
Ciudadano Digital: La Inteligencia Artificial como herramienta de acompañamiento informal en educación sobre ciudadanía y valores morales.

Erick Stiv Junior Guerra Muñoz



UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



**Ciudadano Digital: La Inteligencia Artificial como herramienta
de acompañamiento informal en educación sobre ciudadanía y
valores morales.**

Trabajo de graduación presentado por Erick Stiv Junior Guerra Muñoz
para optar al grado académico de Licenciado en Ingeniería en Ciencias
de la Computación y Tecnologías de la Información

Guatemala,

2025

Vo.Bo.:

(f) _____
MA. Luis Furlán

Tribunal Examinador:

(f) _____
MA. Luis Furlán

(f) _____
MSc. Douglas Barrios

(f) _____
PhD. Gabriel Barrientos

Fecha de aprobación: Guatemala, _____ de _____ de 2025.

Resumen

La inteligencia artificial (IA) ofrece un potencial transformador para la educación, especialmente en contextos marcados por desigualdades sociales, económicas y tecnológicas. En países como Guatemala, donde persiste una amplia brecha educativa, la IA puede convertirse en una herramienta clave para facilitar el acceso a aprendizajes significativos.

Uno de los ámbitos más desatendidos es la formación ciudadana y el desarrollo de valores morales que, aunque incluidos en los programas educativos, suelen abordarse de forma teórica y desvinculada de la realidad social. Es aquí donde se plantea la importancia de una herramienta que brinde acompañamiento a los estudiantes en la reflexión sobre su papel como ciudadanos y en la práctica de valores como el respeto, la empatía y la responsabilidad. Este proyecto combina tecnología accesible con un enfoque centrado en el usuario para fortalecer no solo el aprendizaje, sino también la conciencia social y ética de los jóvenes.

La implementación de esta herramienta incluye el procesamiento de contenidos educativos, el desarrollo de un asistente de IA adaptado a los contenidos seleccionados, a través de una aplicación móvil interactiva y, finalmente, la validación directa de la interacción con la herramienta para verificar que las respuestas dadas coincidan con el material proporcionado. Como resultado, se obtuvo una solución funcional, innovadora y escalable que demuestra cómo la IA puede contribuir significativamente al fortalecimiento de la educación en valores en entornos con recursos limitados.

Abstract

Artificial intelligence (AI) offers transformative potential for education, especially in contexts marked by social, economic, and technological inequalities. In countries like Guatemala, where a significant educational gap persists, AI can become a key tool to facilitate access to meaningful learning.

One of the most neglected areas is civic education and the development of moral values, which, although included in educational programs, are often addressed theoretically and detached from social reality. This is where the importance of a tool that supports students in reflecting on their role as citizens and in practicing values such as respect, empathy, and responsibility arises. This project combines both accessible technology and a user-centered approach to strengthen not only learning but also the social and ethical awareness of young people.

The implementation of this tool includes the digitization of educational content, the development of an AI assistant adapted to the selected contents, and finally, the direct validation of the interaction with the tool to ensure that the answers provided align with the material given. As a result, a functional, innovative, and scalable solution is sought that demonstrates how AI can significantly contribute to strengthening values education in environments with limited resources.

Índice

Resumen	III
Abstract	IV
Lista de cuadros	VIII
Lista de algoritmos	IX
Lista de figuras	X
1. Introducción	1
2. Justificación	3
3. Objetivos	5
3.1. Objetivo general	5
3.2. Objetivos específicos	5
4. Marco teórico	6
4.1. Educación ciudadana y valores	7
4.1.1. Educación en valores	7
4.1.2. Formación ciudadana	7
4.1.3. Competencias cívicas fundamentales	8
4.1.4. Educación moral	9
4.2. Aprendizaje informal y brecha educativa	10
4.2.1. Educación informal	10
4.2.2. Autoformación guiada	10
4.2.3. Brecha educativa y tecnológica	10
4.2.4. Tecnología como herramienta de inclusión educativa	11
4.3. Fundamentos de Inteligencia Artificial en Educación	11
4.3.1. Inteligencia Artificial aplicada a la educación	11
4.3.2. Modelos de Lenguaje de Gran Escala	11
4.3.3. Arquitectura y funcionamiento de sistemas RAG	11

4.3.4.	<i>Embeddings</i> y representación semántica del texto	14
4.3.5.	Bases de datos vectoriales y búsqueda semántica	15
4.3.6.	<i>Prompt Engineering</i> y diseño de instrucciones	15
4.3.7.	Tutoría personalizada con IA	16
4.3.8.	Método socrático aplicado a entornos digitales	16
4.3.9.	Métricas de evaluación de chatbots educativos	16
4.3.10.	Evaluación de calidad de respuestas en sistemas RAG	17
4.3.11.	Validación sin usuarios finales: enfoques y limitaciones	18
4.3.12.	Congruencia y fundamentación en respuestas educativas	19
4.4.	Ética y Responsabilidad en IA Educativa	19
4.4.1.	Principios éticos fundamentales en IA	19
4.4.2.	Sesgos algorítmicos y culturales en contextos latinoamericanos	20
4.4.3.	Transparencia y explicabilidad en sistemas inteligentes	20
4.4.4.	Responsabilidad ante respuestas incorrectas o inadecuadas	20
4.4.5.	Privacidad y seguridad en aplicaciones educativas para menores	20
4.4.6.	Supervisión pedagógica en sistemas automatizados	21
4.5.	Aprendizaje Móvil en Contextos de Recursos Limitados	21
4.5.1.	Panorama del aprendizaje móvil en Guatemala y Centroamérica	21
4.5.2.	Diseño de experiencias móviles para usuarios con baja alfabetización digital	22
4.5.3.	Consideraciones de conectividad intermitente y consumo de datos	22
4.5.4.	Aplicaciones móviles en la educación	22
4.6.	Tecnologías de Implementación	22
4.6.1.	Arquitectura cliente-servidor (<i>frontend/backend</i>)	22
4.6.2.	Frameworks de desarrollo móvil: Kotlin y ecosistema Android	23
4.6.3.	Bases de datos vectoriales y su contraste con bases relacionales	24
4.6.4.	APIs de IA: integración de modelos conversacionales	24
5.	Metodología	25
5.1.	Enfoque metodológico aplicado al contexto del proyecto	26
5.1.1.	Estructura de los <i>Sprints</i>	26
5.2.	<i>Sprint</i> 1: Identificación del perfil de usuario objetivo	27
5.2.1.	Ejecución	27
5.2.2.	Resultado final	28
5.3.	<i>Sprint</i> 2: Recolección y procesamiento del contenido educativo inicial	29
5.3.1.	Ejecución	29
5.3.2.	Resultado final	33
5.4.	<i>Sprint</i> 3: Construcción e implementación del <i>backend</i>	34
5.4.1.	Ejecución	34
5.4.2.	Resultado final	38
5.5.	<i>Sprint</i> 4: Desarrollo de la interfaz móvil en Kotlin	38
5.5.1.	Ejecución	39
5.5.2.	Resultado final	42
5.6.	<i>Sprint</i> 5: Pruebas y validación	43
5.6.1.	Ejecución	43
5.6.2.	Resultado final	44
5.7.	<i>Sprint</i> 6: Documentación y presentación	45
5.7.1.	Ejecución	45

5.7.2. Resultado final	46
6. Resultados	47
6.1. Implementación del modelo LLM y flujo RAG	47
6.2. Integración de la base de datos vectorial y relacional	48
6.3. Desarrollo del backend en NodeJS	49
6.4. Interfaz móvil en Kotlin (Frontend)	53
7. Discusión	55
8. Conclusiones	57
9. Bibliografía	59

Lista de cuadros

1.	Estructura de los <i>Sprints</i>	27
2.	Resultados de consultas y controles	50

Lista de algoritmos

1.	Proceso de limpieza y normalización profunda de texto	31
2.	Estandarización de títulos y numeración en texto	31
3.	Validación de integridad de texto	32
4.	Vectorización de fragmentos	33
5.	Controlador de solicitudes	35
6.	Validación de token	35
7.	Flujo completo de procesamiento de preguntas	38
8.	Flujo básico de un ViewModel	40

Lista de figuras

1.	Vista general del enfoque aplicado en el artículo de Lewis et al. [37]	12
2.	Arquitectura MVC (Modelo, Vista, Controlador)	34
3.	Diagrama entidad-relación (ER) de la base de datos relacional.	36
4.	Arquitectura MVVM (Modelo, Vista, ViewModel)	39
5.	Flujo de procesamiento de preguntas mediante RAG.	48
6.	Vista del índice vectorial en Pinecone con los embeddings cargados.	49
7.	Interfaz de la aplicación móvil desarrollada.	54

CAPÍTULO 1

Introducción

La formación en valores y ciudadanía constituye un componente esencial en la construcción de sociedades democráticas, inclusivas y participativas. A través de ella, los estudiantes desarrollan competencias cívicas como la empatía, la responsabilidad social, el respeto a la diversidad y el compromiso con el bien común. Organismos internacionales como la UNESCO (Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura) y la OCDE (Organización para la Cooperación y el Desarrollo Económicos) han destacado la necesidad de fortalecer estos aprendizajes en un contexto global marcado por tensiones sociales, crisis ambientales y transformaciones digitales profundas [1, 2]. Sin embargo, en muchos países de América Latina esta dimensión formativa sigue relegada frente a enfoques centrados exclusivamente en resultados académicos cuantificables [3, 4].

En Guatemala, el Currículo Nacional Base (CNB) reconoce la educación ciudadana como un eje transversal, pero su aplicación efectiva enfrenta múltiples desafíos: la escasa formación docente en metodologías críticas, la ausencia de recursos digitales adaptados al contexto local y la persistencia de enfoques tradicionales centrados en la memorización. Estas limitaciones dificultan que los estudiantes logren vincular los contenidos cívicos con su vida cotidiana o desarrollar una comprensión profunda de su papel como agentes de cambio [5, 6]. A ello se suma una brecha tecnológica significativa entre zonas urbanas y rurales, que restringe el acceso equitativo a experiencias de aprendizaje innovadoras y limita las oportunidades para fomentar la reflexión ética y la participación ciudadana [7, 8].

En este panorama, la inteligencia artificial (IA) emerge como una herramienta con gran potencial transformador para la educación, especialmente cuando se orienta hacia el fortalecimiento de habilidades humanas y el acompañamiento moral, más que hacia la mera automatización de contenidos. La UNESCO subraya que, para que la IA contribuya al desarrollo de sistemas educativos más justos y democráticos, debe diseñarse bajo principios de equidad, inclusión y supervisión humana, evitando reproducir sesgos o exclusiones [1, 9]. Aplicada con criterios éticos y pedagógicos, la IA puede servir como un medio para ampliar el acceso a materiales formativos, personalizar experiencias de aprendizaje y acompañar procesos de reflexión moral mediante un diálogo guiado y contextualizado [10, 11].

En los últimos años, el avance de los modelos de lenguaje de gran escala (LLMs, por sus siglas en inglés) ha impulsado el desarrollo de asistentes conversacionales capaces de generar tutorías personalizadas, ofrecer retroalimentación inmediata y adaptarse al ritmo de cada estudiante [12, 13]. Estas tecnologías abren la posibilidad de diseñar espacios de aprendizaje informal donde los jóvenes puedan explorar dilemas morales, reflexionar sobre valores y fortalecer su pensamiento crítico mediante la interacción con un sistema empático y culturalmente pertinente.

Desde esta perspectiva, el proyecto *Ciudadano Digital* busca integrar la inteligencia artificial en la formación ciudadana y moral a través de una aplicación accesible, diseñada para acompañar el aprendizaje ético de los jóvenes en entornos digitales. El sistema, basado en un modelo de lenguaje conectado a una base de datos vectorial con materiales educativos, éticos y contextuales, genera respuestas personalizadas y fundamentadas que orientan la reflexión. Su propósito no es sustituir al docente, sino complementar su labor mediante un acompañamiento continuo que promueva la autonomía moral, la empatía y la responsabilidad social. Además, su diseño prioriza la accesibilidad y la escalabilidad, de modo que pueda funcionar eficazmente en dispositivos de bajo costo y en entornos con recursos limitados. En conjunto, el proyecto pretende fortalecer la educación cívica guatemalteca mediante el uso ético y contextualizado de la IA, contribuyendo a una educación más inclusiva, reflexiva y humanista en la era digital [1, 3, 4].

CAPÍTULO 2

Justificación

La rápida expansión de la inteligencia artificial (IA) en el ámbito educativo ha transformado las dinámicas de enseñanza y aprendizaje, lo que ofrece nuevas posibilidades para la personalización, la tutoría automatizada y la retroalimentación inmediata [10]. Sin embargo, su adopción sin una orientación ética y pedagógica clara conlleva riesgos significativos, como la desinformación, la dependencia tecnológica o la reproducción de sesgos culturales [11]. En este escenario, se vuelve imprescindible diseñar propuestas que integren los beneficios de la IA con una visión educativa centrada en el desarrollo humano y la formación en valores.

En América Latina, y particularmente en Guatemala, los esfuerzos por incorporar tecnologías digitales al sistema educativo han estado marcados por desigualdades estructurales, carencias de infraestructura y una limitada capacitación docente en el uso pedagógico de herramientas tecnológicas [3, 4]. Aunque el Programa Nacional de Educación en Valores (Acuerdo Ministerial 2810-2023) reconoce la importancia de fortalecer la formación ciudadana [14], estudios nacionales han evidenciado que su implementación enfrenta obstáculos relacionados con la falta de recursos, la baja cobertura y la escasa integración de medios tecnológicos [6]. Estas condiciones reflejan la necesidad de propuestas educativas innovadoras, accesibles y adaptadas al contexto local que contribuyan a mejorar la educación cívica y moral.

Los modelos de lenguaje de gran escala (LLMs, por sus siglas en inglés) representan una oportunidad para atender esta necesidad. Su capacidad de mantener diálogos personalizados, ajustar el nivel de complejidad y ofrecer acompañamiento continuo los convierte en herramientas idóneas para promover la reflexión ética y el pensamiento crítico [15, 16]. En particular, el enfoque *Retrieval-Augmented Generation* (RAG) permite que estos sistemas fundamenten sus respuestas en documentos educativos verificados, garantizando precisión y coherencia [17]. A través de esta técnica, es posible crear un entorno conversacional confiable, orientado al aprendizaje moral y al fortalecimiento de competencias ciudadanas.

No obstante, el desarrollo de este tipo de tecnologías exige una atención especial a los principios de equidad, inclusión y transparencia algorítmica, tal como lo señalan las recomendaciones éticas de la UNESCO [1, 9]. La pertinencia de este proyecto radica en su propósito

de adaptar dichas innovaciones al contexto guatemalteco, mediante una solución funcional que no solo aproveche los avances de la IA, sino que también incorpore criterios específicos de la realidad nacional.

En este sentido, el proyecto *Ciudadano Digital* se justifica por tres razones fundamentales:

1. **Educativa**, porque propone un modelo de acompañamiento informal que complementa la labor docente y promueve la reflexión crítica en los jóvenes.
2. **Social**, porque busca reducir la brecha tecnológica mediante una aplicación accesible en dispositivos de bajo costo y adaptable a contextos con recursos limitados.
3. **Ética**, porque prioriza la formación ciudadana y moral a través del uso responsable de la inteligencia artificial.

Así, *Ciudadano Digital* responde a una necesidad real del sistema educativo guatemalteco: contar con herramientas tecnológicas que fortalezcan la educación en valores, promuevan el pensamiento crítico y fomenten la participación ciudadana activa desde una perspectiva inclusiva y humanista.

CAPÍTULO 3

Objetivos

3.1. Objetivo general

Desarrollar una herramienta tecnológica de educación informal orientada al acompañamiento en la adquisición de aprendizajes sobre formación ciudadana y valores morales.

3.2. Objetivos específicos

- Implementar un modelo LLM pre-entrenado y optimizado para obtener respuestas coherentes y acordes a la solicitud del usuario.
- Integrar una base de datos vectorial para almacenar y recuperar información actualizada que proporcione respuestas fundamentadas en el contenido preseleccionado.
- Desarrollar una interfaz gráfica atractiva para dispositivos móviles que permita la interacción entre el usuario y el modelo de inteligencia artificial.

CAPÍTULO 4

Marco teórico

La relación entre inteligencia artificial (IA) y educación se ha convertido en un campo de estudio emergente que combina aportes de la pedagogía, la psicología del aprendizaje y las ciencias de la computación. Diversas investigaciones han demostrado que los sistemas basados en IA pueden desempeñar funciones de apoyo al proceso educativo, desde la automatización de tareas administrativas hasta la personalización de la enseñanza mediante algoritmos de aprendizaje adaptativo [12, 13]. Sin embargo, más allá de sus aplicaciones instrumentales, la IA plantea un nuevo paradigma pedagógico que redefine la forma en que se conciben la enseñanza, la interacción docente-estudiante y la construcción del conocimiento en entornos digitales.

