

Creación de un prototipo de mascota virtual móvil con capacidades de interacción y adaptación emocional

**Kevin Andrés Acosta Rengifo
Mónica Andrea Cifuentes Salcedo**

**Universidad del Valle
Facultad de ingeniería
Escuela de ingeniería de sistemas y computación
Tuluá
2025**

Creación de un prototipo de mascota virtual móvil con capacidades de interacción y adaptación emocional

Kevin Andrés Acosta Rengifo
Código 202067578
acosta.kevin@correounivalle.edu.co

Mónica Andrea Cifuentes Salcedo
Código 202067536
cifuentes.monica@correounivalle.edu.co

Director
Joshua David Triana Madrid. Msc
joshua.triana@correounivalle.edu.co

Universidad del Valle
Facultad de ingeniería
Escuela de ingeniería de sistemas y computación
Tuluá
2025

Trabajo de grado presentado por
Kevin Andrés Acosta Rengifo
Mónica Andrea Cifuentes Salcedo
Como requisito parcial para la obtención del título de Ingeniero de Sistemas

Joshua David Triana Madrid
Director

Jurado

Jurado

A Dios por darnos la vida, la salud y permitirnos cumplir este sueño.
A nuestras familias por la compañía, el amor y la confianza depositada en nosotros.
A nuestros profesores por su guía en nuestro aprendizaje y crecimiento personal.
Al profesor Joshua Triana por ser nuestro guía para lograr desarrollar este trabajo.

Tabla de Contenido

1. Introducción	1
1.1. Formulación del problema	3
1.2. Descripción del problema	3
1.3. Objetivos	4
1.3.1. Objetivo general	4
1.3.2. Objetivos específicos	4
1.4. Alcance de la propuesta	5
1.5. Resultados obtenidos	5
1.6. Información sobre los capítulos	6
2. Marco referencial	7
2.1. Antecedentes	7
2.1.1. Interacción Emocional en IA	7
2.1.2. Mascotas Virtuales y su Evolución	7
2.1.3. IA en el bienestar emocional y la salud mental	8
2.1.4. Herramientas para la detección de emociones en texto	9
2.1.5. Chatbots	9
2.1.6. Análisis de sentimientos en práctica	10
2.2. Marco Teórico	11
2.2.1. De lo real a lo virtual: Impacto en el bienestar emocional	11
2.2.2. Inteligencia artificial y procesamiento del lenguaje natural en la interacción humano-computador	11
2.2.3. Diseño de Agentes conversacionales virtuales y simulación de personalidad.	12
2.3. Marco conceptual	13
3. Análisis crítico de la literatura	15
3.1. Análisis crítico de la literatura	15
3.1.1. Metodología de revisión	15
3.1.2. Clasificación y caracterización de los estudios revisados	15
3.1.3. Evolución Histórica de las Arquitecturas de Chatbots	16
3.2. Conclusión del Análisis Crítico	21
4. Librerías y modelos para reconocimiento de emociones	22
4.1. Herramientas y modelos investigados	22
4.2. Comparación de herramientas o modelos	23
4.3. Metodología de Evaluación	24

5. Sistema de Generación de Respuestas Personalizadas 29

5.1. Construcción y Validación del Corpus 29

5.1.1. Recopilación y Diseño de Técnicas 29

5.1.2. Validación por Expertos 30

5.2. Implementación del sistema RAG 30

5.2.1. Base de datos vectorial 31

5.2.2. Modelo de embeddings 32

5.2.3. Proceso de indexación de documentos 33

6. Desarrollo del Prototipo de Mascota Virtual Móvil 36

6.1. Diseño de la Mascota Virtual 36

6.1.1. Concepto Visual y justificación del diseño 36

6.1.2. Proceso de Diseño y Desarrollo en Rive 37

6.1.3. Representación Visual de los Estados Emocionales 38

6.2. Implementación del Chatbot con Análisis de Sentimientos 40

6.2.1. Arquitectura del Sistema Chatbot 40

6.2.2. Desarrollo del backend (FastAPI) 41

6.2.3. Despliegue e infraestructura del backend 50

6.2.4. Integración del Cliente Móvil (Flutter) 50

6.2.5. Interfaz de Usuario y Experiencia Visual 53

6.2.6. Casos de Uso del Sistema 57

6.2.7. Empaquetado y firma de la aplicación 59

7. Pruebas 61

7.1. Pruebas de software 61

7.1.1. Prueba de Rendimiento 61

7.1.2. Pruebas de LLM as a Judge 62

7.2. Pruebas de usabilidad 64

7.2.1. Evaluación experta basada en las heurísticas de Nielsen 65

7.2.2. Evaluación con usuarios finales 67

8. Conclusiones y trabajos futuros 72

8.1. Conclusiones 72

8.2. Trabajos futuros 74

9. Bibliografía 75

Anexos 82

Lista de Figuras

2.1. Línea del tiempo de las mascotas virtuales	8
2.2. Ciclo de procesamiento de lenguaje natural en sistemas conversacionales	12
4.1. Comparación de precisión entre modelos evaluados	26
4.2. Comparación de f1-score entre modelos evaluados	26
4.3. Comparación de recall entre modelos evaluados	27
5.1. Comparación de FAISS vs ChromaDB	31
5.2. Tiempo de respuesta promedio: modelos de vectorización	33
5.3. Proceso de indexación y vectorización de documentos PDF para almacenamiento en ChromaDB	33
5.4. Flujo de recuperación de información en el sistema RAG	34
6.1. Arquitectura del diseño en Rive	38
6.2. Representación visual de la mascota en sus seis estados emocionales.	40
6.3. Diagrama de componentes e interacción Cliente-Servidor.	41
6.4. Diagrama Entidad-Relación de la Base de Datos	49
6.5. Arquitectura en capas de la aplicación Flutter.	51
6.6. Flujo de integración de la mascota.	53
6.7. Wireframe inicial del módulo de chat	54
6.8. Wireframe inicial del módulo de bienestar	54
6.9. Pantalla de inicio de sesión	55
6.10. Interfaz de chat con saludo inicial de la mascota	55
6.11. Interfaz de chat durante conversación activa	55
6.12. Pantalla de bienestar emocional con semáforo y gráfica semanal	56
6.13. Semáforo emocional en rojo con alerta de apoyo psicológico	57
6.14. Diagrama de casos de usos	59
7.1. Estadísticas de tiempo de respuesta: Mínimo, promedio y Máximo	62
7.2. Niveles de fidelidad al contexto RAG: Mínimo, Promedio y Máximo	63
7.3. Niveles de Relevancia de la respuesta: Mínimo, Promedio y Máximo	64
7.4. Resultados prueba Jakob Nielsen: Visibilidad del estado del sistema	65
7.5. Resultados prueba Jakob Nielsen: Control y libertad del usuario	66
7.6. Resultados prueba Jakob Nielsen: Prevención de errores	66
7.7. Resultados prueba Jakob Nielsen: Ayuda y documentación	67
7.8. Claridad de los iconos y elementos visuales	69
7.9. Utilidad percibida de las respuestas del chatbot	69
7.10. Claridad del semáforo emocional	70

7.11. Probabilidad de recomendación de la aplicación 71

Lista de tablas

1.1. Resultados obtenidos según los objetivos específicos	5
1.2. Estructura de capítulos del documento	6
3.1. Distribución de estudios por dominio de aplicación	16
3.2. Modelos de Lenguaje a Gran Escala identificados en la literatura	19
3.3. Componentes de LangChain y su aplicación en salud mental	20
4.1. Comparación de características principales	24
6.1. Componentes de la Capa de Entrada	42
6.2. Endpoints del Módulo de Usuarios	42
6.3. Endpoints del Módulo de Chatbot	43
6.4. Funciones del Controlador de Usuarios	44
6.5. Funciones del Controlador de Chatbot	45
6.6. Umbrales del Semáforo Emocional	47
6.7. Pesos Asignados a las Emociones	48
6.8. Escala de Niveles Emocionales	48
7.1. Ejemplo de evaluación de métricas en interacción positiva	64
7.2. Resultados de la encuesta de usabilidad	68

Resumen

La inteligencia artificial es clave para la atención emocional, especialmente en estudiantes universitarios que enfrentan altos niveles de estrés, ansiedad y depresión. Las mascotas virtuales actuales presentan limitaciones significativas en adaptación emocional, lo que afecta su adopción y efectividad.

Este proyecto desarrolló un prototipo de mascota virtual móvil que utiliza inteligencia artificial para ofrecer soporte emocional adaptativo. Mediante la arquitectura RAG (Retrieval-Augmented Generation) y el modelo Pysentimiento para análisis de emociones en español, el sistema genera respuestas personalizadas basadas en técnicas psicoeducativas validadas. La implementación incluye una interfaz intuitiva con mascota animada, semáforo emocional y visualización de patrones afectivos.

Los resultados obtenidos demuestran que la aplicación cumple completamente con su objetivo de promover el bienestar emocional de los estudiantes de la Universidad del Valle, sede Tuluá, ofreciendo un soporte emocional funcional, efectivo y bien valorado por los usuarios finales.

Capítulo 1

Introducción

En la era digital, la tecnología ha avanzado significativamente, transformando la forma en que nos relacionamos e interactuamos con el mundo virtual. Una de las tendencias más significativas que ha emergido en el campo de la inteligencia artificial y la interacción digital es la creación de mascotas virtuales. Sin embargo, a pesar de los avances en animación e inteligencia artificial, la mayoría de estas mascotas carecen de una adaptación emocional auténtica, limitándose a respuestas predefinidas que no reflejan la complejidad de las emociones humanas. Como señala Cynthia Breazeal [1], pionera en robots emocionales, la capacidad de las máquinas para entender y responder a nuestras emociones abre nuevas fronteras en la manera en que interactuamos con la tecnología.

En este contexto, es relevante considerar que la inteligencia emocional es la capacidad de reconocer, entender y manejar nuestras propias emociones y las de los demás [2], subrayando la importancia de desarrollar tecnologías que no solo imiten, sino que realmente comprendan y respondan a las emociones humanas. La salud mental es un pilar fundamental del bienestar general, afectando todos los aspectos de nuestras vidas, desde la toma de decisiones hasta la calidad de las relaciones personales. Como señala la Organización Mundial de la Salud [3], la salud mental es esencial para nuestro bienestar general y calidad de vida; influye en cómo pensamos, sentimos y actuamos en el día a día.

En un mundo donde la ansiedad, el estrés y otros desafíos emocionales están constantemente presentes, es crucial encontrar formas de apoyo accesibles y efectivas que proporcionen ayuda emocional. Las tecnologías digitales, como las mascotas virtuales, tienen el potencial de ofrecer una compañía empática que contribuya al manejo de emociones. Por esta razón, este proyecto propone la creación de un prototipo de mascota virtual móvil como asistente conversacional que integre capacidades de interacción y adaptación emocional, específicamente dirigido a apoyar el bienestar emocional de los estudiantes de la Universidad del Valle, sede Tuluá. El objetivo es adaptar y entrenar una inteligencia artificial existente capaz de reconocer de manera dinámica las emociones del usuario mediante el procesamiento del texto ingresado, ajustando su comportamiento para proporcionar una interacción más empática y genuina. Utilizando técnicas de procesamiento de lenguaje natural, la mascota adaptará su conducta en función del estado emocional detectado, respondiendo textualmente de manera personalizada.

Se busca no solo innovar en la forma en que las mascotas virtuales interactúan con los usuarios, sino también ofrecer un apoyo valioso para el manejo de emociones. Como destacan Wada y Shibata [4], las mascotas virtuales pueden desempeñar un papel importante en la mejora de la salud mental al ofrecer apoyo emocional, reducir la soledad y proporcionar una forma de interacción social significativa. Reconociendo la importancia de la salud mental y el impacto de cómo se sienten las personas, esta mascota

virtual proporcionará una respuesta adaptativa y empática, ayudando a los estudiantes en la gestión de sus emociones y contribuyendo a su bienestar general.

1.1. Formulación del problema

¿Cómo puede desarrollarse una mascota virtual con capacidad de respuesta empática y personalizada para proporcionar un apoyo emocional efectivo y accesible a estudiantes de la universidad del valle?

1.2. Descripción del problema

Las mascotas virtuales, aunque prometedoras como herramientas de compañía, se quedan cortas en muchos aspectos cuando se trata de interactuar y conectarse emocionalmente, lo que limita su capacidad para ofrecer un apoyo emocional. A pesar de su popularidad, estas mascotas se basan en sistemas preprogramados que generan interacciones repetitivas y predecibles, carentes de la personalización y adaptabilidad necesarias para establecer vínculos emocionales profundos. La ausencia de empatía es particularmente notable en momentos de estrés, ansiedad o soledad, situaciones en las que los estudiantes más necesitan apoyo emocional y una sensación de comprensión.

Estudios como el de Krämer et al [5] han demostrado que las interacciones con mascotas virtuales suelen ser decepcionantes, generando frustración en los usuarios al no satisfacer sus expectativas de conexión emocional. La incapacidad de estas aplicaciones para reconocer y responder a las emociones complejas de los usuarios, así como su falta de personalización, limita su utilidad como herramientas de asistencia en salud mental.

Este cobra relevancia en el contexto actual debido a un aumento de los problemas de salud mental, particularmente entre los grupos demográficos más jóvenes y adolescentes. La OMS (2019 y 2023) [6],[7] señala que el suicidio es una de las principales causas de muerte en América Latina, representando el 30 % de los casos, mientras que los trastornos mentales, como la depresión, afectan a más de 264 millones de personas en todo el mundo. Además, los resultados del estudio de la OMS indican que aproximadamente del 20-25 % de los estudiantes universitarios a nivel mundial experimentan trastornos mentales, siendo la ansiedad y la depresión los más prevalentes. A pesar de la alta prevalencia, se estima que entre el 30-50 % de estos estudiantes no reciben la atención adecuada debido a factores como el estigma y la falta de recursos accesibles [8]. La falta de soluciones accesibles y adaptadas a las necesidades emocionales de los estudiantes resalta aún más la brecha en la atención efectiva.

1.3. Objetivos

1.3.1. Objetivo general

Desarrollar un prototipo de mascota virtual móvil con capacidades de interacción y adaptación emocional, orientado a promover el bienestar emocional de los estudiantes de la Universidad del Valle, sede Tuluá.

1.3.2. Objetivos específicos

1. Analizar críticamente la literatura científica sobre el diseño y uso de chatbots.
2. Investigar, seleccionar y evaluar un modelo de inteligencia artificial adecuado para el reconocimiento de emociones.
3. Entrenar un chatbot utilizando análisis de sentimientos para personalizar la respuesta emocional de la mascota, a través del uso de datasets psicológicos.
4. Desarrollar el prototipo de la mascota virtual móvil integrando las capacidades de interacción emocional, con el fin de ayudar al estudiante a su autoconocimiento y autorregulación emocional.
5. Evaluar la usabilidad de la aplicación y analizar el impacto de las recomendaciones emocionales generadas por la mascota virtual en estudiantes de la Universidad del valle, sede Tuluá.

1.4. Alcance de la propuesta

El alcance de este proyecto abarcó el diseño, desarrollo y evaluación de un prototipo funcional de una mascota virtual móvil, dicho prototipo fue implementado para analizar texto ingresado por el usuario, procesar el sentimiento expresado y generar respuestas adaptadas a sus emociones. El desarrollo estuvo dirigido a promover el bienestar emocional de los estudiantes de la Universidad del Valle, sede Tuluá, alcanzando un nivel de maduración TLR6. Este nivel implicó la validación del prototipo en un entorno cercano al de implementación real, mediante una demostración funcional que confirmó su capacidad para interactuar de manera efectiva y emocionalmente adaptativa. Dicho proceso permitió evaluar, de manera práctica y contextualizada, el impacto potencial de la aplicación en el bienestar emocional de los usuarios.

1.5. Resultados obtenidos

Objetivo específico	Resultado obtenido
Analizar críticamente la literatura científica sobre el diseño y uso de chatbots.	Documento que contiene los hallazgos de la literatura científica, identificando las técnicas más utilizadas, para el diseño y uso de chatbots.
Investigar, seleccionar y evaluar un modelo de inteligencia artificial adecuado para el reconocimiento de emociones.	Documento que presenta los modelos de IA más adecuados para el reconocimiento de emociones, destacando sus características, ventajas, limitaciones y una evaluación comparativa que justifique la selección final.
Entrenar un chatbot utilizando análisis de sentimientos para personalizar la respuesta emocional de la mascota, a través del uso de datasets psicológicos.	Chatbot funcional entrenado, con análisis de sentimientos, capaz de generar respuestas personalizadas basadas en las emociones detectadas y el contexto de la conversación.
Desarrollar el prototipo de la mascota virtual móvil integrando las capacidades de interacción emocional, con el fin de ayudar al estudiante a su autoconocimiento y autorregulación emocional.	Código fuente del prototipo, acompañado de diagramas de arquitectura y una interfaz visual funcional que permita la interacción emocional con el usuario.
Evaluar la usabilidad de la aplicación y analizar el impacto de las recomendaciones emocionales generadas por la mascota virtual en estudiantes de la Universidad del valle, sede Tuluá.	Informe de evaluación de usabilidad, basado en encuestas con usuarios finales, que incluye métricas como facilidad de uso, satisfacción de usuario, y áreas de mejora identificadas.

Tabla 1.1: Resultados obtenidos según los objetivos específicos

Fuente: Elaboración propia

1.6. Información sobre los capítulos

En la siguiente tabla 1.2 se muestra una descripción breve de cada capítulo del documento.

Capítulo	Descripción
Capítulo 2: Marco Referencial	Se explican los conceptos necesarios para comprender el problema y su solución. Se espera que el lector tenga conocimientos básicos en algunos temas relacionados.
Capítulo 3: Modelo	Se describen y analizan los modelos identificados en la literatura, detallando el proceso de selección y justificación de los enfoques adoptados.
Capítulo 4: Sistema de Generación de Respuestas Personalizadas	Se aborda la metodología para la selección y preparación del corpus de entrenamiento, junto con la implementación del proceso de Retrieval-Augmented Generation (RAG).
Capítulo 5: Desarrollo del prototipo	Se detalla el diseño de la mascota virtual, así como el desarrollo del Frontend y el Backend de la aplicación.
Capítulo 6: Pruebas	Se describen los tipos de pruebas de software realizadas al prototipo, las pruebas de usabilidad con usuarios finales y expertos, así como su posterior análisis.
Capítulo 7: Conclusiones y trabajos futuros	Se presentan las conclusiones finales del proyecto y los trabajos futuros que podrían realizarse para mejorar la aplicación propuesta en este trabajo de grado.

Tabla 1.2: Estructura de capítulos del documento

Fuente: Elaboración propia

Capítulo 2

Marco referencial

2.1. Antecedentes

2.1.1. Interacción Emocional en IA

El avance tecnológico ha generado una discusión acerca de cómo equipar a las máquinas con 'sentimientos'. A pesar de que la IA no tiene emociones reales, se han logrado progresos significativos en su capacidad para reconocer las emociones humanas. Este desarrollo permite a los sistemas de IA generar respuestas más naturales y empáticas, adaptándose a los estados emocionales de las personas y ofreciendo interacciones más cercanas a las humanas. Aunque una IA no puede sentir ni querer en sentido estricto, su capacidad para simular empatía y acompañamiento emocional es suficiente para que muchas personas experimenten relaciones de tipo afectivo con estas entidades [9].

En el contexto educativo, la empatía que refleja la IA, también ha mostrado un gran potencial. La inteligencia artificial está ayudando a mejorar la salud mental estudiantil, proporcionando sistemas que detectan señales de estrés o ansiedad en los estudiantes y sugieren actividades o intervenciones personalizadas para mejorar su bienestar[10]. Estas tecnologías se están integrando cada vez más en las universidades, ofreciendo herramientas que van más allá de la simple gestión de información académica, si no que también buscan apoyar el bienestar emocional de los estudiantes.

En la actualidad, los avances tecnológicos nos han llevado a un punto en el que es necesario reflexionar sobre la relevancia de la empatía en las máquinas. La Inteligencia Artificial (IA) está siendo cada vez más integrada en nuestra vida cotidiana, ya sea a través de asistentes de voz en nuestros dispositivos móviles o de algoritmos sofisticados que ayudan en la lucha contra diversas enfermedades [11].

2.1.2. Mascotas Virtuales y su Evolución

El origen de las mascotas virtuales se remonta a mediados de los años 90 con los Tamagotchis, que se convirtió en un fenómeno cultural y dejó una huella profunda en la juventud de la época. Este dispositivo de bolsillo, creado en Japón, se popularizó rápidamente y se convirtió en un accesorio esencial para niños y adolescentes. Con el paso del tiempo, el Tamagotchi dio paso a una serie de mascotas virtuales como Digimon, Neopets, Nintendogs, Petz y Webkinz, que expandieron la idea de interactuar con mascotas virtuales. Como resultado, las aplicaciones móviles también adoptaron esta tendencia, permitiendo a los usuarios cuidar de sus mascotas virtuales directamente desde sus dispositivos móviles. Estas experiencias sentaron las bases para la interacción entre humano y máquina, una dinámica que hoy sigue evolucionando con robots y sistemas de inteligencia artificial[12].

Las mascotas virtuales actuales siguen evolucionando y se han convertido en una parte fundamental del entretenimiento. Un ejemplo notable es Peridot, un juego de mascotas virtuales donde se utiliza la IA generativa, permitiendo que los Dots (mascotas virtuales del juego) reaccionen de manera más dinámica y realista a los estímulos del mundo real, como otros animales u objetos. Esto incrementa la variedad y complejidad de sus comportamientos [13]. Sin embargo, el enfoque principal de Peridot es lúdico, y aunque ofrece una experiencia significativa de usuario, no está centrado en el soporte emocional.

Aunque existen otras mascotas virtuales como Pou o My Cat-Virtual Pet Simulator, que brindan entretenimiento y compañía, ninguna integra específicamente capacidades conversacionales avanzadas con inteligencia artificial para ofrecer apoyo emocional significativo. A pesar de que algunas aplicaciones intentan simular interacciones más complejas, las mascotas virtuales no han alcanzado aún un nivel en el que puedan sustituir el apoyo emocional genuino proporcionado por otros medios, como los animales de compañía reales o sistemas de inteligencia artificial diseñados específicamente para ofrecer apoyo emocional.

A continuación, se presenta una línea de tiempo gráfica que resume la evolución de las mascotas virtuales:

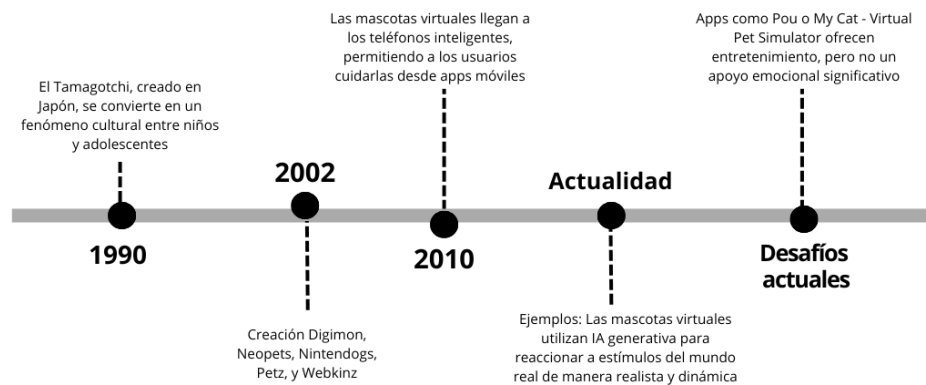


Figura 2.1: Línea del tiempo de las mascotas virtuales

Fuente: Elaboración propia

2.1.3. IA en el bienestar emocional y la salud mental

La aplicación de la inteligencia artificial (IA) para mejorar el bienestar emocional y la salud mental está ganando cada vez más relevancia y se expande de manera constante. Gracias a los avances en procesamiento de lenguaje natural (PLN) y algoritmos de aprendizaje automático, la IA está ofreciendo nuevas formas de intervención en el tratamiento de trastornos emocionales como la depresión, la ansiedad y el estrés. Estos avances han permitido mejorar el acceso a recursos de salud mental y ampliar las opciones de apoyo psicológico para diversas poblaciones. Un estudio realizado por Lopes et al [14] muestra que los chatbots de salud mental ofrecen ventajas significativas en términos de accesibilidad, rentabilidad y mejora del acceso a servicios de apoyo emocional. Sin embargo, el mismo estudio también resalta diversas limitaciones, como la ausencia de conexión humana, el riesgo de diagnósticos erróneos, las preocupaciones de privacidad, el sesgo y las limitaciones en la precisión de la evaluación de riesgos.

Un ejemplo de cómo la IA puede ser integrada para fomentar el bienestar emocional es MyUBot, un chatbot diseñado para ayudar a los estudiantes a gestionar síntomas de ansiedad y depresión. A través

de un sistema de preguntas y respuestas, MyUBot interactúa con los usuarios, permitiendo que tanto el estudiante como el bot aprendan y se adapten mutuamente [15]. Este tipo de tecnologías demuestra el potencial de la IA para ofrecer apoyo emocional personalizado, un concepto que también es aplicable en el desarrollo de mascotas virtuales con capacidades de adaptación emocional, como es el caso de este proyecto.

