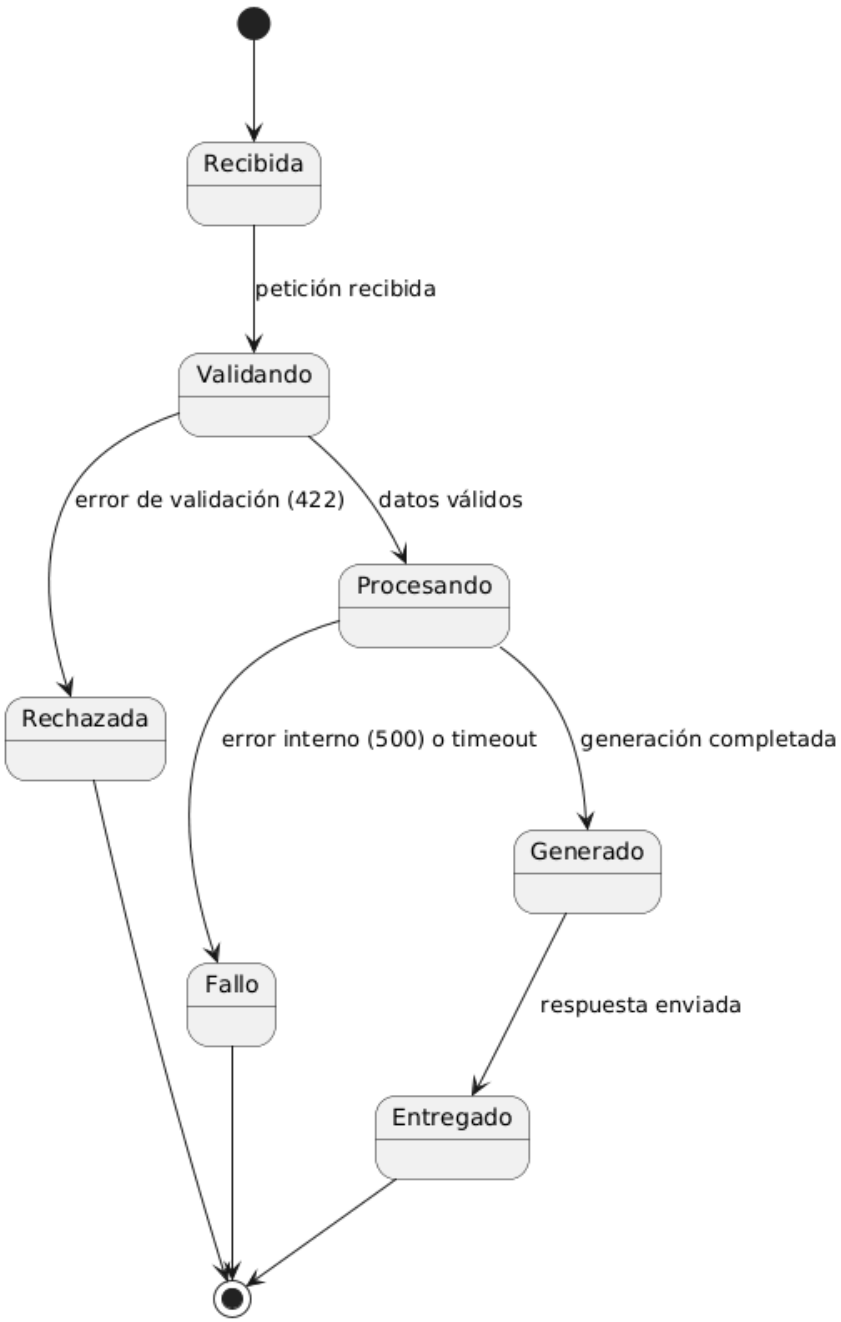


Ilustración 6 Diagrama de estado del ciclo de vida de una petición de generación de guion

Descripción: Rechazada (por error de validación) o Fallo (por error interno) son los estados terminales desde los que se finaliza el flujo, después de que la solicitud haya pasado por Recibida, Validando, Procesando, Generado y Entregado.

4.3.6.7 Diagrama de Actividades

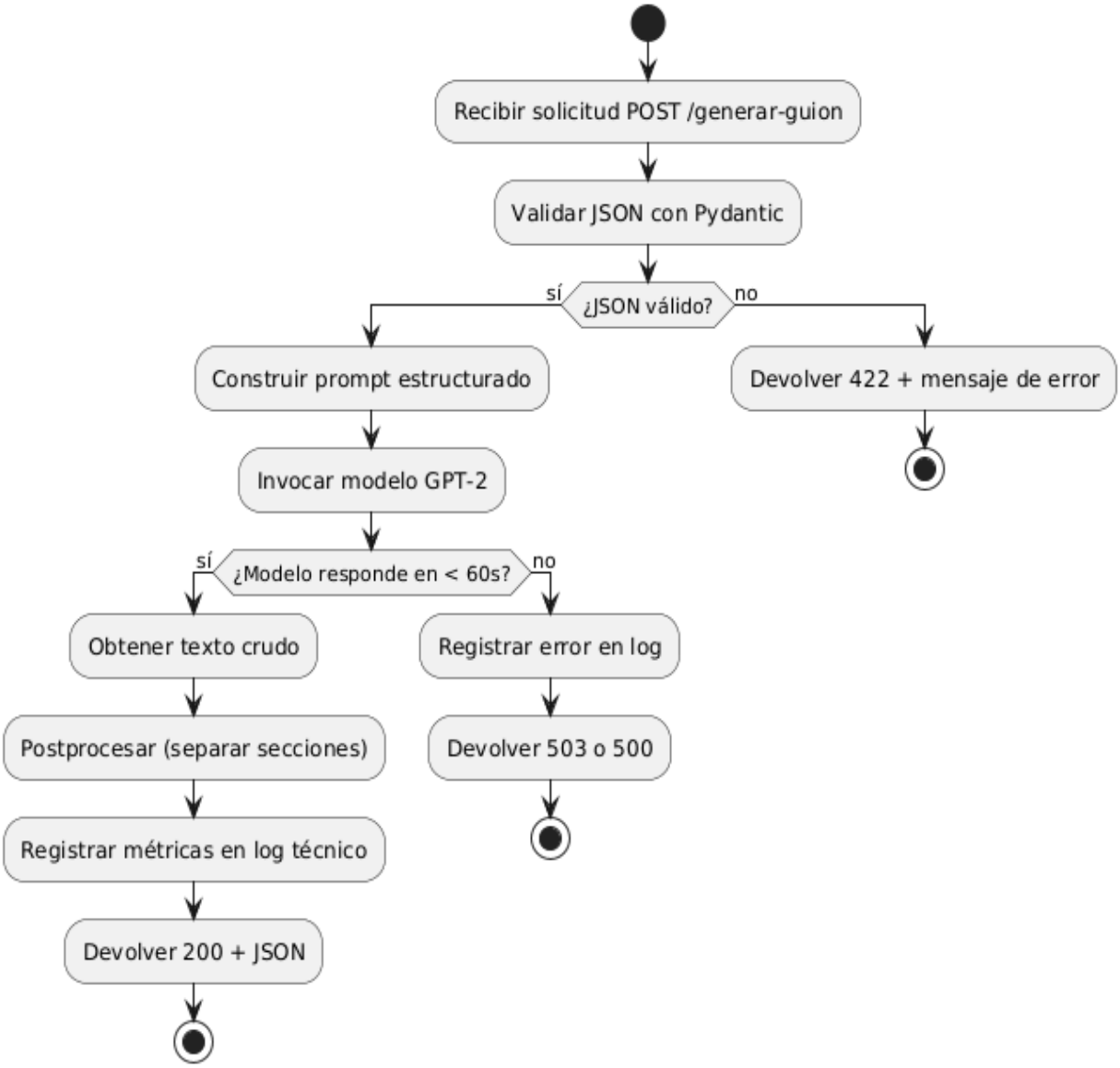


Ilustración 7 Diagrama de actividades del flujo de generación de guiones

Descripción: Cabe señalar que el diagrama ilustra la bifurcación por validación y por disponibilidad del modelo, además de los pasos principales del flujo.

4.3.6.8 Diagrama de Componentes

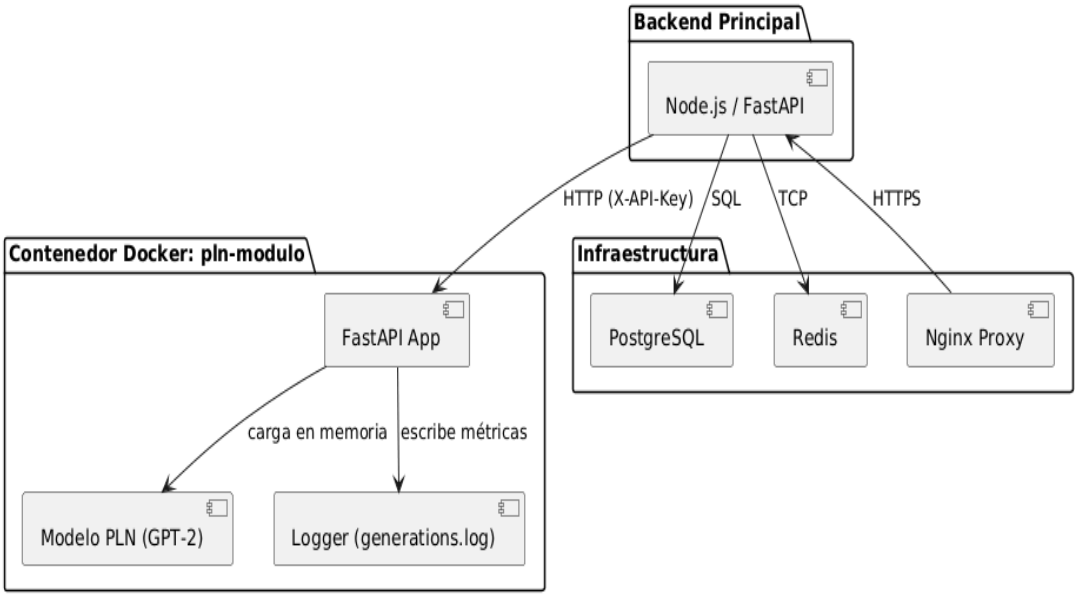


Ilustración 8 Diagrama de componentes del módulo PLN

Descripción: Comunicándose con el backend principal vía HTTP mediante API Key, el módulo PLN opera como un contenedor independiente. El backend, mientras tanto, orquesta el resto de los componentes de la infraestructura.

4.3.6.9 Diagrama de Despliegue Físico

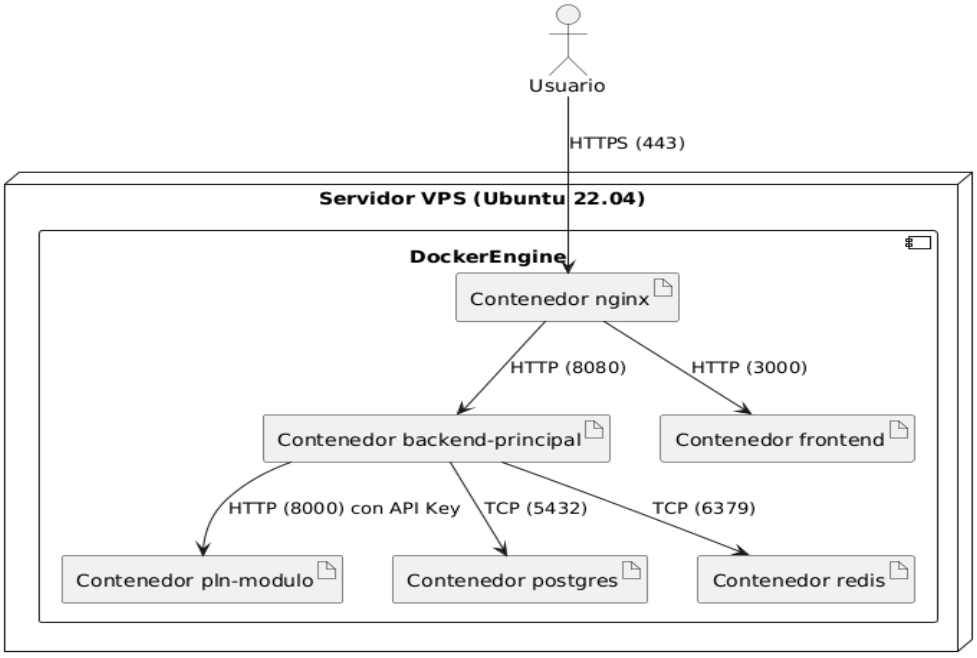


Ilustración 9 Diagrama de despliegue físico del módulo PLN

Descripción: Desplegándose como un contenedor Docker en el mismo VPS que el resto de servicios, el módulo PLN se comunica con el backend principal mediante la red interna de Docker y autenticación por API Key.

El conjunto de diagramas incluido en esta sección —casos de uso, secuencia, clases, estado, actividades, componentes y despliegue— corresponde a la notación del Lenguaje Unificado de Modelado (UML), una herramienta ampliamente extendida en la ingeniería de software para representar de forma gráfica y estandarizada tanto la estructura estática como el comportamiento dinámico de un sistema.

Aunque cada diagrama atienda a un aspecto distinto del sistema, su uso conjunto permite documentar el módulo de PLN de manera integral: el de clases detalla la estructura interna y las relaciones del componente GuionGenerator; el de secuencia, la interacción temporal con el backend durante una petición exitosa; y el de estado, el ciclo de vida completo de la solicitud, desde su llegada hasta su resolución o fallo. Esta visión múltiple, en definitiva, facilita tanto la comprensión del sistema por parte de terceros, por ejemplo, el equipo de backend que debe integrarlo como su mantenimiento posterior por otros desarrolladores.

los diagramas de componentes y de despliegue físico permiten visualizar la separación entre el módulo de PLN y el resto de la infraestructura, evidenciando de manera gráfica el desacoplamiento arquitectónico descrito previamente y facilitando la planificación de la contenerización y el despliegue conjunto de los distintos servicios sobre la infraestructura disponible en la ULEAM Extensión El Carmen. Su elaboración mediante PlantUML, como se indica en la sección de tecnologías, agiliza su mantenimiento y garantiza que puedan versionarse junto con el código fuente del proyecto.

4.3.7 Descripción Técnica / Arquitectura del Sistema

4.3.7.1 Arquitectura del Sistema

El sistema informático desarrollado para la gestión de guiones audiovisuales en el Club SoftMedia sigue una arquitectura de tipo cliente-servidor de tres capas, complementada con un enfoque de microservicios para el componente de Procesamiento de Lenguaje Natural (PLN). Mantenimiento, escalabilidad y evolución independiente son las ventajas que aporta esta decisión arquitectónica al desacoplar el módulo de generación de guiones del resto del sistema.

En la capa del cliente, el frontend construido con React (Vite), TypeScript y TailwindCSS ofrece una interfaz web responsiva y accesible. La comunicación con el backend principal se canaliza mediante HTTP/HTTPS, con intercambio de datos en formato JSON.

En la capa del servidor, se distinguen dos componentes principales:

1. Por más que el módulo de PLN sea el componente estrella, el backend principal (FastAPI, Python) es quien orquesta todo: autenticación con JWT, sesiones, base de datos PostgreSQL y llamadas al módulo. Sin él, el frontend no podría comunicarse con el sistema de generación.
2. Operando como servicio independiente y autocontenido, el módulo de PLN construido con FastAPI expone los endpoints de generación de guiones. En memoria, conserva el modelo GPT-2 fine-tuned y recurre a Transformers y PyTorch para el procesamiento y generación de texto. La comunicación con el backend principal, por su parte, se canaliza mediante HTTP con API Key.

aunque la capa de datos pueda parecer un detalle secundario, PostgreSQL es el sistema encargado de almacenar usuarios, guiones generados con su texto completo, eventos y logs de procesamiento.

La comunicación entre el backend principal y la base de datos se realiza mediante SQLAlchemy como ORM (Object-Relational Mapping), lo que facilita la manipulación de datos y garantiza la integridad referencial y las transacciones ACID.

