UNIVERSIDAD NACIONAL DE SAN LUIS

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS Y NATURALES

Proyecto Integrador Final

para optar por el título en

INGENIERÍA EN INFORMÁTICA

Asistente Conversacional basado en Inteligencia Artificial para la Prevención de Apuestas Online

Autores:

Maria Luciana Loyola

Fabrizio José Riera Bauer

Director:

Mg Lorena Baigorria Fernández

Co-Director:

Dra. Ana Garis

San Luis - Argentina

Diciembre 2025

AGRADECIMIENTOS

Fabrizio

Quisiera agradecer en primer lugar a mis padres, Luciana Bauer y Omar Riera, por hacer que mi única preocupación fuera estudiar. También a mis hermanos Seyla y Damián, por enseñarme el valor de la constancia, el esfuerzo y la lucha por los sueños. A mi pareja y amigos, gracias por estar siempre, por hacerme reír, por ayudarme a despejar la mente cuando lo necesitaba, por ser mi contención y apoyarme en cada paso. A mis compañeros de universidad, por compartir este camino conmigo, por los almuerzos y cenas en el comedor, los desvelos, los empujones necesarios para continuar y el aprendizaje conjunto.

A mi abuela Ana, que desde el cielo me verá concluir mis estudios.

Luciana

Me gustaría agradecer a mi familia, mi pilar de todos los días, quienes me apoyan y me brindan un ejemplo a seguir. Gracias a ellos, todo lo que he vivido y logrado ha sido posible. En especial, quiero agradecer a mis padres, Cristina Almandoz y Waldo Loyola, que me permitieron y acompañaron en el cumplimiento de cada meta que me propuse, estando presentes en cada paso de mi vida. Agradezco también a mis amigos, mi familia elegida, que me acompañan con palabras de ánimo y cariño, y están a mi lado siempre que los necesito. También quiero agradecer a mis compañeros de carrera, por cada noche de estudio, risa y consejo. Por ultimo agradecer a mi gato, que ha sido mi fiel compañero del día a día. Gracias a todos por su amor y apoyo constante, no solo a lo largo de mi carrera, sino a lo largo de toda mi vida.

Agradecimientos generales

Queremos agradecer a la Universidad Nacional de San Luis y a la Facultad de Ciencias Físico, Matemáticas y Naturales por brindarnos la posibilidad de acceder a una educación universitaria pública, libre, gratuita y de calidad. Esta institución no solo nos permitió ampliar nuestros conocimientos y crecer como profesionales y personas, sino que también nos proporcionó herramientas fundamentales y nos dio la oportunidad de conocer personas increíbles que nos acompañaron durante este camino.

Agradecemos especialmente al equipo del proyecto de Psicología: Eliana, Sergio y Nehuen, por confiar en nosotros para llevar adelante este proyecto. Gracias por compartir sus conocimientos y experiencia, y por guiarnos en la comprensión de la problemática. Ha sido un placer trabajar con ustedes y aprender de su experiencia en el campo de la Psicología y la Salud Mental.

Finalmente, nuestro más profundo agradecimiento a nuestras directoras, Lorena Baigorria y Ana Garis, quienes nos acompañaron a lo largo de todo este proceso y cuya guía fue fundamental para que este proyecto sea lo que es hoy. Gracias por sus conocimientos y consejos, sin su apoyo, este trabajo no habría sido posible.

RESUMEN

Las apuestas en línea se han convertido en una problemática creciente entre adolescentes y jóvenes adultos en Argentina. Estudios recientes muestran que el 40 % de los adolescentes ha apostado en línea al menos una vez y muchos lo hacen de manera frecuente. Esta situación representa un riesgo significativo para el desarrollo de ludopatía a edades tempranas, con consecuencias graves para la salud mental y el bienestar de los jóvenes. La accesibilidad constante a plataformas de apuestas y la normalización de esta práctica requieren herramientas innovadoras de prevención y acompañamiento.

En este trabajo se presenta el desarrollo de un asistente conversacional web basado en Large Language Models (Grandes Modelos de Lenguaje) (LLM), diseñado para el abordaje y la prevención del consumo problemático y/o posible adicción a las apuestas en línea en adolescentes, principalmente, pero también disponible al público en general. El asistente utiliza técnicas de Retrieval-Augmented Generation (Recuperación Aumentada por Generación) (RAG), sustentado por una fuente de conocimiento especializada provista por un grupo de investigadores expertos de la Facultad de Psicología de la Universidad Nacional de San Luis (UNSL).

La propuesta surge ante la necesidad planteada por dicho grupo de investigación, interesado en evaluar la efectividad del asistente como instrumento de prevención de apuestas en línea. El proyecto se basa en un enfoque multidisciplinario que integra los conocimientos de la Informática y la Psicología, permitiendo que el asistente mantenga conversaciones naturales y coherentes, brindando información precisa y relevante basada en conocimiento especializado.

El sistema está diseñado para dos contextos de uso diferenciados. Por un lado, permite el acceso público general sin necesidad de registro, funcionando como una herramienta de consulta y orientación disponible para cualquier persona. Por otro lado, facilita pruebas controladas en entornos escolares bajo supervisión de especialistas en Psicología, con autorización de padres o tutores. En ambos casos, el sistema genera estadísticas para evaluar su efectividad como herramienta preventiva y recopila información sobre las interacciones en las conversaciones, datos relevantes para el proyecto de investigación que propuso este desarrollo y que permiten obtener información valiosa sobre esta problemática.

El proyecto se adhiere a las guías de la Agencia de Acceso a la Información Pública de Argentina para asegurar la transparencia, la protección de datos personales y el uso responsable de la Inteligencia Artificial (IA). El sistema busca mantener el anonimato de los usuarios, la confidencialidad de las conversaciones y la protección de los datos personales.

Se espera que esta herramienta sirva como un recurso innovador de prevención, ofreciendo un espacio seguro donde adolescentes y adultos puedan informarse sobre el juego responsable, reconocer señales de riesgo y acceder a recursos de ayuda profesional cuando sea necesario. El proyecto representa un esfuerzo interdisciplinario entre las áreas de informática y Psicología de la UNSL para crear soluciones tecnológicas que contribuyan al bienestar de la comunidad.

Índice general

Índice de Figuras

Índice de Tablas

Lista de Acrónimos

AAIP Agencia de Acceso a la Información Pública de Argentina

API Application Programming Interfaces (Interfaz de Programación de Aplicaciones)

IA Inteligencia Artificial

LLM Large Language Models (Grandes Modelos de Lenguaje)

PLN Procesamiento del Lenguaje Natural

RAG Retrieval-Augmented Generation (Recuperación Aumentada por Generación)

RN Redes Neuronales

SDK Software Development Kit (Kit de Desarrollo de Software)

XP Extreme Programming (Programación Extrema)

Capítulo 1

INTRODUCCIÓN

En la actualidad, está surgiendo una problematica grave relacionada con las apuestas en línea en la etapa de la adolescencia. El acceso temprano a internet, videojuegos y plataformas de apuestas en línea expone a niños y adolescentes a conductas de juego que antes eran poco frecuentes. Esta exposición constante contribuye a normalizar la práctica del juego, lo que incrementa el riesgo de desarrollar ludopatía a edades cada vez más tempranas [1].

La preocupación se intensifica debido a que el juego online es la modalidad más extendida entre jóvenes de 14 a 18 años, con riesgos cada vez mayores de conductas adictivas [2]. Tanto el Manual Diagnóstico y Estadístico de los Trastornos Mentales (DSM, por sus siglas en inglés) como la Clasificación Internacional de Enfermedades (CIE) reconocen al juego patológico o trastorno por apuestas —incluyendo su variante predominantemente online— como una problemática clínica con alta prevalencia y consecuencias significativas en la salud mental. A nivel preventivo, diferentes estudios [3, 4] coinciden en la necesidad de crear instrumentos específicos y adaptados a la población adolescente, que contemplen factores contextuales como la influencia de pares, el anonimato digital y el acceso irrestricto a plataformas de apuestas online.

En Argentina, el 8,29 % de la población declaró haber apostado online al menos una vez en su vida; la cifra asciende al 12,5 % entre 15 y 24 años y al 15,5 % entre 25 a 34 años [5]. Además, un informe de UNICEF Argentina de 2024 reveló que el 40 % de adolescentes y jóvenes había apostado online al menos una vez y dentro de ese grupo, el 37 % lo hace de manera frecuente o diaria [6]. Estos datos coinciden con lo mostrado por estudios de la Universidad Nacional de Córdoba que indican que el 16 % de los jóvenes apostadores online experimentaron problemas personales o sociales relacionados con el juego, mientras que el 22 % presentó signos de riesgo que podrían derivar en complicaciones mayores [7].

A partir de este análisis, la comunidad científica ha indagado sobre estrategias de prevención. Podría darse que, al tratar estos temas sensibles, los adolescentes se sintieran más cómodos hablando con un asistente virtual, quizás porque no se sentirían juzgados y eso les permitiría expresarse con más libertad.

En el ámbito de la salud mental, la IA se ha posicionado como una herramienta transformadora, ofreciendo posibilidades de detección temprana, acompañamiento personalizado y provisión de recursos preventivos mediante asistentes virtuales entrenados en Procesamiento del Lenguaje Natural (PLN) [8].

De esta forma, un equipo de investigación conformado por investigadores y estudiantes de la Facultad de Psicología de la UNSL, planteó una hipótesis que indicaba que el uso de un asistente virtual podría convertirse en un recurso preventivo para identificar conductas de riesgo asociadas al juego online en estudiantes mayores de 13 años, como así también para la población en general. Asimismo, se esperaba que la herramienta fuera accesible y útil para quienes la utilicen.

Desde un equipo interdisciplinario conformado por docentes y estudiantes de la Licenciatura en Psicología y de Ingeniería en Informática, se propuso el diseño, ajuste y evaluación preliminar de un asistente virtual basado en IA como herramienta preventiva frente al consumo abusivo y/o adicción al juego online en población adolescente.

Para dar respuesta a esta problemática emergente, el presente proyecto consta del desarrollo de un asistente conversacional web basado en LLM [9], diseñado para el abordaje y la prevención del consumo problemático y/o posible adicción a las apuestas en línea en adolescentes, principalmente, pero también disponible al público en general. La implementación del sistema utiliza técnicas de RAG [10], sustentado por una base de conocimiento especializada desarrollada en colaboración con investigadores del área de la Psicología. Además, se tendrán en cuenta recomendaciones para el desarrollo de una IA responsable y la protección de los datos personales de los usuarios, por lo que todos los mensajes serán anónimos y estarán cifrados. Este enfoque multidisciplinario integra los conocimientos de la informática y la Psicología, permitiendo crear una herramienta tecnológica que responde tanto a los requerimientos técnicos como a las necesidades específicas de esta problemática.

1.1. Propuesta

A partir del problema planteado en la sección anterior, se propone el desarrollo de un asistente conversacional web basado en IA para la prevención y abordaje de la adicción a las apuestas en línea, dirigido principalmente a jóvenes y adolescentes, aunque también disponible para público en general.

El asistente estará basado en grandes modelos de lenguaje (LLM) y utilizará técnicas de recuperación aumentada de información (RAG) para enriquecer sus respuestas con información especializada y validada por profesionales de Psicología. Esta arquitectura permitirá que se proporcione orientación precisa, actualizada y contextualmente relevante sobre prevención al consumo problemático, abordaje al juego responsable y detección temprana de conductas problemáticas asociadas al juego.

La propuesta contempla dos modalidades de uso diferenciadas. Por un lado, un acceso público general a través de la plataforma web, donde cualquier usuario mayor de 13 años podrá interactuar con el asistente de forma anónima y confidencial, sin necesidad de registro previo. Por otro lado, se habilitarán sesiones controladas en entornos escolares, supervisadas por especialistas y con autorización previa de padres o tutores, donde los estudiantes accederán mediante credenciales temporales proporcionadas por los profesionales a cargo.

El sistema implementará mecanismos de recopilación de datos estadísticos de forma anónima, respetando la privacidad de los usuarios. Esta información permitirá al equipo de investigación de Psicología analizar patrones de interacción, evaluar la efectividad del asistente como herramienta preventiva, identificar niveles de riesgo asociados al juego online y generar conocimiento sobre esta problemática emergente.

El desarrollo del proyecto seguirá una metodología ágil basada en Programación Extrema (XP, por sus siglas en inglés Extreme Programming) [11], priorizando la entrega incremental, la retroalimentación continua con los expertos en Psicología y la adaptabilidad. Se incorporarán además las recomendaciones de la Agencia de Acceso a la Información Pública de Argentina (AAIP) [12] para el desarrollo de sistemas de IA responsables y el proyecto incorporará principios de transparencia, privacidad, minimización de datos y seguridad.

Esta propuesta representa un esfuerzo interdisciplinario entre las áreas de Informática y Psicología, buscando crear una herramienta tecnológica innovadora que contribuya efectivamente a la prevención de conductas adictivas relacionadas con las apuestas en línea, ofreciendo un espacio seguro donde los usuarios puedan informarse, reflexionar sobre sus hábitos y disponer de orientación profesional de manera accesible y no intimidante.

1.2. Objetivos

El objetivo general de este Proyecto Integrador es desarrollar un asistente conversacional web basado en LLM, integrado con técnicas de RAG, orientado a abordar la problemática de las apuestas online en jóvenes, adolescentes y población en general, que contribuya a brindar contención y acompañamiento, promoviendo el juego responsable y ofreciendo recursos para comunicarse con centros de atención en caso de detectarse un comportamiento de juego problemático y/o patológico en el usuario. Dicho asistente servirá de soporte a un proyecto de investigación de la Facultad de Psicología llamado "Evaluación de la Efectividad y Aceptación de un Asistente Virtual Basado en Inteligencia Artificial para la Prevención de la Adicción al Juego Online en Adolescentes" y es implementado con el fin de evaluar la efectividad y aceptación del asistente, recolectar estadísticas de los chats y detectar conductas de consumo problemáticas o patológicas que puedan ser atendidas por profesionales de la salud, especialmente en escuelas secundarias de la provincia de San Luis.

Objetivos particulares:

Para alcanzar el objetivo general planteado, se establecen los siguientes objetivos particulares:

- Analizar los sistemas y tecnologías existentes relacionados con asistentes virtuales para el apoyo en salud mental, así como las soluciones disponibles para abordar problemáticas de conductas adictivas y/o ludopatías.

- Estudiar las normativas, guías y principios éticos para el desarrollo responsable de sistemas de IA, enfocándose en la transparencia, privacidad y seguridad de la información, con el fin de identificar riesgos y diseñar mecanismos de validación apropiados que cumplan con las recomendaciones de la AAIP.

- Diseñar e implementar un asistente conversacional web basado en LLM que utilice técnicas de RAG para proporcionar información precisa y contextualizada sobre prevención, juego responsable y recursos de ayuda profesional.

- Desarrollar funcionalidades que permitan identificar conductas problemáticas y/o patológicas relacionadas con el juego online, facilitando al usuario el acceso a información preventiva y el contacto con profesionales de la salud locales cuando sea necesario.

- Diseñar y desarrollar dos flujos de uso diferenciados: uno para acceso público general sin necesidad de registro y otro para pruebas controladas en entornos escolares bajo supervisión de especialistas.

- Validar el sistema mediante pruebas continuas que permitan minimizar sesgos, errores y respuestas inapropiadas, asegurando la calidad, seguridad y pertinencia de las respuestas generadas por el asistente.

- Garantizar la protección de datos personales mediante la implementación de medidas de seguridad y cifrado de mensajes.

- Implementar un sistema de recopilación y gestión de datos que genere estadísticas sobre las interacciones de los usuarios con el asistente, garantizando el anonimato y la confidencialidad para su posterior análisis por el grupo de investigación de Psicología.

1.3. Estructura del informe

El presente trabajo está estructurado como se sigue:

El segundo capítulo, “Análisis Ingenieril”, se centra en el análisis del contexto de dominio y desarrollo del sistema. Se detalla el contexto actual de la inteligencia artificial y sus aplicaciones en salud mental, se presenta un estudio de antecedentes de sistemas similares y se justifica la elección de la metodología de desarrollo basada en Extreme Programming (Programación Extrema) (XP) integrada con los lineamientos de la AAIP. También se describen las tecnologías, técnicas y enfoques teóricos aplicados para la implementación, incluyendo LLM, técnicas de RAG, Bases de Datos Vectoriales e Ingeniería de Contexto. Finalmente, se presenta la planificación temporal del proyecto mediante un diagrama de Gantt y se realiza un análisis exhaustivo de costos, considerando tanto la infraestructura tecnológica como los servicios de inteligencia artificial, evaluando opciones locales y en la nube.

El tercer capítulo, “Desarrollo Ingenieril”, aborda los conceptos asociados al desarrollo del sistema siguiendo el ciclo de vida del software. Se describe la captura de requerimientos funcionales y no funcionales, destacando la integración de los principios de IA responsable según la guía de la AAIP. Se presentan las etapas de análisis, diseño, implementación y pruebas del sistema, incluyendo diagramas de casos de uso, diagramas de secuencia, arquitectura general, diagrama entidad-relación y diagramas de componentes y conectores. Se detalla, además, el proceso de despliegue y se documentan las pruebas realizadas, incluyendo el test de aceptación con estudiantes universitarios, métricas de usabilidad y rendimiento y la evaluación del sistema RAG.

El cuarto capítulo, “Caso de Estudio”, muestra dos vistas del sistema: la del estudiante de una escuela y la del administrador o investigador. Se muestra cómo funciona cada una en una prueba controlada dentro de una escuela, acompañado de capturas que ilustran cada parte del proceso.

El quinto capítulo, “Conclusión y Trabajos Futuros”, destaca los resultados obtenidos para los objetivos planteados, como así también el impacto académico y social del sistema y posibles trabajos futuros a realizar.

Adicionalmente, se incluyen dos anexos que complementan la documentación del proyecto. El Anexo I presenta el texto completo de las Políticas de Privacidad implementadas en el sistema. El Anexo II contiene los Términos y Condiciones del Servicio.

Capítulo 2

ANÁLISIS INGENIERIL

Este capítulo presenta el análisis integral del proyecto, abordando tanto el contexto de dominio como las decisiones técnicas y metodológicas que fundamentan el desarrollo del asistente conversacional. Se estudia el panorama actual de la inteligencia artificial y su adopción creciente en diversos ámbitos, con énfasis en las aplicaciones orientadas a la salud mental. Se presenta una comparativa de sistemas similares para identificar las características diferenciadoras de la solución propuesta y se justifica la elección de la metodología de desarrollo basada en Programación Extrema (XP), integrada con los lineamientos de la Guía para el desarrollo responsable de Sistemas Inteligentes.

El capítulo profundiza en los fundamentos tecnológicos del sistema, describiendo las técnicas y enfoques teóricos aplicados. Se detalla la selección del proveedor de inteligencia artificial mediante un análisis comparativo de modelos basado en criterios de inteligencia, velocidad y costo. Asimismo, se presentan las tecnologías empleadas para el desarrollo de la aplicación web. Finalmente, se presenta la planificación del proyecto y se realiza un análisis exhaustivo de costos.

2.1. Contexto de Dominio y de Desarrollo

2.1.1. Contexto actual y antecedentes

La IA ha dejado de ser una tecnología reservada exclusivamente para especialistas y se ha convertido en una herramienta cotidiana. Hoy forma parte de nuestra vida diaria a través de asistentes virtuales y chatbots (como ChatGPT, Copilot, Google Assistant, Alexa o Siri), redes sociales y plataformas de contenido personalizado (gracias a los algoritmos de recomendación de TikTok, Instagram, YouTube, Netflix, entre otros), videojuegos y herramientas de creación de contenido. Se trata de un campo en crecimiento exponencial que, cada día, influye en más aspectos de la vida diaria.

En la actualidad, el uso de la IA no solo ha incrementado de manera acelerada, sino que también se ha expandido a diferentes sectores de la sociedad, alcanzando incluso a niños, niñas y adolescentes que tienen acceso a dispositivos digitales y a internet. Según un estudio de UNICEF y UNESCO [13], más de la mitad de las chicas y chicos de entre 9 y 17 años ya utiliza herramientas de IA y de ellos, dos de cada tres lo hacen con fines escolares, como apoyo en tareas, búsqueda de información o asistencia en la resolución de problemas académicos.

El uso de la IA no es inherentemente malo ni bueno. Tal como toda tecnología, rara vez es simplemente buena o mala. Sin embargo, su uso puede tener consecuencias positivas o negativas. Ello depende de quién la cree y de cómo se utilice [14]. En este sentido, su impacto puede variar: en contextos educativos, por ejemplo, puede facilitar el aprendizaje personalizado, ayudar a resolver dudas de manera inmediata o servir como apoyo a docentes; mientras que en otros escenarios puede plantear riesgos relacionados con la privacidad, la exposición a información sesgada o la dependencia excesiva de estas herramientas digitales.

De este modo, la IA se configura como una tecnología con un enorme potencial transformador, pero que requiere un uso responsable y consciente. La manera en que se diseñen estas herramientas, los datos con los que se entrenen y la forma en que se integren en la vida de las personas serán factores determinantes para que sus consecuencias resulten beneficiosas o, por el contrario, generen nuevos desafíos sociales.

2.1.2. Propuesta de solución al problema

Un grupo de investigación de la Facultad de Psicologia de la UNSL, el cual trabaja sobre las problemáticas de adicciones especialmente en jóvenes, cuenta con el proyecto "Evaluación de la Efectividad y Aceptación de un Asistente Virtual Basado en Inteligencia Artificial para la Prevención de la Adicción al Juego Online en Adolescentes", en el que una de sus ramas, la cual trabaja sobre las problemáticas de adicciones especialmente en jóvenes, tiene como objetivo evaluar la efectividad y aceptación de un asistente virtual basado en IA como herramienta preventiva para la adicción a las apuestas online entre estudiantes y población en general. Dentro de este marco, nuestro proyecto integrador consiste en el desarrollo de dicho asistente, priorizando y asegurando la confidencialidad de los datos y el anonimato de los participantes. Se obtendrá el consentimiento informado de los participantes y de sus padres o tutores (en el caso de los menores), asegurando el respeto a los derechos de los menores y las políticas de transparencia y protección de datos personales que estipula la Ley 25326. Además, se velará por la no interferencia en el bienestar emocional de los participantes durante las sesiones con el asistente virtual.

La hipótesis del proyecto de psicología se basa en que el uso de un asistente virtual como herramienta preventiva reducirá las conductas de riesgo relacionadas con el juego online entre los estudiantes de secundaria y población general, siendo aceptado como un recurso útil y accesible por los participantes.

Para la comprobación de esta hipótesis se recolectará información estadística previa al uso del asistente, durante y posterior al uso de dicha herramienta, por lo que uno de nuestros principales objetivos es garantizar estadísticas representativas, confiables y caracterizar de la mejor manera posible a cada uno de los participantes. Esta información permitirá a los psicólogos evaluar los niveles de riesgo y la efectividad de la herramienta. Adicionalmente, se adhiere a pautas éticas estrictas para garantizar la privacidad y transparencia, informando a los usuarios que interactúan con una IA. El objetivo es construir un prototipo funcional que ofrezca respuestas coherentes relevantes, contribuyendo al desarrollo de herramientas preventivas basadas en IA para la salud mental de los jóvenes.

La solución propuesta contempla dos flujos de trabajo diferenciados, en función de los dos tipos de usuarios finales previstos. El primer flujo está orientado al uso público general: cualquier usuario podrá interactuar con el asistente sin necesidad de iniciar sesión previamente. El segundo flujo corresponde a una prueba controlada en un entorno escolar. Como se mencionó anteriormente, esta instancia requiere de una autorización previa por parte de los padres o tutores, permitiendo que el estudiante participe de la interacción con el asistente virtual. En este caso, el alumno deberá iniciar sesión utilizando credenciales proporcionadas por el experto de Psicología que esté llevando a cabo la prueba. Esta prueba estará habilitada únicamente durante el período en el que el experto se encuentre presente en la escuela. Finalizado ese lapso, se desactivará el acceso mediante esas credenciales, a fin de evitar que otros estudiantes y usuarios puedan ingresar fuera de los tiempos estipulados y generar estadísticas no válidas para el análisis.

2.1.3. Estudio de antecedentes

Se llevó a cabo un estudio sobre otras propuestas con objetivos similares a los aquí planteados. Se analizaron tres herramientas de asistentes virtuales: Yana, Wysa y Woebot. Todas las opciones son accesibles a través de aplicaciones en dispositivos móviles y permiten a los usuarios acceder a un chat con un asistente virtual para buscar apoyo emocional.

- **Yana [15]:** Yana es una inteligencia artificial diseñada para ofrecer apoyo emocional. Permite crear una sesión y conversar sobre los temas que sean necesarios, proveyendo apoyo y técnicas para afrontar los problemas. Yana es parcialmente gratuita. La aplicación ofrece una variedad de funciones básicas sin costo alguno, lo que permite a los usuarios acceder a apoyo emocional y herramientas de autoayuda de manera gratuita. Sin embargo, para aquellos que buscan una experiencia más enriquecedora y personalizada, ofrece una suscripción Premium con acceso ilimitado a sus funcionalidades.

- **Wysa [16]:** Wysa es una inteligencia artificial especializada en el autocuidado, completamente gratuita. El chatbot de IA guía a los usuarios a través de una biblioteca de herramientas y ejercicios basados en terapia cognitivo-conductual (TCC), meditación, respiración, entre otras. Facilita el acceso a un coach de bienestar emocional y facilita la interacción entre los usuarios y sus coaches asignados. Los usuarios pueden optar por permanecer anónimos durante estas sesiones de chat y el coach puede referirse a ellos con su apodo. Se trata de una charla sincrónica en vivo de treinta minutos en la que el profesional comienza por comprender el contexto del usuario y las inquietudes que lo llevaron a la sesión.

- **Woebot [17]:** Woebot es una inteligencia artificial especializada en terapia que ayuda a sus usuarios a monitorizar su estado de ánimo y a conocerse mejor. Mediante una combinación de procesamiento del lenguaje natural, redacción meticulosa, sentido del humor y experiencia psicológica, principalmente en terapia cognitivo-conductual (TCC), el asistente pregunta al usuario cómo se siente en breves conversaciones diarias de 10 minutos (máximo). Después, almacena el texto y los emojis en su totalidad, logrando que sus preguntas y respuestas se vuelvan más específicas con el tiempo, teniendo en cuenta conversaciones previas.

Todas las opciones analizadas comparten un mismo propósito, ofrecer un asistente conversacional orientado al apoyo en salud mental. Son herramientas gratuitas que permiten al usuario interactuar con una inteligencia artificial configurada con conocimientos y habilidades específicas, desarrolladas con la colaboración de especialistas en el área.

Sin embargo, no se identificó ninguna herramienta orientada específicamente al apoyo de personas con problemas de ludopatía. Si bien las opciones analizadas podrían ofrecer cierto acompañamiento y técnicas generales de control, carecen de un enfoque especializado en esta problemática. En contraste, el asistente aquí planteado se diseñó sobre bases conceptuales sólidas, desarrolladas a partir de los conocimientos provistos por el equipo de Psicología y se configuró para brindar respuestas precisas y pertinentes frente a las necesidades del usuario.

Adicionalmente, dicho equipo colaboró en la elaboración de un listado de números de contacto y líneas de emergencia especializadas, con cobertura en Argentina. Este listado es utilizado por la herramienta para dar información verificada por profesionales.

Por ultimo y más importante, ninguna de las herramientas analizadas incorpora la posibilidad de recopilar y analizar estadísticas sobre las interacciones mantenidas con los usuarios, una funcionalidad que constituye una de las características que diferencia al proyecto.

2.1.4. Metodología de Desarrollo

En esta sección hablaremos sobre la metodología de desarrollo elegida para llevar a cabo el proyecto, primero haremos mención a algunos aspectos teóricos introductorios para entender los conceptos principales y luego explicaremos nuestra adaptación llevada a cabo para el proyecto.

En el contexto de la ingeniería de software, un proceso de software se define como un conjunto estructurado de actividades interrelacionadas, cuyo propósito es desarrollar o evolucionar un sistema de software [18]. Estas actividades fundamentales, suelen abarcar la especificación del software, el diseño e implementación, la validación (pruebas) y la evolución (mantenimiento y mejora) [18].

Una metodología de desarrollo de software, por su parte, es un enfoque sistemático y estructurado que provee un marco de trabajo (framework) para llevar a cabo estas actividades. Define no solo el proceso a seguir, sino también las técnicas, herramientas, roles y artefactos que se utilizarán durante el ciclo de vida del desarrollo del software. La elección de una metodología adecuada es crucial para el éxito de un proyecto, ya que impacta directamente en la gestión, la calidad del producto final y la capacidad de adaptación a los cambios [18].

Históricamente, las metodologías de desarrollo de software se pueden clasificar en dos grandes categorías:

- **Metodologías Tradicionales (o dirigidas por planes - plan-driven):** Estos enfoques, como el modelo en cascada o el modelo V, se caracterizan por una planificación exhaustiva y detallada al inicio del proyecto. Se pone un fuerte énfasis en la definición completa y estable de los requisitos antes de pasar a las fases siguientes de diseño, implementación y pruebas. La documentación es un componente central y se espera que el proceso siga una secuencia lineal o con pocas iteraciones planificadas. Estos procesos están orientados a proyectos donde los requisitos son bien comprendidos y es improbable que cambien radicalmente, buscando la repetibilidad y la predictibilidad [18]. Su principal desventaja es la rigidez y la dificultad para gestionar cambios en los requisitos una vez que el proyecto está avanzado, lo que puede generar altos costos y retrasos.

- **Metodologías Ágiles:** En contraste, las metodologías ágiles surgieron como una respuesta a la necesidad de mayor flexibilidad y rapidez en entornos de desarrollo donde los requisitos son cambiantes o no se conocen completamente desde el inicio. Enfoques como XP, Scrum o Kanban priorizan la entrega incremental y rápida de software funcional, la colaboración continua con el cliente o los stakeholders (partes interesadas), la adaptación a los cambios y la valoración de los individuos y sus interacciones sobre los procesos y herramientas [18]. El desarrollo ágil se basa en la idea de que el software se desarrolla de forma iterativa, donde cada iteración incorpora retroalimentación y permite refinar el producto [18, 19].

Metodologías Ágiles

Las metodologías ágiles se basan en un conjunto de principios fundamentales, originalmente plasmados en el "Manifiesto por el Desarrollo Ágil de Software". Estos principios valoran a los individuos y sus interacciones por encima de los procesos y herramientas; el software funcional por encima de la documentación exhaustiva; la colaboración con el cliente por encima de la negociación contractual; y la respuesta ante el cambio por encima del seguimiento de un plan. En la práctica, esto se traduce en ciclos de desarrollo cortos llamados iteraciones o sprints, al final de los cuales se entrega un incremento de software funcional. Este enfoque permite una retroalimentación constante, una rápida adaptación a los cambios y una entrega de valor continua al cliente, lo que resulta especialmente útil en proyectos con requisitos que evolucionan con el tiempo o con alta incertidumbre.

Para el desarrollo del presente proyecto, se ha optado por una metodología ágil. Esta elección se fundamenta en varias razones, alineadas con recomendaciones para proyectos con requisitos cambiantes, incertidumbre y necesidad de evolución rápida:

- **Naturaleza Exploratoria y Requisitos Emergentes:** El desarrollo de sistemas que involucran LLMs y RAG es un campo relativamente nuevo y en rápida evolución. Es previsible que los requisitos funcionales y no funcionales, así como las expectativas de los stakeholders, no estén completamente definidos al inicio del proyecto. Una metodología ágil permite descubrir y refinar estos requisitos de manera iterativa a medida que adquirimos un mayor entendimiento del problema y de las capacidades de la tecnología.

- **Necesidad de Retroalimentación Continua:** Dada la sensibilidad del campo de la adicción al juego, es crucial obtener retroalimentación temprana y frecuente de expertos en la materia y, potencialmente, de usuarios finales en las etapas de validación. Los ciclos cortos de desarrollo facilitan la presentación de prototipos funcionales y pequeños incrementos del sistema, permitiendo validar funcionamiento, ajustar el enfoque conversacional del asistente y asegurar que el sistema realmente aporta valor y es percibido como útil, confiable y seguro para el usuario final.
- **Adaptabilidad a Cambios Tecnológicos y de Comprensión:** Tanto los LLMs como las técnicas de RAG están en constante mejora. Un enfoque ágil nos permite ser flexibles para incorporar nuevas versiones de modelos a medida que se publican, mejores prácticas vinculadas al área de la IA, realizando ajustes sin desestabilizar todo el proyecto. A medida que profundizamos en el dominio de la adicción al juego y la IA, pueden surgir nuevas necesidades que requieran cambios en la funcionalidad del asistente; la agilidad facilita la gestión de estos cambios.
- **Entrega Incremental:** El enfoque ágil permite entregar versiones funcionales del asistente de manera incremental. Podemos comenzar con funcionalidades básicas (respuestas a preguntas básicas, preguntas frecuentes, información general sobre la adicción) y luego añadir progresivamente capacidades más complejas (integración RAG para información específica y actualizada, estrategias de afrontamiento personalizadas, acompañamiento, gestión de sesiones y usuarios y estadísticas de los chats). Esto nos permite obtener retroalimentación más rápido, demostrar progreso y entregar valor al cliente de forma continua.
- **Gestión de la Incertidumbre:** Los proyectos con tecnologías en evolución, como las que se utillizaran en este proyecto, conllevan un grado de incertidumbre. Las metodologías ágiles están diseñadas para manejar esta incertidumbre, permitiéndonos aprender, inspeccionar y adaptar el proceso y el sistema regularmente, gestionando los riesgos de forma anticipada.

Programación Extrema

Dentro de las metodologías ágiles, hemos considerado que XP, combinada con los lineamientos de la Guía de la AAIP para el desarrollo de IA responsable, ofrece el marco de trabajo que mejor se adapta al proyecto. Mientras XP proporciona la estructura ágil e iterativa necesaria para el desarrollo técnico del sistema, la Guía establece los requisitos éticos, de transparencia y protección de datos que deben cumplirse en cada etapa del ciclo de vida del software.

XP, popularizada por Kent Beck en "Extreme Programming Explained: Embrace Change" [11], ofrece un conjunto de prácticas y valores beneficiosos para la dinámica y escala de este proyecto. XP lleva los principios de desarrollo iterativo y la participación del cliente a "niveles extremos". Se fundamenta en principios como la comunicación frecuente, la simplicidad en el diseño, la retroalimentación constante y el valor para realizar cambios necesarios [18, 11]. Algunas prácticas clave de XP incluyen:

- **Planificación Incremental (Planning Game):** Los requisitos se registran frecuentemente como historias de usuario y los desarrolladores estiman el esfuerzo. Se planifican entregas frecuentes (releases) e iteraciones cortas. Las historias de usuario son una descripción breve y simple de una funcionalidad desde la perspectiva del usuario final.
- **Pequeñas Entregas (Small Releases):** Se libera funcionalidad en incrementos pequeños y frecuentes, permitiendo al cliente acceder rápidamente al software y proporcionar retroalimentación.
- **Diseño Simple:** Se busca el diseño más simple que funcione para los requisitos actuales, evitando la sobreingeniería (principio YAGNI - You Ain't Gonna Need It).
- **Refactorización (Refactoring):** Se mejora continuamente la estructura interna del código sin cambiar su comportamiento externo, manteniendo el diseño simple y limpio.

- **Programación en Parejas (Pair Programming):** Dos desarrolladores trabajan juntos en una sola computadora, lo que fomenta la revisión continua del código, la transferencia de conocimiento y la producción de software de mayor calidad.
- **Propiedad Colectiva del Código (Collective Ownership):** Cualquier desarrollador puede cambiar cualquier parte del código del sistema. Esto promueve la responsabilidad compartida y evita cuellos de botella.
- **Integración Continua (Continuous Integration):** Se integran nuevas funcionalidades al sistema principal y se prueban tan pronto como se completan.
- **Ritmo Sostenible (Sustainable Pace):** Se evita el trabajo excesivo (horas extra) para mantener la productividad y la calidad a largo plazo.
- **Cliente en el Sitio (On-site Customer):** Un representante del cliente está disponible a tiempo completo para el equipo de desarrollo, para responder preguntas y tomar decisiones sobre los requisitos.

Para este proyecto, desarrollado por un equipo de dos desarrolladores, la adopción de las prácticas de XP resulta ventajosa por las siguientes razones:

- **Programación en Parejas (Pair Programming):** Con solo dos desarrolladores, esta práctica asegura que todo el código sea revisado por ambos miembros del equipo en tiempo real. Esto mejora la calidad del código y reduce defectos; también garantiza un conocimiento compartido completo de la base del código, eliminando la dependencia de un único individuo para ciertas partes del sistema. Facilita la resolución de problemas complejos y la toma de decisiones de diseño de manera colaborativa.
- **Propiedad Colectiva del Código y Diseño Simple:** En un equipo pequeño, es fundamental que ambos desarrolladores puedan trabajar en cualquier parte del sistema. La propiedad colectiva, combinada con un compromiso con el diseño simple y la refactorización constante, asegura que el asistente siga siendo comprensible, mantenible y adaptable a medida que evoluciona.
- **Integración Continua y Pequeñas Entregas:** Estas prácticas nos permiten obtener retroalimentación rápida sobre el trabajo integrado. Al entregar pequeñas porciones funcionales del asistente con frecuencia, se pueden validar las decisiones tomadas y ajustar el sistema si es necesario, sin grandes inversiones de tiempo en direcciones incorrectas.
- **Comunicación y Ritmo Sostenible:** XP fomenta una comunicación intensa, lo cual es natural y fácil de mantener en un equipo de dos desarrolladores. El énfasis en un ritmo sostenible es también necesario para evitar el agotamiento y asegurar que mantengamos un alto nivel de concentración y calidad a lo largo del proyecto.
- **Adaptación del "Cliente en el Sitio":** Si bien un cliente dedicado a tiempo completo podría no ser factible, el principio de comunicación cercana y frecuente con los stakeholders (tutores del proyecto, expertos en el área de la Psicología y en la temática de adicción) se puede adaptar fácilmente. Esto asegura que el desarrollo del asistente se mantenga alineado con los objetivos y las necesidades reales.

Guía para el desarrollo responsable de Sistemas Inteligentes

Como parte integral de nuestra metodología de desarrollo y complementando las prácticas ágiles de XP, se ha tomado como referencia la "Guía para entidades públicas y privadas en materia de Transparencia y Protección de Datos Personales para una IA responsable" [12]. Esta guía no solo orienta aspectos

específicos del diseño del sistema, sino que se integra transversalmente con el ciclo iterativo de XP, asegurando que cada entrega considere los requisitos éticos y legales aplicables.

Dado que el proyecto implica el desarrollo de un sistema de IA que tratará con temas sensibles y usuarios finales potencialmente vulnerables como adolescentes, se ha tomado como referencia la guía elaborada por la AAIP de Argentina, que tiene como objetivo ofrecer lineamientos para incorporar la transparencia y la protección de datos personales como dimensiones transversales en los proyectos de desarrollo responsable de IA.

En la guía se habla de recomendaciones de transparencia y protección de datos personales en el ciclo de vida de sistemas de IA [12]; aquí se hace referencia a un enfoque integral que se aplica a todas las etapas del proyecto, desde su concepción hasta su despliegue y mantenimiento, con el fin de abordar riesgos y posibles impactos adversos. Este ciclo se compone de cuatro etapas principales reflejadas en la **Figura 2.1** extraída de la guía [12].