En este contexto, la formación en valores y ciudadanía adquiere especial relevancia. Aunque tradicionalmente se ha abordado desde marcos filosóficos y éticos, su integración con tecnologías emergentes permite explorar nuevas formas de aprendizaje moral mediadas por el diálogo y la reflexión guiada. La incorporación de sistemas inteligentes en este ámbito representa tanto una oportunidad como un desafío: por un lado, posibilita acompañamientos personalizados que estimulan el pensamiento crítico y la autorregulación ética; por otro, exige garantizar la responsabilidad, transparencia y confiabilidad de los modelos utilizados [18, 19].

Desde esta convergencia entre tecnología y formación ética, la literatura reciente destaca el potencial de los modelos de lenguaje de gran escala (LLMs, por sus siglas en inglés) para generar entornos conversacionales que promuevan la reflexión moral y la toma de decisiones fundamentadas [20, 21]. Estos sistemas, diseñados bajo principios éticos y pedagógicos, pueden convertirse en agentes de acompañamiento educativo informal, capaces de sostener diálogos significativos y culturalmente pertinentes. De esta manera, la IA no solo actúa como herramienta tecnológica, sino como mediadora cognitiva y moral, lo que amplia las posibilidades de aprendizaje y fortalece el desarrollo ciudadano en la era digital.

4.1. Educación ciudadana y valores

La educación ciudadana constituye un proceso educativo integral orientado a formar individuos capaces de ejercer sus derechos y deberes de manera responsable, ética y crítica. Esta formación no se limita al conocimiento de normas y leyes, sino que promueve valores como la solidaridad, la justicia y el respeto por la diversidad, esenciales para la convivencia democrática [22, 23]. Además, la educación ciudadana incorpora competencias sociales y habilidades de pensamiento crítico, fomentando la participación activa en la comunidad y la toma de decisiones informadas [24].

4.1.1. Educación en valores

La educación en valores constituye un enfoque pedagógico reconocido a nivel internacional bajo diversas denominaciones, como educación moral, educación del carácter o educación ética. Si bien cada una presenta matices particulares y distintos énfasis, todas comparten la convicción fundamental de que la formación en valores personales y cívicos representa una responsabilidad legítima de las instituciones educativas a nivel mundial. En la actualidad, este ámbito ya no se considera exclusivo del entorno familiar o religioso, pues diversas investigaciones han evidenciado que una educación desvinculada de los valores puede limitar de forma significativa el desarrollo integral del estudiante, tanto en el plano ético como en el académico [25].

Asimismo, la educación en valores se concibe como un proceso formativo integral que no solo promueve principios fundamentales de ética y ciudadanía, sino que se posiciona como un componente esencial y transversal de la calidad educativa. Lejos de tratarse de un aspecto aislado, establece una relación de mutua interdependencia con la enseñanza de calidad, al punto de integrarse en una dinámica de doble hélice que potencia el desarrollo personal, social y académico del estudiante [25].

4.1.2. Formación ciudadana

La formación ciudadana, bajo el concepto anglosajón de «*civic education*», es el conjunto de procesos, formales e informales, mediante los cuales las personas desarrollan conocimientos, valores, actitudes, habilidades y compromisos que les permiten participar activamente y de manera crítica en la vida democrática y comunitaria; éste no está limitado al ámbito escolar ni a una etapa específica de la vida del individuo, sino que se extiende a lo largo de su ciclo vital e involucra diversos aspectos externos como la familia, los medios de comunicación, su comunidad, instituciones educativas, etc. [26]

Por lo tanto, la formación ciudadana no se limita a la transmisión de contenidos normativos sobre el sistema político, sino que incorpora prácticas educativas activas, como la discusión de temas controversiales, la participación en acciones colectivas y la reflexión crítica, las cuales han demostrado tener efectos significativos en el desarrollo de una ciudadanía activa, consciente y empoderada [26].

4.1.3. Competencias cívicas fundamentales

Las competencias cívicas fundamentales son un conjunto integrado de disposiciones personales y capacidades que permiten a los individuos participar activamente en sociedades democráticas diversas. De acuerdo con el Consejo de Europa, estas competencias se organizan en torno a cuatro dimensiones esenciales: **los valores** que guían el comportamiento ético; **las actitudes** que predisponen a la apertura y al respeto; **las habilidades** necesarias para la interacción democrática; y **los conocimientos y la comprensión crítica del mundo** social, político y cultural. Su desarrollo es clave para convivir como iguales en contextos diversos y democráticos [27].

Valores

Los valores son creencias fundamentales que orientan a las personas hacia metas que consideran deseables en la vida. Funcionan como motores de acción y como criterios que guían la toma de decisiones, al proporcionar marcos de referencia sobre lo que se considera apropiado pensar o hacer en diversas situaciones. Estos principios no se limitan a contextos específicos, sino que ofrecen estándares para evaluar conductas, justificar posturas, elegir entre opciones, planificar acciones e influir en otros [27].

Actitudes

Las actitudes representan la disposición mental general que una persona adopta frente a individuos, grupos, instituciones, temas u objetos simbólicos. Esta orientación suele estar compuesta por cuatro elementos interrelacionados: una creencia o juicio cognitivo sobre el objeto, una respuesta emocional, una valoración positiva o negativa, y una inclinación conductual específica hacia dicho objeto [27].

Habilidades

Las habilidades son capacidades que permiten organizar y ejecutar de forma eficiente patrones complejos de pensamiento o acción, adaptándolos al contexto con el propósito de alcanzar un objetivo específico. [27]

Conocimientos y Comprensión Crítica

Los conocimientos representan el conjunto de información que una persona ha adquirido, mientras que la comprensión crítica implica no solo entender esa información, sino también valorar de forma reflexiva los sentimientos, perspectivas y significados asociados a ella. Este tipo de comprensión es esencial en contextos democráticos e interculturales, ya que permite analizar e interpretar activamente las situaciones, superando respuestas automáticas o no conscientes. En ese sentido, favorece la evaluación crítica de lo que se sabe y de cómo se interpreta el mundo social y político [27].

4.1.4. Educación moral

La educación moral es el proceso educativo centrado en la moralidad, entendida principalmente como la adhesión a normas morales y la creencia en su justificación. Este enfoque puede implicar dos dimensiones fundamentales: por un lado, la formación moral, que busca desarrollar en los individuos disposiciones afectivas, conductuales y motivacionales alineadas con esas normas; y por otro, la indagación moral, que promueve la reflexión crítica y la construcción de creencias fundamentadas sobre la validez de dichas normas. Ambas dimensiones pueden ser abordadas de manera complementaria, aunque conceptualmente son distintas. Además, el autor reconoce que la moralidad podría abarcar elementos adicionales, como ciertas virtudes o disposiciones emocionales, cuya formación también puede formar parte significativa de la educación moral [28].

Formación Moral

La formación moral es una dimensión de la educación moral centrada en el desarrollo de disposiciones afectivas y conductuales que llevan a una persona a adherirse a normas morales y a responder emocionalmente a ellas. No se trata únicamente de enseñar qué está bien o mal, sino de fomentar inclinaciones internas que impulsen a actuar conforme a ciertos estándares, de forma estable y espontánea. Estas disposiciones pueden incluir sentimientos de satisfacción cuando se actúa moralmente, incomodidad al violar principios morales, y expectativas de que otros también se comporten moralmente [28].

Asimismo, este concepto puede abarcar el cultivo de virtudes, entendidas no solo como inclinaciones a seguir normas, sino como capacidades para moderar emociones humanas fundamentales. Bajo esta perspectiva, la formación moral no se reduce a enseñar reglas, sino que apunta a moldear el carácter y las emociones de forma que apoyen una vida moral [28].

Indagación Moral

La indagación moral es la parte de la educación moral que se enfoca en investigar y evaluar la justificación de las normas morales. Consiste en un proceso cognitivo mediante el cual se analiza por qué una norma debería ser aceptada, se examinan los argumentos que la sustentan y se reflexiona críticamente sobre ellos. Creer en la justificación de una norma no es un requisito para adherirse a ella, por lo que esta indagación es distinta de la formación moral, que busca cultivar la adhesión emocional y conductual a esas normas [28].

En la enseñanza de la indagación moral, es posible adoptar un enfoque directivo, orientando al individuo hacia una conclusión particular sobre la validez de una norma, o un enfoque no directivo, en el que se facilita el análisis y la discusión sin influir en la opinión final. Ambos métodos promueven la capacidad del individuo para pensar críticamente sobre las normas morales y su justificación, complementando así la formación moral. [28]

4.2. Aprendizaje informal y brecha educativa

El aprendizaje informal constituye una estrategia educativa que ocurre fuera de los entornos formales, como escuelas o universidades, y se produce de manera espontánea en la vida cotidiana. Este tipo de educación fomenta la autonomía del aprendiz, la creatividad y la resolución de problemas; contribuyendo a reducir la brecha educativa, especialmente cuando el acceso a la educación formal es limitado [29, 30].

4.2.1. Educación informal

La educación informal se refiere a las formas de aprendizaje que ocurren de manera natural en la vida cotidiana, en una amplia variedad de contextos geográficos e históricos. Este tipo de educación no se limita a entornos específicos, sino que suele surgir en espacios donde las personas se sienten cómodas y con la libertad de socializar entre sí. Aunque este concepto es asociado tradicionalmente con actividades fuera de la escuela, hoy en día la educación informal también puede darse dentro de escuelas convencionales o en organizaciones como el voluntariado juvenil o el movimiento scout. [30]

Este tipo de educación se basa en el diálogo y la conversación, fomentando la confianza, el respeto y la empatía. No busca imponer resultados específicos, sino que promueve el aprendizaje a partir de las preocupaciones reales y cotidianas de las personas; generando cambios positivos y significativos en sus vidas. Además, la educación informal puede tener un carácter político, inspirándose en enfoques críticos que buscan que las personas tomen conciencia de las injusticias sociales y encuentren formas de superarlas, conectando lo personal con temas sociales y políticos más amplios [30].

4.2.2. Autoformación guiada

La autoformación guiada es un proceso intencional en el que el sujeto desarrolla su aprendizaje autónomo con el apoyo de una institución, un educador o un colectivo social. Aunque el aprendiz asume responsabilidad sobre sus objetivos, recursos, métodos y ritmos, recibe orientación y acompañamiento que facilitan el desarrollo de su capacidad de aprendizaje y autorregulación [30].

En este enfoque, la autoformación deja de ser un esfuerzo completamente solitario o espontáneo para convertirse en una práctica educativa estructurada, donde el apoyo externo configura las condiciones que permiten que el individuo desarrolle su propio proyecto formativo y consolide su agencia como aprendiz activo en contextos educativos no formales [30].

4.2.3. Brecha educativa y tecnológica

La brecha educativa y tecnológica se refiere a las diferencias en el acceso y aprovechamiento de recursos educativos y tecnológicos entre distintos grupos sociales. Estas desigualdades afectan la calidad del aprendizaje, limitan la participación en entornos digitales y pueden

amplificar la exclusión social. Factores como el acceso desigual a internet, dispositivos digitales y capacitación docente contribuyen a esta brecha, la cual requiere estrategias integrales de inclusión digital y políticas educativas que promuevan la equidad [7, 31].

4.2.4. Tecnología como herramienta de inclusión educativa

La tecnología educativa se ha consolidado como una herramienta estratégica para promover la inclusión educativa, al facilitar el acceso a contenidos y recursos didácticos a estudiantes con diversidad de contextos, habilidades y necesidades. Plataformas digitales, dispositivos móviles y herramientas de aprendizaje asistidas por inteligencia artificial permiten superar barreras geográficas, socioeconómicas y culturales, contribuyendo a mejorar la equidad en la educación [7, 32].

4.3. Fundamentos de Inteligencia Artificial en Educación

La inteligencia artificial (IA), aplicada a la educación, ofrece oportunidades para diseñar entornos de aprendizaje interactivos y personalizados. Una de las estrategias más prometedoras es la implementación de métodos socráticos digitales, donde los sistemas de IA guían a los estudiantes mediante preguntas y diálogos reflexivos, estimulando el pensamiento crítico y la autonomía en la construcción del conocimiento [33, 34].

4.3.1. Inteligencia Artificial aplicada a la educación

La IA en la educación permite automatizar tareas administrativas, ofrecer tutorías personalizadas, monitorear el progreso de los estudiantes y adaptar los contenidos a sus necesidades individuales. Estas aplicaciones han demostrado mejorar la motivación, la eficiencia del aprendizaje y la calidad de la enseñanza, siempre que se acompañen de supervisión pedagógica y criterios éticos claros [12, 13, 19].

4.3.2. Modelos de Lenguaje de Gran Escala

Los modelos de lenguaje de gran escala (LLMs, por sus siglas en inglés) son sistemas de inteligencia artificial entrenados con enormes volúmenes de texto para comprender y generar lenguaje natural. Estos modelos permiten ofrecer respuestas contextualizadas, realizar tutorías personalizadas y asistir en la construcción de conocimiento mediante diálogo interactivo. Su potencial educativo radica en la capacidad de proporcionar retroalimentación inmediata, adaptada al nivel del estudiante, fomentando la reflexión crítica y la autoformación [35, 36].

4.3.3. Arquitectura y funcionamiento de sistemas RAG

La Recuperación Aumentada por Búsqueda (RAG, por sus siglas en inglés) combina modelos de lenguaje (LLM, por sus siglas en inglés) con motores de recuperación de infor-

mación (IR, por sus siglas en inglés), con el fin de proporcionar respuestas fundamentadas en el contenido específico deseado. Esta técnica, introducida por primera vez en 2020 en el artículo *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*, permite la obtención de respuestas fundamentadas en fuentes confiables, que bajo el contexto del sistema promueve la reflexión ética y la resolución de dilemas morales basados en evidencia. RAG amplía las capacidades de tutoría digital al integrar conocimiento externo con generación de lenguaje natural [37, 38].

Arquitectura de RAG

Un sistema RAG opera en dos fases principales:

1. **Fase de indexación:** Los documentos fuente se procesan mediante:

- Segmentación (chunking) en fragmentos semánticamente coherentes.
- Generación de embeddings vectoriales para cada fragmento.
- Almacenamiento en bases de datos vectoriales con metadatos.

Esta fase corresponde con la obtención de conocimiento externo a la implementación del sistema. En el artículo original, la figura 1 muestra cómo este proceso se implementa mediante la combinación de "*memoria paramétrica*" (el LLM utilizado para recibir preguntas y generar respuestas) y la "*memoria no paramétrica*" (índice vectorial del cual se obtiene el contexto para fundamentar la respuesta), con lo cual la generación final obtiene la información requerida para brindar al usuario un resultado fundamentado.

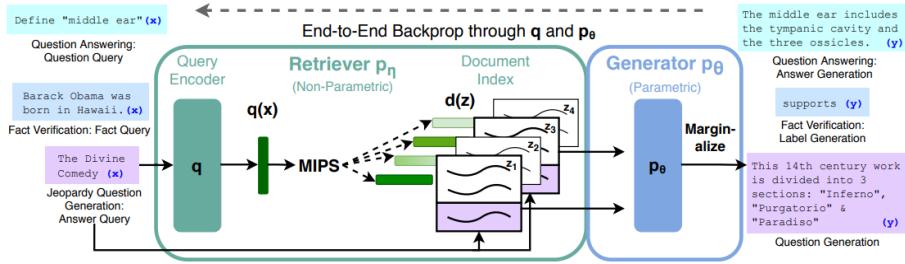


Figura 1: Vista general del enfoque aplicado en el artículo de Lewis et al. [37]

2. **Fase de inferencia:** Al recibir una consulta:

- Se genera un embedding de la pregunta del usuario.
- Se recuperan los K fragmentos más relevantes (típicamente un valor K de entre 3 y 10 elementos), conformando el contexto de la consulta.
- Se construye un prompt contextualizado que combina la pregunta original con los fragmentos recuperados.
- El LLM genera una respuesta basándose exclusivamente en el contexto dado o tomando como guía (el enfoque depende de la estructura utilizada para construir

el *prompt*) de manera que todo resultado se ve anclado a las fuentes verificadas previamente seleccionadas para alimentar el sistema.

Según la implementación, se pueden utilizar distintas estrategias para la combinación de los fragmentos con la generación final. En el documento original se menciona la diferencia de utilizar *RAG sequence* (el modelo selecciona un único documento sobre el cual basará su respuesta) frente a *RAG token* (el modelo genera la respuesta por pasos, seleccionando la fuente a utilizar para cada token independiente, lo que permite combinar más de una fuente).