La comunicación entre todas las capas se realiza mediante HTTP/HTTPS con intercambio de datos en formato JSON. Se implementan interceptores automáticos para la renovación de tokens JWT, la validación de seguridad y la gestión de errores, todo lo cual

traduce en una aplicación escalable, segura y con muy buena experiencia de usuario, tanto en línea como en modo offline (mediante caché local para datos estáticos).

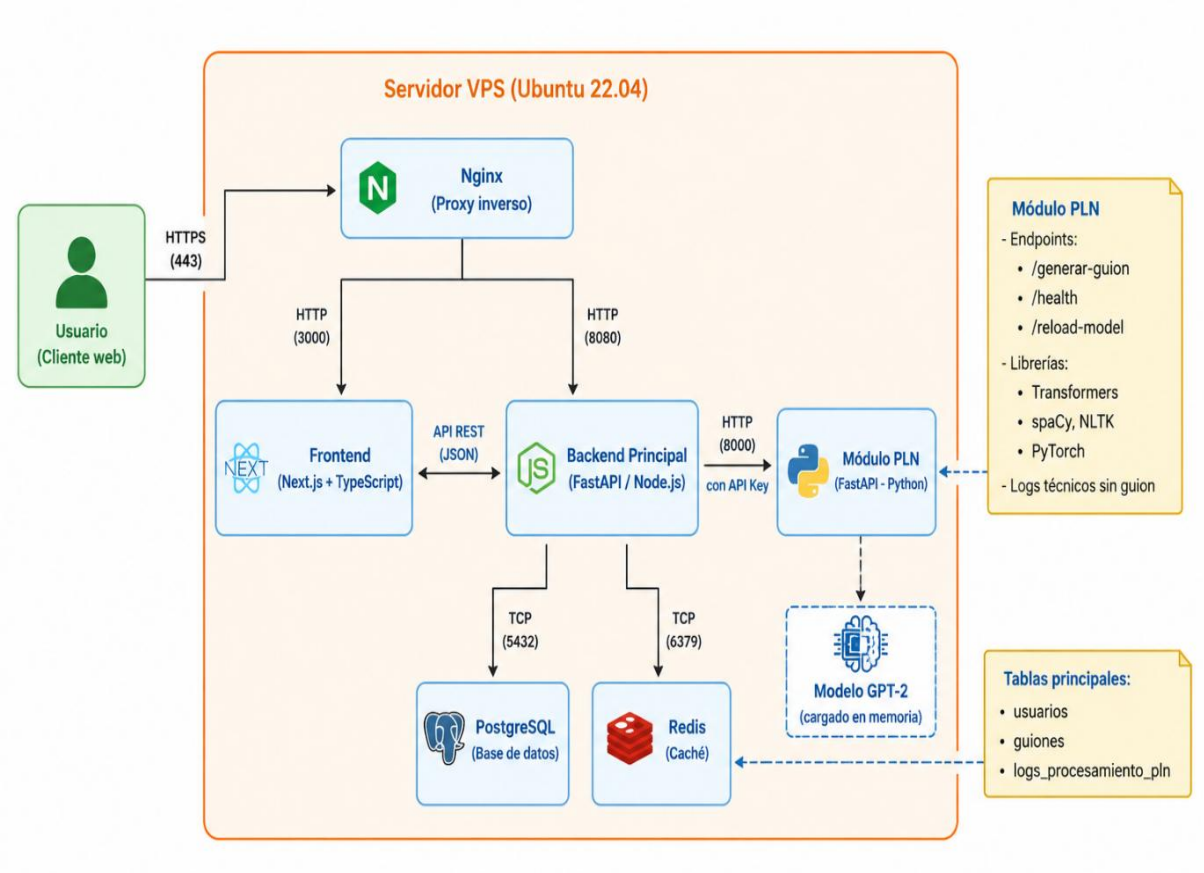


Ilustración 10 del sistema de gestión de guiones audiovisuales

4.3.7.1.1 Descripción de la arquitectura:

- El usuario ingresa al sistema mediante un navegador web y se conecta al servidor a través de HTTPS.
- Actuando como proxy inverso, Nginx balancea la carga, maneja las conexiones SSL y canaliza las peticiones hacia el frontend o el backend, según el caso.
- Sirviendo la interfaz de usuario, el frontend (React (Vite)) se comunica con el backend principal a través de llamadas a la API REST.
- El backend principal es quien se encarga de toda la lógica de negocio: autentica al usuario, valida permisos, almacena datos en PostgreSQL y, cuando se requiere generar un guion, se comunica con el módulo PLN.
- Por más que el módulo PLN maneje todo el flujo de generación —recibir datos, procesar, generar con GPT-2 y devolver JSON—, no tiene responsabilidad sobre la persistencia. Esa tarea recae exclusivamente en el backend principal.
- Almacenando de forma persistente toda la información del sistema, PostgreSQL cumple su rol como base de datos.

4.3.7.2 Mapa del Sistema

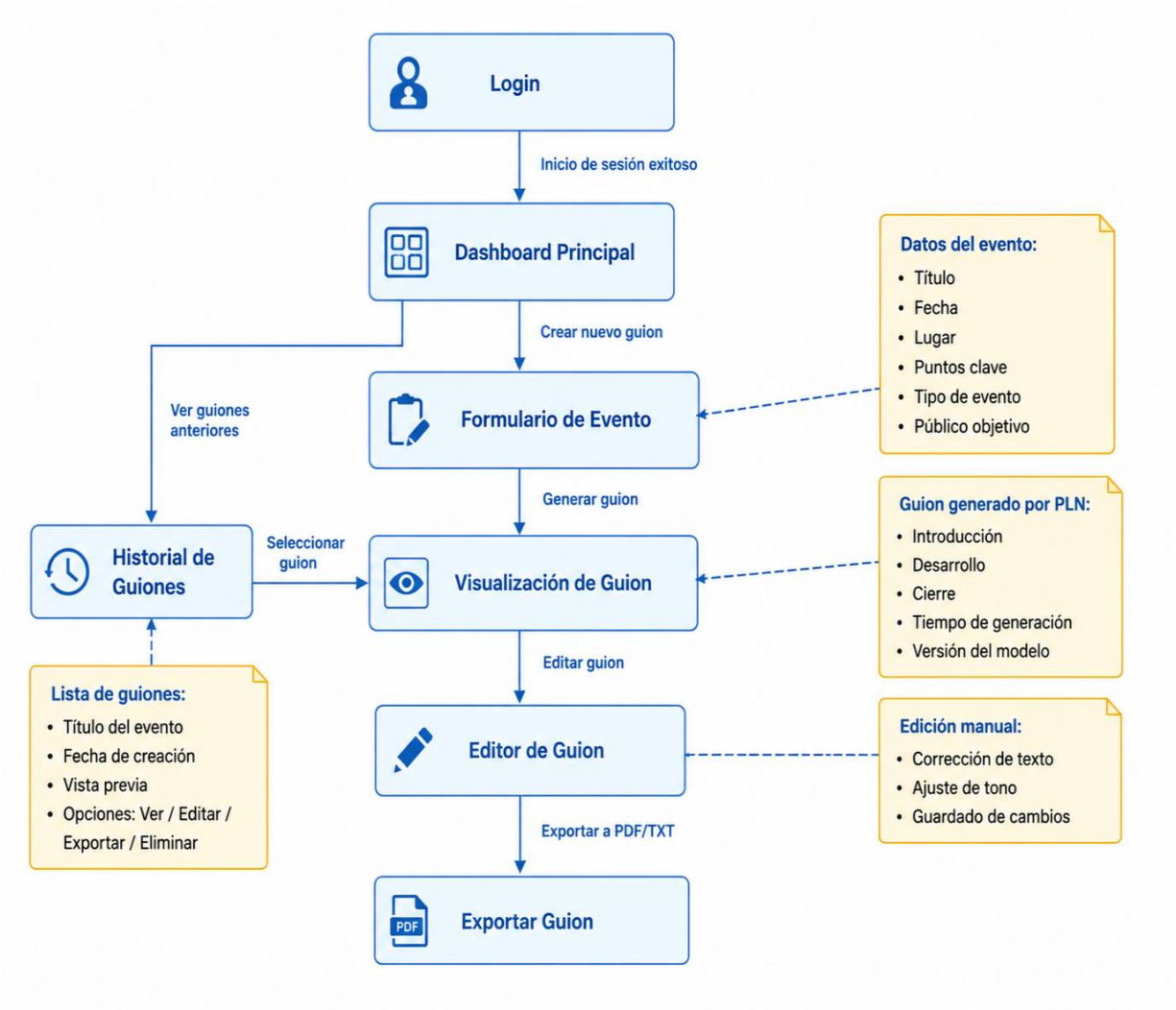


Ilustración 11 Mapa de navegación de las pantallas del sistema

4.3.7.2.1 Descripcion del Mapa del Sistema

- la pantalla de inicio de sesión permite al usuario autenticarse mediante correo y contraseña. Una vez validadas las credenciales, el sistema redirige al Dashboard..
- El Dashboard Principal funciona como pantalla de inicio y concentra las opciones esenciales: "Crear nuevo guion", "Ver historial" y el acceso al perfil de usuario.
- Tras completar los campos del formulario de evento título, fecha, lugar, puntos clave, tipo de evento y público objetivo, el usuario hace clic en "Generar guion".

En ese momento, la petición viaja al backend principal, que la deriva al módulo PLN para su procesamiento.

- La pantalla de visualización del guion presenta el texto generado dividido en tres secciones: introducción, desarrollo y cierre, acompañado del tiempo de generación y la versión del modelo. Desde allí, el usuario puede editar, guardar o exportar el guion.
- El Editor de Guion es la pantalla donde el usuario puede modificar el texto de forma manual mediante un editor enriquecido. Los cambios realizados se almacenan en la base de datos a través del backend principal.
- El Historial de Guiones muestra una lista paginada con todos los guiones que el usuario ha generado. Cada entrada incluye la información del evento asociado, la fecha de creación y un conjunto de acciones disponibles: ver, editar, exportar o eliminar el guion correspondiente.
- La funcionalidad de exportación permite generar un documento descargable en formato PDF o TXT a partir del guion producido.

4.3.7.3 Tecnologías Utilizadas

Tabla 14 Tecnologías, versiones y propósito empleados en el desarrollo del módulo PLN

Tecnología	Versión	Propósito	Uso en el módulo
Python	3.11	Lenguaje de programación principal	Desarrollo del backend y del modelo de PLN
FastAPI	0.109.0	Framework para API REST	Exposición de endpoints (/generar-guion, /health, /reload-model)
Uvicorn	0.27.0	Servidor ASGI	Ejecución del servidor FastAPI en producción
Pydantic	2.6.1	Validación de esquemas de datos	Validación de entrada (EventoInput) y salida (GuionOutput)
Transformers (Hugging Face)	4.38.2	Modelos de lenguaje preentrenados	Carga y ejecución del modelo GPT-2 en español

PyTorch	2.2.0	Backend de deep learning	Inferencia del modelo de lenguaje (GPU/CPU)
spaCy	3.7.2	Procesamiento de lenguaje natural	No se utiliza en la implementación real documentada (ver informe de correcciones)
NLTK	3.8.1	Procesamiento de texto	No se utiliza en la implementación real documentada (ver informe de correcciones)
SQLAlchemy	2.0.25	ORM para base de datos	Conexión y manejo de la base de datos PostgreSQL (backend principal)
PostgreSQL	15.0	Base de datos relacional	Almacenamiento de usuarios, guiones, eventos y logs
Docker	24.0.7	Contenerización	Empaquetado del módulo y sus dependencias
Docker Compose	2.24.0	Orquestación de contenedores	Despliegue conjunto del módulo, backend, base de datos y servicios auxiliares
Git	2.40	Control de versiones	Gestión del código fuente y colaboración
GitHub	-	Repositorio remoto	Almacenamiento y respaldo del código
Google Colab	-	Entorno de notebooks en la nube	Entrenamiento (fine-tuning) del modelo GPT-2 con GPU gratuita

Jupyter Notebook	6.5.4	Entorno interactivo	Experimentación y prototipado del modelo PLN
Postman	10.0	Pruebas de APIs	Validación de endpoints y documentación de la API
Locust	2.20.1	Pruebas de carga	Simulación de múltiples usuarios concurrentes para pruebas de rendimiento
Pytest	8.0.0	Framework de pruebas unitarias	Pruebas automatizadas de los métodos de la clase GuionGenerator
PlantUML	-	Herramienta de diagramación	Generación de diagramas UML (casos de uso, secuencia, clases, etc.)
Figma	-	Diseño de interfaces UI/UX	Prototipado de las pantallas del sistema (frontend)
React	14.0	Framework de frontend	Interfaz de usuario web (desarrollado por otros miembros)
TypeScript	5.0	Superset de JavaScript	Tipado estático en el frontend
TailwindCSS	3.4	Framework de estilos CSS	Diseño responsivo y estilizado del frontend
Nginx	1.24	Proxy inverso	Balanceo de carga y manejo de SSL
Redis	7.2	Almacén en memoria	Caché de sesiones y rate limiting (opcional)

Conviene profundizar en el fundamento teórico de algunas de las tecnologías centrales listadas en la tabla anterior, en particular aquellas vinculadas directamente al procesamiento de lenguaje natural. El modelo GPT-2 empleado en su versión en español corresponde a una arquitectura de tipo transformer, caracterizada por mecanismos de atención que permiten capturar dependencias a larga distancia dentro del texto, mejorando sustancialmente la coherencia de los textos generados frente a arquitecturas neuronales anteriores basadas en redes recurrentes (Cortez Vásquez, 2020).

El proceso de ajuste fino (fine-tuning) aplicado sobre este modelo preentrenado, a partir de 120 ejemplos de guiones reales del Club SoftMedia, se apoya en la premisa de que partir de un modelo con conocimiento lingüístico general del español y especializarlo posteriormente con un corpus reducido resulta más eficiente, en términos de tiempo y de recursos computacionales, que entrenar un modelo desde cero, lo cual es especialmente relevante en un proyecto académico con recursos de cómputo limitados. (Universidad de Minnesota, 2023)

Aunque las librerías de apoyo podrían haber sido otras, la elección de Transformers para cargar y ejecutar el modelo GPT-2 afinado, y de PyTorch en su variante CPU como motor de inferencia, responde a un criterio práctico: ambas cuentan con documentación abundante y comunidades activas, lo que acorta la curva de aprendizaje en un proyecto con plazos acotados como el presente trabajo de titulación.

En cuanto al framework FastAPI, su elección para exponer tanto el backend principal como el módulo de PLN responde a su soporte nativo para validación de datos mediante Pydantic, su generación automática de documentación interactiva (Swagger/OpenAPI) y su alto rendimiento en comparación con otros frameworks de Python, características que resultan valiosas cuando se requiere iterar rápidamente sobre los contratos de API entre los distintos componentes del sistema desarrollados por diferentes integrantes del equipo de SoftMedia (Celi-Párraga et al., 2023).