1. **Diseño del sistema:** Incluye la planificación, el diseño del sistema, el análisis y la selección del modelo IA y los datos, estadísticas e inferencias que se utilizarán en el sistema.
2. **Verificación y validación:** En esta fase, se prueba el sistema diseñado para asegurar su correcto funcionamiento.
3. **Implementación:** Comprende la puesta en marcha del sistema en una infraestructura adecuada.
4. **Operación y mantenimiento:** Una vez que el sistema está operativo, se realiza un monitoreo constante de su funcionamiento y se aplican las actualizaciones necesarias.

Figura 2.1. Ciclo de vida de Sistemas de IA.

Para nuestro proyecto, se adoptarán las siguientes recomendaciones clave, estructuradas según el ciclo de vida de los sistemas de IA.

Etapa 1: Diseño del Sistema

- **Responsabilidad Proactiva:** Se adoptará un enfoque proactivo, no solo cumpliendo con la normativa, sino siendo capaces de demostrar dicho cumplimiento de manera continua y efectiva.
- **Evaluación de Impacto:** Se realizará una evaluación de impacto para identificar y minimizar desde el inicio los riesgos asociados al tratamiento de datos, como posibles sesgos algorítmicos, la reidentificación de personas y la gestión de datos personales.
- **Privacidad por Diseño y por Defecto:** Se garantizará que la privacidad sea un pilar desde la concepción del proyecto. Solo se recopilarán, procesarán y compartirán los datos personales estrictamente necesarios para los fines del asistente, con su debido consentimiento.
- **Minimización y Calidad de los Datos:** Se minimizará la cantidad de datos procesados, excluyendo totalmente los datos sensibles que no sean necesarios. Se establecerán criterios para asegurar que los datos utilizados sean precisos, íntegros y actualizados.

- **Licitud y Consentimiento:** Todo tratamiento de datos se realizará con el consentimiento libre, expreso e informado del usuario.

- **Transparencia y Explicabilidad:** Se proveerá la trazabilidad de los datos y las decisiones del sistema para poder ofrecer explicaciones claras a los usuarios sobre su funcionamiento.

Etapa 2: Verificación y Validación

- **Monitoreo y Testeo de Sesgos:** Se realizarán pruebas para evaluar el rendimiento, la seguridad y, fundamentalmente, para detectar y corregir sesgos y errores en el algoritmo y en los resultados.

- **Auditoría de Datos:** Se auditarán los conjuntos de datos utilizados junto a los expertos del dominio para corregir errores o limitaciones.

Etapa 3: Implementación

- **Política de Privacidad:** Se establecerá una política de privacidad clara y sencilla que informe a los usuarios cómo se recopilan, utilizan y protegen sus datos, así como los derechos que les asisten.

- **Seguridad y Auditabilidad:** La infraestructura tecnológica garantizará un nivel adecuado de seguridad de la información y permitirá la trazabilidad de todas las acciones para facilitar auditorías.

- **Etiquetado de Contenido:** Se notificará explícitamente a los usuarios cuando estén interactuando con un sistema de IA y los contenidos generados por el mismo serán debidamente identificados como tales.

Etapa 4: Operación y Mantenimiento

- **Canales de Comunicación y Reclamo:** Se habilitarán canales de comunicación claros para que los usuarios puedan realizar consultas, interponer reclamos y solicitar la intervención de una persona humana, utilizando un lenguaje comprensible para no expertos.

- **Ficha de Transparencia del Sistema:** Se publicará información detallada sobre el sistema, incluyendo sus objetivos, el tipo de datos que utiliza, su nivel de riesgo y los datos de contacto del responsable de la implementación.

- **Garantía de los Derechos de los Titulares:** Se implementarán procedimientos para garantizar los derechos de los usuarios sobre sus datos, incluyendo el acceso, rectificación, actualización y supresión. Se prestará especial atención al derecho a no ser objeto de decisiones basadas únicamente en el tratamiento automatizado y a solicitar la revisión humana de dichas decisiones.

La implementación de estas recomendaciones busca asegurar el desarrollo de una herramienta tecnológica que ofrezca transparencia y respete los lineamientos que conlleven a una IA responsable.

La adopción de esta metodología híbrida — XP integrado con la Guía de la AAIP — responde a la naturaleza de nuestro proyecto: por un lado, es un desarrollo de software que requiere flexibilidad y adaptabilidad; por otro, es un sistema de IA que trabaja con población potencialmente vulnerable y maneja datos sensibles, lo que exige un desarrollo responsable.

2.1.5. Tecnologías

En esta sección se describen las tecnologías utilizadas para la implementación del asistente conversacional, abarcando tanto aquellas relacionadas con la IA y el procesamiento de lenguaje natural, como las correspondientes al desarrollo de la aplicación web y de estadísticas. Las tecnologías seleccionadas cuentan con amplia adopción en la industria del software y fueron seleccionadas luego de un exhaustivo análisis. Cabe destacar que la innovación del proyecto se fundamenta en la integración de múltiples herramientas y técnicas avanzadas de IA, específicamente LLM y RAG, aplicadas al contexto específico de la prevención de adicciones a las apuestas en línea, buscando ser una herramienta socialmente valiosa.

2.1.5.1. Tecnologías relacionadas a la Inteligencia Artificial

En esta subsección se presentan las tecnologías y conceptos fundamentales de IA que constituyen el núcleo del asistente conversacional. Se describen los LLM, la arquitectura Transformer que los sustenta, las bases de datos vectoriales para almacenar embeddings, la técnica RAG para enriquecer las respuestas con conocimiento especializado, la ingeniería de contexto para optimizar las interacciones con el modelo y finalmente el proveedor de IA seleccionado. Estas tecnologías trabajan de manera integrada para permitir que el asistente comprenda consultas complejas, acceda a información especializada sobre prevención de adicciones y genere respuestas contextualmente apropiadas y fundamentadas en conocimiento validado por expertos.

Grandes Modelos de Lenguaje

En la última década, el área de Procesamiento de Lenguaje Natural ha sido revolucionada por la aparición de los Grandes Modelos de Lenguaje, conocidos como LLM por sus siglas en inglés (Large Language Models). Estos modelos han demostrado capacidades sin precedentes para comprender, generar y manipular el lenguaje humano, convirtiéndose en una tecnología fundamental para una amplia gama de aplicaciones, desde la traducción automática hasta la creación de contenido y los sistemas de diálogo avanzados.

Un LLM es un tipo de modelo de lenguaje basado en Redes Neuronales (RN) profundas que se caracteriza por su escala masiva, tanto en la cantidad de datos con los que se entrena, como en su número de parámetros. El objetivo fundamental de un LLM durante su entrenamiento es simple pero potente: predecir la siguiente palabra (o "*token*") en una secuencia de texto. A partir de esta tarea aparentemente sencilla, emergen capacidades complejas [9], tales como:

- generación de texto coherente y contextualmente relevante,
- traducción entre múltiples idiomas,
- resumen de documentos extensos,
- clasificación de texto (p. ej., análisis de sentimientos),
- respuesta a preguntas de manera conversacional.

Fundamentos: Redes Neuronales y la Arquitectura Transformer

En su núcleo, los LLM son Redes Neuronales (RN). Una RN es un modelo computacional inspirado en el cerebro humano, compuesto por capas de "neuronas" o nodos interconectados. Cada conexión tiene un "peso" (parámetro) que se ajusta durante el entrenamiento para que la red aprenda a mapear entradas (p. ej., texto) a salidas deseadas (p. ej., la siguiente palabra). Los LLM modernos contienen cientos de miles de millones de estos parámetros, lo que les confiere una enorme capacidad para aprender patrones y relaciones del lenguaje.

La arquitectura que ha hecho posibles los LLM actuales es el Transformer, introducida por Vaswani et al. (2017) en su influyente artículo “Attention Is All You Need” [20], del cual se extrae la **Figura 2.2**. A diferencia de arquitecturas anteriores que procesaban el texto de forma secuencial, el Transformer lo procesa en paralelo gracias a un punto clave: el mecanismo de auto-atención (self-attention).

- **Arquitectura Encoder-Decoder:** El modelo Transformer original se compone de un codificador (encoder), que procesa la secuencia de entrada completa y construye una representación enriquecida de la misma y un decodificador (decoder), que utiliza esa representación para generar la secuencia de salida, palabra por palabra. Como se observa en la **Figura 2.2**, el encoder (lado izquierdo) procesa la secuencia completa de entrada en paralelo, mientras que el decoder (lado derecho) genera la salida de forma secuencial, utilizando tanto la representación del encoder como las palabras ya generadas.

- **Mecanismo de Auto-Atención:** Es el corazón del Transformer, implementado mediante los bloques de “Multi-Head Attention” que se observan en la arquitectura. Permite que el modelo, al procesar una palabra, “preste atención” a todas las demás palabras de la secuencia de entrada, ponderando la importancia de cada una para comprender el contexto. Por ejemplo, al procesar la palabra “banco” en “fui al banco a sentarme”, el mecanismo de atención le daría más peso a “sentarme” que a “fui”, ayudando a desambiguar su significado (no es una institución financiera).

Bloques de Construcción: Tokenización, Embeddings y Codificación Posicional

Para que una RN pueda procesar ya sea texto, imagen, video o audio, el contenido debe ser convertido a un formato numérico universal. Este proceso implica varios pasos clave:

1. **Tokenización:** El texto de entrada se segmenta en unidades más pequeñas llamadas tokens. Un token puede ser una palabra, parte de una palabra (sub-palabra) o incluso un solo carácter. Por ejemplo, la frase “Los LLM son fascinantes” podría tokenizarse en [“Los”, “LLM”, “son”, “fasci”, “nantes”] [21].

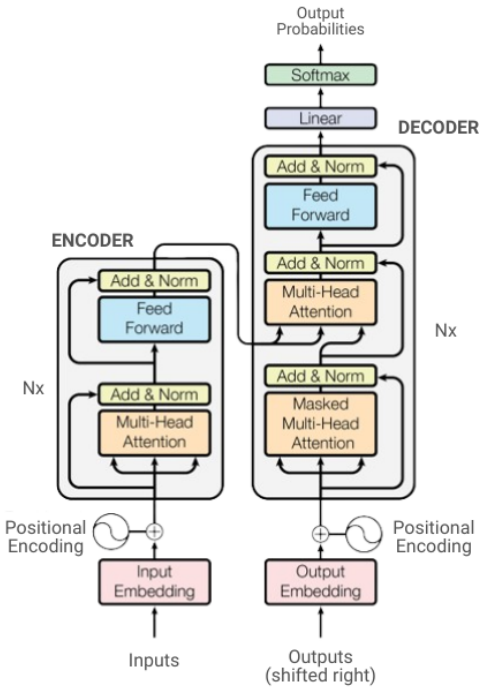

Figura 2.2. Arquitectura Transformer original de “Attention Is All You Need”.

2. **Embeddings:** Cada token se mapea a un vector numérico de alta dimensión conocido como embedding. Estos vectores no son aleatorios; se aprenden durante el entrenamiento de tal manera que tokens con significados semánticos similares tienen representaciones vectoriales cercanas en el espacio.

3. **Codificación Posicional (Positional Encoding):** Dado que el mecanismo de atención procesa todos los tokens simultáneamente, pierde la información sobre el orden de las palabras. Para solucionar esto, se añade un vector de codificación posicional al embedding de cada token, proporcionando al modelo información explícita sobre la posición de cada token en la secuencia original [20].

La **Figura 2.3**, extraída de "Build a Large Language Model (From Scratch)" [9], expresa en la izquierda la universalidad de los embeddings o vectores como representación de una entrada y a la derecha el proceso en el caso del texto. De forma resumida, en el procesamiento de la entrada, el texto se divide en tokens individuales, estos se convierten en identificadores de token mediante un vocabulario. Los identificadores de token se convierten en embeddings a los que se añaden embeddings posicionales de tamaño similar, lo que da como resultado los embeddings que se utilizan como entrada para los LLM.

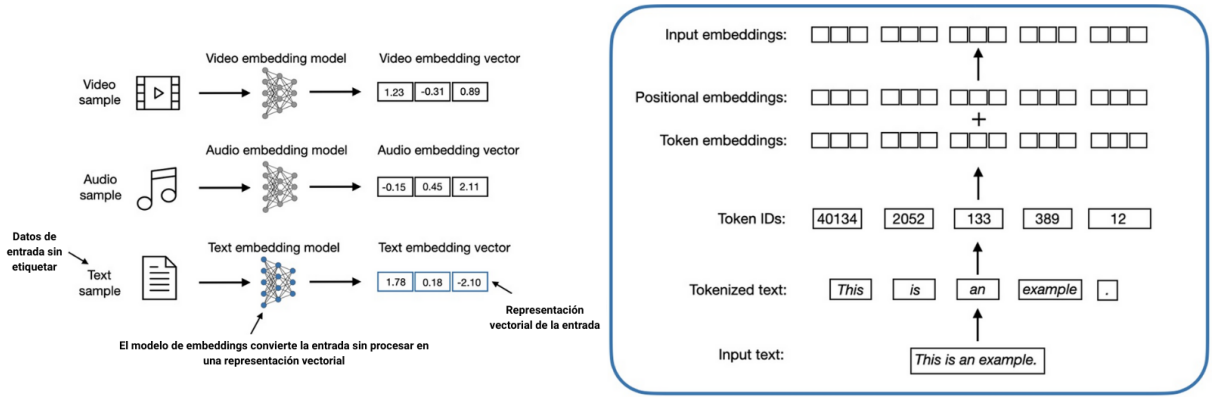

Figura 2.3. Proceso de generación de los embeddings de entrada.

El Proceso de Entrenamiento y Adaptación

El desarrollo de un LLM se realiza en varias fases:

- **Pre-entrenamiento (Pre-training):** En esta primera fase, el modelo se entrena con una cantidad masiva de texto no etiquetado extraído de internet (libros, artículos, sitios web). El objetivo es que el modelo aprenda la gramática, la sintaxis, los hechos del mundo y las relaciones semánticas del lenguaje de forma general.

- **Ajuste Fino (Fine-tuning):** Una vez pre-entrenado, el modelo base puede ser adaptado para tareas específicas. Esto se logra mediante un segundo entrenamiento, o fine-tuning, utilizando un conjunto de datos mucho más pequeño y específico para la tarea deseada (p. ej., un dataset de preguntas y respuestas para crear un chatbot).

- **Aprendizaje por Refuerzo con Retroalimentación Humana (RLHF):** Para alinear mejor las respuestas del modelo con las expectativas y valores humanos, se utiliza una técnica llamada RLHF. En este proceso, los humanos califican y comparan diferentes respuestas generadas por el modelo. Luego, se entrena un "modelo de recompensa" para predecir qué respuestas serían preferidas por los humanos y finalmente se utiliza aprendizaje por refuerzo para "premiar" al LLM por generar respuestas que maximicen esa recompensa [9].

Arquitectura solo decodificador

La arquitectura solo decodificador (decoder-only) es una simplificación de la arquitectura Transformer original, que consta de dos módulos: un codificador (encoder) y un decodificador (decoder). Como su nombre indica, esta arquitectura utiliza únicamente la parte del decodificador del Transformer, eliminando el codificador por completo. Fue popularizada por los modelos GPT (Generative Pretrained Transformer), introducidos originalmente en el artículo "Improving Language Understanding by Generative Pre-Training" en 2018 por investigadores de OpenAI [22], donde adaptaron la arquitectura original para centrarse específicamente en tareas de generación de texto. Modelos más famosos como ChatGPT, Gemini, DeepSeek, LLaMA, entre otros se basan en esta arquitectura decoder-only.

El funcionamiento de un modelo decoder-only es autorregresivo, lo que significa que genera texto prediciendo una palabra a la vez, basándose en la secuencia de palabras que la preceden. Cada nueva palabra generada se incorpora a la entrada para predecir la siguiente, mejorando así la coherencia del texto resultante [9].

Bases de datos vectoriales

Una base de datos vectorial es cualquier base de datos que permita almacenar, indexar y consultar vectores denominados "embeddings" [23].

Los embeddings son representaciones útiles de datos no estructurados, ya que asignan contenido de tal forma que la similitud semántica queda representada por la distancia en el espacio vectorial. De esta forma, es más fácil buscar similitudes, encontrar contenido pertinente en una base de conocimiento o recuperar un elemento que coincida mejor con una consulta compleja generada por un usuario.

Al igual que ocurre con otros tipos de datos, la consulta eficiente de un gran conjunto de vectores requiere un índice. Las bases de datos vectoriales admiten índices especializados para vectores. A diferencia de muchos otros tipos de datos (como texto o números), que tienen un único orden lógico, los vectores no tienen un orden natural que corresponda a casos prácticos. En su lugar, el caso práctico más habitual consiste en consultar los **k** vectores más próximos a otro vector en términos de una métrica de distancia, como el producto escalar, la similitud del coseno o la distancia euclídea. Este tipo de consulta se conoce como "los k vecinos más cercanos (exactamente)" o "KNN".

No existen algoritmos generales para consultas KNN eficientes. Para poder garantizar la búsqueda de los **k** vecinos más cercanos a un vector determinado **q**, es necesario calcular la distancia entre **q** y todos los demás vectores. Sin embargo, hay algoritmos eficientes para encontrar los **k** vecinos más cercanos aproximadamente ("ANN"). Estos algoritmos de ANN sacrifican algo de precisión (en concreto, recuperación, ya que el algoritmo puede omitir algunos de los vecinos más cercanos) para obtener grandes mejoras en la velocidad. Dado que en muchos casos de uso ya se trata el proceso de cálculo de embeddings como algo impreciso, a menudo pueden tolerar cierta pérdida de memoria a cambio de grandes mejoras en el rendimiento.

Un simple ejemplo para entender de mejor manera cómo funcionan las bases de datos vectoriales se refleja en la **Figura 2.4**. En las bases de datos vectoriales, los vectores que son semánticamente parecidos están cerca unos de otros. Por ejemplo, la palabra "Gato" y "Perro" estarán cerca porque los dos son animales, mientras que podemos ver que en otro sector se encuentran "Manzana" y "Banana" ya que semánticamente se reconoce que son frutas.

Cuando se realiza una búsqueda, como podría ser "Gatito", la base de datos no busca coincidencias exactas, sino los vectores más cercanos al de "Gatito". Así devuelve palabras similares dentro de su base de datos vectorial, como "Gato", "Perro" o "Lobo".

Figura 2.4. Ejemplo de una Base de Datos Vectorial.

Generación aumentada por recuperación

La aplicación de LLMs en contextos sensibles —como la prevención de adicciones conductuales— plantea desafíos éticos y legales vinculados a la trazabilidad, el manejo de datos personales, la transparencia algorítmica y la seguridad de la información [24, 25, 26, 27]. Además, los LLM tienen un conocimiento interno que a menudo es fijo, o se actualiza con poca frecuencia debido al alto costo de entrenamiento. Característica que limita su capacidad para responder preguntas sobre eventos actuales o proporcionar conocimiento específico de un dominio, lo que vuelve problemático resolver tareas que requieren gran especialización. Así, suelen generar respuestas incorrectas o alucinaciones cuando se enfrentan a consultas que van más allá de sus datos de entrenamiento o que demandan información actualizada.

En este sentido, la técnica de Recuperación Aumentada por Generación se presenta como una alternativa que permite enriquecer las respuestas generadas por el LLM con información proveniente de bases de conocimiento externas, mejorando la precisión de los resultados. Esta técnica fue propuesta en la investigación de Patrick Lewis y colaboradores, titulada “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks” [10].

El proceso de optimización se basa en referenciar a una base de conocimientos autorizada fuera de los orígenes de datos de entrenamiento, antes de generar una respuesta. De forma que la salida de un modelo de lenguaje de gran tamaño sea mejorada [28].

El uso de RAG ofrece varias ventajas:

- **Información actualizada:** RAG puede acceder y utilizar los datos más recientes, manteniendo las respuestas al día.
- **Experiencia en dominios específicos:** Con bases de conocimiento específicas de cada dominio, RAG puede proporcionar respuestas en áreas concretas.
- **Reducción de alucinaciones:** Basar las respuestas en hechos recuperados ayuda a minimizar la información falsa o inventada.

Para comprender mejor cómo funciona RAG, es importante analizar detalladamente cada una de sus etapas de procesamiento, que se muestran gráficamente en la **Figura 2.5**, basada en el paper “Retrieval-Augmented Generation for Large Language Models: A Survey” [28].

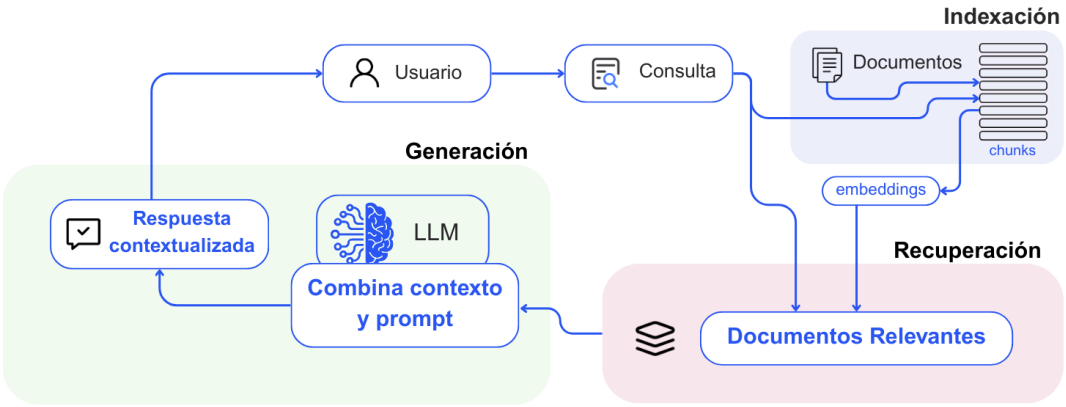

Figura 2.5. Etapas del proceso RAG.

- **Indexación:** El proceso de indexación comienza con la limpieza y extracción de datos en bruto en diversos formatos como PDF, HTML, Word y Markdown, los cuales luego se convierten a un formato de texto plano uniforme. Para adaptarse a las limitaciones de contexto de los modelos de lenguaje, el texto se segmenta en fragmentos más pequeños y manejables denominados "chunks". Estos fragmentos se codifican en representaciones vectoriales mediante un modelo de embedding y se almacenan en una base de datos vectorial. Este paso es crucial para permitir búsquedas de similitud eficientes en la fase de recuperación. El objetivo de la fragmentación es generar unidades de recuperación: partes del texto que puedan ser devueltas como resultados cuando se hace una búsqueda. Un documento completo puede ser demasiado extenso para esto, por lo que es necesario dividirlo en partes más pequeñas. La estrategia de fragmentación más común es la de tamaño fijo, que divide textos largos en secuencias con una cantidad constante de tokens [29].

- **Recuperación:** Al recibir una consulta de un usuario, el sistema RAG emplea el mismo modelo de codificación utilizado durante la fase de indexación para transformar la consulta en una representación vectorial. Luego calcula los puntajes de similitud entre el vector de la consulta y los vectores de los fragmentos dentro del corpus indexado. El sistema prioriza y recupera los K fragmentos que muestran mayor similitud con la consulta. Estos fragmentos se utilizan posteriormente como contexto ampliado en el prompt.

- **Generación:** La consulta planteada y los documentos seleccionados se sintetizan en un prompt coherente, para el cual el modelo de lenguaje de gran tamaño debe generar una respuesta. La manera en que el modelo responde puede variar según criterios específicos de la tarea, permitiéndole tanto recurrir a su conocimiento paramétrico inherente como limitar sus respuestas a la información contenida en los documentos proporcionados. En casos de diálogos en curso, cualquier historial conversacional existente puede integrarse al prompt, permitiendo al modelo participar eficazmente en interacciones de diálogo de múltiples turnos.

El rápido avance y adopción de RAG en PLN ha hecho que la evaluación de estos modelos sea un tema central en la investigación de LLMs. El objetivo principal es entender y optimizar el desempeño de los modelos RAG en distintos escenarios de aplicación, presentando los conjuntos de datos y métodos de evaluación más relevantes [28].

Tradicionalmente, la evaluación de modelos RAG se centra en su desempeño en tareas específicas, usando métricas como:

- **Control de Calidad:** EM (Exact Match) [30] y F1, que miden precisión exacta y solapamiento con la respuesta correcta.

- **Verificación de hechos:** Exactitud (Accuracy), que indica el porcentaje de respuestas correctas.
- **Calidad de respuestas:** BLEU y ROUGE, métricas que comparan el texto generado con referencias de alta calidad. BLEU mide la coincidencia de n palabras consecutivas entre el texto generado y el de referencia, mientras que ROUGE se centra en el grado de recuperación de contenido relevante.

Los objetivos principales en los que se basa la evaluación de un RAG son:

- **Calidad de recuperación:** Evalúa la efectividad del contexto obtenido por el componente de recuperación. Se usan métricas de sistemas de búsqueda y recomendación, que indican qué tan relevantes son los fragmentos recuperados y su orden.
- **Calidad de generación:** Mide la capacidad del generador para producir respuestas coherentes y relevantes a partir del contexto recuperado.

Ingeniería de contexto

La ingeniería de contexto (Context Engineering) es una disciplina formal que se enfoca en la optimización sistemática de la información que se proporciona a los LLM, durante su inferencia. Va más allá del simple diseño de instrucciones (prompts) para abarcar un enfoque más amplio y científico sobre cómo se diseña, gestiona y optimiza la información que reciben estos modelos. El rendimiento de los LLMs está fundamentalmente determinado por el contexto que reciben, por esto, la ingeniería de contexto trata esto no como una simple cadena de texto estática, sino como un conjunto dinámico y estructurado de componentes de información [31].

La ingeniería de contexto adopta un enfoque sistémico y dinámico que gestiona ecosistemas de información en evolución, orquestando múltiples componentes, herramientas y estados para mantener coherencia y relevancia en interacciones prolongadas [32].

Componentes

Algunos de los componentes que podemos mencionar e incluir en la ingeniería de contexto son:

- **Instrucciones y reglas del sistema (system prompt):** Directivas que guían el comportamiento del modelo a nivel general.
- **Conocimiento externo recuperado (RAG):** Información adicional obtenida de fuentes externas, como bases de datos o documentos, que aumenta la entrada del usuario agregando datos recuperados relevantes en contexto.
- **Herramientas externas:** Mecanismos de llamada a funciones, que les permiten utilizar herramientas externas como calculadoras, motores de búsqueda o APIs especializadas para obtener información adicional o realizar tareas específicas.
- **Razonamiento Integrado:** Como por ejemplo cadenas de pensamiento o Chain-of-Thought, donde se descompone problemas complejos en pasos de razonamiento intermedios, guiando al modelo a través de un proceso lógico para llegar a una solución.
- **Persistencia y Gestión de Memoria:** Para sistemas conversacionales que requieren mantener coherencia a través de múltiples interacciones. Permite que el sistema preserve el estado de la conversación, las preferencias del usuario y el contexto histórico relevante, creando una experiencia más personalizada y contextualmente apropiada.

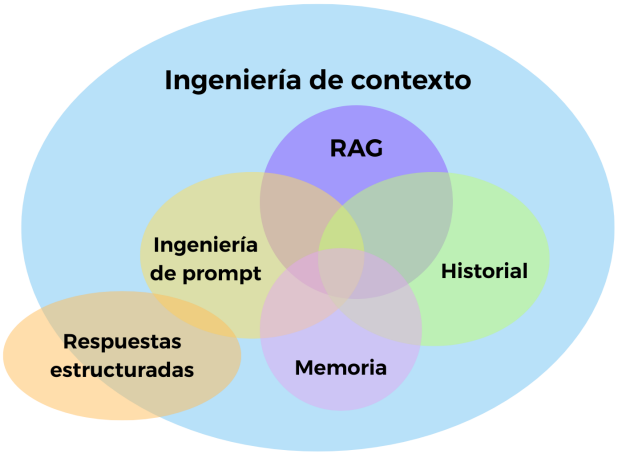

Figura 2.6. Ingeniería de contexto.

- **Estado dinámico:** Información actual sobre el usuario, el mundo o un sistema multiagente. En nuestro proyecto podemos mencionar el nivel de riesgo del usuario, o datos contextuales como la hora y el día, la ubicación, etc.
- **Consulta inmediata:** La solicitud actual o mensaje del usuario.
- **Formato de salida:** Instrucciones específicas sobre cómo debe estructurarse la respuesta del modelo, incluyendo el estilo, el tono y la longitud.

Algunos de estos componentes se ilustran en la **Figura 2.6** extraída de "Context Engineering: A Guide With Examples" [32]. La figura muestra de forma similar a un diagrama de Venn cómo la ingeniería de contexto integra múltiples herramientas y componentes para optimizar la interacción con los LLMs.

En nuestro proyecto se implementan y orquestan varios componentes de la ingeniería de contexto como el "system prompt"; el conocimiento externo recuperado mediante RAG como una herramienta con mecanismo de llamada por el LLM; el razonamiento integrado; la persistencia y gestión de memoria de la conversación; el estado dinámico con variables propias obtenidas del usuario como su nivel de riesgo, edad, ubicación, etc.; la consulta inmediata hecha por el usuario; y en el caso tanto del asistente conversacional como del sistema de estadísticas, usamos formatos de salidas específicos y estructurados.

Etapas

La ingeniería de contexto se puede dividir en varias etapas clave:

- **Análisis del contexto:** Comprender el contexto actual del usuario, incluyendo su historial, preferencias y la situación específica en la que se encuentra.
- **Diseño de la interacción:** Planificar cómo se llevará a cabo la interacción, incluyendo la selección de herramientas, la estructura de la conversación y los posibles flujos de trabajo.
- **Implementación:** Desarrollar y poner en práctica la solución diseñada, integrando todos los componentes necesarios.
- **Evaluación y ajuste:** Monitorear el rendimiento de la solución en el mundo real, recopilando datos y retroalimentando el sistema para realizar ajustes y mejoras continuas.

La ingeniería de contexto representa un cambio de paradigma en la interacción con los modelos de lenguaje: deja de tratarlos como simples sistemas reactivos y los convierte en plataformas inteligentes

capaces de adaptarse a entornos complejos, dinámicos y centrados en el usuario. Su relevancia radica en que habilita aplicaciones más confiables, escalables y personalizadas, lo que la consolida como un componente esencial en el diseño de soluciones basadas en IA avanzada.

Google generative AI Provider

Google Generative AI Provider [33] es una interfaz que permite integrar modelos de IA generativa de Google en aplicaciones desarrolladas con el Software Development Kit (Kit de Desarrollo de Software) (SDK). El SDK de IA generativa de Google proporciona una interfaz unificada para los modelos de Gemini 2.5 y 2.0 a través de la Application Programming Interfaces (Interfaz de Programación de Aplicaciones) (API) de Gemini Developer y la API de Gemini en Vertex AI [34].

Para la selección del modelo y proveedor, utilizamos como referencia Artificial Analysis [35], una plataforma independiente dedicada a la comparación y análisis de modelos de inteligencia artificial y proveedores de APIs. Realizan benchmarks independientes que evalúan métricas clave de rendimiento como calidad, precio, velocidad de respuesta, entre otras, lo que permite a desarrolladores elegir el modelo y proveedor de API más adecuado para sus necesidades específicas. Ofrece un índice de inteligencia propio y métricas detalladas que ayudan a tomar decisiones informadas en un ecosistema de IA cada vez más complejo y en constante evolución.

En el análisis del sistema que debemos construir, un asistente conversacional para la prevención de juego problemático y adicción (adicciones conductuales), se identificaron varios requisitos clave que guían la selección del modelo de IA más adecuado:

- **Inteligencia:** El modelo debe ser capaz de comprender y responder a una amplia gama de consultas relacionadas con la prevención de adicciones conductuales, demostrando un alto nivel de comprensión del lenguaje natural y conocimiento en el dominio.
- **Velocidad:** Dado que las interacciones serán en tiempo real y específicamente construido para el público joven y adolescente, el modelo debe ser capaz de generar respuestas rápidamente para mantener una conversación fluida y no perder el foco de atención del usuario.
- **Costo:** Considerando que el proyecto tiene limitaciones de presupuesto, es crucial seleccionar un modelo que ofrezca un equilibrio adecuado entre costo y rendimiento, asegurando la viabilidad económica del sistema a largo plazo.

Después de evaluar varios modelos y proveedores utilizando los benchmarks proporcionados por Artificial Analysis, se seleccionó el modelo **Gemini 2.5 Flash de Google con razonamiento integrado** [36], como se mencionó anteriormente en la ingeniería de contexto, la importancia del razonamiento, como la opción más adecuada para nuestro proyecto. A continuación se describen las razones de esta elección con las métricas y benchmarks analizados.

Iremos describiendo cada una de las métricas analizadas y se incluyen las figuras correspondientes obtenidas de Artificial Analysis al día 19 de agosto de 2025.

La **Figura 2.7** muestra el índice de inteligencia de varios modelos de IA creado por Artificial Analysis. Este índice de inteligencia se calcula como un promedio ponderado de las evaluaciones constituyentes, equilibrando razonamiento general y conocimiento, razonamiento matemático, generación de código, seguimiento de instrucciones y razonamiento sobre contexto largo. Encontramos al modelo Gemini 2.5 Flash de Google en la mitad de la figura, con un índice de inteligencia de 58 puntos, destacando que este índice solo muestra los mejores modelos dentro del amplio espectro actual.

La **Figura 2.8** presenta la velocidad de respuesta de varios modelos de IA, medida en tokens por segundo. La velocidad de respuesta es crucial para aplicaciones en tiempo real, ya que afecta directamente la experiencia del usuario y en el caso puntual de nuestra aplicación, los usuarios jóvenes tienen una tendencia a un periodo de atención más corto debida a la sobrecarga de estímulos digitales, entre

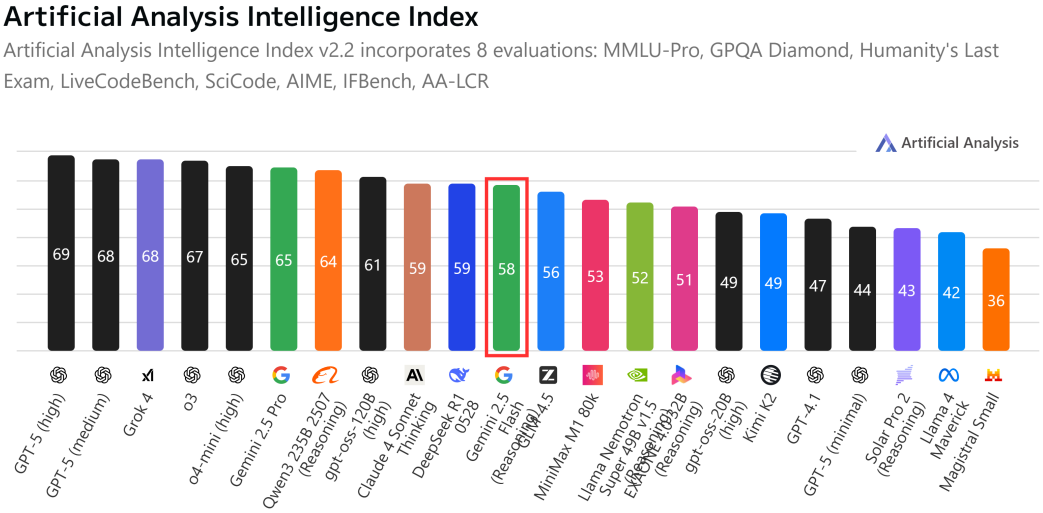

Figura 2.7. Índice de Inteligencia de Artificial Analysis.

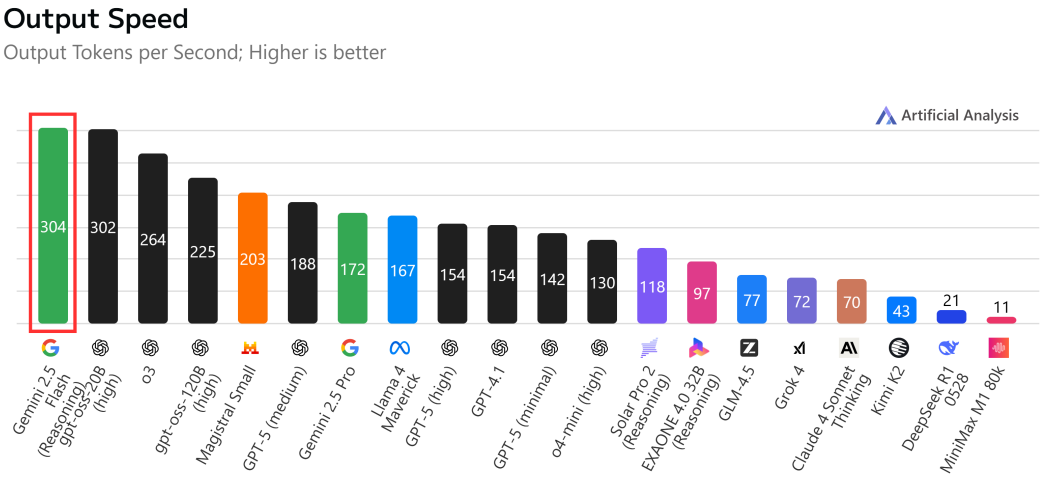

Figura 2.8. Velocidad de Respuesta.

otros factores. En este gráfico, el modelo Gemini 2.5 Flash con razonamiento destaca por su rapidez, encontrándose primero en la figura y ofreciendo los mejores tiempos de respuesta.

La **Figura 2.9** ilustra en un sistema de coordenadas cartesianas la relación entre el índice de inteligencia y la velocidad de respuesta de varios modelos de IA. En el eje de las **X** se representa la velocidad de respuesta en tokens por segundo, mientras que en el eje de las **Y** se muestra el índice de inteligencia. Se busca que el modelo se encuentre en el cuadrante superior derecho de la figura pintado en color verde, ya que esto indica un alto nivel de inteligencia junto con una alta velocidad de respuesta. El modelo Gemini 2.5 Flash de Google se posiciona favorablemente en este gráfico dentro del mencionado cuadrante deseado, ofreciendo un equilibrio óptimo entre inteligencia y rapidez, lo que lo convierte en una opción ideal para aplicaciones que requieren respuestas rápidas y precisas.

Por último, la **Figura 2.10** ilustra en un sistema de coordenadas cartesianas el costo por cada 1,000 tokens procesados por varios modelos de IA. En el eje de las **X** se representa el costo en dólares decreciente, mientras que en el eje de las **Y** se muestra el índice de inteligencia. Se busca que el modelo se encuentre en el cuadrante superior derecho de la figura pintado en color verde, ya que esto indica

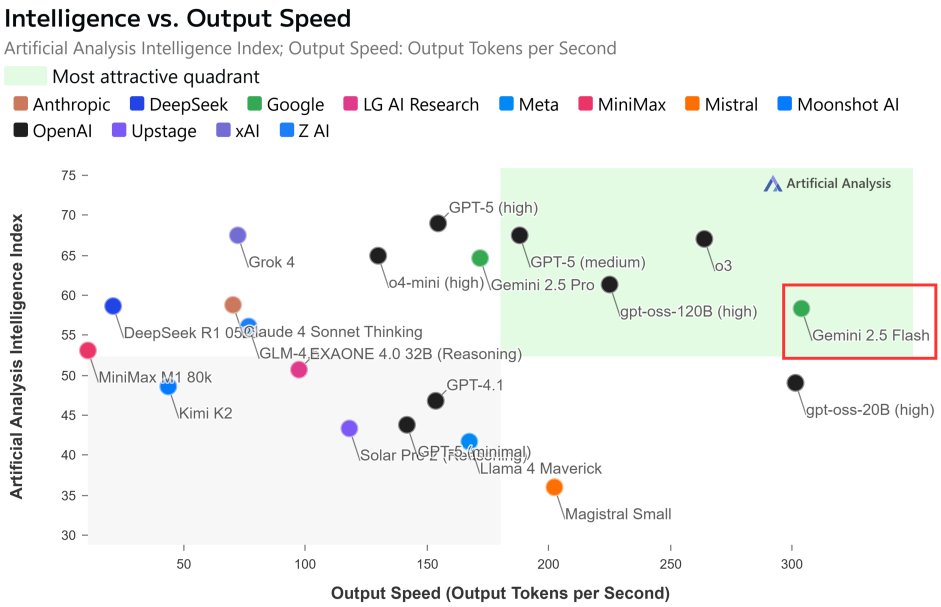

Figura 2.9. Inteligencia vs Velocidad de Respuesta.