Estrategias de segmentación

El método de segmentación (*chunking*) seleccionado puede llegar a afectar directamente la calidad de las respuestas obtenidas por el sistema. Las estrategias más comunes incluyen:

- **Segmentación por tamaño fijo:** se divide el texto en fragmentos de longitud uniforme, independiente de su semántica. Presenta la ventaja de que es el tipo de segmentación más fácil de implementar, ya que basta solamente con establecer un límite de palabras y separar todo el texto en dicho límite. Sin embargo, al no evaluar el sentido semántico en cada fragmento, podría dar lugar a rupturas de contexto o pérdida de sentido. [39]
- **Segmentación semántica:** esta segmentación divide el texto respetando unidades de sentido, identificadas mediante signos de puntuación, saltos de línea, identificación de encabezados, secciones, listas, etc. El propósito de esta estrategia es preservar la continuidad de contexto entre cada fragmento. [39]
- **Segmentación recursiva con solapamiento:** esta técnica divide el texto utilizando alguna técnica anterior, con la peculiaridad de incluir al inicio o al final una parte del fragmento contiguo. Es decir, se define un porcentaje de solapamiento que se refiere a qué tanto del fragmento siguiente (o anterior) se incluirá como parte del nuevo fragmento, de manera que se controle de forma explícita la continuidad. [39]

La estrategia de segmentación dependerá de la implementación del sistema RAG que se desea realizar. Se debe tomar en cuenta que la estrategia utilizada afectará directamente la calidad de las respuestas, ya que esto define cómo el LLM obtendrá el contexto del cual se basará para brindar la respuesta a la consulta dada. [39]

Ventajas de RAG en educación

La implementación de sistemas RAG en enfoques educativos, brinda varias ventajas orientadas al uso de modelos de inteligencia artificial:

- **Reduce alucinaciones** al obligar al sistema a utilizar respuestas fundamentadas en el *corpus* definido para el proyecto. [40]

- Permite **actualización del conocimiento** sin reentrenar el modelo; basta solamente con modificar, añadir o eliminar el corpus del proyecto para que el sistema utilice esta nueva información. [40]
- **Facilita trazabilidad** y citación de fuentes, lo cual es particularmente importante en contextos educativos en los que es necesario fundamentar de dónde se obtiene toda la información proporcionada. [40]
- Es apropiado para **dominios especializados** o corpus limitados, como asignaturas o materiales didácticos que no están bien cubiertos en los datos de entrenamiento general del LLM. [40]

Desafíos conocidos

A pesar de sus beneficios, la implementación de RAG también presenta ciertos retos a cubrir en proyectos educativos de alto impacto:

- **Dependencia crítica** de la calidad del corpus. La calidad de las respuestas depende estrictamente de la calidad del contenido educativo utilizado para alimentar el modelo, por lo que si los documentos están mal organizados, contienen errores o están desactualizados; la recuperación de contexto será débil. [41]
- **Riesgo de fragmentación** que rompa la coherencia contextual. Utilizar una estrategia o combinación de estrategias de segmentación inapropiada, puede llevar a que el contexto obtenido por el modelo pierda de sentido, o bien, que una consulta que sí está relacionada con el contenido del corpus no pueda ser respondida. [41]
- **Limitaciones de la ventana de contexto del LLM** a pesar de que el sistema de recuperación diseñado obtenga una base contextual amplia, según el modelo LLM utilizado, generalmente se tiene un límite de tokens permitido para cada consulta, por lo que el prompt construido también cuenta con limitaciones de longitud y, por lo tanto, del nivel de especificación y claridad exigido al modelo. [41]
- Es apropiado para **Dificultad para sintetizar información de múltiples fragmentos dispersos** Al combinar varios fragmentos en un solo *prompt*, si no se ha seleccionado el *corpus* cuidadosamente, se puede incurrir en contradicciones o redundancia en las consultas. [41]

4.3.4. *Embeddings* y representación semántica del texto

Los *embeddings* son representaciones vectoriales de palabras, frases o documentos que capturan sus significados semánticos. Esta técnica permite que los sistemas de IA comparen y recuperen información de manera eficiente, lo que permite medir la similitud entre conceptos y facilita búsquedas semánticas. En educación, los *embeddings* permiten vincular preguntas de los estudiantes con contenidos relevantes, apoyando la personalización del aprendizaje [42, 43].

4.3.5. Bases de datos vectoriales y búsqueda semántica

Las bases de datos vectoriales permiten almacenar y consultar *embeddings* de manera eficiente, habilitando la búsqueda semántica en grandes volúmenes de información. Este enfoque supera las limitaciones de las búsquedas basadas en palabras clave, lo que permite que los estudiantes y sistemas educativos accedan a contenidos relevantes de manera más precisa y contextualizada, facilitando la recuperación de conocimiento en entornos digitales [44, 45].

4.3.6. *Prompt Engineering* y diseño de instrucciones

La disciplina de *Prompt Engineering* consiste en el diseño de instrucciones efectivas para guiar el comportamiento de modelos de lenguaje hacia objetivos específicos. [46] El enfoque principal es brindar al modelo directivas claras con el fin de obtener el resultado final esperado, enfocados en ser tan específicos y directos como el modelo permita.

Componentes de un *prompt* efectivo

- **System prompt:** Define el rol que el asistente debe adoptar ante cada consulta que se le solicite, así como el tono de las respuestas generadas y los límites que debe cumplir (por ejemplo, indicar que debe responder basado solamente en el *corpus* del proyecto e ignorar todo lo externo). [47]
- **Contexto:** Aquí se especifica la base que debe fundamentar todas las respuestas del modelo. En el caso de un sistema RAG, es aquí donde se incluyen todos los fragmentos recuperados de la fuente de datos creada previamente. [47]
- **Instrucción:** Define la tarea específica que se espera que deba cumplir el asistente. Aquí puede ir la pregunta, solicitud de información o generación de contenido multimedia (si aplica). [47]
- **Formato de salida:** Se debe definir también cómo se espera que el modelo responda, ya sea porque se busca obtener una respuesta segmentada en separadores identificables, o bien, para especificar un formato específico. [47]
- **Ejemplos (opcional):** Si se quiere ser aún más explícito sobre cómo se espera que el modelo se comporte, se pueden indicar ejemplos claros del comportamiento que debe tener el modelo a partir de las consultas recibidas. [47]

Estrategias en contextos educativos

Con el objetivo de enfocar el diseño de *prompts* al campo de la educación, se enfatizan prácticas específicas que permiten obtener el flujo de pensamiento del asistente, establecer un tipo de interacción específica con el estudiante (por ejemplo, el uso del método socrático) o solicitar las fuentes utilizadas.

- ***Chain-of-thoutht prompting:*** Solicita al modelo el razonamiento que utilizó para responder la pregunta, se exige el paso a paso de cómo llegó hasta la respuesta brindada. [48]
- ***Socratic prompting:*** Indica al modelo que se debe guiar por el método socrático, el cual consiste en incentivar al usuario a obtener una respuesta final por sí mismo, en lugar de brindar una respuesta directa a la consulta dada.[48]
- ***Constitutional AI:*** Incorpora principios éticos en las instrucciones, por ejemplo, la omisión de palabras o temas sensibles.[48]
- ***Retrieval-aware prompting:*** Exige al modelo citar fuentes en todas sus respuestas. Puede ser útil, aunque también vale la pena analizar si conviene más esta estrategia o simplemente almacenar en los metadatos de los fragmentos los documentos originales.[48]

4.3.7. Tutoría personalizada con IA

La tutoría personalizada con IA permite adaptar los contenidos y las estrategias de enseñanza al nivel, intereses y ritmo de cada estudiante. Los sistemas inteligentes analizan patrones de aprendizaje y ofrecen retroalimentación inmediata, identificando áreas de dificultad y recomendando recursos específicos. Esta personalización mejora la motivación, la retención de conocimiento y promueve la autonomía del aprendiz [34, 49].

4.3.8. Método socrático aplicado a entornos digitales

Los entornos digitales permiten implementar el método socrático mediante sistemas de IA que guían a los estudiantes a través de preguntas reflexivas y secuencias de razonamiento. Esta estrategia fomenta el pensamiento crítico y la autonomía, ya que los alumnos deben analizar, argumentar y evaluar sus propias respuestas antes de recibir retroalimentación. El uso de chatbots y asistentes inteligentes basados en este método facilita un aprendizaje personalizado y continuo, replicando la interacción dialógica propia del enfoque socrático tradicional [34, 50].

4.3.9. Métricas de evaluación de chatbots educativos

La evaluación de asistentes conversacionales educativos requiere métricas más allá de la precisión técnica, incorporando dimensiones pedagógicas incluso si no son herramientas planificadas para su utilización en entornos formales tradicionales de educación. [51]

Métricas de calidad de respuesta

A nivel semántico, es importante identificar aspectos que pueden definir una respuesta dada como exitosa o no, enfocados en la pregunta inicial dada, el *corpus* utilizado para la alimentación del modelo y las instrucciones de tono y comprensión indicadas al modelo. [52]

- **Relevancia:** ¿La respuesta aborda la pregunta planteada?
- **Precisión:** ¿La información es correcta?
- **Compleitud:** ¿Cubre todos los aspectos necesarios?
- **Claridad:** ¿Es comprensible para el público objetivo?

Métricas de desempeño técnico

Por otro lado, a nivel técnico también es útil identificar el desempeño del sistema en cuanto a tiempos de respuesta, integridad de los componentes, coste de operaciones y recursos utilizados.

- **Tasa de éxito** (% de preguntas respondidas apropiadamente)
- **Precisión y Recall** en detección de preguntas fuera de alcance
- **Latencia** (tiempos de respuesta)
- **Consumo de tokens**

4.3.10. Evaluación de calidad de respuestas en sistemas RAG

Por su parte, los sistemas que implementan RAG presentan sus propios desafíos específicos de evaluación relacionados con la recuperación y síntesis de la información almacenada y recuperada. Es necesario identificar si la recuperación semántica fue exitosa, si el almacenamiento de los fragmentos y metadatos se está haciendo correctamente y si el modelo está tomando de referencia los documentos apropiados dentro del *corpus*.

Métricas de recuperación (*Retrieval*)

Para poder obtener estas métricas, las pruebas deben consistir de un conjunto de preguntas previamente analizadas, así como la identificación de qué fragmentos son realmente relevantes para abordar cada pregunta.

- **Precisión@K:** Se refiere al porcentaje de fragmentos obtenidos que corresponden a los fragmentos relevantes identificados para la consulta.
- **Recall@K:** Se refiere al porcentaje de fragmentos relevantes recuperados por el sistema. Es decir, de todos los fragmentos relevantes identificados para la consulta, qué porcentaje fue seleccionado por el modelo.
- **Mean Reciprocal Rank (MRR):** Del top K obtenido por el modelo, en qué posición se sitúa el primer fragmento relevante identificado. Se guía bajo la siguiente ecuación:

$$MRR = \frac{1}{|N|} \sum_{i=1}^{|N|} \frac{1}{K_i} \quad (1)$$

En donde K el número de fragmentos recuperados, i representa cada fragmento relevante y N la posición del fragmento dentro del top- K .

Métricas de generación (*Generation*)

A diferencia de los sistemas puramente extractivos, los sistemas RAG deben ser evaluados tanto por la calidad de la información generada como por su fidelidad al contexto recuperado. Para ello, se emplean métricas que permiten analizar si la respuesta es coherente, pertinente y derivada correctamente del material de referencia.

- **Faithfulness (Fidelidad):** Evalúa si la respuesta se limita a la información contenida en los documentos recuperados, evitando alucinaciones o afirmaciones no sustentadas. [53]
- **Answer Relevance (Relevancia de la respuesta):** Determina el grado en que la respuesta generada responde efectivamente a la pregunta formulada, sin desviarse hacia información tangencial o irrelevante. [54]
- **Context Utilization (Utilización del contexto):** Mide la dependencia efectiva del modelo respecto al contexto recuperado. Una respuesta es de alta calidad cuando demuestra haber utilizado fragmentos relevantes del corpus y no únicamente el conocimiento previo del modelo. [37]
- **Coherencia y fluidez:** Se relaciona con la estructura interna de la respuesta, su legibilidad y consistencia discursiva. En el contexto educativo, la claridad también constituye un indicador pedagógico de calidad. [52]

4.3.11. Validación sin usuarios finales: enfoques y limitaciones

La validación técnica de sistemas educativos sin participación de usuarios finales presenta limitaciones inherentes, pero permite establecer la viabilidad funcional del sistema antes de su despliegue [55]. Estos procedimientos son fundamentales en etapas tempranas del desarrollo.

Enfoques de validación técnica:

1. **Validación basada en corpus:** Verifica que las respuestas provengan de los materiales fuente y que no contengan información ajena al contenido educativo.
2. **Validación por expertos:** Permite asegurar la pertinencia pedagógica y detectar errores conceptuales o éticos en las respuestas.
3. **Pruebas de estrés:** Incluyen preguntas fuera de alcance, ambiguas o sensibles para evaluar la resiliencia del modelo.
4. **Simulación de usuarios:** Reproduce escenarios conversacionales con perfiles variados, como estudiantes de diferentes edades o niveles educativos. [33, 49, 55]

Limitaciones reconocidas:

- No mide el impacto pedagógico real ni el aprendizaje logrado.
- La validación manual puede estar sujeta a sesgos del evaluador.
- Los escenarios simulados no reflejan completamente la complejidad del uso real.
- No existe retroalimentación iterativa basada en la experiencia de usuarios finales. [33, 49, 55]

Implicaciones:

Los resultados obtenidos de esta validación deben interpretarse como una demostración de factibilidad técnica y coherencia funcional, pero no como evidencia de efectividad educativa. Estudios posteriores con usuarios reales serán necesarios para medir impacto pedagógico y aceptación. [33]

4.3.12. Congruencia y fundamentación en respuestas educativas

En contextos educativos, además de la precisión técnica, se espera que el sistema responda de forma congruente con las expectativas del rol asignado (alumno, tutor o asistente). La **congruencia conversacional** se define como el grado en que el sistema actúa según lo esperado ante cada tipo de pregunta: responder apropiadamente cuando debe hacerlo y abstenerse en situaciones fuera de alcance [56].

Este tipo de métrica puede calcularse empíricamente como el porcentaje de interacciones en las que el modelo actúa conforme al comportamiento esperado. En el presente proyecto, dicha métrica complementa las tradicionales de precisión y fidelidad, proporcionando una visión integral de la calidad del comportamiento conversacional [56].

4.4. Ética y Responsabilidad en IA Educativa

La ética en IA educativa aborda la responsabilidad en el diseño, implementación y uso de sistemas inteligentes en contextos de aprendizaje. Incluye consideraciones sobre privacidad de los datos, equidad, transparencia, inclusión e impacto social. Garantizar que los estudiantes sean tratados de manera justa y que los sistemas no reproduzcan sesgos existentes es crucial para la confianza y efectividad de la educación asistida por IA [57, 58].

4.4.1. Principios éticos fundamentales en IA

Los principios éticos fundamentales en IA incluyen transparencia, justicia, no discriminación, responsabilidad, privacidad y seguridad. En el ámbito educativo, estos principios guían el desarrollo de sistemas que respeten la dignidad del estudiante, promuevan equidad en el

aprendizaje y faciliten la rendición de cuentas por parte de desarrolladores y educadores. La aplicación de estos principios permite aprovechar el potencial de la IA sin comprometer la integridad pedagógica [59, 60].

4.4.2. Sesgos algorítmicos y culturales en contextos latinoamericanos

La prevención de sesgos algorítmicos se centra en garantizar que los sistemas de IA no reproduzcan ni amplifiquen desigualdades existentes en la educación. Esto implica analizar los datos de entrenamiento, identificar posibles sesgos y aplicar técnicas de mitigación, como ajuste de ponderaciones, diversificación de datasets y pruebas de equidad en los resultados. La prevención de sesgos asegura que todos los estudiantes reciban oportunidades de aprendizaje justas y equitativas [61, 62].

4.4.3. Transparencia y explicabilidad en sistemas inteligentes

La transparencia y explicabilidad son fundamentales para que docentes, estudiantes y desarrolladores comprendan cómo un sistema de IA toma decisiones. Esto incluye técnicas de interpretabilidad que permitan visualizar la lógica de los modelos y justificar las recomendaciones que generan. En educación, la explicabilidad ayuda a confiar en las decisiones automatizadas, facilita la supervisión pedagógica y permite detectar errores o sesgos [63, 64].

4.4.4. Responsabilidad ante respuestas incorrectas o inadecuadas

La responsabilidad en sistemas de IA educativa implica definir con claridad los mecanismos para abordar errores, recomendaciones inadecuadas o información potencialmente nociva generada por los algoritmos. Cuando un sistema automatizado produce contenidos incorrectos, los efectos pueden ser especialmente sensibles en contextos educativos, ya que influyen directamente en el aprendizaje, la motivación y las decisiones académicas de los estudiantes [59, 60].

Esta responsabilidad recae tanto en los desarrolladores, quienes deben implementar modelos seguros, mecanismos de verificación y pruebas continuas, como en los docentes y las instituciones que integran la tecnología. Esto incluye ofrecer rutas de corrección, permitir retroalimentación humana y asegurar canales claros para reportar fallos. De esta manera, la IA se integra como una herramienta asistiva bajo supervisión profesional, en lugar de delegar completamente la evaluación y orientación pedagógica [59, 60].