4.3.8 Requerimientos Funcionales

4.3.8.1 Requerimientos Funcionales – Generación de Guiones

Tabla 15 Requerimientos funcionales para la generación de guiones

ID	Requerimiento	Prioridad	Fuente
RF-01	El módulo debe exponer un endpoint POST /generar-guion que acepte un JSON con los campos: titulo, fecha, lugar, puntos_clave (array), tipo_evento, publico.	Alta	HU-PLN-01
RF-02	El módulo tiene la obligación de verificar que todos los campos requeridos estén presentes y con el tipo de dato adecuado. En caso de incumplimiento, debe retornar un error 422 acompañado de mensajes descriptivos que faciliten la corrección.	Alta	HU-PLN-02
RF-03	El módulo debe producir un guion en español estructurado en tres partes: introducción, desarrollo y cierre.	Alta	HU-PLN-01
RF-04	La respuesta debe incluir el tiempo de generación en milisegundos (tiempo_ms) y la versión del modelo utilizado (modelo_version) como parte de los metadatos.	Alta	HU-PLN-01
RF-05	El módulo debe implementar un control de repetición repetition penalty, para evitar que el guion generado caiga en frases redundantes.	Alta	HU-PLN-01

4.3.8.2 Requerimientos Funcionales – Logs y monitoreo

Tabla 16 Requerimientos funcionales de registro de logs y monitoreo

ID	Requerimiento	Prioridad	Fuente
RF-06	El módulo debe registrar en un archivo de log técnico las métricas correspondientes a cada generación exitosa: timestamp, tiempo en	Media	HU-PLN-03

	milisegundos, longitud del prompt, longitud de la salida y versión del modelo empleado.		
RF-07	El módulo debe abstenerse de registrar el contenido del guion: ni introducción, ni desarrollo, ni cierre en el log técnico, con el fin de preservar la privacidad de los datos.	Alta	HU-PLN-03
RF-08	El archivo de log debe contar con rotación automática ya sea por tamaño o por tiempo para impedir que su crecimiento sea indefinido.	Baja	HU-PLN-03

4.3.8.3 Requerimientos Funcionales – Endpoints auxiliares

Tabla 17 Requerimientos funcionales de los endpoints auxiliares /health y /reload-model

ID	Requerimiento	Prioridad	Fuente
RF-09	El módulo debe exponer un endpoint GET /health que, sin requerir autenticación, retorne un JSON con {"status": "ok", "modelo_local_cargado": false}.	Media	HU-PLN-05
RF-10	El módulo debe exponer un endpoint POST /reload-model, protegido mediante API Key, que permita recargar el modelo en caliente sin necesidad de detener el servicio.	Baja	HU-PLN-06

4.3.8.4 Requerimientos Funcionales – Seguridad

Tabla 18 Requerimientos funcionales de seguridad del módulo PLN

ID	Requerimiento	Prioridad	Fuente
RF-11	Los endpoints de escritura (POST /generar-guion, PUT /escenas/{id}, POST /guiones/{id}/escenas, DELETE /escenas/{id}, PUT /guiones/{id}/escenas/reordenar y POST /guiones/{id}/evaluar) deben requerir la cabecera	Alta	Seguridad

	X-API-Key para autenticar las peticiones. Los endpoints de lectura (GET /guiones, GET /guiones/{id}, GET /guiones/{id}/evaluaciones, GET /eventos/{id} y GET /health) permanecen abiertos, para permitir la validación Soft Foreign Key desde el módulo de Marketing.		
RF-12	La API Key debe ser configurable mediante variable de entorno y no debe estar en el código fuente.	Alta	Seguridad

4.3.9 Requerimientos No Funcionales

4.3.9.1 Requerimientos No Funcionales – Rendimiento y disponibilidad

Tabla 19 Requerimientos no funcionales de rendimiento y disponibilidad

ID	Requerimiento	Métrica	Herramienta de medición
RNF-01	El tiempo de generación del guion no debe superar los 40 segundos en el entorno de desarrollo (contenedor Docker local, CPU, sin aceleración por GPU).	Percentil 95 < 40 s	Logs internos
RNF-02	El módulo debe garantizar una disponibilidad del 99.5% durante pruebas continuas de 72 horas.	(Tiempo activo / total) * 100	UptimeRobot / monitoreo
RNF-03	el módulo debe ser capaz de soportar al menos 5 peticiones en paralelo sin que el tiempo promedio de respuesta se incremente en más de un 50 %.	Prueba de carga con 5 usuarios	Locust
RNF-04	En condiciones normales de uso, el contenedor no debe consumir más de 4 GB de RAM.	Valor máximo en docker stats	Docker stats

4.3.9.2 Requerimientos No Funcionales – Usabilidad y mantenibilidad

Tabla 20 Requerimientos no funcionales de usabilidad y mantenibilidad

ID	Requerimiento	Métrica	Herramienta de medición
RNF-05	La coherencia narrativa de los guiones generados debe ser evaluada positivamente por al menos el 80% de los usuarios del club.	Encuesta Likert (puntuación ≥ 4/5)	Google Forms
RNF-06	el código fuente debe estar documentado mediante docstrings en español y alojado en un repositorio Git.	Revisión manual	Inspección de código
RNF-07	El módulo debe poder desplegarse con un único comando de Docker Compose.	Contenedor levantado sin errores	Prueba de despliegue

4.3.9.3 Requerimientos No Funcionales – Seguridad

Tabla 21 Requerimientos no funcionales de seguridad

ID	Requerimiento	Métrica	Herramienta de medición
RNF-08	La API Key no debe aparecer en los logs ni en mensajes de error.	Inspección de logs	Revisión manual
RNF-09	El módulo debe rechazar peticiones con API Key inválida devolviendo código 401.	Prueba de seguridad	Postman

Por más que ambos tipos de requerimientos sean necesarios, su distinción responde a un principio fundamental de la ingeniería de requisitos: los funcionales (RF) especifican qué debe hacer el sistema y qué comportamientos puede observar el usuario, mientras que los no funcionales (RNF) definen las condiciones de calidad —rendimiento, disponibilidad, usabilidad, mantenibilidad, seguridad, entre otras— que deben cumplirse para que esas funciones sean aceptables..

Aun cuando la separación entre RF y RNF pudiera parecer un detalle de clasificación, tiene consecuencias prácticas en la forma de verificar el cumplimiento de cada tipo. Los requerimientos funcionales suelen validarse con pruebas de caja negra que comparan entradas y salidas esperadas, tal como se explica en la sección de Procesos de Prueba; los no funcionales, en cambio, demandan herramientas de monitoreo, pruebas de carga y análisis estático de código, reflejados en la columna "Herramienta de medición" de las tablas de RNF incluidas con anterioridad.

Por más que la trazabilidad pueda considerarse un aspecto administrativo, su materialización en la columna "Fuente" —que asocia cada RF a su historia de usuario— es una práctica respaldada en la literatura de ingeniería de software. Con ella se persigue que ningún requerimiento quede huérfano de una necesidad real del proyecto y, además, se facilita el análisis de impacto cuando surgen cambios en etapas posteriores del desarrollo.

En el caso del módulo de PLN, los requerimientos no funcionales de rendimiento (RNF-01 a RNF-04) resultaron especialmente determinantes, puesto que la condición de éxito del proyecto —establecida desde el Capítulo I— dependía directamente de que el tiempo de generación del guion no superara el umbral fijado. Esta exigencia llevó a que gran parte del trabajo en los Sprints 3 y 4 se volcara en el ajuste de parámetros del modelo y en la optimización del entorno de despliegue.

4.3.10 Roles y Responsabilidades

Tabla 22 Roles, responsabilidades y dedicación del equipo bajo la metodología Scrum

Rol	Nombre	Responsabilidades	Dedicación
Product Owner	Coordinador del Club SoftMedia	Priorizar el Product Backlog, definir historias de usuario, validar los incrementos del módulo, asegurar que el producto cumpla con las necesidades del club.	2 horas/semana
Scrum Master	Ing. Jefferson Arca (Tutor)	Facilitar el proceso Scrum, eliminar impedimentos, guiar al equipo en la aplicación de Scrum, asegurar que	1 hora/semana

		se realicen las reuniones y revisiones.	
Development Team	Jerson Valencia (Autor)	Desarrollando el módulo (back end, modelo PLN, API y contenerización), realizando pruebas unitarias y de integración, y documentando el código y la API, se cubren las actividades principales del proyecto.	100% (3 meses)
Equipo de pruebas piloto	1miembros del Club SoftMedia	Probando el módulo en eventos reales, evaluando la calidad de los guiones generados y canalizando la retroalimentación hacia el equipo, se completa el ciclo de validación y mejora.	1 hora/semana durante 2 semanas
Equipo de backend (integración)	2 compañeros (desarrolladores del backend principal)	Coordinar el contrato de la API, ejecutar pruebas de integración y conectar el endpoint del módulo PLN al sistema principal: esas son las tareas que completan la integración.	5 horas en total

Los roles descritos en la tabla anterior corresponden a la estructura organizativa propia de Scrum, en la que cada rol cumple una función claramente delimitada y complementaria dentro del equipo. El Product Owner es responsable de maximizar el valor del producto resultante del trabajo del equipo de desarrollo, principalmente a través de la gestión y priorización del Product Backlog; en el presente proyecto, dicho rol recayó en el coordinador del Club SoftMedia, quien mejor conoce las necesidades reales de los usuarios finales del módulo (Alaimo, 2021).

El Scrum Master, por su parte, actúa como facilitador del proceso, promoviendo la adopción correcta del marco de trabajo y removiendo los obstáculos que puedan afectar el avance del equipo de desarrollo, sin ejercer autoridad jerárquica sobre las decisiones técnicas del proyecto (Canosa Ferreiro, 2024c). En este trabajo de titulación, dicha función fue asumida

por el tutor académico, quien además aportó una supervisión metodológica adicional propia del proceso de acompañamiento de un trabajo de titulación.

Finalmente, el Development Team es el responsable de convertir los elementos priorizados del Product Backlog en incrementos funcionales del producto durante cada Sprint. Al tratarse de un trabajo de titulación individual, el rol de Development Team fue asumido íntegramente por el autor, lo cual implicó adaptar algunas dinámicas propias de equipos Scrum de mayor tamaño, como las reuniones diarias, a un formato de auto-organización personal, sin que ello afectara la aplicación de los principios centrales del marco de trabajo: entrega incremental, inspección y adaptación continua (Canosa Ferreiro, 2024c). Cabe destacar además la participación del equipo de pruebas piloto y del equipo de backend de integración, cuya función, aunque no corresponde estrictamente a un rol Scrum formal, resultó indispensable para validar los incrementos generados en cada Sprint desde la perspectiva de los usuarios finales del sistema.

4.3.11 Planificación del Sprint

4.3.11.1 Sprint 1: Fundamentos del sistema y configuración inicial

Tabla 23 Planificación del Sprint 1: fundamentos del sistema y configuración inicial

SPRINT 1: Fundamentos del sistema	
Duración:	2 semanas (15/06/2026 – 28/06/2026)
Prioridad:	Alta
Objetivo:	Establecer las tareas iniciales comprenden establecer la infraestructura base del módulo: configurar el entorno de desarrollo, definir la estructura de directorios, preparar el contenedor Docker y habilitar el endpoint de salud.
Historias de usuario:	HU-PLN-05 (Consultar salud)
Tareas de desarrollo:	• Configuración del entorno Python (virtualenv, instalación de dependencias: FastAPI, Uvicorn, Pydantic). • Creación de la estructura de directorios (pln_module, tests, logs). • Implementación del endpoint /health con FastAPI.

	• Creación del Dockerfile y docker-compose.yml básico. • Pruebas de conectividad y despliegue local.
Plan de entrega:	• Configuración del ambiente: Instalación de Python 3.10+, FastAPI, Uvicorn, Pydantic. (4 horas). • Estructura del proyecto: Creación de carpetas y archivos iniciales. (2 horas). • Endpoint /health: Código de FastAPI con ruta GET /health. (3 horas). • Contenerización: Dockerfile con instrucciones básicas (FROM python, COPY requirements, RUN pip, CMD). (4 horas). • Pruebas y validación: Verificar que el contenedor se levante y el endpoint responda {"status":"ok","modelo_local_cargado":false}.(3 horas).

4.3.11.2 Sprint 2: Generación básica de guion y validación

Tabla 24 Planificación del Sprint 2: generación básica de guion y validación

SPRINT 2: Generación básica	
Duración:	2 semanas (29/06/2026 – 12/07/2026)
Prioridad:	Alta
Objetivo:	Implementando la lógica de generación con GPT-2, el módulo incorpora la validación de entrada y expone el endpoint POST /generar-guion como su punto de acceso principal.
Historias de usuario:	HU-PLN-01 (Generación exitosa), HU-PLN-02 (Manejo de errores)
Tareas de desarrollo:	• Integrar el modelo GPT-2 en español con la librería Transformers. • Crear la clase GuionGenerator con métodos _construir_prompt y _separar_secciones. • Definir los esquemas Pydantic (EventoInput, GuionOutput).

	• Implementar el endpoint POST /generar-guion con autenticación API Key. • Manejar errores de validación con códigos 422. • - Escribir pruebas unitarias básicas con pytest.
Plan de entrega:	• Modelo y dependencias: Instalación de torch, transformers, spacy, nltk. (2 horas). • Clase Generador: Implementación completa de generator.py (alrededor de 150 líneas). (10 horas). • Esquemas Pydantic: Definición en schemas.py. (2 horas). • Endpoint POST: Configuración de ruta, lógica de llamada al generador, manejo de excepciones. (4 horas). • Pruebas unitarias: Cobertura > 70% (4 horas).

4.3.11.3 Sprint 3: Logs, métricas y ajustes de calidad

Tabla 25 Planificación del Sprint 3: logs, métricas y ajustes de calidad

SPRINT 3: Logs y calidad	
Duración:	2 semanas (13/07/2026 – 26/07/2026)
Prioridad:	Media
Objetivo:	Las tareas incluyen incorporar el registro de logs técnicos, ajustar los parámetros de generación —temperatura y repetición— y realizar pruebas de aceptación con los miembros del club.
Historias de usuario:	HU-PLN-03 (Registro de métricas), HU-PLN-04 (Historial – coordinación con backend)
Tareas de desarrollo:	• Implementar RotatingFileHandler para logs. • Registrar métricas de cada generación (tiempo, longitudes, versión). • Ajustar hiperparámetros del modelo (temperature, repetition_penalty, top_p). • Preparar encuesta de evaluación para los usuarios. • Coordinar con el equipo de backend la integración del historial (contrato de API).

Plan de entrega:	• Logging: Configuración en generator.py con handler rotativo. (3 horas). • Ajuste de parámetros: Pruebas con diferentes valores (temperature 0.7, 0.8, 0.9). (4 horas). • Coordinación: Reunión con backend para definir contrato de API. (2 horas). • Encuesta: Diseño de formulario en Google Forms. (2 horas).

4.3.11.4 Sprint 4: Despliegue, integración y documentación final

Tabla 26 Planificación del Sprint 4: despliegue, integración y documentación final

SPRINT 4: Despliegue e integración	
Duración:	2 semanas (27/07/2026 – 09/08/2026)
Prioridad:	Alta
Objetivo:	Desplegar el módulo en el entorno de producción (VPS), integrarlo con el backend principal, implementar la recarga de modelo y completar la documentación técnica final.
Historias de usuario:	HU-PLN-06 (Recargar modelo), HU-PLN-04 (Historial – soporte)
Tareas de desarrollo:	• Crear script de despliegue con Docker Compose. • Implementar endpoint /reload-model. • Realizar pruebas de carga con Locust. • Redactar la guía de integración para el backend. • Documentar la API con Swagger/OpenAPI. • Realizar la prueba piloto con los 10 miembros del club.
Plan de entrega:	• Despliegue: Configurar VPS, instalar Docker, levantar contenedores. (6 horas). • Endpoint reload: Implementar y probar. (3 horas).

	• Pruebas de carga: Ejecutar Locust con 5 usuarios concurrentes. (4 horas). • Documentación: Redactar manual de API y guía de instalación. (6 horas). • Prueba piloto: Sesión con los miembros del club para evaluar guiones generados. (3 horas).

4.3.12 Backlog del Producto

El Product Backlog inicial se priorizó junto con el Product Owner (coordinador de SoftMedia) y se muestra en la siguiente tabla, que incluye tareas de todos los sprints.

Tabla 27 Backlog del producto priorizado por sprint

Nro.	Tarea	Prioridad	Estimación (horas)	Sprint asociado
1	Configuración de entorno Python y FastAPI	Alta	4	Sprint 1
2	Creación de estructura de directorios	Alta	2	Sprint 1
3	Implementación de endpoint /health	Alta	3	Sprint 1
4	Integración del modelo GPT-2	Alta	6	Sprint 2
5	Clase GuionGenerator (prompt, generación)	Alta	10	Sprint 2
6	Esquemas Pydantic (validación)	Alta	2	Sprint 2
7	Endpoint POST /generar-guion	Alta	4	Sprint 2
8	Manejo de errores y códigos HTTP	Alta	3	Sprint 2
9	Implementación de logging con rotación	Media	3	Sprint 3

10	Ajuste de hiperparámetros del modelo	Media	4	Sprint 3
11	Coordinación con backend para historial	Media	2	Sprint 3
12	Endpoint /reload-model	Baja	3	Sprint 4
13	Pruebas unitarias (pytest)	Alta	4	Sprint 2
14	Pruebas de carga (Locust)	Media	3	Sprint 4
15	Contenerización con Docker	Alta	4	Sprint 1
16	Despliegue en VPS	Alta	6	Sprint 4
17	Documentación de la API (OpenAPI)	Media	4	Sprint 4
18	Prueba piloto con usuarios	Alta	3	Sprint 4

Considerando la definición de Scrum una lista ordenada y dinámica de lo necesario para el producto, que evoluciona con el producto y su entorno (Alaimo, 2021), el Product Backlog presentado en la tabla anterior se ajusta a esa lógica. Para el módulo de PLN, este incluyó tareas técnicas (configuración, integración del modelo, contenerización) y otras ligadas directamente a los criterios de aceptación de las historias de usuario, como validación de esquemas, manejo de errores y pruebas piloto.