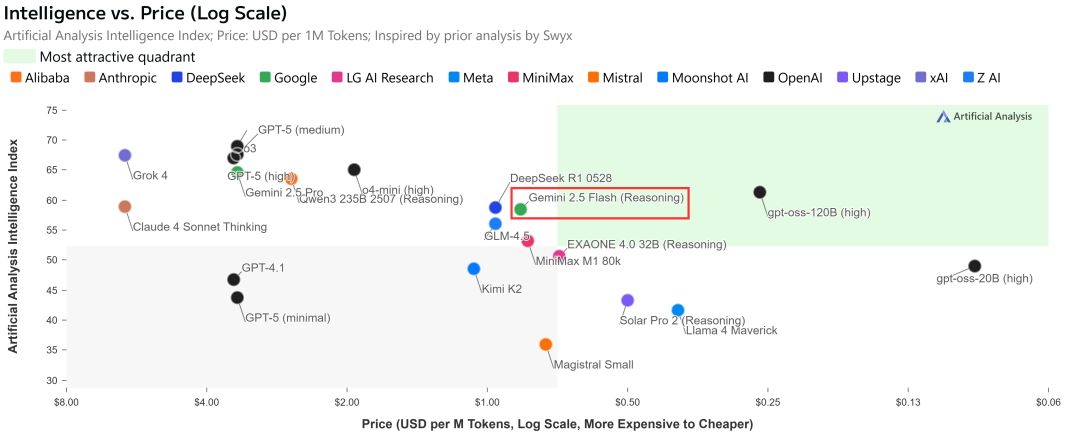

Figura 2.10. Inteligencia vs Precio.

un alto nivel de inteligencia a un costo bajo. El costo es un factor crítico en la selección de un modelo, especialmente para proyectos con restricciones presupuestarias. En este gráfico, el modelo Gemini 2.5 Flash de Google se posiciona muy cercano al cuadrante deseado, ofreciendo un equilibrio entre costo y rendimiento, solamente superado por el modelo GPT-OSS-120B el cual no posee actualmente un proveedor oficial con bajo costo, sino que debe ejecutarse de manera local o alquilando servidores lo que no conviene en términos de costos, este análisis se profundiza en la **subsección 2.2.2**.

2.1.5.2. Tecnologías del Desarrollo

En esta subsección se describen las tecnologías y herramientas utilizadas para la construcción de la aplicación web que aloja al asistente conversacional y la API de estadísticas. Se abordan tanto tecnologías de frontend que conforman la interfaz de usuario, como de backend que gestionan la lógica de negocio y la comunicación entre componentes. También se incluyen las herramientas para la gestión de bases de datos, autenticación, librerías de análisis de sentimientos, integración con modelos de IA, la plataforma de despliegue en la nube y despliegue mediante contenedores. Esta infraestructura tecnológica proporciona el soporte necesario para que el asistente funcione de manera eficiente, segura y escalable, posibilitando una buena experiencia de usuario y la correcta gestión de datos.

React

React es una biblioteca de JavaScript para el desarrollo front-end y es una herramienta para construir componentes de interfaz de usuario. React fue desarrollado por el ingeniero de software de Facebook Jordan Walke [37].

React funciona como un framework que integra y potencia las tecnologías fundamentales del desarrollo web: HTML, CSS y JavaScript. Proporciona una estructura organizada y metodologías específicas para trabajar con estas tecnologías de manera más eficiente, permitiendo crear aplicaciones web interactivas y dinámicas mediante un enfoque basado en componentes reutilizables.

En lugar de manipular directamente el Modelo de Objetos del Documento (DOM, por sus siglas en inglés) del navegador, que es la estructura de objetos generada a partir del HTML, React trabaja con un DOM virtual en memoria. Allí realiza todas las actualizaciones y comparaciones necesarias. Después detecta exactamente qué partes cambiaron y aplica al DOM real únicamente esos cambios puntuales, evitando operaciones innecesarias.

Vercel

Vercel fue fundada en 2015 por Guillermo Rauch, un desarrollador argentino conocido por haber creado bibliotecas populares como Socket.IO. La empresa originalmente se llamaba ZEIT y cambió su nombre a Vercel en abril de 2020. Desde entonces, ha crecido significativamente hasta convertirse en una de las plataformas líderes para el despliegue de aplicaciones web modernas [38].

Es una plataforma en la nube especializada en el desarrollo, despliegue y escalado de aplicaciones web modernas. La empresa proporciona herramientas para desarrolladores e infraestructura cloud optimizada que permite construir experiencias web más rápidas y personalizadas, enfocándose en la optimización del rendimiento, la escalabilidad automática y una experiencia de desarrollo ágil. Vercel es especialmente conocida por ser la creadora y principal mantenedora del framework Next.js, uno de los frameworks más populares para React.

Next.js

Next.js es un framework para crear aplicaciones web full-stack de código abierto construido sobre React, creado por la empresa Vercel. Este framework surge como una solución para abordar las limitaciones del desarrollo tradicional con React, proporcionando una estructura más robusta y optimizada para la creación de aplicaciones web modernas. Se ha posicionado como una de las herramientas más populares en el ecosistema de JavaScript, siendo adoptado por empresas de gran envergadura como Netflix, TikTok y Twitch para el desarrollo de sus plataformas web [39].

Una de las principales ventajas es su capacidad de renderizado híbrido, que combina la generación estática de sitios (Static Site Generation - SSG) con el renderizado del lado del servidor (Server-Side Rendering - SSR), permitiendo a los desarrolladores elegir la estrategia más adecuada para cada página de su aplicación. Además, incluye características avanzadas como la optimización automática de imágenes, el code splitting (división de código) inteligente, el enrutamiento basado en el sistema de archivos y el soporte nativo para TypeScript, el lenguaje base elegido en el proyecto. Estas funcionalidades contribuyen a mejorar el rendimiento de las aplicaciones, la experiencia del usuario y los tiempos de carga de las páginas.

El framework también destaca por su enfoque en la optimización para motores de búsqueda y la experiencia del desarrollador, ofreciendo múltiples herramientas integradas como API Routes para crear endpoints de backend, un sistema de despliegue simplificado, entre otras herramientas. Permite concentrarse en la lógica de negocio en lugar de la configuración del entorno, mientras mantiene la flexibilidad para personalizaciones avanzadas cuando sea necesario.

Shadcn

Shadcn es una herramienta con un conjunto de componentes para front-end accesibles y atractivos a la vista. Esta herramienta permite el acceso al código directamente para que se pueda modificar y adaptar según el diseño que se necesite [40].

Todos los componentes siguen una interfaz común para que se pueda componer de manera predecible y el sistema de distribución permite que se puedan compartir e instalar en distintos proyectos. Además, incluye estilos predeterminados pensados para que se puedan usar de inmediato y está diseñado para que se pueda integrar fácilmente con herramientas de IA.

Auth.js

Auth.js es una biblioteca de autenticación completa y flexible para aplicaciones web modernas que proporciona soluciones de autenticación seguras y escalables para diversos frameworks de JavaScript y Typescript. Soporta cuatro métodos principales de autenticación: autenticación OAuth (como Google, GitHub, LinkedIn), enlaces mágicos a través de proveedores de email, credenciales tradicionales (usuario y contraseña, método elegido para el proyecto) y puede integrarse opcionalmente con bases de datos externas mediante adaptadores. Auth.js puede integrarse con cualquier capa de datos (base de datos, ORM, o API backend) para crear usuarios, manejar vinculación de cuentas, soportar inicio de sesión sin contraseña y almacenar información de sesión. La biblioteca incluye adaptadores nativos para ORMs populares como Drizzle ORM, Prisma, entre otros, facilitando la integración con bases de datos PostgreSQL, MySQL, SQLite, etc [41].

La arquitectura se centra en la simplicidad de implementación y la seguridad robusta, permitiendo integrar sistemas de autenticación con configuraciones mínimas. Para la gestión de sesiones, ofrece múltiples estrategias, incluyendo el uso de JSON Web Tokens (JWT) como una opción flexible y seleccionada para nuestro proyecto. Los JWT son medios compactos y seguros para URLs que representan reclamaciones a ser transferidas entre dos partes, codificadas como objetos JSON firmados digitalmente.

Los JWT son tokens autocontenidos y sin estado que llevan toda la información necesaria para autenticación y autorización dentro del token mismo, eliminando la necesidad de que los servidores mantengan datos de sesión. El funcionamiento de JWT implica que cuando el usuario se autentica correctamente, el servidor genera un JWT usando una clave secreta, lo envía al cliente y para solicitudes posteriores, el cliente envía el JWT junto con la petición para que el servidor verifique el token antes de otorgar acceso, convirtiéndose en una opción ideal para aplicaciones distribuidas que requieren escalabilidad y autenticación sin estado [42].

Python

Python es un lenguaje de programación muy popular. Fue creado por Guido van Rossum y lanzado en 1991. Se utiliza en distintos ámbitos como el desarrollo web del lado del servidor, el desarrollo de software, las matemáticas y la automatización de sistemas mediante scripts [43].

Con Python se pueden crear aplicaciones web en un servidor, integrar flujos de trabajo junto a otros programas, conectar con sistemas de bases de datos, leer y modificar archivos, trabajar con grandes volúmenes de datos y realizar cálculos matemáticos complejos. También puede usarse tanto para hacer prototipos de manera rápida como para desarrollar software listo para producción.

Python funciona en diferentes plataformas, como Windows, Mac, Linux o Raspberry Pi. Su sintaxis es simple y se asemeja al lenguaje inglés, lo que permite escribir programas con menos líneas que en otros lenguajes. Además, se ejecuta en un sistema interpretado, lo que significa que el código puede probarse de inmediato, facilitando así el prototipado.

Pysentimiento

Pysentimiento es una librería basada en Transformers pensada para tareas de procesamiento del lenguaje natural enfocadas específicamente en el lenguaje de redes sociales. Es open-source y permite analizar opiniones y otros aspectos del lenguaje en varios idiomas, especialmente español [44].

Incluye tareas como análisis de sentimiento, donde el modelo identifica si un texto transmite una actitud positiva, negativa o neutra; y análisis de emociones, que permite detectar qué emoción concreta aparece en lo que la persona escribió (alegría, tristeza, enojo, etc.). También ofrece detección de hate speech, ironía y otras herramientas útiles. Todo esto usando modelos Transformers ya entrenados y listos para usar.

Además, presenta una evaluación completa del desempeño de varios modelos de lenguaje preentrenados en diferentes tareas, idiomas y datasets, incluyendo un análisis sobre la equidad de los resultados, lo que permite entender mejor el rendimiento y las limitaciones de cada modelo en contextos reales.

Postgres y pgvector

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional (ORDBMS) basado en POSTGRES, Versión 4.2, desarrollado en el Departamento de Ciencias de la Computación de la Universidad de California en Berkeley. POSTGRES introdujo muchos conceptos que solo estuvieron disponibles en algunos sistemas comerciales de bases de datos mucho tiempo después [45].

PostgreSQL es un descendiente de código abierto de este proyecto original de Berkeley. Es compatible con gran parte del estándar SQL y ofrece muchas funcionalidades modernas, como consultas complejas, claves foráneas, triggers, vistas actualizables, integridad transaccional y control de concurrencia multiversión.

Pgvector es una extensión de PostgreSQL que permite realizar búsqueda de similitud de vectores, almacenar vectores junto con el resto de los datos y ofrece soporte para búsqueda de vecinos más cercanos, tanto exacta como aproximada [46].

Admite vectores en precisión simple, media, binarios y dispersos, así como distintas métricas de distancia.

NeonDB

NeonDB es una plataforma de base de datos PostgreSQL serverless diseñada para el desarrollo de aplicaciones modernas. Se caracteriza por su arquitectura separada de almacenamiento y computación, que permite escalar automáticamente los recursos según la demanda y reducir costos al pausar la base de datos cuando no está en uso. Esta característica resulta especialmente valiosa para el proyecto, ya que ofrece una capa gratuita generosa, suficiente para el desarrollo del proyecto. Además, NeonDB proporciona soporte nativo para pgvector, lo que permite almacenar y consultar eficientemente los vectores de embeddings necesarios para la implementación del RAG, sin requerir configuraciones complejas ni infraestructura adicional. La combinación de su modelo serverless, el rendimiento, la compatibilidad total con PostgreSQL y su capa gratuita son los puntos por los cuales se eligió [47].

Drizzle ORM

Un ORM (Object-Relational Mapping) permite trabajar con bases de datos relacionales mediante objetos en lugar de SQL directo. Drizzle ORM destaca por ser ligero, sin dependencias, basado en TypeScript y con un enfoque SQL-like que reduce la curva de aprendizaje para desarrolladores familiarizados con SQL. Ofrece compatibilidad nativa con PostgreSQL, un diseño optimizado para entornos serverless y genera una sola consulta por operación, lo que lo hace ideal para aplicaciones escalables y de alto rendimiento [48].

Interfaz de programación de aplicaciones

Las APIs (interfaces de programación de aplicaciones) son un software que permite que dos aplicaciones se comuniquen entre sí. Son componentes fundamentales de la tecnología moderna y de la infraestructura empresarial. Las APIs permiten la conexión y el intercambio de información entre aplicaciones y servidores de manera directa [49].

AI-SDK

El AI SDK es un conjunto de herramientas en TypeScript, diseñado para ayudar a los desarrolladores a crear aplicaciones y agentes con IA usando React, Next.js, Node.js y más. Estandariza la integración de modelos de IA entre los proveedores soportados, evitando la necesidad de lidiar con problemas técnicos al integrarlos manualmente [50].

Uno de los aspectos fundamentales por los que se eligió este SDK es su facilidad para integrar modelos, simplicidad, comunidad y documentación, pero además porque utiliza streaming de datos. El streaming (transmisión) de datos de salida, tal como lo describe la documentación de Vercel AI SDK en el apartado "fundamentos", es un método de entrega de respuestas desde un LLM al usuario. A diferencia de una interfaz de bloqueo, que requiere que el LLM genere la respuesta completa antes de mostrar algo, obligando al usuario a esperar frente a un indicador de carga, una interfaz de streaming envía la respuesta en pequeños fragmentos (tokens) a medida que se generan.

La ventaja fundamental de este enfoque es la mejora de la experiencia de usuario y la reducción de la latencia percibida. Aunque el tiempo total que tarda el modelo en pensar la respuesta completa es el mismo, el usuario comienza a recibir retroalimentación visual casi instantáneamente. Esto hace que la aplicación se sienta mucho más rápida, interactiva y receptiva, similar a la experiencia de ver las palabras aparecer una por una en interfaces conversacionales conocidas.

Este método es crucial para las aplicaciones de IA conversacionales modernas, al transmitir la respuesta token por token, se evita la espera y se proporciona respuesta casi de inmediato al usuario, confirmando que el sistema está procesando activamente el mensaje.

Docker

Docker es una plataforma abierta para desarrollar, enviar y ejecutar aplicaciones. Docker permite separar aplicaciones de la infraestructura para que se pueda entregar software rápidamente. Permite gestionar la infraestructura de la misma manera que se gestionan las aplicaciones [51].

Docker proporciona la capacidad de empaquetar y ejecutar una aplicación en un entorno ligeramente aislado llamado contenedor. El aislamiento y la seguridad permiten ejecutar muchos contenedores simultáneamente en un mismo host. Los contenedores son livianos y contienen todo lo necesario para ejecutar la aplicación, por lo que no se necesita depender de lo que esté instalado en el host. Docker ofrece políticas de reinicio para controlar si los contenedores se inician automáticamente cuando se detienen o cuando Docker se reinicia. Las políticas de reinicio permiten que los contenedores vinculados se inicien en el orden correcto.

2.2. Planificación y Costos del Proyecto

2.2.1. Planificación

La planificación del proyecto se fundamenta en los principios de XP descritos en la sección anterior, adoptando un enfoque iterativo e incremental que permite la entrega de funcionalidad de manera progresiva y la adaptación continua a los cambios. El desarrollo se estructura en cuatro etapas principales que

abarcan desde la investigación inicial hasta la capacitación del cliente y las pruebas controladas o test de aceptación.

Organización del equipo y prácticas de XP

Si bien para facilitar la organización y el seguimiento del proyecto se han definido áreas de responsabilidad principal para cada integrante del equipo, es fundamental destacar que el desarrollo se llevó a cabo siguiendo las prácticas de XP, especialmente la **programación en parejas** y la **propiedad colectiva del código**. Esto significa que, aunque cada desarrollador lidera ciertas funcionalidades, ambos integrantes trabajan de manera colaborativa en todas las áreas del sistema, asegurando que todo el código sea revisado, comprendido y mejorado conjuntamente.
Las áreas de responsabilidad principal establecidas son:

- Desarrollo del chatbot con LLM, manejo de sesiones y cifrado de mensajes: Riera Bauer Fabrizio.
- Desarrollo de las estadísticas, integración con RAG y evaluación del comportamiento del modelo: Loyola Maria Luciana.

Etapa 1: Investigación, análisis y captura de requisitos (8 semanas)

El objetivo de esta etapa es lograr un entendimiento completo de la problemática planteada y de los sistemas y herramientas disponibles para el desarrollo del proyecto. En consonancia con los principios de XP, se enfatiza la comunicación frecuente con los clientes y la colaboración continua entre ambos desarrolladores.

- **Investigación, análisis y captura de requisitos generales (ambos integrantes):** A través de reuniones semanales con los clientes se capturaron los requisitos funcionales y no funcionales, considerando el manejo de datos y la interacción con el sistema. Se identificaron las necesidades y se recopiló la información necesaria para el desarrollo del asistente virtual. Esta actividad refleja la práctica de XP del "cliente en el sitio", adaptada mediante comunicación cercana y frecuente con los stakeholders.
- **Análisis de modelos y arquitectura (liderado por Fabrizio):** Se realizó un análisis comparativo de modelos de lenguaje y arquitecturas disponibles, evaluando su compatibilidad técnica, eficiencia y escalabilidad. Se seleccionó el modelo y la arquitectura que mejor se adaptan a las necesidades del asistente y a la infraestructura planificada.
- **Análisis de modelos y aspectos éticos (liderado por Luciana):** Se llevó a cabo un análisis comparativo de modelos con enfoque en los costos de implementación y mantenimiento, evaluando el impacto ético y la adherencia a criterios de privacidad y anonimización establecidos en la "Guía de Transparencia y Protección de Datos para una Inteligencia Artificial responsable". Se definió cómo estos aspectos influirán en la lógica de respuestas del asistente. Se trabajó de manera colaborativa para garantizar la integración de estos principios éticos en todo el sistema.

Etapa 2: Diseño y desarrollo (16 semanas)

Esta etapa abarca la planificación de la arquitectura del sistema y su implementación incremental. Siguiendo los principios de XP, se prioriza el diseño simple, la refactorización continua y la integración constante del trabajo de ambos desarrolladores.

- **Diseño y desarrollo general (ambos integrantes):** Se desarrolló la plataforma web que aloja al asistente conversacional, incluyendo las interfaces de usuario y la estructura de la base de datos

que almacena tanto los mensajes de los chats como la información requerida para la herramienta RAG. También se integró el contenido de la página web actual de los clientes en la plataforma, permitiendo que los usuarios accedan a información relevante sobre el proyecto. Durante esta fase se elaboraron diagramas de clases, casos de uso y arquitectura de componentes para representar la estructura y el flujo del sistema. Se definió el modelo de base de datos con sus entidades, relaciones y restricciones para garantizar consistencia, seguridad y eficiencia en el acceso a los datos.

- **Asistente y manejo de sesiones (liderado por Fabrizio):** Se implementó el asistente conversacional, considerando el cifrado de todos los mensajes intercambiados. Se gestionaron las sesiones de tres tipos de usuarios registrados (investigadores, estudiantes y administradores), así como las sesiones de chat de los usuarios no registrados, tanto desde la interfaz pública como desde la interfaz de muestra controlada, protegiendo y cifrando todos los datos. Se implementó la arquitectura del sistema elegida para contar con una plataforma adecuada para el tráfico de datos considerado. Además, se optimizó el rendimiento general del sistema, mejorando la eficiencia de las consultas a la base de datos y reduciendo los tiempos de respuesta.
- **RAG y estadísticas (liderado por Luciana):** Se implementó la técnica RAG, generando una correcta integración con el asistente y desarrollando métricas específicas para evaluar su desempeño. Se evaluó el comportamiento del modelo ajustando el system prompt y revisando la coherencia de las respuestas según los parámetros establecidos por los clientes. También se generaron estadísticas basadas en los mensajes de chat que permitieron a los clientes analizar el nivel de riesgo de los usuarios y sus patrones de interacción con el modelo, garantizando que la herramienta sea útil para el desarrollo de su proyecto de investigación.

Etapa 3: Testing, validación y despliegue (6 semanas)

El testing del sistema tiene como objetivo verificar que el sistema cumple con los requerimientos que guiaron su diseño y desarrollo, ejecuta las funcionalidades planteadas en los tiempos requeridos y cumple con los estándares de privacidad necesarios dada la sensibilidad de la temática. La validación de las respuestas del modelo constituye una parte fundamental para el correcto funcionamiento del sistema y representa un requerimiento esencial del proyecto. Esta etapa refleja el énfasis de XP en las pruebas continuas y la integración constante.

- **Testing, validación y despliegue general (ambos integrantes):** Se probó el flujo completo del sistema, verificando que cada caso de prueba sea exitoso y cumpla con los requerimientos analizados en etapas anteriores. Se validaron las respuestas del modelo de manera automatizada y manual. En el caso de la validación manual, se contó con el apoyo de los clientes para confirmar que las respuestas generadas por el modelo cumplen con sus expectativas y requisitos profesionales.
- **Sesiones y chat (liderado por Fabrizio):** Se verificó que las sesiones se generen correctamente para los distintos roles de usuario (investigadores, estudiantes y administradores) y que los permisos y restricciones funcionen según lo definido. Se probaron las sesiones de chat con el asistente para asegurar que funcionen según los plazos de tiempo establecidos. Se verificó que los datos intercambiados en los chats se almacenen y cifren de manera segura, manteniendo la integridad y privacidad de los datos.
- **RAG y modelo (liderado por Luciana):** Se verificó que la información obtenida mediante la técnica RAG sea relevante y se utilice en los momentos adecuados, garantizando que el modelo no genere respuestas inapropiadas, ofensivas o fuera de contexto. Se evaluó la precisión, coherencia y consistencia de las respuestas en diversos escenarios, incluyendo casos extremos, verificando también que el modelo proporcione correctamente contactos de emergencia ante situaciones de alto riesgo.

El proceso de despliegue comprende el conjunto de actividades necesarias para poner el sistema a disposición del cliente. Este proceso se llevó a cabo mediante dos etapas. En la primera fase, se efectuó pruebas en un entorno local, utilizando como servidor las computadoras de los desarrolladores. Posteriormente, en la segunda fase, se procedió a implementar el sistema en infraestructura de nube, lo que permite el acceso remoto al mismo. El equipo asumió la responsabilidad compartida de ejecutar todas las tareas requeridas para completar el despliegue del sistema en el entorno de nube correspondiente.

Etapa 4: Capacitación del cliente y prueba controlada (2 semanas)

En esta etapa final, el equipo capacitó al equipo de Psicología, lo que implica enseñar el uso del sistema, explicar su funcionamiento técnico y operativo y detallar cómo se llevará a cabo un flujo de muestra controlada (visita a una institución para el uso del asistente). En este contexto, una prueba controlada consiste en que el equipo de Psicología realiza un taller sobre la temática con un grupo de interés y, durante el mismo, solicita a los participantes que utilicen el sistema para comunicarse con el asistente.

Los desarrolladores participaron conjuntamente con el equipo de Psicología en la realización de una prueba controlada o también conocida como test de aceptación, con grupos de alumnos de primer año de la UNSL, ya que constituyen un grupo etario similar al público y usuarios finales de interés para la investigación. También se hizo una visita a una escuela secundaria de la ciudad de San Luis para realizar la prueba controlada con estudiantes de sexto año.

Diagrama de Gantt y distribución temporal

La planificación establecida para el desarrollo del proyecto abarcó 20 semanas. Aunque la suma de las duraciones individuales de cada etapa supera este período, la construcción del sistema fue viable en el plazo establecido gracias a la ejecución paralela de las actividades, siempre que las dependencias entre tareas lo permitieron.

Para facilitar la visualización de la planificación temporal y la distribución de tareas, se presenta a continuación un diagrama de Gantt **Figura 2.11** junto con una tabla explicativa **Tabla 2.1** que detallan las actividades, sus duraciones y las responsabilidades principales de cada integrante, recordando que ambos desarrolladores trabajan colaborativamente en todas las áreas del proyecto.

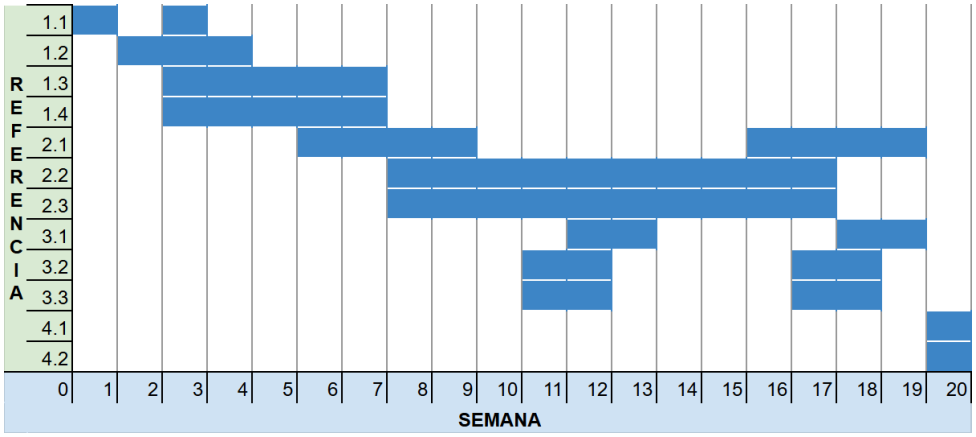

Figura 2.11. Diagrama de Gantt del proyecto.

Referencia	Tarea	Tiempo (semanas)	Estudiante
1.1	Reuniones y Captura de Requisitos	2	Riera - Loyola
1.2	Investigación y Análisis General	3	Riera - Loyola
1.3	Investigación y Análisis de Modelo y Arquitectura	5	Riera
1.4	Investigación y Análisis de Modelo y Ética	5	Loyola
2.1	Diseño y Desarrollo General	8	Riera - Loyola
2.2	Desarrollo del Chatbot y Manejo de Sesiones	10	Riera
2.3	Desarrollo de RAG y Estadísticas	10	Loyola
3.1	Testing, Validación y Despliegue General	4	Riera - Loyola
3.2	Testing, Validación y Despliegue de Sesiones y Chat	4	Riera
3.3	Testing, Validación y Despliegue de RAG y Estadísticas	4	Loyola
4.1	Capacitación del cliente	1	Riera - Loyola
4.2	Prueba controlada	1	Riera - Loyola

Tabla 2.1. Detalle de tareas y tiempos del proyecto.

2.2.2. Costos

El análisis de costos se divide en varios aspectos: la infraestructura tecnológica necesaria para ejecutar el modelo de IA, tanto en un entorno local como en la nube, los servicios de hosting y dominio del sistema y los costos asociados al desarrollo.

En primer lugar, el aspecto del modelo de IA conlleva un costo complejo que se tuvo que analizar en profundidad. Para ello, separamos el análisis de nuestros posibles proveedores de modelos en dos secciones: Servicio Local y Servicio en la nube.

2.2.2.1. Servicio Local

El servicio local se refiere a ejecutar los modelos de IA directamente en una computadora personal o servidor propio, sin depender de plataformas en la nube. Esta opción gratuita elimina la necesidad de controlar el tiempo de uso o la cantidad de tokens procesados. Se analizaron dos de las opciones más populares en la actualiadad: Ollama [52] y Hugging Face [53].

Ollama

Ollama es framework de código abierto que establece un entorno de ejecución preconfigurado y optimizado para correr versiones de modelos de IA que requieren menos recursos.

La característica más importante de esta implementación es que todos los datos se procesan de forma interna, dentro de nuestros propios servidores. Esto contribuye a que la información potencialmente sensible o privada no se envíe a servidores de terceros.

De esta manera, logramos un mayor control sobre la privacidad, la seguridad y la trazabilidad de la información dentro del proyecto. Así, además, alineándose mejor con la "Guía de transparencia y protección de datos personales para una IA responsable" [12]. No depender de servicios de terceros, elimina la posibilidad de que los datos pasen por intermediarios o plataformas en la nube que podrían llegar a tener interés en usar los datos bajo sus propias políticas de privacidad.

Con el objetivo de evaluar los costos y el rendimiento de la implementación utilizando el servicio local Ollama, se utilizó una computadora con las especificaciones detalladas en la **Tabla 2.2**

Con un costo de inversión inicial aproximado de $USD 1.215, esta computadora logró realizar pruebas de tres modelos de LLM accesibles por Ollama con muy buen rendimiento basado en tiempo de

procesamiento y calidad de las respuestas provistas, el cuál será analizado a continuación.

Se hicieron pruebas con un pequeño código de Python accediendo con Ollama a los siguientes modelos: Gemma, Llama y DeepSeek R1. Al evaluar DeepSeek R1 en nuestro prototipo, observamos que, aunque cuenta con una capacidad avanzada de razonamiento y análisis profundo, su tiempo de respuesta es significativamente largo (3 segundos), lo que no lo hace adecuado para un chatbot en tiempo real. En cambio, al analizar Gemma 3B y LLaMA 3.2, ambos presentaron tiempos de respuesta muy rápidos (menos de un segundo), sin embargo, Gemma se destacó por generar respuestas de mayor calidad. Este modelo utilizó de forma adecuada el contexto proporcionado y respondió de manera más entendible y razonable. Como Gemma logró entender mejor el contexto y responder de mejor manera en tiempos adecuados, nos pareció la opción más adecuada.

Si bien esta opción se presentó inicialmente como la más prometedora, tras analizar en detalle los distintos aspectos de su implementación, se concluyó que no resulta la alternativa más conveniente para nuestro proyecto. Ya que aunque la computadora fue capaz de responder de manera adecuada a todas las pruebas realizadas, en un entorno con más de 30 consultas simultáneas es muy probable que no logre sostener un rendimiento óptimo. Para ejecutar modelos de IA con Ollama de esta forma, aun siendo una alternativa gratuita, se requiere de un servidor con mayor capacidad de procesamiento y recursos dedicados.

Hugging Face

Además de Ollama, también analizamos Hugging Face como opción de servicio local. Hugging Face es una de las plataformas de modelos de IA más reconocidas. Su Hub aloja más de 1,7 millones de modelos, 400.000 conjuntos de datos y 600.000 aplicaciones de demostración (“Spaces”), todos de código abierto y disponibles públicamente para facilitar la colaboración y el desarrollo de proyectos de aprendizaje automático.

Para una implementación gratuita utilizaríamos la librería *Transformers*, que reúne modelos preentrenados para tareas de PLN, visión artificial, audio y tareas multimodales, tanto para inferencia como para entrenamiento.

Además, una de las principales ventajas de esta librería es la amplia variedad de modelos preentrenados disponibles, que permite elegir según la tarea que se desea resolver y adaptar el sistema a distintos requerimientos de precisión, tamaño y velocidad.

En nuestro caso, el modelo tendrá la tarea de generación de texto. En la página oficial de Hugging Face Model Hub se puede acceder a todos los modelos disponibles, con una lista que supera los 200,000. Entre los modelos más destacados para generación de texto se encuentran *deep-seek*, *NemoTron (NVIDIA)* y *LLaMA*.

En el caso de Hugging Face, se realizaron las mismas pruebas utilizando la computadora con las especificaciones detalladas en la **Tabla 2.2**. Los resultados fueron similares a los obtenidos con Ollama, aunque con tiempos levemente mayores, lo que convierte a Hugging Face en la opción más lenta entre ambos proveedores locales.

Descripción	**Componente**
Intel Core i7-12700	Microprocesador
MB 1700 Gigabyte	Motherboard
DDR4 – 16 GB * 2	Memoria RAM
M.2 – 1 TB	Disco sólido
Aerocool 750W Plus Bronce	Fuente
Placa de Video 4060 Ti 8 GB	Placa de video
ID Cooling	Disipador de aire

Tabla 2.2. Configuración de hardware utilizada para las pruebas locales

Al igual que en la opción de Ollama, a pesar de la gran accesibilidad y flexibilidad de Hugging Face, los requerimientos de hardware dificultan su aplicación en nuestro proyecto. El uso de la librería Transformers y la carga de modelos grandes puede requerir recursos computacionales considerables, especialmente RAM y GPU, para obtener un rendimiento fluido y rápido en tiempo real. La implementación gratuita en entornos locales o sin infraestructura dedicada puede generar latencias y limitaciones en el procesamiento de solicitudes concurrentes.

Por lo tanto, a pesar de su flexibilidad y variedad de modelos, la necesidad de GPUs con alta VRAM hace que esta alternativa sea inviable para nuestro proyecto en esta etapa.

2.2.2.2. Servicio en la Nube

Ante estas limitaciones de hardware local, exploramos la opción de servicio en la nube.

El servicio en la nube implica el uso de modelos de IA que están alojados en servidores de terceros, a los que se accede remotamente a través de internet. Esta implementación elimina la necesidad de contar con hardware de alto rendimiento y reduce la complejidad del mantenimiento local. Sin embargo, implica gestionar cuidadosamente el tiempo de uso, la cantidad de tokens procesados por el modelo utilizado y la forma en que se procesan los datos enviados según las políticas de privacidad y protección de datos bajo los que funcione la plataforma de servicio.

Hugging Face

La implementación paga del servicio con Hugging Face se realiza a través de los Inference EndPoint. Inference Endpoints (Puntos de Inferencia) ofrece una solución de producción segura para desplegar fácilmente cualquier modelo de Transformers, con la posibilidad de aplicar autoescalado.

La creación de un Hugging Face Inference Endpoint comienza a partir de un modelo alojado en Hugging Face Hub. Durante el proceso, el servicio genera artefactos de imagen a partir del modelo seleccionado, ya sea directamente desde el repositorio o mediante un contenedor personalizado proporcionado por el usuario. Estos artefactos se construyen de manera independiente del repositorio original, lo que permite que el endpoint funcione de forma autónoma una vez desplegado [54].

Al configurar el endpoint, es posible seleccionar el tipo de instancia de cómputo, ya sea CPU o GPU y ajustar la escala del modelo según la demanda. El cobro se realiza según la tarifa por hora de la instancia utilizada y se aplica mientras el endpoint se encuentre en ejecución o en proceso de inicialización.

Esta implementación también cuenta con el autoescalado. Esto permite ajustar dinámicamente el número de réplicas del endpoint que ejecutan los modelos basándose en el tráfico y la utilización del acelerador. Al aprovechar el autoescalado, se puede manejar sin problemas las diferentes cargas de trabajo, optimizando los costos y asegurando una alta disponibilidad.

Para hacer una estimación del costo que conlleva esta opción utilizamos el siguiente ejemplo tomado de la documentación de los precios por Inference Endpoint [55]. Para calcular los costos se utiliza la siguiente fórmula (2.1).

$$THI * (Hrs. * Nro.\quad Replicas\quad min.) + (Hrs.\quad de\quad escalado * Nro.\quad Replicas\quad adic.) \quad (2.1)$$

Los términos mencionados en la fórmula (2.1) son:

- **Tarifa Horaria de la Instancia (THI):** Es el precio por hora que se paga por usar una instancia (una máquina virtual de CPU o GPU) en Hugging Face.
- **Número de Replicas mínina (Nro. Replicas min.):** Es la cantidad mínima de réplicas (copias de la instancia) que siempre están activas cuando el endpoint está desplegado.
- **Horas de uso (Hrs.):** Se refiere al total de horas durante las cuales el endpoint está activo y funcionando.

- **Horas de escalado (Hrs. de escalado):** Si se usa auto-escalado, durante momentos de mayor demanda el sistema puede crear réplicas adicionales para atender el tráfico. Horas de escalado indica cuántas horas (o fracción) de esas réplicas adicionales estuvieron activas.
- **Número de Replicas adicionales (Nro. Replicas adic.):** Es cuántas réplicas extra (además de la mínima) se usan durante las horas de escalado.

Según la documentación, un Endpoint Básico requiere la utilización de los siguientes recursos:

- AWS CPU intel-spr x2 (2 vCPUs, 4GB RAM)
- Autoescalado (mínimo 1 réplica, máximo 1 réplica)

De esta forma, el costo mensual resulta en:

$$\$USD0{,}064/hora * (730horas * 1replica) = \$USD46,72/mes \tag{2.2}$$

Debido a que el acceso a los servicios en la nube de Hugging Face requiere una suscripción de pago, no se realizaron pruebas utilizando esta opción.

Finalmente, aunque esta opción ofrece los mejores resultados en términos de usabilidad, mantenimiento y escalabilidad, también se descartó dado que no se dispone de financiamiento para afrontar el costo mostrado en la fórmula (2.2).

API de Gemini

La API de Gemini, desarrollada por Google, permite interactuar con modelos de lenguaje de gran escala y modelos multimodales mediante solicitudes HTTP. Esta interfaz facilita el envío de prompts y la recepción de respuestas generadas por modelos avanzados como Gemini 2.5 Pro, Gemini Flash, Gemini 2.0, entre otros.

Versión gratuita

La API de Gemini permite acceder a Gemma como servicio alojado para el desarrollo de aplicaciones o la creación de prototipos de forma gratuita [56].

Gemma es una serie de modelos de IA de código abierto de última generación. Se basan en la misma tecnología que se utilizó para crear los modelos Gemini.

La versión Gemma 3 es multimodal, lo que significa que puede procesar tanto texto como imágenes como entrada y generar texto como salida. Gemma 3 se destaca por su amplia ventana de contexto de 128K (lo que le permite procesar y entender grandes cantidades de información a la vez), su soporte multilingüe en más de 140 idiomas y su disponibilidad en más tamaños que versiones anteriores. Esto la hace ideal para diversas tareas de generación de texto y comprensión de imágenes, como la respuesta a preguntas, el resumen y el razonamiento. Gracias a su tamaño relativamente compacto, los modelos Gemma 3 pueden implementarse en entornos con recursos limitados, como laptops o computadoras de escritorio[57].