2.1.4. Herramientas para la detección de emociones en texto

Las herramientas para la detección de emociones en texto son usadas por investigadores, estudiantes y desarrolladores para analizar el contenido emocional presente en diversas fuentes, como redes sociales, chats y correos electrónicos, respondiendo a sus necesidades. Una de las herramientas más conocidas es IBM Watson Tone Analyzer, que nació como un reto de IBM Research para desarrollar un sistema informático que pudiera competir con los campeones en el juego de Jeopardy, y que ahora es usado para encontrar tonos de lenguaje y tonos emocionales [16].

Por otro lado, Google Cloud Natural Language permite a los interesados trabajar con distintos aspectos de las funciones del lenguaje, entre ellas el análisis de sentimiento, incorporando información sobre la opinión emocional predominante de un contenido seleccionado previamente y permitiendo obtener un score que analiza la valencia de las emociones con valores dicotómicos [17].

Estas herramientas no solo transforman la capacidad de analizar emociones en textos, sino que también representan un pilar para proyectos que buscan comprender y responder a emociones humanas. Su capacidad para identificar tonos emocionales y analizar sentimientos resulta esencial para desarrollar interacciones más empáticas y adaptativas, permitiendo que la mascota virtual ofrezca respuestas alineadas con el estado emocional del usuario.

2.1.5. Chatbots

El concepto de chatbots no es nuevo; su origen se remonta a la década de 1960, cuando se desarrollaron los primeros programas capaces de simular conversaciones humanas. Desde entonces, la tecnología que los impulsa ha experimentado cambios significativos [18], especialmente en áreas como el procesamiento del lenguaje natural (PLN) y la inteligencia artificial (IA). Gracias a estos avances, los chatbots modernos han alcanzado un nivel de sofisticación notable, permitiéndoles interactuar de manera más fluida y resolver tareas complejas en tiempo real.

En la actualidad, los chatbots son elementos esenciales en diversos aspectos de nuestra vida cotidiana. Desde los asistentes inteligentes integrados en dispositivos móviles, que ayudan con recordatorios y búsquedas rápidas, hasta los sistemas avanzados de atención al cliente, que resuelven dudas sin necesidad de esperar largas filas. Estas herramientas no solo simplifican la vida de los usuarios, sino que también optimizan procesos y proporcionan soluciones rápidas. Además, los chatbots han demostrado ser aliados clave en áreas como la salud, el comercio electrónico y la educación.

En el ámbito de la salud mental, Woebot es un agente conversacional automatizado diseñado para ofrecer terapia cognitivo-conductual mediante conversaciones diarias breves, mientras realiza un seguimiento constante del estado de ánimo del usuario. Esta herramienta se integra a una aplicación de mensajería instantánea multiplataforma, lo que permite su uso tanto en computadoras como en dispositivos móviles [19]. Woebot utiliza algoritmos de PLN para comprender las emociones del usuario y emplea árboles de decisiones para personalizar las respuestas, lo que facilita una interacción más fluida. Además, ayuda

en el establecimiento de metas y proporciona seguimiento emocional a través de gráficos semanales [20]. La efectividad de Woebot fue evaluada en un ensayo realizado con 70 estudiantes que experimentaban ansiedad y depresión, mostrando una reducción significativa de los síntomas entre los participantes [20].

Por otro lado, Replika se presenta como un compañero virtual diseñado para brindar apoyo emocional y compañía, alcanzando más de 30 millones de usuarios en 2024 [21]. Este chatbot aprende y adapta sus patrones conversacionales según cómo interactúa con el usuario, combinando modelos generativos con contenido de diálogo predefinido para optimizar la interacción. A medida que interactúa con el usuario, Replika ajusta sus respuestas para satisfacer mejor sus necesidades emocionales [22] [23].

Estos ejemplos demuestran el potencial de los chatbots para ofrecer apoyo emocional accesible y personalizado. Sin embargo, tanto Woebot como Replika se presentan principalmente como interfaces conversacionales tradicionales. La integración de estos principios con el concepto de mascota virtual representa una oportunidad para combinar la efectividad terapéutica de los chatbots de salud mental con la conexión emocional característica de las mascotas virtuales, creando así una experiencia más inmersiva y atractiva para los estudiantes universitarios.

2.1.6. Análisis de sentimientos en práctica

El análisis de sentimientos ha sido utilizado en diversas áreas para evaluar el estado emocional de los usuarios a partir de textos escritos. Un ejemplo relevante es el estudio *Impacto de Covid-19 en la salud mental: análisis de sentimientos usando NLTK*, donde se empleó procesamiento del lenguaje natural (PLN) para analizar la condición mental de los estudiantes durante la pandemia.

En este estudio, se recolectaron datos a través de una encuesta dirigida a estudiantes entre 18 y 22 años mediante un formulario de Google. Los participantes expresaron sus pensamientos y sentimientos sobre la pandemia. Posteriormente, se realizó un preprocesamiento de los datos, que incluyó la conversión de texto a minúsculas, eliminación de signos de puntuación, espacios vacíos, emojis, números y caracteres no alfabéticos.

Para la tokenización (proceso de dividir un texto en unidades más pequeñas, llamadas tokens), se utilizó la biblioteca NLTK para construir una colección de palabras. También se eliminaron las palabras irrelevantes (como artículos y conectores).

El análisis de sentimientos se llevó a cabo con la función de polaridad de NLTK, asignando puntuaciones positivas, negativas y neutrales a los textos. Los resultados se representaron mediante gráficos de frecuencia y nubes de palabras generadas con Matplotlib y PIL, lo cual facilitó visualizar la distribución de ideas en el conjunto de datos.

Finalmente, el modelo fue implementado en una aplicación web utilizando Flask, con una interfaz HTML básica que permitía visualizar los resultados desde un navegador. Este enfoque metodológico sirve como referencia valiosa para el desarrollo de sistemas de análisis emocional en texto, y puede ser aplicado en el contexto de mascotas virtuales para identificar los sentimientos de estudiantes universitarios durante conversaciones, permitiendo generar respuestas empáticas adaptadas a sus necesidades emocionales [24].

2.2. Marco Teórico

2.2.1. De lo real a lo virtual: Impacto en el bienestar emocional

La interacción con mascotas ha demostrado tener un impacto positivo significativo en el bienestar emocional de las personas, ayudando a reducir el estrés y mejorar la salud mental. Diversos estudios han señalado que los dueños de mascotas suelen reportar niveles más altos de felicidad en comparación con aquellos que no los tienen. Este fenómeno puede explicarse a través de mecanismos psicosociales y psicofisiológicos, como la liberación de oxitocina, asociada al apego y la confianza, y la reducción de los niveles de cortisol, la hormona primaria del estrés [25]. Un estudio realizado en 2014 [26] encontró que la presencia de una mascota, especialmente un perro, durante situaciones de estrés disminuía notablemente la ansiedad y el estrés de los participantes, sustentando así la base biológica de este vínculo.

Este impacto es particularmente relevante en poblaciones bajo estrés crónico, como los universitarios. La transición a la vida universitaria, caracterizada por altas demandas académicas y, en muchos casos, el distanciamiento del núcleo familiar, genera una vulnerabilidad emocional donde la ansiedad y la soledad son factores comunes. En este contexto, la conexión emocional se convierte en un apoyo esencial. El trabajo de Guzmán y Valencia [27] corrobora esto, al encontrar que los estudiantes que tienen o interactúan con animales no solo reportan menor estrés, sino que tienden a mostrar una mayor empatía hacia otras personas. Esto sugiere que los beneficios se extienden más allá del alivio inmediato, fomentando habilidades socioemocionales duraderas.

Partiendo de la evidencia sobre el vínculo humano-animal, la evolución natural hacia el entorno digital ha dado lugar al desarrollo de mascotas virtuales. Aunque sus capacidades para ofrecer apoyo emocional profundo son aún incipientes, estas tecnologías, como los chatbots y acompañantes virtuales, utilizan inteligencia artificial para generar interacciones empáticas con los usuarios. Estos avances representan un punto de partida crucial para el desarrollo de herramientas tecnológicas escalables que brinden soporte emocional accesible, especialmente en entornos donde el acceso a una mascota real es limitado.

2.2.2. Inteligencia artificial y procesamiento del lenguaje natural en la interacción humano-computador

La inteligencia artificial (IA) ha evolucionado para permitir interacciones entre humanos y computadoras que se asemejan cada vez más a la comunicación humana. Este campo persigue dos objetivos principales, que resultan complementarios, por un lado, un objetivo tecnológico que se centra en usar sistemas computacionales, para realizar tareas útiles de forma eficiente (a veces empleando métodos muy distintos a los de la mente), y, por otro, un objetivo científico, que busca usar conceptos y modelos de IA para comprender mejor facultades humanas como el lenguaje y la emoción. Esta dualidad ofrece una base sólida para enriquecer la interacción humano-computadora.

En el corazón de esta evolución se encuentra el procesamiento del lenguaje natural (PLN), la rama de la IA que dota a las máquinas de la capacidad de comprender, interpretar y generar lenguaje humano de manera útil. El PLN utiliza técnicas como el análisis sintáctico, semántico y de sentimientos para procesar el texto del usuario. Esto permite identificar no solo el significado literal de las palabras, sino también las emociones subyacentes y la intención comunicativa. Los avances en este ámbito han proporcionado, de hecho, una base de conocimientos para crear sistemas que comprendan aspectos de la mente humana [28]. En la práctica, esto se traduce en una colaboración donde el usuario y la computadora se comunican a través de un diálogo estructurado que va más allá de comandos básicos [29]. Este diálogo, facilitado por el PLN, permite que los sistemas comprendan y respondan a las intenciones y emociones del usuario,

haciendo que la interacción sea fluida y personalizada.

Estos avances en IA y PLN no solo mejoran la comprensión del lenguaje, sino que permiten una interacción adaptativa. Gracias a ello, agentes virtuales como la mascota virtual propuesta en este proyecto pueden analizar el contenido emocional del texto del usuario, detectar su estado anímico, y generar respuestas contextualmente apropiadas que reflejen empatía y comprensión. Esto hace posible que el sistema ofrezca una experiencia más inmersiva, donde el usuario se sienta que interactúa con una entidad "viva", capaz de entender y reaccionar a sus emociones y necesidades.

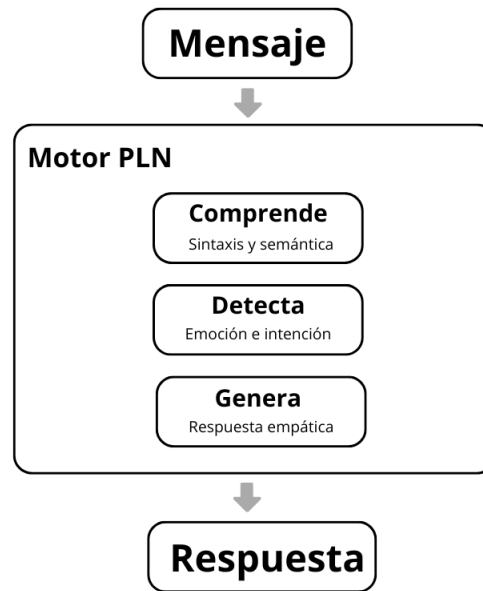


Figura 2.2: Ciclo de procesamiento de lenguaje natural en sistemas conversacionales

Fuente: Elaboración propia

2.2.3. Diseño de Agentes conversacionales virtuales y simulación de personalidad.

Los agentes conversacionales virtuales son aplicaciones de software diseñadas para entablar un diálogo con humanos mediante el uso del lenguaje natural. Entre estos, los chatbots destacan por su capacidad para interactuar a través de texto [30]. Desde sus orígenes con Eliza, desarrollado en la década de 1960, hasta modelos avanzados como GPT, estas herramientas han evolucionado notablemente gracias a los avances en el procesamiento de lenguaje natural (PLN) y el aprendizaje automático.

El PLN permite funciones esenciales como la tokenización, normalización del texto y corrección ortográfica. A partir de esta información procesada, el agente realiza búsquedas y compara patrones lingüísticos en su base de datos para generar respuestas precisas [31]. Los actuales modelos como GPT pueden entender el lenguaje natural gracias a su red neuronal. Una red neuronal es un modelo matemático, una red interconectada y organizada con base en una jerarquía diseñada para que una máquina simule el sistema nervioso humano interactuando con el mundo real [32].

En el ámbito de los agentes virtuales, la simulación de una personalidad se ha convertido en un componente clave para lograr interacciones más naturales y afectivas. Dicha personalidad, entendida como el conjunto de patrones consistentes en la comunicación que se transmiten a través del estilo (tono, lenguaje

y modo de interacción), encuentra sustento en investigaciones recientes. Un estudio concluye que, efectivamente, el estilo comunicativo de los agentes conversacionales tiene un impacto significativo en cómo los usuarios perciben la personalidad del agente, lo que demuestra que la personalidad no solo depende de características internas, sino que puede ser creada a través de la comunicación[33]. En el contexto de una mascota virtual para apoyo emocional, resulta especialmente relevante diseñar una personalidad empática, cálida y paciente, que permita generar confianza y conexión emocional con los estudiantes universitarios, facilitando así un espacio seguro para una buena interacción emocional.

Para que esta personalidad simulada sea efectiva en contextos de apoyo emocional, el estudio de la emoción en el lenguaje es de vital importancia. Esto ha dado lugar a una subdisciplina dentro del PLN, llamada análisis de sentimiento, cuyo objetivo es identificar y extraer el contenido subjetivo y emocional del texto [31]. Este análisis se integra con la personalidad del agente para generar respuestas no solo contextuales sino también emocionalmente apropiadas, permitiendo que la mascota virtual detecte el estado anímico del usuario y responda de manera coherente con su perfil empático.

2.3. Marco conceptual

Inteligencia Artificial (IA)

La IA es la capacidad de las máquinas para usar algoritmos, aprender de los datos y utilizar lo aprendido en la toma de decisiones tal y como lo haría un ser humano. Sin embargo, a diferencia de las personas, los dispositivos basados en IA no necesitan descansar y pueden analizar grandes volúmenes de información a la vez [34].

La IA es crucial en este proyecto, porque permite que la mascota virtual simule comportamientos que se adaptan y evolucionan de acuerdo con las interacciones del usuario. La mascota, gracias a la IA, puede aprender sobre el usuario y ajustar sus respuestas y comportamiento, para ofrecer así una experiencia más satisfactoria y adaptable emocionalmente. Esto no solo potencializa la interacción humano-máquina, sino que es un buen punto de partida para la relación empática entre la mascota y el usuario.

Procesamiento de Lenguaje Natural (PLN)

El procesamiento de lenguaje natural (PLN) es un subcampo de la informática y la inteligencia artificial (IA) que emplea el aprendizaje automático para permitir que las computadoras comprendan y se comuniquen con el lenguaje humano. Permite que las computadoras y los dispositivos digitales reconozcan, comprendan y generen texto y voz mediante la combinación de la lingüística computacional (el modelado basado en reglas del lenguaje humano) y el modelado estadístico, el aprendizaje automático y el aprendizaje profundo [35].

El PLN es fundamental para interpretar y responder adecuadamente a lo expresado por el usuario. Al poder comprender el lenguaje natural, la mascota va a crear una interacción empática, dado que podrá interpretar de manera más precisa los sentimientos e intenciones del usuario, ajustando las respuestas al estado emocional del usuario. Esto hace que la mascota pueda brindar un soporte emocional más genuino.

Aprendizaje Automático (ML)

Es un subcampo de la IA que se centra en entrenar algoritmos para aprender y mejorar automáticamente de la experiencia sin intervención humana directa. Su objetivo es descubrir patrones en datos

estructurados y no estructurados a través de modelos matemáticos contruidos con un conjunto de datos de entrenamiento [36].

El ML es fundamental para entrenar algoritmos que permitan a la mascota virtual aprender de sus interacciones previas y ajustar su comportamiento según los patrones detectados. Además, desempeña un papel clave en la interpretación y adaptación a las emociones del usuario, logrando esto a través de modelos que analizan y procesan los datos de entrada de manera dinámica.

Interacción Humano-Computadora (HCI)

Es una disciplina que estudia cómo los seres humanos interactúan con las computadoras. El objetivo principal de la HCI es facilitar las interacciones entre los usuarios finales y los sistemas computacionales, haciendo que estos sean más utilizables y adaptables a los requisitos de los usuarios [37]. Su propósito radica en mejorar la usabilidad y accesibilidad de las aplicaciones, promoviendo una interacción más eficiente y efectiva.

La HCI es esencial porque posibilita una interacción más natural y efectiva entre los usuarios y las máquinas. Esto resulta especialmente relevante en el contexto de las mascotas virtuales, ya que contribuye a ofrecer una experiencia de usuario más enriquecedora, facilitando una interacción más intuitiva, natural y satisfactoria.

Modelos emocionales y teoría de las emociones básicas

En el ámbito de la informática afectiva y el desarrollo de agentes inteligentes, un modelo emocional es una estructura teórica que permite categorizar, medir y representar los estados afectivos del usuario. La adopción de un modelo específico es fundamental para que el sistema pueda transformar los datos analizados en una respuesta coherente (interacción).

Este trabajo se apoya en la Teoría de las Emociones Básicas de Paul Ekman [38], la cual postula que el ser humano posee un conjunto de emociones discretas, universales y biológicamente programadas. A diferencia de los modelos dimensionales, la teoría de Ekman define categorías cerradas que se manifiestan a través de expresiones faciales involuntarias y constantes en todas las culturas. Las seis emociones primarias definidas en este modelo son:

- **Ira:** Respuesta ante la frustración o amenaza, que prepara al organismo para la defensa.
- **Asco:** Reacción de rechazo o protección ante estímulos potencialmente tóxicos o aversivos.
- **Miedo:** Activación defensiva ante peligros reales o percibidos, orientado a la supervivencia.
- **Alegría:** Estado placentero asociado a una valoración positiva de la situación y altos niveles de energía.
- **Tristeza:** Estado de decaimiento anímico ante una pérdida o situación negativa, caracterizado por baja energía.
- **Sorpresa:** Reacción física y cognitiva breve ante un evento inesperado o que no entraba en los esquemas mentales de la persona.

La universalidad de las expresiones faciales asociadas a estas emociones permite a los sistemas digitales simular empatía mediante señales visuales reconocibles.

Capítulo 3

Análisis crítico de la literatura

3.1. Análisis crítico de la literatura

3.1.1. Metodología de revisión

Se realizó una revisión sistemática de 30 artículos científicos publicados entre 2018-2024 en bases de datos IEEE Xplore, Scopus y ScienceDirect. Los criterios de inclusión fueron: estudios empíricos sobre chatbots con capacidades emocionales, procesamiento de lenguaje natural aplicado a salud mental, y arquitecturas conversacionales avanzadas. La búsqueda se focalizó en investigaciones que presentaran implementaciones concretas o evaluaciones de chatbots.

3.1.2. Clasificación y caracterización de los estudios revisados

Del análisis de los 30 estudios seleccionados, se identificó la distribución por dominios de aplicación que se presenta en la tabla 3.1.

Esta distribución revela tres focos principales de investigación: la aplicación práctica en entornos empresariales (30 %), el desarrollo de tecnologías base (27 %) y de salud mental (23 %), siendo la salud mental el dominio aplicado más representado.

Dominio de aplicación	Cantidad	Porcentaje	Referencias
Salud mental	7	23 %	[39, 40, 41, 42, 43, 44, 45]
Salud general	2	7 %	[46, 47]
Soporte Empresarial/Técnico	9	30 %	[48, 49, 50, 51, 52, 53, 54, 55, 56]
Educación	4	13 %	[57, 58, 59, 60]
Estudios generales	8	27 %	[61, 62, 63, 64, 65, 66, 67, 68]
Total	30	100 %	

Tabla 3.1: Distribución de estudios por dominio de aplicación

Fuente: Elaboración propia

3.1.3. Evolución Histórica de las Arquitecturas de Chatbots

Primera generación: La rigidez de los modelos basados en reglas

El inicio de los asistentes conversacionales se fundamentó en dos enfoques principales que marcaron la primera generación de chatbots. El primero se basó en el sistema de reglas predefinidas, caracterizado por ser rígido; el segundo introdujo el uso de procesamiento de lenguaje natural clásico y métodos estadísticos, buscando mejorar la flexibilidad en la interpretación de la entrada de los usuarios.

Los primeros intentos de simular la conversación humana se basaron en árboles de decisión y reconocimiento de patrones. Como se analiza en [61], este enfoque (AIML) utilizó un lenguaje de marcado similar a XML para crear “categorías” que mapeaban directamente una entrada del usuario a una respuesta específica.

Si bien este método era fácil de entender y de implementar para diálogos simples, su capacidad de interpretación contextual fue limitada. La respuesta del chatbot dependía de coincidencias exactas o simples, sin una comprensión profunda del discurso. Esto se evidenció en dificultades para resolver referencias anafóricas (hacer referencia a un objeto previo en una conversación), ambigüedad léxica u omisión, lo que restringía la fluidez y coherencia del diálogo. A pesar de estas limitaciones, AIML representó un avance significativo al permitir la implementación de interfaces conversacionales accesibles y reutilizables, marcando un gran comienzo en la evolución de los sistemas de diálogo automatizados [61].

Segunda generación: Aproximaciones estadísticas y deep learning clásico

La evolución de los sistemas basados en reglas hacia aproximaciones estadísticas marcó un avance significativo. En lugar de depender exclusivamente de patrones predefinidos, estos sistemas comenzaron a aprender a partir de datos. El análisis de la literatura identifica técnicas clave en esta generación:

- **TF-IDF (Term Frequency-Inverse Document Frequency):** Mencionado en tres estudios [49, 47, 44], se empleó para calcular la relevancia de palabras clave. Por ejemplo, en el chatbot de

comercio electrónico [49], TF-IDF facilitaba la búsqueda de productos, mientras que en el sistema de salud mental [44], ayudaba a identificar términos emocionales relevantes.

- **Clasificadores Naïve Bayes:** Implementados en sistemas empresariales [48] para identificar intenciones detrás de consultas técnicas. Si bien demostró eficacia para categorizar solicitudes en el ámbito de ITSM, su suposición de independencia entre palabras lo hacía inadecuado para el análisis emocional complejo, donde el contexto era determinante.
- **K-Vecinos más Cercanos (KNN):** Utilizado en diagnóstico médico [46] para clasificar enfermedades basadas en síntomas. Aunque funcional para condiciones con síntomas físicos discretos, su aplicación a estados emocionales resulta limitada debido a la naturaleza continua, subjetiva y multidimensional de las emociones.

La limitación fundamental de estos métodos radica en su tratamiento del texto como una "bolsa de palabras", ignorando el orden secuencial y las dependencias sintácticas. Esta aproximación es particularmente problemática en salud mental. Un hallazgo crucial surge del análisis del chatbot médico [46]: herramientas estadísticas como KNN son insuficientes para captar la complejidad de este dominio. Estas técnicas asumen que los síntomas se agrupan de manera ordenada, pero la realidad clínica es más desordenada. Un paciente con ansiedad puede presentar tanto nerviosismo como agotamiento, dos manifestaciones aparentemente opuestas. Forzar esta realidad en un modelo que busca 'vecindades' simples de síntomas conduce a errores de diagnóstico, demostrando la necesidad de enfoques que comprendan estas relaciones no lineales.

En conclusión, las técnicas estadísticas clásicas marcaron un avance al permitir el aprendizaje desde datos, pero mantuvieron una limitación crítica: no capturaban dependencias secuenciales ni contexto a largo plazo. En el dominio de salud mental, donde las emociones se expresan de forma indirecta y contextual (por ejemplo, "Me siento como si nadara contra la corriente"), estos métodos resultaron insuficientes para una comprensión profunda. No obstante, la literatura sugiere que estas técnicas aún poseen valor como componentes auxiliares en sistemas híbridos. El clasificador Random Forest en MindWell [40], por ejemplo, logró un 89 % de precisión en la detección inicial, pero requirió complementarse con modelos contextuales más avanzados para la generación de respuestas empáticas.

Tercera generación: Deep Learning y la Revolución del Contexto Secuencial

La incapacidad de los modelos estadísticos (como TF-IDF) para capturar el orden de las palabras y el contexto en diálogos prolongados llevó a la adopción del aprendizaje profundo (Deep Learning). Este paradigma, basado en redes neuronales artificiales con múltiples capas, permitió a los chatbots adquirir una memoria secuencial que transformó radicalmente la calidad de la interacción.

Las redes neuronales se convirtieron en la base para el modelado de datos secuenciales, como el lenguaje natural. Su estructura cíclica les permite retener información de entradas anteriores, lo cual es necesario para el análisis de texto y la interpretación del contexto [64].

Los modelos más utilizados en esta fase son las LSTM (Long Short-Term Memory) y las GRU (Gated Recurrent Unit), diseñadas específicamente para el problema del "desvanecimiento del gradiente" de las RNN tradicionales, la solución a la memoria a largo plazo.