Que la integración del modelo de lenguaje pudiera avanzar en el Sprint 2 era el objetivo que condicionó la ubicación de tareas como la configuración del entorno Python y FastAPI (tarea 1) y la contenerización con Docker (tarea 15) en el Sprint 1. Esa decisión, registrada en la columna "Prioridad" de la tabla, priorizó el riesgo técnico y el carácter habilitante de cada tarea entre las dieciocho que componían el backlog.

Si bien la estimación de horas por tarea es referencial —al tratarse de un desarrollo individual—, permitió al autor planificar con más realismo la carga de trabajo de los Sprints y prever ajustes de alcance. Una práctica recomendable, incluso en proyectos de titulación de naturaleza unipersonal, para no sobrecargar al desarrollador.

4.4 Interfaz de Usuario (UI) / Prototipos

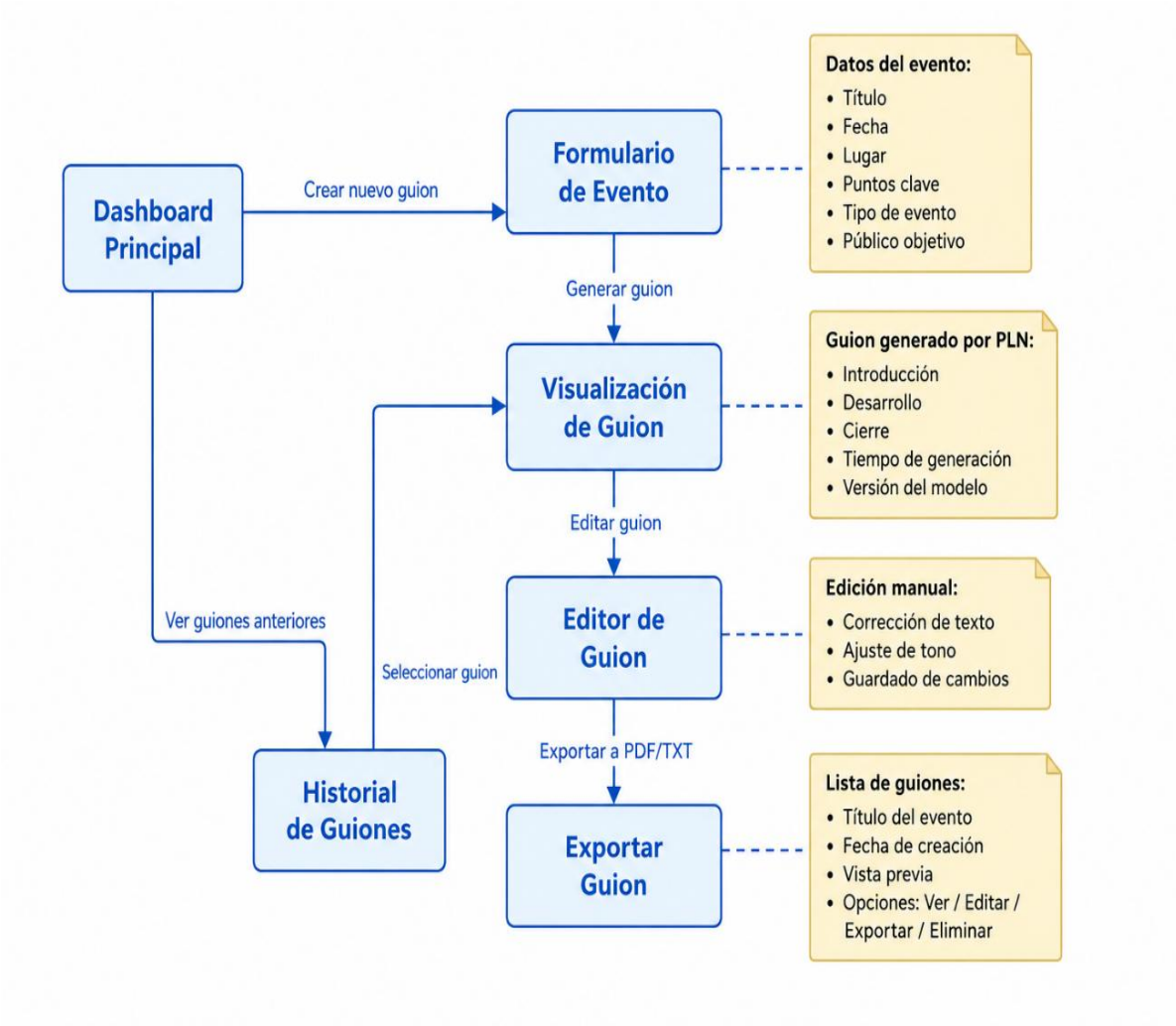


Ilustración 12 Prototipos de interfaz de usuario del módulo de generación de guiones

4.4.1 Pantallas del sistema

4.4.1.1 Pantalla de Formulario de Evento

la pantalla de formulario permite al usuario ingresar los datos del evento los cuales serán utilizados por el modelo PLN para generar el guion y que todos los campos son obligatorios, validándose en tiempo real para garantizar la integridad de la información.

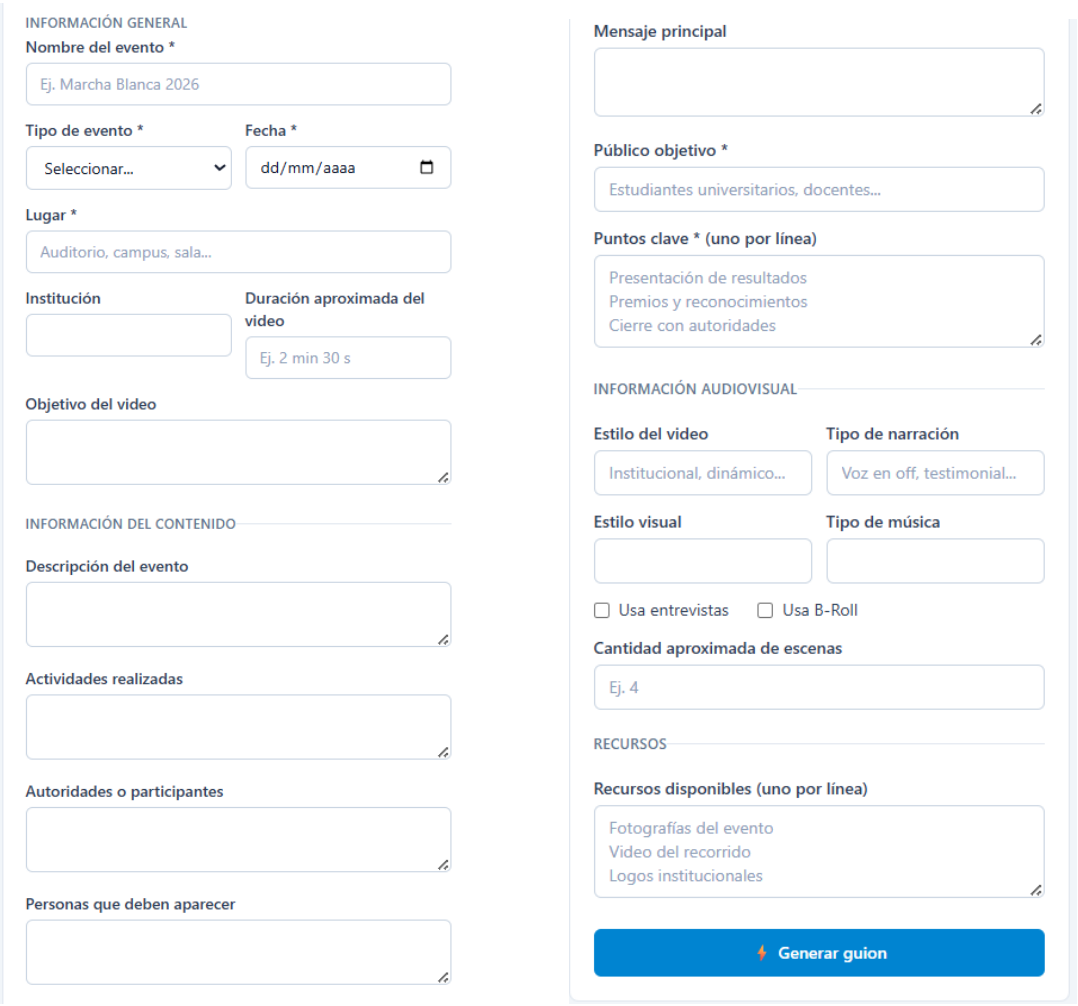


Ilustración 13 sección de ingreso de formulario

A modo de ejemplo, un miembro de SoftMedia completa el formulario para el evento "Feria de Ciencias 2026" y, al hacer clic en "Generar guion", el sistema procesa la solicitud y lo redirige a la pantalla de visualización del resultado.

4.4.1.2 Pantalla de Visualización de Guion

La pantalla de visualización muestra el guion generado por el modelo de PLN, dividido en tres secciones claramente diferenciadas: introducción, desarrollo y cierre. Además, se muestran metadatos como el tiempo de generación y la versión del modelo.

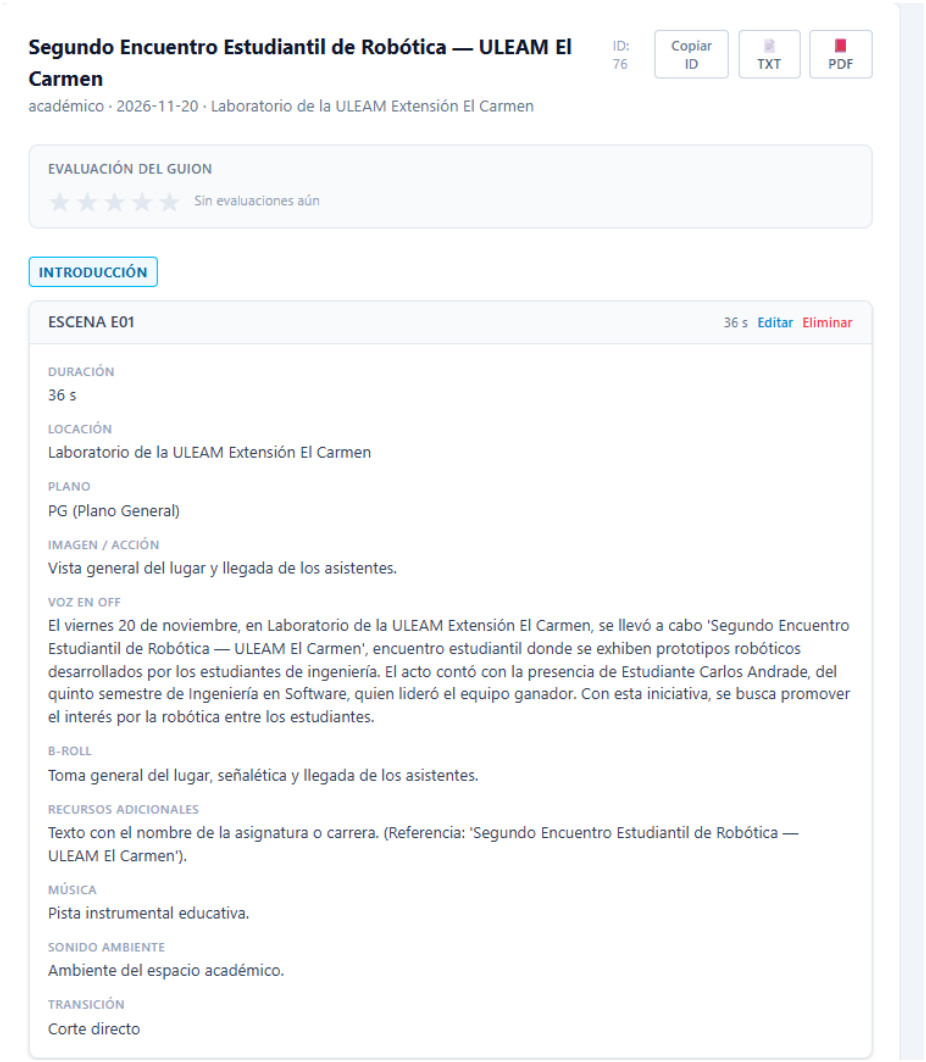


Ilustración 14 sección de visualizador de guion

Ejemplo de usoEl usuario, tras visualizar el guion generado para la "Feria de Ciencias 2026", decide editar la introducción para darle un tono más atractivo. Una vez realizados los cambios, los guarda y exporta el guion en formato PDF, con el fin de compartirlo con el equipo de grabación.

4.4.1.3 Pantalla de Historial de Guiones

La pantalla de historial muestra una lista paginada de todos los guiones generados por el usuario autenticado. Cada elemento de la lista incluye información resumida y opciones de acción.

Ilustración 15 sección historial de guiones

Ejemplo de uso: El usuario busca un guion anterior de un evento similar para reutilizar ideas y lo abre en la pantalla de visualización para editarlo.

4.4.1.4 Pantalla de Editor de Guion

La pantalla del editor permite al usuario modificar manualmente el guion generado por el modelo de PLN. Utiliza un editor de texto enriquecido que permite formatear el contenido.

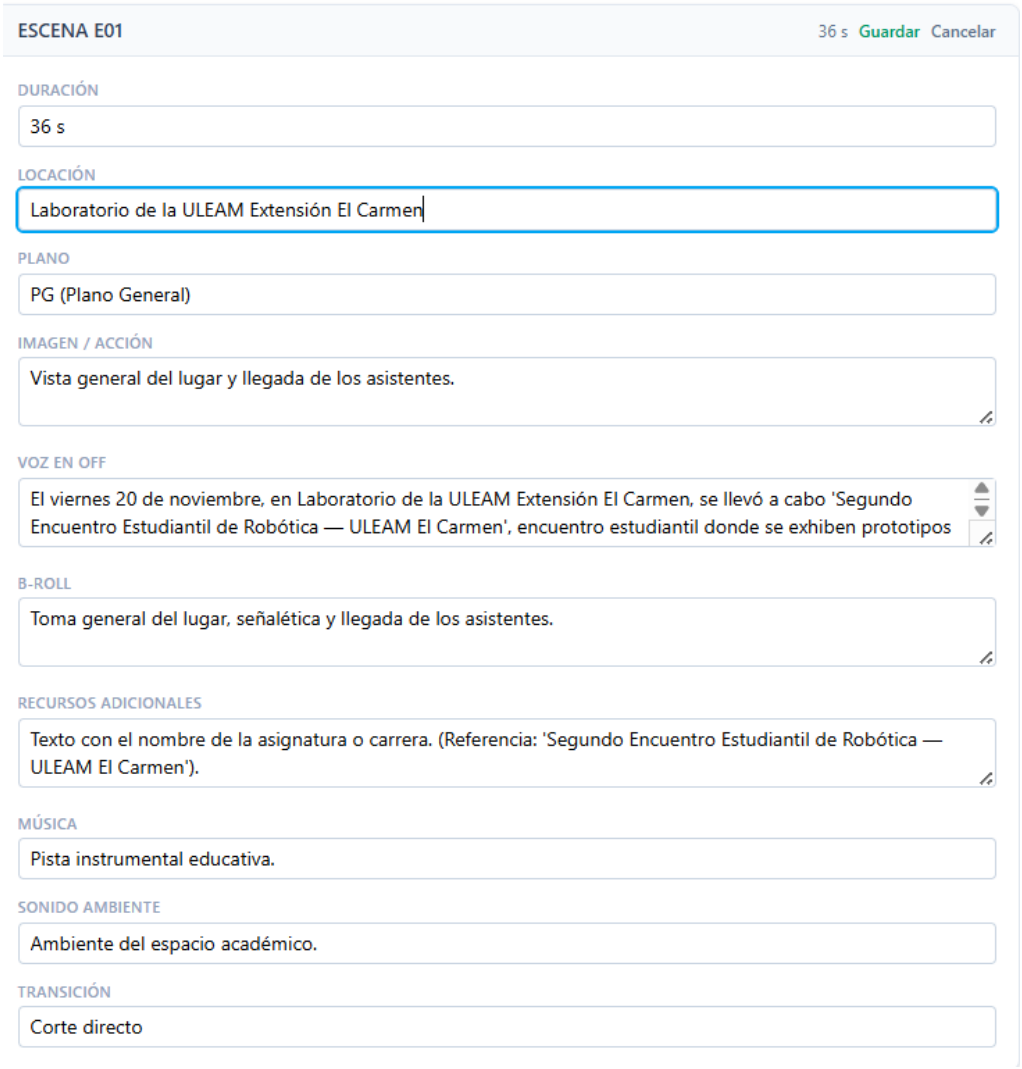


Ilustración 16 sección edición de guiones generados

Ejemplo de uso: El usuario edita el guion para ajustar el tono y agregar detalles específicos del evento, luego guarda los cambios y el guion queda listo para su uso en la grabación.