Al ser un servicio en la nube, una de sus principales ventajas es que no requiere instalación ni ejecución local del modelo. Esto simplifica el desarrollo, evita la necesidad de equipos de alto rendimiento, configuraciones avanzadas o mantenimiento y permite implementar un asistente eficiente, preciso, veloz y de bajo costo. En nuestro proyecto, esto resulta especialmente útil, ya que buscamos implementar un asistente que sea lo más eficiente posible, con respuestas no solo precisas, sino también veloces, además que implique bajos costos de desarrollo y mantenimiento.

La mayoría de las alternativas disponibles cobran por la cantidad de tokens procesados por mes o por el tiempo en que el modelo está en uso, lo cual, al pensar en un asistente conversacional que estará disponible todo el día, implicaría un costo importante.

Sin embargo, la versión gratuita de la API tiene una serie de restricciones que debemos tener en cuenta al momento de incorporarla a nuestro proyecto. La API de Gemini impone un límite en cuanto a

la frecuencia con la que se pueden realizar llamadas desde un mismo proyecto. Estos límites de frecuencia se miden en cuatro dimensiones:

- **Solicitudes por minuto (RPM):** Define el número máximo de peticiones que se pueden enviar al modelo en 60 segundos.
- **Solicitudes por día (RPD):** Establece el total de peticiones permitidas en un periodo de 24 horas.
- **Tokens por minuto (TPM):** Indica la cantidad máxima de unidades de texto (entrada y salida) que el modelo puede procesar en un minuto.
- **Tokens por día (TPD):** Limita el volumen total de tokens que pueden ser procesados o generados en un día completo.

El uso se evalúa en función de cada límite y, si se supera alguno de ellos, se activará un error de límite de frecuencia. Por ejemplo, si el límite de RPM es de 20, realizar 21 solicitudes en un minuto generará un error, incluso si no se superó el TPM ni otros límites.

Un aspecto importante a tener en cuenta en esta implementación es que, al trabajar con un modelo alojado en servidores externos, toda la información ingresada por el usuario, incluyendo preguntas, frases personales o datos potencialmente sensibles, se envía a los servidores de Google para ser procesada. Según la documentación oficial de los Términos de la API de Gemini, vigente desde mayo de 2025, Google aclara que puede utilizar el contenido que los usuarios envíen, tanto prompts como datos y respuestas generadas, para mantener, desarrollar y mejorar sus modelos y servicios. Además, se establece explícitamente que revisores humanos pueden acceder y revisar el contenido enviado a la API, con fines de control de calidad y mejora continua [58].

Además, tal como advierte la Guía para una IA responsable, el uso de modelos de IA en la nube puede implicar riesgos vinculados a la violación de la privacidad, la reidentificación de datos anónimos y el uso de la información con fines no autorizados o sin el consentimiento explícito del titular. Incluso si no se guarda una copia local, el simple hecho de transferir información sensible a un tercero ya genera obligaciones legales.

En el caso de nuestro asistente conversacional, es clave reconocer que estamos trabajando con un público que podría formar parte de grupos vulnerables, lo que conlleva un mayor riesgo. Por eso, deberemos reforzar el principio de responsabilidad proactiva: asegurarnos de minimizar al máximo los datos recolectados, no registrar información sensible innecesaria y minimizar el tiempo de guardado de historiales. También será necesario dejar bien claro cuándo se está interactuando con un sistema de IA, e informar al usuario sobre cómo se almacenan y usan sus datos.

Teniendo en cuenta todos los aspectos analizados previamente, consideramos que esta no es la opción más adecuada para nuestro proyecto. Al revisar en detalle la documentación de Google sobre el uso y análisis de los datos, concluimos que sus políticas no se alinean con los estándares de privacidad y seguridad que se especifican en la Guía para una IA responsable.

Versión paga

Para la versión paga se analizó el modelo Gemini 2.5 Flash [36], que forma parte de la familia de modelos de Google. A diferencia de Gemma 3, este modelo no es gratuito, pero su precio por uso es uno de los más accesibles dentro de las versiones pagas disponibles, como se analizó anteriormente en la **Figura 2.10**.

Además, debido a la integración del sistema RAG y a que esta opción ya cuenta con créditos disponibles en la API de Gemini, se optó por utilizar los generadores de embeddings de Google. Estos se emplearán para generar los embeddings tanto de los documentos que se suben a la página como de las consultas de los usuarios. El costo de este servicio es de **$USD 0,15 por millón de tokens de entrada**.

El costo se calcula en función de la cantidad de tokens de entrada y de salida. Por ejemplo, procesar un millón de tokens de entrada cuesta aproximadamente $USD 0,30, mientras que generar un millón

de tokens de salida cuesta unos $USD 2,50. Esto lo convierte en una alternativa viable en términos económicos si se busca una mayor capacidad de respuesta sin comprometer el presupuesto.

$$CM = (TEM/1M) * \$USD0,3 + (TSM/1M) * \$USD2,5 + (TEE/1M) * \$USD0,15 \quad (2.3)$$

CM: Costo Mensual
TEM: Tokens de Entrada Mensual
TSM: Tokens de Salida Mensual
TEE: Tokens de Entrada Embedding mensual

Para realizar un estimado del costo mensual que conllevaría la utilización de este servicio, se asumieron los siguientes aspectos: una utilización de aproximadamente 70 sesiones por mes, en la cual cada sesión realiza 15 preguntas, un estimado de tokens de entrada y salida es el siguiente. Debido a que cada prompt será tokenizado y no tenemos una forma concreta de calcular la cantidad justa de tokens que necesitaremos, asumiremos los siguientes valores para calcular un aproximado:

Tokens de entrada:

$$\begin{aligned} &50(pregunta\ del\ usuario) + 1000(informacion\ de\ RAG) \\ &+1000(mensajes\ del\ historial) + 300(instruccion\ del\ sistema) \\ &\approx 2350\ tokens\ de\ entrada \end{aligned} \quad (2.4)$$

Tokens de salida:

Asumiendo que la respuesta del asistente será corta, con un aproximado de 300 palabras, en total serían: *400 tokens de salida*

Por lo que, para cada pregunta se tendrá un total de *2.350 tokens de entrada* y *400 tokens de salida*.

Si se asumen un total de 15 preguntas por sesión: *35.250 tokens de entrada* y *6.000 tokens de salida*. Y si asumimos un total de 70 sesiones: *2.467.500 tokens de entrada* y *420.000 tokens de salida*.

Para el cálculo de embeddings se toma el costo de subida de PDFs fijo y ya cubierto por la versión gratuita, pero para los embeddings de las consultas se tienen *52.500 tokens de entrada* (50 tokens, con 15 preguntas por 70 sesiones).

Ahora, tomando estas aproximaciones finales podemos reemplazar los valores en la ecuación inicial.

$$CM = (2\,467\,500/1M) * \$USD0,3 + (420\,000/1M) * \$USD2,5 + (52\,500/1M) * \$USD0,15 \quad (2.5)$$

$$CM = 2,4675 * \$USD0,3 + 0,42 * \$USD2,5 + 0,0525 * \$USD0,15 \quad (2.6)$$

$$CM = \$USD0,74 + \$USD1,05 + \$USD0,0078 \quad (2.7)$$

$$CM = \$USD1,79 \quad (2.8)$$

Podemos concluir que, considerando el costo mensual aproximado que se muestra en la ecuación (2.8), se trata de un valor accesible. Este costo nos permite acceder a un modelo de gran potencia y capacidad de procesamiento, ofreciendo un rendimiento elevado.

Además, este modelo no solo tiene un costo accesible, sino que también permite realizar más solicitudes por minuto y mantener flujos de interacción constantes sin interrupciones, lo cual es fundamental para una herramienta disponible 24 horas y con múltiples usuarios simultáneos. Al tratarse de un modelo más potente que Gemma en su versión flash, el tiempo de respuesta habitualmente es más rápido y la calidad de las respuestas mejora, especialmente en conversaciones más complejas o con múltiples turnos.

Debido a que vamos a optar por una versión paga con crédito disponible, el modelo Gemini 2.5 Flash cuenta con un máximo de 1.000 solicitudes por minuto (RPM), lo que indica que el modelo puede gestionar hasta mil peticiones en un minuto, un límite totalmente aceptable para el uso estimado que tiene nuestro proyecto. Además, se permite un total de 1.000.000 de tokens por minuto (TPM), lo que

Tipo	Características	Ventajas	Desventajas	Costo
Servidor propio	Acceso local a los modelos. Mínimo de memoria RAM de 16 GB libre. Recomendable una tarjeta de video potente.	Procesamiento de datos interno en el proyecto. Uso gratuito de los modelos sin límite de uso.	Requerimiento computacional alto.	Inversión inicial de $USD 1.250 aprox.
En la nube sin pago	Acceso a través de API de Gemini. Restricción de uso por cantidad de consultas y tokens.	Requerimiento computacional bajo. Acceso gratuito a modelos avanzados.	Límite bajo de cantidad de tokens y consultas por minuto. Posibles problemas de seguridad y privacidad. Procesamiento completo de datos por Google. Revisores humanos de Google pueden leer la entrada y salida de la API.	No aplica
En la nube pago	Acceso a través de API de Gemini. Límite amplio de utilización de tokens y consultas.	Acceso a modelos avanzados. Requerimiento computacional bajo. No se utilizan los datos para entrenar modelos.	Almacena las instrucciones y respuestas durante un período limitado.	Costo mensual de $USD 1,79

Tabla 2.3. Comparación de opciones de uso de modelos de IA.

asegura la capacidad de procesar hasta un millón de tokens, de entrada y salida combinados, por minuto, asegurando así una experiencia fluida en las interacciones. Por otro lado, las solicitudes por día (RPD) tienen un límite de 10.000, un límite totalmente aceptable teniendo en cuenta la utilización que se planea tendrá el asistente.

En términos de privacidad y seguridad para los servicios pagos, la documentación oficial de los Términos de la API de Gemini [58] establece que Google no utiliza los prompts enviados ni las respuestas generadas para entrenar modelos o mejorar sus productos. En su lugar, estos datos se procesan de acuerdo con el Data Processing Addendum; esto significa que Google solo procesa esa información para que la herramienta funcione correctamente y no la usa para otros fines.

Sin embargo, Google registra temporalmente los prompts y las respuestas con el único objetivo de detectar posibles violaciones a su Política de Uso Prohibido o cumplir con obligaciones legales o regulatorias. Esta información puede almacenarse de forma transitoria o en caché en cualquier país donde Google o sus agentes operen. Además, otros datos operativos y administrativos, como la información de la cuenta, historial de facturación, métricas de uso, quedan sujetos a la Política de Privacidad de Google, lo que significa que Google sigue siendo responsable de cómo se usan y protegen esos datos.

En la **Tabla 2.3** se comparan las principales alternativas que analizamos anteriormente: servidor propio, acceso en la nube y acceso en la nube pago. Podemos observar no solo en términos de costos, sino de ventajas y desventajas.

En conclusión, la **versión paga de la API de Gemini** y en específico Gemini 2.5 Flash, resulta ser la opción más recomendable para nuestro proyecto. Ofrece un mejor control sobre el uso de datos y permite

un alto volumen de solicitudes y tokens por minuto, ofreciendo fluidez y calidad en las respuestas con un costo accesible. Consideramos que esta opción es viable si contamos con el presupuesto necesario y mantenemos una comunicación clara con los usuarios acerca de cómo la API manejará su información.

Servidor

Por último, debido a que en todos los casos necesitaremos que nuestro proyecto se encuentre disponible en línea para su uso por parte de los usuarios, el costo asociado al dominio y al hosting es fijo. Esto implica la contratación de un dominio propio, así como de un servicio de hosting que permita alojar tanto la aplicación como el modelo de IA en funcionamiento y la API que se encargará del cálculo de estadísticas.

En nuestro caso particular, desarrollamos una API utilizando FastAPI, la cual requiere un entorno con al menos 8 GB de memoria debido al uso de la librería PySentimiento, que incorpora modelos de procesamiento de lenguaje natural de gran tamaño. Este requerimiento incrementa la demanda de recursos del servidor, motivo por el cual se evaluó cuidadosamente el costo de las instancias en la nube.

Se realizará el análisis de posibles proveedores y también la posibilidad de contar con un servidor de la universidad. Entre las posibles opciones de servidores se analizó Hostinger [59], Vercel [38], AWS [60] y el servidor de la Universidad Nacional de San Luis.

Hostinger

Hostinger es una empresa tecnológica global fundada en 2004 en Lituania, especializada en soluciones de alojamiento web, VPS, dominios y herramientas digitales para proyectos online. Actualmente, Hostinger atiende a más de 3 millones de usuarios en todo el mundo, ofreciendo tecnología de alto rendimiento.

El plan más vendido y más adecuado para el proyecto se denomina KVM 2, el cual tiene un precio de $13.999 pesos por mes. Este plan incluye un servidor que provee 2 núcleos de CPU, 8 GB de RAM, 100 GB en disco NVMe y 8 TB de ancho de banda. Estas características serían más que suficientes para alojar la aplicación de Next.js y la API de Python.

Vercel

Vercel, compañia introducida anteriormente en la sección de tecnologías de desarrollo, también ofrece diversos planes, el plan más popular entre los desarrolladores es el plan Hobby. Es un plan gratuito que permite obtener muy buenos recursos, entre estos se encuentran 4 horas de CPU activo por mes, 360 GB-hrs, seguridad con Firewall, acceso con HTTPS, entre otros.

La desventaja que presenta el plan Hobby es que no se puede cambiar el dominio al que está desplegado. Al desplegar con el plan Hobby, la página se va a encontrar en un dominio de Vercel que no podrá ser posible de encontrar en Google.

AWS

Amazon Web Services (AWS) es una plataforma de servicios en la nube ofrecida por Amazon que proporciona infraestructura tecnológica bajo demanda a empresas y desarrolladores. Permite acceder a recursos como servidores virtuales, bases de datos, almacenamiento, redes y herramientas de inteligencia artificial, sin necesidad de mantener equipos físicos propios. Gracias a su modelo escalable y flexible, los usuarios pueden pagar solo por lo que utilizan, adaptando fácilmente la capacidad según las necesidades del proyecto y optimizando así los costos operativos.

En el modelo On-Demand de Amazon EC2, el usuario paga únicamente por el tiempo en que la instancia está activa. Para este proyecto, que requiere una instancia con aproximadamente 8 GB de memoria RAM y CPU de capacidad moderada, si se utiliza 5 horas por semana, el costo estimado de cómputo es de aproximadamente $USD 1,80 por mes. El almacenamiento EBS necesario para conservar el sistema operativo, las librerías y el código de la aplicación representa un gasto fijo de $USD 3,30 por mes. El costo total estimado del servidor sería de aproximadamente $USD 5,10 por mes.

Servidor de la Universidad Nacional de San Luis

La Universidad Nacional de San Luis proporcionó un servidor con 20 GB de almacenamiento, 4 GB de memoria RAM y sistema operativo Linux. Este espacio y capacidad de memoria resultan suficientes para desplegar la aplicación y la API que se necesita.

Opción seleccionada de servidor

Luego de analizar las cuatro opciones disponibles, decidimos desplegar la aplicación de Next.js en *Vercel Hobby* y la API de estadísticas en el *servidor de la Universidad Nacional de San Luis*. Esta combinación nos permite aprovechar los recursos de ambas opciones con la ventaja de que no es necesario ningún costo adicional.

Dominio

Para la gestión del dominio se analizó la posibilidad de adquirir uno a través de NIC Argentina[61]. Según la información provista por el organismo, el costo de alta para un dominio .com.ar es de $8.500, con una renovación anual del mismo valor.

En cuanto al despliegue y considerando las opciones de servidor seleccionadas, la aplicación desarrollada en Next.js utilizará un dominio proporcionado por Vercel, correspondiente a su plan de hosting. Por otro lado, la API alojada en el servidor de la Universidad Nacional de San Luis no contará con un dominio propio, dado que su acceso será interno y restringido exclusivamente a la aplicación principal. Además, se implementarán mecanismos de seguridad que limitarán las solicitudes únicamente a orígenes permitidos y con autorización, garantizando la protección de los datos y del servicio.

Costo de desarrollo

En esta evaluación, se contempla el trabajo de un equipo compuesto por un líder (Director de Proyecto) y dos desarrolladores Backend, con una estimación inicial del tiempo total de desarrollo de tres meses, además de un mes adicional para la captura de requisitos y un mes para el despliegue y pruebas. El cálculo estimado de la remuneración del equipo, durante el período de desarrollo del software, se realizó teniendo en cuenta los honorarios mensuales de octubre de 2025 obtenidos del "Consejo Profesional de Ciencias Informáticas de la Provincia de Córdoba" [62]: $3.638.207,81 para el líder y $2.616.089,35 para cada desarrollador.

En la **Tabla 2.4** se presenta el cálculo detallado de cada una de las actividades, desglosado según los roles que conforman el equipo de trabajo y las respectivas remuneraciones asociadas. A partir de este análisis, se obtiene un costo total estimado de $44.411.932,52, distribuido de acuerdo con las tareas realizadas y las responsabilidades de cada integrante del equipo.

Actividad	**Duración**	**Rol**	**Remuneración Mensual (ARS)**	**Remuneración Total (ARS)**
Captura de Requisitos	1 mes	Líder 2 Desarrolladores	3.638.207,81 2.616.089,35	3.638.207,81 5.262.178,7
Desarrollo de la Aplicación	3 meses	Líder 2 Desarrolladores	3.638.207,81 2.616.089,35	10.914.623,4 15.696.536,1
Testing y Despliegue	1 mes	Líder 2 Desarrolladores	3.638.207,81 2.616.089,35	3.638.207,81 5.262.178,7
Total	-	-	-	44.411.932,52

Tabla 2.4. Estimación de costos de remuneración del equipo de desarrollo

Aportes positivos a la sociedad

Aunque hasta el momento se han detallado los costos asociados al desarrollo del sistema, es importante destacar que este proyecto trabaja sobre una problemática social muy relevante. La prevención de adicciones, especialmente la ludopatía en adolescentes, es una temática que está en creciente auge y que preocupa cada vez más por su aparición en personas de edades cada vez más jóvenes.

Este sistema busca ofrecer una solución preventiva que, a largo plazo, puede representar un efecto muy positivo para la sociedad. Recuperar a una persona con ludopatía tiene un costo muy alto, no solo por el tratamiento psicológico que requiere, sino también por las consecuencias sociales, familiares y, en algunos casos, judiciales que trae el problema.

Frente a esto, un sistema de prevención temprana como el propuesto tiene el potencial de reducir significativamente esos costos, actuando antes de que el problema se agrave. Además, al fomentar un uso consciente y responsable de la tecnología, también contribuye a disminuir los riesgos psicológicos y económicos asociados a las adicciones digitales.

A futuro, con un uso continuo y responsable, la aplicación del sistema podría conllevar:

- Ayudar a crear conciencia sobre los riesgos de las nuevas tecnologías.
- Reducir los gastos en tratamientos psicológicos y programas de rehabilitación.
- Disminuir la necesidad de invertir en campañas de prevención intensivas.
- Generar un impacto positivo en la salud mental y el bienestar social de las personas jóvenes.

De esta forma, la relevancia del proyecto no se mide únicamente en términos económicos, sino también en su capacidad de generar bienestar, reducir riesgos sociales y ofrecer un aporte positivo a la sociedad.

Conclusión de costos

Por último, para cerrar esta sección de costos, se decidió reunir todos los valores analizados hasta el momento con el objetivo de obtener una estimación final del costo total del proyecto. En este resumen se tienen en cuenta tanto los costos asociados al acceso al modelo de lenguaje (LLM) a través de la API de Gemini, como los costos del equipo de desarrollo durante las diferentes etapas del trabajo.

El acceso al LLM representa un costo mensual, ya que depende del uso continuo del servicio, mientras que el costo del equipo de desarrollo se calcula para un período total de cinco meses, que incluye la captura de requisitos, el desarrollo, las pruebas y el despliegue del sistema. Se indica “no aplica” para servidor y dominio porque ninguno genera costos adicionales. La aplicación web utiliza el dominio incluido en el hosting de Vercel y la API se ejecuta en un servidor de la UNSL con acceso interno, por lo que no requiere un dominio propio.

En la **Tabla 2.5** se muestran todos los costos divididos por las categorías analizadas previamente, utilizando como referencia la cotización del dólar del Banco de la Nación Argentina del 24 de octubre, cuyo valor era de $1.515.

Acceso a LLM	**Servidor**	**Dominio**	**Total Equipo de Desarrollo**
$USD 1,79 / mes	No aplica	No aplica	$USD 29.314,80

Tabla 2.5. Estimación de costos totales en dólares

Capítulo 3

DESARROLLO INGENIERIL

Este capítulo documenta el proceso de construcción del asistente conversacional, siguiendo las etapas del ciclo de vida del software y los principios de desarrollo responsable de inteligencia artificial. Se inicia con la implementación de mecanismos de IA responsable, incluyendo la evaluación de impacto que identifica posibles amenazas y vulnerabilidades, junto con sus estrategias de mitigación. Además, se presenta un resumen de las políticas de privacidad y los términos y condiciones diseñados.

El capítulo continua con el desarrollo sistemático del software, documentando las etapas de captura de requerimientos funcionales y no funcionales, análisis, diseño, implementación y estrategias de despliegue mediante integración continua. Finalmente, se presentan las pruebas realizadas durante el desarrollo y previas al lanzamiento, incluyendo el test de aceptación con estudiantes universitarios que validó la usabilidad del sistema, métricas de rendimiento y experiencia de usuario obtenidas, y la evaluación del sistema RAG que muestra tanto la relevancia de la información recuperada como su correcta utilización por el asistente en la generación de respuestas.

3.1. Desarrollo Responsable de Sistema de Inteligencia Artificial

Siguiendo los lineamientos planteados por la Guía de Protección de Datos de la AAIP para una inteligencia artificial responsable, se desarrollan diferentes tareas que permiten diseñar instrumentos que serán utilizados en el sistema [12].

Durante el desarrollo del sistema, como se mencionó anteriormente, se adoptó la metodología XP. Esta metodología ágil favorece la colaboración constante, la revisión continua del código y la adaptación rápida ante cambios, lo que permite identificar y corregir posibles fallas o sesgos desde etapas tempranas. Las iteraciones cortas y las pruebas permanentes permitieron que, a lo largo del proceso de desarrollo, se tuvieran en cuenta los diversos principios de los que se habla en la Guía de Protección de Datos de la AAIP, garantizando que cada avance técnico contribuya a un sistema de IA más seguro y transparente.

3.1.1. Evaluación de Impacto

La Evaluación de Impacto en la Protección de Datos en la etapa de diseño, es una herramienta fundamental para identificar y mitigar riesgos asociados al tratamiento de datos personales en un sistema de IA desde el principio.

Por lo general aquellas organizaciones que deberían realizar una evaluación de impacto son las que tratan grandes volúmenes de datos, que realizan operaciones con datos sensibles [12].

Dado que trabajamos con una temática sensible y un público potencialmente vulnerable, es necesario evaluar los riesgos asociados al uso de IA para la prevención de problemas de adicción.

Entre las etapas de una evaluación de impacto está la “gestión de riesgos”, proceso mediante el cual se identifica, analiza y valora la probabilidad e impacto de las ocurrencias de amenazas. El objetivo

es establecer cuáles son las hipótesis de riesgo para luego, en una etapa posterior, definir el plan de tratamiento necesario para minimizarlos.

La relación entre los riesgos y la probabilidad de impacto hace posible construir una matriz que permite anticipar si el sistema que se está diseñando es de alto, medio o bajo riesgo. Un proyecto que en el diseño dé como resultado una matriz de alto riesgo probablemente deba ser desestimado y rediseñado.

La fórmula para la evaluación de riesgos propuesta por la AAIP es:

$$Riesgo = Probabilidad \quad x \quad Impacto \tag{3.1}$$

La *probabilidad* hace referencia a la frecuencia con la que una amenaza puede llegar a materializarse. Se clasifica en los siguientes niveles:

- **Baja (1):** La amenaza podría ocurrir únicamente en circunstancias excepcionales.
- **Media (2):** La amenaza es poco frecuente, pero existe la posibilidad de que ocurra.
- **Alta (3):** Es probable que la amenaza se materialice al menos una vez al año.
- **Crítica (4):** Es muy probable que la amenaza ocurra varias veces en un mismo año.

El *impacto* se refiere al nivel de daño material o moral que puede producirse en caso de que la amenaza se concrete. Sus valores son los siguientes:

- **Bajo (1):** No se generan daños relevantes o estos pueden resolverse fácilmente. Ejemplo: sensación de invasión de la privacidad sin consecuencias objetivas.
- **Medio (2):** Los efectos no son significativos o la persona puede afrontarlos sin dificultad. Ejemplo: invasión de la privacidad sin un perjuicio grave.
- **Alto (3):** La persona experimenta consecuencias negativas que afectan su bienestar o reputación. Ejemplo: temor a usar un servicio, daños psicológicos leves, afectación de la reputación o el honor.
- **Crítico (4):** Se producen consecuencias graves o irreversibles. Ejemplo: daño psicológico permanente.

La **Figura 3.1**, extraída de "Evaluación de riesgo" de la AAIP [63], muestra la matriz de riesgo según la combinación de probabilidad e impacto.

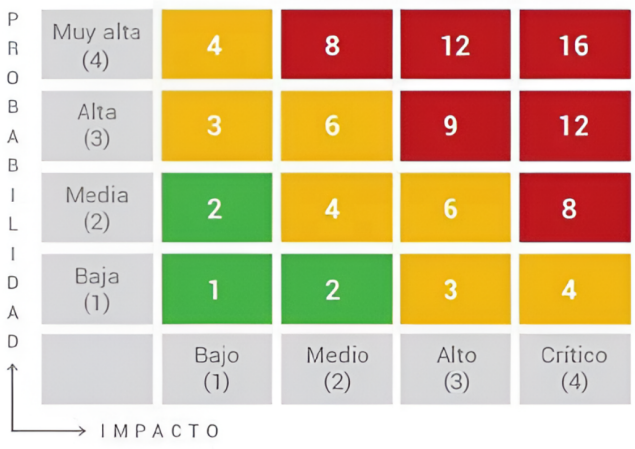

Figura 3.1. Matriz de riesgo.

Luego de analizar debidamente nuestro sistema, el público a quién está dirigido y los lineamientos que da la guía, definimos las siguientes amenazas.

- **Sesgo de activación:** Se produce cuando las salidas del sistema se utilizan en el entorno de manera sesgada.
- **Apego emocional:** La dependencia emocional se da cuando una persona busca estabilidad emocional en algo o alguien externo y, en este caso, esa búsqueda recae en la IA. Con el tiempo, el usuario puede preferir este "soporte" artificial por encima de las relaciones humanas, pues la IA siempre está disponible, no juzga y se adapta a sus necesidades.[64]
- **Daño psicológico:** La IA, si no se especifica de forma correcta, puede generar respuestas con índole negativa, ofensiva o discriminatoria. Estas respuestas pueden afectar la autoestima y el bienestar emocional del usuario.
- **Re identificación de la persona:** Cuando los datos del usuario no son adecuadamente seleccionados, anonimizados o procesados, existe la posibilidad de que se logre identificar indirectamente a la persona que interactuó con la IA, comprometiendo su privacidad y exponiéndola a riesgos de seguridad o uso indebido de su información.
- **Pérdida de integridad de los mensajes:** Existe la posibilidad de que los mensajes enviados por el usuario sean alterados, interceptados o manipulados durante su transmisión o almacenamiento. Esto podría exponer datos sensibles, generar accesos no autorizados o provocar divulgación indebida de información confidencial.

Para analizar estas amenazas se elaboró la **Tabla 3.1**, donde se detallan los valores de Impacto y Probabilidad de Ocurrencia junto con su Riesgo total.

Amenaza	**Vulnerabilidad**	**Impacto (I)**	**Probabilidad Ocurrencia (P)**	**Riesgo (P x I)**
Sesgo de activación	Comportamiento mal configurado del modelo y falta de información	Alto (3)	Alto (3)	Alto (9)
Re identificación de personas	Guardado de datos de conversación	Alto (3)	Alto (3)	Alto (9)
Daño psicológico	Comportamiento mal configurado del modelo	Muy alto (4)	Alto (3)	Alto (12)
Apego emocional	Comportamiento mal configurado del modelo y falta de especificación de comunicación con IA	Alto (3)	Muy alta (4)	Alto (12)
Pérdida de integridad de los mensajes	Posibilidad de acceso y modificación de los datos	Muy alto (4)	Medio (2)	Alto (8)

Tabla 3.1. Amenazas con su riesgo

Estos riesgos podrían tener un mayor impacto en personas que se encuentran en una situación especial de vulnerabilidad, tales como, niños, niñas y adolescentes, géneros y disidencias, personas con discapacidades, minorías étnicas y raciales, personas de la tercera edad, personas en situación de pobreza, entre otras.

Por cada una de las amenazas con riesgo alto se proponen mecanismos de control para minimizarlos:

- **Sesgo de activación:** Se configuró el comportamiento del asistente de forma que sea crítico y piense antes de dar una respuesta final. Además, con las respuestas que se obtienen del RAG se puede asegurar que las respuestas que da el asistente son correctas, validadas por información provista por el equipo de Psicología.
- **Re Identificación de personas:** Solo se almacenan los datos estrictamente necesarios para el funcionamiento del sistema. Además, los mensajes de los usuarios se eliminan tras un período de tiempo determinado, reduciendo el riesgo de exposición o vinculación con una identidad específica.
- **Daño psicológico:** Se configuró el comportamiento del modelo, orientándolo a generar respuestas amigables, respetuosas y ajustadas a la información validada por profesionales de la Psicología, minimizando el riesgo de respuestas dañinas.
- **Apego emocional:** El comportamiento definido limita al modelo a mantenerse en la temática de interés, redirigiendo al usuario cuando intenta desviarse. Adicionalmente, se implementaron restricciones de tiempo para cada sesion previniendo asi la dependencia hacia la IA.
- **Pérdida de integridad de los mensajes:** Los mensajes obtenidos del usuario se cifran utilizando AES-256-GCM [65], el cual es un algoritmo de cifrado simétrico del Estándar de Cifrado Avanzado (AES) que usa claves de 256 bits y el modo GCM (Modo Galois/Contador) que garantiza confidencialidad, integridad y autenticidad de los datos. Además, todas las comunicaciones entre el cliente y el servidor se realizan a través de HTTPS, asegurando que los datos transmitidos estén protegidos contra interceptaciones y manipulaciones.

3.1.2. Políticas, términos y condiciones

Políticas de Privacidad

La definición de las políticas de privacidad tiene en cuenta los ítems propuestos por la guía:

- **Privacidad por Diseño y por Defecto:** Solo se recopila y procesan datos personales estrictamente necesarios con su debido consentimiento.
- **Minimización y Calidad de los Datos:** Se minimiza la cantidad de datos procesados.
- **Licitud y Consentimiento:** Se permit el uso solo con el consentimiento libre e informado del usuario, el cual es presentado antes de acceder al asistente conversacional.

Las políticas de privacidad fueron diseñadas considerando los principios establecidos en la guía de la AAIP y se encuentran publicadas en la plataforma web con acceso a ellas en todo momento. A continuación se resumen los aspectos principales:

Finalidad del tratamiento de datos

- Investigar y analizar científica y psicológicamente los hábitos de juego online y factores de riesgo asociados al consumo abusivo o ludopatía.

- Evaluar la efectividad, aceptación y utilidad del asistente virtual en las intervenciones conversacionales.
- Realizar un análisis estadístico de resultados con fines académicos en formato completamente anónimo. Los datos no se utilizarán con fines comerciales, publicitarios ni se compartirán con terceros ajenos al proyecto.

Datos recopilados y procesados

- *Proporcionados por el usuario*
- *Generados automáticamente*
- *Datos técnicos*

Almacenamiento y ubicación de los datos

- *Aplicación web:* Servidores en la nube de Vercel [38].
- *Base de datos:* Neon (PostgreSQL en la nube) [47].
- *Estadísticas:* Servidor local de la Universidad Nacional de San Luis.

Tiempo de conservación

- *Mensajes del chat:* Se conservan durante 30 días desde su envío, tras lo cual se eliminan automáticamente.
- *Estadísticas anónimas:* Se conservan indefinidamente para fines de investigación científica, histórica y estadística.

Medidas de seguridad informática

- *Cifrado en tránsito:* Protocolos HTTPS en todas las comunicaciones.
- *Cifrado en reposo:* Mensajes almacenados cifrados.
- *Control de acceso:* Solo el administrador del sistema y los investigadores autorizados con credenciales específicas tendrán acceso a los datos.

Servicios de terceros

- *Google Generative AI (API de Gemini):* Procesa las conversaciones sin vincularse con información personal identificable.
- *Vercel Analytics:* Métricas técnicas anónimas sobre el uso del sistema.

Anonimización y limitaciones

- Se aplican procesos de anonimización para garantizar la protección de identidad.
- El asistente funciona de manera anónima sin requerir registro de usuario.
- Una vez anonimizados, los datos no podrán ser identificados ni eliminados de forma individualizada.

Derechos del usuario y contacto

- Se proporciona un correo electrónico de contacto para consultas sobre privacidad y ejercicio de derechos.
- La aceptación de las políticas se realiza previo a utilizar el asistente, confirmando que el usuario ha leído, comprendido y aceptado los términos.

Las políticas de privacidad completas se presentan en el **ANEXO I**.

Transparencia y Explicabilidad

Los términos y condiciones fueron diseñados para proporcionar transparencia sobre el funcionamiento del sistema, sus limitaciones y las responsabilidades de los usuarios. Estos términos son publicados en la plataforma web y deben ser aceptados antes de utilizar el asistente. A continuación se resumen algunos de los aspectos principales:

Naturaleza del sistema

- Es una aplicación web que forma parte de una investigación académica como herramienta para un proyecto de Psicología.
- El sistema tiene carácter experimental y de investigación científica, enfocado en la prevención de adicción al juego online en adolescentes.
- Tiene como propósito brindar información educativa, orientación y apoyo preventivo relacionado con juego responsable, apuestas online y prevención de conductas problemáticas.
- No sustituye la intervención profesional directa ni constituye una herramienta de diagnóstico, terapia o tratamiento psicológico.

Funcionamiento con inteligencia artificial

- El asistente funciona mediante el modelo de lenguaje Gemini 2.5 Flash, complementado con RAG.
- Las respuestas se basan en documentos científicos, guías clínicas, investigaciones académicas y material de organizaciones especializadas, validados por el grupo de psicólogos.
- El comportamiento del asistente fue configurado mediante un prompt específico diseñado en colaboración con profesionales de la psicología, y posteriormente revisado y aprobado por especialistas.

Limitaciones del sistema

- No reemplaza la atención de un profesional de la salud mental, ni constituye terapia psicológica, asesoramiento médico o diagnóstico clínico.
- Está específicamente diseñado para tratar temas de prevención de adicción al juego online y apuestas digitales. No debe ser utilizado para consultas sobre otros temas.
- No está diseñado para gestionar crisis o emergencias. En esos casos, proporcionará números de contacto de servicios de emergencia según la ubicación geográfica.
- El asistente puede cometer errores o proporcionar información inexacta debido a las limitaciones inherentes de los sistemas de IA.

Edad mínima y requisitos de uso

- Para utilizar el asistente se debe tener al menos 13 años de edad. Los usuarios entre 13 y 17 años deberán contar con autorización expresa de un padre, madre o tutor legal.
- Se requiere consentimiento libre, informado y expreso previo al uso del asistente.

Conductas prohibidas

- Manipular el asistente, utilizar bots o herramientas automatizadas, enviar mensajes repetitivos o intentar acceder a bases de datos.
- Proporcionar información falsa, suplantar identidades, acosar, intimidar o usar lenguaje ofensivo.
- Introducir contenido ilegal o usar el sistema con fines comerciales no autorizados.

También se incluyen otros aspectos, tales como limitación de responsabilidad, los cuales se encuentran detallados en el texto completo de Términos y Condiciones, mostrado en el **ANEXO II**.

3.2. Etapas del ciclo de desarrollo de software

El desarrollo de software implica llevar a cabo una serie de actividades organizadas y estructuradas según la metodología adoptada para producir software de alta calidad. Como se introdujo en el capítulo anterior, se adoptó XP como metodología ágil para el desarrollo del presente sistema. Si bien XP se caracteriza por su naturaleza iterativa e incremental, el proceso de desarrollo sigue cumpliendo con actividades estructurales fundamentales que son parte integral del proceso de software y se basan en modelos genéricos diseñados para facilitar la producción de software de alta calidad.

Estas actividades estructurales generales son: comunicación, planificación, modelado, construcción y despliegue. A continuación, se describen brevemente cada una de estas actividades:

- **Comunicación:** El objetivo es comprender las metas de los involucrados en el proyecto (stakeholders) y recopilar los requisitos que contribuyan a definir las propiedades y capacidades del software.
- **Planificación:** Implica la elaboración de un plan detallado para el proyecto. Comprende identificar los posibles riesgos, especificar los recursos necesarios, determinar los resultados o productos del trabajo a obtener y establecer un plan de actividades en un cronograma.
- **Modelado:** Supone la creación de representaciones visuales o abstractas que ayudan a comprender y diseñar el sistema. Facilita la comunicación entre los miembros del equipo y permite una comprensión más clara de cómo se desarrollará el software antes de la implementación.
- **Construcción:** Conlleva el desarrollo de código, se desarrolla el software real siguiendo los diseños y especificaciones previamente establecidos. Se transforman los conceptos y diseños en un producto de software funcional.
- **Despliegue:** Se implementa y pone en funcionamiento el software en el entorno de producción. El software puede ser entregado como entidad completa o como un incremento parcialmente terminado. El consumidor lo evalúa y da retroalimentación.

En el contexto de XP, estas actividades no se ejecutaron de manera secuencial y rígida, sino que se desarrollaron de forma iterativa y entrelazada. Cada iteración incluyó elementos de todas estas actividades, permitiendo la evolución continua del sistema mediante entregas incrementales y retroalimentación constante.

Durante el desarrollo del proyecto se llevaron a cabo reuniones periódicas de 30 minutos a 1 hora de duración con los directores del proyecto y los stakeholders (equipo de psicólogos en nuestro caso) para asegurar que los requisitos y expectativas se mantuvieran alineados con el progreso del desarrollo. Estas reuniones permitieron ajustar y refinar los objetivos del proyecto a medida que se avanzaba en cada iteración.

A continuación, se detallan las etapas específicas llevadas a cabo durante el desarrollo del asistente conversacional. Estas etapas se organizan siguiendo las actividades estructurales mencionadas y se complementan con las recomendaciones de la Guía de la AAIP para el ciclo de vida de sistemas de IA. La **captura de requerimientos** corresponde a la actividad de comunicación; el **análisis** y **diseño** forman parte del modelado; la **implementación** constituye la construcción del software; y finalmente, las **pruebas** validan el correcto funcionamiento antes del despliegue.

3.2.1. Captura de requerimientos

La etapa de captura de requerimientos fue la primera del proceso de desarrollo. En esta instancia, luego de diversas reuniones con el equipo de Psicología, se logró definir con claridad qué se espera del sistema y cuáles son los aspectos que resultan imprescindibles para su correcto funcionamiento. El trabajo conjunto permitió identificar tanto los requerimientos funcionales, relacionados directamente con las acciones que el sistema debe realizar, como los requerimientos no funcionales, que establecen las condiciones de calidad, rendimiento y seguridad esperadas.