4.4.5. Privacidad y seguridad en aplicaciones educativas para menores

El uso de aplicaciones educativas basadas en IA en contextos escolares requiere un enfoque riguroso de protección de datos, especialmente cuando se trata de menores de edad. La información académica, conductual y biométrica recopilada por estos sistemas consti-

tuye un activo sensible que debe ser gestionado bajo principios de minimización de datos, consentimiento informado y almacenamiento seguro [57, 58].

Organismos internacionales han enfatizado la importancia de resguardar la identidad digital de los estudiantes, evitar usos secundarios no autorizados y garantizar que los datos no se utilicen para prácticas discriminatorias o comerciales. Las instituciones tienen la responsabilidad de establecer políticas claras de acceso, supervisar proveedores tecnológicos y aplicar estándares robustos de ciberseguridad. La protección de los datos de menores no solo es una obligación legal y ética, sino también una condición necesaria para preservar la confianza en entornos educativos mediados por IA [57, 58].

4.4.6. Supervisión pedagógica en sistemas automatizados

A pesar de la autonomía de los sistemas de IA, la supervisión pedagógica es esencial para garantizar la calidad del aprendizaje. Docentes y tutores deben monitorear el funcionamiento de los sistemas automatizados, evaluar la relevancia y exactitud de las respuestas generadas, y ajustar los parámetros de personalización según las necesidades de los estudiantes. Este enfoque mixto asegura que la tecnología complemente, y no reemplace, la guía educativa [33, 55].

4.5. Aprendizaje Móvil en Contextos de Recursos Limitados

El aprendizaje móvil (*mobile learning* o m-learning) se ha convertido en un medio clave para ampliar el acceso a experiencias educativas, especialmente en regiones donde las limitaciones tecnológicas, de infraestructura o económicas dificultan el aprendizaje tradicional. En contextos con recursos limitados, los dispositivos móviles permiten superar barreras geográficas y temporales, democratizando oportunidades de acceso a información, formación técnica y herramientas de apoyo educativo [65, 66].

Sin embargo, la implementación efectiva del aprendizaje móvil requiere considerar retos como la disponibilidad de dispositivos, la alfabetización digital de los usuarios, los costos de conectividad y las brechas de infraestructura. El diseño de soluciones educativas móviles sostenibles debe responder a estos factores para garantizar accesibilidad, pertinencia cultural y equidad tecnológica. [66]

4.5.1. Panorama del aprendizaje móvil en Guatemala y Centroamérica

El crecimiento del aprendizaje móvil en Guatemala y Centroamérica ha sido gradual pero progresivo, impulsado por iniciativas de digitalización, comunidades tecnológicas emergentes y el interés institucional por modernizar procesos educativos y productivos. Según el BID, la región ha avanzado significativamente en adopción tecnológica, pero aún enfrenta brechas en cuanto a infraestructura digital, capacidad de investigación e inversión en innovación [3].

4.5.2. Diseño de experiencias móviles para usuarios con baja alfabetización digital

El diseño de experiencias educativas móviles para usuarios con baja alfabetización digital requiere estrategias centradas en usabilidad, simplicidad y acompañamiento formativo. UNESCO destaca que las interfaces visuales claras, los flujos guiados y los recursos multimedia accesibles pueden favorecer la participación de usuarios principiantes [67].

4.5.3. Consideraciones de conectividad intermitente y consumo de datos

En muchos contextos latinoamericanos, incluidos sectores rurales de Guatemala, el acceso a Internet es costoso e intermitente. Por ello, las aplicaciones educativas móviles deben optimizar el consumo de datos, ofrecer funcionalidad fuera de línea y emplear técnicas de sincronización diferida para resguardar el progreso del usuario cuando no haya conexión [66, 68, 69].

Prácticas recomendadas incluyen compresión de recursos multimedia, almacenamiento local temporal, caching inteligente y utilización de formatos eficientes. La capacidad de operar con conectividad limitada no solo reduce barreras de acceso, sino que también mejora la adopción sostenida de herramientas educativas en zonas marginadas [70].

4.5.4. Aplicaciones móviles en la educación

Las aplicaciones móviles educativas permiten acceder a recursos y experiencias de aprendizaje en cualquier momento y lugar. Integradas con IA, estas apps pueden ofrecer tutorías personalizadas, seguimiento del progreso, retroalimentación inmediata y gamificación del aprendizaje. Su portabilidad y accesibilidad contribuyen a reducir la brecha educativa y facilitan la inclusión digital [71, 72].

4.6. Tecnologías de Implementación

El uso responsable de IA en educación implica enseñar a los estudiantes a utilizar herramientas inteligentes sin vulnerar normas éticas ni académicas. Esto incluye fomentar la autoría propia, la citación adecuada de fuentes y el desarrollo de habilidades de pensamiento crítico para interpretar la información generada por la IA. La integridad académica asegura que la tecnología complementa el aprendizaje sin reemplazar la reflexión y el esfuerzo personal [73, 74].

4.6.1. Arquitectura cliente-servidor (*frontend/backend*)

La arquitectura cliente-servidor es un modelo de diseño de software en el que el cliente (por ejemplo, una app móvil o navegador web) solicita servicios al servidor, el cual procesa

la información, ejecuta lógica de negocio y responde con datos. En educación digital, esta arquitectura permite centralizar recursos educativos, gestionar bases de datos y ofrecer aplicaciones interactivas seguras y escalables. El *frontend* se encarga de la interfaz y la experiencia de usuario, mientras que el *backend* gestiona la lógica, la seguridad y la integración con IA y bases de datos [75, 76].

Patrón de diseño MVC

El patrón de diseño *Model-View-Controller* (MVC) es una arquitectura ampliamente utilizada para la construcción de interfaces de usuario. Su objetivo principal es separar la representación de la información de la lógica que la gestiona, promoviendo modularidad, reutilización y facilidad de mantenimiento. [77]

En este patrón, el **Model** contiene los datos y la lógica de negocio; la **View** es responsable de mostrar la información al usuario; y el **Controller** actúa como intermediario, recibiendo entradas del usuario y coordinando las actualizaciones entre el modelo y la vista. Esta separación permite que cambios en la interfaz no afecten directamente a la lógica del sistema y viceversa. [77]

El patrón MVC tuvo sus orígenes en el entorno *Smalltalk-80* y ha sido ampliamente adoptado en múltiples *frameworks* modernos de desarrollo tanto web como de escritorio debido a su capacidad de estructurar aplicaciones complejas de forma eficiente [77].

4.6.2. Frameworks de desarrollo móvil: Kotlin y ecosistema Android

Kotlin es un lenguaje de programación moderno y seguro que se utiliza para el desarrollo de aplicaciones Android. Presenta características como tipado estático, interoperabilidad con Java, sintaxis concisa y soporte nativo en Android Studio. Su uso permite crear aplicaciones robustas, escalables y fáciles de mantener, integrando librerías modernas y *frameworks* de IA para educación digital [78, 79].

Patrón de diseño MVVM

El patrón *Model-View-ViewModel* (MVVM) es una arquitectura de software que separa de forma clara la lógica de negocio de la interfaz de usuario, promoviendo el desacoplamiento y facilitando la mantención del código. [80]

En este patrón, el **Model** encapsula los datos y reglas de negocio; la **View** representa la interfaz de usuario; y el **ViewModel** actúa como un intermediario que gestiona el estado de la vista, expone datos al usuario y maneja la lógica de presentación. La comunicación suele realizarse mediante mecanismos de enlace de datos (*data binding*), lo que permite que la interfaz se actualice automáticamente ante cambios en los datos. [80]

4.6.3. Bases de datos vectoriales y su contraste con bases relationales

Las bases de datos vectoriales almacenan representaciones numéricas (*embeddings*) de información, permitiendo búsquedas semánticas rápidas y precisas. En cambio, las bases de datos relationales organizan información en tablas con relaciones explícitas y consultas estructuradas. Para educación digital basada en IA, las bases vectoriales permiten recuperar contenido relevante según el significado, mientras que las relationales son útiles para gestión de usuarios, cursos y registros administrativos. Integrar ambos tipos optimiza tanto la eficiencia semántica como la consistencia estructural de los datos [44, 81].

4.6.4. APIs de IA: integración de modelos conversacionales

Las APIs de IA, como *OpenAI* y *Gemini*, permiten integrar modelos de lenguaje conversacionales en aplicaciones educativas. Estos servicios ofrecen capacidades de generación de texto, comprensión de lenguaje natural y tutoría personalizada, facilitando la interacción del estudiante con sistemas de IA. La integración se realiza mediante solicitudes a la API, manejo de *tokens* y adaptación de respuestas al contexto educativo, permitiendo desarrollar tutores digitales eficientes y éticos [82, 83].

CAPÍTULO 5

Metodología

El desarrollo del proyecto *Ciudadano Digital* se llevó a cabo bajo el marco de trabajo SCRUM, un enfoque ágil ampliamente utilizado en ingeniería de software que permite la entrega incremental de productos funcionales mediante ciclos cortos de desarrollo denominados *sprints*. Esta metodología fue seleccionada debido a su flexibilidad, capacidad de adaptación a cambios en los requerimientos y enfoque en la mejora continua, elementos clave en un proyecto de innovación educativa..

A lo largo del proceso, se definieron seis *sprints* principales, cada uno con objetivos concretos y entregables verificables, orientados a la obtención progresiva de un prototipo funcional y validado de la aplicación. Cada *sprint* tuvo una duración de entre tres y cuatro semanas, ajustándose según la complejidad técnica y la carga académica del periodo correspondiente.

Cada ciclo SCRUM siguió las fases de planificación, desarrollo, revisión y retrospectiva, bajo los siguientes principios:

- Planificación (*Sprint Planning*): se definieron los objetivos y alcance del *sprint*, así como las tareas específicas necesarias para cumplir la meta establecida.
- Desarrollo (*Sprint Execution*): se ejecutaron las tareas asignadas con enfoque en la funcionalidad incremental, priorizando siempre la obtención de resultados medibles.
- Revisión (*Sprint Review*): al cierre de cada *sprint*, se evaluó el cumplimiento de los objetivos, la calidad del producto obtenido y la satisfacción de los criterios de aceptación definidos.
- Retrospectiva (*Sprint Retrospective*): se analizaron los aprendizajes obtenidos, los obstáculos encontrados y las oportunidades de mejora para el siguiente *sprint*.

El enfoque SCRUM permitió mantener un flujo de trabajo iterativo, controlado y adaptable, asegurando que cada componente técnico se ejecutara tomando como enfoque la experiencia esperada del usuario objetivo. En este caso, el usuario fue representado a través de

una *Persona* desarrollada con base en un proceso de investigación y perfilamiento descrito en el primer *sprint*.

A partir del segundo *sprint*, los entregables se enfocaron en la construcción progresiva del sistema técnico, desde la recopilación y procesamiento de contenido educativo, hasta la implementación del *backend*, el desarrollo de la interfaz móvil y las fases finales de validación y documentación.

El producto mínimo viable (MVP, por sus siglas en inglés) obtenido al finalizar el último *sprint* constituye una versión funcional del asistente inteligente de educación ciudadana, capaz de interactuar con el usuario, contextualizar sus preguntas y generar respuestas basadas en la información previamente curada y vectorizada.

5.1. Enfoque metodológico aplicado al contexto del proyecto

A diferencia de proyectos puramente técnicos, *Ciudadano Digital* combina aspectos de ingeniería de software, inteligencia artificial y educación en valores. Por ello, la aplicación de SCRUM fue adaptada a un enfoque sociotecnológico, que no solo prioriza la funcionalidad del sistema, sino también la pertinencia ética y pedagógica del contenido.

En cada *sprint*, se incluyeron tareas de análisis cualitativo y cuantitativo relacionadas con el perfil del usuario objetivo: estudiantes de nivel medio en Guatemala, de entre 14 y 20 años, con acceso limitado a formación cívica más allá del aula formal. Este enfoque garantizó que las decisiones técnicas (estructura del *backend*, procesamiento de datos, interfaz y validación) respondieran a necesidades reales detectadas en el público meta.

Así, el proceso metodológico buscó alinear el desarrollo tecnológico con la misión educativa del proyecto, entendiendo que la calidad del producto no se mide solo por su rendimiento, sino también por su capacidad de promover la reflexión moral y la ciudadanía responsable en contextos informales de aprendizaje.

5.1.1. Estructura de los *Sprints*

La fase metodológica del desarrollo del proyecto estuvo compuesta por seis *sprints*, cada uno con un enfoque y duración específicos (Cuadro 1). El propósito de esta organización fue facilitar la gestión del proyecto, permitiendo una entrega progresiva de resultados y la incorporación de retroalimentación continua, la cual se llevó a cabo al finalizar cada *sprint* evaluando si se alcanzó el objetivo establecido para el mismo a través de las tareas completadas.

Sprint	Meta Principal	Duración Estimada
<i>Sprint 1</i>	Identificación del perfil de usuario objetivo (Persona).	3 semanas
<i>Sprint 2</i>	Recolección y procesamiento del contenido educativo.	4 semanas
<i>Sprint 3</i>	Construcción e implementación del <i>backend</i> .	4 semanas
<i>Sprint 4</i>	Desarrollo de la interfaz móvil en Kotlin.	4 semanas
<i>Sprint 5</i>	Pruebas y validación funcional.	3 semanas
<i>Sprint 6</i>	Documentación, presentación y cierre del proyecto.	3 semanas

Cuadro 1: Estructura de los *sprints* del proyecto, incluyendo la meta principal y duración estimada de cada uno.

Al final, cada *sprint* culminó con un entregable verificable que sirvió como criterio de avance para el siguiente ciclo, asegurando así la trazabilidad y coherencia entre la visión inicial del proyecto y el producto final obtenido.

5.2. *Sprint 1: Identificación del perfil de usuario objetivo*

Duración estimada: 3 semanas

Este *sprint* tuvo como objetivo desarrollar un perfil de usuario (Persona) que sirviera como insumo accionable para orientar las decisiones de diseño interactivo y priorización técnica del proyecto. Dado que no fue posible realizar entrevistas ni trabajo de campo, el perfil se elaboró exclusivamente a partir del análisis de fuentes documentales que reflejan la situación actual de los estudiantes en el país, considerando aspectos demográficos, académicos y sociales. Con base en esta información, se construyó una ficha de *Persona* completa, acompañada de criterios de diseño alineados con las necesidades y características identificadas.

5.2.1. Ejecución

La culminación del *sprint* se evaluó tomando en cuenta la culminación exitosa de las siguientes tareas:

1. Investigación documental

- Revisión de informes académicos y/o gubernamentales sobre educación ciudadana, competencias cívicas y valores en jóvenes guatemaltecos.
- Consulta de programas educativos oficiales, como el Currículo Nacional Base (CNB) y materiales de formación en valores del Ministerio de Educación de Guatemala, así como contenido internacional enfocado en brindar una educación más completa.

- Análisis de estudios internacionales de organismos como UNESCO (Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura), CEPAL (Comisión Económica para América Latina y el Caribe), CIEN (Centro de Investigaciones Económicas Nacionales) y BID (Banco Interamericano de Desarrollo) sobre hábitos digitales, desigualdad educativa y desarrollo de competencias ciudadanas en adolescentes y jóvenes.

2. Análisis e interpretación de la información

- Sistematización de datos demográficos, educativos y tecnológicos relevantes para el contexto juvenil guatemalteco.
- Identificación de patrones generales de comportamiento, motivaciones, frustraciones y objetivos (enfocados en aspiraciones cívicas), a partir de tendencias reportadas en las fuentes analizadas.
- Construcción de categorías de análisis que permitieran traducir los hallazgos documentales en insumos para el diseño centrado en el usuario.

3. Definición del perfil Persona

- Elaboración de una ficha de usuario basada en la interpretación crítica de los datos documentales, con los siguientes componentes:
 - **Perfil base:** edad estimada, nivel educativo, ubicación, etnia, acceso tecnológico y contexto social.
 - **Motivaciones:** interés por la participación comunitaria y el aprendizaje de ciudadanía.
 - **Frustraciones:** barreras de acceso a recursos educativos y desconfianza en la calidad o adecuación de los materiales disponibles.
 - **Objetivos:** qué quisiera conseguir el usuario a través de sus motivaciones y frustraciones, bajo el contexto de educación en valores y formación ciudadana.
 - **Consideraciones especiales:** limitaciones de conectividad, recursos económicos y brechas culturales.
- Producción de una ficha visual que sirviera como base para las decisiones de diseño en *sprints* posteriores.

4. Documentación de criterios de diseño

- Derivación de recomendaciones de diseño UX basadas en el perfil construido: tono comunicativo, estructura de funciones, rol a asumir por el asistente, y adaptabilidad tecnológica.
- Identificación de necesidades prioritarias que el asistente debe ser capaz de abordar a través de la interacción pregunta-respuesta.

5.2.2. Resultado final

Como resultado de este primer *sprint*, se construyó un perfil de *Persona* detallado, basado en fuentes documentales, que permitió comprender las necesidades, barreras y expectativas del usuario objetivo frente a una herramienta de apoyo educativo.