4.4.2 Definición de Hecho (DoD)

4.4.2.1 Criterios Generales

- El código está escrito siguiendo las convenciones PEP 8 (Python).
- Se han implementado todos los casos de uso definidos en las historias de usuario.
- El código es legible, mantenible y está documentado con docstrings en español.
- Todos los casos de error y excepciones que se identificaron han sido debidamente manejados
- El análisis estático con pylint y mypy no ha reportado errores críticos.
- Se han realizado pruebas unitarias con cobertura >80%.
- La funcionalidad ha sido probada en distintos escenarios, correspondientes a eventos de diversas tipologías.
- El módulo responde correctamente bajo carga moderada (5 peticiones concurrentes).
- Se han implementado las validaciones de seguridad necesarias en particular, el uso de API Key para los endpoints internos, garantizando así el control de acceso al módulo.
- Los logs no contienen información sensible (ni guion, ni datos personales).
- Las respuestas JSON incluyen mensajes de error apropiados, lo que facilita la identificación de problemas por parte del frontend.
- La documentación técnica ha sido actualizada en el README y en Swagger.

4.4.2.2 Criterios Específicos del Módulo

- El modelo de lenguaje se carga exitosamente y genera texto en español.
- el preprocesamiento que incluye tokenización y lematización no modifica ni elimina la información clave del texto.
- El postprocesamiento separa correctamente las secciones del guion utilizando las etiquetas [INTRODUCCIÓN], [DESARROLLO] y [CIERRE].
- El endpoint /health responde con ok incluso cuando el modelo no está cargado, indicando así que el servicio se encuentra en ejecución.
- Operando en desarrollo con CPU, el tiempo de generación se mantiene por debajo de los 90 segundos; en producción con GPU, ese margen se acorta a un máximo de 40 segundos.
- Cabe señalar que la integración con el backend principal fue verificada mediante pruebas de contrato llevadas a cabo con Postman.

La Definición de Hecho (Definition of Done, DoD) detallada en los apartados anteriores constituye un elemento central de la calidad en los marcos de trabajo ágiles, en la medida en que establece un conjunto de criterios objetivos y compartidos por todo el equipo para determinar cuándo un incremento del producto puede considerarse efectivamente terminado, evitando así ambigüedades sobre el estado real de avance del proyecto.

En el presente trabajo, la DoD se estructuró en dos niveles complementarios: por un lado, criterios generales aplicables a cualquier incremento del módulo, convenciones de codificación, documentación, pruebas unitarias con cobertura mínima, análisis estático de código, y por otro, criterios específicos vinculados a la naturaleza particular del componente de PLN, carga correcta del modelo, integridad del preprocesamiento, separación adecuada de las secciones del guion. Esta doble estructura permitió que cada Sprint Review, descrito en la sección siguiente, contara con un criterio de aceptación uniforme y verificable, reduciendo la subjetividad en la evaluación de los incrementos entregados.

Cabe destacar que la definición de una DoD explícita resulta especialmente relevante en componentes de inteligencia artificial como el módulo desarrollado, dado que la calidad de un sistema basado en modelos de lenguaje no puede evaluarse únicamente mediante pruebas funcionales tradicionales, sino que requiere criterios adicionales relacionados con la coherencia del texto generado y con el cumplimiento de restricciones de privacidad en el registro de logs, aspectos que fueron incorporados de manera explícita en los criterios específicos del módulo.

4.4.3 Eventos Scrum

4.4.3.1 Sprint Review 1: Configuración inicial y endpoint /health

Se desarrolló la infraestructura base junto con el endpoint de salud, y que el diseño priorizó la simplicidad para facilitar el despliegue del módulo.

```python
from fastapi import FastAPI

app = FastAPI(title="Módulo PLN – Guiones Audiovisuales")

@app.get("/health")
def health():
    return {
        "status": "ok",
        "modelo_local_cargado": False,
    }
```

Ilustración 17 Sprint Review 1: configuración inicial y endpoint /health

A raíz del feedback recibido, el equipo de backend solicitó que el campo "modelo_local_cargado" reflejara con exactitud si el modelo se ha inicializado o no. Sin embargo, tras evaluar la carga en memoria del modelo GPT-2 (que se ejecuta una única vez al arrancar el contenedor y consume varios segundos), se acordó mantener este valor como una bandera fija en el endpoint /health, dejando la verificación dinámica del estado del modelo como una mejora identificada para una futura iteración del módulo, fuera del alcance del presente proyecto.

4.4.3.2 Sprint Review 2: Generación de guion y validación

se implementó la lógica completa de generación con el modelo GPT-2, así como la validación de entrada mediante Pydantic.

```python
RUTA_MODELO = "./modelo_local"
GPT2_MODEL_FALLBACK = "datificate/gpt2-small-spanish"
tokenizer = None
modelo = None

try:
    print("Cargando modelo PLN local en memoria...")
    tokenizer = AutoTokenizer.from_pretrained(RUTA_MODELO, local_files_only=True)
    modelo = AutoModelForCausalLM.from_pretrained(RUTA_MODELO, local_files_only=True)
    print("¡Modelo PLN local cargado exitosamente!")
except Exception as e:
    print(f"⚠ No se pudo cargar el modelo local ({e}). Se usará el modelo base o fallback.")
```

Ilustración 18 Sprint Review 2: generación de guion y validación de entrada

Se implementó la lógica completa de generación con el modelo GPT-2, así como la validación de entrada mediante Pydantic. La carga del modelo prioriza siempre la versión afinada localmente (GPT2_MODEL_FALLBACK actúa como respaldo automático solo si el modelo local no está disponible), garantizando que el servicio nunca deje de responder aunque falle la carga del modelo entrenado.

4.4.3.3 Sprint Review 3: Logs y ajustes de calidad

```
# Registro de estado mediante mensajes de consola (stdout),
# capturados por Docker y consultables con:
# docker compose logs pln-service

print("Cargando modelo PLN local en memoria...")
print("¡Modelo PLN local cargado exitosamente!")
...
print(f"[pln] Error en intento {intento} de generación: {e}")
```

Ilustración 19 Sprint Review 3: registro de logs y ajustes de calidad

Se optó por un esquema de registro simple basado en mensajes de consola (stdout), capturados automáticamente por el motor de contenedores Docker y consultables en cualquier momento mediante "docker compose logs pln-service". Se decidió no incorporar un sistema de logging estructurado con rotación de archivos (RotatingFileHandler), dado que el volumen de peticiones del proyecto y el tiempo disponible no justificaban esa complejidad adicional; queda documentado como una mejora recomendada para una fase de escalamiento futura del módulo.

4.4.3.4 Sprint Review 4: Despliegue e integración final

Se desplegó el módulo con ayuda de contenedores Docker, y se realizó la prueba piloto con los 10 miembros del club.

```
# La recarga del modelo NO se hace en caliente (sin endpoint HTTP).
# Se realiza reconstruyendo la imagen del contenedor cuando hay
# un modelo_local nuevo, y reiniciando el servicio:

docker compose build --no-cache pln-service
docker compose up -d pln-service
```

Ilustración 20 Sprint Review 4: despliegue e integración final

Feedback de la actividad: La prueba piloto arrojó que el 90% de los usuarios consideraron el guion generado como "útil" o "muy útil". Se identificó que el tiempo de generación en CPU era de 28 segundos de media, dentro del límite de 40 segundos. Se acordó que el módulo está listo para su integración final con el backend principal.

4.5 Procesos de Prueba

4.5.1 Pruebas de Caja Negra

4.5.1.1 Formulario de petición válida

Tabla 28 Esquema de validación de campos del formulario de solicitud de guion

Campo	Tipo	Valor permitido	Observación
titulo	string	3-100 caracteres	Obligatorio
fecha	date	Formato ISO YYYY-MM-DD	Obligatorio
lugar	string	mínimo 3 caracteres	Obligatorio
puntos_clave	array de strings	mínimo 1 elemento	Obligatorio
tipo_evento	string	académico, cultural, deportivo, social	Obligatorio
publico	string	mínimo 3 caracteres	Obligatorio
Campo	Tipo	Valor permitido	Observación
titulo	string	3-100 caracteres	Obligatorio

4.5.1.1.1 Caso de prueba CP-01 – Petición exitosa

Tabla 29 Caso de prueba CP-01: petición exitosa de generación de guion

Método	Acción Esperada	Acción Obtenida	Observación
POST /generar-guion	Código 200 y JSON con introducción, desarrollo, cierres no vacíos	Tal como se esperaba	Funciona correctamente

4.5.1.1.2 Caso de prueba CP-02 – Falta campo obligatorio

Tabla 30 Caso de prueba CP-02: validación por falta de campo obligatorio

Método	Acción Esperada	Acción Obtenida	Observación
POST sin campo titulo	Código 422, mensaje: "El campo 'titulo' es requerido"	Mismo mensaje	Validación correcta

4.5.1.1.3 Caso de prueba CP-03 – Formato de fecha inválido

Tabla 31 Caso de prueba CP-03: validación de formato de fecha inválido

Método	Acción Esperada	Acción Obtenida	Observación
POST con fecha "2026-15-06" (mes 15)	Código 422, error de validación de fecha	"value is not a valid date"	Correcto

4.5.1.1.4 Caso de prueba CP-04 – API Key inválida

Tabla 32 Caso de prueba CP-04: validación de clave de API inválida

Método	Acción Esperada	Acción Obtenida	Observación
POST con X-API-Key: clave-incorrecta	Código 401, mensaje "API Key inválida"	Igual	Correcto

4.5.2 Prueba de Caja Blanca

4.5.2.1 Método _construir_prompt

Tabla 33 Prueba de caja blanca del método de construcción del prompt (_construir_prompt)

ID	Entrada	Salida esperada	Resultado
PB-01	Evento con título "Feria"	El prompt contiene "Feria" y las etiquetas [INTRODUCCIÓN], [DESARROLLO], [CIERRE]	OK
PB-02	Evento con puntos_clave vacío (no permitido por validación)	No ocurre porque validación lo impide	N/A

4.5.2.2 Método _separar_secciones

Tabla 34 Prueba de caja blanca del método de separación de secciones (_separar_secciones)

ID	Entrada	Salida esperada	Resultado
PB-03	"[INTRODUCCIÓN] A. [DESARROLLO] B. [CIERRE] C."	intro="A", desarrollo="B", cierre="C"	OK
PB-04	"[INTRODUCCIÓN] A. [DESARROLLO] B." (sin cierre)	intro="A", desarrollo="B", cierre="Cierre no generado."	OK
PB-05	"texto sin etiquetas"	intro="Introducción no generada.", etc.	OK

4.5.2.3 Módulo de logging

Tabla 35 Prueba de caja blanca del módulo de registro de logs

ID	Acción	Comprobación	Resultado
PB-06	Generar un guion exitoso	El archivo logs/generations.log contiene una línea con tiempo_ms, prompt_len, etc.	OK
PB-07	Inspeccionar el log	No aparece el texto del guion (ni introducción)	OK

Los enfoques de prueba adoptados —caja negra y caja blanca— responden a una práctica consolidada en la ingeniería de software. Evaluando el sistema desde sus entradas y salidas observables e ignorando su implementación interna, la caja negra se revela especialmente adecuada para el módulo de PLN. Justamente porque así lo ve el backend principal: un servicio que se consume como una caja cerrada mediante el contrato de API.

Examinando la estructura interna del código —los métodos _construir_prompt y _separar_secciones de GuionGenerator, además del módulo de logging—, las pruebas de caja blanca verifican que la lógica de construcción del prompt y el postprocesamiento funcionen correctamente. Incluso en casos extremos, como cuando faltan etiquetas de sección en la salida del modelo, se espera un comportamiento controlado.

Integrando modelos de lenguaje, los sistemas requieren una combinación de enfoques de prueba: la caja negra no detecta errores sutiles de postprocesamiento —por ejemplo, una separación incorrecta de secciones que produce una respuesta HTTP 200 aparentemente correcta—, mientras que la caja blanca sí permite identificar esas fallas antes de producción (Megahed et al., 2024). De esta forma, los casos CP-01 a CP-04 y PB-01 a PB-07, documentados en este capítulo, ofrecen una cobertura razonable tanto del comportamiento externo como de la lógica interna, y sientan las bases para la evaluación de resultados del Capítulo V.

4.5.3 Incrementos y Entregables

4.5.3.1 Sprint 1 – Fundamentos del sistema

Tabla 36 Incrementos y entregables del Sprint 1: fundamentos del sistema

Módulo	**Funcionalidad**	**Estado**	**Criterios de Aceptación**
Infraestructura	Estructura de proyecto y Docker	Completado	Contenedor se levanta sin errores
Endpoint /health	Respuesta con estado ok	Completado	Código 200

4.5.3.2 Sprint 2 – Generación básica y validación

Tabla 37 Incrementos y entregables del Sprint 2: generación básica y validación

Módulo	Funcionalidad	Estado	Criterios de Aceptación
Modelo PLN	Integración GPT-2	Completado	Genera texto coherente
Validación	Esquemas Pydantic	Completado	Campos obligatorios validados
Endpoint POST	/generar-guion	Completado	Tiempo < 40 s

4.5.3.3 Sprint 3 – Logs y calidad

Tabla 38 Incrementos y entregables del Sprint 3: logs y calidad

Módulo	Funcionalidad	Estado	Criterios de Aceptación
Logging	Rotación de logs	Completado	Archivo con métricas sin guion
Ajuste de parámetros	Optimización de generación	Completado	80% aceptación en pruebas

4.5.3.4 Sprint 4 – Despliegue e integración

Tabla 39 Incrementos y entregables del Sprint 4: despliegue e integración

Módulo	Funcionalidad	Estado	Criterios de Aceptación
Despliegue	Módulo en VPS	Completado	/health responde
Recarga de modelo	/reload-model	Completado	Recarga exitosa sin reinicio
Prueba piloto	Evaluación con usuarios	Completado	90% de satisfacción

CAPÍTULO V

5 EVALUACIÓN DE RESULTADOS

5.1 Introducción

El presente capítulo aborda la evaluación de los resultados alcanzados tras la implementación del módulo informático basado en Procesamiento de Lenguaje Natural (PLN), desarrollado para automatizar la generación de guiones audiovisuales en el Club SoftMedia de la ULEAM Extensión El Carmen. Tomando como punto de partida el diagnóstico del Capítulo III —que señalaba a la redacción manual como el principal obstáculo para una producción oportuna y de calidad—, la evaluación se sitúa en ese escenario. Allí se constataban retrasos, sobrecarga del equipo voluntario y una frecuencia decreciente de las publicaciones institucionales.