En las reuniones con el equipo se propuso nombrar al asistente “No Va Más”, jugando con la frase típica que se utiliza en una ronda del casino cuando no se aceptan más apuestas. Para darle un tono más amigable y moderno, se decidió modificar el nombre a “NoVa+”, que no se encuentra registrado como marca en el Instituto Nacional de Propiedad Industrial y que, desde este punto en adelante, será la forma de referirse al asistente.

Requerimientos funcionales

Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares.[18]

- **Interacción con la IA**

  El sistema debe permitir al usuario mantener una conversación fluida con una IA, que funcione como asistente virtual y brinde acompañamiento emocional dentro de los límites definidos por el equipo de Psicología.

- **Mantenimiento del contexto**

  El asistente debe conservar el contexto de la conversación dentro de una misma sesión, de manera que las respuestas resulten coherentes y personalizadas.

- **Sesiones grupales**

  El sistema debe permitir crear sesiones grupales garantizando la separación de los distintos entornos.

- **Administración de usuarios**

  El sistema debe permitir gestionar las sesiones activas y la información básica de los usuarios, asegurando la correcta asignación de roles y el control de acceso.

- **Generación de estadísticas**

  El sistema debe poder generar y visualizar estadísticas sobre el uso del asistente, incluyendo la cantidad de sesiones, los niveles de riesgo y los tipos de apuestas mencionadas por los usuarios, siempre de forma anónima.

- **Exportación de datos**

  Las estadísticas generadas deben poder descargarse en formato Excel, para su análisis posterior por parte del equipo de Psicología.

- **Integración de base de conocimiento**

  El asistente debe tener acceso a una base de conocimiento validada por profesionales, que sirva como fuente de información segura para brindar respuestas precisas y responsables.

- **Detección de riesgo**

  El sistema debe ser capaz de detectar sesiones que indiquen un posible nivel de riesgo alto en el usuario y en esos casos, ofrecer recursos de ayuda profesional o derivaciones pertinentes.

Requerimientos no funcionales

Son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Los requerimientos no funcionales a menudo se aplican al sistema en su totalidad.[18]

- **Usabilidad**

  Hace referencia a la facilidad con la que los usuarios pueden comprender, aprender y utilizar el sistema, así como la satisfacción que obtienen al hacerlo. Un software usable debe ofrecer una interfaz intuitiva, coherente y accesible. Dentro de este requerimiento se analizan tres aspectos fundamentales:

  - **Protección contra errores de usuario:** el sistema debe minimizar las posibilidades de error y, cuando éstos ocurran, ofrecer mensajes claros y mecanismos de recuperación. En este proyecto se logró mediante la validación de datos en los campos de texto y la presentación de mensajes informativos cuando ocurre un error.
  - **Facilidad de aprendizaje:** capacidad del sistema para que los usuarios, incluso sin experiencia previa, puedan aprender a usarlo de manera eficiente y en un tiempo reducido. Se refiere específicamente al esfuerzo y tiempo que requiere un nuevo usuario para familiarizarse con el sistema y alcanzar un nivel básico de competencia. Para cumplir con esto, se diseñó una interfaz con una estructura simple y elementos distribuidos, procurando obtener un estilo visual amigable que facilite la interacción inicial.
  - **Facilidad de uso:** capacidad para que los usuarios puedan realizar tareas de manera efectiva, eficiente y satisfactoria una vez que ya conocen el funcionamiento básico. A diferencia de la facilidad de aprendizaje que se centra en la curva inicial, este aspecto evalúa la experiencia continua durante el uso habitual del sistema. En el proyecto se procuró definir una interfaz consistente que además brindará respuestas rápidas del asistente.

- **Privacidad**

  Este requerimiento se centra en el respeto y la protección de los datos personales de los usuarios. En el proyecto, se lleva a cabo el tratamiento de la privacidad evitando la recopilación de datos identificables y aplicando políticas claras sobre el uso y retención de la información.

  - **Anonimización de la información del usuario:** se realiza evitando almacenar cualquier dato que permita identificar al usuario. Para esto se siguen los lineamientos especificados por la guía [12]. Las conversaciones se guardan de forma anónima, sin nombres, números de identificación u otra información personal. Además, las estadísticas generadas a partir de los chats no permiten inferir la identidad de los participantes.
  - **Retención y eliminación de mensajes:** los mensajes generados durante las conversaciones se conservan indefinidamente en la etapa de prueba para facilitar el desarrollo y ajuste del sistema. Una vez en funcionamiento final, los mensajes serán almacenados únicamente durante 30 días, tras lo cual se eliminarán de manera automática, limitando la exposición de información sensible.

- **Seguridad**

  Se refiere a la capacidad del sistema de proteger la información y los datos frente a accesos no autorizados, modificaciones indebidas o pérdidas. Este requerimiento es clave, dado que el sistema maneja conversaciones que pueden incluir información sensible. Se implementaron diferentes mecanismos para contribuir a la seguridad de los datos y la privacidad de los usuarios.

  - **Confidencialidad:** los mensajes se almacenan encriptados, de modo que no puedan ser leídos por personas no autorizadas. Solo los administradores registrados pueden acceder al sistema y el acceso a los datos está restringido mediante autenticación segura. Además, las APIs utilizadas están protegidas y solo pueden ser invocadas desde el servidor.
  - **Integridad:** se asegura que los datos no puedan ser modificados o eliminados de forma indebida. Para esto, el sistema controla los permisos de escritura en las bases de datos y valida las operaciones de guardado y lectura de información. Además, como los mensajes están encriptados, cualquier alteración indebida puede detectarse fácilmente, ya que el proceso de desencriptado o la verificación de integridad del mensaje fallarían si el contenido fue modificado.

- **Disponibilidad**

  Representa la capacidad del sistema para mantenerse accesible y operativo en todo momento. El asistente se encuentra alojado en la nube mediante la plataforma Vercel, lo que proporciona una alta disponibilidad, tolerancia a fallos y escalabilidad automática. Además, la infraestructura permite monitorear el rendimiento y detectar interrupciones en tiempo real, reduciendo el riesgo de caídas del servicio.

- **Performance**

  Mide la eficiencia con la que el sistema ejecuta sus operaciones bajo distintas condiciones de carga. El proyecto fue diseñado para ofrecer tiempos de respuesta rápidos, incluso con múltiples usuarios concurrentes, optimizando las consultas a la base de datos y las llamadas al modelo de IA. La comunicación entre los componentes está optimizada mediante el uso de APIs y server actions, lo que mejora la velocidad general del sistema.

- **Escalabilidad**

  El sistema puede aumentar su rendimiento y capacidad operativa a medida que crece la cantidad de usuarios. Gracias al despliegue en la nube, es posible ampliar los recursos sin afectar la estabilidad o el funcionamiento del sistema. La arquitectura modular permite además incorporar nuevas funcionalidades o servicios, como la integración de modelos adicionales o módulos de análisis, sin necesidad de rediseñar el sistema principal.

- **Conformidad**

  Hace referencia al cumplimiento de normas, estándares y regulaciones aplicables en materia de protección de datos y buenas prácticas de desarrollo. El sistema cumple con principios establecidos en la AAIP, priorizando el manejo ético de los datos de los usuarios, la transparencia y la confidencialidad.

- **Mantenibilidad**

  Capacidad del sistema para ser modificado, actualizado o corregido de forma rápida y efectiva, con el menor impacto posible sobre el resto del sistema. El proyecto fue estructurado siguiendo una arquitectura modular que permite realizar cambios localizados en componentes sin afectar el funcionamiento general.

  - **Modularidad:** el sistema está dividido en múltiples módulos, incluyendo APIs, componentes y server actions, lo que permite aislar las responsabilidades de cada parte. Esto facilita las tareas de depuración, mantenimiento y ampliación de funcionalidades.
  - **Reusabilidad:** los componentes fueron diseñados para poder ser reutilizados en diferentes partes del sistema o incluso en proyectos futuros. Esta estrategia reduce el tiempo de desarrollo, mejora la consistencia visual y funcional y contribuye a la sostenibilidad del código.

A partir del estudio de estos requerimientos se pudo conceptualizar con mayor detalle el dominio sobre el cual se desarrolla el proyecto, descripto gráficamente en la **Figura 3.2**.

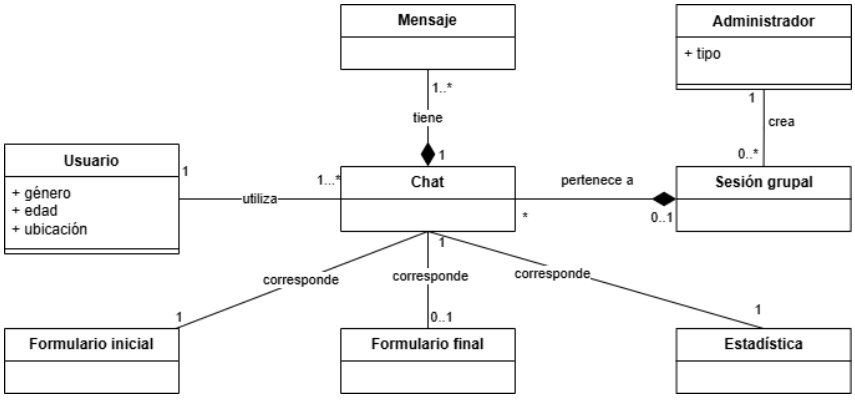

Figura 3.2. Modelo de Dominio.

Dentro del dominio de nuestro proyecto se encuentra el concepto más importante, el chat. El chat hace referencia a la conversación que el usuario tiene con el asistente. A este se relacionan todos los otros conceptos del dominio, ya sea de forma directa o indirecta.

Cada chat se relaciona con la información relevante del usuario, como su edad, género y ubicación aproximada, manteniendo en todo momento su anonimato y cumpliendo con los criterios de privacidad mencionados previamente.

Además, cada chat cuenta obligatoriamente con un formulario inicial, el cual incluye un set de preguntas específicas que permiten estimar el nivel de riesgo del usuario y una estadística que describe

distintos aspectos analizados del chat. El chat puede opcionalmente derivar a un cuestionario final, que permiten evaluar cómo le resultó el asistente al usuario.

Por otra parte, cada chat puede estar relacionado con una sesión grupal, en caso de que haya sido creado dentro de un grupo particular. Las sesiones grupales permiten al equipo de Psicología diferenciar y dividir las estadísticas que se generen, permitiendo analizar de mejor forma los resultados y comportamientos específicos según los grupos que hayan participado.

3.2.2. Etapa de análisis

Una vez definidos y documentados los requerimientos funcionales y no funcionales, se dio inicio a la etapa de análisis. El objetivo de esta fase fue profundizar en la comprensión del comportamiento esperado del sistema, asegurando que respondiera adecuadamente a las necesidades y expectativas del equipo de Psicología. Durante el análisis, se examinó cada requerimiento para determinar cómo se reflejaría en el funcionamiento general del sistema e identificar posibles dependencias entre las distintas funcionalidades.

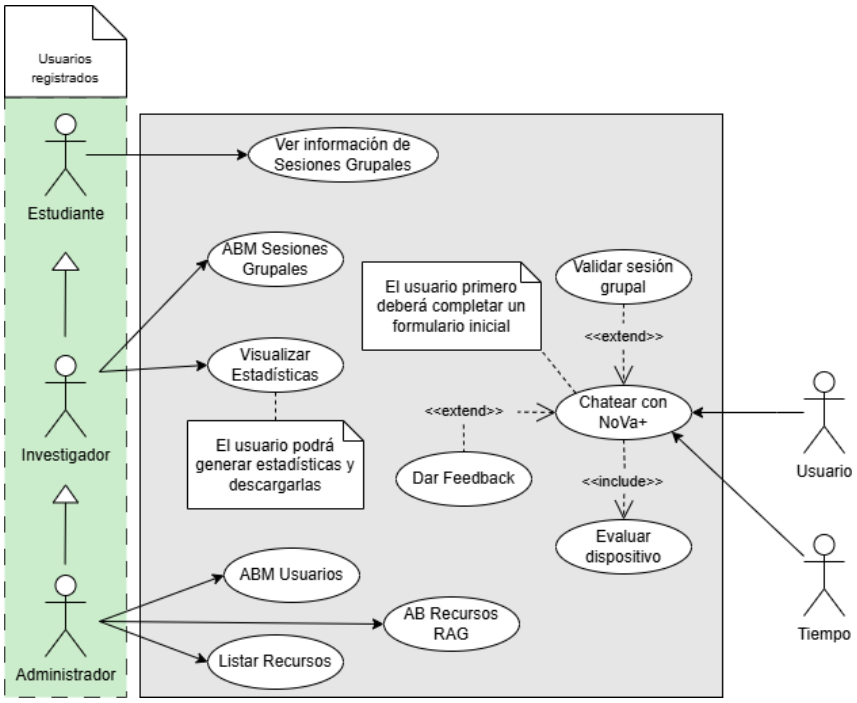

Figura 3.3. Diagrama de Casos de Uso.

A partir de los requerimientos, se pudieron definir los casos de uso del sistema. El diagrama de Casos de Uso resultante se puede ver en la **Figura 3.3**, éste permitió comprender el alcance del sistema de forma clara.

Entre los casos de uso especificados en la **Figura 3.3**, se encuentran los casos de uso principales “Chatear con NoVa+” y “Visualizar Estadísticas”.

Chatear con NoVa+

Este caso de uso representa una de las funcionalidades más importante del sistema, en el cuál el usuario puede comunicarse con “NoVa+”.

1. El caso de uso comienza con el usuario accediendo al sistema para hablar con el asistente; si el usuario pertenece a una sesión grupal necesita validar su sesión con contraseña.

2. El usuario completa un formulario inicial que sirve para estimar el nivel de exposición que éste tiene en relación a las apuestas en línea.

3. El usuario acepta los Términos y Condiciones y las Políticas de Privacidad y envía el formulario.

4. El sistema muestra el chat con "NoVa+" con un mensaje de bienvenida.

5. El usuario envía un mensaje a "NoVa+" y se inicia la conversación.

6. El usuario puede completar un comentario sobre la idea del asistente.

7. El caso de uso finaliza cuando el usuario decide finalizar el chat con "NoVa+" siempre que le quede tiempo disponible, sino después del tiempo estipulado se finaliza la sesión automáticamente.

Luego que el caso de uso finaliza comienza el caso de uso "Evaluar Dispositivo".

Visualizar Estadísticas

Este caso de uso se encarga de mostrar al usuario todas las estadísticas que se obtuvieron a partir de las conversaciones con el asistente. También permite generar estadísticas y descargarlas.

1. El sistema muestra las estadísticas generales de todas las conversaciones.

2. El usuario puede generar estadísticas nuevas.

3. El usuario puede ver las estadísticas de un grupo en específico.

4. El sistema muestra las estadísticas generales del grupo pedido.

5. El usuario puede ver todas las conversaciones de un grupo.

6. El usuario puede descargar las estadísticas.

7. El caso de uso finaliza con el usuario pudiendo ver las estadísticas de una conversación en particular.

3.2.3. Diseño del sistema

La etapa de diseño del sistema se abordó con el diseño arquitectónico del sistema. El diseño arquitectónico es la primera etapa en el proceso de diseño y representa un enlace crítico entre los procesos de ingeniería de diseño y de requerimientos. El proceso de diseño arquitectónico está relacionado con el establecimiento de un marco estructural básico que identifica los principales componentes de un sistema y las comunicaciones entre estos componentes.

Arquitectura general del proyecto

La arquitectura general que se implementó se representa en la **Figura 3.4**, donde se cuenta con dos servidores. El servidor "NoVa+" será el servidor principal desarrollado con el framework NextJS[39]. Este es el encargado de gestionar todas las solicitudes del lado del cliente y es donde se lleva a cabo la función principal del sistema que es el asistente conversacional con RAG. Por otro lado, se cuenta con el servidor "Analytics", desarrollado en Python, el cual es el encargado de calcular las estadísticas para cada una de las conversaciones de los usuarios.

Ambos servidores acceden a una base de datos relacional, donde se almacenan todos los datos relevantes para el funcionamiento de la aplicación web, incluyendo información de usuarios, sesiones grupales, sesiones de chat públicas y estadísticas.

Además, el servidor principal tiene acceso a la base de datos vectorial, necesaria para la técnica de RAG, a la cual recurre cuando se requiera cargar bibliografía relevante al contexto o recopilar información en función de una consulta específica del usuario.

Esta arquitectura fue diseñada teniendo en cuenta algunos de los requerimientos no funcionales más importantes mencionados en la etapa anterior, como la escalabilidad, performance, disponibilidad y la mantenibilidad del sistema.

El enfoque distribuido en dos servidores permite escalar cada componente de forma independiente según su carga específica: el servidor "NoVa+" puede dimensionarse para manejar la concurrencia de usuarios y consultas RAG, mientras que el servidor "Analytics" puede escalarse según la intensidad de los cálculos estadísticos, evitando que ambas cargas compitan por los mismos recursos computacionales.

Además, la separación de responsabilidades mejora significativamente la mantenibilidad al permitir que cada servidor utilice la tecnología más adecuada para su propósito: Next.js para la aplicación web, interfaz de usuario y gestión de conversaciones en tiempo real con su conjunto de herramientas para IA y Python con sus bibliotecas científicas para el procesamiento estadístico y análisis de datos. Esta arquitectura también incrementa la disponibilidad del sistema, ya que una falla en el servidor Analytics no afecta la funcionalidad principal de conversación del asistente. Por último, el rendimiento general mejora al procesar las estadísticas de forma asíncrona en un servidor dedicado, evitando que estos cálculos bloqueen o ralenticen las respuestas del asistente conversacional.

Si bien esta arquitectura distribuida introduce potenciales vulnerabilidades en las comunicaciones inter-servicios y en el acceso concurrente a las bases de datos, se implementaron medidas de seguridad como:

- Todas las comunicaciones entre servidores utilizan el protocolo HTTPS para asegurar confidencialidad e integridad en tránsito.
- El acceso a ambas bases de datos está protegido y las conexiones cifradas.
- Los datos sensibles almacenados en la base de datos PostgreSQL, como el contenido de conversaciones, se encriptan en reposo utilizando cifrado con algoritmos estándar (AES-256-GCM [65]).

Cumpliendo así con los requisitos de confidencialidad, integridad y privacidad de datos personales establecidos en los requerimientos no funcionales del sistema.

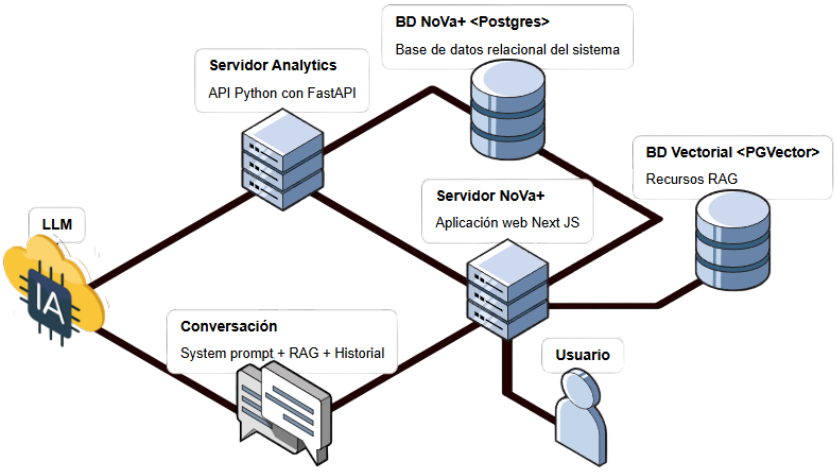

Figura 3.4. Esquema de la arquitectura general del Sistema.

Flujo de Datos de una conversación con NoVa+.

Por otro lado, en esta etapa del ciclo se decidió realizar un esquema que muestre de forma clara cómo fluyen los datos durante una conversación con "NoVa+". Este esquema permite entender el recorrido completo de la información desde que el usuario envía un mensaje hasta que recibe la respuesta del asistente.

El flujo de datos de una conversación con "NoVa+" se representa en la **Figura 3.5**. Cuando el usuario inicia una interacción y envía una pregunta o mensaje, esta información es procesada por el sistema mediante el sistema RAG.

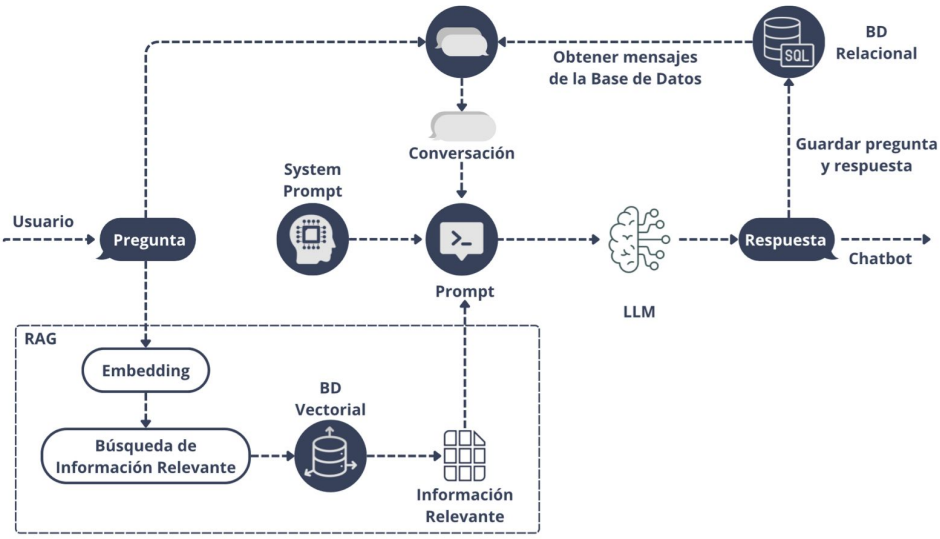

Figura 3.5. Esquema del Flujo de Datos de una conversación.

En primer lugar, la consulta del usuario se transforma en un embedding, que es una representación numérica del significado del mensaje. A partir de eso, el sistema compara dicho vector con los almacenados en la base de conocimiento para recuperar los fragmentos de texto más relevantes según su similitud semántica. Esos fragmentos contienen información validada y actualizada y se incorporan al prompt que se enviará al modelo, ampliando así su contexto y permitiendo que responda con datos más precisos.

Además, al prompt se le agrega el system prompt, que define el comportamiento del asistente, las reglas de interacción y los límites de sus respuestas, con el objetivo de que mantenga un tono adecuado al contexto del proyecto. También se incluye el historial de la conversación, lo que permite mantener la coherencia entre los mensajes y que el asistente recuerde lo que se habló previamente durante la sesión.

Finalmente, los fragmentos recuperados, el system prompt y el historial de mensajes se combinan y se envían al modelo para generar la respuesta. Esta respuesta se devuelve al usuario a través de la interfaz del asistente y tanto la pregunta como la respuesta se almacenan en la base de datos relacional, quedando registradas para su posterior análisis.

Diagramas de Secuencia

Los diagramas de secuencia son herramientas que permiten visualizar el flujo de las interacciones entre objetos durante la ejecución de un caso de uso, mostrando el orden cronológico y facilitando la comprensión de la lógica y las responsabilidades de cada componente. Debido a la complejidad de las interacciones entre los distintos objetos involucrados, es posible elaborar diagramas de secuencia para documentar cómo estos elementos interactúan entre sí.

Diagrama de Secuencia de Chatear con “NoVa+”

En la sección anterior se presentó en la **Figura 3.5** el flujo de datos que sigue una conversación con “NoVa+”. A partir de este esquema, se desarrolló el diagrama de secuencia del caso de uso “Chatear con NoVa+”, representado en la **Figura 3.6**. Este diagrama detalla más en profundidad la implementación y las interacciones entre los diferentes componentes del sistema durante una sesión de chat.

En el diagrama de flujo de datos se hizo énfasis en el recorrido de la información, mientras que en el diagrama de secuencia se destacan las interacciones entre los distintos módulos del sistema y cómo se coordinan para brindar la funcionalidad del asistente conversacional. Se incluyen las restricciones temporales que en un principio se omitieron para una comprensión más simple y abstracta de la realidad del sistema.

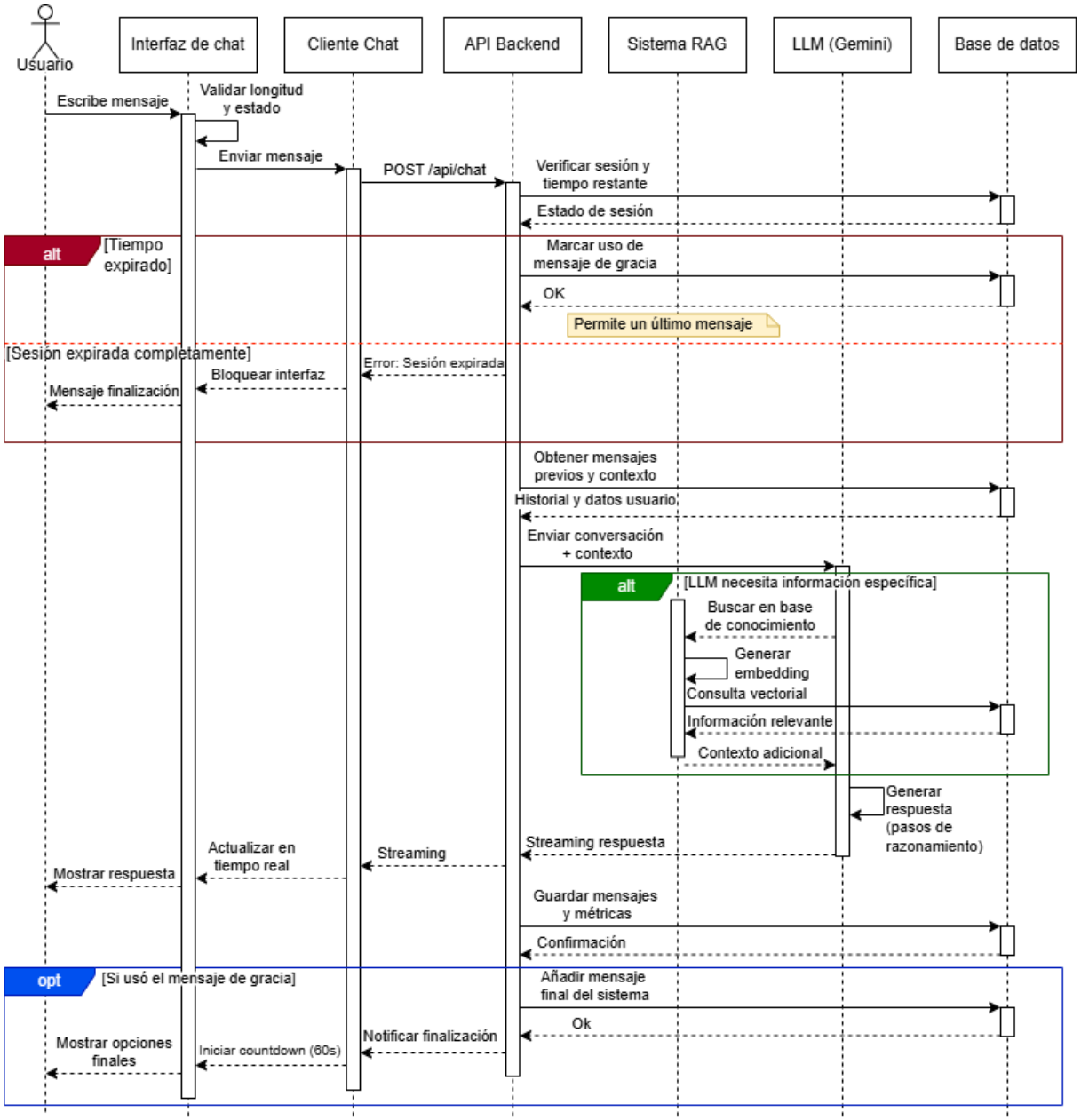

Figura 3.6. Diagrama de secuencia de “Chatear con NoVa+”.

El diagrama ilustra la secuencia completa desde el envío del mensaje por el usuario, hasta la recepción de la respuesta en la interfaz gráfica. El proceso inicia cuando el usuario escribe y envía un mensaje a

través de la interfaz de chat. El backend verifica que la sesión del usuario esté activa (en caso de ser sesión grupal) y su tiempo restante (tanto para sesiones públicas como grupales), consultando esta información en la base de datos. Si el tiempo expiró el usuario puede enviar un último mensaje final, por lo que continúa el flujo. Si la sesión ha expirado completamente, es decir finalizó el tiempo y ya se envió el mensaje final, el sistema bloquea la interfaz del chat y notifica al usuario sobre la finalización de su sesión.

Una vez validada la sesión, la API del backend recupera la información obtenida en el formulario previo para aumentar el contexto de la conversación y el historial de mensajes previos para mantener la coherencia del diálogo. En caso de ser necesario, se invoca al Sistema RAG, que se encarga de buscar información específica relevante para la conversación en la base de conocimiento vectorial. El sistema identifica el contexto adicional relevante y lo envía al LLM (Gemini), que genera una respuesta luego de un proceso de razonamiento. En caso de no ser necesario el uso de RAG, se genera la respuesta con el historial de mensajes y la información del usuario del formulario previo.

Finalmente, el LLM transmite la respuesta mediante streaming en tiempo real hacia el backend, que lo reenvía al cliente del chat para mostrarla progresivamente al usuario. Una vez completada la respuesta, se guarda tanto el mensaje del usuario como la respuesta generada, encriptados ambos, en la base de datos, junto con las métricas asociadas a la interacción, tales como tokens de entrada, de salida y de razonamiento, entre otros.

Diagrama de Secuencia de Visualizar Estadísticas

Luego de haber analizado en detalle las interacciones que se espera que se tengan en los diferentes casos de uso del sistema, se realizó el diagrama de Secuencia del caso de uso Visualizar Estadísticas, representado en la **Figura 3.7**.

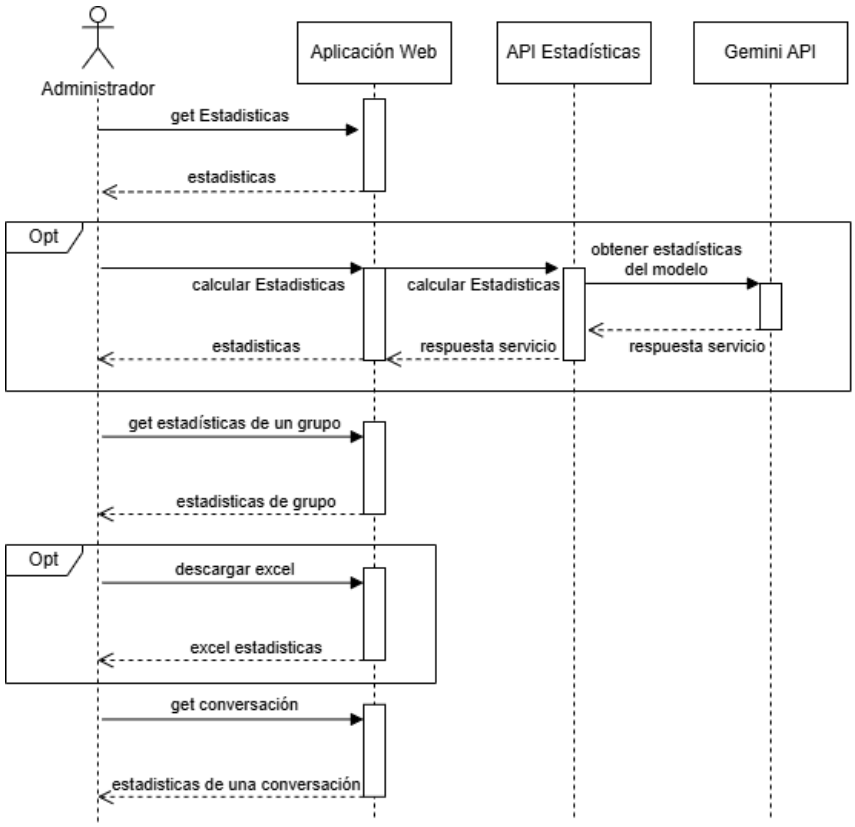

Figura 3.7. Diagrama de Secuencia de "Visualizar Estadísticas".

El flujo principal del caso de uso se basa en la muestra de estadísticas generadas a partir de las interacciones con el asistente. Estas estadísticas pueden visualizarse de tres formas diferentes: todas las estadísticas generales, las estadísticas generales de un grupo particular, o las estadísticas de conversaciones que se hayan hecho en la versión pública. En cualquiera de los casos, el tipo de información mostrada es la misma, mostrando datos como género, niveles de riesgo detectados, evaluación del asistente, entre otros indicadores relevantes para el proyecto del equipo de Psicología.

En caso de que existan sesiones que aún no han sido analizadas, el sistema le permite al administrador generar estas estadísticas, actualizando así los valores mostrados en pantalla.

El usuario puede, al seleccionar un grupo de prueba que desee analizar, acceder a un listado completo de todas las conversaciones que pertenecen a dicho grupo. Desde este listado, el usuario cuenta con la posibilidad de visualizar todas las estadísticas disponibles para cada conversación individual, accediendo así a toda la información de las estadísticas, formulario inicial y formulario final si es que se completó.

Además, tanto para las estadísticas generales como para las de un grupo particular, el sistema ofrece la posibilidad de exportar los datos a un archivo Excel. Este archivo replica toda la información disponible en la interfaz y se genera de forma automática, facilitando así el trabajo posterior del equipo de Psicología. El formato Excel fue elegido intencionalmente, ya que es una herramienta habitual para el equipo, permitiendo que puedan realizar análisis adicionales, aplicar fórmulas, filtrar información y detectar patrones de comportamiento en las conversaciones.

Etapas de procesamiento del sistema RAG

Para que el sistema RAG funcione de manera eficiente, no es recomendable cargar un PDF completo directamente en la base de datos vectorial. Si se guardara así, el sistema no podría realizar búsquedas precisas ni rápidas, porque el contenido sería muy extenso y difícil de utilizar para ampliar el contexto al asistente. Por eso, el sistema aplica una serie de transformaciones previas a los archivos ingresados. Todas estas etapas se muestran en la **Figura 3.8** y serán explicadas a continuación.

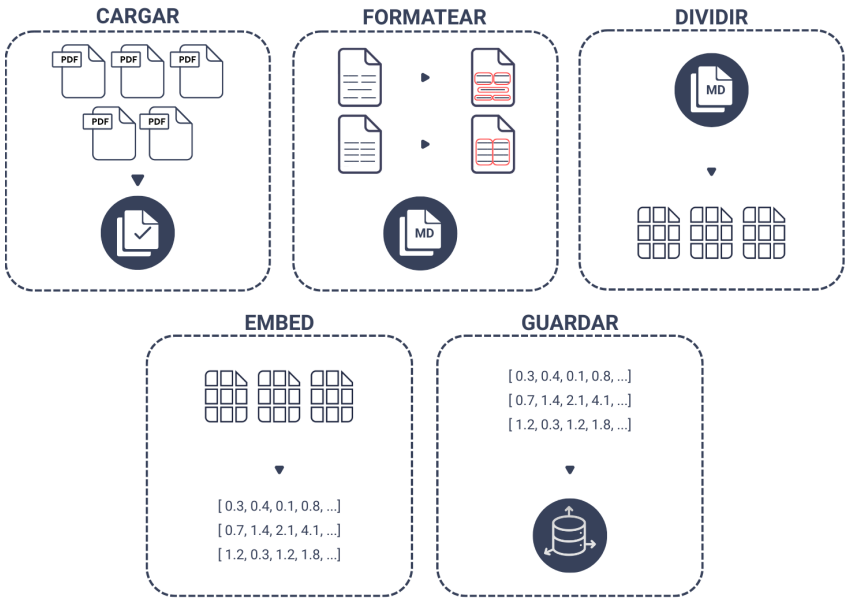

Figura 3.8. Esquema de las etapas del procesamiento de RAG.

El procesamiento comienza con el usuario cargando su archivo PDF por la página. Ese archivo es enviado a una API especializada que se encarga de convertirlo al formato Markdown. Markdown es un lenguaje basado en texto plano que permite representar títulos, listas, tablas y enlaces sin necesidad de mantener el diseño original del PDF. Esto es necesario porque la mayoría de los PDFs académicos cuentan con formatos que complican la lectura automática, como tablas, columnas dobles o encabezados

y pies de página repetidos. Al transformarlo a Markdown, el contenido queda formateado de una manera que es mucho más entendible para el asistente.

Una vez que el documento está convertido, se debe recortar el contenido, de forma que el asistente no tenga que tratar de entender un libro entero para una simple consulta. El contenido se divide en fragmentos más pequeños, llamados chunks. El sistema trata de hacer los cortes siguiendo límites naturales del texto, como el final de una oración o de un párrafo. En los casos en los que no es posible reconocer esas estructuras, se aplica un corte cada 600 caracteres. Además de esto, se utiliza una técnica de solapamiento entre fragmentos. Básicamente, una parte final del chunk anterior se repite al principio del siguiente. Esto evita que una idea quede fragmentada de forma brusca, y también ayuda a que el RAG mantenga el contexto cuando recupera los fragmentos más similares durante una consulta. Sin este solapamiento, el asistente podría perder información clave o interpretar frases fuera de su contexto original. Finalmente, una vez que todos los chunks están listos, se genera el embedding de cada uno.

3.2.4. Implementación

La etapa de implementación constituye la fase de construcción del sistema, donde se materializan las especificaciones y los diseños definidos en las etapas anteriores. Durante esta fase se desarrolla el código fuente, se configuran las bases de datos y se integran los diferentes componentes para dar forma al sistema completo. En esta sección se presentan los principales artefactos resultantes de la implementación: el diagrama entidad-relación que define la estructura de la base de datos, el diagrama de componentes y conectores que muestra la arquitectura en ejecución y el diagrama de despliegue que describe la distribución física del sistema en la infraestructura de producción.

Diagrama Entidad Relación

El diagrama Entidad Relación es una herramienta fundamental en el diseño de bases de datos, ya que permite representar gráficamente las entidades del sistema y las relaciones entre ellas, facilita el diseño, la documentación y la comunicación. En la **Figura 3.9** se presenta el diagrama de entidad relación de la base de datos utilizada en el proyecto, en este diagrama se pueden observar el esquema final de la base de datos, el cual fue escalando durante el desarrollo del proyecto, a medida que se iban definiendo, analizando y modelando nuevos requerimientos y funcionalidades.

A continuación, se describen brevemente, sin entrar en demasiado detalle, las principales entidades y sus relaciones:

- **ChatSession:** es la entidad central del sistema y del diagrama, representa una sesión de chat entre un usuario y el asistente “NoVa+”. Cada sesión está relacionada con un respectivo formulario de evaluación previo necesario para iniciar la conversación, los mensajes del chat, un formulario (opcional) de retroalimentación acerca de la idea de la aplicación (relevante para el equipo de desarrollo), un formulario final (opcional) de evaluación del asistente y su impacto en el usuario y una estadística que resume los datos analizados de la conversación. También puede o no estar relacionada con una sesión grupal de chat.

- **ChatGroup:** representa una sesión grupal de chat, donde múltiples usuarios pueden participar en conversaciones separadas pero dentro del mismo entorno controlado. Esta entidad permite organizar y gestionar las sesiones de chat en función de grupos específicos cuando el equipo de Psicología realice sus relevamientos en diferentes instituciones.