Estos dos modelos controlan el flujo de información, permitiendo al modelo recordar información relevante durante periodos prolongados y detectar patrones emocionales en textos largos [62, 64]. Esta capacidad

es crucial para los chatbots emocionales. Por ejemplo, el modelo conversacional [64] utiliza LSTM para mantener el historial emocional y capturar el contexto. En el desarrollo de chatbots multi-turno, los modelos Seq2Seq (secuencia a secuencia) basados en GRU son empleados para codificar el contexto de la conversación y generar respuestas fluidas [63].

Si bien estos modelos permitieron a los chatbots generar respuestas coherentes basadas en la conversación previa [63] y capturar el contexto, conllevan desafíos. Principalmente, su entrenamiento es muy complejo y requiere un alto costo computacional, además de que su capacidad para hacer predicciones precisas se limita si la conversación es muy corta [62, 64].

A pesar de estos avances, las arquitecturas recurrentes mantuvieron tres limitaciones fundamentales:

1. **Procesamiento secuencial obligatorio:** Las RNN debían procesar el texto palabra por palabra de forma secuencial, imposibilitando la paralelización del entrenamiento e inferencia. Esto generaba latencias perceptibles en conversaciones largas [65].
2. **Degradación de dependencias lejanas:** Aunque LSTM mejoró la capacidad de memoria, las relaciones entre palabras separadas por más de 100 tokens aún perdían fuerza en la representación interna, limitando la comprensión de contextos extensos [62, 64].
3. **Paradoja de la brevedad en salud mental:** Como documentaron los estudios [62, 64], estos modelos requerían conversaciones extensas para generar predicciones precisas. Sin embargo, usuarios en situaciones de crisis emocional frecuentemente proporcionaban respuestas muy breves, precisamente cuando más se necesitaba comprensión contextual profunda.

Estas limitaciones establecieron la necesidad de una arquitectura capaz de: (1) procesar texto en paralelo para mayor eficiencia, (2) capturar dependencias independientemente de la distancia en el texto, y (3) funcionar efectivamente con entradas cortas. Los Transformers resolvieron estos tres desafíos simultáneamente.

Cuarta generación: Transformers y la Comprensión Contextual Bidireccional

La arquitectura Transformer introduce un cambio fundamental, en lugar de procesar texto palabra por palabra secuencialmente, utiliza mecanismos de atención que permiten examinar simultáneamente todas las palabras de una oración, identificando relaciones de dependencia independientemente de la distancia [42, 65, 66].

El análisis del corpus evidencia la predominancia de BERT (Bidirectional Encoder Representations from Transformers) como modelo de referencia, presente en 6 de los estudios analizados [62, 41, 42, 65, 67, 68]. Su arquitectura bidireccional demuestra especial eficacia en la interpretación de construcciones lingüísticas complejas características del discurso en salud mental, donde la negación y las referencias contextuales resultan determinantes para la correcta interpretación emocional.

La evidencia demuestra una tendencia creciente hacia implementaciones híbridas donde los Transformers especializados en comprensión contextual se complementan con modelos generativos avanzados [52, 42, 54]. Este patrón arquitectónico anticipa la evolución hacia la quinta generación de sistemas conversacionales, estableciendo los cimientos para la integración con Modelos de Lenguaje a Gran Escala (LLMs).

La cuarta generación constituye así un punto de inflexión en la madurez tecnológica del procesamiento de lenguaje natural, resolviendo definitivamente el desafío de la comprensión contextual profunda mientras revela nuevas fronteras en la generación conversacional adaptativa.

Quinta generación: Modelos de Lenguaje de Gran Escala (LLM)

La introducción de Modelos de Lenguaje a Gran Escala (LLMs) marcó un cambio paradigmático en el desarrollo de sistemas conversacionales. A diferencia de aproximaciones anteriores que requerían entrenamiento especializado para tareas específicas, estos modelos se pre-entrenaron en corpus masivos, desarrollando capacidades emergentes de razonamiento contextual, generación de lenguaje natural y seguimiento de instrucciones sin programación explícita.

El análisis del corpus identifica una diversificación en la adopción de LLMs, como se documenta en la tabla 3.2.

Modelos	Artículos	Parámetros
GPT-4	[39, 45, 66]	~1,7T (estimado)
LLaMA2	[39, 45, 66]	7B-70B
Halcón	[58]	7B-40B
ChatGLM	[58]	6.2B

Tabla 3.2: Modelos de Lenguaje a Gran Escala identificados en la literatura
Fuente: Elaboración propia

El estudio MindGuide [39] documenta una implementación de GPT-4 en salud mental, revelando capacidades cualitativamente superiores:

- 1. **Razonamiento contextual complejo:** El sistema demostró capacidad para interpretar expresiones indirectas sin requerir entrenamiento específico [39].
- 2. **Seguimiento de instrucciones terapéuticas:** Mantuvo adherencia del 95 % a prompts estructurados en 200 conversaciones simuladas [39].
- 3. **Generación de respuestas naturales:** Produjo texto fluido y adaptado al tono emocional, superando la rigidez de sistemas como RASA [41].

Sin embargo, el estudio MindGuide identificó una limitación fundamental compartida por todos los LLMs: la ausencia de memoria entre sesiones. Los LLMs son sistemas “sin estado”, lo que significa que cada interacción es completamente independiente de las anteriores una vez finalizada la sesión. Por ende, cuando el usuario quiera interactuar de nuevo con el sistema, este no recordará nada haciendo que la interacción no sea fluida y no se pueden detectar patrones emocionales a lo largo del tiempo. Esta limitación destruye la continuidad terapéutica, obligando al usuario a re-contextualizar completamente su situación en cada sesión, lo cual puede ser retraumatizante cuando se trata de eventos difíciles o traumáticos que deben ser re-narrados repetidamente.

Para mitigar este problema dentro de una sesión activa, MindGuide implementó `ConversationBufferMemory` de LangChain, componente que mantiene el historial completo de la conversación en memoria temporal. Esta solución funcionó efectivamente durante la sesión activa, pero la memoria no persistió entre sesiones separadas por tiempo (típicamente tras 1-2 horas de inactividad o al cerrar el navegador).

Frameworks de orquestación

Los sistemas conversacionales contemporáneos han evolucionado hacia arquitecturas modulares que requieren la integración coordinada de múltiples componentes especializados: modelos de lenguaje (LLM), sistemas de recuperación (RAG), memoria conversacional, procesamiento de embeddings, gestión de prompts y conectores a fuentes de datos externos.

El análisis del corpus identifica dos categorías principales de soluciones: frameworks de orquestación general (LangChain, LlamaIndex), diseñados para componer flujos personalizados, y frameworks de desarrollo completo (RASA, DialogFlow), que proporcionan ecosistemas integrados. LangChain emerge como la herramienta predominante, presente en 11 de los 30 estudios analizados (37%), constituyéndose como el estándar para la construcción de chatbots modernos.

La tabla 3.3 detalla los componentes de LangChain más utilizados y su aplicación específica en el dominio de salud mental, evidenciando patrones de implementación recurrentes.

Componente	Función principal	Uso en salud mental	Artículo
ConversationBufferMemory	Mantener histórico	Recordar contexto emocional	[39, 58, 52]
RecursiveCharacterTextSplitter	Fragmentar documentos	Dividir guías terapéuticas largas	[52, 67]
FAISS/Milvus	Vectorial	Recuperar técnicas relevantes	[50, 52, 54]
PromptTemplate	Estructurar prompts	Guiar respuestas empáticas	[52, 66]
ConversationalRetrievalChain	RAG conversacional	Combinar contexto + conocimiento	[39, 54, 55]

Tabla 3.3: Componentes de LangChain y su aplicación en salud mental
Fuente: Elaboración propia

FAISS y Milvus no son componentes nativos de LangChain, sino bases de datos vectoriales externas. Sin embargo, LangChain proporciona conectores pre-construidos (`VectorStoreRetriever`) que facilitan su integración, razón por la cual se emplearon frecuentemente en conjunto en los estudios analizados [50, 52, 54]. Estas bases vectoriales son esenciales para implementar RAG, almacenando representaciones vectoriales de documentos que permiten búsquedas semánticas eficientes.

RAG

La técnica de Generación Aumentada por Recuperación (RAG) se consolidó como fundamental para la especialización de chatbots. Si bien aparece explícitamente mencionada en 6 estudios [50, 51, 52, 67, 55, 56], su implementación implícita —inferida a partir de descripciones arquitectónicas— está presente en 12 estudios adicionales. Esta mención en el corpus revela que RAG se ha establecido como la técnica preferida para crear sistemas especializados.

Esta adopción masiva respondió directamente a una limitación inherente de los LLMs: si bien poseen conocimiento amplio y general, adolecen de superficialidad en dominios técnicos o especializados. Adicionalmente, los LLM tienden a “alucinar” (generar información convincente, pero incorrecta) cuando enfrentan preguntas fuera de su distribución de entrenamiento.

RAG se consolidó como técnica esencial para chatbots especializados. Su capacidad para reducir alucinaciones en 72 % [55] mientras permite actualización de conocimiento sin reentrenamiento costoso.

3.2. Conclusión del Análisis Crítico

Este análisis trazó la evolución de las arquitecturas de chatbots a través de 5 generaciones tecnológicas. La revisión de los 30 estudios revela una trayectoria clara: la tecnología ha migrado desde sistemas rígidos basados en reglas y los modelos estadísticos, a Transformers que resolvían el desafío de la comprensión contextual profunda.

Sin embargo, la Quinta generación y los frameworks de orquestación asociados marcan el estado del arte actual y la solución a los desafíos centrales de este trabajo de grado:

- Los LLMs como GPT-4 [39] y Llama2 [52, 55] pueden generar respuestas fluidas y coherentes, superando la naturaleza “robótica” de las generaciones anteriores.
- El principal desafío de los LLMs es su falta de conocimiento especializado y su tendencia a la “alucinación”. Este análisis demuestra que la Generación Aumentada por Recuperación (RAG) se ha consolidado como la técnica central y preferida para solucionar este problema. RAG permite “aterrizar” al LLM en un cuerpo de conocimiento válido en el contexto de este proyecto (“los textos de apoyo psicológico”), sin incurrir en altos costos de reentrenamiento.
- Las bases de datos vectoriales como Faiss o ChromaDB y los LLMs requieren un sistema de conexión entre ellos, y funciones que faciliten la implementación de un prototipo de sistema conversacional eficiente y personalizado. La literatura identifica a LangChain como el framework de orquestación dominante [39, 58, 52, 54, 56], valorado por su simplicidad para integrar múltiples procesos y modelos para cualquier uso.

Capítulo 4

Librerías y modelos para reconocimiento de emociones

Se realiza una evaluación de las librerías y modelos, disponibles y pre-entrenados, mediante pruebas con conjuntos de datos reales de reconocimiento emocional. La selección de esta herramienta es un componente crítico para la mascota virtual, ya que determina la capacidad del sistema para:

- Detectar el estado anímico del estudiante en tiempo real durante la conversación
- Adaptar respuestas empáticas apropiadas al contexto emocional identificado
- Mantener coherencia emocional a lo largo de múltiples interacciones
- Identificar momentos que requieren escalamiento a apoyo profesional humano

Para la elección se tuvieron en cuenta criterios como: precisión, soporte multilingüe, capacidad de procesamiento en tiempo real, facilidad de integración con arquitecturas existentes, y disponibilidad de modelos preentrenados.

4.1. Herramientas y modelos investigados

Para el sistema de reconocimiento de emociones se investigaron diversas herramientas y librerías especializadas en procesamiento de lenguaje natural (PLN). A continuación se describen las principales opciones evaluadas:

Librería PySentimiento

PySentimiento es una biblioteca de Python multilingüe y de código abierto diseñada para la minería de opiniones y tareas de procesamiento de lenguaje natural. Permite realizar el análisis de sentimientos, emociones, discurso de odio e ironía de manera sencilla y accesible. Está construido sobre HuggingFace Transformers, lo que le permite aprovechar potentes modelos optimizados para diversos idiomas, incluidos español, inglés, italiano y portugués. Para la realización de tareas de procesamiento en español, PySentimiento utiliza RoBERTuito, un modelo basado en RoBERTa específicamente entrenado para español. Este modelo presenta un puntaje promedio en macro F1 de 68.9% en tareas de análisis emocional[69].

Modelo Monologg (BERT Ekman)

El modelo monologg/bert-base-cased-goemotions-ekman está basado en BERT y utiliza una de las tres taxonomías implementadas en GoEmotions PyTorch. Los modelos fueron entrenados con el conjunto de datos GoEmotions, un conjunto de datos desarrollado por Google que contiene 58.000 comentarios de Reddit etiquetados con 28 emociones.

Las etiquetas de esta taxonomía específica se basan en la teoría de Paul Ekman, que identifica seis emociones básicas universales: ira, asco, miedo, alegría, tristeza y sorpresa, además de la categoría neutral. El modelo de Ekman obtuvo una puntuación F1 en el conjunto de prueba de 62,38 % [70].

Modelo Daveni (XLM-RoBERTa)

Este modelo utiliza la arquitectura de XLM-RoBERTa-base, fue entrenado en 198 millones de tweets y ajustado específicamente para el análisis de emociones en el idioma español. Este modelo clasifica textos entre siete clases diferentes: ira, disgusto, miedo, alegría, tristeza, sorpresa y otros. El modelo logró la primera posición con un puntaje F1 macro-averaged de 71.70 % en el EmoEvalEs@IberLEF 2021, una competencia orientada a comprender las emociones expresadas por los usuarios en las redes sociales [71].

Librería Spanish-nlp

Spanish-nlp es una biblioteca de Python diseñada para facilitar las tareas de procesamiento de lenguaje natural en español. Esta biblioteca proporciona un conjunto de herramientas para el preprocesamiento de texto, la clasificación y el aumento de datos, lo que facilita a los investigadores y desarrolladores el trabajo con datos de texto en español. Cuenta con clasificadores como análisis de emociones y análisis de sentimientos [72].

4.2. Comparación de herramientas o modelos

La Tabla 4.1 resume las principales características de las herramientas y modelos investigados que pudieron ser evaluados exitosamente.

Herramienta / Modelo	Idioma	Ventajas	Desventajas
PySentimiento	Multilingüe (incluye español)	■ Fácil de implementar ■ Soporta detección de emociones, discurso de odio, ironía ■ Reconoce uso de emojis ■ Implementa el modelo RoBERTuito	En textos muy técnicos o especializados pueden presentar limitaciones en la detección emocional.
monologg/bert-base-cased-goemotions-ekman	Inglés	■ Entrenado con gran conjunto de datos (58K comentarios) ■ Basado en teoría de Ekman (6 emociones básicas) ■ Alta precisión en inglés	Al hacer la traducción al español se puede reducir significativamente la precisión y perder contexto cultural.
daveni/twitter-xlm-roberta-emoción-es	Español	■ Excelente precisión en emociones de redes sociales (F1: 71,70 %) ■ Entrenado en 198M tweets en español ■ Ganador IberLEF 2021	Menor precisión en textos formales o especializados debido a entrenamiento con lenguaje informal de Twitter.
Spanish-nlp	Español	■ Diseñado nativamente para tareas de procesamiento en español. ■ Integración de herramientas de clasificación y preprocesamiento.	Alta complejidad en la gestión de dependencias y conflictos de versiones con librerías modernas, lo que dificulta su ejecución estable.

Tabla 4.1: Comparación de características principales
Fuente: Elaboración propia

4.3. Metodología de Evaluación

Conjunto de datos utilizado

Para asegurar una comparación equitativa entre los modelos analizados, todos fueron evaluados utilizando el mismo conjunto de datos de prueba: `EmoSPEech_phase_1_test_codalab.csv` [73]. Este dataset fue utilizado en el reto EmoSPEech 2024, desarrollado como parte del taller IberLEF 2024, cuyo objetivo principal es investigar el campo del reconocimiento automático de emociones en habla [74].

Para ver el conjunto de datos con los resultados de las pruebas realizadas, consulte el anexo 1.

Características del conjunto de datos:

- **Tamaño:** 382 filas (muestras de prueba).
- **Atributos:**
 - **transcription:** Transcripción del texto (oración en español)

- **label:** Etiqueta emocional
- **Emociones clasificadas:** 6 categorías:
 - Neutral
 - Disgusto
 - Ira
 - Tristeza
 - Alegría
 - Miedo

Métricas de evaluación

Se utilizaron tres métricas estándar para evaluar el desempeño de los modelos en la tarea de clasificación emocional: Precisión, Recall y F1-Score.

Se evaluaron exitosamente tres alternativas:

- PySentimiento
- monologg/bert-base-cased-goemotions-ekman
- daveni/twitter-xlm-roberta-emoción-es

Spanish-nlp presentó problemas graves de incompatibilidad de dependencias que impidieron su uso. El error específico fue:

```
ValueError: numpy.dtype size changed, may indicate binary incompatibility.  
Expected 96 from C header, got 88 from PyObject
```

Este error no pudo ser resuelto a pesar de diversos intentos utilizando diferentes versiones de dependencias y soluciones documentadas en foros técnicos, por lo que se descartó del análisis comparativo.

Resultados

A continuación se presentan los resultados obtenidos por cada modelo en las tres métricas evaluadas.

Precisión: Representa la proporción de predicciones correctas entre todos los casos que fueron clasificados como positivos. Como se observa en la Figura 4.1, PySentimiento alcanzó la mayor precisión con 0.36, superando a daveni (0.34) y monologg (0.27) por 2 y 9 puntos porcentuales respectivamente.

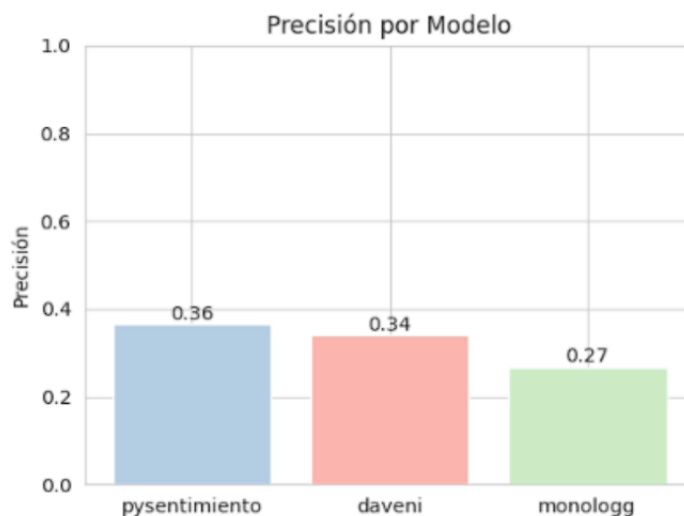


Figura 4.1: Comparación de precisión entre modelos evaluados

Fuente: Elaboración propia

F1-Score: Representa el balance entre precisión y recall. PySentimiento logró el mayor F1-Score con 0.39, seguido por daveni (0.37) y monologg (0.31), como se muestra en la Figura 4.2.

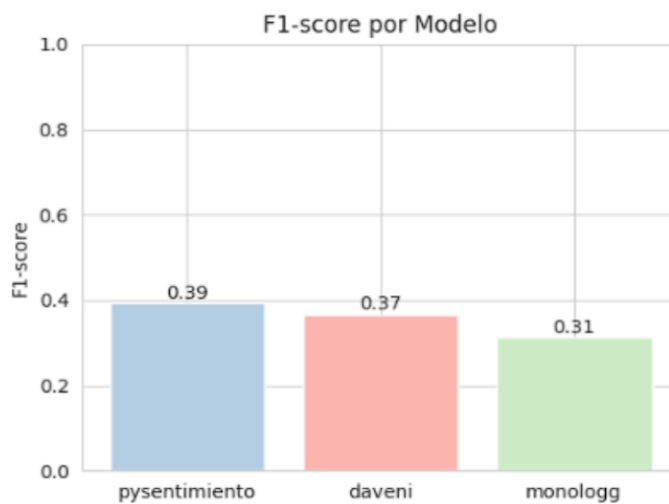


Figura 4.2: Comparación de f1-score entre modelos evaluados

Fuente: Elaboración propia

Recall: PySentimiento obtuvo el recall más alto con 0.47, superando a daveni (0.43) y monologg (0.41) por 4 y 6 puntos porcentuales respectivamente, como se observa en la Figura 4.3.

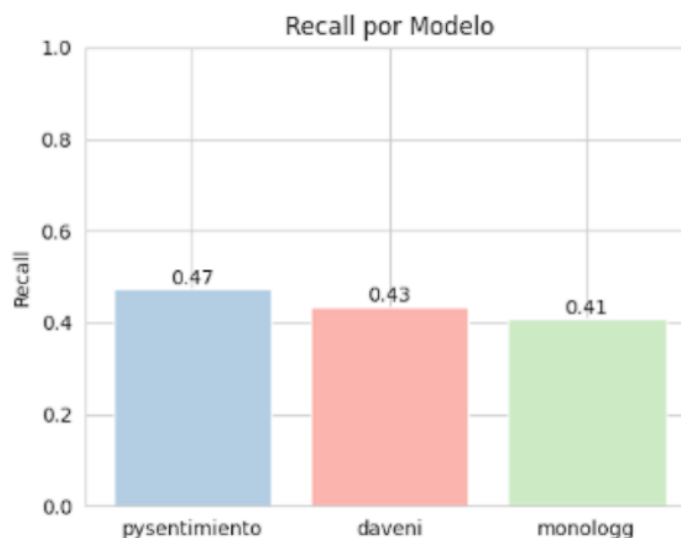


Figura 4.3: Comparación de recall entre modelos evaluados

Fuente: Elaboración propia

De acuerdo con los resultados obtenidos, **PySentimiento fue seleccionado como la solución definitiva** para el reconocimiento de emociones en la mascota virtual, alcanzando las métricas más altas en:

- **Precisión:** 0.36 (mejor por +2 y +9 puntos porcentuales)
- **F1-Score:** 0.39 (mejor por +2 y +8 puntos porcentuales)
- **Recall:** 0.47 (mejor por +4 y +6 puntos porcentuales)

Si bien estos valores no superan el umbral deseable de 0.5, su superioridad frente a las otras opciones evaluadas es estadísticamente significativa. La selección se fundamenta en:

Mejor desempeño relativo consistente: PySentimiento no solo alcanzó las métricas más altas, sino que lo hizo de manera consistente en las tres mediciones, demostrando robustez y equilibrio.

Recall superior (crítico para salud mental): En aplicaciones de apoyo emocional, el recall es la métrica más crítica. Un recall de 0.47 significa que el modelo detecta correctamente casi la mitad de los casos emocionales reales. Es preferible detectar una emoción negativa cuando existe (aunque ocasionalmente se equivoque) que no detectarla cuando el estudiante necesita apoyo.

Naturaleza desafiante del dataset: El bajo rendimiento absoluto de todos los modelos (ninguno superó 0.5) se atribuye a dos factores del dataset EmoSPeech:

- **Textos poco naturales:** El conjunto representa textos fragmentados, expresiones poco comunes y transcripciones de habla, lo que dificulta la extracción de emociones para todos los modelos.
- **Dataset desbalanceado:** La distribución desigual de clases emocionales afecta las métricas globales, ya que los modelos tienden a sesgarse hacia la clase mayoritaria.

A pesar de estas condiciones adversas, PySentimiento mostró mayor robustez, especialmente en recall (0.47), indicando mejor capacidad para identificar casos positivos incluso en un escenario desbalanceado.

Ventajas adicionales:

- Soporte nativo para español sin necesidad de traducción
- Procesamiento de emojis como características adicionales
- Facilidad de implementación mediante API de HuggingFace

Capítulo 5

Sistema de Generación de Respuestas Personalizadas

5.1. Construcción y Validación del Corpus

Una vez seleccionado el modelo para el reconocimiento emocional, el siguiente paso fue desarrollar el sistema de generación de respuestas. Para lograr esto, se optó por una arquitectura de Retrieval-Augmented Generation (RAG), la cual se nutre de un corpus de estrategias previamente validadas por expertos.

El objetivo de esta fase fue entrenar un chatbot, a través del uso de un corpus especializado y validado. Este capítulo detalla el proceso de construcción, validación de dicho corpus e implementación del RAG.

5.1.1. Recopilación y Diseño de Técnicas

Para dotar al sistema RAG de una base de conocimiento robusta y contextualmente relevante, se desarrolló un corpus exhaustivo de técnicas de bienestar emocional dirigido específicamente a poblaciones universitarias. La construcción del corpus se realizó mediante una revisión de información pertinente al entorno educativo [75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85], priorizando estrategias para manejar el estrés, la ansiedad, depresión, gestión del tiempo y otros problemas que afrontan los universitarios.

El proceso de recopilación contempló la búsqueda y síntesis de información relacionada con el contexto universitario. El corpus resultante se organizó en nueve dominios clave de intervención, estructurados según la naturaleza del problema y el tipo de estrategia requerida:

1. **Gestión del tiempo y productividad:** Incluye técnicas como establecimiento de horarios realistas, priorización de tareas, técnica Pomodoro y fragmentación de proyectos extensos.
2. **Autocuidado y hábitos saludables:** Abarca higiene del sueño, alimentación consciente, desintoxicación digital y práctica de gratitud.
3. **Técnicas de concentración:** Estrategias prácticas para eliminar distracciones y entrenar la capacidad de atención.
4. **Manejo preventivo del estrés:** Se centra en el desarrollo de actitud positiva, establecimiento de límites y comunicación asertiva.