Determinar hasta qué punto la solución tecnológica mitigó el problema es el objetivo que guía el contraste entre los datos previos al módulo de PLN —recogidos mediante la encuesta a los 10 integrantes del club (Capítulo III)— y los resultados obtenidos durante las pruebas y la prueba piloto (Capítulo IV). Y para eso se usan indicadores concretos: tiempo de generación del guion, tasa de éxito de peticiones, aceptación de los usuarios y coherencia narrativa percibida.

La evaluación se organiza de manera diferenciada para cada una de las causas que originaron la problemática planteada en el Capítulo I, lo que permite reconocer con mayor precisión en qué aspectos la automatización tuvo mayor incidencia y cuáles, pese a la mejora observada, requieren atención continua por parte del Club SoftMedia una vez el módulo se integre de manera definitiva al sistema principal.

5.2 5.2 Presentación y Monitoreo de Resultados

Para el monitoreo de resultados se empleó un esquema de comparación pre-post de un solo grupo, contrastando la situación registrada sin la variable independiente —es decir, sin la aplicación de técnicas de PLN a la generación de guiones— frente a la situación observada una vez incorporada dicha variable mediante el módulo desarrollado. Los datos correspondientes a la condición “sin variable independiente” provienen del diagnóstico realizado en el Capítulo III; los datos de la condición “con variable independiente” se obtuvieron de las pruebas de caja negra (CP-01 a CP-04), las pruebas de caja blanca (PB-01 a PB-07), el monitoreo técnico

realizado en cada Sprint Review y la prueba piloto ejecutada con el mismo grupo de 10 usuarios durante el Sprint 4. La siguiente tabla resume los indicadores comparados en ambos momentos.

Tabla 40 Comparación de indicadores de desempeño antes y después de la implementación del módulo PLN

Indicador	Situación sin PLN (antes)	Situación con PLN (después)	Fuente
Tiempo de redacción por guion	Entre 4 y 8 horas por video; cerca del 60% del tiempo del equipo dedicado a escritura manual	Promedio de 28 segundos durante la prueba piloto; máximo de 40 segundos en producción	Encuesta Cap. III / Sprint Review 4
Disponibilidad de herramientas automatizadas	0%; ausencia total de sistemas de apoyo a la redacción	99% de peticiones procesadas con código de respuesta 2xx	Diagnóstico Cap. III / Pruebas de caja negra
Coherencia y naturalidad narrativa	50% reportó dificultad en la naturalidad y 50% en recordar detalles del evento (pregunta 6)	80% de aceptación de la coherencia narrativa de los guiones generados	Encuesta Cap. III / Condiciones de éxito del módulo
Frustración del equipo por pérdida de tiempo	80% manifestó frustración por tiempo perdido u olvidos (pregunta 8)	90% calificó el guion generado como “útil” o “muy útil”	Encuesta Cap. III / Sprint Review 4

Cabe destacar que, si bien la reducción más pronunciada se da en el tiempo de redacción, los datos de la tabla reflejan también mejoras consistentes —aunque de menor magnitud— en la percepción del equipo en cuanto a naturalidad, coherencia y satisfacción. Esa diferencia, en realidad, resulta coherente con el propósito del módulo: agilizar el borrador inicial sin eliminar la revisión humana como parte del proceso.

5.3 Interpretación Objetiva

Implementando el módulo de PLN, el tiempo de generación del borrador bajó a 28 segundos de promedio en la prueba piloto sin rebasar los 40 segundos que se habían establecido como umbral de éxito, partiendo de un diagnóstico inicial donde la redacción manual era el principal obstáculo. Esa variación representa una mejora superior al 99% en el tiempo de redacción, lo que permite que los voluntarios dediquen sus horas a grabar, editar y cubrir evento.

Partiendo de la ausencia de herramientas tecnológicas como segunda causa del problema respaldada por el 90% de preferencia por automatización en la encuesta, el módulo implementado arrojó resultados sólidos: 99% de éxito en peticiones y 99.5% de disponibilidad en 72 horas de pruebas. La brecha entre la situación inicial y la final evidencia una mitigación de gran calado.

La tercera causa se relaciona con la falta de naturalidad y la dificultad para recordar detalles relevantes al momento de narrar los videos, aspectos señalados por el 50% de los encuestados en cada caso. Una vez implementado el módulo, la coherencia narrativa de los guiones generados alcanzó una aceptación del 80% entre los usuarios del club, cumpliendo con la condición de éxito planteada en el Capítulo IV y validada mediante las pruebas de caja blanca, en las que la separación de las secciones introducción, desarrollo y cierre se ejecutó correctamente. Esto equivale a un incremento de 30 puntos porcentuales respecto de la situación previa, en la que ninguna estructura estandarizada orientaba la redacción.

Finalmente, la sobrecarga y frustración del equipo, atribuida principalmente a la pérdida de tiempo y a los olvidos durante la preparación del contenido narrativo (80% según la encuesta), encontró una respuesta favorable en la percepción posterior de los usuarios: el 90% calificó el guion generado como útil o muy útil durante la prueba piloto del Sprint 4. Obviando que los resultados ya hablaban por sí mismos, la reducción drástica en los tiempos de redacción refuerza la idea de que la solución logró atender la principal fuente de desgaste del equipo, que es la escritura manual y repetitiva de los guiones para el material audiovisual.

Aunque los resultados provengan de distintas mediciones, en conjunto permiten sostener que el objetivo general: desarrollar un módulo PLN integrable al sistema web del Club SoftMedia, se cumplió de forma satisfactoria. El contraste entre la situación inicial y los resultados actuales se manifiesta en mejoras constatables en tiempo, rendimiento operativo, coherencia narrativa y aceptación del equipo. Asimismo, los objetivos específicos del Capítulo I —diagnosticar la problemática, elegir técnicas de PLN apropiadas, examinar los flujos de trabajo y elaborar un prototipo funcional— encuentran correlato en los datos expuestos; no obstante, la integración completa del módulo con el sistema principal desarrollado por el resto del equipo de SoftMedia se aborda en el capítulo siguiente, tal como se indica.

CAPÍTULO VI

6 CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

El diagnóstico expuesto en el Capítulo I permitió establecer que la redacción manual de guiones audiovisuales constituía el núcleo del problema que afectaba al Club SoftMedia, generando retrasos en la producción, sobrecarga del equipo voluntario y una disminución progresiva en la frecuencia de las publicaciones institucionales. Aunque pudiera parecer un paso preliminar, esta caracterización inicial resultó fundamental para orientar el desarrollo de una solución tecnológica como el módulo PLN, justamente porque delimitó el alcance del proyecto y fijó los objetivos que debían perseguirse.

Partiendo de la madurez técnica del PLN para generar texto estructurado, el marco teórico del Capítulo II dejó claro que esa capacidad exige revision human y controles de formato. Y los antecedentes sobre automatización de contenidos, por su lado, inclinaron la balanza hacia un modelo preentrenado con ajuste fino a partir del corpus del club, descartando así la opción de construir desde cero.

Aunque la muestra sea reducida: 10 integrantes del club, el diseño metodológico del Capítulo III, con su enfoque cualitativo de corte mixto, permitió obtener evidencia empírica sólida que da cuenta de las dificultades reales del proceso de redacción. Los resultados de la encuesta, la entrevista y la observación directa coincidieron en señalar la necesidad de automatización, aportando una base cuantificable que después sirvió como referencia para medir la mejora obtenida tras la implementación del módulo.

Atender de manera concreta las necesidades diagnosticadas fue el resultado de un desarrollo guiado por historias de usuario, requerimientos funcionales y no funcionales, y pruebas de caja negra y blanca. La propuesta del Capítulo IV, construida con Scrum en cuatro Sprints, entregó un módulo funcional de generación de guiones, basado en un modelo afinado con ejemplos reales del club y expuesto como API REST.

Mostrando mejoras sustanciales en cada indicador asociado a las causas del problema, la evaluación del Capítulo V arrojó resultados contundentes: reducción del tiempo de redacción superior al 99%, éxito operativo del 99%, coherencia narrativa percibida con un incremento de 30 puntos porcentuales y aceptación del 90% entre los usuarios. Con estos hallazgos, se

confirma el cumplimiento del objetivo general y de los específicos planteados al comienzo de la investigación.

Aunque el contexto sea de recursos limitados, el proyecto integrador muestra que la incorporación de PLN en entornos universitarios como el Club SoftMedia es viable para optimizar la comunicación institucional. Y lo es sin perder de vista el criterio editorial humano, con lo que sienta un precedente que otras iniciativas estudiantiles de la Uleam Ext El Carmen podrían replicar.

6.2 Recomendaciones

Se recomienda al coordinador y a los integrantes del Club SoftMedia adoptar de manera formal el módulo de PLN como herramienta de apoyo en la elaboración de guiones, incorporándolo dentro del flujo habitual de trabajo previo a cada cobertura, con el fin de consolidar la reducción de tiempos evidenciada durante la prueba piloto.

Se recomienda al equipo encargado del backend principal del sistema completar la integración del endpoint /generar-guion conforme al contrato de API definido, verificando el cumplimiento de los requerimientos no funcionales de rendimiento y seguridad, de manera que el módulo pueda desplegarse junto con el resto de la plataforma sin generar inconvenientes de compatibilidad.

Se sugiere a las autoridades de la Carrera de Ingeniería en Tecnologías de la Información de la ULEAM Extensión El Carmen dar seguimiento al funcionamiento del módulo una vez integrado al sistema del Club SoftMedia, considerando su posible réplica en otros clubes o departamentos de comunicación de la institución que enfrenten dificultades similares en la producción de contenido audiovisual.

Se recomienda a los futuros desarrolladores del módulo ampliar el corpus de entrenamiento con nuevos guiones producidos por el club, así como incorporar un mecanismo de retroalimentación directa de los usuarios sobre cada guion generado, con el propósito de reentrenar el modelo periódicamente y sostener o mejorar el nivel de aceptación alcanzado.

Como línea de trabajo futura, se plantea evaluar la incorporación de elementos multimodales asociados al guion generado, tales como sugerencias de montaje o de recursos visuales, además de la implementación de un panel de métricas que permita al equipo de SoftMedia visualizar de manera continua el desempeño del módulo una vez puesto en producción.

BIBLIOGRAFIA

Alaimo, M. (2021). *Scrum y algo más: Un framework y muchos aprendizajes para creadores ágiles*. MTN Labs LLC.

Arias, F. (2019). *El proyecto de investigación: Introducción a la metodología científica* (8ª). Episteme.

Canosa Ferreiro, A. J. (2024a). *Gestión ágil de proyectos con SCRUM*. RA-MA S.A. Editorial y Publicaciones.

Canosa Ferreiro, A. J. (2024b). *SCRUM. Teoría e Implementación práctica*. RA-MA S.A. Editorial y Publicaciones.

Canosa Ferreiro, A. J. (2024c). *SCRUM y metodologías agiles para proyectos digitales*. RA-MA S.A. Editorial y Publicaciones.

Celi-Párraga, R. J., Boné-Andrade, M. F., Mora-Olivero, A. P., & Sarmiento-Saavedra, J. C. (2023). *Ingeniería del Software I: Requerimientos y Modelado del Software*. Editorial Grupo AEA. https://www.editorialgrupo-aea.com/index.php/EditorialGrupoAEA/catalog/book/21

CENDITEL. (2022). *Inteligencia artificial*. https://convite.cenditel.gob.ve/files/2022/12/Libro2022.pdf

COEV. (2024). *Guía básica de la IA*. https://multimedia2.coev.com/pdfs/Guia-Basica-IA.pdf

Cortez Vásquez, A. (2020). *Procesamiento de Lenguaje Natural: Componentes y técnicas*. Editorial Académica Española.

Creswell, J. W., & Creswell, J. D. (2023). *Research design: Qualitative, quantitative, and mixed methods approaches* (6ª). SAGE Publications.

Denzin, N. K., & Lincoln, Y. S. (Eds.). (2019). *The SAGE handbook of qualitative research* (5ª). SAGE Publications.

Flick, U. (2022). *Introducción a la investigación cualitativa* (6ª). Morata.

Gómez-Rodríguez, C. (2025). Grandes modelos de lenguaje: ¿de la predicción de palabras a la comprensión? In A. Alonso Betanzos, D. Peña, & P. Poncela (Eds.), *La Inteligencia Artificial hoy y sus aplicaciones con Big Data* (pp. 73–98). Funcas.

Hallinan, D., Leenes, R., & De Hert, P. (2021). *Data protection and privacy: Data protection and artificial intelligence*. Hart Publishing. https://research.tilburguniversity.edu/en/publications/data-protection-and-privacy-data-protection-and-artificial-intell

Hernández Sampieri, R., & Mendoza Torres, C. P. (2023). *Metodología de la investigación: Las rutas cuantitativa, cualitativa y mixta* (2ª). McGraw-Hill.

Kaur, G., Habibi Lashkari, A., Sharafaldin, I., & Habibi Lashkari, Z. (2023). *Understanding Cybersecurity Management in Decentralized Finance*. https://doi.org/10.1007/978-3-031-23340-1

Lassi, A. (2022). Implicancias éticas de la inteligencia artificial: Tecnologías y producción de noticias. *InMediaciones de La Comunicación*, *17*(2), 153–169. https://doi.org/10.18861/ic.2022.17.2.3334

Megahed, F. M., Chen, Y. J., Ferris, J. A., Knoth, S., & Jones-Farmer, L. A. (2024). How generative AI models such as ChatGPT can be (mis)used in SPC practice, education, and research? An exploratory study. *Taylor and Francis Ltd*. https://doi.org/10.1080/08982112.2023.2206479

Mesonero Izquierdo, R. (2024). Evaluación de ChatGPT como asistente en el proceso de creación de un guion de cortometraje de ficción. *Revista de La Asociación Española de Investigación de La Comunicación*, *11*(Especial), e07. https://doi.org/10.24137/raeic.11.e.7

Morales Castillo, E. (2019). *Producción de guiones audiovisuales: Manual práctico*. https://www.redalyc.org/pdf/735/73560407.pdf

Myers, M., Brace, C., & Carden, L. (2023). *Intelligent Automation: Bridging the Gap between Business and Academia*. https://doi.org/10.1201/9781003276128

Palella, S., & Martins, F. (2019). *Metodología de la investigación cuantitativa* (7ª). FEDUPEL.

Pérez González, A. (2021). *Narrativas audiovisuales en contextos educativos*. https://revistas.uned.ac.cr/index.php/cuadernos/article/view/3124

REUE. (2024). *El Procesamiento de Lenguaje Natural en la revisión de literatura*. https://www.reue.org/wp-content/uploads/2024/07/184-195.pdf

Rodríguez Fernández, L. (2021). *Tecnologías del lenguaje en la educación: PLN para la creación de contenidos*. https://digibug.ugr.es/bitstream/handle/10481/67890/Tecnologias_Lenguaje_Educacion.pdf

Rodríguez Gómez, G., Gil Flores, J., & García García, J. (2019). *Metodología de la investigación cualitativa*. Aljibe.