- **Message:** representa cada mensaje individual enviado durante una sesión de chat. Cada mensaje está asociado a una sesión específica y contiene información sobre el contenido del mensaje, el remitente (usuario o asistente), marca de tiempo e información relevante para calcular costos del uso del LLM.

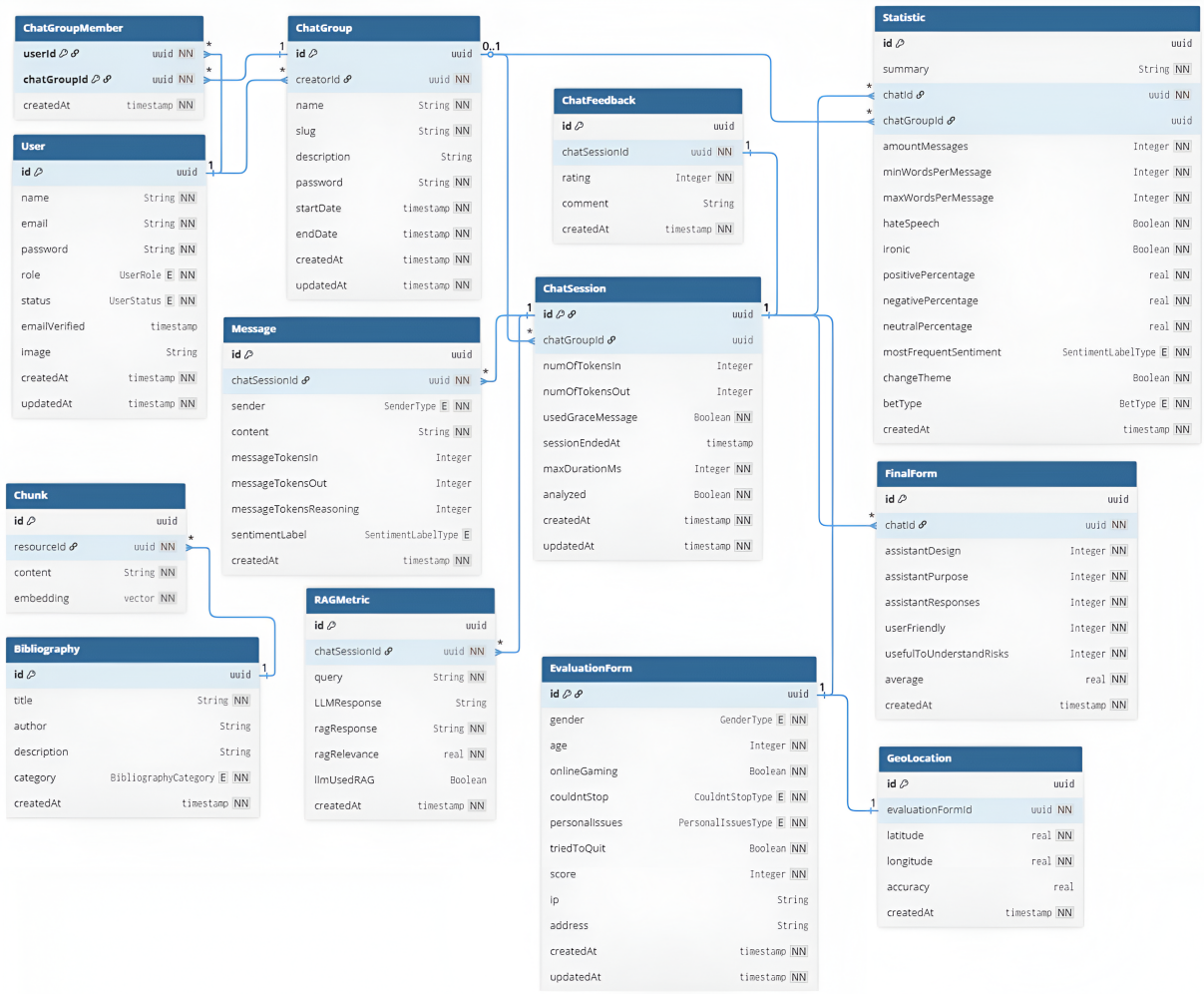

Figura 3.9. Diagrama Entidad Relación de la Base de Datos.

- **Bibliography:** representa los documentos que conforman la base de conocimiento utilizada por el asistente "NoVa+" para brindar respuestas informadas y precisas. Cada entrada en esta entidad contiene metadatos relevantes del documento.

- **Chunk:** representa los fragmentos individuales en los que se divide cada documento de la bibliografía para su almacenamiento en la base de datos vectorial. Cada chunk está asociado a un documento específico, contiene el texto del fragmento y su embedding.

- **Statistic:** representa las estadísticas generadas a partir de las conversaciones en una sesión de chat. Cada estadística está vinculada a una sesión específica y contiene diversas métricas solicitadas por el equipo de Psicología para evaluar de una mejor manera el uso del asistente. Entre las estadísticas principales se encuentra el tipo de apuesta, si trató de cambiar de tema, si usó lenguaje ofensivo o irónico, entre otros.

- **RagMetric:** representa las métricas específicas relacionadas con la técnica de RAG utilizadas durante las conversaciones. Cada métrica está asociada a una sesión de chat y proporciona información sobre el rendimiento y la efectividad del proceso RAG en esa sesión.

- **User:** representa a los usuarios registrados en el sistema. Estos son los posibles actores anteriormente mencionados como el administrador, investigadores o estudiantes que ayuden en los relevamientos del equipo de Psicología.

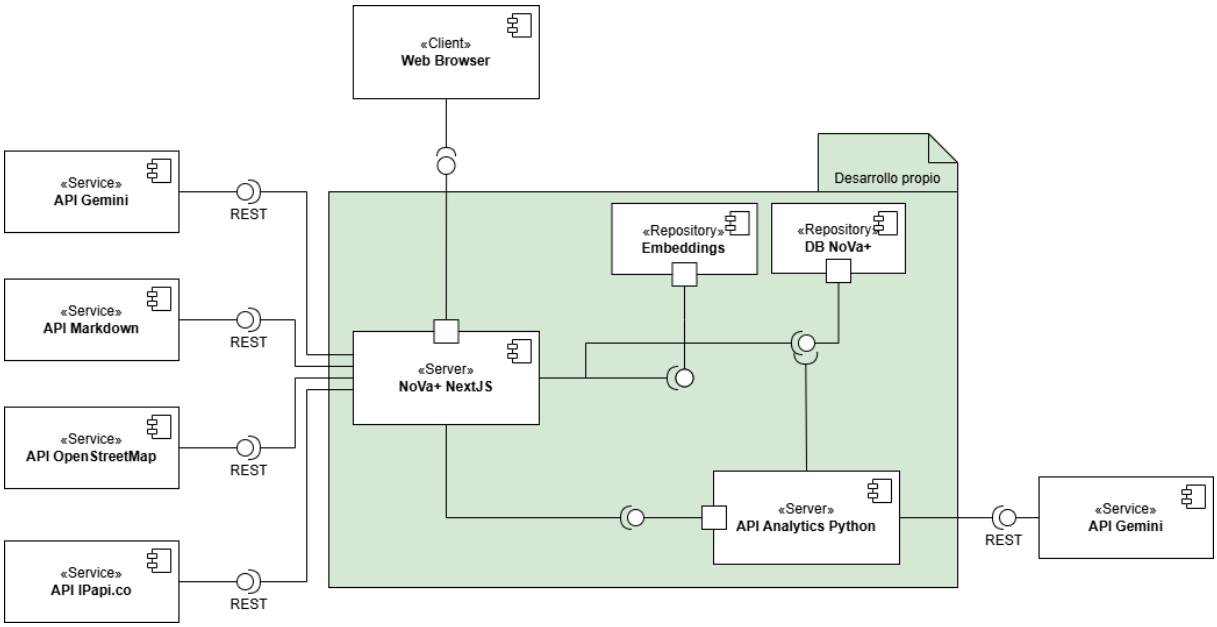

Figura 3.10. Diagrama Componente y Conector.

Diagrama Componente y Conector

Para comprender cómo los diferentes módulos del sistema interactúan durante su ejecución y cómo se comunican entre sí en un entorno de producción, se elaboró el diagrama de componentes y conectores. Este diagrama resulta fundamental para visualizar la arquitectura dinámica del sistema y entender los mecanismos de comunicación entre "NoVa+", el servidor de estadísticas, las bases de datos y los servicios externos.

Los diagramas de componentes y conectores expresan el comportamiento en tiempo de ejecución. Se describen en términos de componentes y conectores. Un componente es una de las principales unidades de procesamiento del sistema en ejecución. Los componentes pueden ser servicios, procesos, hilos, filtros, repositorios, pares o clientes y servidores, entre otros.

Un conector es el mecanismo de interacción entre los componentes. Los conectores incluyen tuberías, colas, protocolos de solicitud/respuesta, invocación directa, invocación dirigida por eventos, entre otros. Tanto los componentes como los conectores pueden descomponerse en otros componentes y conectores. La descomposición de un componente puede incluir conectores y viceversa.

En la **Figura 3.10** se presenta el diagrama Componente y Conector del sistema, donde los componentes dentro del recuadro representan los módulos desarrollados específicamente para el asistente "NoVa+", mientras que los elementos externos corresponden a servicios externos utilizados por el sistema. El componente principal, "NoVa+", es el punto de acceso de los usuarios y actúa como núcleo del sistema, siendo responsable de coordinar la interacción entre los diferentes componentes, como el servidor de Estadísticas, APIs utilizadas por el sistema y las bases de datos.

El diagrama Componente y Conector permite visualizar la estructura lógica del sistema en ejecución, identificando los componentes principales y sus mecanismos de comunicación. Los conectores representados utilizan principalmente protocolos REST sobre HTTPS para las comunicaciones entre "NoVa+" y los servicios externos, así como invocaciones directas para la interacción con las bases de datos. El servidor de Estadísticas se comunica con "NoVa+" mediante una arquitectura de solicitud/respuesta, procesando las conversaciones para generar las estadísticas requeridas por el equipo de Psicología.

Si bien este diagrama proporciona una visión general de los componentes y sus interacciones, no se profundizará en mayor detalle sobre aspectos específicos de implementación o comunicación, ya que el siguiente diagrama de despliegue complementará esta información mostrando la distribución física de es-

tos componentes en la infraestructura real, permitiendo una comprensión más completa de la arquitectura del sistema en su entorno de producción.

Diagrama de Despliegue

Los diagramas de despliegue describen la forma en que las unidades de software se mapean a los elementos del entorno en el que el software se desarrolla o se ejecuta. Dicho entorno puede ser el hardware, los sistemas de archivos que soportan el desarrollo o la implementación, o las propias organizaciones de desarrollo.

En la **Figura 3.11** se muestra el diagrama de Despliegue del sistema. Ya en la etapa de diseño se había definido una arquitectura distribuida con dos servidores principales. En este diagrama de despliegue se representa la arquitectura del sistema, utilizando la notación estándar de UML.

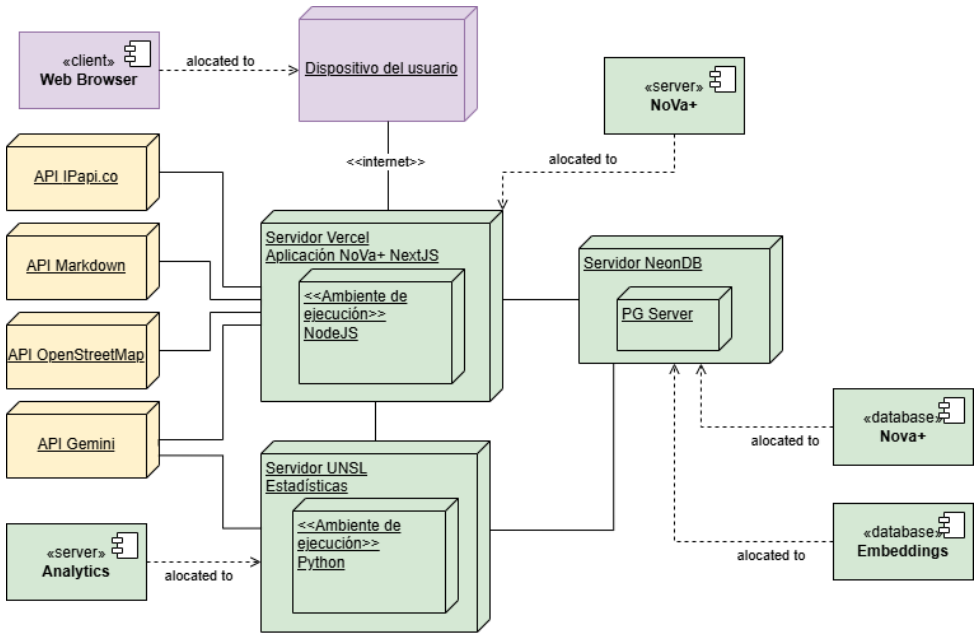

Figura 3.11. Diagrama de Despliegue.

En este diagrama se puede observar dónde se encuentran alojados los servidores.

- El servidor de la aplicación web “NoVa+” se encuentra alojado en la nube utilizando la plataforma Vercel.
- El servidor de estadísticas se encuentra alojado en un servidor físico en las instalaciones de la UNSL.
- La base de datos relacional PostgreSQL se encuentra alojada en la plataforma de NeonDB.
- La base de datos vectorial que también se encuentra alojada en la plataforma de NeonDB.

Si bien podría considerarse que esta distribución de servidores genera problemas de latencia o rendimiento, en la práctica no se observan estos inconvenientes debido a que los servidores de Vercel y NeonDB están desplegados en la misma región geográfica. La comunicación entre ambos es rápida y eficiente, por lo que el caso de uso principal del sistema, el chat con “NoVa+”, donde se requiere el mejor rendimiento, no se ve afectado por esta arquitectura.

Tanto en el diagrama de componentes y conectores de la **Figura 3.10** y en el diagrama de despliegue de la **Figura 3.11** se pueden observar los diferentes servicios externos que utiliza el sistema, donde podemos mencionar:

- El servicio de “API IPapi.co” utilizado para obtener información en base a una IP.
- El servicio de “API Markdown” utilizado para convertir archivos como PDFs en formato markdown de más fácil lectura para el LLM.
- El servicio de “API OpenStreetMap” utilizado para obtener información geográfica en base a coordenadas obtenidas de la ubicación del usuario.
- El servicio de “API Gemini” utilizado para acceder a las capacidades de Google generative AI mencionados en la sección de tecnologías.

Todas las comunicaciones con estos servicios utilizan una arquitectura REST y se realizan de forma segura mediante el protocolo HTTPS.

3.2.5. Despliegue, Operación y Mantenimiento

En la presente sección se describen los aspectos relacionados con el despliegue, operación y mantenimiento del sistema desarrollado. Se explica cómo se realizaron estos procesos para cada uno de los servidores mencionados y diagramados en la sección anterior.

Servidor de Estadísticas

El servidor de estadísticas fue desplegado utilizando Docker. El uso de Docker permitió estandarizar el entorno de ejecución, evitando inconsistencias entre la infraestructura de las computadoras de desarrollo y producción.

Como se mencionó anteriormente, para el análisis de los mensajes enviados se utilizó la librería PySentimiento [44], una librería basada en modelos Transformer capaz de realizar análisis de sentimientos, detección de ironía, emociones y discurso de odio. Esta genera problemas con el espacio, su instalación completa pesa aproximadamente 10 GB, debido a la gran cantidad de dependencias que incluye, entre las cuales se encuentran bibliotecas para aceleración por GPU, especialmente las relacionadas con NVIDIA. Luego del analisis de las dependencias que incluye la librería se observó que estas no son necesarias para el analisis de mensajes dado que el sistema no entrena modelos.

Este problema de espacio se volvió crítico debido a que el servidor provisto por la UNSL cuenta con un almacenamiento disponible reducido. Teniendo esto en cuenta, se tuvieron que realizar unas modificaciones adicionales al momento de configurar el Docker.

Docker puede construir imágenes de forma automática leyendo las instrucciones definidas en un Dockerfile. Un Dockerfile es un archivo de texto que contiene todas las instrucciones que un usuario podría ejecutar manualmente en la línea de comandos para construir una imagen. Describe paso a paso cómo debe ensamblarse el entorno, permitiendo crear imágenes reproducibles, portables y fáciles de mantener.

De esta forma, se definió un Dockerfile que instala únicamente las dependencias esenciales para el funcionamiento de la API, utilizando las versiones más livianas disponibles de PySentimiento. Además, incluye la copia del código fuente y las instrucciones necesarias para ejecutar la API. La imagen resultante se construyó mediante el comando docker build, que procesa todas las instrucciones del Dockerfile y genera una imagen lista para su despliegue.

Finalmente, el contenedor resultante fue ejecutado con la política –restart=unless-stopped, esta política indica que el contenedor debe reiniciarse automáticamente si se detiene inesperadamente o si el demonio Docker se reinicia, pero no si fue detenido manualmente.

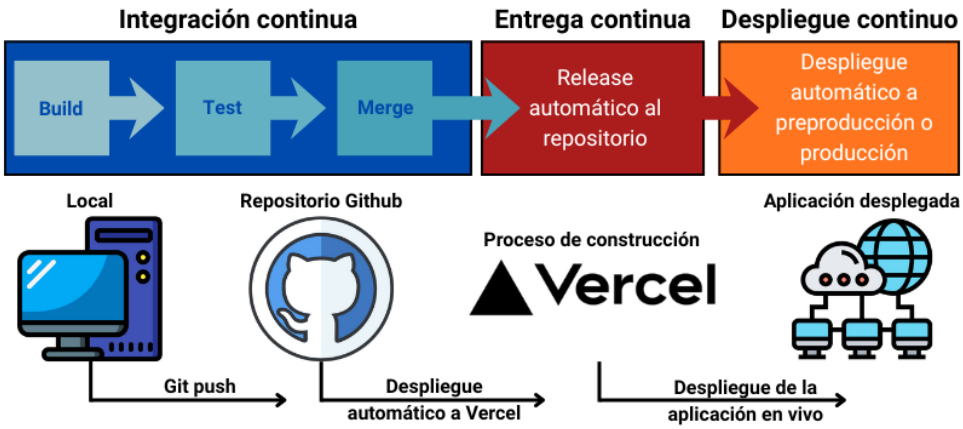

Figura 3.12. Esquema del proceso de despliegue en Vercel.

Servidor de aplicación web NoVa+

El servidor de aplicación web "NoVa+" está desplegado utilizando la plataforma Vercel, que ofrece un entorno optimizado para aplicaciones desarrolladas con Next.js. Desde etapas tempranas del desarrollo, se configuró el proyecto para su despliegue en Vercel, debido a sus capacidades de integración, entrega y despliegue continuo (CI/CD).

Se configuraron 2 ramas principales para el control de versiones en GitHub: una rama principal y una rama de desarrollo. La rama principal contiene la versión estable del sistema, mientras que la rama de desarrollo se utiliza para integrar nuevas funcionalidades y realizar pruebas antes de fusionarlas con la rama principal. Cuando se hace un commit en alguna rama, Vercel despliega los cambios automáticamente, en un enlace privado para el caso de desarrollo y en el dominio público de la aplicación cuando es en la rama principal. Esto asegura que la ultima version del sistema esté siempre disponible para los usuarios.

En la **Figura 3.12** se representa en forma de diagrama, el flujo del proceso llevado a cabo para el despliegue del sistema en Vercel.

El despliegue en dicha plataforma permite un acceso directo a funcionalidades como:

- Analíticas de uso y rendimiento del sistema.
- Monitoreo de errores y logs de la aplicación.
- Gestión de variables de entorno.
- Seguimiento de versiones desplegadas.

Estas herramientas facilitan significativamente las tareas de operación y mantenimiento del sistema, permitiendo identificar y resolver problemas rápidamente, así como optimizar el rendimiento y la disponibilidad del asistente. Además de estas ventajas, Vercel provee una dirección URL personalizada para la aplicación de forma gratuita, facilitando el acceso de los usuarios al sistema sin que se necesite una inversión inicial para el despliegue del sistema.

3.2.6. Pruebas

Las pruebas de software se aplican durante el proceso de desarrollo para detectar y corregir errores antes de que se conviertan en defectos visibles para los usuarios finales. Cuanto antes se identifique un problema, menor será el esfuerzo requerido para solucionarlo y menor el riesgo de que afecte otros componentes del sistema.

En términos de fiabilidad y calidad, un error de programación se define como el el defecto que resulta en la introducción de fallas en el sistema. Una falla (o defecto / bug) es una característica del software que, cuando se ejecuta, puede conducir a un error del sistema (un estado erróneo durante la ejecución). Ambos (fallas y errores) están relacionados con la falta de confiabilidad del software. El objetivo principal del testing es descubrir tantos defectos (bugs) como sea posible para demostrar que el producto es apto para su propósito y está listo para ser lanzado antes de que se ponga en uso [18].

Para el desarrollo de "NoVa+" se implementaron diferentes tipos de pruebas en distintas etapas del ciclo de vida, con el objetivo de garantizar la calidad, funcionalidad y usabilidad del sistema. Se realizaron pruebas continuas para validar que cada componente y funcionalidad implementada cumpla con los requerimientos definidos, haciendo énfasis especialmente en la calidad de las respuestas generadas por el asistente, ya que este aspecto es el más relevante del sistema. Las pruebas fueron llevadas a cabo tanto por el equipo de desarrollo, como también por equipo de Psicología, quienes validaron aspectos específicos como las respuestas de "NoVa+", dado que su conocimiento en el área de la salud mental es fundamental para asegurar la pertinencia y precisión de las mismas.

Pruebas del equipo de desarrollo

Durante el desarrollo de cada funcionalidad y caso de uso se realizaron pruebas unitarias y de integración para asegurar que cada módulo del sistema funcionara correctamente de forma aislada y en conjunto. Se implementaron pruebas automatizadas utilizando frameworks de testing compatibles con Next.js, lo que permitió ejecutar pruebas cada vez que se realizaban cambios en el código. Esto ayudó a identificar rápidamente cualquier error introducido durante el desarrollo y resolverlo.

Además, al permitir la creación de versiones de prueba (previews) en Vercel, se pudo validar nuevas funcionalidades en un entorno real pero controlado antes de integrarlas a la versión principal del sistema. Esto facilitó la detección temprana de problemas y permitió realizar ajustes necesarios basados en las observaciones del equipo de Psicología.

Pruebas del equipo de Psicología

Luego del primer despliegue funcional del sistema, con el "Mínimo producto viable", se generaron 3 usuarios diferentes para que el equipo de Psicología comience a interactuar con "NoVa+" y así evaluar la calidad de las respuestas generadas por el asistente. Los expertos participaron activamente en la validación de las respuestas generadas por "NoVa+". Se realizaron sesiones de prueba donde interactuaron con el asistente, evaluando la calidad, relevancia y adecuación de las respuestas en función del contexto y las necesidades del usuario.

Se recopilaron comentarios detallados sobre el asistente y se realizaron ajustes en el sistema basados en sus observaciones. Este proceso iterativo de prueba y ajuste fue sumamente necesario para mejorar la efectividad del asistente y asegurar que cumpliera con los estándares esperados por el equipo de Psicología. De estas retroalimentaciones se fue puliendo el "system prompt" y ajustando ciertos parámetros del modelo para mejorar la calidad de las respuestas.

Test de aceptación y prueba de estrés

Posterior a las pruebas de desarrollo, se llevó a cabo el "Test de Aceptación", que es el proceso de verificar que una versión de software cumple con los objetivos establecidos para el producto, buscando asegurar que este es eficiente y confiable. Generalmente, se describe como pruebas que realiza el cliente de un sistema para decidir si es adecuado para satisfacer sus necesidades. Es una forma de prueba de validación, cuyo propósito es demostrar que el software satisface las expectativas, incluso si van más allá de la simple conformidad con la especificación [18]. Es la última fase de pruebas en el desarrollo de software, donde los usuarios finales o el cliente verifican si el producto cumple con los requisitos del

negocio y las expectativas del usuario antes de su lanzamiento final. El objetivo principal es asegurar la satisfacción del cliente y validar que el producto final sea viable para su uso en un entorno real.

Dada la naturaleza crítica y sensible del proyecto, se decidió realizar el test de aceptación en un entorno controlado, lo más similar posible al entorno de usuarios reales, para evaluar el desempeño del sistema bajo condiciones de uso típicas. Como se mencionó anteriormente, el público principal que interactuará con "NoVa+" serán estudiantes de escuelas secundarias, por lo que se decidió realizar la prueba de aceptación en una materia de primer año de la universidad, con estudiantes voluntarios, todos mayores de edad para no solicitar autorización parental.

Los test mencionados se realizaron en 2 jornadas, en diferentes comisiones de la misma materia. Se dio una charla previa explicando el fin del proyecto y la problemática que busca atacar, luego, cada estudiante interactuó con "NoVa+" durante un período de 15 minutos en sesiones grupales, realizando preguntas y evaluando las respuestas generadas por el asistente. Tanto el equipo de desarrollo como el equipo de Psicología estuvieron presentes para observar, asistir y anotar posibles fallos no descubiertos en caso de ser necesario.

A continuación, se detalla información obtenida en el test de aceptación, obtenidas de las analíticas web implementadas en el sistema y valores obtenidos de la base de datos.

- Se registraron **92** visitantes en el período de las 2 jornadas con **493** páginas vistas (un mismo visitante puede ver la misma página varias veces, lo que genera múltiples eventos).
- Se crearon **125** sesiones de chat, con aproximadamente **2300** mensajes intercambiados entre usuario y asistente.
- El formulario de retroalimentación acerca de la idea de la aplicación obtuvo un promedio de **4.8 sobre 5** estrellas.
- El formulario final de evaluación del asistente obtuvo un promedio de **4.4 sobre 5** en las preguntas realizadas en la escala de Likert [66].
- Un **55 %** de los usuarios ingresaron desde dispositivos móviles, un **43 %** desde computadoras y un **1 %** desde tablets.

Otros datos obtenidos de las analíticas web muestran información como los navegadores usados, el sistema operativo y la ubicación geográfica de los usuarios (país), entre otros datos. En la **Figura 3.13** se muestran algunas de las analíticas obtenidas.

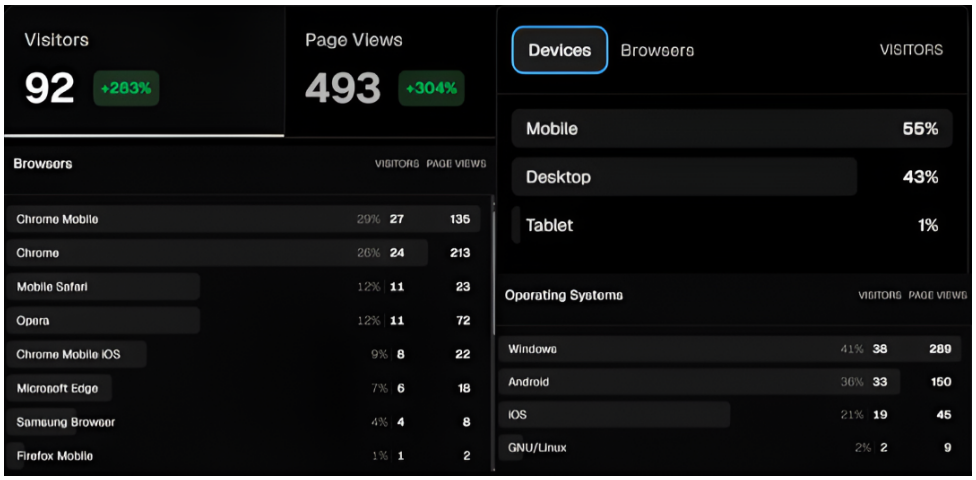

Figura 3.13. Analíticas de Vercel obtenidas en el test de aceptación.

Dado que en el Test de aceptación se tuvieron más de 50 usuarios concurrentes en el mismo momento, se puede considerar que también se realizó una prueba de estrés al sistema, evaluando su desempeño bajo

condiciones de alta carga. Un número estimado de alumnos en simultáneo en condiciones reales sería de aproximadamente 30, por lo que se probó con un 66.67 % más de carga. Durante estas pruebas, el sistema mantuvo un rendimiento estable, con tiempos de respuesta adecuados y sin caídas o errores, mostrando la capacidad para manejar múltiples usuarios simultáneamente.

En conclusión, el nivel de aceptación del sistema fue muy positivo, tanto por parte de los usuarios finales (estudiantes) como por docentes de la asignatura donde se llevó a cabo la prueba y del equipo de Psicología.

El test de aceptación tuvo como objetivo principal evaluar la calidad de las respuestas generadas por "NoVa+" en un entorno realista, así como la usabilidad y experiencia del usuario al interactuar con el asistente. Con el objetivo de analizar el desempeño del sistema se recopilaron métricas específicas en estas áreas que se explican en la siguiente subsección.

Métricas de usabilidad y experiencia de usuario

Para evaluar la usabilidad y experiencia del usuario al interactuar con "NoVa+", se utilizó el paquete de "Speed Insights" [67] (en español Información sobre velocidad) de Vercel. Esta herramienta nos proporciona facilidades para medir el rendimiento de la aplicación web desde la perspectiva del usuario final, analizando diversos aspectos que veremos a continuación. Como se mencionó anteriormente, la performance es uno de los puntos claves de nuestra aplicación, ya que un asistente que responda de manera correcta y rápida es fundamental para una buena experiencia de usuario, especialmente en jóvenes y los tiempos actuales donde el período de atención es más corto.

Métricas principales

- **Real Experience Score (RES - Puntuación de Experiencia Real):** Mide la experiencia general del usuario al interactuar con la aplicación. Es un promedio ponderado de todas las puntuaciones individuales de las métricas. El valor objetivo es de 90 a 100.

- **First Contentful Paint (FCP - Primer Renderizado de Contenido):** Mide el tiempo transcurrido desde el inicio de la página hasta la representación del primer elemento del contenido DOM. El valor objetivo es de 1.8 segundos o menos.

- **Largest Contentful Paint (LCP - Renderizado del Contenido más Grande):** Mide el tiempo transcurrido desde el inicio de la página hasta que el elemento de contenido más grande sea completamente visible. El valor objetivo es de 2.5 segundos o menos.

- **Interaction to Next Paint (INP - Interacción hasta Siguiente Renderizado):** Mide el tiempo transcurrido desde la interacción del usuario hasta que el navegador renderiza el siguiente fotograma. El valor objetivo es de 200 milisegundos o menos.

- **Cumulative Layout Shift (CLS - Cambio Acumulativo de Diseño):** Cuantifica la fracción de cambio de diseño que experimenta el usuario durante la vida útil de la página. El valor objetivo es de 0.1 o menos.

- **First Input Delay (FID - Retraso de Primera Interacción):** Mide el tiempo transcurrido desde la primera interacción del usuario hasta el momento en que el navegador puede responder. El valor objetivo es de 100 milisegundos o menos.

- **Time to First Byte (TTFB - Tiempo hasta el Primer Byte):** Mide el tiempo transcurrido desde la solicitud de un recurso hasta que comienza a llegar el primer byte de una respuesta. El valor objetivo es de menos de 800 milisegundos.

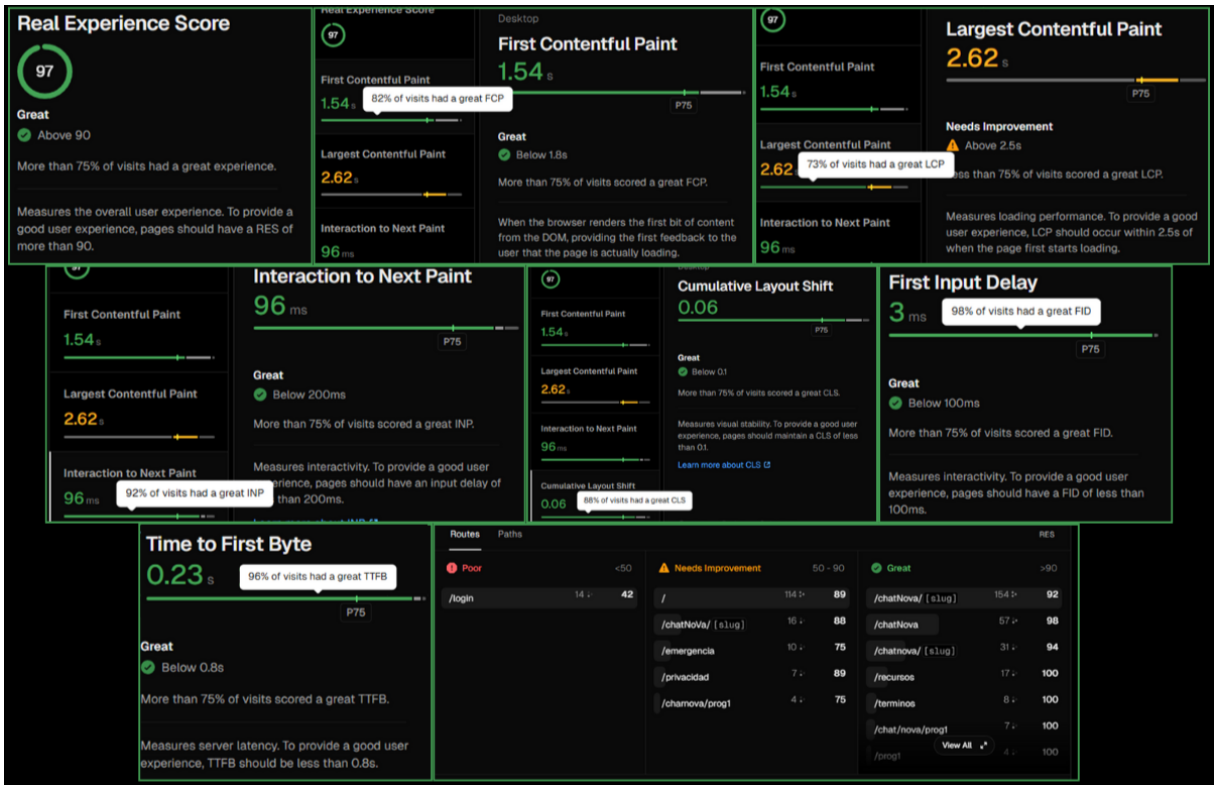

Figura 3.14. Información sobre velocidad y rendimiento obtenidas en el test de aceptación.

De las métricas mencionadas, se obtuvieron los siguientes resultados en las 2 pruebas del test de aceptación:

- **RES:** La métrica principal de experiencia de usuario dio 97 de 100 puntos.
- **FCP:** Promedio de 1.54 segundos, el 82 % tuvo excelente FCP.
- **LCP:** Promedio de 2.62 segundos, el 73 % tuvo excelente LCP.
- **INP:** Promedio de 96 milisegundos, el 92 % tuvo excelente INP.
- **CLS:** Promedio de 0.06, el 88 % tuvo excelente CLS.
- **FID:** Promedio de 3ms, el 98 % tuvo excelente FID.
- **TTFB:** Promedio de 0.23 segundos, el 96 % tuvo excelente TTFB.

Además de esto, se hizo un análisis de las rutas para identificar qué páginas eran visitadas y cuáles tenían mejor y peor rendimiento. En la **Figura 3.14** se muestran las métricas mencionadas anteriormente y el análisis de las rutas visitadas con su respectivo rendimiento.

Sobre las rutas, solo el "/login" tiene rendimiento "Pobre" donde se priorizo la seguridad sobre el rendimiento. Las páginas que "Necesitan mejorar" poseen un rendimiento bueno pero no excelente, son solamente páginas informativas. El apartado "Excelente" concentra la mayoría de las visitas, específicamente del chat con "NoVa+", que es el punto crítico de la aplicación y su rendimiento define gran parte de la experiencia final del usuario.

Se puede observar que todas las métricas obtenidas están dentro de los valores objetivos definidos por Vercel, lo que indica que la aplicación web "NoVa+" ofrece una experiencia de usuario rápida y fluida. Estos resultados reflejan un buen desempeño en términos de tiempos de carga, interactividad y estabilidad visual, contribuyendo a una experiencia positiva para los usuarios finales.

Validación del RAG

Uno de los pilares fundamentales para garantizar el correcto funcionamiento de "NoVa+" fue la integración del sistema RAG. Si bien el foco principal del proyecto no fue en el desarrollo de pruebas y mediciones avanzadas, resultó importante poder validar que el RAG recupera información relevante a la consulta y que el asistente efectivamente la utilizara para generar sus respuestas. Por esto se utilizaron mediciones que permiten de simple manera comprobar el correcto funcionamiento del sistema de recuperación [68].

Estas mediciones se realizaron para evaluar si las respuestas del RAG y el asistente funcionan de la manera esperada.

El sistema de RAG se analizó desde dos aspectos. Por un lado, se analiza si las búsquedas realizadas por el RAG devuelven información relevante a la consulta del usuario. Y, por otro lado, se analiza si el asistente efectivamente utilizaba la información obtenida del RAG para construir sus respuestas o si, por el contrario, ignoraba parcialmente esos resultados y respondía únicamente basándose en su conocimiento previo.

Relevancia de las respuestas del RAG

Para evaluar la relevancia de las búsquedas realizadas, se analizó si las respuestas recuperadas de la base de conocimiento respondían a la consulta original del usuario. Se buscaron y ajustaron métricas propias del área de la IA, tomando como referencia la métrica Exact Match [30], la cual analiza si el texto obtenido concuerda exactamente con el texto esperado. Esta métrica se ajustó a nuestro uso, posibilitando un mejor análisis. La medición utilizada consistió en identificar las palabras clave de la consulta y verificar si estas aparecían en la respuesta proporcionada por el RAG. De este modo, se buscó obtener una aproximación cuantitativa sobre el grado de alineación entre la consulta y el resultado recuperado.

Dado que el RAG realiza búsquedas conceptuales, no necesariamente devuelve coincidencias textuales exactas, sino información relacionada en términos de significado. Esto significa que esta medición basada en coincidencias de palabras exactas puede subestimar la verdadera relevancia de una respuesta.

Se consideró la implementación de una medición más adecuada al contexto, consistiendo en un modelo de IA adicional capaz de evaluar la relevancia desde un enfoque semántico, como realiza el RAG. Sin embargo, su integración hubiera aumentado significativamente la complejidad y los costos computacionales, excediendo el alcance definido para el proyecto. Por esa razón, se optó por mantener la medición inicial, que, aun siendo más simple, permitió obtener conclusiones sobre el funcionamiento del sistema.

Para calcular la relevancia de una respuesta, primero se identificaron las palabras clave presentes en la consulta del usuario y luego se verificó si dichas palabras aparecían en la respuesta obtenida a través del RAG. Sobre estas, se calculó un valor de relevancia con la división entre la cantidad de palabras clave presentes en la respuesta y el total de palabras clave detectadas en la consulta. Es decir, la medición devolverá valores entre 0 y 1 (es decir, 0 % a 100 %), si una consulta devuelve 50 %, significa que la mitad de las palabras presentes en la consulta original se encuentran mencionadas en la respuesta del RAG. Este cálculo nos da el valor de relevancia al que haremos referencia a continuación.

Para realizar este análisis de relevancia, se realizaron 22 consultas específicas sobre conocimiento de la base de conocimiento. De todas ellas, un 64 % obtuvo un valor de relevancia superior a 50 %, umbral que se definió como aceptable.