5. **Técnicas de afrontamiento Durante el Estrés:** Incluye protocolos estructurados de respiración (4-7-8, respiración tranquila) y técnicas prácticas inmediatas como la “Descarga Mental”.
6. **Manejo de pensamientos negativos:** Estrategias basadas en terapias contextuales para flexibilizar pensamientos y conectar con valores personales.
7. **Estrategias específicas por emoción:** Contiene protocolos diferenciados para manejo de ansiedad ante exámenes, síntomas depresivos, soledad y frustración.
8. **Protocolo de seguridad:** Procedimiento claro para casos de crisis, con datos de contacto y ruta de atención del Consultorio Psicológico de la Universidad del Valle, sede Tuluá.
9. **Nota ética:** Declaración explícita sobre los límites de la mascota virtual, dirigiendo al usuario hacia ayuda profesional cuando sea necesario.

5.1.2. Validación por Expertos

El corpus desarrollado fue sometido a un proceso de validación con dos profesionales de la psicología con perfiles complementarios:

- **Psicólogo institucional:** Profesional en ejercicio dentro del departamento de bienestar universitario de la Universidad del Valle, sede Tuluá, con experiencia directa en la atención de estudiantes.
- **Psicóloga recién graduada:** Profesional con formación reciente y actualizada en enfoques terapéuticos contemporáneos.

La validación se realizó mediante un instrumento estructurado que evaluó:

- Pertinencia de las técnicas para el bienestar emocional de estudiantes universitarios
- Adecuación de los protocolos de derivación y crisis
- Claridad y seguridad de las intervenciones propuestas

Los resultados de la validación indicaron que las técnicas fueron consideradas apropiadas para la población universitaria, con sugerencias de ajustes menores para su optimización. En cuanto a los protocolos de seguridad, los evaluadores confirmaron su claridad y adecuación, resaltando la importancia de los mecanismos de derivación establecidos.

Las observaciones de los expertos permitieron refinar aspectos específicos del corpus, particularmente en la dosificación de técnicas y la especificidad de algunos protocolos. Ambos validadores autorizaron la inclusión de sus nombres como respaldo profesional del contenido.

Esta validación cruzada entre experiencia práctica institucional y conocimiento actualizado confirma que el corpus integra técnicas psicológicas válidas, seguras y éticamente responsables para su implementación en un sistema conversacional de apoyo emocional básico.

5.2. Implementación del sistema RAG

Tras la construcción y validación del corpus psicológico, se procedió a la implementación técnica del sistema encargado de procesar y recuperar esta información de manera dinámica. Este componente, denominado RAG (*Retrieval-Augmented Generation*), dota al chatbot con la capacidad de personalizar sus respuestas basándose en técnicas para el bienestar universitario.

A diferencia del reentrenamiento tradicional de modelos (*fine-tuning*), que es estático y computacionalmente costoso, el enfoque RAG permite instruir dinámicamente al modelo generativo [86]. Este sistema actúa como un mecanismo de contextualización especializada, cuya función es recuperar la estrategia precisa del corpus validado y suministrarla al modelo de lenguaje en tiempo real. De esta manera, se asegura que la interacción de la mascota no se base en el conocimiento general del modelo, sino que esté estrictamente alineada con los protocolos de bienestar emocional definidos en la fase anterior.

La implementación técnica para lograr esta “especialización en contexto” se estructuró en tres componentes: vectorización, almacenamiento y recuperación.

5.2.1. Base de datos vectorial

El almacenamiento del corpus requería una solución que permitiera búsquedas rápidas y filtrado por contexto. Para ello, se seleccionó **ChromaDB**, tras una evaluación técnica comparativa frente a **FAISS** (*Facebook AI Similarity Search*) [86].

Si bien **FAISS** es un estándar en la industria para búsquedas vectoriales a gran escala, la implementación de **ChromaDB** demostró ser superior para los objetivos específicos de este trabajo de grado por las siguientes razones:

- **Velocidad de respuesta en el prototipo:** Las pruebas empíricas revelaron una diferencia significativa en los tiempos de recuperación.

Para medir esto, se utilizó el conjunto de control de 20 preguntas (descrito en la sección de pruebas) ejecutado bajo las mismas condiciones para ambas bases de datos (mismo *prompt* y modelo base **paraphrase-multilingual-MiniLM-L12-v2**). Mientras que la configuración inicial con **FAISS** arrojó un promedio de 22 segundos, la implementación con **ChromaDB** logró estabilizar el promedio en 12.8 segundos. Esto representa una reducción de latencia cercana al 42 % atribuible únicamente al motor de base de datos, como se observa en la siguiente figura:

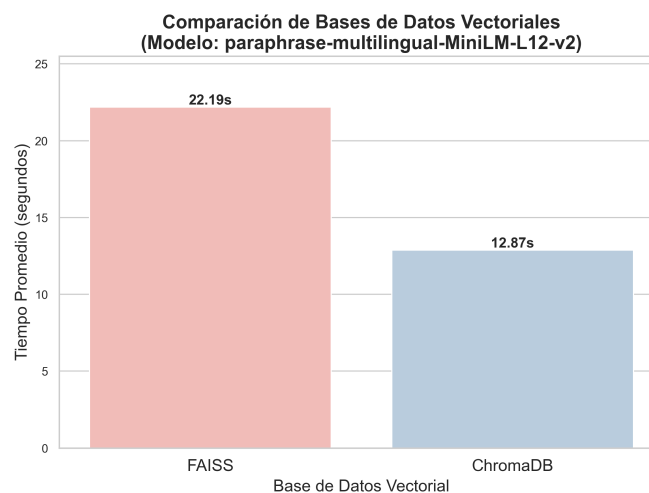


Figura 5.1: Comparación de FAISS vs ChromaDB

Fuente: Elaboración propia

- **Gestión de metadatos:** Aunque actualmente el sistema utiliza búsqueda por similitud semántica, **ChromaDB** ofrece capacidades nativas de filtrado por metadatos que permitirían, en futuras itera-

ciones, personalizar las respuestas según el tipo de técnica o estado emocional del usuario. Una funcionalidad que en FAISS hubiera requerido desarrollo adicional complejo.

- **Facilidad de despliegue:** La persistencia automática en disco mediante `PersistentClient` y la integración directa con el flujo de trabajo en Python permitieron centrar el esfuerzo en la calidad de la recuperación más que en la gestión de la infraestructura.

5.2.2. Modelo de embeddings

Para que el sistema pudiera interpretar semánticamente las consultas de los estudiantes y asociarlas con las técnicas correctas, se implementó el modelo `SentenceTransformer` con la arquitectura `all-MiniLM-L6-v2`.

La selección del modelo `all-MiniLM-L6-v2` se realizó mediante un proceso experimental comparativo. Inicialmente se evaluó el modelo `paraphrase-multilingual-MiniLM-L12-v2` por su capacidad multilingüe, pero las pruebas de rendimiento mostraron latencias superiores a 12 segundos. Al migrar a `all-MiniLM-L6-v2`, se logró reducir el tiempo de respuesta promedio a 4.5 segundos, validando que esta arquitectura ofrece el mejor equilibrio entre velocidad de inferencia y precisión semántica para el dominio de la aplicación.

Hiperparámetros de la Arquitectura: Aunque el modelo se implementa pre-entrenado (Transfer Learning), su eficiencia se debe a los siguientes hiperparámetros estructurales definidos durante su entrenamiento masivo con más de 1 billón de pares de oraciones (Contrastive Learning):

- **Longitud de Secuencia:** Limitada a 128 tokens durante el entrenamiento (con truncamiento a 256 en inferencia), optimizando el procesamiento de frases cortas típicas de un chat.
- **Optimizador Base:** Fue entrenado utilizando AdamW con una tasa de aprendizaje (learning rate) de $2e-5$ y un calentamiento (warm up) de 500 pasos, garantizando una convergencia estable que se refleja en la calidad de los vectores generados.
- **Infraestructura de Origen:** El modelo fue desarrollado sobre hardware TPU v3-8 con un tamaño de lote (batch size) de 1024, lo que le permite generar representaciones densas de alta calidad que el sistema actual aprovecha sin necesidad de reentrenamiento [87].

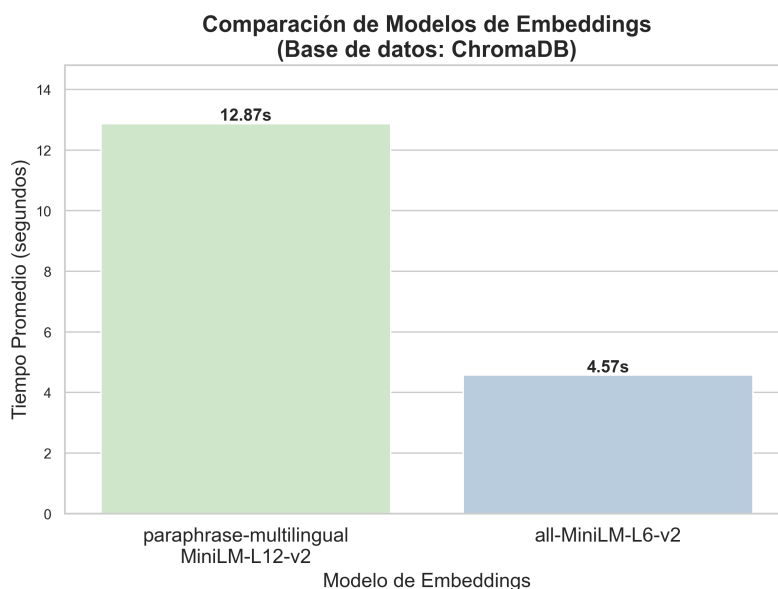


Figura 5.2: Tiempo de respuesta promedio: modelos de vectorización
Fuente: Elaboración propia

El modelo `all-MiniLM-L6-v2` mapea oraciones y párrafos a un espacio vectorial denso de 384 dimensiones. Esta dimensionalidad compacta (inferior a las 768 estándar) es la característica clave que permite ejecutar el sistema eficientemente en arquitecturas de solo CPU, eliminando la dependencia de hardware especializado (GPU) [87]. Aunque su entrenamiento base es en inglés, las pruebas de desarrollo confirmaron su eficacia para procesar y vincular conceptos de bienestar emocional en español.

5.2.3. Proceso de indexación de documentos

Previo a la interacción con el usuario, se ejecuta un procedimiento de indexación y vectorización del material bibliográfico. Este mecanismo de inicialización es el responsable de convertir los archivos PDF en representaciones vectoriales almacenadas en ChromaDB, tal como se ilustra en la figura 5.3.

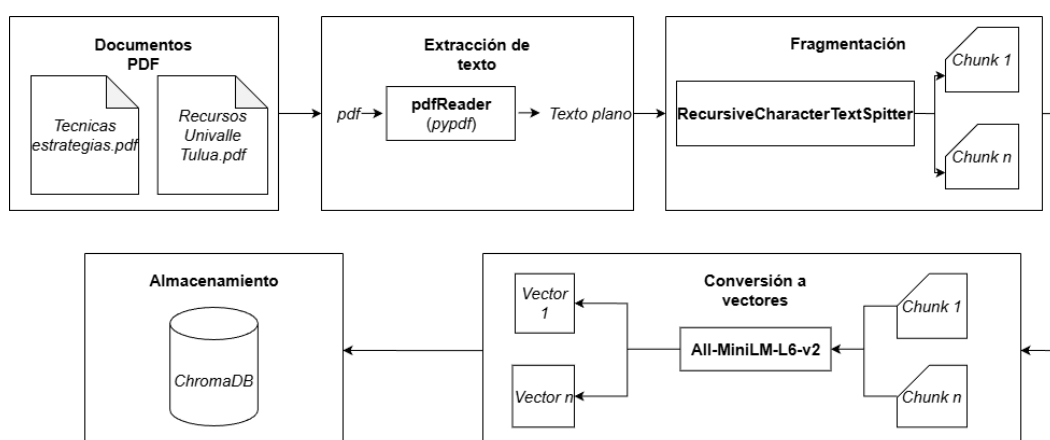


Figura 5.3: Proceso de indexación y vectorización de documentos PDF para almacenamiento en ChromaDB

Fuente: Elaboración propia

La arquitectura de ingesta se diseñó siguiendo un flujo secuencial de cuatro etapas técnicas:

1. **Ingesta y extracción de texto:** La primera etapa consiste en la carga de los documentos. Para ello, se implementó la función `pdfReader`, iterando página por página y concatenando el texto extraído.
2. **Fragmentación:** El texto extraído se divide en fragmentos manejables llamados “Chunks”. Utilizando `RecursiveCharacterTextSplitter`, cada chunk tiene un tamaño de 600 caracteres y un solapamiento (*overlap*) de 100 caracteres. Este mecanismo asegura la continuidad del contexto, evitando cortes abruptos en la información clave.
3. **Vectorización:** Cada fragmento se transforma en un vector utilizando el modelo `SentenceTransformer` (`all-MiniLM-L6-v2`). En esta fase, el lenguaje natural se convierte en representaciones matemáticas (vectores densos de 384 dimensiones) que capturan el significado del contenido. Formalmente, para un fragmento de texto t , se obtiene un vector $\mathbf{v} \in \mathbb{R}^{384}$:

$$\mathbf{v} = \text{SentenceTransformer}(t)$$

4. **Almacenamiento persistente:** Finalmente, los vectores generados, junto con sus metadatos asociados (como la fuente del documento y la ubicación del texto), son indexados y almacenados en una colección dentro de `ChromaDB`.

Este flujo secuencial garantiza una preparación eficiente del corpus para su posterior recuperación semántica durante las consultas de los usuarios.

El proceso mediante el cual el sistema RAG entrega la información ante una situación específica se ejecuta en tiempo real a través del flujo secuencial comprendido en la figura 5.4:

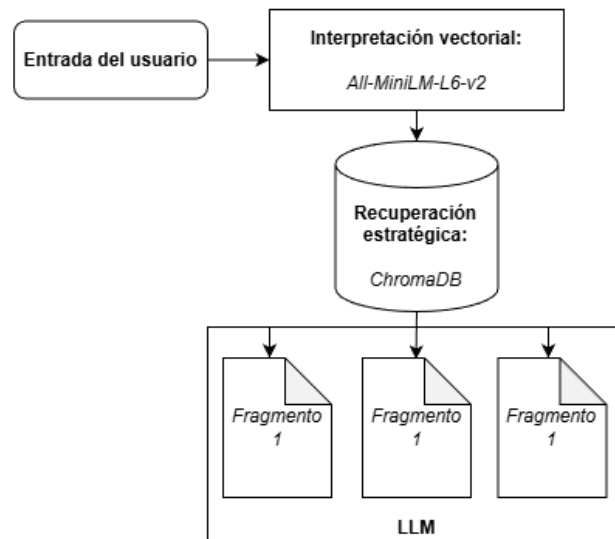


Figura 5.4: Flujo de recuperación de información en el sistema RAG

Fuente: Elaboración propia

1. **Interpretación Vectorial:** Al recibir la entrada del usuario, el sistema la transforma en un vector numérico utilizando el modelo `all-MiniLM-L6-v2`, capturando la esencia semántica de la necesidad emocional expresada. Formalmente, para una consulta del usuario q , se obtiene un vector $\mathbf{v}_q \in \mathbb{R}^{384}$:

$$\mathbf{v}_q = \text{SentenceTransformer}(q)$$

2. **Recuperación Estratégica:** El sistema consulta la base de datos `ChromaDB` mediante similitud coseno para identificar las estrategias del corpus que mejor se alinean con la situación del estudiante. Dados los vectores almacenados $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$ y el vector de consulta $\mathbf{v}_q$, se calcula:

$$\text{similitud}(\mathbf{v}_q, \mathbf{v}_i) = \frac{\mathbf{v}_q \cdot \mathbf{v}_i}{\|\mathbf{v}_q\| \|\mathbf{v}_i\|}$$

y se seleccionan los fragmentos con mayor similitud.

3. **Construcción del Contexto de Instrucción:** Se seleccionan los tres fragmentos más relevantes y se inyectan en las instrucciones del sistema (*system prompt*). Este paso es el equivalente funcional al entrenamiento, ya que condiciona la generación de texto del modelo, obligándolo a responder utilizando únicamente las técnicas de bienestar validadas y el tono empático definido para la mascota virtual.

Capítulo 6

Desarrollo del Prototipo de Mascota Virtual Móvil

6.1. Diseño de la Mascota Virtual

La mascota virtual funciona como interfaz principal del sistema, operando como mediador entre el estudiante y las capacidades de reconocimiento afectivo. Este componente incorpora una capa de interacción visual, con los siguientes objetivos:

- **Facilitar una experiencia** más natural para construir confianza y conexión emocional inmediata
- **Fortalecer el sentido de pertenencia** institucional mediante elementos de identidad

El componente visual está diseñado para responder a las seis categorías emocionales que determina el modelo de reconocimiento emocional: felicidad, tristeza, enojo, sorpresa, disgusto y neutral. Tras recibir la clasificación emocional, la mascota adapta dinámicamente sus expresiones faciales y animaciones, proporcionando retroalimentación visual coherente con las respuestas textuales generadas por el sistema.

6.1.1. Concepto Visual y justificación del diseño

La mascota fue conceptualizada como un personaje híbrido que combina características de la ardilla—símbolo institucional de la Universidad del Valle— con atributos felinos. Esta elección se fundamenta en dos pilares:

- **Identidad institucional:**
 - *Vínculo identitario:* La ardilla como mascota oficial establece una conexión inmediata con el contexto universitario
 - *Sentido de pertenencia:* Refuerza la identidad colectiva de los estudiantes de Univalle
 - *Reconocimiento visual:* Aprovecha un símbolo ya consolidado en la cultura institucional
- **Atributos emocionales complementarios:**
 - *Características felinas:* Asociadas con calma, compañía no invasiva y estabilidad, aspectos vinculados a la reducción del estrés y generación de sentimientos positivos [88]
 - *Características de ardilla:* Aportan energía, vitalidad y adaptabilidad contextual
 - *Equilibrio afectivo:* La combinación permite responder efectivamente a diversos estados emocionales

6.1.2. Proceso de Diseño y Desarrollo en Rive

La mascota virtual fue diseñada y animada completamente en Rive, una plataforma especializada en animaciones vectoriales interactivas para aplicaciones móviles [89]. La selección de Rive se fundamentó en tres criterios técnicos principales:

1. **Gráficos vectoriales escalables:** Permite crear animaciones de alta calidad que se adaptan a diferentes tamaños de pantalla sin pérdida de resolución, optimizando el rendimiento en dispositivos móviles con recursos limitados.
2. **Integración nativa con Flutter:** Rive ofrece un SDK oficial para Flutter que facilita la integración directa de animaciones en el código de la aplicación, reduciendo la complejidad técnica y mejorando el rendimiento.
3. **Sistema de máquina de estados:** Proporciona herramientas integradas para crear máquinas de estados finitos (FSM) que gestionan transiciones fluidas entre animaciones, permitiendo sincronización precisa con el modelo de reconocimiento emocional.

El flujo de trabajo estructurado implementado, contempló las siguientes etapas:

- **Diseño vectorial:** Creación de los elementos gráficos del personaje (cuerpo, rostro, extremidades, cola) utilizando las herramientas de dibujo vectorial nativas de Rive, definiendo la paleta de colores y el estilo visual.
- **Desarrollo de animaciones individuales:** Implementación de siete animaciones independientes (felicidad, tristeza, enojo, sorpresa, disgusto, respiración (neutral) y parpadeo), manteniendo coherencia en estilo, duración y naturalidad de movimientos.
- **Máquina de estados finitos (FSM):** Configuración de un sistema de estados que gestiona las transiciones entre emociones de manera fluida y predecible, evitando cambios abruptos o conflictos visuales.
- **Sistema de inputs programáticos:** Creación de controles booleanos para cada estado emocional, facilitando la integración con el modelo de reconocimiento emocional desde el código de la aplicación.

Esta arquitectura modular garantiza transiciones precisas entre diferentes expresiones emocionales y permite que el sistema responda en tiempo real a los cambios detectados en el estado emocional del usuario.

La máquina de estados finitos implementada en Rive (Figura 6.1) configura los siguientes elementos:

- **7 animaciones independientes:** Una para cada estado emocional (felicidad, tristeza, enojo, sorpresa, disgusto, respirar (neutral) y parpadeo)
- **Inputs booleanos:** Controles individuales para activar cada emoción (Joy, Sadness, Anger, Surprise, Disgust), que son manipulados programáticamente desde Flutter, según la salida del modelo de reconocimiento emocional
- **Transiciones automáticas:** Cambios fluidos entre estados mediante inputs booleanos programáticos (true/false) que activan cada estado emocional
- **Exclusividad mutua y gestión de capas:**
 - *Estados emocionales principales:* Son mutuamente excluyentes, solo uno puede estar activo a la vez

- *Estado neutral (respiración)*: Funciona como estado base con animación cíclica continua, que se reproduce en bucle durante períodos de inactividad emocional.
- *Parpadeo*: Opera como capa superpuesta independiente sobre el estado neutral, manteniéndose activo durante la respiración, para aportar naturalidad al personaje.
- *Activación de emoción*: Al detectarse una emoción específica mediante el modelo de análisis de sentimientos, el sistema ejecuta la siguiente secuencia:
 1. Detiene la respiración y el parpadeo
 2. Activa la animación emocional completa
 3. Mantiene la expresión durante aproximadamente 5 segundos
 4. Retorna automáticamente al estado de respiración neutral
 5. Reactiva el parpadeo superpuesto
- *Objetivo*: Garantizar claridad visual de cada emoción sin interferencias, mientras se mantiene naturalidad y "vida" del personaje durante inactividad.

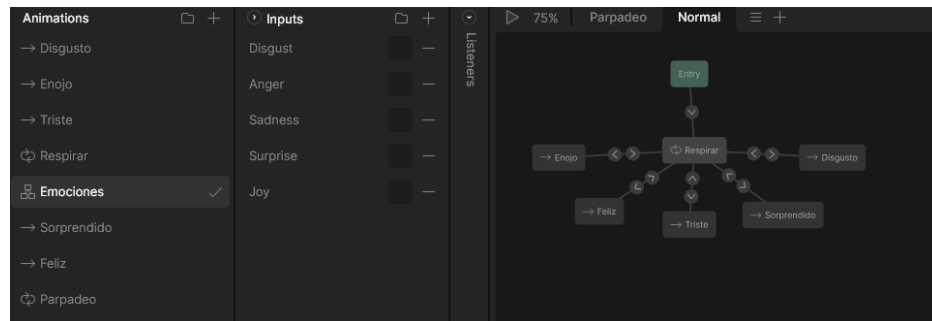


Figura 6.1: Arquitectura del diseño en Rive

Fuente: Elaboración propia

6.1.3. Representación Visual de los Estados Emocionales

Cada estado emocional fue diseñado con características visuales distintivas que permiten al usuario identificar rápidamente la "sintonía emocional" de la mascota con su propio estado afectivo. Esta retroalimentación visual inmediata refuerza la sensación de ser comprendido y acompañado emocionalmente, elemento clave en intervenciones de apoyo emocional digital.

La Figura 6.2 muestra las expresiones faciales y elementos visuales diseñados para cada estado emocional. A continuación se describen las características distintivas de cada representación:

Características Visuales por Estado

Neutral/Respiración

- **Expresión facial**: Relajada y acogedora, con ligera curvatura en la boca transmitiendo amabilidad.
- **Ojos**: Abiertos con mirada frontal amigable y accesible.
- **Postura**: Natural y equilibrada, con las manos frente al pecho.

- **Animación característica:** Movimiento suave y cíclico de respiración que se repite aproximadamente cada 4 segundos, durante el cual cada cuarto de segundo (15 fotogramas por segundo) se ejecuta el proceso de expansión-contracción del cuerpo simulando inhalación y exhalación. El parpadeo natural ocurre cada 4 segundos de manera sincronizada.
- **Función:** Mantener al personaje perceptivamente "vivo" durante períodos de inactividad o conversación sin carga emocional significativa.

Sorpresa (Surprised)

- **Expresión facial:** Boca en forma de media luna invertida, con ligera curvatura hacia abajo, transmitiendo asombro genuino.
- **Ojos:** Muy abiertos con pupilas ligeramente dilatadas, maximizando la expresión de sorpresa.

Tristeza (Sadness)

- **Expresión facial:** Boca en forma de "U" invertida con las comisuras hacia abajo, representando desánimo. **Postura:** Manos juntas frente al cuerpo en gesto contenido que transmite comprensión y disponibilidad para escuchar sin invadir el espacio emocional.
- **Consideración de diseño:** La expresión busca validar la emoción del usuario sin dramatizar excesivamente, manteniendo un tono empático pero esperanzador.