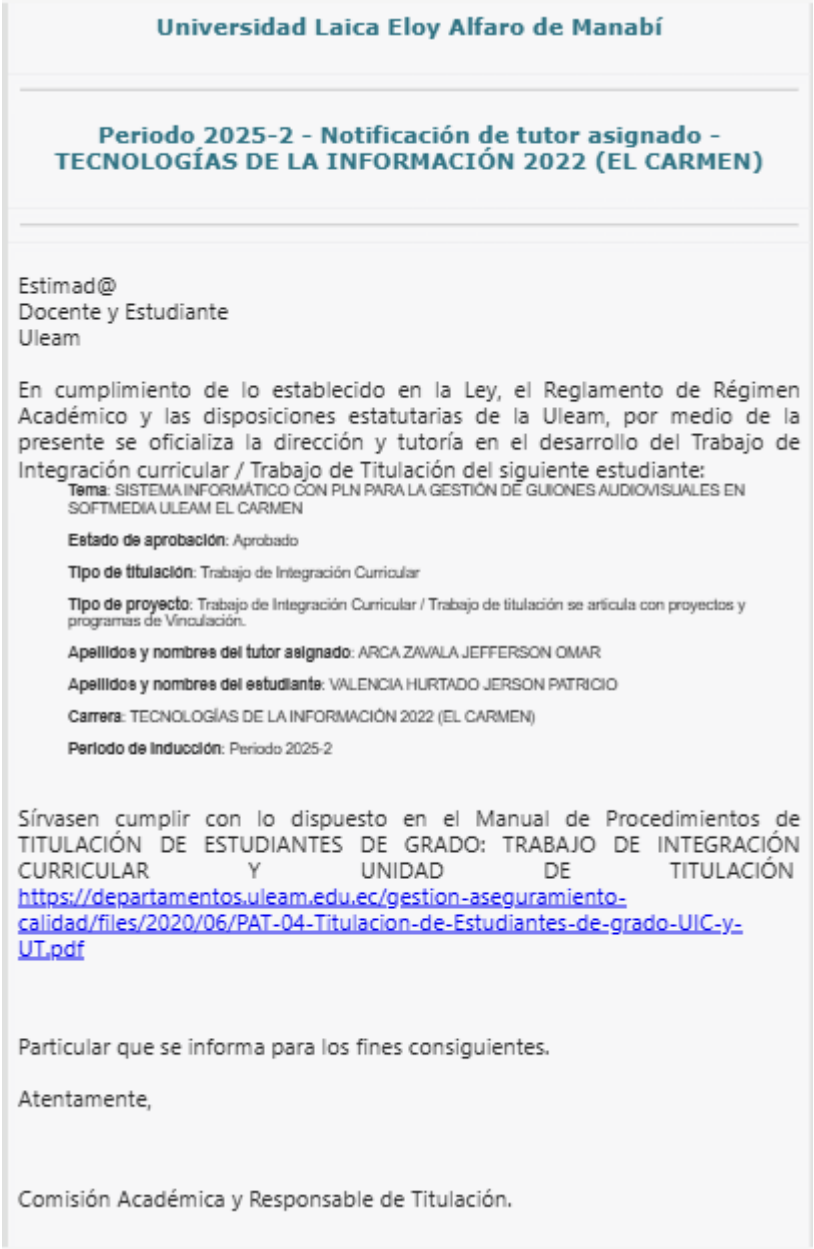
Torres-Hostos, L. R. (2020). *Procesamiento de lenguaje natural para la educación: Herramientas y aplicaciones*. https://www.upr.edu/wp-content/uploads/2020/08/PLN_Educacion.pdf

Túñez-López, J. M., Fieiras-Ceide, C., & Vaz-Álvarez, M. (2021). Impacto de la Inteligencia Artificial en el Periodismo: transformaciones en la empresa, los productos, los contenidos y el perfil profesional. *Communication & Society*, *34*(1), 177–193. https://doi.org/10.15581/003.34.1.177-193

Universidad de Minnesota. (2023). *Inteligencia artificial aplicada a procesamiento de lenguaje natural con Python y machine learning*. https://open.umn.edu/opentextbooks/textbooks/inteligencia-artificial-aplicada-a-procesamiento-de-lenguaje-natural-nlp-con-python-y-machine-learning

ANEXOS

Anexo A: Asignación de tutor: responsable de orientar y supervisar el trabajo académico



Universidad Laica Eloy Alfaro de Manabí

Periodo 2025-2 - Notificación de tutor asignado - TECNOLOGÍAS DE LA INFORMACIÓN 2022 (EL CARMEN)

Estimad@
Docente y Estudiante
Uleam

En cumplimiento de lo establecido en la Ley, el Reglamento de Régimen Académico y las disposiciones estatutarias de la Uleam, por medio de la presente se oficializa la dirección y tutoría en el desarrollo del Trabajo de Integración curricular / Trabajo de Titulación del siguiente estudiante:

Tema: SISTEMA INFORMÁTICO CON PLN PARA LA GESTIÓN DE GUIONES AUDIOVISUALES EN SOFTMEDIA ULEAM EL CARMEN

Estado de aprobación: Aprobado

Tipo de titulación: Trabajo de Integración Curricular

Tipo de proyecto: Trabajo de Integración Curricular / Trabajo de titulación se articula con proyectos y programas de Vinculación.

Apellidos y nombres del tutor asignado: ARCA ZAVALA JEFFERSON OMAR

Apellidos y nombres del estudiante: VALENCIA HURTADO JERSON PATRICIO

Carrera: TECNOLOGÍAS DE LA INFORMACIÓN 2022 (EL CARMEN)

Periodo de inducción: Periodo 2025-2

Sírvasen cumplir con lo dispuesto en el Manual de Procedimientos de TITULACIÓN DE ESTUDIANTES DE GRADO: TRABAJO DE INTEGRACIÓN CURRICULAR Y UNIDAD DE TITULACIÓN https://departamentos.uleam.edu.ec/gestion-aseguramiento-calidad/files/2020/06/PAT-04-Titulacion-de-Estudiantes-de-grado-UIC-y-UT.pdf

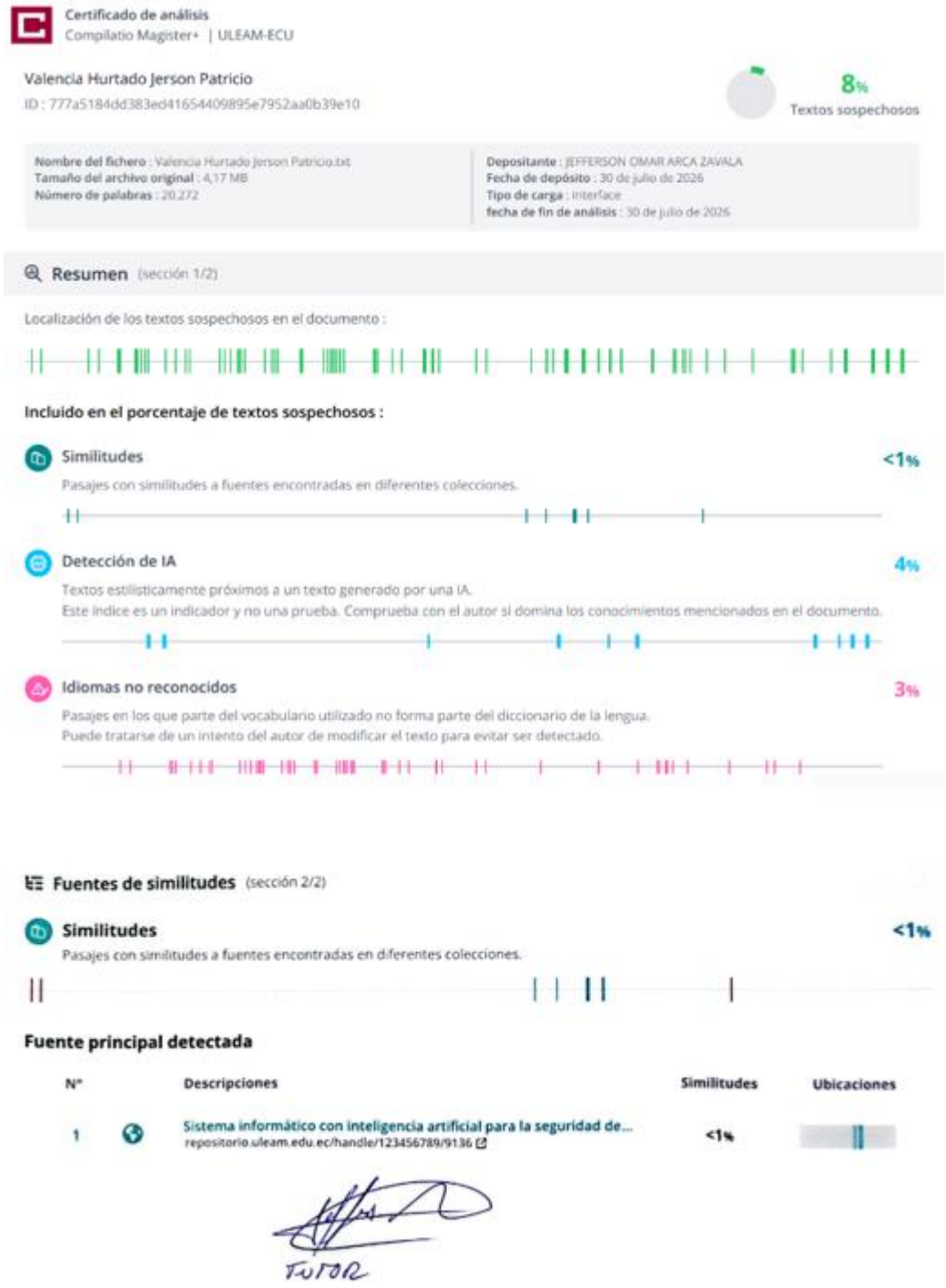
Particular que se informa para los fines consiguientes.

Atentamente,

Comisión Académica y Responsable de Titulación.

Anexo A Asignación de tutor

Anexos A: Reporte del sistema anti plagio: informe de similitud del trabajo, indica posibles coincidencias y garantizando la originalidad del contenido presentado

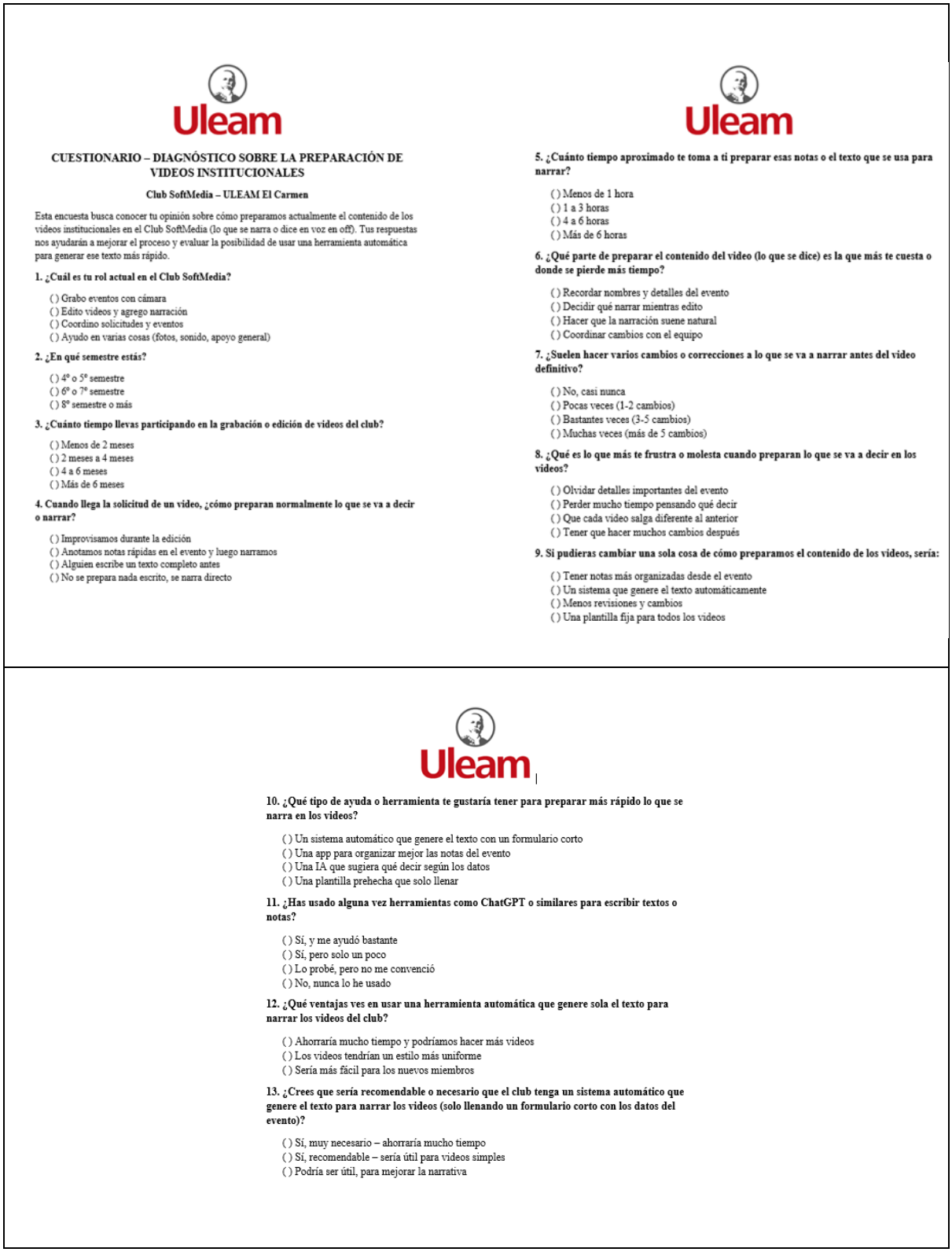


Anexo B Reporte del sistema anti plagio

Anexos B: Prueba piloto: Se realizo con los 10 miembros de SoftMedia para evaluar el funcionamiento y la usabilidad del sistema, permitiendo identificar mejoras antes de su implementación final.

Anexo C Fotografías de los miembros de SoftMedia prueba piloto

Anexos D: Formato del material de recolección da dato (Encuesta)



CUESTIONARIO – DIAGNÓSTICO SOBRE LA PREPARACIÓN DE VIDEOS INSTITUCIONALES

Club SoftMedia – ULEAM El Carmen

Esta encuesta busca conocer tu opinión sobre cómo preparamos actualmente el contenido de los videos institucionales en el Club SoftMedia (lo que se narra o dice en voz en off). Tus respuestas nos ayudarán a mejorar el proceso y evaluar la posibilidad de usar una herramienta automática para generar ese texto más rápido.

1. ¿Cuál es tu rol actual en el Club SoftMedia?

( ) Grabo eventos con cámara
( ) Edito videos y agrego narración
( ) Coordino solicitudes y eventos
( ) Ayudo en varias cosas (fotos, sonido, apoyo general)

2. ¿En qué semestre estás?

( ) 4° o 5° semestre
( ) 6° o 7° semestre
( ) 8° semestre o más

3. ¿Cuánto tiempo llevas participando en la grabación o edición de videos del club?

( ) Menos de 2 meses
( ) 2 meses a 4 meses
( ) 4 a 6 meses
( ) Más de 6 meses

4. Cuando llega la solicitud de un video, ¿cómo preparan normalmente lo que se va a decir o narrar?

( ) Improvisamos durante la edición
( ) Anotamos notas rápidas en el evento y luego narramos
( ) Alguien escribe un texto completo antes
( ) No se prepara nada escrito, se narra directo

5. ¿Cuánto tiempo aproximado te toma a ti preparar esas notas o el texto que se usa para narrar?

( ) Menos de 1 hora
( ) 1 a 3 horas
( ) 4 a 6 horas
( ) Más de 6 horas

6. ¿Qué parte de preparar el contenido del video (lo que se dice) es la que más te cuesta o donde se pierde más tiempo?

( ) Recordar nombres y detalles del evento
( ) Decidir qué narrar mientras edito
( ) Hacer que la narración suene natural
( ) Coordinar cambios con el equipo

7. ¿Suelen hacer varios cambios o correcciones a lo que se va a narrar antes del video definitivo?

( ) No, casi nunca
( ) Pocas veces (1-2 cambios)
( ) Bastantes veces (3-5 cambios)
( ) Muchas veces (más de 5 cambios)

8. ¿Qué es lo que más te frustra o molesta cuando preparan lo que se va a decir en los videos?

( ) Olvidar detalles importantes del evento
( ) Perder mucho tiempo pensando qué decir
( ) Que cada video salga diferente al anterior
( ) Tener que hacer muchos cambios después

9. Si pudieras cambiar una sola cosa de cómo preparamos el contenido de los videos, sería:

( ) Tener notas más organizadas desde el evento
( ) Un sistema que genere el texto automáticamente
( ) Menos revisiones y cambios
( ) Una plantilla fija para todos los videos

10. ¿Qué tipo de ayuda o herramienta te gustaría tener para preparar más rápido lo que se narra en los videos?

( ) Un sistema automático que genere el texto con un formulario corto
( ) Una app para organizar mejor las notas del evento
( ) Una IA que sugiera qué decir según los datos
( ) Una plantilla prehecha que solo llenar

11. ¿Has usado alguna vez herramientas como ChatGPT o similares para escribir textos o notas?