- Edad promedio: 14-20 años.
- Contexto educativo: estudiantes de nivel medio y universitario inicial.
- Motivaciones: aprender de forma práctica y reflexiva, mejorar su comprensión de ciudadanía y valores.
- Frustraciones: enseñanza teórica, falta de espacios de diálogo y escasez de herramientas interactivas.
- Competencias digitales: nivel bajo a medio en uso de aplicaciones y herramientas digitales.
- Contexto de uso de la aplicación: dispositivos móviles, principalmente Android, con sesiones cortas de interacción y preferencia por contenidos dinámicos y cercanos a su realidad.

Este perfil se utilizó como base para orientar el diseño conversacional, las estrategias de análisis documental y los lineamientos pedagógicos que guiarán las siguientes etapas del desarrollo del proyecto.

5.3. *Sprint 2: Recolección y procesamiento del contenido educativo inicial*

Duración estimada: 4 semanas

Este *sprint* se centró en recopilar, procesar y estructurar el contenido educativo inicial que alimentará al asistente virtual de inteligencia artificial, con la finalidad de garantizar que el sistema pueda generar respuestas precisas y contextualizadas sobre formación ciudadana y valores morales, basándose en información confiable y organizada de manera semántica. Se combinó la selección documental, curación de contenido, segmentación temática y almacenamiento vectorial de manera sistemática, asegurando la trazabilidad y calidad de los datos utilizados.

5.3.1. Ejecución

Para cumplir el objetivo se desarrolló un proceso sistemático dividido en cuatro etapas principales: selección documental, curación, segmentación temática, y vectorización (a través de *OpenAI*) con almacenamiento en *Pinecone*. Este flujo se diseñó de forma reproducible para permitir futuras ampliaciones o actualizaciones del *corpus* de información.

1. Selección documental

- **Identificación de fuentes oficiales y confiables:** se recopilaron documentos emitidos por el Ministerio de Educación de Guatemala, tales como contenidos contemplados en el *Curriculum Nacional Base (CNB)* para los grados educativos

abarcados por el rango de edad establecido, así como guías orientacionales dirigidas a los educadores, con el fin de que el asistente también tenga conocimiento de cómo interactuar con los usuarios objetivo de forma correcta.

- **Revisión de fuentes internacionales:** se incorporaron publicaciones y estudios de entidades internacionales como la OEA (Organización de Estados Americanos), universidades extranjeras o la UNESCO (Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura) los cuales buscan alimentar aún más el conocimiento teórico del asistente, así como diversificar las fuentes de información a contextos internacionales, con lo que la herramienta no cuenta con sesgos específicos para tratar temas exclusivos de un área específica, sino que busca contar con una base sólida que aplique en la mayoría de situaciones.
- **Estudios complementarios y casos prácticos:** Además de las fuentes mencionadas, también se incluye la recopilación de libros educativos de entidades independientes (tales como IGER, Instituto Guatemalteco de Educación Radiónica) así como de autores externos. También se prioriza la obtención de casos prácticos resueltos, de manera que el asistente tenga una base contextual de cómo actuar ante situaciones en las que sea necesaria la aplicación de conceptos relacionados con formación ciudadana y valores morales.
- **Registro de metadatos:** cada documento fue almacenado en un contenedor tipo **Amazon S3** (*Simple Storage Service*), mientras que sus metadatos asociados (título, autor, año de publicación) así como la ruta de almacenamiento relativa dentro del contenedor, fueron almacenados en la base de datos vectorial, siguiendo el siguiente esquema:
 - Identificador único (**Document ID**)
 - Identificador del usuario que sube el documento (**User ID**)
 - Título del documento (**Title**)
 - Fuente o Autor (**Author**)
 - Año de publicación (**Year**)
 - Categoría temática (**Category**)
 - Ruta dentro del contenedor S3 (**Document URL**)

Este registro garantiza trazabilidad desde la fuente original hasta el fragmento vectorizado.

2. Curación y digitalización

- **Conversión de documentos:** los archivos se transformaron a texto plano (**.txt**) con codificación UTF-8 mediante herramientas como **NFKD** o **Tesseract OCR** (esta última extrae de forma automática el texto reconocible de imágenes o documentos PDF escaneados).
- **Limpieza y normalización:** se eliminaron saltos de línea innecesarios, espacios vacíos múltiples y caracteres especiales, para lo cual se aplicó el proceso descrito en el Algoritmo 1.

Algoritmo 1 Proceso de limpieza y normalización profunda de texto

```
1: procedure LIMPIARTEXTO(texto)
2:   texto ← NormalizarUnicode(texto, “NFKD”)
3:   texto ← EliminarCaracteresNoASCII(texto)
4:   texto ← Reemplazar(texto, {“\r”, “\n”, “\t”}, “ ”)
5:   texto ← EliminarCaracteresEspeciales(texto, “manteniendo letras, números y pun-
tuación básica”)
6:   texto ← ReemplazarMúltiplesEspaciosPorUno(texto)
7:   texto ← EliminarEspaciosExtremos(texto)
8:   return texto
9: end procedure
```

- **Estandarización de formato:** se uniformaron títulos y subtítulos con reglas jerárquicas para facilitar la segmentación automática, como se muestra en el Algoritmo 2.

Algoritmo 2 Estandarización de títulos y numeración en texto

```
1: procedure ESTANDARIZARFORMATO(texto)
2:   texto ← ReemplazarMarkdownConTitulo(texto)
3:   texto ← ReemplazarNumeracionConTitulo(texto)
4:   texto ← ConvertirTitulosMayusculas(texto)
5:   texto ← UniformarNumeracion(texto)
6:   texto ← EliminarEspaciosExtremos(texto)
7:   return texto
8: end procedure
```

- **Validación de integridad:** se verificó que los textos conservaran coherencia y completitud, eliminando duplicados o secciones ilegibles, siguiendo el flujo del Algoritmo 3.

Algoritmo 3 Validación de integridad de texto

```
1: procedure VALIDARINTEGRIDAD(texto)
2:   líneas ← DividirEnLineas(texto)
3:   líneas_limpias ← ListaVacía()
4:   for cada línea en líneas do
5:     if Longitud(Trim(linea)) < 3 then
6:       Continuar
7:     end if
8:     caracteres_válidos ← ContarCaracteresAlfanumericosYEspacios(linea)
9:     if caracteres_válidos / Max(Longitud(linea), 1) > 0.6 then
10:      Añadir(linea, líneas_limpias)
11:    end if
12:  end for
13:  líneas_sin_duplicados ← EliminarDuplicados(líneas_limpias)
14:  texto_limpio ← UnirLineas(líneas_sin_duplicados)
15:  return texto_limpio
16: end procedure
```

3. Segmentación temática

- **Diseño del esquema de categorías:** se definieron seis temas guía iniciales: *ética y moral, participación ciudadana, derechos humanos, convivencia y respeto, responsabilidad social y cultura digital*. Sin embargo, esta lista puede incrementar con el tiempo, a medida que el modelo procese una mayor cantidad de archivos y no sea capaz de incluirlos en una de las categorías predefinidas.
- **División en fragmentos:** los textos fueron segmentados automáticamente en bloques de 20 a 150 palabras, conservando coherencia semántica.
- **Etiquetado y registro:** cada fragmento se asoció a una categoría temática, con lo cual la metadata de cada fragmento quedó estructurada de la siguiente manera: **document_id** (identificador único del documento en la base de datos relacional), **text** (contenido original del documento), **source** (título original del documento), **author** (autor o institución que realizó el documento), **year** (año de publicación del documento original), **category** (categoría temática asociada al fragmento), **sha1** (*hash* único del fragmento, para evitar duplicados), **uploaded_at** (fecha y hora de publicación del fragmento, en formato ISO).

4. Vectorización y almacenamiento en *Pinecone*

- **Generación de embeddings:** cada fragmento fue procesado con el modelo *text-embedding-3-small* de OpenAI, generando vectores de 1536 dimensiones.
- **Normalización final:** se verificó la unicidad de cada **fragment_id** y la ausencia de duplicados o errores de codificación.
- **Creación del índice vectorial:** se configuró un índice en Pinecone con los parámetros:
 - **name** = “ciudadano-digital”
 - **namespace** = “ciudadania”
 - **metric** = “cosine”

- dimension = 1536
- **Inserción de vectores:** los embeddings se insertaron junto con sus metadatos (fuente, categoría, bloque, documento de origen) para permitir consultas semánticas eficientes. El proceso completo se muestra en el Algoritmo 4.

Algoritmo 4 Vectorización de fragmentos

```

1: procedure VECTORIZARFRAGMENTOS(fragmentos, identificador, fuente, autor, año,
   BATCH_SIZE)
2:   lote ← ListaVacía()
3:   for cada frag en fragmentos do
4:     sha1 ← CalcularSHA1(frag)
5:     if FragmentoYaIndexado(sha1) or EsVacio(frag) then
6:       Continuar
7:     end if
8:     categoria ← ClasificarCategoria(frag)
9:     embedding ← GenerarEmbedding(modelo=“text-embedding-3-small”, tex-
   to=frag)
10:    metadatos ← CrearDiccionario({ “document_id”: identificador, “text”: frag, “sour-
      ce”: fuente, “author”: autor, “year”: año, “category”: categoria, “sha1”: sha1, “uploaded_-
      at”: FechaHoraActual() })
11:    AñadirAlLote(lote, CrearVector(id=GenerarUUID(), valores=embedding, meta-
      datos=metadatos))
12:    if Longitud(lote) ≥ BATCH_SIZE then
13:      RegistrarLote(lote, namespace=“ciudadanía”)
14:      lote ← ListaVacía()
15:    end if
16:   end for
17:   if lote no vacío then
18:     RegistrarLote(lote)
19:   end if
20: end procedure

```

- **Control de calidad:** se revisó manualmente una muestra del 10 % de los fragmentos para validar su correcta asignación temática y coherencia contextual.

5.3.2. Resultado final

Al finalizar el *Sprint 2*, se obtuvo:

- Una base documental curada y segmentada en categorías temáticas.
- *Embeddings* generados para cada fragmento de texto, con metadatos completos para garantizar trazabilidad.
- Un índice en *Pinecone* listo para consultas semánticas, capaz de proporcionar contexto preciso al asistente virtual para cualquier pregunta del usuario.

- Establecimiento de un flujo reproducible de selección, curación, segmentación y vectorización de contenido para una continua actualización del *corpus* del proyecto.

Este *sprint* permitió sentar las bases para un sistema de respuesta contextualizada, alineado con los objetivos de formación ciudadana y valores morales definidos en el proyecto, asegurando que el asistente virtual cuente con información confiable, organizada y accesible para generar respuestas pertinentes y fundamentadas.

5.4. *Sprint 3: Construcción e implementación del backend*

Duración estimada: 4 semanas

Este *sprint* se enfocó en el diseño, construcción e implementación de la arquitectura *backend* del asistente virtual, garantizando la integración de bases de datos relacionales y vectoriales, y estableciendo la comunicación segura y eficiente con el modelo de lenguaje (LLM, por sus siglas en inglés) mediante un flujo RAG (*Retrieval-Augmented Generation*). Se definieron módulos claros bajo el patrón de diseño MVC (Modelo-Vista-Controlador), así como servicios complementarios internos en *Python* tanto para la curación y procesamiento de documentos, como para procesar consultas y generar respuestas contextualizadas con base en los mismos.

5.4.1. Ejecución

1. Diseño de arquitectura

- Se adoptó el patrón de diseño **Modelo–Vista–Controlador (MVC)**, siguiendo la estructura de la Figura 2. Esta arquitectura permite separar las responsabilidades del sistema, facilitando el mantenimiento y escalabilidad.

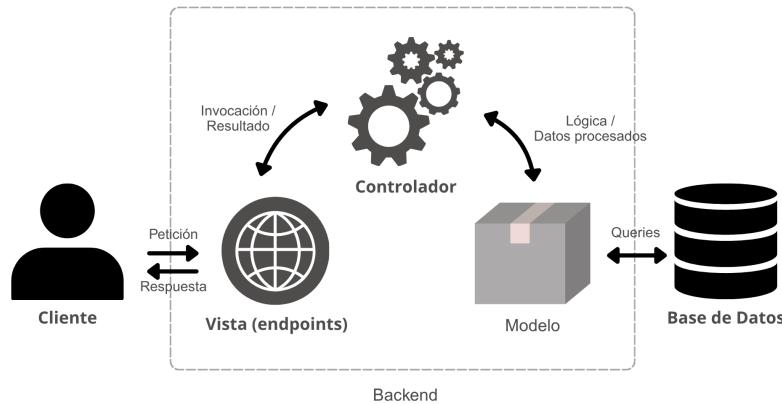


Figura 2: Arquitectura MVC (Modelo, Vista, Controlador).

- **Modelos:**

- Representan entidades del sistema: usuarios, chats, mensajes, sesiones, documentos, categorías.
- Cada modelo incluye operaciones CRUD (*Create, Read, Update, Delete*) que se ejecutan directamente sobre la base de datos, permitiendo la creación, consulta, actualización y eliminación de registros, según sea requerido.
- A través de la comunicación con los controladores, los modelos gestionan la persistencia y recuperación de datos de manera eficiente, siguiendo la lógica de negocio definida.
- **Rutas (Vistas):** Endpoints REST para interacción con el usuario, como registro, login, listado de chats, envío de preguntas, entre otros.
- **Controladores:** Gestionan la lógica de negocio: validación de datos, comunicación con modelos, manejo de errores y generación de respuestas. El Algoritmo 5 ilustra el flujo básico de un controlador típico.

Algoritmo 5 Controlador de solicitudes

```

1: procedure CONTROLADOR(request)
2:   Validar(request.datos)
3:   resultado ← modelo.operacion(request)
4:   devolver(resultado)
5: end procedure

```

- **Módulos auxiliares:**

- **Middlewares:** validan autenticación y seguridad antes de pasar al controlador. El Algoritmo 6 muestra un ejemplo de middleware para validar tokens JWT.

Algoritmo 6 Validación de token

```

1: procedure VALIDARTOKENREQUEST(request)
2:   if not ValidarToken(request.token) then
3:     DevolverError(401, "Token inválido")
4:   else
5:     Continuar(request)
6:   end if
7: end procedure

```

- **Helpers:** funciones reutilizables como:
 - encriptarContraseña(contraseña)
 - generarToken(usuarioID)
 - formatearFecha(fecha)

2. Diseño y construcción de bases de datos

Base de datos relacional

Esta será la encargada de almacenar la información estructurada del sistema, como usuarios, chats, mensajes y sesiones, de manera que se mantenga la persistencia de

datos y se facilite la gestión de las interacciones del usuario con el asistente virtual. La Figura 3 muestra el diagrama entidad-relación (ER) de la base de datos relacional diseñada.

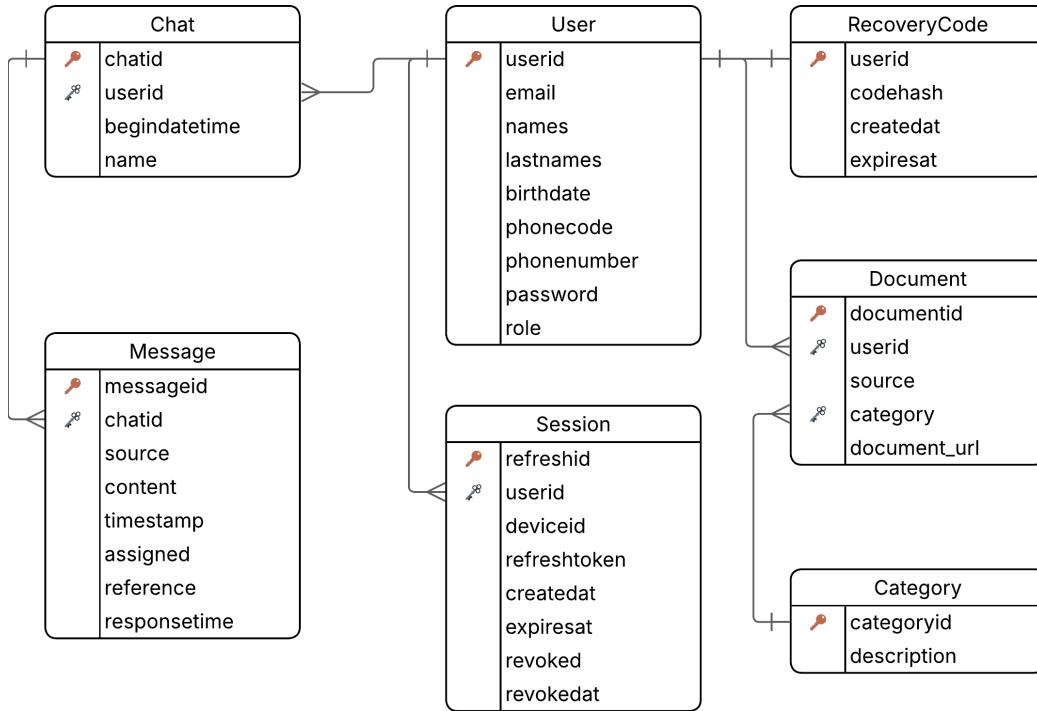


Figura 3: Diagrama entidad-relación (ER) de la base de datos relacional.