Felicidad (Joy)

- **Expresión facial:** Sonrisa amplia y genuina con boca completamente curvada hacia arriba, para enfatizar alegría.
- **Ojos:** Cerrados un poco por la intensidad de la sonrisa, transmitiendo felicidad auténtica.
- **Elementos adicionales:** Estrellas brillantes alrededor de la cabeza que refuerzan visualmente el estado de felicidad y logro, funcionando como indicadores gráficos de emoción positiva intensa.

Enojo (Anger)

- **Expresión facial:** Boca con ligera sonrisa contenida que busca transmitir comprensión más que enfado, evitando expresiones agresivas que podrían intensificar la emoción negativa del usuario.
- **Postura:** Manos juntas en gesto de contención y estabilidad emocional, transmitiendo solidez y apoyo.
- **Elemento simbólico:** Una hoja cayendo representa el proceso de dejar ir la tensión y volver a la calma, sugiriendo sutilmente la transitoriedad de la emoción.
- **Consideración de diseño:** La expresión fue diseñada para validar el enojo del usuario sin adoptar agresividad propia, manteniendo el rol de acompañamiento empático y orientando implícitamente hacia la regulación emocional.

Disgusto (Disgusted)

- **Expresión facial:** Boca ligeramente torcida hacia un lado.
- **Función:** Validar sentimientos de aversión o rechazo del usuario ante situaciones desagradables, manteniendo un tono empático.



Figura 6.2: Representación visual de la mascota en sus seis estados emocionales.

Fuente: Elaboración propia

El diseño de las emociones se orientó a transmitir empatía y validación, evitando reforzar los estados negativos. Para ello, se usaron metáforas visuales como estrellas (felicidad) y hoja cayendo (enojo), que acompañan al usuario sin intensificar la carga emocional.

6.2. Implementación del Chatbot con Análisis de Sentimientos

Este componente constituye el núcleo inteligente del sistema, integrando capacidades de procesamiento de lenguaje natural con mecanismos de análisis afectivo para generar respuestas contextualizadas y empáticas. El chatbot no solo procesa consultas textuales, sino que incorpora una capa de comprensión emocional que le permite adaptar su tono, contenido y estrategias de respuesta según el estado afectivo detectado del estudiante. La implementación combina un modelo de lenguaje avanzado (GPT-4o-mini) con un sistema de recuperación aumentada de generación (RAG) especializado en contenidos de bienestar emocional, creando así un amigo virtual capaz de proporcionar apoyo emocional fundamentado en técnicas para fortalecer el estado emocional de los estudiantes universitarios.

6.2.1. Arquitectura del Sistema Chatbot

El chatbot opera mediante una arquitectura cliente-servidor donde el frontend en Flutter captura las interacciones del usuario, el backend en FastAPI procesa el análisis emocional y la generación de respuestas, y el modelo GPT-4o-mini actúa como núcleo de inteligencia artificial. La comunicación entre componentes sigue un patrón RESTful donde el cliente realiza peticiones HTTP al servidor

(FastAPI) el cual retorna respuestas en formato JSON, que incluyen tanto la respuesta conversacional como la clasificación emocional para la sincronización con la mascota virtual.

Para ilustrar estos componentes y el flujo de datos dentro del sistema, se presenta la figura 6.3, que detalla la organización de la arquitectura antes mencionada y su integración con los servicios externos.

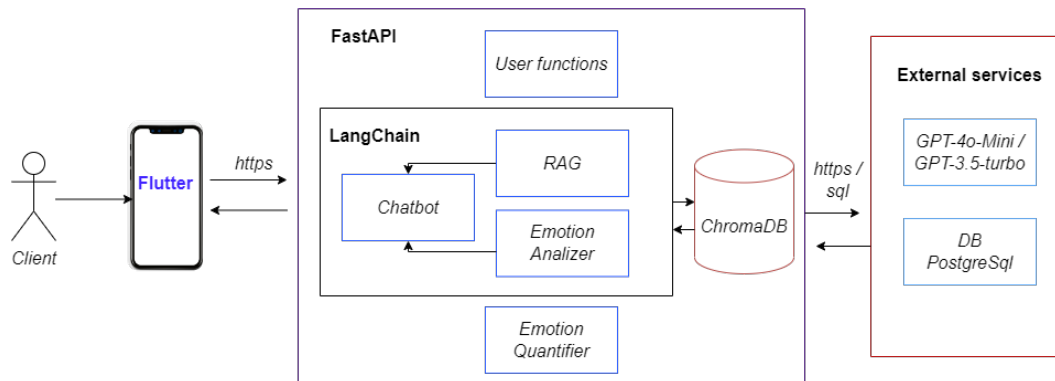


Figura 6.3: Diagrama de componentes e interacción Cliente-Servidor.

Fuente: Elaboración propia

6.2.2. Desarrollo del backend (FastAPI)

Para la construcción de la mascota virtual en el backend se utilizó **FastAPI**, un framework web utilizado para el desarrollo de APIs RESTful en Python. Permite la creación de APIs de manera rápida y sencilla, con un alto rendimiento en tiempos de procesamiento de peticiones. Además, genera de forma automática documentación en métodos implementados [90]. Algunas de las características por las que este framework se destaca para la realización de este trabajo de grado son:

- Facilidad de desarrollo
- Alto rendimiento, a la par con NodeJS y Go
- RESTful en Python, lo que lo hace útil para la implementación de modelos de IA

El diseño del sistema se estructuró siguiendo una arquitectura por capas inspirada en el patrón MVC (Modelo-Vista-Controlador), donde las rutas actúan como punto de entrada de las peticiones HTTP, los controladores gestionan la validación y orquestación de la lógica, y los servicios encapsulan las reglas de negocio. Al ser una API REST que sirve datos a una aplicación móvil Flutter, las respuestas son entregadas en formato JSON en lugar de vistas renderizadas. Esta separación se complementa con un módulo dedicado al sistema RAG (Sección 5.2), que opera de forma independiente, permitiendo que el chatbot fundamente sus respuestas en documentos especializados.

La estructura del backend está compuesta por los componentes descritos en adelante:

Capa de Entrada

El punto de entrada de la aplicación configura **FastAPI** y establece los componentes globales de la aplicación; para su visualización se presenta la relación de componentes en la Tabla 6.1:

Componente	Función
Global Exception Handler	Captura errores no manejados en cualquier endpoint, registra el error en logs y retorna una respuesta JSON con código 500, evitando que la aplicación se detenga.
Validation Exception Handler	Intercepta errores de validación de Pydantic, proporcionando mensajes personalizados al usuario (ej: “El email no es válido”).
Static Files	Sirve archivos estáticos (CSS) utilizados en la página de restablecimiento de contraseña.
Logging	Sistema de registro que almacena información de errores y eventos para depuración.

Tabla 6.1: Componentes de la Capa de Entrada
Fuente: Elaboración propia

Capa de Rutas

Define los endpoints HTTP disponibles y los agrupa por dominio funcional. A continuación, se detallan los servicios expuestos para la gestión de usuarios:

Endpoint	Método	Descripción
/users/register	POST	Registro de nuevos usuarios con nombre, correo, contraseña y nombre de mascota.
/users/login	POST	Autenticación de usuario y generación de token JWT con vigencia de 7 días.
/users/forgot-password	POST	Envío de correo electrónico con enlace para restablecer contraseña.
/users/reset-password	GET	Sirve la página HTML del formulario para ingresar nueva contraseña.
/users/reset-password	PUT	Confirma el cambio de contraseña utilizando el token recibido por correo.
/users/delete-account	DELETE	Elimina la cuenta del usuario y todos sus datos asociados (historial, perfil).

Tabla 6.2: Endpoints del Módulo de Usuarios
Fuente: Elaboración propia

De manera complementaria, el archivo `chatbot_route.py` gestiona la lógica de interacción con el

asistente virtual. Este módulo expone los puntos de acceso para el envío de mensajes, la administración del historial de conversaciones y la consulta del análisis emocional del usuario. Los detalles de estos endpoints se describen en la Tabla 6.3.

Endpoint	Método	Descripción
/chat	POST	Envío de mensaje al chatbot y recepción de respuesta con emoción detectada.
/chat/history	GET	Obtención del historial de conversación del usuario autenticado.
/chat/clear	DELETE	Vaciado del historial de chat conservando datos emocionales para estadísticas.
/chat/emotion-status	GET	Estado emocional actual del usuario representado como semáforo (verde/amarillo/rojo).
/chat/emotion-weekly-status	GET	Resumen de emociones agrupadas por día de la semana actual.

Tabla 6.3: Endpoints del Módulo de Chatbot
Fuente: Elaboración propia

Capa de Controladores

La capa de controladores contiene la lógica de orquestación, encargándose de validar las peticiones recibidas, invocar los servicios correspondientes y estructurar las respuestas.

`user_controller.py`

Este controlador gestiona el ciclo de vida de la cuenta del usuario y la autenticación. Las funciones principales se detallan en la Tabla 6.4.

Función	Responsabilidad
<code>register()</code>	Verifica si el email ya existe en la base de datos. Si no existe, crea el usuario con contraseña hashada y retorna el ID del usuario creado.
<code>login()</code>	Autentica credenciales (email/contraseña), genera token JWT con vigencia de 7 días y retorna datos del usuario (id, nombre, email, nombre de mascota, fecha de creación).
<code>forgot_password()</code>	Verifica que el usuario exista, genera token de restablecimiento y envía correo electrónico con enlace al formulario de cambio de contraseña.
<code>serve_reset_password_page()</code>	Sirve la plantilla HTML del formulario de restablecimiento de contraseña, pasando el token y URL del backend.
<code>reset_password_confirm()</code>	Verifica validez del token de restablecimiento, busca el usuario por email y actualiza la contraseña con el nuevo valor hashado.
<code>delete_account()</code>	Valida token de acceso, elimina en cascada: historial de chat → perfil de usuario → usuario, y confirma la eliminación.

Tabla 6.4: Funciones del Controlador de Usuarios

Fuente: Elaboración propia

chatbot_controller.py

Por otro lado, la gestión de las peticiones referentes a la interacción con la inteligencia artificial y el análisis emocional es manejada por este controlador, cuyas funciones se describen en la Tabla 6.5.

Función	Responsabilidad
<code>chat()</code>	Valida token JWT, obtiene usuario y sus últimas 5 interacciones como memoria, invoca el servicio de chatbot, guarda la nueva interacción (pregunta, respuesta, emoción) y retorna la respuesta con la emoción detectada.
<code>get_chat_history()</code>	Valida token JWT, obtiene historial completo ordenado por fecha ascendente. Si está vacío, genera y guarda mensaje de bienvenida personalizado con el nombre de la mascota.
<code>get_emotion_status()</code>	Valida token JWT, obtiene las últimas 10 interacciones con emoción y calcula el estado semáforo (rojo: alerta, amarillo: precaución, verde: estable).
<code>get_weekly_emotion_levels()</code>	Valida token JWT, recibe rango de fechas como parámetros, filtra historial en ese período y calcula nivel emocional por día traduciendo nombres al español.
<code>clear_chat_history()</code>	Valida token JWT, vacía el contenido de <i>question</i> y <i>answer</i> en todos los registros del usuario, pero conserva <i>emotion</i> y <i>timestamp</i> para estadísticas.

Tabla 6.5: Funciones del Controlador de Chatbot

Fuente: Elaboración propia

Capa de Servicios

La capa de servicios es el núcleo de la lógica de negocio del sistema, reuniendo las operaciones más complejas de manera independiente del protocolo HTTP. Esta separación permite que los servicios sean testeables y reutilizables. Para esta capa, en adelante se detallan los tres servicios principales que sustentan la funcionalidad del chatbot, el análisis del estado emocional actual y el seguimiento emocional semanal.

Servicio de chatbot (`response_chatbot()`)

Este servicio es el componente principal del sistema conversacional implementado, orquestando dos modelos de inteligencia artificial y fuentes de datos extraídos del usuario para generar respuestas personalizadas y contextualmente relevantes. La función recibe como parámetros:

- El mensaje del usuario.
- El historial de conversación (memoria).
- El identificador del usuario.
- La sesión de la base de datos.
- El nombre de la mascota virtual.

Para la orquestación del flujo conversacional se implementó **LangGraph**, una librería open-source de LangChain diseñada para construir flujos de trabajo y agentes basados en grafos de estado

[91]. Específicamente, se utiliza **StateGraph** para definir un nodo principal (**Chatbot_node**), el cual procesa el historial de mensajes y genera las respuestas. Esta arquitectura favorece la modularización, lo que a su vez, permite la adición de nodos de procesamiento futuros sin alterar la estructura base.

Complementariamente, se hace uso del ecosistema de **LangChain** para la manipulación y estructuración de la interacción con los modelos de lenguaje (LLMs). Se emplean las abstracciones **HumanMessage** y **AIMessage** para normalizar el historial de chat independientemente del proveedor del modelo. Además, se utilizan cadenas de procesamiento (LangChain Expression Language) que integran los *Prompt Templates* (**CHATBOT_PROMPT** y **EXTRACTION_PROMPT**) con *parsers* de salida específicos: **JsonOutputParser** para garantizar que la extracción de datos del usuario se entregue en formato JSON estructurado, y **StrOutputParser** para limpiar y procesar la respuesta final de texto generada por el chatbot.

El flujo de trabajo inicia con el análisis de la emoción implícita en el mensaje del usuario mediante la función interna **create_analyzer()**, configurada con los parámetros de tarea ‘emotion’ e idioma ‘español’.

Posteriormente, se ejecuta el modelo **gpt-3.5-turbo** con un prompt especializado en extracción de atributos clave del usuario (como estudios, hobbies, entre otros). Esta información se almacena en la base de datos (**UserProfile**) bajo un esquema clave-valor para enriquecer el contexto del modelo principal.

Finalmente, se recupera el contexto del sistema RAG a partir del mensaje del usuario y se invoca el grafo inyectando todos los datos estructurados en el prompt de la mascota virtual:

- **messages**: Historial de la conversación.
- **input**: Entrada actual del usuario.
- **emotion**: Emoción predicha.
- **profile**: Perfil del usuario recuperado.
- **chroma_context**: Contexto obtenido de la base de datos vectorial.
- **pet_name**: Nombre de la mascota.

Funcionalidades de bienestar emocional

Como complemento al chatbot, se incorporaron dos funciones estadísticas que permiten visualizar el estado emocional del usuario a lo largo del tiempo.

Estado emocional actual (**calculate_emotional_status()**)

Esta función evalúa el estado emocional reciente del usuario mediante un sistema de semáforo basado en el porcentaje de emociones negativas detectadas en sus últimas interacciones.

Se define el conjunto de emociones negativas E_{neg} :

- anger
- disgust
- fear
- sadness

La decisión de agrupar las emociones discretas propuestas anteriormente bajo un único indicador de ”**carga de afecto negativo**” se fundamenta en el principio de **Valencia emocional** [92]. Aunque estas emociones poseen características cualitativas distintas, todas comparten una valencia negativa que indica un estado de malestar psicológico.

El objetivo del algoritmo es identificar estados de riesgo, no realizar un diagnóstico clínico diferencial. Al agrupar estas emociones, se incrementa la sensibilidad del sistema, asegurando que cualquier manifestación de malestar contribuya al umbral de alerta.

Estas emociones deben ser transformadas a valores numéricos para poder entregar un umbral que dependa de la cantidad de emociones negativas que hay en las interacciones del usuario.

A partir del historial de las últimas n interacciones del usuario, se cuenta la cantidad de emociones pertenecientes a este conjunto $count_{neg}$ y se calcula el porcentaje de emociones negativas mediante la fórmula:

$$P_{neg} = \left(\frac{count_{neg}}{n} \right) \times 100 \quad (6.1)$$

El estado del semáforo se asigna según los umbrales descritos en la Tabla 6.6:

Estado	Condición	Interpretación
Verde	$P_{neg} \leq 40\%$	Estado emocional estable
Amarillo	$P_{neg} < 70\%$	Precaución, estado cambiante
Rojo	$P_{neg} \geq 70\%$	Alerta, considerar la derivación a recursos de apoyo

Tabla 6.6: Umbrales del Semáforo Emocional

Fuente: Elaboración propia

La selección del 70 % como punto de corte para el estado de ’Alerta’ obedece a un criterio de dominancia estadística. Un valor superior a dos tercios (66 %) indica que las emociones negativas no son eventos aislados, sino que constituyen la mayoría cualificada de las interacciones recientes.

Este umbral elevado se diseñó intencionalmente para garantizar la especificidad del sistema. Si el umbral fuera menor (ej. 51 %), el sistema generaría alertas ante cualquier malestar pasajero, provocando ’fatiga de alertas’ en el usuario. El 70 % actúa como un filtro de seguridad, asegurando que la recomendación de buscar apoyo profesional solo se active ante un patrón de malestar persistente y predominante, minimizando los falsos positivos.

Además, se establece un umbral mínimo de 8 interacciones para garantizar una buena interpretación. Si el historial contiene menos registros, el sistema retorna estado verde con un detalle indicando datos insuficientes.

Nivel emocional semanal (`calculate_weekly_emotional_levels()`)

Esta función proporciona un análisis extenso a lo largo de los días del estado emocional del usuario, calculando un nivel emocional diario dentro de un rango de fechas especificadas. Para esto, se asignan pesos emocionales para cada emoción. Estos valores no son arbitrarios, sino que se fundamentan en teoría psicológica: el modelo circuplejo de Russell [92] que ubica las emociones según valencia (positiva o negativa) e intensidad, y escalas como PANA [93] que asignan puntajes numéricos a emociones. De esta manera, se puede cuantificar y promediar las emociones de un usuario, generando

un indicador diario de bienestar emocional que refleja la tendencia positiva o negativa de su estado afectivo, como se muestra en la Tabla 6.7:

Emoción	Peso
Joy	+2
Surprise	+1
Others	0
Anger	-1
Sadness	-2
Fear	-2
Disgust	-2

Tabla 6.7: Pesos Asignados a las Emociones
Fuente: Elaboración propia

Se mapea cada una de las emociones de un día especificado, y se calcula el promedio diario. Para cada día d con k interacciones registradas, se calcula el nivel emocional promedio mediante:

$$E_d = \frac{1}{k} \sum_{i=1}^k w_i \quad (6.2)$$

Donde w_i representa el peso asignado a la emoción de cada interacción i . El promedio obtenido se clasifica según la siguiente escala, donde E_d es el promedio de la emoción diaria:

Nivel	Rango del promedio
Muy negativo	$E_d \leq -1,5$
Negativo	$-1,5 < E_d \leq -0,5$
Neutral	$-0,5 < E_d \leq 0,5$
Positivo	$0,5 < E_d \leq 1,5$
Muy positivo	$E_d \geq 1,5$

Tabla 6.8: Escala de Niveles Emocionales
Fuente: Elaboración propia

La definición de los rangos se basa en el criterio de equidistancia aritmética entre los pesos enteros asignados $(-2, -1, 0, 1, 2)$. Los límites se establecen en los puntos medios $(0,5$ y $1,5)$ para delimitar la pertenencia a cada categoría:

Categorías Centrales (Amplitud = 1.0): Los estados *Neutral*, *Positivo* y *Negativo* poseen un intervalo de amplitud de 1 unidad. Esto permite que el promedio oscile alrededor del valor entero central (por ejemplo, el estado *Neutral* abarca desde $-0,5$ hasta $+0,5$), capturando la variabilidad normal de las interacciones sin cambiar de categoría bruscamente.

Categorías Extremas (Amplitud Efectiva = 0.5): Los estados *Muy Positivo* y *Muy Negativo* poseen un intervalo de selección más estricto de $0,5$ unidades (de $1,5$ a $2,0$). Al ser los límites máximos de la escala, se requiere que el promedio supere el punto medio $(1,5)$ para ser clasificado como extremo, exigiendo una dominancia casi total de emociones fuertes para activar esta etiqueta.

Esta información permite a la aplicación móvil visualizar gráficamente el estado emocional a lo largo de la semana, facilitando la identificación de patrones o avances en su estado emocional.

Las dos funcionalidades anteriores facilitan la conciencia inmediata de estados internos, lo que se relaciona con el proceso de aceptación de la atención plena (mindfulness) descrito en la Terapia de Aceptación y Compromiso (ACT). En este enfoque, se guía a los pacientes a centrarse en el momento presente y aceptar aspectos de la vida que no pueden cambiarse, reduciendo la evitación emocional y aliviando síntomas depresivos [94].

Capa de Modelos

La capa de modelos define la estructura de datos persistidos mediante **SQLAlchemy ORM**, proporcionando una abstracción orientada a objetos sobre la base de datos relacional PostgreSQL.

El sistema utiliza tres entidades que modelan la información del usuario, su historial conversacional y el perfil personalizado extraído durante las interacciones con el chatbot. La Figura 6.4 presenta el diagrama entidad-relación que ilustra la estructura de la base de datos.

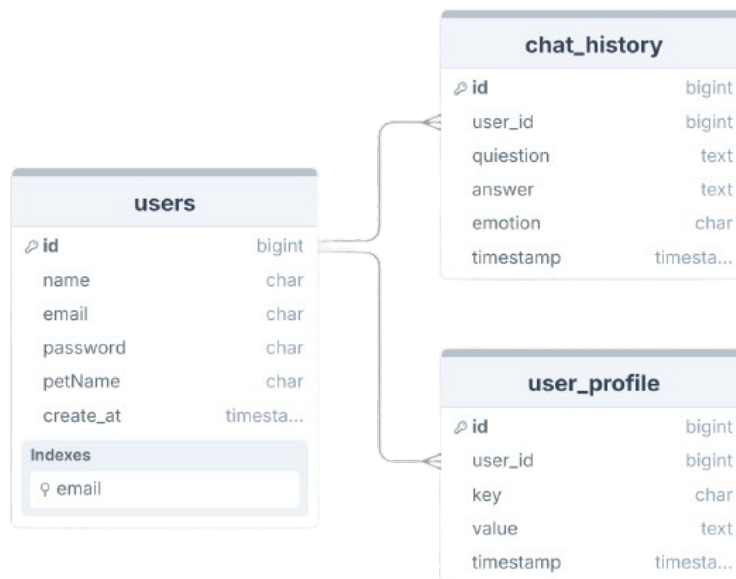


Figura 6.4: Diagrama Entidad-Relación de la Base de Datos

Fuente: Elaboración propia

Como se observa en el esquema, la entidad **users** actúa como la tabla central de la base de datos, albergando los datos de autenticación y la identidad de la mascota virtual. A partir de este nodo central se proyectan dos asociaciones de uno a muchos (1 : N) fundamentales para la lógica del sistema.

Por un lado, la relación con **chat_history** permite consolidar el registro de interacciones; aquí no solo se guardan el mensaje de usuario y la respuesta, sino también la emoción inferida por Pysentimiento y la marca temporal. Esta trazabilidad es clave, pues facilita tanto la reconstrucción del hilo conversacional como el análisis longitudinal del estado anímico del estudiante.

Por otro lado, la vinculación con **user_profile** responde a la necesidad de flexibilidad. Al adoptar un esquema de tipo clave-valor, el sistema puede almacenar atributos heterogéneos extraídos

dinámicamente mediante un modelo LLM, sin la necesidad de modificar la estructura de la base de datos, lo que garantiza una personalización progresiva y escalable.

6.2.3. Despliegue e infraestructura del backend

Para garantizar la disponibilidad y escalabilidad de la lógica de negocio, se utilizó DigitalOcean como proveedor de servicios en la nube. La implementación se realizó sobre la solución *App Platform*, lo que permite una gestión abstracta del servidor y un enfoque centrado en el rendimiento del contenedor. El proceso de despliegue técnico siguió los siguientes pasos:

Contenerización y Registro: La aplicación *pet-virtual-app* fue empaquetada mediante Docker. La imagen resultante se alojó en el *DigitalOcean Container Registry*. Se configuró el uso de la etiqueta *latest*, garantizando que el servicio productivo siempre ejecute la versión actual del código fuente.

Aprovisionamiento de recursos: Para soportar la carga operativa de este prototipo se asignaron recursos de cómputo específicos: 1 VCPU compartida y 2 GB de RAM. Esta configuración fue seleccionada para mantener equilibrio en 2 *workers* durante las peticiones concurrentes y el procesamiento de datos.

Automatización del despliegue: Se habilitó la característica *Autodeploy*. El sistema monitorea constantemente el sistema de contenedores y automáticamente reemplaza las nuevas versiones sin intervención manual.

Persistencia de datos: La base de datos se implementó como un servicio gestionado (*Managed Database*), separándose del contenedor de la aplicación y gestionando copias de seguridad automáticas.

6.2.4. Integración del Cliente Móvil (Flutter)

La aplicación móvil constituye la interfaz principal de interacción entre el estudiante y el sistema de apoyo emocional. Desarrollada con Flutter y Dart, la aplicación integra tres componentes fundamentales: el sistema conversacional del chatbot, la visualización animada de la mascota virtual mediante Rive y una sección de bienestar emocional.

Arquitectura en Capas de la Aplicación

La aplicación implementa una organización modular que prioriza la practicidad y separación de responsabilidades. Se estructura en tres capas principales:

1. **Capa de Infraestructura:** Contiene los modelos de datos para comunicación con la API, como los modelos de autenticación *UserRegister* y *UserLogin* que encapsulan la información que se envía al backend.
2. **Capa de Configuración y Servicios:** Combina configuración estática (constantes, temas) con servicios de aplicación como la autenticación, comunicación con el chatbot y persistencia de datos.
3. **Capa de Presentación:** Gestiona la interfaz de usuario (screens y widgets) y el estado reactivo mediante el patrón Provider, renderizando componentes visuales y respondiendo a cambios de estado.