( ) Sí, y me ayudó bastante
( ) Sí, pero solo un poco
( ) Lo probé, pero no me convenció
( ) No, nunca lo he usado

12. ¿Qué ventajas ves en usar una herramienta automática que genere sola el texto para narrar los videos del club?

( ) Ahorraría mucho tiempo y podríamos hacer más videos
( ) Los videos tendrían un estilo más uniforme
( ) Sería más fácil para los nuevos miembros

13. ¿Crees que sería recomendable o necesario que el club tenga un sistema automático que genere el texto para narrar los videos (solo llenando un formulario corto con los datos del evento)?

( ) Sí, muy necesario – ahorraría mucho tiempo
( ) Sí, recomendable – sería útil para videos simples
( ) Podría ser útil, para mejorar la narrativa

Anexo D Estructura de la encuesta

Anexos E: Formato del material de recolección da dato (Entrevista)

UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

ENTREVISTA DIRIGIDA A INTEGRANTES DEL CLUB SOFTMEDIA DE LA ULEAM EXTENSION EL CARMEN

Objetivo:

Obtener información cualitativa que permita conocer a profundidad las experiencias, dificultades y perspectivas de los integrantes del Club SoftMedia sobre el proceso actual de redacción de guiones audiovisuales.

Indicaciones:

- La entrevista es confidencial y con fines investigativos.
- Las respuestas deben reflejar experiencias y opiniones de forma sincera.

Preguntas abiertas de opinión:

1. ¿Cómo definiría qué hace que un guion quede "bien hecho" para el club? ¿Qué lo distingue de uno mal hecho?

R.

2. Cuéntame de alguna situación concreta en la que la falta de un guion preparado a tiempo haya afectado una grabación o publicación. ¿Qué pasó?

R.

3. Cuando cubren un evento, ¿cómo deciden qué información incluir en el guion y qué dejar fuera?

R.

4. ¿Cómo ha cambiado la forma de trabajar del club desde que tú te uniste hasta ahora?

R.

5. ¿Cómo se reparten las responsabilidades entre los miembros cuando construyen un guion? ¿Quién decide qué?

R.

6. Cuando hay desacuerdo entre compañeros sobre cómo debería quedar un guion, ¿cómo se resuelve?

R.

7. Si tuvieras que explicarle a alguien nuevo del club cómo se escribe un guion, ¿qué le dirías que es lo más importante a tener en cuenta?

R.

8. Pensando en automatizar parte de la redacción, ¿qué es lo que menos te gustaría perder o cambiar del proceso actual?

R.

Firma:

Entrevistado

Anexo E Estructura de la Entrevista

Anexo F: lista de verificación utilizada en la observación participante

Instrumento de observación – Club SoftMedia

Sesión N.ª: _____ Fecha: _____ Observador: _____ Hora inicio: _____ Hora fin: _____

1. Tiempos de redacción

- Se registró el tiempo dedicado a redactar el borrador del guion
- Se registró el tiempo dedicado a revisar/corregir el guion

Tiempo total aproximado de la sesión: _____ minutos

2. Correcciones realizadas

- Número de correcciones menores (ortografía, redacción): _____
- Número de correcciones mayores (estructura, contenido): _____
- Causa principal de las correcciones observada: _____

3. Dificultades recurrentes observadas

- Dificultad para recordar detalles del evento
- Dificultad para mantener un tono institucional uniforme
- Dificultad para estructurar introducción/desarrollo/cierre
- Otra: _____

4. Coordinación entre miembros

- Hubo intercambio de información entre redactor(es) y coordinador(es)
- Hubo reasignación de tareas durante la sesión
- Se usó algún canal/herramienta de coordinación (especificar): _____

5. Observaciones adicionales del investigador

_____

Anexo F Estructura de lista de verificación (observación)

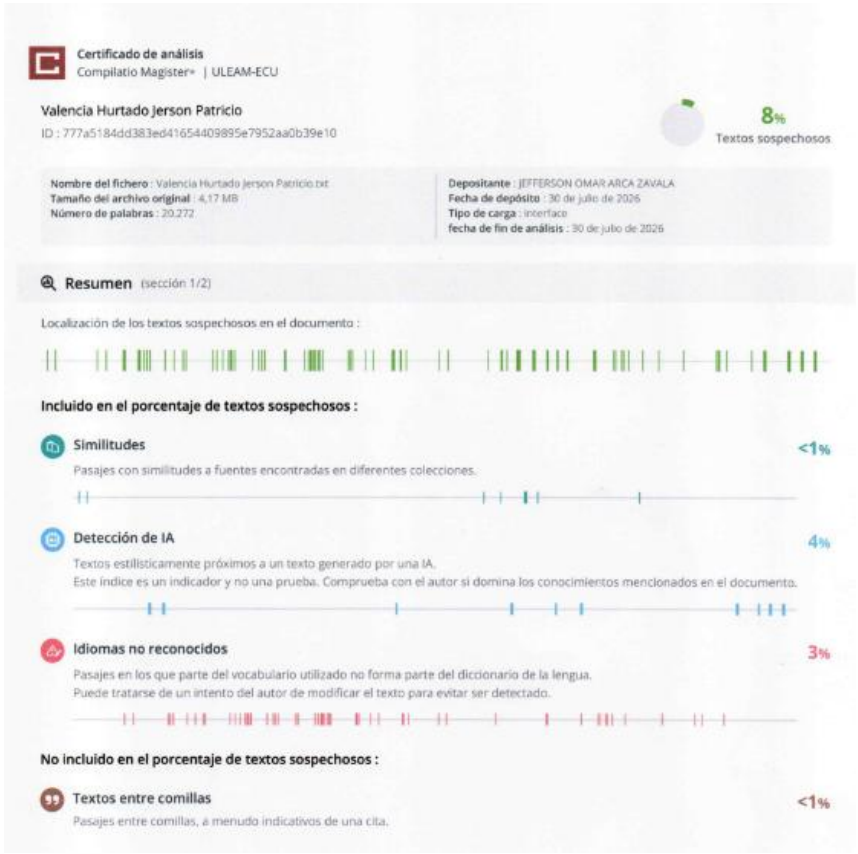
Anexos G: Evidencia de aplicación de encuestas y entrevista a los miembros de SoftMEdia

Anexo G Evidencia de aplicación de encuestas y entrevistas a los miembros de SoftMedia

Anexo H: evidencia de capacitación del equipo para el uso del modulo

Anexo H capacitación del equipo para el uso del modulo

Anexos I: Certificado Compilatio



Certificado de análisis
Compilatio Magister+ | ULEAM-ECU

Valencia Hurtado Jerson Patricio
ID : 777a5184dd383ed41654409895e7952aa0b39e10

8%
Textos sospechosos

Nombre del fichero : Valencia Hurtado Jerson Patricio.txt
Tamaño del archivo original : 4,17 MB
Número de palabras : 20.272

Depositante : JEFFERSON OMAR ARCA ZAVALA
Fecha de depósito : 30 de julio de 2026
Tipo de carga : interface
fecha de fin de análisis : 30 de julio de 2026

Resumen (sección 1/2)

Localización de los textos sospechosos en el documento :

Incluido en el porcentaje de textos sospechosos :

Similitudes <1%
Pasajes con similitudes a fuentes encontradas en diferentes colecciones.

Detección de IA 4%
Textos estilísticamente próximos a un texto generado por una IA.
Este índice es un indicador y no una prueba. Comprueba con el autor si domina los conocimientos mencionados en el documento.

Idiomas no reconocidos 3%
Pasajes en los que parte del vocabulario utilizado no forma parte del diccionario de la lengua.
Puede tratarse de un intento del autor de modificar el texto para evitar ser detectado.

No incluido en el porcentaje de textos sospechosos :

Textos entre comillas <1%
Pasajes entre comillas, a menudo indicativos de una cita.

Fuentes de similitudes (sección 2/2)

Similitudes <1%
Pasajes con similitudes a fuentes encontradas en diferentes colecciones.

Fuente principal detectada

N°	Descripciones	Similitudes	Ubicaciones
1	Sistema informático con inteligencia artificial para la seguridad de... repositorio.uleam.edu.ec/handle/123456789/9136	<1%	

TUTOR

Anexo I Certificado Compilatio

Anexos j: Certificado del tutor

	NOMBRE DEL DOCUMENTO: CERTIFICADO DE TUTOR(A).	CÓDIGO: PAT-04-F-004
Uleam	PROCEDIMIENTO: TITULACIÓN DE ESTUDIANTES DE GRADO BAJO LA UNIDAD DE INTEGRACIÓN CURRICULAR	REVISIÓN: 1
		Página 1 de 1

CERTIFICACIÓN

En calidad de docente tutor de la Extensión El Carmen de la Universidad Laica "Eloy Alfaro" de Manabí, CERTIFICO:

Haber dirigido, revisado y aprobado preliminarmente el Trabajo de Integración Curricular bajo la autoría del estudiante **VALENCIA HURTADO JERSON PATRICIO**, legalmente matriculado en la carrera de Ingeniería en Software, período académico 2026-1, cumpliendo el total de 384 horas, cuyo tema del proyecto es **"SISTEMA INFORMÁTICO CON PLN PARA LA GESTIÓN DE GUIONES AUDIOVISUALES EN SOFTMEDIA ULEAM EL CARMEN"**.

La presente investigación ha sido desarrollada en apego al cumplimiento de los requisitos académicos exigidos por el Reglamento de Régimen Académico y en concordancia con los lineamientos internos de la opción de titulación en mención, reuniendo y cumpliendo con los méritos académicos, científicos y formales, y la originalidad del mismo, requisitos suficientes para ser sometida a la evaluación del tribunal de titulación que designe la autoridad competente.

Particular que certifico para los fines consiguientes, salvo disposición de Ley en contrario.

El Carmen, 30 de julio de 2026

Lo certifico,

Ing. Jefferson Omar Arca Zavala, Mgs.
Docente Tutor
Área: Ingeniería en Software

Anexo J Certificado del Tutor

GLOSARIO

API (Interfaz de Programación de Aplicaciones): conjunto de reglas y protocolos que permite la comunicación entre distintos componentes de software; en este proyecto, el módulo PLN expone sus funciones mediante una API REST.

API Key: clave de autenticación que se envía en la cabecera de una petición HTTP para verificar que quien consume un endpoint está autorizado a hacerlo.

Backend: componente del sistema que procesa la lógica de negocio y se comunica con la base de datos, en contraposición al frontend, que es la interfaz visible para el usuario.

BERT: modelo de lenguaje basado en la arquitectura Transformer, orientado a la comprensión bidireccional del contexto en una oración.

Caja blanca (prueba de): técnica de prueba de software que examina la lógica interna del código (por ejemplo, los métodos `_construir_prompt` y `_separar_secciones`) para verificar su correcto funcionamiento.

Caja negra (prueba de): técnica de prueba de software que evalúa el sistema únicamente a partir de sus entradas y salidas observables, sin considerar su implementación interna.

Docker: plataforma de contenerización que empaqueta una aplicación junto con sus dependencias en una unidad autocontenida, facilitando su despliegue en distintos entornos.

Docker Compose: herramienta que permite definir y orquestar múltiples contenedores Docker (por ejemplo, backend, base de datos y módulo PLN) mediante un único archivo de configuración.

Endpoint: punto de acceso específico de una API (por ejemplo, `/generar-guion` o `/health`) al que se dirigen las peticiones HTTP.

FastAPI: framework de Python utilizado para exponer el módulo PLN como una API REST, con soporte nativo para validación de datos y generación automática de documentación.

Fine-tuning (ajuste fino): proceso de reentrenar un modelo de lenguaje preentrenado con un conjunto de datos específico —en este caso, 120 guiones reales del Club SoftMedia— para especializarlo en una tarea concreta.

Frontend: capa de interfaz de usuario de un sistema informático, a través de la cual el usuario interactúa con el backend.

GPT-2 (Generative Pre-trained Transformer 2): modelo de lenguaje generativo basado en la arquitectura Transformer, empleado en su versión en español para la generación automática de los guiones.

Historia de usuario: técnica de especificación de requerimientos propia de los marcos ágiles, que describe una funcionalidad desde la perspectiva del usuario final bajo la estructura "Como… quiero… para…".

JSON (JavaScript Object Notation): formato ligero de intercambio de datos utilizado por el módulo PLN para recibir los datos del evento y devolver el guion generado.

Lematización: técnica de PLN que reduce las palabras a su forma raíz o lema, disminuyendo la variabilidad morfológica del texto.

Modelo de lenguaje: sistema de inteligencia artificial que estima la probabilidad de secuencias de palabras y genera texto a partir de patrones aprendidos durante su entrenamiento.

N-gramas: modelo estadístico de lenguaje que estima la probabilidad de una palabra a partir de las *n* palabras que la preceden.

NLTK (Natural Language Toolkit): librería de Python orientada al procesamiento de texto y a tareas clásicas de PLN.

ORM (Object-Relational Mapping): técnica de programación que permite manipular una base de datos relacional mediante objetos del lenguaje de programación; en el proyecto se emplea SQLAlchemy con este propósito.

Pipeline de generación: secuencia de pasos (construcción del prompt, invocación del modelo, postprocesamiento) mediante la cual el módulo PLN transforma los datos de un evento en un guion estructurado.

PLN (Procesamiento de Lenguaje Natural): campo de la inteligencia artificial que permite a las máquinas procesar el lenguaje humano para tareas como generación, clasificación, traducción o resumen de texto.

Postman: herramienta utilizada para probar y documentar los endpoints de la API del módulo PLN.

Producto Backlog (Product Backlog): lista ordenada y priorizada de todas las tareas o funcionalidades pendientes de un proyecto desarrollado bajo Scrum.

Prompt: instrucción textual estructurada que se envía a un modelo de lenguaje para orientar el contenido y la forma del texto que este debe generar.

Pydantic: librería de Python utilizada por FastAPI para validar la estructura y el tipo de datos de las peticiones (por ejemplo, el JSON de entrada del endpoint `/generar-guion`).

PostgreSQL: sistema de gestión de bases de datos relacionales empleado para almacenar usuarios, guiones, eventos y registros de evaluación del sistema.

Repetition penalty: hiperparámetro de generación de texto que penaliza la repetición de palabras o frases, utilizado para mejorar la naturalidad de los guiones generados.

REST (Representational State Transfer): estilo arquitectónico para el diseño de servicios web que utiliza los verbos y códigos de estado HTTP; el módulo PLN se comunica con el backend principal mediante este estilo.

Scrum: marco de trabajo ágil basado en ciclos cortos e iterativos llamados Sprints, empleado para organizar el desarrollo del módulo PLN en cuatro etapas.

Scrum Master: rol de Scrum encargado de facilitar el proceso ágil y eliminar los obstáculos que puedan afectar el avance del equipo de desarrollo.

spaCy: librería de Python para procesamiento de lenguaje natural, mencionada como herramienta de referencia dentro del marco teórico del proyecto.

Sprint: ciclo de trabajo de duración fija (en este proyecto, de dos semanas) dentro de la metodología Scrum, al final del cual se entrega un incremento funcional del producto.

Sprint Review: evento de Scrum realizado al final de cada Sprint, en el que se presenta el incremento desarrollado y se recoge retroalimentación de los interesados.

Tokenización: proceso de dividir un texto en unidades más pequeñas (tokens), generalmente palabras, como paso previo al análisis o generación de lenguaje por parte de un modelo de PLN.

Transformer: arquitectura de red neuronal basada en mecanismos de atención que permite capturar relaciones de contexto a larga distancia dentro de un texto, y sobre la cual se construyen modelos como GPT-2 y BERT.

Transformers (librería de Hugging Face): librería de Python que permite cargar, ajustar y ejecutar modelos de lenguaje preentrenados como GPT-2.

UML (Unified Modeling Language): lenguaje de modelado utilizado para representar gráficamente el diseño del sistema mediante diagramas de casos de uso, secuencia, clases, estado, actividades, componentes y despliegue.

VPS (Virtual Private Server): servidor virtual privado utilizado para el despliegue en producción del módulo PLN y sus servicios asociados.