- Motor: PostgreSQL en AWS RDS (*Relational Database Service*).
- Entidades: usuarios, chats, mensajes, sesiones, documentos, categorías, códigos de recuperación.
- Relaciones:
 - Un usuario puede tener varios chats.
 - Cada chat contiene múltiples mensajes.
 - Un usuario puede tener varias sesiones.
 - Un usuario puede subir varios documentos.
 - Una categoría puede incluir múltiples documentos.
 - Un usuario solo puede tener un único código de recuperación a la vez.
- Mantenimiento: restricciones de claves foráneas, eliminación en cascada y validaciones para asegurar la integridad referencial.

Base de datos vectorial

Para llevar a cabo búsquedas semánticas eficientes y recuperar fragmentos de documentos relevantes en función de las preguntas del usuario (sistema *Retrieval-Augmented Generation*), se implementó una base de datos vectorial utilizando Pinecone. Esta base almacena los *embeddings* generados previamente a partir de los fragmentos de texto,

permitiendo consultas rápidas y precisas basadas en similitud semántica. La métrica de similitud de coseno fue seleccionada para evaluar la cercanía entre vectores, optimizando la relevancia de los resultados obtenidos.

- Motor: Pinecone, con métrica de similitud *cosine*.
- Contenido: embeddings de fragmentos de documentos con metadatos (fuente, categoría, documento, bloque, relevancia).
- Consultas top-K para recuperar los fragmentos más relevantes en relación al vector generado a partir de la pregunta realizada. Esto lo realiza *Pinecone* de forma automática, dependiendo del método de similitud configurado al momento de crear el índice. En este caso, se indicó que se utilice la métrica de similitud *cosine*, puesto que dicha métrica mide el ángulo de inclinación entre vectores, lo que determina qué tan similar es la dirección a la que se dirigen (orientación semántica). La fórmula que guía este cálculo es la siguiente:

$$\text{Similitud Coseno}(A, B) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

3. Servicio complementario de *Python* para comunicación con LLM

- **Función:** gestionar el flujo de *Retrieval-Augmented Generation (RAG)*, integrando la base vectorial con el modelo de lenguaje para generar respuestas contextualizadas y fundamentadas.
- **Flujo realizado:**
 - a) **Recepción de la pregunta en NodeJS:**
 - 1) El usuario envía una pregunta en texto plano a través de la interfaz de la aplicación.
 - 2) NodeJS recibe la pregunta y prepara la solicitud para el microservicio Python.
 - b) **Procesamiento en Python:**
 - 1) El microservicio Python recibe la pregunta enviada desde NodeJS.
 - 2) Genera un *embedding* del texto de la pregunta utilizando el modelo *text-embedding-3-small* de OpenAI, transformando la información textual en un vector semántico.
 - 3) Se realiza una consulta al índice de Pinecone con el *embedding* generado, recuperando los cinco fragmentos más relevantes del *corpus* vectorizado, que servirán como contexto para la respuesta.
 - 4) Si no se encuentran fragmentos relevantes, se procede a generar una respuesta estándar indicando la falta de información suficiente.
 - 5) Si la búsqueda semántica obtuvo resultados, se extrae el texto plano almacenado en la metadata de los vectores y se combina la pregunta original, construyendo un *prompt* en texto plano que resume la información relevante y plantea la consulta al modelo de lenguaje.
 - 6) Envía el *prompt* al modelo LLM para generar la respuesta.
 - 7) Genera un objeto JSON que incluye la pregunta original, la respuesta obtenida y el tiempo de procesamiento, garantizando trazabilidad de la interacción.

- c) **Devolución de la respuesta a NodeJS:**
- 1) NodeJS recibe el JSON con la respuesta generada por el LLM.
 - 2) Formatea y entrega la respuesta al usuario final a través de la interfaz de la aplicación.
 - 3) Simultáneamente, guarda la pregunta y la respuesta en la base de datos relacional para mantener un historial de interacciones.

El flujo completo de procesamiento de preguntas se ilustra en el Algoritmo 7.

Algoritmo 7 Flujo completo de procesamiento de preguntas

```

1: procedure PROCESARPREGUNTA(usuario)
2:   pregunta ← RecibirPregunta(usuario)
3:   EnviarPreguntaAPython(pregunta)
4:   embedding ← GenerarEmbedding(pregunta)
5:   contexto ← ConsultarPinecone(embedding, topK=5)
6:   prompt ← ConstruirPrompt(pregunta, contexto)
7:   respuesta ← ConsultarLLM(prompt)
8:   jsonRespuesta ← ArmarJSON(pregunta, respuesta, tiempoProcesamiento)
9:   EnviarAlCliente(jsonRespuesta)
10:  GuardarMensaje(usuario, respuesta)
11: end procedure

```

5.4.2. Resultado final

Al finalizar este *sprint*, se obtuvo:

- *Backend* modular bajo MVC, con rutas, controladores y modelos independientes.
- Base de datos relacional (PostgreSQL) con integridad referencial y seguridad.
- Base vectorial en Pinecone, indexada y lista para búsquedas semánticas eficientes.
- Servicio Python que integra recuperación contextual y generación ética de respuestas mediante LLM.
- Flujo completo validado: desde envío de pregunta hasta devolución de respuesta fundamentada.

Este *sprint* consolidó la infraestructura técnica del sistema, asegurando operación confiable, trazabilidad de datos y escalabilidad futura para el asistente educativo.

5.5. *Sprint 4: Desarrollo de la interfaz móvil en Kotlin*

Duración estimada: 4 semanas

Este *sprint* tuvo como objetivo diseñar e implementar la interfaz móvil de la aplicación del asistente educativo, utilizando Kotlin para garantizar integración nativa con Android y un flujo de interacción intuitivo para el usuario, de manera que se permitiera una interacción eficiente con el asistente al integrar las funcionalidades proporcionadas por el *backend* a través de servicios *REST*. El diseño de la arquitectura se basó en el patrón de diseño MVVM (*Modelo-Vista-ViewModel*), lo que permitió separar responsabilidades, facilitar la escalabilidad del código y mantener una clara independencia entre la lógica de negocio, la gestión de datos y la capa de presentación.

5.5.1. Ejecución

1. Diseño de arquitectura

- Se adoptó el patrón de diseño **Modelo-Vista-ViewModel (MVVM)**, siguiendo la estructura de la Figura 4. Esta arquitectura permite separar las responsabilidades del sistema, facilitando el mantenimiento y escalabilidad; a su vez, mediante este patrón de diseño se logra una comunicación reactiva entre la interfaz de usuario y las fuentes de datos, mediante el uso de componentes de arquitectura de Android como *LiveData*, *ViewModel* y *Repository*.

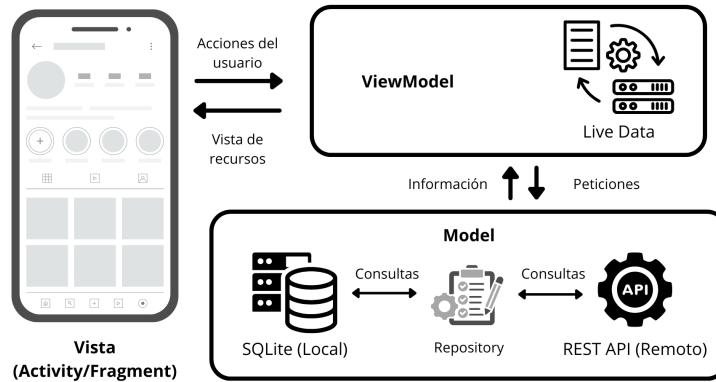


Figura 4: Arquitectura MVVM (Modelo, Vista, ViewModel).

■ Vista (View):

- Implementada a través de *Activities* y *Fragments* organizados modularmente.
- Cada fragmento representa una sección funcional del sistema (por ejemplo: inicio de sesión, chat, documentos, perfil).
- Se aplicó el patrón de navegación basado en *NavHostFragment* y *SafeArgs*, garantizando transiciones seguras y controladas entre vistas.
- Se usaron componentes de diseño *Material Design 3* para mantener consistencia visual, accesibilidad y adaptabilidad en distintos tamaños de pantalla.
- Cada *Fragment* o *Activity* cuenta con su propia plantilla, definida en formato XML (*Extensible Markup Language*), que especifica la disposición de los elementos visuales.

■ **ViewModel:**

- Actúa como intermediario entre la vista y las fuentes de datos, manejando la lógica de presentación.
- Emplea *LiveData* y *StateFlow* para notificar automáticamente a la vista sobre cambios en los datos.
- Encapsula la interacción con los repositorios, garantizando que la vista permanezca libre de lógica de negocio.
- El Algoritmo 8 ilustra el flujo básico de un ViewModel típico.

Algoritmo 8 Flujo básico de un ViewModel

```
1: procedure OBTENERMENSAJES(chatId)
2:   emitar(Estado.Cargando)
3:   datos ← repositorio.obtenerMensajes(chatId)
4:   emitar(Estado.Exitoso(datos))
5: end procedure
```

■ **Modelo (Repository y Data Sources):**

- Los repositorios centralizan el acceso a las fuentes de datos, tanto locales como remotas.
- Se definieron dos capas de origen de datos:
 - a) **Data Local:** implementada con *Room Database*, que gestiona entidades como usuarios, mensajes, chats y documentos. Esta capa permite el acceso a información sin conexión a internet (historiales de chat, información del usuario, listado de documentos disponibles), así como el almacenamiento persistente de la información obtenida mediante otras fuentes de datos que sí requieran comunicación con internet.
 - b) **Data Remota:** implementada mediante *Retrofit* y *OkHttp*, se utiliza para consumir los endpoints del *backend* desarrollado en el *sprint* anterior.
- Los repositorios determinan la fuente más apropiada según la disponibilidad de conexión y estado de sincronización.

■ **Gestión de dependencias:**

- Se empleó *Hilt (Dagger)* para la inyección de dependencias, simplificando la creación de instancias de *ViewModels*, repositorios y servicios. Este enfoque garantiza bajo acoplamiento entre componentes y favorece la escalabilidad del sistema.
- La configuración de los módulos de *Hilt* se realizó en la carpeta *di/*, donde se definieron las dependencias necesarias para la aplicación, como clientes *Retrofit*, bases de datos *Room* y repositorios.
- Se utilizaron otras dependencias básicas como *AppCompat* y *ConstraintLayout* para asegurar compatibilidad y flexibilidad en el diseño de la interfaz, así como *NavigationFragment* para gestionar la navegación entre pantallas.
- Por su parte, el uso de *ThreeTen* permitió mantener la compatibilidad de manejo de fechas para versiones antiguas de Android (con soporte desde Android 7 en adelante), mientras que *Glide* facilitó la carga y gestión eficiente de archivos dentro de la aplicación.

- **Manejo de estado y persistencia:**

- Se usaron *ViewModelScope* y *CoroutineScope* para ejecutar tareas asíncronas sin bloquear la interfaz.
- Se implementó almacenamiento persistente mediante *SharedPreferences* y *Room*, de manera que incluso sin internet la aplicación aún fuera utilizable, a pesar de no poder realizar peticiones al asistente ya que esta función sí requiere comunicación con internet para realizar solicitudes a *OpenAI*.
- La base de datos local permite al usuario acceder a su historial de chats, datos personales e historial de documentos (en el caso de usuarios con rol **Administrador**) aún sin conexión a internet.

2. Diseño de flujo de interacción

- **Activities**

- *SplashActivity*: esta es la vista principal de la aplicación. Si bien visualmente tan solo muestra el logo del proyecto, su función principal es verificar si el usuario ya ha iniciado sesión previamente (mediante un token JWT almacenado en *SharedPreferences*) o bien si debe ser redirigido a la pantalla de login. A partir de esta lógica se permite que el usuario no deba iniciar sesión cada vez que abre la aplicación, mejorando la experiencia de uso.
- *UnloggedActivity*: esta actividad contiene los fragmentos relacionados con la autenticación del usuario, incluyendo *LoginFragment*, *RegisterFragment* y *RecoverPasswordFragment*. Cada fragmento maneja su propia lógica de validación y comunicación con el *backend* para gestionar el acceso seguro a la aplicación.
- *MainActivity*: esta actividad permite el acceso a las funciones principales de la aplicación una vez el usuario ya haya sido autenticado. Como componente principal, alberga el *NavHostFragment*, el cual gestiona la navegación entre los distintos fragmentos funcionales, que incluyen el chat con el asistente, perfil de usuario y visualización de documentos. Por otro lado, esta actividad también incluye un menú lateral que facilita el acceso rápido al historial de chats del usuario.

- **Fragments**

- *LoginFragment*: permite al usuario ingresar sus credenciales (correo electrónico y contraseña) para autenticarse en el sistema. Incluye validaciones de formato y manejo de errores en caso de credenciales incorrectas.
- *RegisterFragment*: permite a nuevos usuarios crear una cuenta proporcionando información básica como correo electrónico, nombre, apellido, número de teléfono (con código de teléfono, en caso sea un número extranjero), contraseña y confirmación de contraseña. Incluye validaciones para asegurar la integridad y unicidad de los datos ingresados. La solicitud de un número telefónico cumple únicamente una función de verificación de identidad, ya que el sistema de mensajería SMS no fue implementado en esta fase del proyecto. Por el contrario, el correo electrónico sí es indispensable, ya que se utiliza para la recuperación de contraseña y notificaciones importantes como la creación o eliminación de documentos.

- *SendRecoveryFragment*: permite a los usuarios solicitar un código de recuperación de contraseña, el cual es enviado al correo electrónico registrado. El código será enviado **únicamente** si se encuentra una cuenta asociada al correo proporcionado.
- *VerifyCodeFragment*: permite a los usuarios ingresar el código de recuperación recibido por correo electrónico, así como validar la validez del mismo.
- *ResetPasswordFragment*: permite a los usuarios establecer una nueva contraseña tras haber validado el código de recuperación.
- *ChatFragment*: permite la interacción directa con el asistente virtual. Incluye un campo de texto para ingresar preguntas, un botón para enviar las consultas y una vista de lista que muestra el historial de mensajes intercambiados con el asistente.
- *ProfileFragment*: permite al usuario visualizar y actualizar su información personal, como nombre, apellido, correo electrónico y número de teléfono. La contraseña solo puede ser modificada mediante el flujo de recuperación de contraseña.
- *DocumentsFragment*: permite a los usuarios con rol **Administrador** visualizar el listado de documentos educativos cargados en el sistema, así como acceder al documento original de ser requerido, o bien, agregar o eliminar documentos según sea necesario.
- Implementación de navegación mediante *Navigation Component*, garantizando consistencia y control del back stack.
- Integración de indicadores de carga y estado de conexión, ofreciendo retroalimentación inmediata al usuario sobre la consulta al LLM.

3. Integración con *backend* y servicios de *Python*

- Consumo de endpoints REST del *backend* para autenticación, gestión de sesiones, envío de preguntas y recuperación de respuestas.
- Procesamiento de respuestas JSON, parseo y renderizado en la interfaz de usuario de manera clara y comprensible.
- Manejo de errores y reconexión ante fallos de red, asegurando robustez en la experiencia de usuario.

5.5.2. Resultado final

Al finalizar este *sprint*, se obtuvo:

- Aplicación móvil funcional en Android, con integración nativa mediante Kotlin y comunicación estable con el *backend*.
- Estructura modular clara (*Data, Dependency Injection, Helpers, UI, Resources*) que facilita mantenimiento y escalabilidad.
- Flujo de interacción optimizado para el usuario, incluyendo envío de preguntas, recepción de respuestas contextuales y visualización de documentos y administración de documentos para usuarios autorizados.

Este *sprint* permitió contar con una interfaz móvil operativa, lista para el despliegue y pruebas de usabilidad, estableciendo las bases para las fases finales de evaluación y documentación del proyecto.

5.6. *Sprint 5: Pruebas y validación*

Duración estimada: 3 semanas

Este *sprint* se centró en validar el funcionamiento integral del sistema, asegurando la correcta interacción entre la aplicación móvil, el *backend*, la base de datos relacional y vectorial, y el modelo de lenguaje (LLM, por sus siglas en inglés). Además, se buscó evaluar la calidad, precisión y confiabilidad de las respuestas generadas por el asistente virtual, así como recopilar retroalimentación mediante pruebas internas para ajustar y mejorar la herramienta antes de su entrega.