La Figura 6.5 ilustra la relación jerárquica entre capas, donde la presentación depende de servicios, y estos a su vez de la infraestructura de datos. Los componentes mencionados son ejemplos representativos de cada nivel, existiendo adicionalmente otros elementos específicos para funcionalidades particulares de la aplicación.

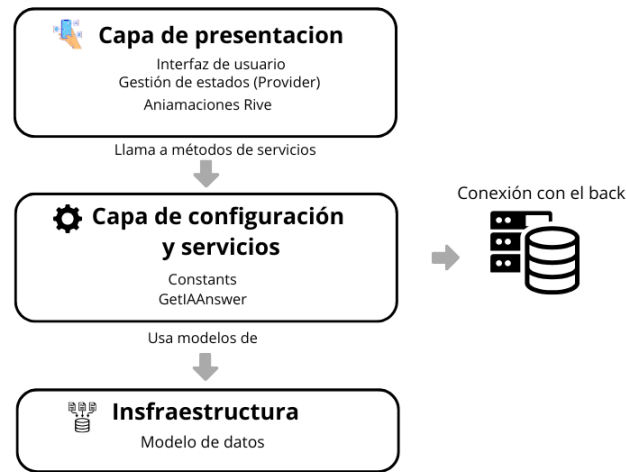


Figura 6.5: Arquitectura en capas de la aplicación Flutter.

Fuente: Elaboración propia

Selección Tecnológica: Flutter y Dart

Flutter es un framework de código abierto desarrollado por Google para la creación de aplicaciones nativas multiplataforma desde una única base de código [95]. Su adopción para este proyecto se fundamentó en los siguientes criterios técnicos:

- **Reactividad y Estado:** Implementación del patrón Provider como solución oficial para gestión de estado, permitiendo actualizaciones reactivas de la interfaz.
- **Integración con Rive:** Soporte nativo para renderizar animaciones vectoriales interactivas, utilizada para la mascota virtual.
- **Ecosistema robusto:** Disponibilidad de paquetes como `dio` para comunicación HTTP avanzada, `flutter_secure_storage` para persistencia local segura, y `provider` para gestión de estado.
- **Productividad en desarrollo:** Hot Reload para iteración rápida, widget tree declarativa, y sintaxis moderna de Dart que facilita el desarrollo y mantenimiento del código.

Dart como Lenguaje de Programación

Dart es un lenguaje de programación de código abierto optimizado para desarrollo de interfaces de usuario, diseñado específicamente por Google para el ecosistema Flutter [96]. Las características principales que beneficiaron este proyecto incluyen:

- **Sistema de tipos con null-safety:** Dart utiliza sistema de tipos con null safety, que maneja los valores nulos de forma segura.

- **Sistema de tipos estático:** Verificación en tiempo de compilación que garantiza la consistencia de datos, esencial para los modelos estructurados como `Message` y `UserLogin` utilizados en la comunicación con el backend.
- **Async/Await Nativo:** Soporte integrado para operaciones asíncronas mediante sintaxis clara, utilizado extensivamente en los servicios `AuthService` y `GetIAAnswer` para peticiones HTTP sin bloquear la interfaz de usuario.

La combinación de `dio` para comunicación HTTP y `rive` para animaciones permite un flujo integrado donde: (1) el chatbot detecta emociones, (2) `ChatProvider` actualiza el estado emocional, (3) Rive reproduce la animación correspondiente, y (4) la interfaz refleja estos cambios de forma reactiva mediante Provider.

Gestión de Estado con Provider

Para la gestión de estado se implementó el patrón Provider utilizando el paquete oficial `provider` de Flutter. El `ChatProvider`, que extiende `ChangeNotifier`, actúa como gestor central del estado del chat, notificando a los widgets suscritos mediante `notifyListeners()` cuando cambian datos como el historial de mensajes o la emoción actual.

Este componente especializado es responsable de coordinar la interfaz de usuario con la lógica de negocio y los servicios de backend, gestionando:

- **Historial completo de mensajes:** Lista ordenada cronológicamente de todos los mensajes intercambiados, almacenada en `messageList`.
- **Estado emocional actual:** Emoción detectada almacenada en `_currentEmotion`, utilizada para sincronizar la animación de la mascota mediante `setEmotion()`.
- **Indicadores de proceso:** Estados de carga (`_isLoading`), indicador de escritura (mensajes con `FromWho.typing`), y control de sesión expirada (`_sessionExpired`).
- **Control inteligente de navegación:** Gestión automática del scroll mediante `scrollToBottom()`, `ensureScrollOnKeyboard()`, y `handleViewInsetsChange()`, asegurando visibilidad del mensaje más reciente.
- **Comunicación con servicios:** Interacción con `GetIAAnswer` para envío/recepción de mensajes y gestión de emociones mediante `setEmotion()`, actuando como punto de coordinación central.

La implementación de Provider permite que múltiples widgets de la interfaz (lista de mensajes, campo de entrada, indicador de estado, mascota animada) reaccionen automáticamente a cambios de estado sin necesidad de callbacks manuales.

Integración de la Mascota Animada con Rive

La mascota virtual, diseñada en Rive (Sección 6.1), se integra en la aplicación Flutter como un widget reactivo que responde a los cambios emocionales detectados durante la conversación. El flujo de integración sigue un patrón unidireccional: el estado de la aplicación, gestionado por el `ChatProvider`, determina qué animación debe reproducir la mascota.

Flujo de Integración

1. El usuario envía un mensaje a través de la interfaz de chat.
2. El `ChatProvider` envía el mensaje al backend a través del servicio `GetIAAnswer`.
3. El modelo de IA en el backend clasifica la emoción del mensaje y devuelve una respuesta junto con una etiqueta emocional (ej: "joy").
4. El `ChatProvider` actualiza el estado emocional con `setEmotion()` y notifica a los widgets suscritos.
5. El widget `MascotaAnimation`, que está suscrito al `ChatProvider` mediante un `Consumer`, recibe la nueva emoción y activa la animación correspondiente en Rive.

El widget `MascotaAnimation` se encarga de gestionar la animación Rive. Carga el archivo `animacion.riv` y obtiene los controles programáticos (inputs booleanos) de la máquina de estados. Cuando la emoción cambia, el widget activa el input correspondiente (por ejemplo, `_joyInput.value = true`) y programa un temporizador para que, tras unos segundos, la mascota vuelva al estado neutral de respiración". Esto asegura una retroalimentación visual inmediata y coherente con la conversación.

Además, durante la generación de respuestas del chatbot, la mascota permanece en estado neutral independientemente de la emoción detectada, evitando transiciones visuales que puedan distraer al usuario mientras lee la respuesta. Esta integración permite que la mascota reaccione de forma dinámica y empática, reforzando la experiencia de acompañamiento emocional. La figura 6.6 ilustra el proceso:

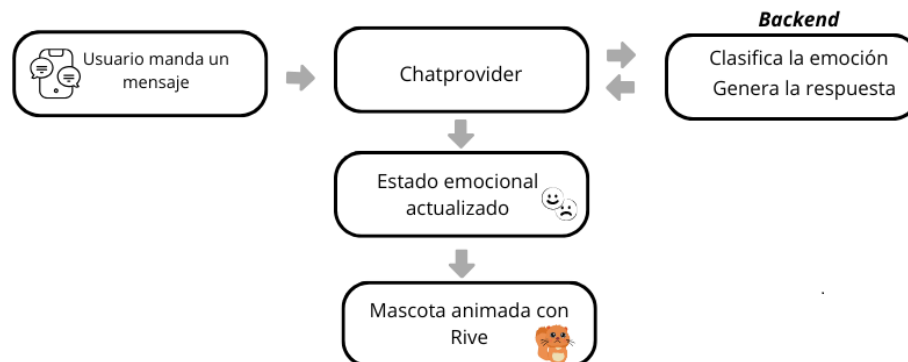


Figura 6.6: Flujo de integración de la mascota.

Fuente: Elaboración propia

6.2.5. Interfaz de Usuario y Experiencia Visual

El diseño de la interfaz siguió un proceso iterativo que comenzó con la creación de wireframes de baja fidelidad, lo que permitió validar la estructura y flujos de navegación antes de pasar al desarrollo visual. Estos prototipos iniciales, como los que se muestran en las Figuras 6.7 y 6.8, establecieron la disposición básica de los elementos y la secuencia de interacción, los cuales evolucionaron hacia la interfaz final que prioriza la simplicidad, accesibilidad y calidez emocional. El conjunto completo de wireframes se encuentra disponible en el Anexo 2.

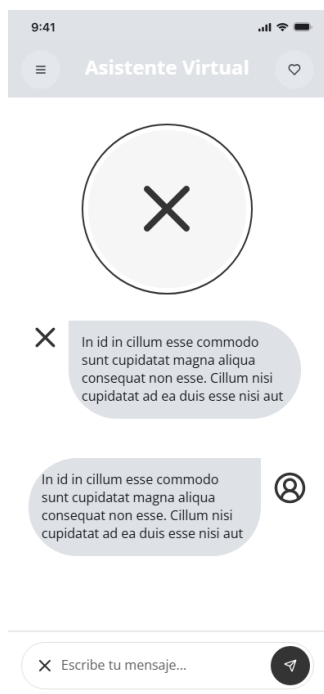


Figura 6.7: Wireframe inicial del módulo de chat



Figura 6.8: Wireframe inicial del módulo de bienestar

Fuente: Elaboración propia

La interfaz de la aplicación se diseñó con un enfoque centrado en el usuario, priorizando la simplicidad, accesibilidad y calidez emocional. Para ello se empleó una paleta de colores con un rojo claro que transmite cercanía, dado que es el color representativo de la Universidad del Valle. La navegación se estructuró en flujos lineales simples, minimizando la profundidad de menús para que el usuario pueda enfocarse en la interacción emocional sin distracciones. A continuación se presentan las interfaces principales de cada módulo.

Módulo de Autenticación

El módulo de autenticación gestiona el acceso seguro a la aplicación. La Figura 6.9 muestra la pantalla de inicio de sesión como interfaz inicial.

Esta pantalla presenta campos para correo electrónico y contraseña con validación en tiempo real, botón de acción principal destacado en el color institucional, y enlaces para registro de nuevos usuarios y recuperación de contraseña.

Chat Principal con Mascota Virtual

La pantalla de chat constituye el núcleo de la experiencia de usuario, donde se desarrolla la interacción conversacional y el reconocimiento emocional. La Figura 6.10 muestra la interfaz cuando el usuario ingresa por primera vez al chatbot, y la Figura 6.11 presenta una conversación activa.



Figura 6.9: Pantalla de inicio de sesión
Fuente: Elaboración propia



Figura 6.10: Interfaz de chat con saludo inicial de la mascota



Figura 6.11: Interfaz de chat durante conversación activa

Fuente: Elaboración propia

La interfaz integra tres elementos principales: la mascota animada en la parte superior que reacciona visualmente a las emociones detectadas, el área central de mensajes que diferencia visualmente los mensajes del usuario (alineados a la derecha) y las respuestas del chatbot (alineadas a la izquierda), y la barra inferior con campo de texto y botón de envío.

Módulo de Bienestar Emocional

Este módulo permite visualizar y reflexionar sobre el estado emocional mediante representaciones gráficas. La Figura 6.12 presenta el semáforo emocional y la gráfica de evolución semanal.



Figura 6.12: Pantalla de bienestar emocional con semáforo y gráfica semanal

Fuente: Elaboración propia

El semáforo emocional utiliza codificación de colores que resume el estado actual del usuario: verde indica estado positivo, amarillo representa estado moderado, y rojo señala estado emocional negativo. La Figura 6.13 muestra el semáforo en estado rojo, el cual activa automáticamente una alerta con información de contacto del servicio de psicología universitaria.

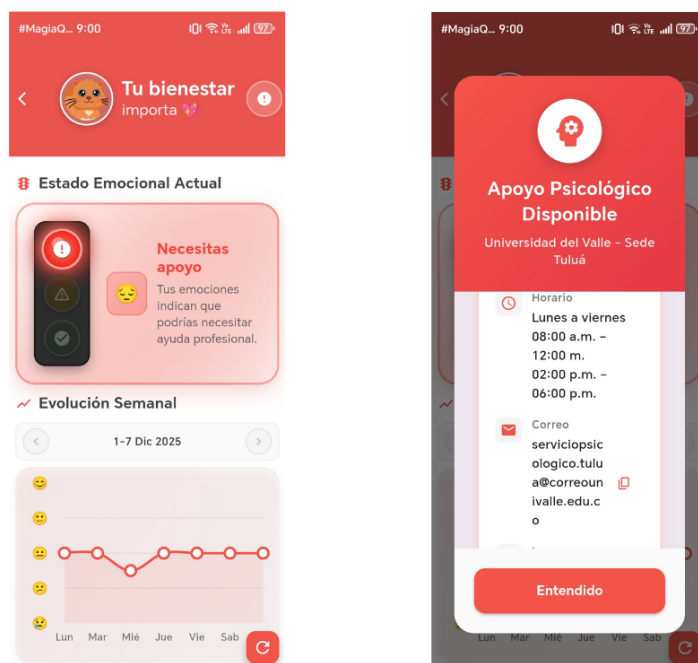


Figura 6.13: Semáforo emocional en rojo con alerta de apoyo psicológico
Fuente: Elaboración propia

Esta funcionalidad de alerta representa una medida de seguridad que conecta al usuario con recursos profesionales cuando el sistema detecta un patrón emocional preocupante sostenido en el tiempo.

6.2.6. Casos de Uso del Sistema

Los casos de uso describen las interacciones principales entre el usuario (estudiante) y el sistema de apoyo emocional. A continuación se presentan los cuatro flujos de uso más importantes.

Caso de Uso 1: Registro y Autenticación

Actor: Estudiante

Precondiciones: Usuario tiene la aplicación instalada y conexión a Internet.

Flujo principal (Registro de nuevo usuario):

1. Usuario abre la aplicación.
2. Sistema muestra pantalla de inicio de sesión.
3. Usuario selecciona “Registrarse”.
4. Usuario completa formulario con nombre, email, contraseña y nombre para la mascota.
5. Sistema valida los datos y crea la cuenta en el backend.

Flujo alternativo 1 (Inicio de sesión con credenciales existentes):

1. Usuario ingresa email y contraseña.

2. Sistema verifica las credenciales.
3. Si son correctas, redirige al chat.

Flujo alternativo 2 (Recuperación de contraseña):

1. Usuario selecciona "¿Olvidó su contraseña?".
2. Sistema solicita el email.
3. Usuario ingresa su email.
4. Sistema envía un correo con instrucciones para restablecer la contraseña.
5. Usuario sigue las instrucciones y establece una nueva contraseña.

Postcondición: Usuario autenticado accede a la pantalla principal de chat.

Caso de Uso 2: Interacción Conversacional con Detección Emocional

Actor: Estudiante

Precondiciones: Usuario ha iniciado sesión correctamente.

Flujo principal:

1. Usuario escribe un mensaje en el campo de texto.
2. Sistema envía el mensaje al backend para procesamiento.
3. Modelo de IA analiza el mensaje y clasifica la emoción.
4. Backend genera respuesta textual y devuelve etiqueta emocional.
5. Sistema actualiza la interfaz con la respuesta.
6. Mascota virtual reproduce animación correspondiente a la emoción detectada.
7. Tras 5 segundos, mascota regresa a estado neutral de respiración.

Postcondición: Historial de chat actualizado, estado emocional reflejado visualmente.

Caso de Uso 3: Monitoreo de Estado Emocional

Actor: Estudiante

Precondiciones: Usuario ha interactuado con el chatbot al menos una vez.

Flujo principal:

1. Usuario navega a la sección de bienestar emocional.
2. Sistema muestra el semáforo emocional (rojo/amarillo/verde).
3. Sistema muestra gráfica de evolución semanal de emociones.
4. Si el semáforo está en rojo, sistema muestra alerta con información de contacto psicológico.
5. Usuario puede revisar sus tendencias emocionales históricas.

Postcondición: Usuario comprende su estado emocional actual e histórico.

Caso de Uso 4: Gestión de Cuenta y Configuración

Actor: Estudiante

Precondiciones: Usuario ha iniciado sesión.

Flujo principal:

1. Usuario accede al menú de opciones desde el chat principal.
2. Sistema muestra opciones: cerrar sesión, eliminar cuenta, limpiar chat.
3. Usuario selecciona una opción.
4. Sistema solicita confirmación para acciones irreversibles.
5. Ejecuta la acción seleccionada y redirige según corresponda.

Flujo alternativo 1: Si selecciona "Limpiar chat", sistema borra historial local y en backend.

Flujo alternativo 2: Si selecciona "Cerrar sesión", sistema borra tokens y redirige a pantalla de login.

Flujo alternativo 3: Si selecciona "Eliminar cuenta", sistema elimina datos del usuario y redirige a pantalla de login.

Postcondición: Acción completada, interfaz actualizada.

En la Figura 6.14 se ve el diagrama de casos de uso que muestra la interacción del usuario con el sistema y los casos de uso posible.

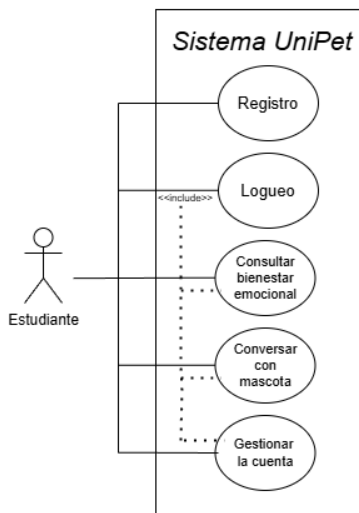


Figura 6.14: Diagrama de casos de usos
Fuente: Elaboración propia

6.2.7. Empaquetado y firma de la aplicación

Para el desarrollo en Flutter, la fase de generación y firma del ejecutable se centró en la construcción de un binario (APK), optimizado y seguro para el sistema Android. En primera instancia, se implementó el firmado digital mediante un archivo de claves (*keystore*), requisito indispensable para garantizar la integridad y autoría del paquete. Posteriormente, se ejecutó el comando **flutter**

`build apk --release` para generar un archivo APK optimizado y apto para su instalación en sistemas operativos Android.

Capítulo 7

Pruebas

7.1. Pruebas de software

7.1.1. Prueba de Rendimiento

Se realizó una prueba de rendimiento para evaluar el tiempo de respuesta del chatbot. Se utilizaron 20 preguntas independientes, no relacionadas entre sí, correspondientes a situaciones emocionales reales del contexto universitario, relacionadas con estrés, ansiedad, motivación, bienestar y conflictos académicos y sociales. Las preguntas presentaban longitudes variables, con el fin de simular mensajes reales de los usuarios. La prueba se ejecutó en un entorno local; se ejecutaron concurrentemente en el mismo dispositivo de desarrollo: un computador portátil Lenovo IdeaPad 3 (procesador Intel Pentium Gold Dual-Core a 2.4 GHz, 8 GB de memoria RAM DDR4, disco SSD de 512 GB y sistema operativo Windows 11 Home). Las peticiones se realizaron a través de la interfaz de red de bucle local (localhost), midiendo el tiempo de procesamiento puro del modelo y la base de datos vectorial sin latencia de internet, pero con la poca capacidad de procesamiento de un portátil de uso doméstico.

Cada pregunta se envía mediante una petición HTTP, utilizando un script en Python, de forma secuencial, midiendo el tiempo transcurrido desde el envío de la solicitud hasta la recepción de la respuesta. Se registró el tiempo de respuesta individual de cada consulta, permitiendo calcular métricas como tiempo promedio, mínimo y máximo, con el objetivo de evaluar la capacidad de respuesta y la estabilidad del chatbot en escenarios realistas.

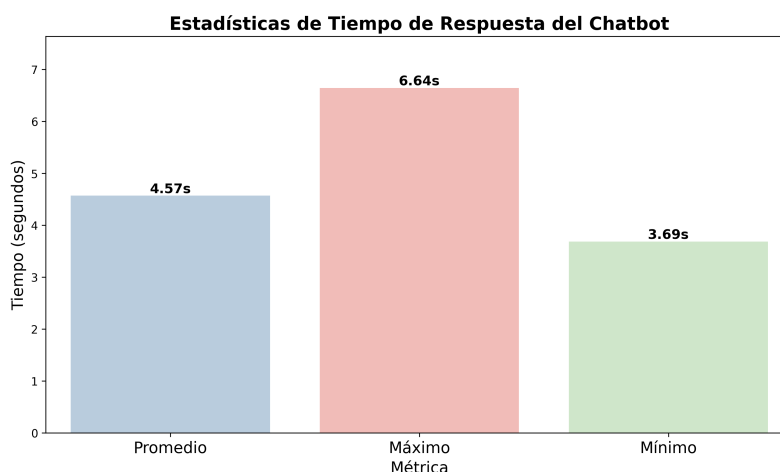


Figura 7.1: Estadísticas de tiempo de respuesta: Mínimo, promedio y Máximo
Fuente: Elaboración propia

Según Robert B. Miller en [97], los tiempos de respuesta en la interacción humano-computadora no tienen un único límite, y dependen del tipo de tarea, la complejidad de ella y del contexto. El autor señala que, aunque los retrasos mayores a dos segundos pueden afectar la continuidad cognitiva del usuario, en situaciones complejas los usuarios toleran hasta 4.6 segundos para percibir demora en peticiones complejas, que además indican el procesamiento de la información para generar una respuesta. En este sentido, el tiempo de respuesta promedio de este chatbot (**4.57 segundos**) se encuentra dentro del rango aceptable para mantener la validez de la conversación. Mostrando además que el 60 % de las peticiones fueron resueltas por debajo del promedio.

Los datos de prueba de rendimiento del chatbot están disponibles en el anexo 3.

7.1.2. Pruebas de LLM as a Judge

El *LLM-as-a-judge* es un método de evaluación de respuestas donde los grandes modelos de lenguaje sustituyen tanto a los evaluadores humanos como a métricas estadísticas estáticas. Los jueces basados en este sistema pueden realizar evaluaciones más detalladas y sensibles, ajustando dinámicamente los criterios según los requisitos específicos de la tarea. Más allá de la simple puntuación, este sistema proporciona una explicación en lenguaje natural que ofrece retroalimentación más rica e interpretable [98].

Para efectos de este sistema, se medirán dos puntos importantes para este trabajo de grado: la fundamentación y la relevancia. Para ello se utiliza un modelo más avanzado que el modelo usado para generar las respuestas del chatbot, como lo es **gpt-4o**.

Cabe mencionar que el conjunto de datos completo, incluyendo los prompts utilizados, las respuestas del chatbot y las evaluaciones detalladas generadas por el juez para cada interacción, se encuentra disponible para su consulta en el anexo 4.

Fundamentación (Test Grounding)

Esta prueba verifica que las respuestas generadas estén respaldadas por la información contenida en el contexto (RAG) proporcionado, asignando puntuaciones de 0 a 5 según qué tan fielmente la

respuesta se basa en las fuentes de información del RAG. Lo cual es importante para evidenciar que no haya alucinaciones.

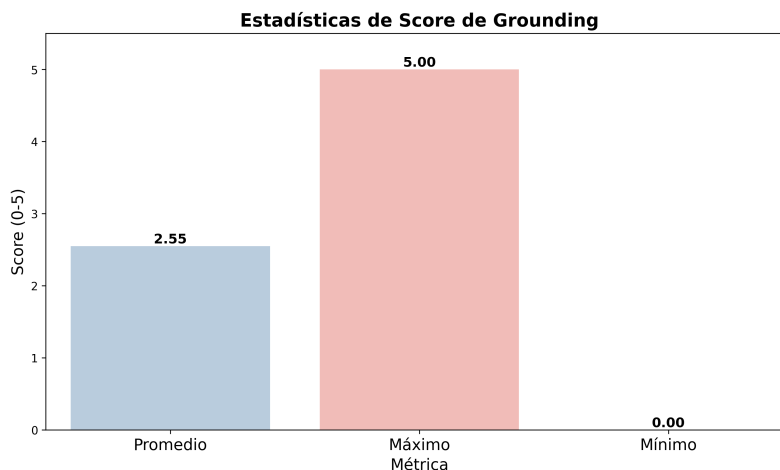


Figura 7.2: Niveles de fidelidad al contexto RAG: Mínimo, Promedio y Máximo

Fuente: Elaboración propia

La Figura 7.2 permite observar que hay un bajo uso del contexto en las respuestas, identificando un promedio de 2.55 en la puntuación de las respuestas, con algunas puntuaciones en 0, lo que indica un uso inexistente del contexto RAG. Sin embargo, este resultado debe interpretarse considerando el propósito y diseño del sistema RAG implementado.

La base de conocimiento del chatbot está orientada específicamente a proporcionar técnicas y recursos de apoyo para situaciones de deterioro emocional como estrés, ansiedad, depresión y otros estados negativos que puedan experimentar los estudiantes universitarios. Por lo tanto, el sistema rag es imprescindible únicamente cuando el usuario experimenta y comparte un estado emocional negativo.