Para evaluar el funcionamiento del RAG, se llevó a cabo un análisis manual de los resultados. Esto permitió observar que, incluso en aquellos casos con valores más bajos, el RAG había recuperado información relevante a la consulta realizada. Por ejemplo, en una consulta sobre "estadísticas de juego", el sistema no devolvió literalmente las palabras "estadísticas" o "juego", pero sí devolvió estadísticas vinculadas a las apuestas, que respondían de forma coherente a la intención de la consulta. Lo mismo

ocurrió en el resto de las consultas realizadas que obtuvieron un valor menor a 50 %, lo que permitió inferir que el RAG identifica información relevante dentro de la base de conocimiento.

Uso de las respuestas del RAG

Además de analizar la relevancia de las respuestas recuperadas, se evaluó si "NoVa+" utilizaba efectivamente esta información en la construcción de sus respuestas finales. Para ello, se compararon los términos relevantes devueltos por el RAG con las respuestas generadas por el asistente.

Al igual que en la medición anterior, se realizaron 22 consultas relacionadas con contenidos almacenados en la base de conocimiento. Las pruebas dieron que el 95 % de las respuestas de "NoVa+" incorporaron de manera explícita la información recuperada. En el único caso en el que no se detectó el uso de la información provista, a partir de un análisis manual se pudo ver que el asistente sí utilizó la información. La medición no pudo reconocerlo, ya que el formato de respuesta de salida del RAG y del asistente eran diferentes, por lo que la búsqueda de palabras exactas no lo reconoció como la misma información.

Este análisis mostró que "NoVa+" accede a la base de conocimiento, encuentra información relevante y la utiliza adecuadamente en sus respuestas. De este modo, el asistente no solo recupera información relevante a las preguntas realizadas, sino que también las utiliza correctamente para dar respuestas acertadas.

Capítulo 4

CASO DE ESTUDIO

Este capítulo presenta un caso de estudio de ejemplo, que ilustra el funcionamiento integral del sistema “NoVa+” en un escenario similar al de uso real en un relevamiento a una escuela. El objetivo es mostrar, de manera clara y detallada, cómo se comporta la aplicación web cuando es utilizada tanto por el equipo de Psicología en un entorno controlado como por los propios estudiantes durante la prueba. De esta manera buscamos validar no solo el flujo técnico del sistema, sino también su potencial utilidad como herramienta preventiva para jóvenes en riesgo, tal como se plantea en los objetivos generales de este proyecto.

A través de capturas de pantalla y descripciones paso a paso, se muestra cómo opera el sistema desde la perspectiva de los diferentes actores. Se desarrolla en el contexto de las funcionalidades para gestión de sesiones grupales, Chatear con NoVa+, Visualizar estadísticas, entre otras. Podemos dividirlo en dos vistas:

1. Vista del actor Administrador o Investigador responsable de coordinar la prueba.
2. Vista del estudiante que interactúa con “NoVa+”.

El caso simula una prueba controlada en un entorno escolar, contexto para el cual fue diseñado principalmente el sistema, mostrando desde la creación de la sesión por parte del profesional de Psicología, pasando por la interacción del estudiante con el asistente conversacional, hasta la generación y visualización de estadísticas para análisis posterior. El objetivo es demostrar de manera concreta y visual cómo cada funcionalidad implementada se integra en una experiencia de usuario fluida, intuitiva y adecuada para el propósito preventivo del proyecto, permitiendo validar no solo los aspectos técnicos del sistema, sino también su aplicabilidad práctica en contextos reales de intervención.

4.1. Gestión de sesión grupal

Vista del Administrador o Investigador

En la **Figura 4.1** se muestra la pantalla del inicio del flujo de ejecución, con el listado de las sesiones grupales ya creadas. El Administrador o Investigador puede decidir crear una sesión grupal cuando deba realizar charlas, talleres o actividades en escuelas o instituciones relacionadas con la prevención de las apuestas en línea. Puede agendarla en cualquier momento y asignarle la fecha y el horario que necesite. La creación de una nueva sesión comienza al presionar el botón “Crear nuevo grupo”.

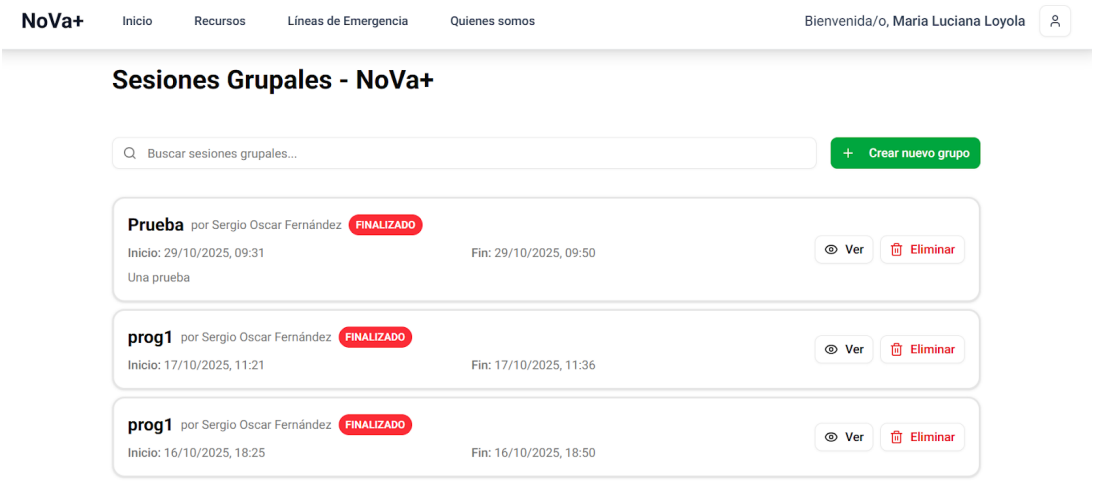

Figura 4.1. Listado de las sesiones grupales.

Luego se ingresan los datos de la sesión grupal como se muestra en la **Figura 4.2**, pantalla en la cual podrán ingresar todos los datos para crear una sesión grupal. Esta sesión se utiliza para organizar una prueba grupal específica, por ejemplo, una charla en una escuela o una actividad programada dentro de un estudio experimental. Durante esta creación, se debe completar una serie de datos:

- Un nombre representativo para identificar la prueba (por ejemplo, nombre de la escuela o del curso).
- Una descripción breve con información sobre la prueba, como el motivo de la misma.
- Una contraseña, necesaria para que los estudiantes ingresen.
- Una fecha de inicio y una duración total, que determinan el período en el cual los estudiantes podrán hablar con “NoVa+” en esta prueba.

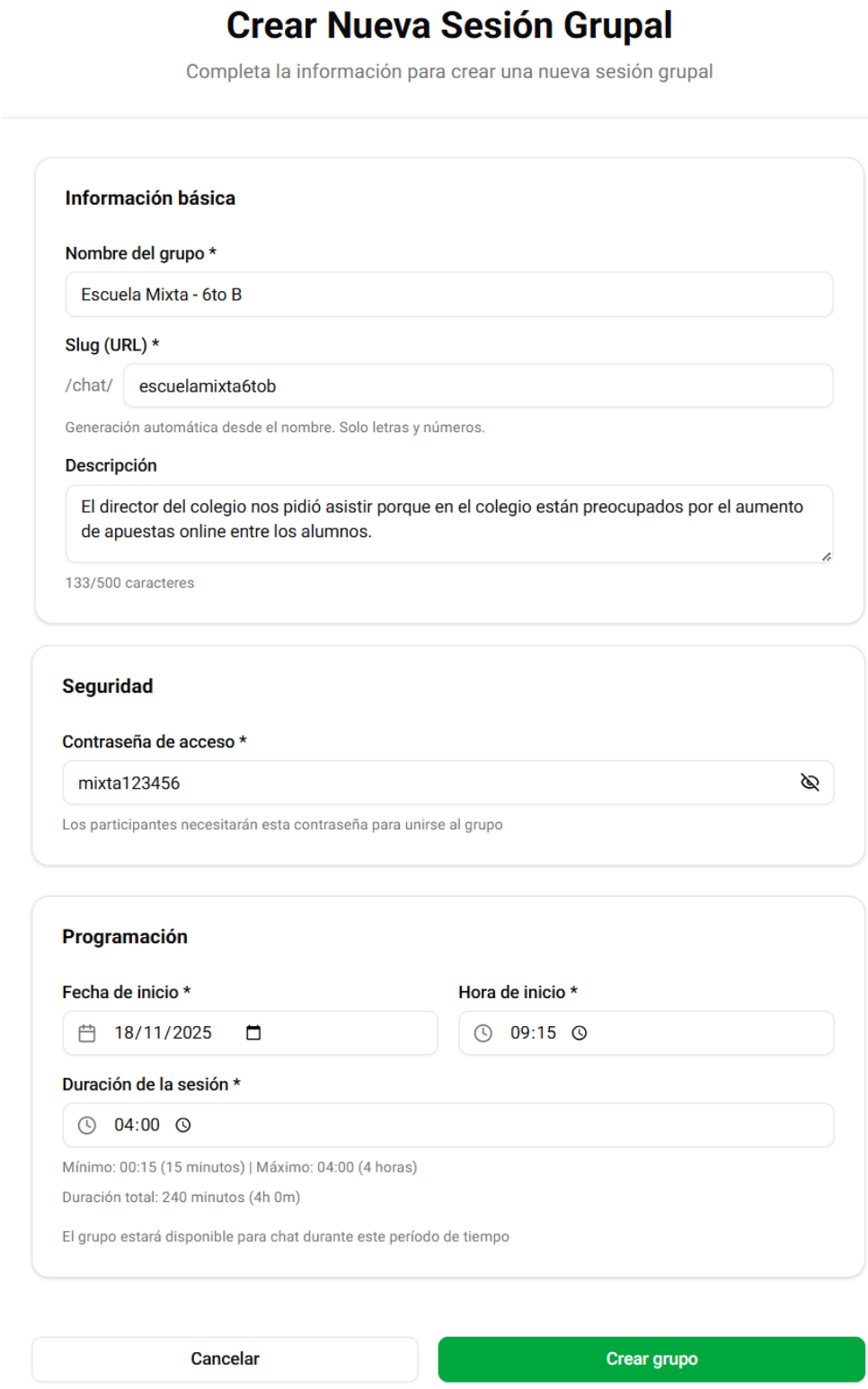

Figura 4.2. Creación de una sesión grupal.

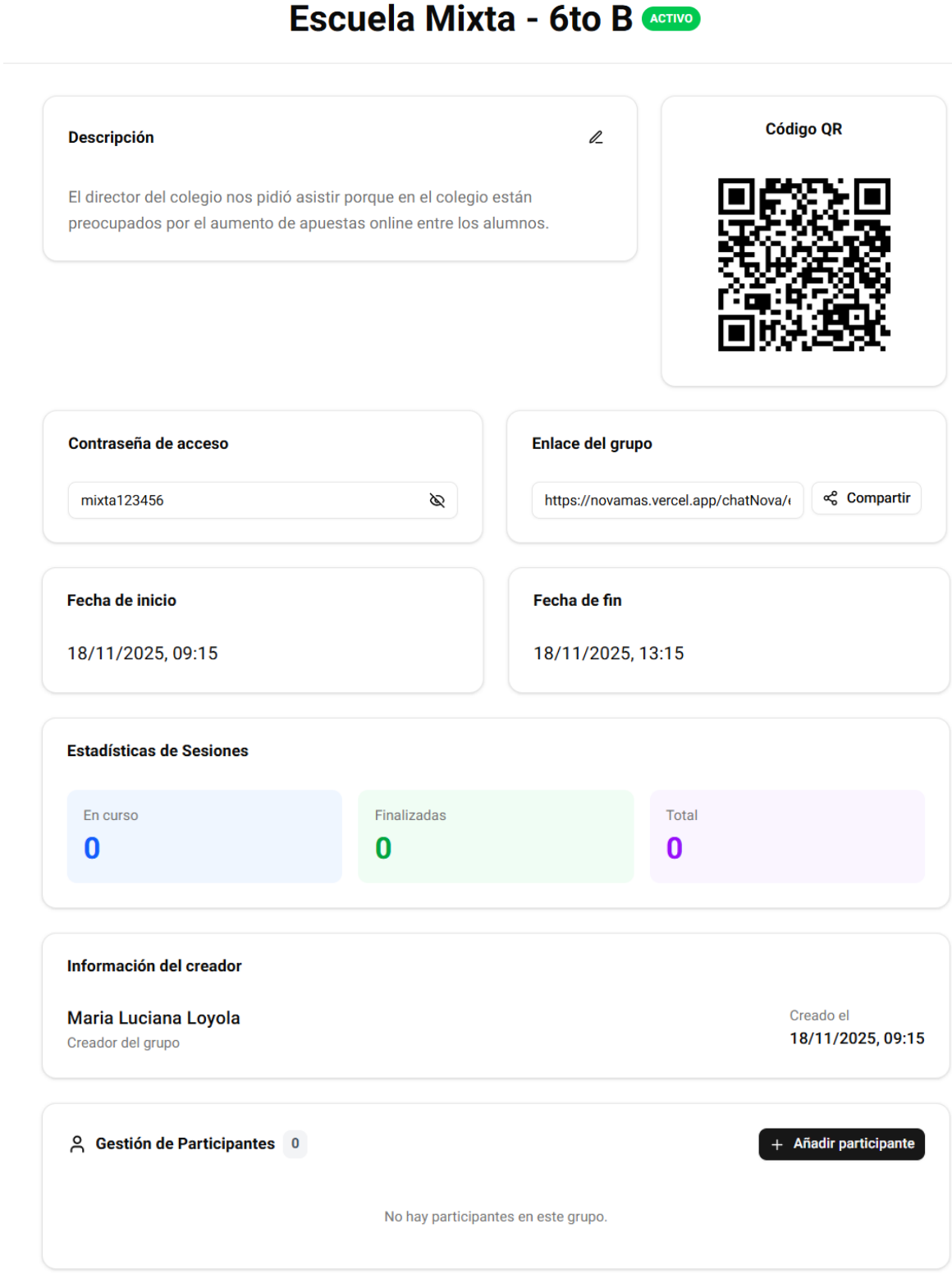

Figura 4.3. Pantalla de la información de la sesión grupal recién creada.

Una vez creada la sesión, el sistema muestra un resumen con todos los datos ingresados, representado en la **Figura 4.3**. Desde esta misma pantalla se puede compartir el enlace directo o un código QR para que los estudiantes accedan de forma rápida.

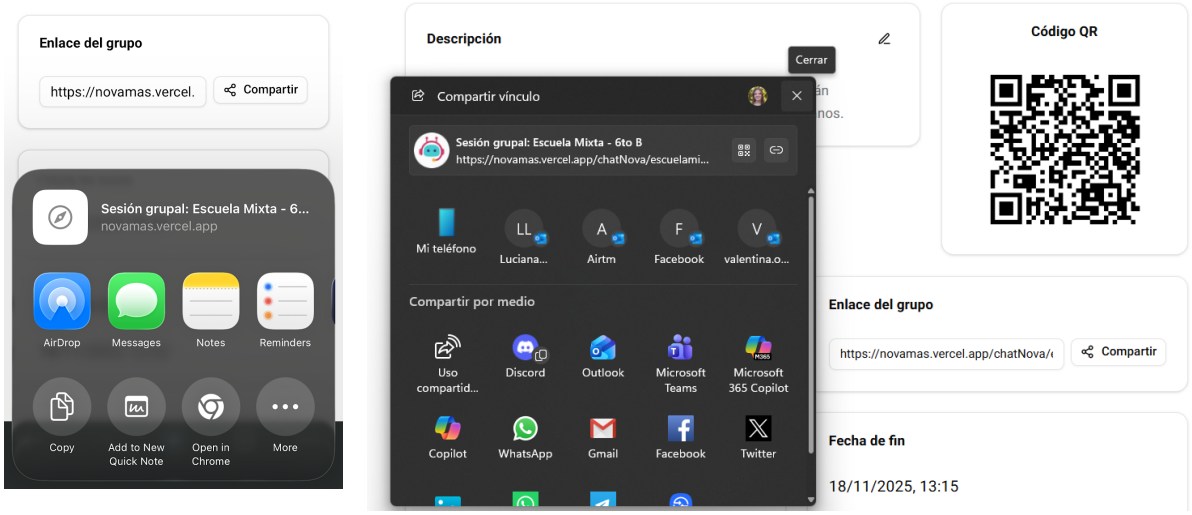

Figura 4.4. Pantallas de móvil y computadora para compartir sesión.

Es posible enviar el acceso de la sesion grupal, a traves de la funcionalidad compartir, con quienes se desee. La misma se muestra en la **Figura 4.4** con sus variantes en computadora y celular.

4.2. Chatear con “NoVa+”

Vista del Estudiante

La ejecución inicia cuando intenta acceder a la sesión grupal con la pantalla que se muestra en la **Figura 4.5**. Para ingresar, debe validar la contraseña definida previamente por el Administrador o Investigador, garantizando así que solo los participantes invitados puedan acceder a la prueba.

Figura 4.5. Validación de la sesión grupal.

Figura 4.6. Primer paso del fomulario inicial.

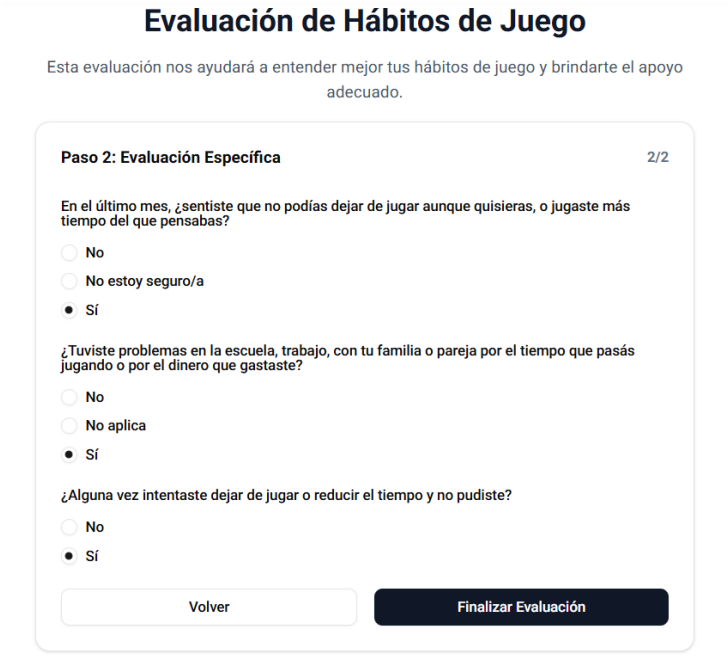

Figura 4.7. Segundo paso del formulario inicial.

Una vez validada la contraseña, el estudiante completa un formulario inicial. El formulario está dividido en dos pasos:

- En el primero se recopila información general como edad y género, junto con una pregunta sobre exposición previa al juego y su consentimiento sobre los Términos y Condiciones y las Políticas de Privacidad. La pantalla de este paso se ve reflejada en la **Figura 4.6**.
- Si el estudiante indica que tuvo exposición, pasa al segundo paso donde se profundiza en el grado de contacto con las apuestas. La pantalla de este paso se ve reflejada en la **Figura 4.7**.

Al finalizar la encuesta inicial, el estudiante puede comenzar la conversación con “NoVa+”. En la Figura **Figura 4.8** se expone un ejemplo de un chat con NoVa+. Cabe aclarar que el mismo es ilustrativo; es decir, no muestra la interacción con un usuario real.

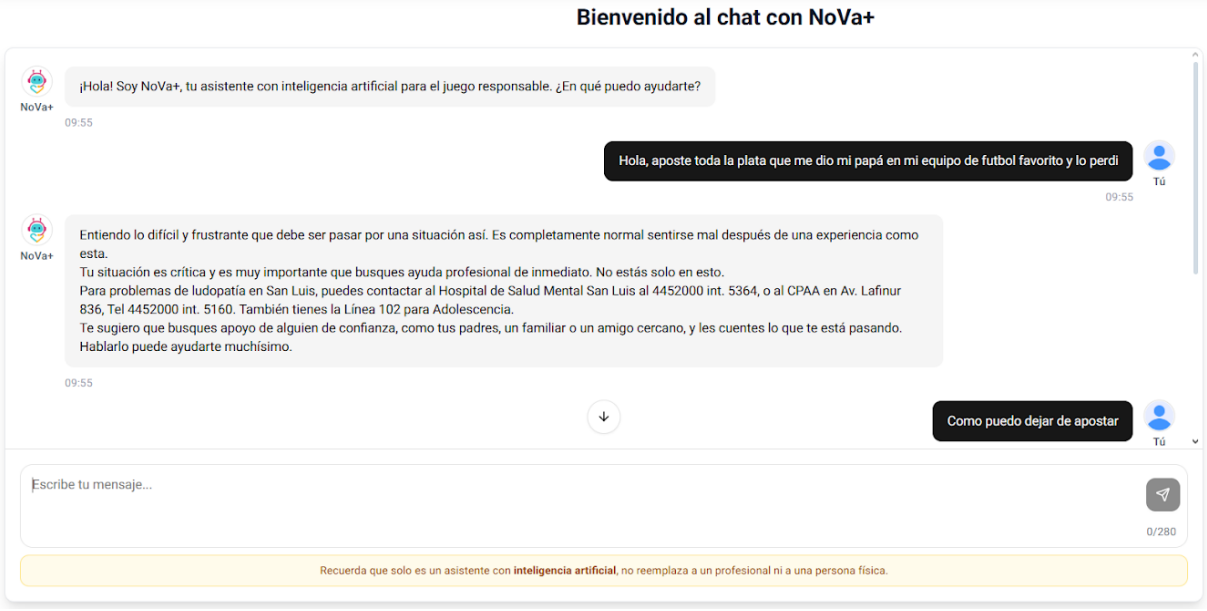

Figura 4.8. Pantalla del chat con “NoVa+”.

De manera opcional, el estudiante puede dejar una valoración del funcionamiento del chat en una escala del 1 al 5, tal como se observa en la **Figura 4.9**. Esta pregunta no busca evaluar aspectos personales del estudiante, sino la percepción general sobre el asistente.

Cuando el estudiante lo desee, o cuando el tiempo de la sesión haya terminado, puede finalizar la conversación presionando “Finalizar chat” en el recuadro posicionado a la derecha del chat. Este se ve reflejado en la **Figura 4.10**. Con el fin de evitar conversaciones vacías o sin contenido fundamental, el botón solo se habilita cuando el usuario ya envió al menos dos mensajes a “NoVa+”.

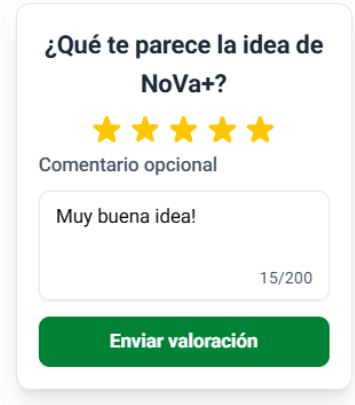

Figura 4.9. Comentario opcional.

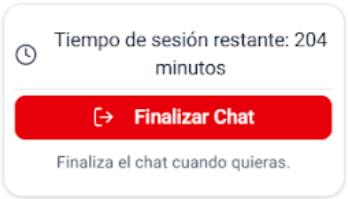

Figura 4.10. Botón para finalizar el chat.

Al finalizar el chat, se pasa al cuestionario final, el mismo se muestra en la **Figura 4.11**. Este formulario complementa la información inicial y permite evaluar cómo percibió el estudiante al asistente luego de interactuar con él, aportando información útil para el equipo de investigación de Psicología.

Figura 4.11. Formulario final.

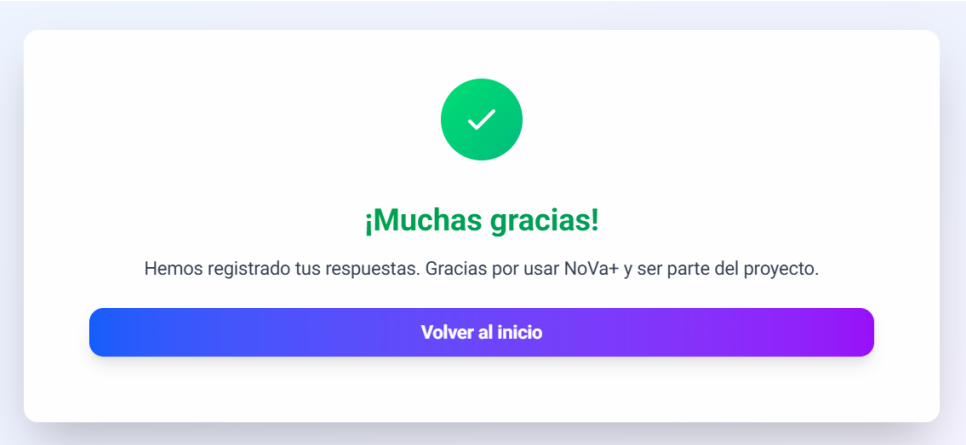

Figura 4.12. Pantalla del proceso del chat con "NoVa+" finalizado.

Se muestra en la **Figura 4.12** el mensaje de finalización del flujo de ejecución. El chatbot se presenta como un dispositivo de prevención accesible y confiable para los adolescentes, brindando información y orientación inmediata sobre los riesgos de las apuestas en línea.

4.3. Visualizar Estadísticas

Una vez que las sesiones han finalizado, el Administrador o Investigador puede acceder a la sección de estadísticas. Allí se muestran datos generales sobre el total de conversaciones, cuántas han sido analizadas y cuántas aún están pendientes de procesar. Si hay sesiones disponibles para analizar, el sistema habilita la generación de nuevas estadísticas, permitiendo actualizar la información con los resultados más recientes. Esta pantalla se ve reflejada en la **Figura 4.13**.

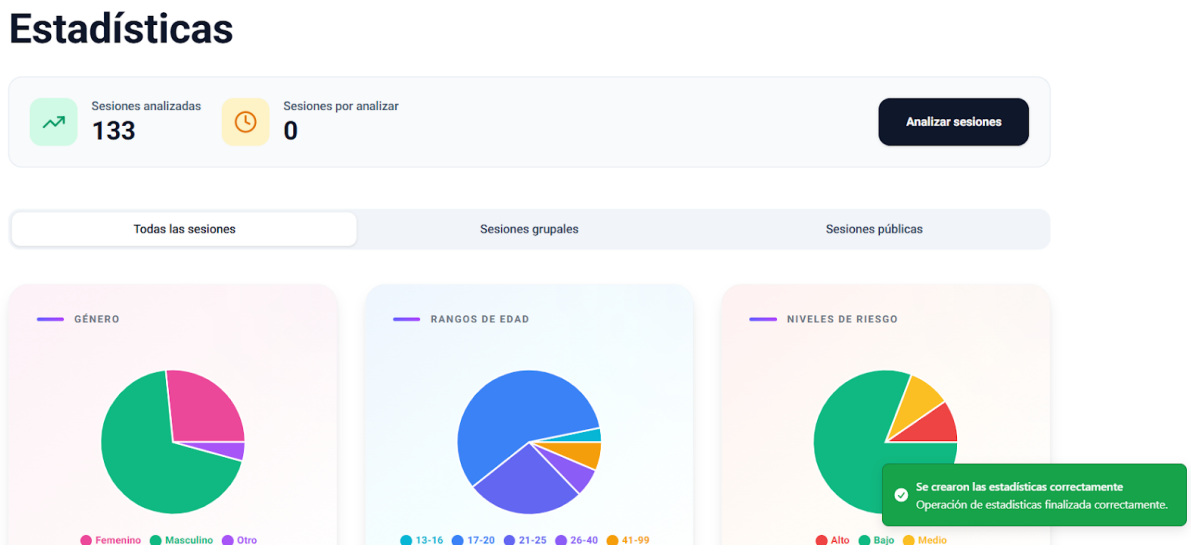

Figura 4.13. Pantalla de estadísticas.

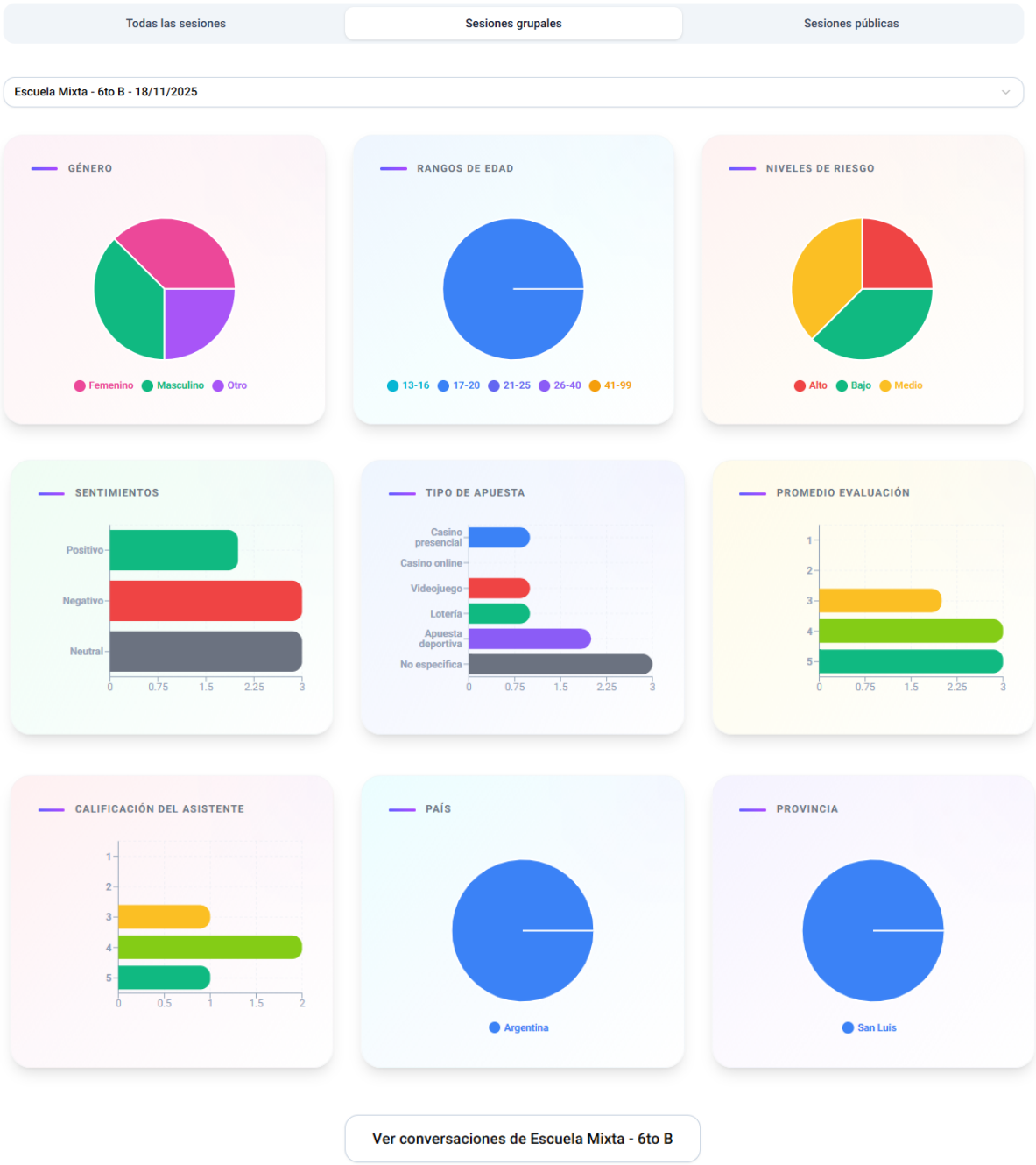

Figura 4.14. Estadísticas generales de la sesión grupal creada.

Después del análisis, el investigador puede buscar información de cualquier grupo que haya creado previamente. En la **Figura 4.14** se muestran las estadísticas de la prueba que se creó al inicio del caso de estudio.

Figura 4.15. Lista de conversaciones analizadas de la sesión grupal.

Es posible visualizar todas las conversaciones realizadas durante la prueba al presionar el botón “Ver conversaciones de ...”. El listado de conversaciones se muestra en la **Figura 4.15**. Éste permite filtrar por nivel de riesgo, género y ordenar por riesgo o por fecha, lo cual facilita la revisión puntual de casos de interés.

El sistema también posibilita exportar un archivo Excel con toda la información recopilada. Esta herramienta es especialmente útil para el equipo de investigación de Psicología, ya que les permite revisar los datos con mayor flexibilidad, aplicar filtros propios o generar informes adicionales. Los datos en el Excel generado se ven diferenciados en dos hojas de excel, una de ella es las estadísticas generales de la prueba, representada en la **Figura 4.16**.

Cantidad de sesiones	Sexo		Rango de edad		Nivel de riesgo		Sentimiento más frecuente		Tipo de apuesta		Cambió tema de conversación		Promedio evaluación del asistente		País	
8	Femenino	3	13-16	0	Alto	2	Positivo	2	Casino presencial	1	Si	1	1	0	Argentina	8
	Masculino	3	17-20	8	Medio	3	Negativo	3	Casino online	0	No	7	2	0		
	Otro	2	21-25	0	Bajo	3	Neutral	3	Videojuego	1			3	2		
			26-40	0					Lotería	1			4	3		
			41-99	0					Apuesta deportiva	2			5	3		
									No especifica	3						

Estadisticas generales | Estadisticas individuales

Figura 4.16. Excel generado de la sesión grupal, pantalla de estadísticas generales.

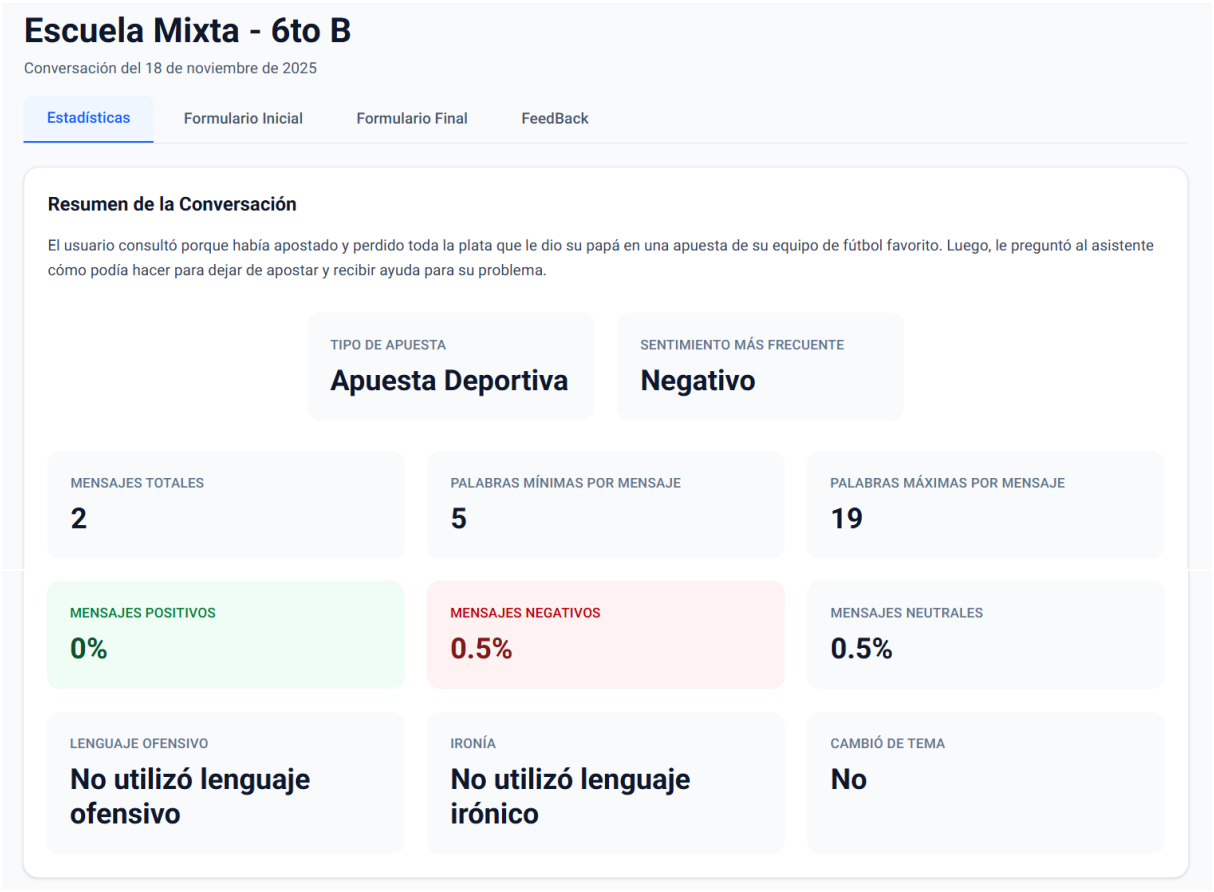

Figura 4.17. Pantalla de las estadísticas generales de una conversación en particular.

Finalmente, si el investigador desea indagar en profundidad una conversación en particular, puede entrar a sus estadísticas individuales en “Ver estadísticas” de la conversación de interés. Allí se muestran todas las estadísticas propias de esa conversación, como se ve en la **Figura 4.17**. También aparece el formulario inicial que completó el estudiante, mostrado en la **Figura 4.18**. Si el estudiante completó el formulario final, ese también se incluye, tal como se visualiza en la **Figura 4.19**, junto con la valoración opcional que aparece en la **Figura 4.20**.

En conclusión, la generación y visualización de estadísticas aporta un valor fundamental al sistema, ya que permite comprender mejor el uso del chatbot y las necesidades reales de los adolescentes que interactúan con él. Contar con estos datos facilita identificar patrones, ajustar estrategias de prevención y tomar decisiones basadas en evidencia, algo que antes era difícil de obtener de manera sistemática.

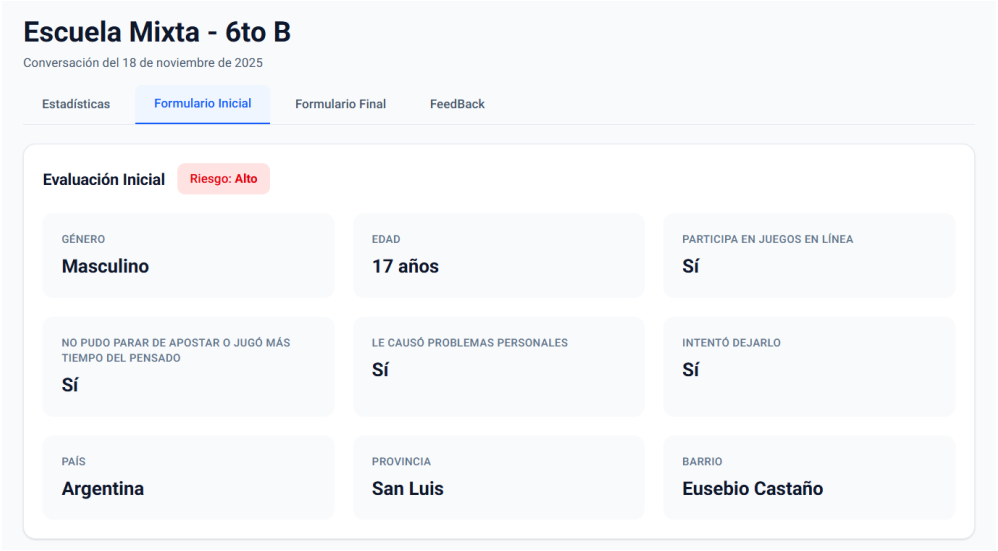

Figura 4.18. Pantalla del formulario inicial de una conversación en particular.