5.6.1. Ejecución

1. Pruebas funcionales del sistema

- Verificación de la comunicación entre la aplicación móvil (Kotlin), el *backend* (NodeJS), la base de datos relacional (PostgreSQL) y la base de datos vectorial (*Pinecone*). Se considera exitoso si cada módulo responde correctamente a las solicitudes y envía datos esperados al módulo contiguo. Este procedimiento se detalla en mayor profundidad en la sección de resultados.
- Pruebas de endpoints mediante *Postman* para asegurar correcta autenticación de usuarios, envío de preguntas, recuperación de respuestas y gestión de historial de chats. Se considera exitoso si cada uno de los endpoints funciona como se espera, incluyendo la validación de parámetros, restricciones de acceso, formato de respuestas y manejo de errores.
- Comprobación de la integridad de los datos entre los distintos módulos, incluyendo creación, lectura, actualización y eliminación de información (CRUD). Se considera exitoso si todos los datos manipulados a través del cliente (ya sea *Postman* o la aplicación final obtenida) se reflejan correctamente en la base de datos relacional y vectorial, manteniendo consistencia y precisión, así como la integridad referencial a los documentos originales almacenados en el contenedor S3 (es decir, si un elemento se crea o se elimina en cualquiera de las tres fuentes de datos, este cambio debe reflejarse en las otras dos).

2. Pruebas de calidad y confiabilidad de las respuestas

- Definición de un conjunto de preguntas de prueba basadas en escenarios educativos reales, cubriendo temas clave de formación ciudadana y valores morales. Se utiliza un total de 50 preguntas, distribuidas en categorías como ética, ciudadanía, formación ciudadana o democracia, así como consultas que evalúan la capacidad del asistente para manejar situaciones hipotéticas o dilemas morales en los cuales podría incurrir el usuario. También se incluyen preguntas de control

que no están relacionadas con el contenido base, para evaluar la capacidad del modelo de rechazar consultas fuera de contexto y verificar la no alucinación del modelo.

- Validación directa de las respuestas generadas por el asistente comparando con el contenido base documentado utilizado para entrenar y alimentar al modelo. No basta solamente con que el modelo sea capaz de responder coherentemente, sino que toda información proporcionada debe estar alineada con el contenido educativo aprobado por expertos.
- Identificación de casos en los que el modelo no proporcione información suficiente o presente inconsistencias, documentando hallazgos para ajuste de contenido o configuración de *embeddings*. Dentro del conjunto de preguntas utilizadas para la evaluación del sistema, se toman en cuenta aquellas preguntas que el modelo sí se espera que sea capaz de responder, pero que no lo hace correctamente (ya sea por falta de información en el contenido base, por errores en la generación de respuestas, o bien, por una estructuración incorrecta del *prompt* utilizado para solicitar la respuesta al modelo). Estos casos se documentan para su posterior análisis y ajuste en futuras iteraciones.
- Evaluación del tiempo de respuesta del sistema. A partir del conjunto de preguntas de prueba, se mide el tiempo transcurrido desde que el usuario envía la consulta hasta que recibe la respuesta generada por el asistente. Se considera exitoso si el tiempo de respuesta promedio se mantiene dentro de un rango que permita una experiencia de usuario fluida y eficiente.

5.6.2. Resultado final

Al concluir este *sprint*, se logró:

- Validación completa de la integración entre *frontend*, *backend* y bases de datos, garantizando estabilidad y funcionalidad del sistema, así como el correcto funcionamiento de todos los componentes involucrados antes de incurrir en un análisis específico para identificar puntos de mejora. Se verifica que todos los componentes técnicos funcionen según lo previsto, encaminados al cumplimiento de los objetivos del proyecto.
- A partir de las pruebas de calidad y confiabilidad, se obtuvieron resultados cuantitativos que permiten obtener el estado actual del modelo RAG (*Retrieval-Augmented Generation* en términos de)
 - **Eficiencia**
 - **Exito (%)**
 - **Nivel de congruencia (%)**
- Implementación de mejoras en la interfaz y en la lógica de generación de prompts para optimizar la experiencia de usuario y la pertinencia pedagógica.
- Documentación de resultados de prueba y recomendaciones para mantenimiento futuro y escalabilidad del proyecto.

Este *sprint* permitió asegurar que la aplicación estuviera lista para su uso efectivo público, a través de proporcionar respuestas precisas y contextualizadas, así como el establecimiento de las bases para fases futuras de despliegue y monitoreo continuo del asistente virtual.

5.7. *Sprint 6: Documentación y presentación*

Duración estimada: 3 semanas

Este *sprint* se centró en consolidar toda la documentación generada durante el desarrollo del proyecto y preparar la presentación final del asistente virtual de formación ciudadana y valores morales. El objetivo fue garantizar que tanto los resultados como los procesos utilizados quedaran claramente registrados, así como asegurar que el producto final estuviera disponible para revisión, prueba y entrega formal al cliente.

5.7.1. Ejecución

1. Elaboración del informe final

- Integración de la información de todos los *sprints* previos en un documento único, estructurado y coherente.
- Inclusión de resultados de cada *sprint*, análisis de hallazgos, decisiones de diseño y mejoras implementadas.
- Redacción de conclusiones generales y recomendaciones para futuras iteraciones, escalabilidad o mejoras del asistente virtual.
- Formateo del documento en LaTeX, asegurando uniformidad, claridad y cumplimiento de estándares académicos y de presentación profesional.

2. Preparación de la presentación final

- Desarrollo de material visual que resuma el proyecto, incluyendo diagramas de arquitectura, capturas de pantalla del prototipo móvil, flujo de interacción y ejemplos de uso del asistente.
- Elaboración de una presentación estructurada para explicar el proceso de desarrollo, resultados obtenidos y funcionalidades del sistema.
- Ensayo de la presentación y ajuste de contenido para garantizar claridad, concisión y relevancia para el público objetivo.

3. Traslado y entrega de documentación al cliente

- Consolidación de repositorios de código, documentos de investigación, fichas de usuario, diagramas de arquitectura y demás materiales generados.
- Entrega formal de toda la documentación y repositorios a la Fundación de Scouts de Guatemala, asegurando que puedan acceder a todos los recursos para pruebas, mantenimiento y futuras actualizaciones.
- Registro de la entrega, incluyendo inventario de archivos, versión final de documentación y evidencia de disponibilidad del producto para pruebas finales.

5.7.2. Resultado final

Como resultado de este *sprint*, se logró:

- Un informe final consolidado, claro y completo que documenta todo el proceso de desarrollo, análisis y resultados del proyecto.
- Material de presentación profesional listo para exponer ante el cliente y otros interesados.
- Entrega formal de toda la documentación y repositorios al cliente, asegurando disponibilidad total de recursos para pruebas, evaluación y futuras mejoras.
- Registro de la entrega y validación de que el producto final está operativo y listo para su uso y pruebas definitivas.

Este *sprint* concluyó con la transferencia completa del conocimiento y del producto, cerrando oficialmente el ciclo de desarrollo inicial del asistente virtual y dejando una base sólida para el mantenimiento y escalabilidad futura del proyecto.

CAPÍTULO 6

Resultados

Esta fase del desarrollo del proyecto *Sapien - Ciudadano Digital* concluyó con la implementación de un prototipo funcional de asistente virtual orientado a la educación informal en ciudadanía y valores morales, el cual integra de manera coherente los componentes de procesamiento de lenguaje natural, recuperación semántico de contexto y el diseño y desarrollo de una interfaz nativa móvil para dispositivos *Android*. Aunque los resultados obtenidos no constituyen una validación empírica del impacto educativo del sistema, esta primera versión permitió demostrar la viabilidad técnica y conceptual del proyecto, así como su alineación con los principios establecidos en el marco teórico y los objetivos específicos planteados, enfatizando los logros técnicos y visuales alcanzados durante los *sprints* definidos previamente.

6.1. Implementación del modelo LLM y flujo RAG

Durante el tercer *sprint* del desarrollo, se implementó un flujo RAG (*Retrieval-Augmented Generation*), el cual permite al asistente virtual generar respuestas fundamentadas en los contenidos educativos previamente procesados y almacenados en la base de datos. Este flujo se compone de los siguientes pasos. Este flujo integra los siguientes componentes principales:

1. **REST API en NodeJS** como *backend* del sistema, encargado de orquestar la comunicación entre la aplicación móvil y los servicios de procesamiento de lenguaje natural.
2. **Microservicio en Python** para la interacción con el modelo LLM (*Large Language Model*) de OpenAI, responsable de generar las respuestas, así como del procesamiento de documentos.
3. **API de OpenAI** para la generación de embeddings y respuestas.
4. **Base de datos vectorial en Pinecone** para el almacenamiento y recuperación de embeddings generados a partir de los contenidos educativos.

5. **Base de datos relacional en PostgreSQL**, la cual se encarga de gestionar usuarios, mensajes, sesiones y la información relacionada con los documentos procesados para asegurar trazabilidad desde los vectores hasta el documento original almacenado en el sistema.

El flujo completo del procesamiento RAG, desde que se recibe una respuesta a través del cliente (aplicación móvil) hasta que se devuelve la respuesta generada, se ilustra en la Figura 5

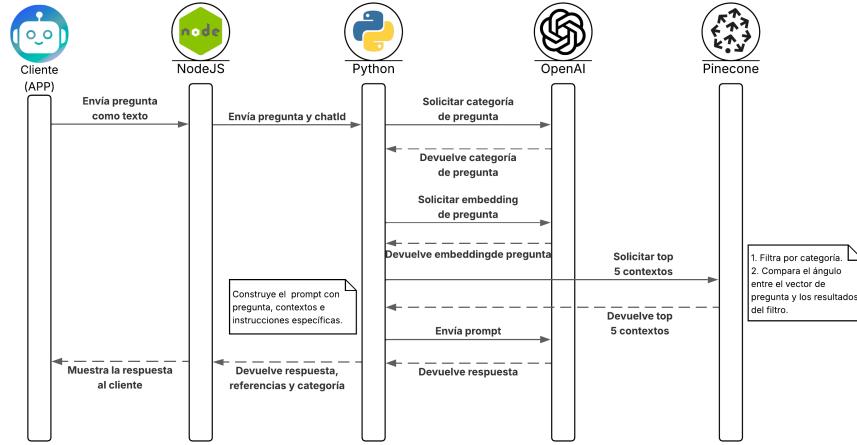


Figura 5: Flujo de procesamiento de preguntas mediante RAG.

6.2. Integración de la base de datos vectorial y relacional

El almacenamiento de información a lo largo del sistema se describe en 3 elementos principales:

- **Base de datos relacional (PostgreSQL)**: gestor principal de los datos del sistema; aquí se almacenan los datos de los usuarios, chats, mensajes, sesiones, códigos de recuperación de contraseñas y metadatos básicos de los documentos procesados.
- **Base de datos vectorial (Pinecone)**: encargada de almacenar los embeddings generados a partir de los documentos seleccionados. Esta es la base del funcionamiento RAG, ya que permite recuperar el contexto necesario, según la pregunta realizada, para que el modelo LLM pueda generar respuestas fundamentadas.
- **Documentos originales (Contenedor AWS S3)**: aquí se almacenan todos los documentos originales cargados al sistema, lo que permite su posterior consulta o descarga por parte de los usuarios con rol de **Administrador**. Cabe aclarar que se maneja un protocolo de trazabilidad basado en el identificador único generado para el contenedor, el cual se almacena como parte de los metadatos tanto en la base de datos vectorial como en la relacional; de esta forma, se mantiene la coherencia al momento de eliminar documentos o consultar su fuente original.

permitieron verificar la correcta sincronización entre todas las fuentes de datos mediante los servicios expuestos por el *backend*, lo que permitió que las consultas del usuario se asocien con el contenido adecuado. La Figura 6 muestra la vista desde el panel de pinecone, una vez se han cargado vectores correctamente.

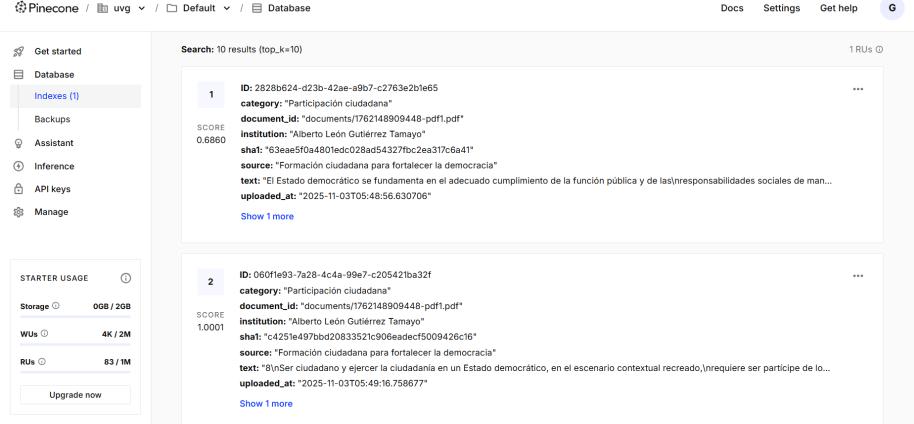


Figura 6: Vista del índice vectorial en Pinecone con los embeddings cargados.

6.3. Desarrollo del backend en NodeJS

El *backend* implementado durante el *sprint 3* se comprobó mediante pruebas de endpoints *REST*, a través de la herramienta *Postman*, lo cual demostró la existencia de comunicación estable con los servicios de Python y la aplicación móvil. Los *endpoints* principales cubren autenticación, recuperación de contraseña, envío de mensajes, historial de chats y administración de documentos.

Las pruebas realizadas a través de *Postman* permitieron verificar la correcta integración entre los módulos del sistema, la estabilidad del flujo de información y la capacidad del sistema para generar respuestas basadas en los fragmentos de texto almacenados. El Cuadro 2 presenta los resultados de las consultas y controles realizados durante la fase de validación técnica y semántica del sistema.

NOTA: Durante las pruebas realizadas, se incluyen solicitudes de *control*, pensadas para validar que el modelo no sea capaz de responder a preguntas fuera del contexto establecido mediante la base de datos vectorial. Se espera que estas preguntas no sean respondidas, para evitar *alucinaciones* del modelo y mantener la congruencia del sistema. Por otro lado, las preguntas de *consulta* están relacionadas directamente con los contenidos educativos procesados y almacenados en el sistema, por lo que su congruencia se refiere a que el modelo haya sido capaz no solo de responderla, sino que el contenido retornado tenga relación con los contenidos y sepa responder apropiadamente a la pregunta planteada.

Cuadro 2: Resultados de las consultas y controles con sus tiempos de respuesta, éxito y congruencia.

Consulta	Tiempo de respuesta (s)	Exitosa	Congruente	Tipo
Dime en una frase qué es el civismo	9.643	Sí	Sí	Consulta
¿Qué es la ciudadanía?	12.285	Sí	Sí	Consulta
¿Cómo se pueden abordar los dilemas morales en la educación cívica?	16.715	Sí	Sí	Consulta
¿Qué es Formación Ciudadana?	10.854	Sí	Sí	Consulta
Dame el listado de todos los departamentos de Guatemala	10.329	No	Sí	Control
¿Cuál es la relación entre Formación Ciudadana y Democracia?	17.456	Sí	Sí	Consulta
¿Qué importancia crees que tiene la formación ciudadana en la educación actual?	14.502	Sí	Sí	Consulta
¿Cuál es el rol de la escuela en la educación en valores?	19.409	Sí	Sí	Consulta
¿Cómo puedo practicar el civismo en mi familia?	20.086	Sí	Sí	Consulta
¡Es una emergencia! ¿Cómo aplico un torniquete?	8.12	No	Sí	Control
Dame 5 tips prácticos para ser mejor ciudadano	13.652	Sí	Sí	Consulta
¿Cómo crees que la empatía puede influir en la convivencia escolar?	11.255	Sí	Sí	Consulta
¿Qué importancia tiene la discusión sobre ética en el aula para la formación ciudadana?	16.29	Sí	Sí	Consulta
¿Qué es la democracia?	12.53	Sí	Sí	Consulta
¿Cuál es el procedimiento de emergencia si se incendia un televisor?	11.418	No	Sí	Control

Continúa en la siguiente página

<i>Consulta</i>	<i>Tiempo de respuesta (s)</i>	<i>Exitosa</i>	<i>Congruente</i>	<i>Tipo</i>
¿Cómo practico la democracia si soy estudiante?	16.49	Sí	Sí	Consulta
¿Es posible ser corrupto sin ser político?	7.252	No	No	Consulta
Si soy buen ciudadano, ¿qué puedo esperar?	13.584	Sí	Sí	Consulta
¿Cómo se reconoce la formación ciudadana?	16.203	Sí	Sí	Consulta
Le di chocolate a mi perro, ¿cómo puedo salvarlo?	8.183	No	Sí	Control
¿La democracia representativa es perfecta?	10.79	No	No	Consulta
¿Cómo es un ciudadano ejemplar?	21.064	Sí	Sí	Consulta
¿Una nación sin democracia tiene ciudadanía?	12.842	No	No	Consulta
¿Votar es importante?	10.135	Sí	Sí	Consulta
¿Cuáles son los componentes básicos de un motor?	9.246	No	Sí	Control
Lista 5 ideas para que mi círculo sea más democrático.	14.8	Sí	Sí	Consulta
¿Qué puedo hacer para mostrar civismo además de votar?	18.837	Sí	Sí	Consulta
Si veo corrupción y no denuncio, ¿fallé como ciudadano?	12.974	Sí	Sí	Consulta
¿Qué caracteriza un país democrático?	17.86	Sí	Sí	Consulta
¿Cuáles son los países más democráticos del mundo?	10.319	No	Sí	Control
¿Qué valores son esenciales para vivir en sociedad?	13.914	Sí	Sí	Consulta
¿Qué acciones diarias me hacen buena persona?	10.533	No	No	Consulta