Al analizar los casos con puntuaciones bajas en fundamentación, se identificó un patrón consistente: corresponden a interacciones donde el usuario expresa su estado emocional positivo (alegría, entusiasmo, satisfacción). En estos escenarios, el chatbot responde de manera empática y conversacional sin recurrir a las técnicas del RAG, ya que no existe una necesidad de intervención o apoyo emocional.

Relevancia (test relevance)

De manera complementaria, esta prueba evaluó la pertinencia de las respuestas, midiendo que tan alineada está la conversación con la intención del usuario, la emoción detectada y las reglas de personalidad del sistema. A diferencia de la fundamentación que tan solo recibía el contexto, pregunta y respuesta, verificando solo el origen de la información, la relevancia juzga la calidad de la interacción.

La figura 7.3 arroja los resultados, con un promedio destacado de 4.6 sobre 5.0 en relevancia. Este puntaje alto, contrastado con el puntaje obtenido en fundamentación, confirma la hipótesis planteada anteriormente sobre el manejo de emociones positivas.

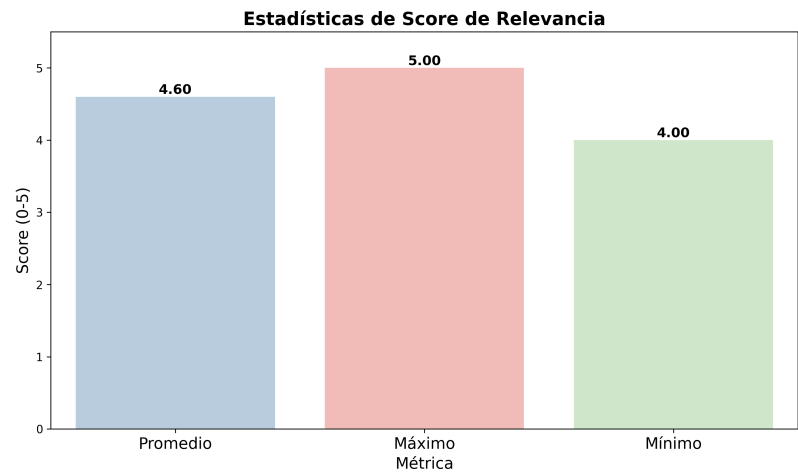


Figura 7.3: Niveles de Relevancia de la respuesta: Mínimo, Promedio y Máximo
Fuente: Elaboración propia

Para evaluar el comportamiento del chatbot en emociones positivas, la tabla 7.1 presenta un caso real extraído de las pruebas, donde se observa la relación inversa entre fundamentación y relevancia en estos casos.

Componente	Contenido/Valor
Entrada usuario	Saqué buena nota en la expo, valió la pena trasnochar.
Respuesta chatbot	¡Qué bien! Felicitaciones por esa buena nota en la expo. Todo el esfuerzo y las trasnochadas valieron la pena. Esos momentos de sacrificio a menudo traen grandes recompensas. ¿Tienes algún plan para celebrar tu éxito? ¡Disfrútalo!
Puntaje fundamentación	0.5
Puntaje relevancia	5
Descripción evaluaciones	Grounding: La respuesta del chatbot no está fundamentada en el contexto proporcionado. El contexto se centra en técnicas de manejo del estrés y la planificación, mientras que la respuesta del chatbot se enfoca en felicitar al usuario por su éxito en una exposición, sin relación directa con el contenido del contexto. Relevance: La respuesta es totalmente pertinente, celebra el logro del usuario y se adapta a la emoción de alegría detectada. Usa un tono amigable y cálido, cumpliendo con las pautas del PROMPT_ORIGINAL.

Tabla 7.1: Ejemplo de evaluación de métricas en interacción positiva
Fuente: Elaboración propia

7.2. Pruebas de usabilidad

Para evaluar la usabilidad del sistema, se consideraron dos perspectivas diferentes pero complementarias. Por una parte, se llevó a cabo una evaluación heurística basada en los principios de Jakob

Nielsen, realizada a profesionales con formación y experiencia en diseño y análisis de software, capaces de identificar problemas de usabilidad de manera técnica. Por otra parte, se aplicó una encuesta con escala de Likert a estudiantes universitarios, quienes representan a los usuarios finales. Esta estrategia permitió obtener una perspectiva equilibrada entre la experiencia real de uso y el análisis técnico especializado, proporcionando una validación más completa y fundamentada del sistema.

7.2.1. Evaluación experta basada en las heurísticas de Nielsen

Para la evaluación heurística se emplearon los diez principios de usabilidad de Jakob Nielsen. Estos principios funcionan como lineamientos generales ampliamente aceptados en el diseño de interfaces, orientados a identificar áreas de mejora en la experiencia del usuario, más que como reglas estrictas. La evaluación se realizó mediante un cuestionario estructurado basado en dichas heurísticas, aplicado a un pequeño grupo de cuatro profesionales en Ingeniería de Sistemas. Los participantes revisaron la aplicación y proporcionaron su valoración de acuerdo con cada principio, permitiendo identificar aspectos de la interfaz que requerirían ajustes y mejoras desde una perspectiva técnica.

El 100 % de los encuestados consideró que el prototipo informa claramente lo que está ocurriendo, otorgando la puntuación máxima. Este resultado refleja que el sistema comunica adecuadamente su estado mediante indicadores visuales como animaciones de carga y el estado “escribiendo...” del chatbot, como se observa en la figura 7.4.

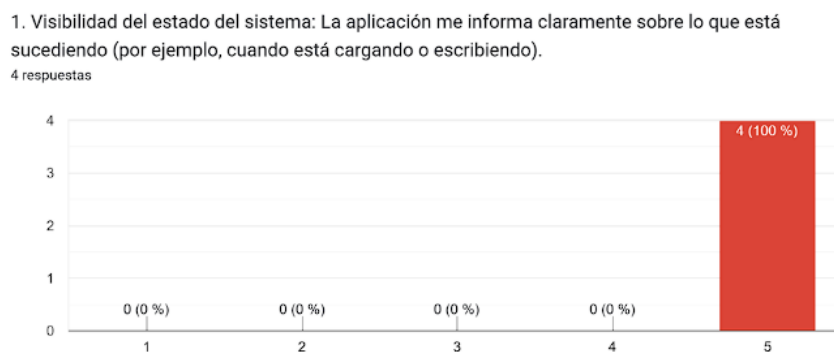


Figura 7.4: Resultados prueba Jakob Nielsen: Visibilidad del estado del sistema

Fuente: Elaboración propia

Respecto a la correspondencia entre el sistema y el mundo real, el 100 % de los evaluadores otorgó puntuación máxima también indicando que el lenguaje y los iconos son intuitivos y familiares para los usuarios, lo que facilita la comprensión inmediata del sistema. En lo relacionado con el control y libertad del usuario, obtuvo una valoración mixta: el 75 % de los expertos asignó una puntuación neutral y el 25 % restante otorgó la puntuación máxima (5). Este resultado sugiere que existen oportunidades para mejorar los mecanismos de deshacer o cancelar acciones, como se observa en la figura 7.5.

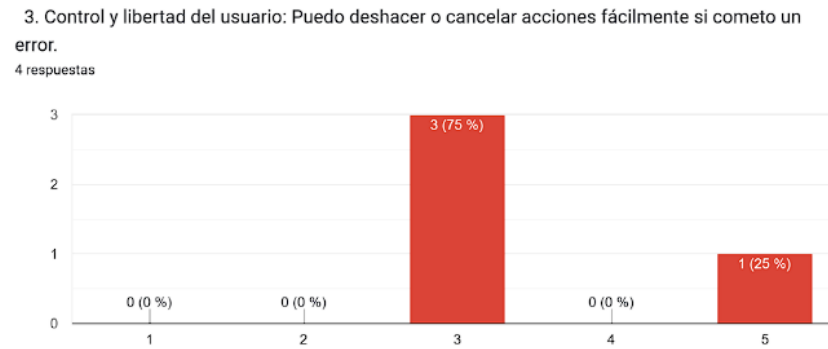


Figura 7.5: Resultados prueba Jakob Nielsen: Control y libertad del usuario
Fuente: Elaboración propia

En cuanto a la heurística de consistencia y estándares, los resultados fueron positivos, con 75 % completamente de acuerdo y 25 % de acuerdo, indicando que la interfaz mantiene consistencia en sus elementos visuales, patrones de navegación y comportamiento de componentes. Esta uniformidad contribuye significativamente a la experiencia del usuario al reducir la curva de aprendizaje.

La prevención de errores presentó una distribución más variada: 50 % muy de acuerdo, 25 % de acuerdo, y 25 % neutral. Esto indica que, aunque la mayoría considera adecuadas las medidas preventivas, existe margen para reforzar advertencias y confirmaciones en acciones importantes, como muestra la figura 7.6.

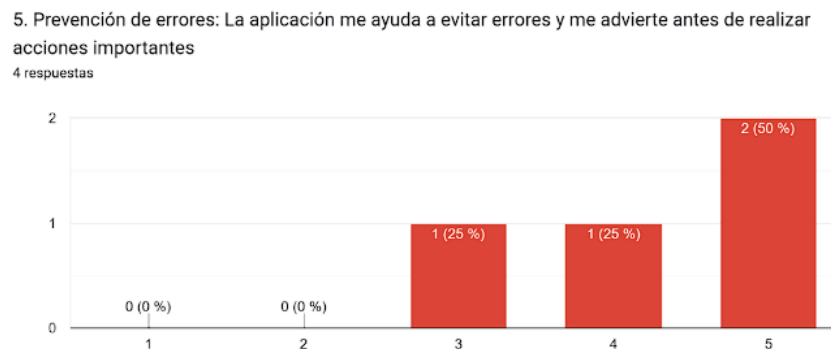


Figura 7.6: Resultados prueba Jakob Nielsen: Prevención de errores
Fuente: Elaboración propia

Por el lado de Reconocer antes que recordar, los resultados fueron completamente positivos con un 75 % muy de acuerdo y un 25 % de acuerdo, ningún evaluador dio una calificación negativa o neutral, lo que indica que los elementos de la interfaz son fácilmente reconocibles sin requerir que el usuario memorice información entre pantallas, esto contribuye a una curva de aprendizaje rápida.

En cuanto a flexibilidad y eficiencia de uso, los resultados estuvieron divididos pero completamente positivos: 50 % de acuerdo y 50 % muy de acuerdo. Esto indica que la aplicación es accesible tanto para usuarios novatos como para aquellos con mayor experiencia en aplicaciones móviles.

La heurística de diseño estético y minimalista obtuvo 100 % de valoración máxima, lo que con-

firma que la interfaz es visualmente limpia, sin elementos necesarios que distraigan al usuario. El diseño minimalista facilita el enfoque en la tarea principal (la conversación emocional con la mascota) y contribuye a una experiencia visual agradable.

Por otro lado, ayudar a reconocer, diagnosticar y recuperarse de errores, los resultados fueron positivos pero divididos: 50 % de acuerdo y 50 % muy de acuerdo. Esto indica que los mensajes de error son generalmente claros y útiles, aunque existe margen de mejora para proporcionar orientación más específica sobre cómo resolver problemas cuando estos ocurren.

Para finalizar, la heurística ayuda y documentación presentó la mayor dispersión en las valoraciones: 25 % neutral, 50 % de acuerdo, y 25 % muy de acuerdo. Los resultados sugieren que, aunque la mayoría encuentra la aplicación suficientemente intuitiva, algunos usuarios podrían beneficiarse de documentación adicional, especialmente para funcionalidades como la interpretación del semáforo emocional y las gráficas de evolución, como muestra la figura 7.7.

10. Ayuda y documentación: Si lo necesito, puedo encontrar fácilmente ayuda o instrucciones sobre cómo usar la aplicación.

4 respuestas

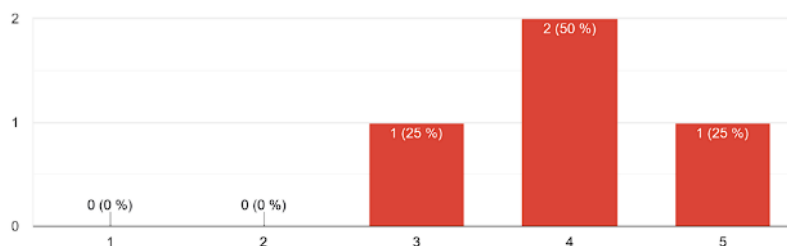


Figura 7.7: Resultados prueba Jakob Nielsen: Ayuda y documentación

Fuente: Elaboración propia

La evaluación heurística reveló un nivel de usabilidad muy satisfactorio, donde tres heurísticas obtuvieron puntuación perfecta, lo que indica fortalezas significativas en: comunicación clara del estado del sistema, uso de lenguaje familiar e intuitivo y diseño visual limpio y enfocado. Se identificaron áreas de mejora en control y libertad del usuario, prevención de errores y ayuda y documentación. Ninguna heurística obtuvo valoración negativa (por debajo de 3.0), lo que confirma que el diseño base es sólido y no requiere reestructuración fundamental, solo refinamientos específicos en las áreas identificadas.

7.2.2. Evaluación con usuarios finales

La encuesta evaluó cuatro aspectos claves de la experiencia de usuario: diseño visual e interfaz, funcionalidades del chatbot, visualización de los datos emocionales y utilidad percibida e intención de uso. En total, participaron 40 estudiantes de la Universidad del Valle, sede Tuluá, quienes interactuaron con la aplicación antes de responder el cuestionario. La recolección de datos se llevó a cabo a través de una sesión virtual programada, en la que los participantes probaron la aplicación en tiempo real y, de manera complementaria, se invitó a otros estudiantes interesados a participar. Se formularon 11 preguntas cerradas con escala Likert de 5 puntos y dos preguntas abiertas

La Tabla 7.2 presenta los resultados consolidados, incluyendo promedios por dimensión evaluada.

Aspecto evaluado	Prom. (1-5)	% Acuerdo
Diseño e Interfaz		
Diseño atractivo y agradable	4,3	90 %
Iconos y elementos claros	4,55	97,5 %
Mascota hace interacción amena	4,2	85 %
Promedio general	4,35	
Funcionalidad Chatbot		
Comprensión de emociones	4,05	80 %
Utilidad de respuestas	4,17	87,5 %
Naturalidad conversación	4,01	82,5 %
Promedio general	4,07	
Visualización Datos		
Semáforo emocional	4,37	90 %
Gráfica semanal	4,05	80 %
Promedio general	4,21	
Utilidad e Intención		
Ayuda gestión emocional	3,9	72,5 %
Intención uso regular	3,62	62,5 %
Probabilidad recomendación	4,1	82,5 %
Promedio general	3,87	
PROMEDIO TOTAL	4,12	82,7 %

Tabla 7.2: Resultados de la encuesta de usabilidad

Fuente: Elaboración propia.

Diseño e interfaz

Analizando los resultados detalladamente por ítem, se encuentra que para el 60 % de los estudiantes los iconos son "muy fáciles" de entender, y para un 37,5 % son "fáciles", lo que confirma que el 97,5 % total de los usuarios encuentra la interfaz coherente y entendible. Este aspecto fundamental de la usabilidad contribuye significativamente a la percepción general positiva del sistema, como se aprecia en la figura 7.8.

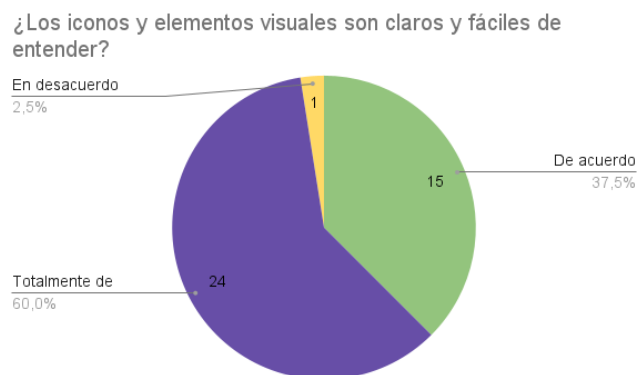


Figura 7.8: Claridad de los iconos y elementos visuales
Fuente: Elaboración propia

La mascota virtual cumplió exitosamente su función de hacer la interacción más amena (4,2 con 85 % de acuerdo), confirmando que este elemento diferenciador aporta valor percibido a la experiencia. El diseño general fue percibido como atractivo y agradable (4,3 con 90 % de acuerdo).

Funcionalidad del chatbot

Por otro lado, la funcionalidad del chatbot obtuvo una puntuación satisfactoria, indicando que cumple su propósito de apoyo emocional. Los usuarios percibieron las respuestas como útiles y naturales (4,17), donde el 87,5 % estuvieron de acuerdo en que el chatbot era útil (el 55 % de acuerdo y el 32,5 % totalmente de acuerdo).

Sin embargo, la comprensión de emociones (4,05 con 80 %) fue el aspecto con menor valoración dentro del ítem, sugiriendo oportunidades de mejora en la precisión del modelo de análisis emocional. Aunque el 80 % de acuerdo indica que la mayoría de usuarios considera que el chatbot identifica correctamente sus emociones, existe una proporción significativa (19 %) que experimenta detección imprecisa, como se observa en la figura 7.9.

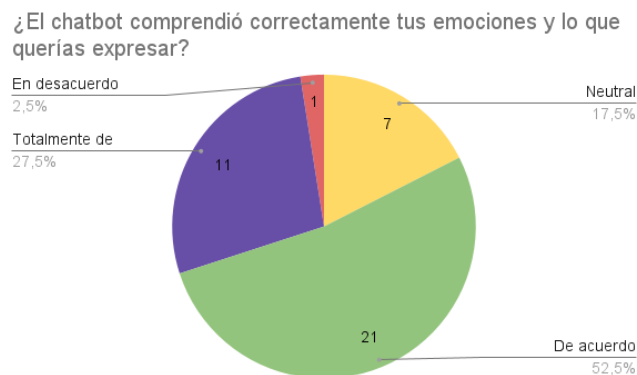


Figura 7.9: Utilidad percibida de las respuestas del chatbot
Fuente: Elaboración propia

Visualización de datos

En cuanto a la visualización de los datos, se obtuvo una valoración alta confirmando la efectividad de la parte gráfica. El semáforo emocional destacó, con un 52,5 % de usuarios totalmente de acuerdo y un 37,5 % de acuerdo, alcanzando una puntuación de 4,37 y un 90 % de acuerdo. Esto indica que este elemento cumple con su función, como se aprecia en la figura 7.10.

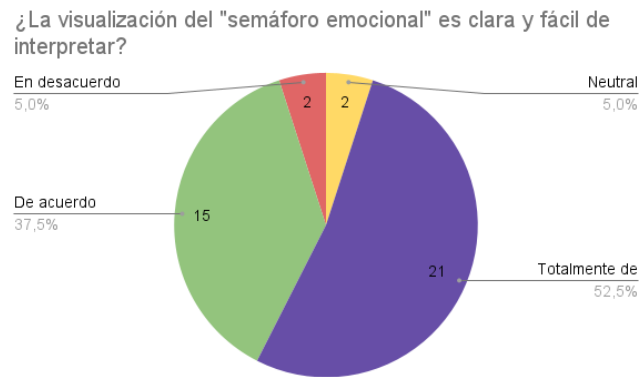


Figura 7.10: Claridad del semáforo emocional
Fuente: Elaboración propia

A su vez, la gráfica semanal de evolución emocional obtuvo 4,05 con 80 % de acuerdo, demostrando que los usuarios encuentran valor en poder visualizar sus patrones emocionales a lo largo del tiempo, lo cual facilita la reflexión y el autoconocimiento emocional.

Utilidad percibida e intención

Por último, los ítems de utilidad percibida e intención muestran una valoración positiva pero ligeramente inferior, especialmente en la intención de uso regular (3,62). Esto podría indicar que, a pesar de encontrar la herramienta útil y atractiva, los usuarios necesitan un incentivo o recordatorio más fuerte para integrarla en su rutina diaria. Cabe destacar que lo mejor valorado fue la probabilidad de recomendación, donde el 30 % optó por "probable" el 52,5 % por "muy probable", lo que significa que el 82,5 % de los usuarios recomendaría la aplicación. Esto indica que los usuarios están dispuestos a promoverla entre sus conocidos, un indicador importante de satisfacción general, como se observa en la figura 7.11.

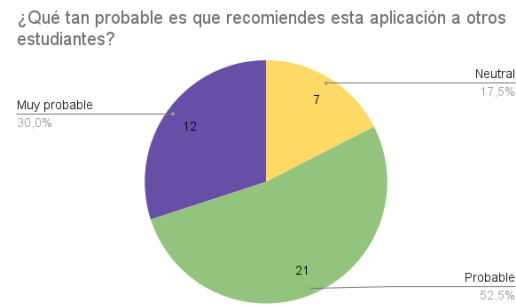


Figura 7.11: Probabilidad de recomendación de la aplicación
Fuente: Elaboración propia

El promedio general de 4,12 indica una aceptación positiva del prototipo por parte de los usuarios finales, aunque se identificaron aspectos de mejora para optimizar su validación. En las respuestas a las preguntas abiertas, los participantes destacaron especialmente el diseño de la mascota virtual, que resultó ser atractivo y logró generar la cercanía emocional esperada. Asimismo, entre las sugerencias recogidas, se mencionó la posibilidad de incorporar diferentes apariencias (skins) para la mascota, lo que permitiría personalizar la experiencia de uso.

Capítulo 8

Conclusiones y trabajos futuros

8.1. Conclusiones

- Durante el desarrollo de este trabajo, se hizo evidente la creciente demanda de recursos de apoyo emocional que sean accesibles y no generen estigmas en el ámbito universitario, donde los alumnos se enfrentan a elevados niveles de estrés, ansiedad y depresión. La creación de una mascota virtual con inteligencia emocional demostró ser una alternativa efectiva y bien aceptada, que integra nociones de técnicas para el bienestar universitario, diseño enfocado en el usuario e inteligencia artificial conversacional.
- Se eligió el modelo **Pysentimiento** para el reconocimiento de emociones, al estar preentrenado específicamente para español y no requerir fine-tuning. No obstante, la evaluación con usuarios mostró que un 19 % de ellos percibió que el chatbot no detectó correctamente su estado emocional. Esto indica que, pese a las ventajas de modelos especializados como Pysentimiento, la detección de emociones en lenguaje natural sigue siendo un desafío debido a la complejidad del lenguaje, que incluye matices como la ironía y el sarcasmo.
- El contexto tecnológico subrayó la importancia de los frameworks de orquestación para desarrollar sistemas conversacionales. En este ecosistema, LangChain se destaca como una de las herramientas más empleadas, y su extensión LangGraph es especialmente útil al posibilitar la representación del diálogo como un grafo de estados; además, facilita la orquestación del modelo de lenguaje, la memoria de la conversación y el sistema de recuperación de información (RAG), convirtiéndolo en un entorno ideal para desarrollar chatbots conversacionales.
- La Generación Aumentada por Recuperación (RAG) fue reconocida, después de revisar la literatura, como la técnica de Procesamiento del Lenguaje Natural más idónea para este proyecto. Su implementación permitió el desarrollo del sistema conversacional sin requerir el entrenamiento de un modelo desde el inicio, al extraer información de una base de conocimiento establecida y producir respuestas rápidas. Asimismo, permite la renovación e inclusión de información reciente de forma eficiente, asegurando que el chatbot pueda ajustarse a nuevas exigencias o datos sin altos costos de reentrenamiento.
- El desarrollo de una aplicación móvil beneficia a los usuarios al garantizar un acceso inmediato y constante a la herramienta, dado que los teléfonos son de uso cotidiano. Esto permite que puedan recurrir a ella en cualquier momento, estableciendo una conexión emocional cada vez que el usuario lo necesite.
- En conjunto, la integración de tecnologías como el modelo preentrenado Pysentimiento, los frameworks de orquestación LangChain y LangGraph, la técnica RAG y la implementación

en aplicación móvil y entorno cloud, permitieron desarrollar un chatbot capaz de reconocer emociones, gestionar diálogos complejos y responder de manera rápida y accesible. Estas herramientas tecnológicas consolidan la efectividad del sistema, asegurando su viabilidad y su aporte real al bienestar emocional de los estudiantes universitarios

- Con base en los resultados de las pruebas de usabilidad, se concluye que la aplicación móvil fue bien recibida, con un promedio general de 4,12 que refleja una alta aceptación. Destacan elementos específicos como la mascota virtual (85 %), la claridad de los iconos (97,5 %) y el semáforo emocional (90 %), los cuales fueron particularmente valorados por los usuarios. Asimismo, el 87,5 % de los participantes consideró que el chatbot cumplía con su funcionalidad. Por otro lado, la evaluación heurística también arrojó resultados positivos, donde tres de sus ítems alcanzaron una puntuación perfecta del 100 %, destacando fortalezas significativas en visibilidad del estado del sistema, correspondencia con el mundo real y diseño estético minimalista. En conjunto, los resultados evidencian que el prototipo cumple con su propósito de ofrecer un apoyo emocional accesible, intuitivo y empático, validando su viabilidad y potencial impacto en el bienestar estudiantil.