Figura 4.19. Pantalla del formulario final de una conversación en particular.

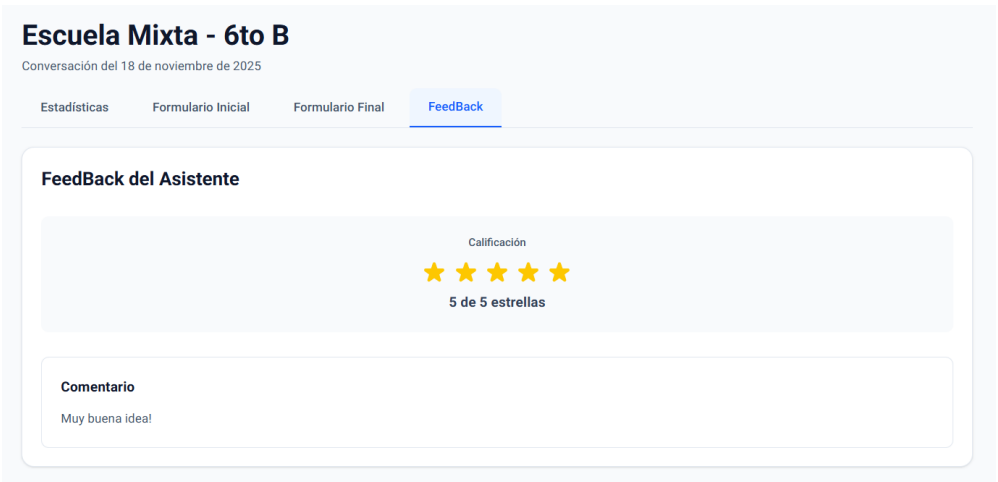

Figura 4.20. Pantalla del comentario de una conversación en particular.

Capítulo 5

CONCLUSIÓN Y TRABAJOS FUTUROS

En el presente informe de Proyecto Final Integrador de la Carrera se detalló el desarrollo de "NoVa+", un asistente conversacional web basado en LLM e integrado con técnicas de RAG, diseñado para el abordaje y la prevención del consumo problemático y/o posible adicción a las apuestas en línea en adolescentes y población general. El asistente fue desarrollado como parte de un proyecto interdisciplinario en colaboración con un grupo de investigadores de la Facultad de Psicología de la UNSL, quienes proveyeron el conocimiento especializado que sustenta las respuestas del sistema. La propuesta surge ante la necesidad de crear una herramienta preventiva innovadora que permita brindar información, orientación y acompañamiento a jóvenes en situación de riesgo, así como recopilar datos estadísticos anónimos para su posterior análisis científico sobre esta problemática emergente.

El sistema implementado no solo ha permitido cumplir con los objetivos técnicos y funcionales propuestos, sino que ha evolucionado para convertirse en una herramienta con potencial impacto social. Los usuarios que han interactuado con "NoVa+" durante las pruebas de aceptación experimentaron una interfaz intuitiva y accesible, demostrando la aceptación del asistente como recurso preventivo. La arquitectura basada en RAG posibilita que las respuestas proporcionadas se sustenten en conocimiento validado por profesionales de la Psicología, minimizando riesgos, asegurando información precisa y contextualmente relevante. Además, el módulo estadístico desarrollado permite al equipo de investigación acceder a métricas anónimas detalladas sobre las interacciones, facilitando el análisis científico de patrones de conducta y la efectividad de las intervenciones preventivas. La implementación de principios de IA responsable, siguiendo las recomendaciones de la AAIP, velando por la protección de datos personales, la transparencia y el uso ético de la tecnología.

5.1. Conclusiones

Aportes del sistema

El desarrollo del presente proyecto ha permitido alcanzar satisfactoriamente el objetivo general de desarrollar un asistente conversacional web basado en LLM, integrado con técnicas de RAG, orientado a abordar la problemática de las apuestas online en jóvenes, adolescentes y población en general, que contribuya a brindar contención y acompañamiento, promoviendo el juego responsable y ofreciendo recursos para comunicarse con centros de atención en caso de detectarse un comportamiento de juego problemático y/o patológico en el usuario.

Este objetivo general está materializado en el asistente "NoVa+". Para lograr esto, a lo largo del ciclo de desarrollo, se cumplieron exitosamente los objetivos particulares planteados:

- Se realizó un análisis de sistemas existentes, identificando que ninguno aborda específicamente la ludopatía ni incorpora capacidades de recopilación estadística, validando la importancia de la propuesta.

- Se incorporaron transversalmente las recomendaciones de la Agencia de Acceso a la Información Pública para IA responsable, integrando principios de transparencia, privacidad y seguridad en cada etapa, con políticas claras.

- La arquitectura implementada incluye una base de datos que almacena conocimiento especializado validado por expertos en Psicología, permitiendo respuestas contextualizadas sobre prevención, juego responsable y recursos profesionales.

- El sistema incorpora capacidades para detectar patrones de conducta problemática, facilitando el acceso a información preventiva y contacto con especialistas, respetando principios de no diagnóstico.

- Se diseñaron dos formas de uso diferenciados: acceso público anónimo sin registro y acceso controlado para entornos escolares mediante credenciales temporales.

- Se implementó un proceso de validación, incluyendo pruebas con expertos del área de Psicología, test de aceptación con estudiantes universitarios, pruebas en escuelas secundarias, documentando métricas de usabilidad, rendimiento y evaluación del sistema RAG.

- La protección de datos se garantizó mediante almacenamiento anónimo con identificadores de sesión, cifrado de transmisión, políticas de retención y encriptado de datos.

- Se desarrolló un módulo estadístico completo que registra estadísticas anónimas (duración, temas, patrones, niveles de riesgo), disponible para el equipo de Psicología mediante un panel de visualización con exportación a Excel.

- El sistema fue desplegado en producción con integración continua.

El trabajo ha sido expuesto en múltiples eventos, tal como se mencionará en la siguiente sección, mostrando su relevancia no solo a nivel académico sino también como potencial herramienta social para la prevención de la ludopatía. En síntesis, el proyecto generó un producto funcional utilizado activamente, contribuyendo al desarrollo tecnológico y al abordaje de una problemática social en auge.

Impacto

El asistente conversacional “NoVa+”, no solo constituye el “Proyecto Final Integrador de Ingeniería en Informática” presentado en este informe, sino que, además forma parte del proyecto de investigación “Evaluación de la Efectividad y Aceptación de un Asistente Virtual Basado en Inteligencia Artificial para la Prevención de la Adicción al Juego Online en Adolescentes” de la Facultad de Psicología.

Esta idea y desarrollo del proyecto “NoVa+” ha demostrado su impacto en la sociedad formando parte de charlas y publicaciones como:

- Charla en la Tercera edición de la Jornada Salud y Consumos en la Vida Universitaria - UNSL 2025.

- Divulgación en el Honorable Concejo Deliberante de la municipalidad de San Luis para el Día Internacional del Programador Informático.

- Publicación de paper y poster en el Congreso de Salud Mental - Mendoza 2025.

- Publicación de paper y poster en el Congreso Nacional de Ingeniería Informática / Sistemas de Información - Córdoba 2025.

Además de estas presentaciones académicas, el sistema ha sido demostrado en funcionamiento real ante diferentes públicos, permitiendo validar sus capacidades y recopilar retroalimentación directa de usuarios potenciales en diversos contextos:

- Muestra y uso en vivo del asistente en la Tercera edición de la Jornada Salud y Consumos en la Vida Universitaria.
- Charla, muestra y uso en vivo del asistente en la Promoción de Carreras UNSL 2025.
- Pruebas de uso por estudiantes en dos diferentes comisiones de primer año de la UNSL.
- Validación con el usuario final en una escuela secundaria, en la jornada de puertas abiertas de la Escuela Normal Juan Pascual Pringles.

5.2. Trabajos futuros

A lo largo de este proyecto se logró implementar un asistente virtual funcional con una base sólida para la prevención de apuestas online en adolescentes y jóvenes adultos. Durante el proceso fueron surgiendo muchas ideas, mejoras y posibles extensiones que, por cuestiones de tiempo y alcance, no pudieron incorporarse en esta etapa. Sin embargo, todas ellas abren un camino interesante para seguir ampliando y fortaleciendo el proyecto en el futuro.

A continuación, se presentan las principales ideas de trabajo que podrían desarrollarse:

- **Integrar una IA especializada para la detección de ludopatía.** Para trabajos futuros, sería interesante incorporar un modelo más específico, que se destaque por detectar patrones de riesgo en mensajes de redes sociales. Integrar este tipo de modelo en la sección de estadísticas permitiría analizar y evaluar las conversaciones con métricas mucho más precisas.
- **Implementar un mapa interactivo de zonas de riesgo por barrio.** Se propone desarrollar un mapa de Argentina que muestre regiones con mayor nivel de vulnerabilidad, basándose en los niveles de riesgo detectados y el uso del asistente.
- **Utilizar una IA de forma local.** Inicialmente se evaluó la posibilidad de ejecutar el modelo de manera local para reforzar la privacidad y el control sobre los datos. Si bien no fue posible por limitaciones técnicas, continúa siendo una línea de trabajo a futuro.
- **Automatizar el cálculo de estadísticas al finalizar cada sesión.** Se plantea la generación de las estadísticas automáticamente al cierre de cada conversación, como por ejemplo con el uso de colas que permitan procesar la información sin intervención manual.
- **Desarrollar una versión mobile o una PWA.** Crear una aplicación móvil o una Progressive Web App ampliaría notablemente el acceso al asistente.
- **Incorporar monitoreo en tiempo real de emociones o señales de alarma.** Se propone agregar análisis emocional durante la conversación para detectar cambios bruscos en el estado del usuario y activar alertas o protocolos de contención cuando sea necesario.
- **Integración con WhatsApp para facilitar el acceso.** Esta fue una de las ideas propuestas por el equipo de Psicología, en el cual se planteó integrar el asistente a un chat especializado en WhatsApp.
- **Integración con entidades del sector de juegos de azar.** Explorar colaboraciones con agencias de lotería, casinos o empresas del rubro, a fin de implementar políticas preventivas y generar sistemas de alerta temprana.

REFERENCIAS

[1] Josefa Rodríguez. «Ludopatía en la infancia. ¿Una nueva pandemia?» En: *Medicina Infantil* (2024). [En línea]. URL: https://pesquisa.bvsalud.org/portal/resource/pt/biblio-1578950.

[2] Isabel Sales Triguero y Alexis Cloquell Lozano. «La adicción al juego online entre los adolescentes españoles: propuestas de prevención en el marco educativo». En: (2021). [En línea]. URL: https://revistas.ucv.es/edetania/index.php/Edetania/article/view/810.

[3] Pablo García Ruiz, Pilar Buil y María José Solé Moratilla. «Consumos de riesgo: menores y juego de azar online. El problema del "juego responsable"». En: *Política y Sociedad* (2016). [En línea]. URL: https://dialnet.unirioja.es/servlet/articulo?codigo=6009037.

[4] Marc N. Potenza et al. «Correlates of at-risk/problem Internet gambling in adolescents acknowledging or denying gambling on the Internet». En: *Journal of Adolescent Health* (). [En línea]. URL: https://pubmed.ncbi.nlm.nih.gov/21241952/.

[5] Walter Martello. *Presentación del Observatorio de Adicciones y Consumos Problemáticos*. [En línea]. 2024. URL: https://drive.google.com/file/d/165JgLjvY-Ag6vvhRfcvkDowEpIIzVvD9/view.

[6] UNICEF Argentina. *Apuestas online: un llamado de atención*. Artículo. [En línea]. Nov. de 2024. URL: https://www.unicef.org/argentina/historias/apuestas-online.

[7] Andrés Rubino. *Casi 4 de cada 10 estudiantes que apuestan en línea tienen conductas de riesgo y problemas de juego*. Artículo. [En línea]. Nov. de 2024. URL: https://unciencia.unc.edu.ar/psicologia/casi-4-de-cada-10-estudiantes-que-apuestan-en-linea-tienen-conductas-de-riesgo-y-problemas-de-juego.

[8] Veton Këpuska y Gamal Bohouta. *Next-Generation of Virtual Personal Assistants (Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home)*. [En línea]. 2018. URL: https://ieeexplore.ieee.org/document/8301638.

[9] Sebastian Raschka. *Build a Large Language Model (From Scratch)*. Shelter Island, NY: Manning Publications, 2024.

[10] Patrick Lewis, Ethan Perez y et al. Aleksandra Piktus. «Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks». En: *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020)*. [En línea]. 2020. URL: https://arxiv.org/abs/2005.11401.

[11] Kent Beck y Cynthia Andres. *Extreme Programming Explained: Embrace Change*. 2.ª ed. Addison-Wesley, 2004.

[12] Agencia de Acceso a la Información Pública. *Guía para entidades públicas y privadas en materia de Transparencia y Protección de Datos Personales para una Inteligencia Artificial responsable*. [En línea]. 2024. URL: https://www.argentina.gob.ar/sites/default/files/aaip-argentina-guia_para_usar_la_ia_de_manera_responsable.pdf.

[13] UNICEF Argentina. *Más de la mitad de las chicas y los chicos usa Inteligencia Artificial: dos de cada tres, con fines escolares*. Comunicado de prensa. [En línea]. Mayo de 2025. URL: https://www.unicef.org/argentina/comunicados-prensa/mas-de-la-mitad-de-las-chicas-y-los-chicos-usa-inteligencia-artificial.

[14] UNICEF Innocenti. *Guía de IA para adolescentes*. Informe. [En línea]. 2021. URL: https://www.unicef.org/innocenti/media/1391/file/UNICEF-Global-Insight-AI-guide-for-teens-2021_ES.pdf.

[15] Yana. *Yana: Tu acompañante emocional*. [En línea]. 2025. URL: https://www.yana.ai/es/home.

[16] Wysa. *Wysa: Everyday Mental Health*. [En línea]. 2025. URL: https://www.wysa.com.

[17] Woebot Health. *Woebot Health: AI-powered mental health support*. [En línea]. 2025. URL: https://woebothealth.com.

[18] Ian Sommerville. *Software Engineering*. 10.ª ed. Pearson, 2016.

[19] Ian Sommerville. *Engineering Software Products: An Introduction to Modern Software Engineering*. Pearson, 2019.

[20] Ashish Vaswani et al. «Attention Is All You Need». En: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 30. 2017. URL: https://arxiv.org/abs/1706.03762.

[21] Daniel Jurafsky y James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. 3.ª ed. Draft edition, disponible en línea. Draft, 2023. URL: https://web.stanford.edu/~jurafsky/slp3/.

[22] Alec Radford et al. *Improving Language Understanding by Generative Pre-Training*. Technical Report, OpenAI. [En línea]. 2018. URL: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

[23] Google Cloud. *What is a Vector Database?* Artículo. [En línea]. 2025. URL: https://cloud.google.com/discover/what-is-a-vector-database?hl=en.

[24] Tom B. Brown et al. «Language Models are Few-Shot Learners». En: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 33. 2020, págs. 1877-1901.

[25] Laura Weidinger et al. «Ethical and social risks of harm from Language Models». En: *Proc. ACM Conf. Fairness, Accountability, and Transparency (FAccT)*. [En línea]. 2022. URL: https://dl.acm.org/doi/10.1145/3531146.3533088.

[26] European Commission. *Ethics Guidelines for Trustworthy AI*. [En línea]. 2019. URL: https://ec.europa.eu/digital-strategy/sites/default/files/ethics-guidelines-trustworthy-ai.pdf.

[27] European Union. «Artificial Intelligence Act». En: *Official Journal of the European Union* L 1689 (2024).

[28] Yunfan Gao et al. *Retrieval-Augmented Generation for Large Language Models: A Survey*. [En línea]. 2023. URL: https://arxiv.org/abs/2312.10997.

[29] Cameron R. Wolfe. *A Practitioner's Guide to Retrieval*. Artículo. [En línea]. Sep. de 2023. URL: https://cameronrwolfe.substack.com/p/a-practitioners-guide-to-retrieval.

[30] IBM. *Exact Match Metrics*. [En línea]. 2025. URL: https://www.ibm.com/docs/en/waasfgm?topic=metrics-exact-match.

[31] Lingrui Mei, Jiayu Yao et al. «A Survey of Context Engineering for Large Language Models». En: *arXiv preprint* arXiv:2507.13334 (2025). [En línea]. URL: https://arxiv.org/pdf/2507.13334.

[32] DataCamp. *Context Engineering: A Guide With Examples*. [En línea]. Jul. de 2025. URL: https://www.datacamp.com/es/blog/context-engineering.

[33] Google Cloud. *Google Gen AI SDK: Interfaz unificada para modelos Gemini 2.5 Pro y 2.0*. Documentación oficial. [En línea]. 2025. URL: https://cloud.google.com/vertex-ai/generative-ai/docs/sdks/overview?hl=es-419.

[34] Google Cloud. *Introducción a Vertex AI*. Documentación oficial. [En línea]. 2025. URL: https://docs.cloud.google.com/vertex-ai/docs/start/introduction-unified-platform?hl=es.

[35] Artificial Analysis. *Artificial Analysis*. [En línea]. 2025. URL: https://artificialanalysis.ai/.

[36] Google AI. *Gemini API Documentation: Gemini Models*. [En línea]. 2025. URL: https://ai.google.dev/gemini-api/docs/models?hl=es-419#gemini-2.5-flash.

[37] W3Schools. *Introduction to React*. Tutorial web. [En línea]. 2025. URL: https://www.w3schools.com/react/react_intro.asp.

[38] Vercel. *Vercel: Build and deploy the best web experiences with the Frontend Cloud*. [En línea]. 2025. URL: https://vercel.com.

[39] Vercel. *Next.js: The React Framework for the Web*. Documentación oficial. [En línea]. 2025. URL: https://nextjs.org/.

[40] shadcn/ui. *shadcn/ui Documentation: Beautiful, Accessible Components*. Documentación web. [En línea]. 2025. URL: https://ui.shadcn.com/docs.

[41] Auth.js. *Auth.js: Authentication for JavaScript/TypeScript*. Documentación oficial. [En línea]. 2025. URL: https://authjs.dev/.

[42] JWT.io. *Introduction to JSON Web Tokens*. [En línea]. 2025. URL: https://jwt.io/introduction#what-is-json-web-token.

[43] W3Schools. *Introducción a Python*. Tutorial web. [En línea]. 2025. URL: https://www.w3schools.com/python/python_intro.asp.

[44] Juan Manuel Pérez et al. *pysentimiento: A Python Toolkit for Opinion Mining and Social NLP tasks*. 2023. URL: https://arxiv.org/abs/2106.09462.

[45] PostgreSQL Global Development Group. *¿Qué es PostgreSQL?* Documentación oficial. [En línea]. 2025. URL: https://www.postgresql.org/docs/current/intro-whatis.html.

[46] pgvector. *pgvector: Open-source vector similarity search for Postgres*. Repositorio en GitHub. [En línea]. 2025. URL: https://github.com/pgvector/pgvector.

[47] Neon. *Neon: Serverless Postgres*. Plataforma web. [En línea]. 2025. URL: https://neon.com/.

[48] Drizzle Team. *Drizzle ORM: ¿Por qué Drizzle?* Documentación oficial. [En línea]. 2025. URL: https://orm.drizzle.team/docs/overview.

[49] Google Cloud. *¿Qué es la gestión de APIs?* Artículo. [En línea]. 2025. URL: https://cloud.google.com/learn/what-is-api-management.

[50] Vercel. *AI SDK Documentation*. Documentación oficial. [En línea]. 2025. URL: https://ai-sdk.dev/docs/introduction.

[51] Docker, Inc. *What is Docker? – Docker Overview*. [En línea]. 2025. URL: https://docs.docker.com/get-started/docker-overview/.

[52] Ollama, Inc. *Ollama: Chat and build with open models*. [En línea]. 2025. URL: https://ollama.com.

[53] Hugging Face, Inc. *Hugging Face: The AI community building the future*. [En línea]. 2025. URL: https://huggingface.co.

[54] Hugging Face. *Inference Endpoints: Despliegue de modelos de IA en producción*. Documentación oficial. [En línea]. 2025. URL: https://huggingface.co/docs/inference-endpoints/index.

[55] Hugging Face, Inc. *Inference Endpoints Pricing*. [En línea]. 2025. URL: https://huggingface.co/docs/inference-endpoints/pricing.

[56] Google AI Developers. *Run Gemma with the Gemini API*. Documentación técnica. [En línea]. 2025. URL: https://ai.google.dev/gemma/docs/core/gemma_on_gemini_api.

[57] Google DeepMind. *Tarjeta de Modelo Gemma 3*. Documentación técnica. [En línea]. 2025. URL: https://ai.google.dev/gemma/docs/core/model_card_3?hl=es-419.

[58] Google AI Developers. *Gemini API Additional Terms of Service*. Términos y Condiciones. [En línea]. 2025. URL: https://ai.google.dev/gemini-api/terms.

[59] Hostinger. *VPS Argentina | Servidor Virtual en Argentina*. [En línea]. 2025. URL: https://www.hostinger.com/ar/vps-argentina.

[60] Amazon Web Services. *AWS Free Tier | Explora servicios sin costo*. [En línea]. 2025. URL: https://aws.amazon.com/es/free.

[61] NIC Argentina. *Dominios y Aranceles*. [En línea]. 2025. URL: https://nic.ar/es/dominios/dominios_y_aranceles.

[62] Consejo Profesional de Ciencias Informáticas de la Provincia de Córdoba (CPCIPC). *Honorarios recomendados*. [En línea]. 2025. URL: https://cpcipc.org.ar/honorarios-recomendados/.

[63] Agencia de Acceso a la Información Pública (AAIP). *Evaluación de riesgo: Módulo Protección de Datos Personales del RITE*. [En línea]. 2024. URL: https://www.argentina.gob.ar/sites/default/files/aaip_rite_modulo_proteccion_de_datos_evaluacion.pdf.

[64] Ángel Mena Rodríguez. *El peligro de la dependencia emocional con la IA*. Artículo. [En línea]. Nov. de 2024. URL: https://psicologiaymente.com/psicologia/peligro-dependencia-emocional-ia.

[65] Morris Dworkin. *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC*. NIST Special Publication 800-38D. [En línea]. National Institute of Standards y Technology (NIST), 2007. URL: https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-38d.pdf.

[66] Antonio Matas. «Diseño del formato de escalas tipo Likert». En: *Revista Electrónica de Investigación Educativa (REDIE)* (2018). [En línea]. URL: https://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1607-40412018000100038.

[67] Vercel. *Speed Insights: Metrics*. Documentación oficial. [En línea]. 2025. URL: https://vercel.com/docs/speed-insights/metrics.

[68] Confident AI. *RAG Evaluation Metrics*. [En línea]. 2025. URL: https://www.confident-ai.com/blog/rag-evaluation-metrics-answer-relevancy-faithfulness-and-more.

ANEXO I

Políticas de Privacidad

1. Información General

NoVa+ es un asistente virtual conversacional orientado a la investigación científica, desarrollado en la Universidad Nacional de San Luis. Participan en su diseño e implementación investigadores del área de Psicología e Informática.

2. Compromiso con la Privacidad

En NoVa+ adoptamos los principios de Privacidad por Defecto, lo que significa que:

- Solo recopilamos y procesamos datos personales estrictamente necesarios para los fines de investigación científica.
- Implementamos medidas técnicas y organizativas para proteger tu información desde el momento de su recolección.
- Minimizamos la cantidad de datos procesados.

3. Consentimiento Informado

El uso de NoVa+ requiere tu consentimiento explícito, el cual será solicitado antes de acceder al asistente conversacional.

4. Finalidad del Tratamiento de Datos

Los datos personales que recopilamos se utilizan exclusivamente para:

- Investigar y analizar de manera científica y psicológica los hábitos de juego online y los factores de riesgo asociados al consumo abusivo o la ludopatía, para llevar a cabo acciones de prevención en el campo de las adicciones al juego online.
- Evaluar la efectividad, aceptación y utilidad del asistente virtual en las intervenciones conversacionales proporcionadas.
- Publicar resultados académicos en formato completamente anónimo.

En ningún caso los datos personales serán utilizados con fines comerciales, publicitarios ni compartidos con terceros ajenos al proyecto de investigación.

5. Datos que Recopilamos

Conforme al principio de minimización de datos, recolectamos únicamente la información necesaria para cumplir con los fines de investigación:

5.1 Datos Proporcionados por el Usuario

- Género
- Edad
- Ubicación geográfica
- Respuestas sobre hábitos de juego
- Respuestas al formulario de evaluación del asistente
- Experiencia del usuario

5.2 Datos Generados Automáticamente

- Contenido de los mensajes enviados al asistente
- Estadísticas de uso: número de mensajes, cantidad de palabras, duración de las conversaciones
- Análisis de sentimiento: valoraciones sobre emociones expresadas durante las conversaciones (tratado de forma anónima)

5.3 Datos Técnicos

- Dirección IP
- Sistema operativo
- Navegador
- Tipo de dispositivo (móvil o escritorio)
- Datos analíticos (proveedor de hosting y métricas técnicas anónimas sobre el uso de la web)

6. Cómo Almacenamos y Protegemos tus Datos

6.1 Almacenamiento

Los datos se almacenan en entornos seguros bajo estándares internacionales de protección:

- Aplicación web: Servidores en la nube de Vercel
- Base de datos: Neon (PostgreSQL en la nube)
- Estadísticas: Servidor local de la Universidad Nacional de San Luis

6.2 Medidas de Seguridad

Implementamos robustas medidas de seguridad técnicas y organizativas para proteger tus datos:

- Cifrado en tránsito: Utilizamos protocolos HTTPS/TLS en todas las comunicaciones para proteger los datos durante su transmisión.
- Cifrado en reposo: Los mensajes se almacenan cifrados mediante el algoritmo AES-256-GCM.
- Control de acceso: Solo el administrador del sistema y los investigadores autorizados con credenciales específicas tienen acceso a los datos.

6.3 Servicios de Terceros

Para el funcionamiento del asistente conversacional, utilizamos los siguientes servicios externos:

- Google Generative AI (API de Gemini): Modelo de lenguaje que procesa las conversaciones de manera automática. Los datos enviados a Gemini se procesan de acuerdo con sus estándares de seguridad y privacidad, sin vincularse con información personal identificable.
- Vercel Analytics: Métricas técnicas anónimas sobre el uso del sistema.

El uso de estos servicios está sujeto a las políticas de privacidad de sus respectivos proveedores. El proyecto no comparte datos personales identificables con terceros ajenos a la investigación.

7. Tiempo de Conservación de los Datos

7.1 Mensajes del Chat

Los mensajes completos enviados al asistente conversacional se conservan durante un período de 30 días desde su envío. Transcurrido este plazo, se eliminan automáticamente de nuestros sistemas.

7.2 Estadísticas Anónimas

Las estadísticas agregadas y completamente anónimas derivadas de las conversaciones (sin contenido textual identificable) se conservan indefinidamente para fines de investigación científica, histórica y estadística.

8. Anonimización y Minimización de Datos

Aplicamos procesos de anonimización para garantizar la protección de tu identidad:

- Minimización: Solo recopilamos los datos estrictamente necesarios para cumplir con los objetivos de investigación.
- Uso anónimo: Los resultados académicos y publicaciones científicas garantizan el anonimato de los participantes.

Importante: Una vez que los datos se anonimizan de forma irreversible dejan de ser considerados “datos personales”.

Dado que NoVa+ funciona de manera anónima y no requiere registro de usuario, una vez que los datos han sido anonimizados y agregados a las estadísticas de investigación, no será posible identificarlos ni eliminarlos de forma individualizada.

9. Modificaciones a estas Políticas

El equipo de investigación podrá actualizar estas Políticas de Privacidad cuando sea necesario para reflejar cambios en nuestras prácticas, aspectos técnicos y legales.

Cualquier modificación será publicada en esta página con la fecha de actualización correspondiente. Se recomienda revisar periódicamente este documento.

10. Contacto

Si tienes preguntas, inquietudes o deseas ejercer tus derechos sobre tus datos personales, puedes contactarnos en:

Correo electrónico: programa.adicciones.unsl1@gmail.com

Aceptación de las Políticas

Al utilizar NoVa+, confirmas que has leído, comprendido y aceptado estas Políticas de Privacidad.
El uso del asistente conversacional implica tu consentimiento expreso a los términos aquí establecidos.

ANEXO II

Términos y Condiciones

1. Aceptación de los Términos

Bienvenido/a a NoVa+, un asistente virtual con Inteligencia Artificial para la prevención e investigación del juego online en jóvenes y adolescentes, desarrollado en la Universidad Nacional de San Luis. Participan en su diseño e implementación investigadores del área de Psicología e Informática.

Al acceder y utilizar este sistema, aceptas quedar vinculado/a por estos Términos y Condiciones, así como por nuestras Políticas de Privacidad.

2. Naturaleza del Sistema

2.1 Proyecto de Investigación

NoVa+ es un proyecto de investigación académica desarrollado por estudiantes de ingeniería en informática como herramienta para una investigación de un grupo de Psicología, enfocado en la evaluación de la efectividad y aceptación de un asistente virtual basado en inteligencia artificial para la prevención de la adicción al juego online en adolescentes.

Este sistema tiene carácter experimental y de investigación científica. Su objetivo es:

- Evaluar la efectividad de las intervenciones conversacionales basadas en inteligencia artificial para la prevención de adicciones al juego online
- Contribuir al desarrollo del conocimiento en el campo de la Psicología y la tecnología aplicada a la salud mental
- Brindar información, orientación y apoyo preventivo sobre la adicción al juego online y el uso responsable de las apuestas digitales

Como proyecto en desarrollo, NoVa+ puede presentar limitaciones propias de sistemas experimentales y su disponibilidad no está garantizada.

2.2 Finalidad del Sistema

El asistente virtual tiene como propósito brindar información educativa, orientación y apoyo preventivo relacionado con:

- Juego responsable
- Apuestas online
- Adicción al juego online
- Prevención de conductas problemáticas relacionadas con el juego

El sistema no sustituye la intervención profesional directa ni constituye una herramienta de diagnóstico, terapia o tratamiento psicológico.

2.3 Interacción con Inteligencia Artificial

Funcionamiento y Explicabilidad del Sistema

El asistente conversacional de NoVa+ funciona mediante:

- Tecnología: Modelo de lenguaje Gemini 2.5 Flash desarrollado por Google, complementado mediante un sistema de recuperación de información (RAG - Retrieval Augmented Generation)
- Contenido especializado: Las respuestas del asistente se basan en:
    - Documentos confiables y guías clínicas
    - Investigaciones académicas
    - Material de organizaciones de salud mental especializadas en adicción al juego
    - Documentos cuidadosamente seleccionados y validados por el equipo de psicólogos del proyecto
- Diseño supervisado: El comportamiento del asistente ha sido configurado mediante un prompt específico (instrucciones de sistema) diseñado conjuntamente por profesionales de la Psicología e informática y ha sido revisado y aprobado por psicólogos especializados

El sistema procesa tus mensajes, analiza el contexto de la conversación y genera respuestas personalizadas basándose en la información con la que ha sido entrenado y los documentos especializados proporcionados. El contenido de las respuestas puede variar en función del contexto, pero en ningún caso representa una recomendación que suplante a la de un profesional especializado.

3. Limitaciones del Sistema

NoVa+ NO reemplaza la atención de un profesional de la salud mental, ni constituye terapia psicológica, asesoramiento médico o diagnóstico clínico.

El asistente virtual es una herramienta de apoyo, información y prevención:

- No puede realizar diagnósticos de adicción al juego ni de ninguna otra condición psicológica o médica.
- No puede prescribir tratamientos ni ofrecer planes terapéuticos.
- No sustituye la evaluación profesional.
- No emite juicios clínicos.

3.2 Alcance Temático

El asistente de NoVa+ está específicamente diseñado para tratar temas relacionados con la prevención de adicción al juego online y apuestas digitales.

El sistema NO debe ser utilizado para consultas sobre otros temas. Si introduces consultas fuera de este alcance, el asistente te redirigirá al tema central o indicará que no puede ayudarte con ese tipo de solicitudes.

3.3 Situaciones de Emergencia

NoVa+ NO está diseñado para gestionar crisis o emergencias.

Si te encuentras en una situación de emergencia, crisis emocional, riesgo de autolesión o cualquier situación que requiera atención inmediata:

- Contacta inmediatamente con servicios de emergencia, líneas de crisis o profesionales de la salud mental

El sistema está programado para detectar situaciones de riesgo elevado y, en esos casos, proporcionará números de contacto útiles de servicios de emergencia y apoyo psicológico según tu ubicación geográfica. Sin embargo, esta funcionalidad no garantiza la detección de todas las situaciones de riesgo.

3.4 Errores

Aunque NoVa+ ha sido desarrollado con rigor académico y supervisión profesional, el asistente puede cometer errores, proporcionar información inexacta o incompleta, o no comprender correctamente tu consulta.

Los sistemas de inteligencia artificial tienen limitaciones inherentes y no son infalibles. El funcionamiento del sistema depende de modelos de lenguaje probabilísticos que pueden presentar errores, omisiones o interpretaciones inexactas. Por ello:

- Verifica información crítica con fuentes adicionales o profesionales
- No tomes decisiones importantes basándote únicamente en las respuestas del asistente
- Usa tu criterio personal al evaluar la información proporcionada

4. Requisitos de Uso

4.1 Edad Mínima

Para utilizar NoVa+ debes tener al menos 13 años de edad.

- Usuarios entre 13 y 17 años: Deben contar con la autorización expresa de un padre, madre o tutor legal para participar en este estudio de investigación

Al aceptar estos Términos y Condiciones, el usuario declara bajo su responsabilidad que cumple con los requisitos de edad.

4.2 Consentimiento Informado

El uso de NoVa+ requiere tu consentimiento libre, informado y expreso. Este consentimiento se otorga al marcar la casilla de aceptación en el formulario de Evaluación de Hábitos de Juego antes de acceder al asistente conversacional.

Al otorgar tu consentimiento, confirmas que:

- Has leído y comprendido estos Términos y Condiciones y las Políticas de Privacidad
- Comprendes que participas en un proyecto de investigación académica con fines científicos
- Aceptas que tus datos serán procesados conforme a lo descrito en las Políticas de Privacidad
- Eres mayor de 13 años y, si eres menor de 18, cuentas con autorización parental
- Proporcionarás información veraz en los formularios previos

5. Conductas Prohibidas

Para garantizar el correcto funcionamiento de NoVa+ y proteger la integridad de la investigación, quedan expresamente prohibidas las siguientes conductas:

5.1 Uso Inapropiado

- Manipular al asistente para alterar el funcionamiento del sistema
- Utilizar bots, scripts, herramientas automatizadas o cualquier medio que no sea la interacción humana directa para acceder al sistema
- Enviar mensajes repetitivos o con el único propósito de saturar el sistema
- Intentar acceder, modificar o extraer información del sistema o sus bases de datos
- Proporcionar deliberadamente información falsa, engañosa o fraudulenta en los formularios o durante las conversaciones
- Suplantar la identidad de otra persona
- Utilizar el asistente para acosar, intimidar, insultar o agredir verbalmente
- Usar lenguaje ofensivo, discriminatorio o que promueva el odio durante la interacción
- Introducir contenido ilegal, difamatorio, obsceno o que promueva actividades ilegales
- Usar el sistema con fines comerciales no autorizados o para generar contenido inapropiado
- Reproducir, copiar o distribuir el contenido generado sin autorización del equipo de investigación

5.4 Consecuencias del Incumplimiento

El incumplimiento de estas normas podrá derivar en la suspensión o bloqueo del acceso al Asistente. La Universidad Nacional de San Luis y el equipo de investigación no se hacen responsables de las consecuencias derivadas del uso indebido o prohibido del sistema por parte de los usuarios.

6. Derechos de Propiedad Intelectual

Todos los contenidos de NoVa+, incluyendo pero no limitado a:

- Software y código fuente
- Diseño, textos y gráficos
- Estructura y arquitectura del sistema
- Documentación especializada

Son propiedad del equipo de investigación del proyecto.

Los componentes tecnológicos provistos por terceros (como Google Gemini o Vercel) mantienen sus respectivos derechos de propiedad.

Queda prohibida la reproducción, distribución, modificación o cualquier forma de explotación del contenido de NoVa+ sin autorización expresa y por escrito del equipo de investigación y desarrollo.

7. Limitación de Responsabilidad

7.1 Uso Bajo Tu Propia Responsabilidad

El uso de NoVa+ es completamente voluntario y bajo tu propia responsabilidad.

Al utilizar este asistente, reconoces y aceptas que el equipo de NoVa+ no será responsable por cualquier daño, perjuicio, pérdida o inconveniente derivado de:

- El uso o la imposibilidad de uso de NoVa+
- Las decisiones o acciones que adoptes en base a la información brindada por el asistente
- Errores, inexactitudes, omisiones o interpretaciones inexactas en las respuestas del asistente
- Decisiones tomadas basándose en la información proporcionada por el sistema
- Interrupciones, caídas, fallos técnicos o problemas de disponibilidad del sistema
- Interrupciones temporales del sistema, errores en el procesamiento de datos o pérdida de información
- Daños directos o indirectos derivados del uso del sistema

7.2 Servicios de Terceros

NoVa+ utiliza servicios de terceros (Vercel para hosting, Neon para base de datos, API de Gemini para el asistente conversacional). El equipo no se hace responsable por interrupciones, errores o problemas de privacidad derivados de estos servicios externos.

7.3 Enlaces Externos

Si el asistente proporciona enlaces a recursos externos, estos se ofrecen únicamente con fines informativos. No controlamos ni respaldamos el contenido de sitios web de terceros y no asumimos responsabilidad por su contenido, políticas de privacidad o prácticas.

8. Disponibilidad del Servicio

No se garantiza la disponibilidad continua del servicio ni la ausencia de interrupciones, fallas técnicas o demoras.

Nos reservamos el derecho de:

- Modificar, suspender o discontinuar el servicio en cualquier momento sin previo aviso
- Realizar mantenimientos programados o no programados que puedan afectar la disponibilidad
- Finalizar el proyecto de investigación una vez cumplidos sus objetivos académicos

9. Modificaciones a los Términos y Condiciones

Si por razones excepcionales fuera necesario realizar modificaciones, se notificarán claramente en la plataforma y se solicitará la aceptación de los nuevos términos antes de continuar utilizando la aplicación.

10. Jurisdicción y Ley Aplicable

Estos Términos y Condiciones se rigen por las leyes de la República Argentina.

Cualquier controversia, conflicto o reclamación derivada de estos Términos y Condiciones o del uso del asistente será sometida a la jurisdicción exclusiva de los tribunales ordinarios de la ciudad de San Luis, Provincia de San Luis, Argentina, renunciando las partes a cualquier otro fuero o jurisdicción que pudiera corresponder.

11. Aceptación Final

El uso del asistente implica la aceptación plena y sin reservas de estos Términos y Condiciones y las Políticas de Privacidad correspondientes.

El uso del asistente debe ser informado y responsable, reconociendo sus límites como herramienta de apoyo y no de tratamiento.

12. Contacto

Si tienes preguntas, comentarios, inquietudes, consultas, reclamos o solicitudes relacionadas con estos Términos y Condiciones, el funcionamiento del asistente o el tratamiento de los datos personales, puedes contactarnos en:

Correo electrónico: programa.adicciones.unsl1@gmail.com

Al utilizar NoVa+, confirmas que has leído, comprendido y aceptado estos Términos y Condiciones en su totalidad.