8.2. Trabajos futuros

A lo largo del desarrollo de este trabajo y con base en las pruebas realizadas, se evidenciaron varios aspectos susceptibles de mejora, entre los cuales se encuentran:

- **Skins de la mascota:** Implementar un sistema de múltiples apariencias (skins) para la mascota virtual. Aunque el diseño actual se basó en la identidad institucional y fue bien recibido, ofrecer variedad de estilos podría fortalecer la identificación y conexión emocional de los usuarios, permitiendo mayor personalización y accesibilidad afectiva. Esta necesidad fue una de las sugerencias más recurrentes en las respuestas abiertas de la encuesta a usuarios finales.
- **Interacción táctil con la mascota:** Implementar animaciones que permitan a la mascota reaccionar a los toques del usuario en la pantalla, mejorando la interactividad y fortaleciendo el vínculo emocional mediante una experiencia más dinámica.
- **Ampliación de la base de conocimiento:** Aunque la base actual aborda aspectos clave de la vida estudiantil (como el estrés, la ansiedad y la gestión del tiempo), sería conveniente incorporar más temas de relevancia para los estudiantes, con el fin de ampliar y enriquecer el soporte emocional ofrecido.
- **Compatibilidad con iOS:** Actualmente, la aplicación solo está disponible para dispositivos Android. Sería deseable extender su desarrollo para que también sea compatible con iOS, permitiendo así llegar a un mayor número de usuarios.
- **Panel de control para el usuario:** Crear una vista más detallada con estadísticas, gráficas interactivas y recomendaciones personalizadas sobre su evolución emocional y patrones detectados, permitiendo al usuario tener una visión más integral de su estado emocional a lo largo del tiempo.
- **Integrar comunicación por voz:** Incorporar el reconocimiento y síntesis de voz que permita a los usuarios interactuar con la mascota mediante comandos de voz y recibir respuestas auditivas. Esta funcionalidad mejoraría la accesibilidad y ofrecería una experiencia más natural e inmediata.
- **Detección emocional por visión computacional:** Integrar un módulo que, utilizando la cámara frontal del dispositivo, detecte expresiones faciales y las clasifique emocionalmente. Aunque su implementación representa un desafío técnico considerable, su potencial impacto en la precisión y experiencia de usuario la convierte en una mejora altamente prometedora.
- **Modelos emocionales multidimensionales:** Investigar la adopción de modelos que clasifiquen las emociones considerando múltiples factores, más allá de la polaridad positiva o negativa utilizada actualmente, como por ejemplo la integración de la dimensión de activación (el segundo eje del modelo de russell, además de la valencia explicada en las funcionalidades de bienestar emocional). Mejorando la precisión de las estrategias de regulación emocional.

Capítulo 9

Bibliografía

- [1] C. Breazeal, “Diseño de máquinas sociables,” in *Agentes socialmente inteligentes* (K. Dautenhahn, A. Bond, L. Cañamero, and B. Edmonds, eds.), vol. 3, Boston, MA, USA: Springer, 2002.
- [2] D. Goleman, *Inteligencia emocional*. Barcelona, España: Editorial Kairós, 1995.
- [3] OMS, “Promoting mental health: Concepts, emerging evidence, practice: Summary report,” 2020.
- [4] T. Shibata and K. Wada, “Robot therapy: A new approach for mental healthcare of the elderly - a mini-review,” 2011.
- [5] N. C. Krämer, S. Eimler, A. V. D. Pütten, and S. Payr, “Theory of companions: What can theoretical models contribute to applications and understanding of human-robot interaction?,” *Applied Artificial Intelligence*, vol. 25, 2011.
- [6] Organización Mundial de la Salud, “Presencia de la oms en los países, territorios y zonas: informe de 2019: resumen de orientación,” Tech. Rep. WHO/CCU/19.06, Organización Mundial de la Salud (OMS), 2019. Consultado el 26 oct 2024. Disponible en: <https://iris.who.int/bitstream/handle/10665/324774/WHO-CCU-19.06-spa.pdf>.
- [7] Organización Mundial de la Salud, “Trastorno depresivo (depresión),” Mar. 2023. Consultado el 27 oct 2024. Disponible en: <https://www.who.int/en/news-room/fact-sheets/detail/depression>.
- [8] R. P. Auerbach, P. Mortier, R. Bruffaerts, J. Alonso, C. Benjet, P. Cuijpers, and K. Demyttenaere, “Proyecto de estudiantes universitarios internacionales de las encuestas mundiales de salud mental de la oms: prevalencia y distribución de los trastornos mentales,” *Abnormal Psychology*, 2018.
- [9] J. M. A. López and J. A. F. Lozano, “Inteligencia artificial emocional. desafíos y limitaciones de la interacción entre humanos y máquinas,” *Medicina Estética. Revista Científica de la Sociedad Española de Medicina Estética (SEME)*, 2025.
- [10] J. Bossio, “Bienestar estudiantil... ¿también con ia?,” Oct. 2024. Consultado el 26 oct. 2024. Disponible en: <https://innovacioneducativa.upc.edu.pe/2024/10/22/bienestar-estudiantil-tambien-con-ia/>.
- [11] Microsoft News España, “Trabajos del futuro: enseñar empatía a la inteligencia artificial,” June 2019. Consultado el 27 oct. 2024. Disponible en: <https://news.microsoft.com/es-es/2019/06/17/trabajos-del-futuro-ensenar-empatia-a-la-inteligencia-artificial/>.

- [12] J. D. Polo, "Historia del tamagotchi," Sept. 2023. Consultado el 26 oct. 2024. Disponible en: <https://www.muyinteresante.com/tecnologia/959.html>.
- [13] E. de Peridot, "Las mascotas virtuales de peridot son ahora más reales (y adorables) que nunca gracias a la ia generativa.," 11 2023.
- [14] R. Lopes, A. Silva, A. Rodrigues, and M. V, "Chatbots para el bienestar: exploración del impacto de la inteligencia artificial en la mejora del estado de ánimo y la salud mental," 2024.
- [15] M. J. Osorio, C. Zepeda, and J. L. Carballido, "Myubot: Towards an artificial intelligence agent system chat-bot for well-being and mental health," in *CEUR Workshop Proceedings*, vol. 2820, 2020.
- [16] 2020, "De ibm watson a watsonx," 2023. Consultado el 26 oct. 2024. Disponible en: <https://www.ibm.com/watson>.
- [17] Ángel Quintana Gómez, "Análisis de los procesos de tratamiento de información en un estudio de análisis de sentimiento utilizando la tecnología de google," *Vivat Academia*, 2021.
- [18] Amazon Web Services, "¿qué es un chatbot?," Consultado el 27 oct. 2024. Disponible en: <https://aws.amazon.com/es/what-is/chatbot/>.
- [19] Eve, "Woebot – the bleeding intelligent self-help therapist and companion," Apr. 2020. Consultado el 29 oct. 2024. Disponible en: <https://d3.harvard.edu/platform-digit/submission/woebot-the-bleeding-intelligent-self-help-therapist-and-companion/>.
- [20] K. K. Fitzpatrick, A. Darcy, and M. Vierhile, "Delivering cognitive behavior therapy to young adults with symptoms of depression and anxiety using a fully automated conversational agent (woebot): A randomized controlled trial," *JMIR Mental Health*, vol. 4, 2017.
- [21] M. Wang, "The limits of care: A critical analysis of ai companions' capacity for good care," in *2025 IEEE International Symposium on Ethics in Engineering, Science, and Technology (ETHICS)*, pp. 1–4, IEEE, 6 2025.
- [22] EBIS Business Techschool, "Replika ai: Qué es, cómo funciona y cuánto cuesta," Aug. 2024. Consultado el 29 oct 2024. Disponible en: <https://www.ebiseducation.com/replika-ai-que-es-como-funciona-y-cuanto-cuesta>.
- [23] MiTSoftware, "¿qué es replika ai friend?," Feb. 2024. Consultado el 30 oct. 2024. Disponible en: <https://mitsoftware.com/blog/Que-es-Replika-AI-Friend/>.
- [24] N. Agarwal and Vijayalaxmi, "Covid-19 impact on mental health: Sentiment analysis using nltk," in *Proceedings of International Conference on Contemporary Computing and Informatics, IC3I 2023*, 2023.
- [25] A. Beetz, K. Uvnäs-Moberg, H. Julius, and K. Kotrschal, "Psychosocial and psychophysiological effects of human-animal interactions: The possible role of oxytocin," 2012.
- [26] J. P. Polheber and R. L. Matchock, "The presence of a dog attenuates cortisol and heart rate in the trier social stress test compared to human friends," *Journal of Behavioral Medicine*, vol. 37, 2014.
- [27] G. Pérez, E. V, and V. M. C., "Estrés percibido y empatía en estudiantes universitarios que tienen mascotas," 4 2024.
- [28] B. M. A, *Inteligencia Artificial - Margaret A. Boden - Google Libros*. 2016.
- [29] G. M. de la Teja, "Ergonomía e interfases de interacción humano-computadora," *IX Congreso Internacional de Ergonomía*, 2007.
- [30] R. Dale, "The return of the chatbots," *Natural Language Engineering*, vol. 22, 2016.

- [31] R. J. Celi-Parraga, E. A. Varela-Tapia, I. L. Acosta-Guzmán, and N. R. Montaña-Pulzara, “Técnicas de procesamiento de lenguaje natural en la inteligencia artificial conversacional textual,” *AlfaPublicaciones*, vol. 3, 2021.
- [32] M. D. Jorge, “Redes neuronales: Conceptos básicos y aplicaciones.,” *Historia*, 2001.
- [33] H. Heppner, B. Schiffhauer, and U. Seelmeyer, “Conveying chatbot personality through conversational cues in social media messages,” *Computers in Human Behavior: Artificial Humans*, vol. 2, 2024.
- [34] L. Ruhiainen, “Inteligencia artificial 101 cosas que debes saber hoy sobre inteligencia artificial,” *Inteligencia Artificial*, vol. 1, 2018.
- [35] C. Stryker and J. Holdsworth, “¿qué es el procesamiento de lenguaje natural (nlp)?,” 2024. Consultado el 1 nov. 2024. Disponible en: <https://www.monologg.com/mx-es/think/topics/natural-language-processing>.
- [36] S. Rafatirad, H. Homayoun, Z. Chen, and S. M. P. Dinakarrao, *What Is Applied Machine Learning?* 2022.
- [37] A. K. Dubey, K. Gulabani, and R. Rathi, “Empirical study to appraise consciousness of hci technologies,” in *2014 International Conference on Signal Propagation and Computer Technology, ICSPCT 2014*, 2014.
- [38] P. Ekman, *El rostro de las emociones*. RBA Libros, 2003. Consultado el 26 ene. 2026. Disponible en: https://books.google.com.co/books?id=HOTODwAAQBAJ&printsec=frontcover&source=gbs_ge_summary_r&cad=0.
- [39] A. Singh, A. Ehtesham, S. Mahmud, and J. H. Kim, “Revolutionizing mental health care through langchain: A journey with a large language model,” in *2024 IEEE 14th Annual Computing and Communication Workshop and Conference, CCWC 2024*, 2024.
- [40] U. Bhawe, M. M. Narendra, D. J. Bhadresh, J. A. Suhas, and R. R. Ajit, “Mindwell solace: Your mental health companion,” in *2024 4th Asian Conference on Innovation in Technology (ASIANCON)*, pp. 1–4, IEEE, 8 2024.
- [41] R. Vineeth, S. Maskey, U. S. Vishakan, and Y. Singh, “A proposed chatbot psykh your personal therapist and stress buster using rasa open-source framework,” in *2022 OPJU International Technology Conference on Emerging Technologies for Sustainable Development, OTCON 2022*, 2023.
- [42] S. H. Noorani, S. Khan, A. Mahmood, M. Ishtiaq, U. Rauf, and Z. Ali, “Transformative conversational ai: Sentiment recognition in chatbots via transformers,” in *2023 25th International Multitopic Conference (INMIC)*, pp. 1–6, IEEE, 11 2023.
- [43] P. Kaushik, Renuka, S. Parihar, M. S. Khalid, U. C. Sharma, and J. J. Mary, “Enhancing emotional support chatbots using deep learning & nlp: A study on personalization and effectiveness,” in *2024 1st International Conference on Advances in Computing, Communication and Networking (ICAC2N)*, pp. 1416–1421, IEEE, 12 2024.
- [44] A. Vuyyuru, T. L. Praveena, A. Sharma, M. Yelagandula, and S. Nelli, “Mental health therapist chatbot using nlp,” in *2024 International Conference on Artificial Intelligence and Quantum Computation-Based Sensor Application (ICAIQSA)*, pp. 1–6, IEEE, 12 2024.
- [45] K. Patil, R. Patil, V. Koyande, A. S. Thakur, and K. Kadam, “Analyzing chatbot architectures utilising deep neural networks,” in *2024 IEEE 6th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA)*, pp. 15–19, IEEE, 10 2024.

- [46] R. B. Mathew, S. Varghese, S. E. Joy, and S. S. Alex, "Chatbot for disease prediction and treatment recommendation using machine learning," in *Proceedings of the International Conference on Trends in Electronics and Informatics, ICOEI 2019*, 2019.
- [47] M. Sureshkumar, S. N. Kumar, J. S. Chris, and V. Lokesh, "Heltrak - a medical application with chatbot based on ai," in *2022 1st International Conference on Computational Science and Technology, ICCST 2022 - Proceedings*, 2022.
- [48] N. A. Godse, S. Deodhar, S. Raut, and P. Jagdale, "Implementation of chatbot for itsm application using ibm watson," in *Proceedings - 2018 4th International Conference on Computing, Communication Control and Automation, ICCUBEA 2018*, 2018.
- [49] S. S. F. Sacoto, *Diseño y desarrollo de un chatbot usando redes neuronales recurrentes y procesamiento de lenguaje natural para tiendas virtuales en comercio electrónico*. PhD thesis, 10 2021.
- [50] e. J. Anuj, "Retrieval augmented generation approach for multipdf chatbot using langchain," tech. rep., 2024.
- [51] R. Akkiraju, "Facts about building retrieval augmented generation-based chatbots," tech. rep., 2024.
- [52] R. R. G. Tripathi and S. k. Gopalaiyenga H. K. Chaubey, "Comparative analysis of rag, fine-tuning, and prompt engineering in chatbot development," in *2024 International Conference on Future Technologies for Smart Society (ICFTSS)*, pp. 169–172, 2024.
- [53] C. Ouaddi, L. Benaddi, L. Naimi, E. M. Bouziane, and A. Jakimi, "A graphical dsl for accelerating chatbot development," in *2024 7th International Conference on Advanced Communication Technologies and Networking (CommNet)*, pp. 1–6, IEEE, 12 2024.
- [54] P. K, A. Kamath, C. K. M, P. C, O. S. N, and A. S. J, "Enhancing q&a systems with multilingual text conversion and speech integration: Harnessing the power of langchain and large language models," in *2024 8th International Conference on Computational System and Information Technology for Sustainable Solutions (CSITSS)*, pp. 1–6, IEEE, 11 2024.
- [55] H.-C. Lee, K. Hung, G. M.-T. Man, R. Ho, and M. Leung, "Development of an rag-based llm chatbot for enhancing technical support service," in *TENCON 2024 - 2024 IEEE Region 10 Conference (TENCON)*, pp. 1080–1083, IEEE, 12 2024.
- [56] K. Suryavanshi, N. Thikekar, R. Pawar, and S. Ashtekar, "Implementation of rag based question-answering application," in *2025 International Conference on Data Science and Business Systems (ICDSBS)*, pp. 1–6, IEEE, 4 2025.
- [57] R. Singh, M. Paste, N. Shinde, H. Patel, and N. Mishra, "Chatbot using tensorflow for small businesses," in *Proceedings of the International Conference on Inventive Communication and Computational Technologies, ICICCT 2018*, 2018.
- [58] S. C. Qice Yan, "A chatbot for learning and life guide in the university based on langchain+chatglm," 6 2024.
- [59] V. Agatha and I. Setyawan, "Web chat-based application with large language model and transformers from hugging face for self-learning on storytelling skills," in *2024 International Electronics Symposium (IES)*, pp. 614–618, 2024.
- [60] A. A. Elsayed, A. I. Hafez, R. Ceprián, G. Martínez, A. Granados, B. Soriano, C. Llorens, and J. M. Sempere, "Practical design and implementation of virtual chatbot assistants for bioinformatics based on a nlu open framework," *Big Data and Cognitive Computing*, vol. 8, p. 163, 11 2024.

- [61] J. M. Rodríguez, H. Merlino, and E. Fernández, “Comportamiento adaptable de chatbots dependiente del contexto,” *Revista Latinoamericana de Ingenieria de Software*, vol. 2, 2014.
- [62] L. G. Westermann, P. Eberle, and G. Lu, “Plug-and-play text-based emotion recognition for chatbots as virtual companions for older people,” in *International Conference on Wireless and Mobile Computing, Networking and Communications*, vol. 2023-June, 2023.
- [63] C. H. Kao, C. C. Chen, and Y. T. Tsai, “Model of multi-turn dialogue in emotional chatbot,” in *Proceedings - 2019 International Conference on Technologies and Applications of Artificial Intelligence, TAAI 2019*, 2019.
- [64] M. Anastasiia and T. Korotyeva, “A chatbot of a person’s emotional state using a neural network,” in *2022 IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT)*, pp. 13–16, IEEE, 11 2022.
- [65] B. Rawat, A. S. Bist, U. Rahardja, Q. Aini, and Y. P. A. Sanjaya, “Recent deep learning based nlp techniques for chatbot development: An exhaustive survey,” in *2022 10th International Conference on Cyber and IT Service Management (CITSM)*, pp. 1–4, IEEE, 9 2022.
- [66] I. Naik, D. Naik, and N. Naik, “Chat generative pre-trained transformer (chatgpt): Comprehending its operational structure, ai techniques, working, features and limitations,” in *2023 IEEE International Conference on ICT in Business Industry & Government (ICTBIG)*, pp. 1–9, IEEE, 12 2023.
- [67] H. S. A. Paramarthalingam, S. Sundaramurthy, and S. Cirillo, “A study on word embeddings in local llm-based chatbot applications,” in *2024 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, pp. 232–237, IEEE, 11 2024.
- [68] N. Mathew, N. Chooramun, and M. S. Sharif, “Implementing a chatbot music recommender system based on user emotion,” in *2023 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies, 3ICT 2023*, 2023.
- [69] J. M. Pérez, J. C. Giudici, and F. Luque, “pysentimiento: A python toolkit for sentiment analysis and socialnlp tasks,” *ArXiv*, 2021.
- [70] Monologg, “Goemotions-pytorch,” 2020. Repositorio en GitHub. Consultado el 26 may. 2025. Disponible en: <https://github.com/monologg/GoEmotions-pytorch>.
- [71] Amontejo, “Emoeval@iberlef 2021,” Feb. 2021. Competición en Codalab. Consultado el 26 may. 2025. Disponible en: <https://competitions.codalab.org/competitions/28682>.
- [72] J. Ortiz Fuentes, “Spanish nlp,” 2024. Paquete de Python en PyPI. Consultado el 26 may. 2025. Disponible en: <https://pypi.org/project/spanish-nlp/#classification>.
- [73] JoseAGD, “Tarea emospeech 2024: Reconocimiento multimodal de emociones en español,” 2024.
- [74] F. G. S. R. V. G. R. P. José Antonio García Díaz, Miguel Ángel Rodríguez García, “Overview of emospeech at iberlef 2024 multimodal speech-text emotion recognition in spanish.”
- [75] I. Stuart, “Técnicas de gestión del tiempo para estudiantes,” June 2023. Consultado el 29 sep. 2025. Disponible en: <https://www.vitastudent.com/es/blog/productividad/tecnicas-de-gestion-del-tiempo-para-estudiantes/>.
- [76] Unifranz, “8 micro-hábitos de estudiantes para alcanzar el éxito académico que transformarán el 2025,” Feb. 2025. Consultado el 29 sep. 2025. Disponible en: <https://unifranz.edu.bo/blog/8-micro-habitos-de-estudiantes-para-alcanzar-el-exito-academico-que-transformaran-el-2025>.
- [77] U. S. Arboleda, “Diez consejos para mejorar tu concentración.” Consultado el 1 oct. 2025. Disponible en: <https://www.usergioarboleda.edu.co/noticias/diez-consejos-para-mejorar-tu-concentracion/>.

- [78] SHAIN, “Técnicas para mejorar la memoria y la concentración,” July 2023. Consultado el 1 oct. 2025. Disponible en: <https://www.shaio.org/estrategias-para-ejercitar-la-memoria-y-la-concentracion>.
- [79] M. Ceballo, “¿cómo superar la frustración?,” Sept. 2025. Consultado el 1 oct. 2025. Disponible en: <https://www.areandina.edu.co/blogs/como-superar-la-frustracion>.
- [80] C. S. M. España, “Guía de autoayuda para el manejo de la depresión y la ansiedad,” tech. rep., Aug. 2013. Consultado el 2 oct. 2025. Disponible en: <https://consaludmental.org/publicaciones/Guiasautoayudadepresionansiedad.pdf>.
- [81] A. Canada, “Respiración tranquila (respiración diafragmática).” Consultado el 2 oct. 2025. Disponible en: <https://www.anxietycanada.com/downloadables/calm-breathing-diaphragmatic-breathing/>.
- [82] A. C. Huertas, “Superar el estrés universitario: estrategias efectivas para una vida en equilibrio,” Feb. 2024.
- [83] B. W. L. G. F. P. D. d. l. A. E. Eileen Rosa, Soria, “Manejo del estrés académico,” *Revista Atlante: Cuadernos de Educación y Desarrollo*, 8 2018.
- [84] Área Humana, “Cómo combatir la ansiedad: guía de técnicas esenciales.” Consultado el 3 oct. 2025. Disponible en: <https://www.areahumana.es/como-combatir-la-ansiedad/>.
- [85] J. J. Demetrius, “Cómo aceptar la 2023,” Apr. 2023. Consultado el 4 oct. 2025. Disponible en: <https://www.arcadia.edu/student-life/meet-our-students/jj-demetrius-26/how-to-embrace-being-alone/>.
- [86] R. Shan, “A deep dive into vector stores: Classifying the backbone of retrieval-augmented generation,” in *2024 IEEE International Conference on Big Data (BigData)*, pp. 8831–8833, IEEE, 12 2024.
- [87] H. Face, “all-minilm-l6-v2,” 2023. Consultado el 29 nov. 2025. Disponible en: <https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>.
- [88] M. Londoño-Taborda, M. Lemos, and J. J. Orejuela, “Impacto de la adopción de una mascota en las percepciones de bienestar físico y emocional,” *Revista de Psicología Universidad de Antioquia*, vol. 10, pp. 53–74, 6 2019.
- [89] R. Inc., “What is rive?,” 2024. Consultado 27 nov. 2025. Disponible en: <https://www.learnrive.com/getting-started/what-is-rive>.
- [90] FastAPI, “Fastapi: framework web moderno y rápido para python,” 2024. Consultado el 28 nov. 2024. Disponible en: <https://fastapi.tiangolo.com/es/>.
- [91] LangChain, “Langgraph: workflows and agents.” Consultado el 29 nov. 2025. Disponible en: <https://docs.langchain.com/oss/python/langgraph/workflows-agents>.
- [92] J. A. Russell, “A circumplex model of affect.,” *Journal of Personality and Social Psychology*, vol. 39, pp. 1161–1178, 12 1980.
- [93] L. Riopel, “Escala panas: Escala de afectos positivos y negativos,” 8. Consultado el 26 ene. 2026. Disponible en: <https://positivepsychology.com/positive-and-negative-affect-schedule-panas/>.
- [94] Y. Zou, R. Wang, X. Xiong, C. Bian, S. Yan, and Y. Zhang, “Effects of acceptance and commitment therapy on negative emotions, automatic thoughts and psychological flexibility for depression and its acceptability: a meta-analysis,” *BMC Psychiatry*, vol. 25, p. 602, 7 2025.
- [95] A. W. Services, “¿qué es flutter?.” Consultado el 30 nov. 2025. Disponible en: <https://aws.amazon.com/es/what-is/flutter/>.

- [96] U. V. Parra, “Introducción a dart: la guía definitiva para comenzar a programar.” Consultado el 28 nov. 2024. Disponible en: <https://urianviera.com/dart/aprende-dart-desde-cero>.
- [97] R. B. Miller, “Response time in man-computer conversational transactions,” in *Proceedings of the December 9-11, 1968, fall joint computer conference, part I on - AFIPS '68 (Fall, part I)*, p. 267, ACM Press, 1968.
- [98] Y. Shang and C. Wang, “A method for enhancing invocation capabilities in tool calling with llm-as-a-judge system,” in *2025 5th International Conference on Advanced Algorithms and Neural Networks (AANN)*, pp. 1–5, IEEE, 8 2025.

Anexos

1. Pruebas de detección de emociones
2. Repositorio de los wireframes completos
3. Datos de prueba de rendimiento
4. Datos de prueba de LLM
5. Repositorio de la aplicación