Continúa en la siguiente página

<i>Consulta</i>	<i>Tiempo de respuesta (s)</i>	<i>Exitosa</i>	<i>Congruente</i>	<i>Tipo</i>
¿Qué implica la democracia en el bienestar social?	14.528	Sí	Sí	Consulta
¿Cómo demuestro valores en mi día a día?	18.311	Sí	Sí	Consulta
¿Qué tan algo puede respirar un ser humano?	10.846	No	Sí	Control
Además de la escuela y la familia, ¿dónde más se obtienen los valores?	9.213	No	No	Consulta
¿Una persona en el tráfico casi me choca, cómo puedo actuar de forma ética?	8.178	No	No	Consulta
¿Qué papel representa la tecnología en la ciudadanía moderna?	15.59	Sí	Sí	Consulta
¿Por qué me debería preocupar por mis valores y ciudadanía?	13.73	Sí	Sí	Consulta
¿Cuántos vasos de agua al día bebe un buen ciudadano?	8.943	No	Sí	Control
¿La democracia es un modelo de gobierno ideal?	10.591	Sí	Sí	Consulta
¿Cómo puedo ser fiel a mis valores en un mal día?	7.045	No	No	Consulta
¿Por qué es importante la ética?	14.326	Sí	Sí	Consulta
¿Ser buen ciudadano es ser alguien ético?	10.728	Sí	Sí	Consulta
¿Cuáles son los componentes de un motor por combustión?	9.004	No	Sí	Control
¿Cuál es la diferencia entre moral y ética?	8.176	No	No	Consulta
¿Qué acciones demuestran ética en mi conducta?	16.241	Sí	Sí	Consulta

Continúa en la siguiente página

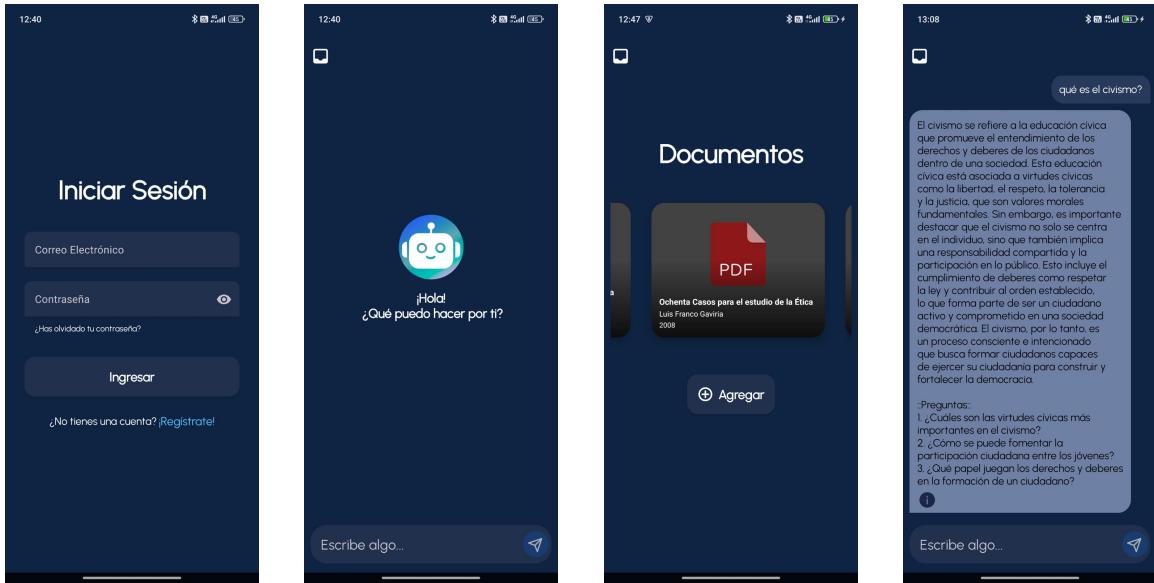
<i>Consulta</i>	<i>Tiempo de respuesta (s)</i>	<i>Exitosa</i>	<i>Congruente</i>	<i>Tipo</i>
¿Cuál es la forma ética de actuar ante alguien que me ha hecho un mal?	9.471	No	No	Consulta
Una persona en el tráfico casi me choca, ¿cómo puedo actuar de forma ética?	16.114	Sí	Sí	Consulta
¿Qué animales guatemaltecos están en peligro de extinción?	8.531	No	Sí	Control
Promedio / Totales	12.783	77.5 %	82 %	

Como se puede observar en el Cuadro 2, el sistema logró responder correctamente el 77.5 % de las consultas realizadas, con un nivel de congruencia del 82 %. Estos resultados, aunque preliminares, indican un buen desempeño del sistema en la generación de respuestas fundamentadas en los contenidos educativos procesados.

6.4. Interfaz móvil en Kotlin (Frontend)

Durante el cuarto *sprint*, se desarrolló la aplicación móvil nativa para dispositivos *Android* utilizando el lenguaje de programación Kotlin, implementando el patrón de diseño MVVM (Modelo-Vista-ViewModel) para asegurar una arquitectura modular y mantenible. Como resultado, se obtuvo una interfaz funcional, amigable con el usuario y cuyo uso no requiere estar familiarizado previamente con herramientas parecidas. A través de esta herramienta, se permite a los usuarios acceder al sistema mediante autenticación, mantener conversaciones con el asistente, visualizar el historial de chats, administrar el perfil del usuario y, para usuarios con rol de **Administrador**, gestionar los documentos cargados en el sistema.

La Figura 7 muestra algunas de las pantallas principales de la aplicación móvil desarrollada.



(a) Pantalla de inicio de sesión.

(b) Pantalla de chat con asistente virtual.

(c) Pantalla de gestión de documentos.

(d) Pantalla de chat con mensajes.

Figura 7: Interfaz de la aplicación móvil desarrollada.

El diseño de la interfaz emplea principios y componentes de *Material Design*, así como el uso de navegación por componentes (*Navigation Component*) y *Hilt*, para la inyección de dependencias.

CAPÍTULO 7

Discusión

El desarrollo del proyecto *Ciudadano Digital* permitió obtener un prototipo funcional de asistente virtual orientado a la educación ciudadana y en valores morales, lo que evidencia la viabilidad técnica y conceptual de la propuesta inicial, así como el desarrollo de una arquitectura tecnológica y un flujo de trabajo adecuados para lograr el alcance de los objetivos planteados. La integración de un mecanismo de recuperación aumentada de generación (RAG) con modelos de lenguaje grande (LLM) demostró ser una estrategia efectiva para ofrecer respuestas contextualizadas y fundamentadas en un *corpus* educativo que, por su parte, también cuenta con una metodología específica para la selección y procesado del mismo.

Si bien los resultados obtenidos se ven limitados a un entorno de validación externa, puesto que no se realizaron pruebas con usuarios finales, permiten analizar el desempeño del sistema a partir de 3 perspectivas reconocibles: técnica, semántica y visual.

Desde el punto de vista técnico, el sistema construido logró integrar exitosamente las tecnologías planteadas desde el inicio del proyecto; se logró la generación de un corpus dedicado a la fundamentación de las respuestas obtenidas del modelo, esto a través de la base de datos vectorial *Pinecone* que, a su vez, se comunica correctamente con la base de datos relacional y con el contenedor de archivos. Todo este sistema permite la obtención apropiada de respuestas basadas únicamente en los documentos seleccionados y procesados, garantizando que no exista alucinación por parte del modelo y que este se limite a responder únicamente las cuestiones que estén relacionadas con el contenido de la base de datos vectorial.

Como se evidencia en la Figura 5, el flujo de recuperación aumentada de generación (RAG) se ejecuta de manera sistemática al dividir correctamente las tareas entre los distintos componentes del sistema.

Sin embargo, si bien se cumplieron las expectativas técnicas de implementación, curado y procesamiento de datos, se identifican claramente nuevas oportunidades de mejora a abordar en iteraciones futuras del proyecto, tales como la optimización del tiempo de respuesta, la mejora en la gestión de errores y la incorporación de mecanismos de supervisión pedagógica

que garanticen continuamente la validación de los documentos utilizados para alimentar el modelo.

En el plano semántico; con una tasa de éxito del 77.5 %, acompañado de un 82 % de congruencia en las respuestas obtenidas, el sistema demostró un desempeño adecuado en la generación de respuestas fundamentadas y coherentes con el material curado. No obstante, estos resultados también indican áreas de oportunidad para mejorar la precisión y profundidad de las respuestas, especialmente en temas complejos o ambigüos, así como en temáticas más específicas a la realidad sociocultural y contextual del usuario que realiza la petición.

Desde una perspectiva pedagógica, si bien el proyecto fue ideado no como una herramienta de enseñanza directa, sino como un asistente secundario que陪伴e a los estudiantes en las inquietudes que los aquejen al salir del salón de clases, se reconoce la importancia de evaluar el impacto real que este tipo de tecnologías pueden tener en el aprendizaje y desarrollo ético de los jóvenes. En este sentido, futuras iteraciones deberían centrarse en la implementación de pruebas piloto con estudiantes y educadores, para medir no solo la eficacia técnica del asistente, sino también su capacidad para fomentar la reflexión crítica y el compromiso ciudadano.

Finalmente, desde una perspectiva visual, el diseño de la aplicación móvil fue construido a partir de componentes y directrices de diseño centradas en la experiencia de usuario usual de un propietario de dispositivos *Android*, pensado así para asegurar un mayor alcance entre la población objetivo. No obstante, se identifican oportunidades de mejora en la interfaz gráfica y en la experiencia de usuario, las cuales podrían optimizarse a través de pruebas de usabilidad con usuarios reales, para garantizar que la herramienta sea intuitiva, accesible y atractiva para los jóvenes. Cabe resaltar que, al no realizarse contacto directo con el usuario objetivo en ningún momento del desarrollo de esta fase, si bien se tomaron en cuenta buenas prácticas de diseño y la información documentada, esto podría no reflejar de manera fiel las necesidades y preferencias reales de los estudiantes guatemaltecos que se pretende que utilicen la herramienta; valdría la pena realizar estudios centrados en cómo los colores, tipografías, distribución de elementos y flujos de navegación impactan en la experiencia del usuario final.

Resulta pertinente afirmar, entonces, que el proyecto *Sapien - Ciudadano Digital* busca sentar las bases para la mejora continua del proyecto, enfocado en implementar la estructura tecnológica básica necesaria para la interacción con el modelo, la generación continua del *corpus* y los criterios de diseño iniciales para continuar el desarrollo de la aplicación móvil. Se logró la implementación técnica esperada, tomando en cuenta las limitaciones relacionadas con la falta de retroalimentación directa y validación empírica con usuarios finales. Sin embargo, esta primera fase del proyecto permite demostrar cómo la integración de tecnologías de inteligencia artificial puede contribuir al fortalecimiento de la educación en valores y formación ciudadana en contextos con recursos limitados, lo que abre la puerta a futuras investigaciones y desarrollos en este campo.

CAPÍTULO 8

Conclusiones

El desarrollo del proyecto *Sapien – Ciudadano Digital* permitió cumplir satisfactoriamente la creación de una herramienta tecnológica de educación informal orientada al acompañamiento en la adquisición de aprendizajes sobre formación ciudadana y valores morales; puesto que el prototipo construido demuestra la factibilidad técnica y conceptual de emplear modelos de lenguaje de gran escala (LLM) como apoyo en procesos educativos no formales, con especial atención en la coherencia discursiva, pertinencia temática y alineación con los principios establecidos en el marco teórico.

Asimismo, se logró alcanzar la implementación de un modelo LLM preentrenado y optimizado mediante la integración de un flujo RAG (*Retrieval-Augmented Generation*), el cual permite generar respuestas coherentes y contextualizadas con base en los documentos educativos procesados. Los resultados obtenidos en la validación técnica y semántica evidencian un comportamiento consistente y un adecuado control de respuestas no relacionadas con el *corpus* del proyecto, evitando así la alucinación. Esto respalda el cumplimiento del primer objetivo específico planteado y respalda la viabilidad del enfoque adoptado, así como su escalabilidad para futuras iteraciones del proyecto.

En cuanto a la integración de una base de datos vectorial, esto fue alcanzado mediante la implementación de *Pinecone* como índice semántico, el cual permitió el almacenamiento y recuperación eficiente de fragmentos relevantes a partir de los contenidos educativos almacenados, garantizando la trazabilidad entre los vectores, los metadatos y los documentos originales mediante la conexión con la base de datos relacional en *PostgreSQL* y el uso de contenedores de tipo S3. Esta integración facilitó la optimización del proceso de recuperación de información y permitió el desarrollo de un flujo RAG completo y funcional.

Por consiguiente, se obtuvo como producto final una aplicación móvil nativa para dispositivos *Android* a través de *Kotlin* y el patrón de diseño MVVM, logrando alcanzar una interfaz clara, funcional y visualmente coherente con los principios de *Material Design*. Esta herramienta facilita la interacción del usuario con todos los elementos que componen el sistema principal, lo que da paso a una portabilidad y accesibilidad adecuada para el público objetivo planteado, ya que permite el acceso al flujo RAG desde cualquier lugar y en

cualquier momento.

Cabe resaltar también que, al haber desarrollado el flujo RAG mediante un sistema modular y desacoplado, el producto final presenta una arquitectura flexible y escalable, encapsulado en una API REST capaz de ser utilizada por cualquier otro cliente, por lo que este proyecto no está limitado únicamente a la aplicación móvil desarrollada, sino que queda abierto a su uso tanto en plataformas web como en otros entornos tecnológicos.

De manera general, se concluye que el prototipo constituye una base sólida para futuras fases del proyecto, demostrando la viabilidad de integrar inteligencia artificial generativa en entornos educativos informales. Si bien se reconoce que los resultados se limitan al ámbito técnico y conceptual, los logros alcanzados hasta esta fase actúan como incentivo para avanzar hacia etapas de validación empírica con usuarios reales para evaluar el impacto pedagógico, la pertinencia cultural de las respuestas y la efectividad del acompañamiento en la formación ciudadana.

En síntesis, *Ciudadano Digital* representa un avance significativo en la aplicación de tecnologías emergentes al fortalecimiento de la educación en valores y ciudadanía. Su diseño modular, su arquitectura funcional y su enfoque ético ofrecen una base prometedora para continuar el desarrollo de soluciones educativas innovadoras, centradas en la reflexión moral, el pensamiento crítico y la participación ciudadana responsable.

CAPÍTULO 9

Bibliografía

- [1] UNESCO, *Recommendation on the Ethics of Artificial Intelligence*, <https://unesdoc.unesco.org/ark:/48223/pf0000381137>, París, 2021.
- [2] OECD, *AI and the Future of Skills, Volume 1: Capabilities and Assessments*, https://www.oecd.org/content/dam/oecd/en/publications/reports/2021/11/ai-and-the-future-of-skills-volume-1_2f19d213/5ee71f34-en.pdf, Paris, 2021.
- [3] Banco Mundial, *The Revolution of AI in Education: Innovations and Opportunities in Latin America and the Caribbean*, <https://profuturo.education/en/observatory/trends/the-revolution-of-ai-in-education-innovations-and-opportunities-in-latin-america-and-the-caribbean/>, Washington, DC, 2022.
- [4] A. Rivas, N. Buchbinder e I. Barrenechea, *The future of Artificial Intelligence in education in Latin America*, <https://oei.int/wp-content/uploads/2023/04/the-future-of-artificial-intelligence-in-education-in-latin-america-oei-profuturo.pdf>, Madrid, 2023.
- [5] Ministerio de Educación de Guatemala (MINEDUC), *Curriculum Nacional Base (CNB)*, <https://digeex.mineduc.gob.gt/digeex/wp-content/uploads/2020/09/CNB-Modalidades-Flexibles-Basico-Etapa-1.pdf>, Guatemala, 2020.
- [6] Centro de Investigaciones Económicas Nacionales (CIEN), *Diagnóstico de Educación y Tecnología: ¿Cuáles son los principales problemas?* <https://cien.org.gt/wp-content/uploads/2019/03/Resumen-ejecutivo-Educacion%CC%81n-VF.pdf>, Guatemala, 2019.
- [7] UNESCO, *Global education monitoring report, 2023: technology in education: a tool on whose terms?* <https://unesdoc.unesco.org/ark:/48223/pf0000385723>, París, 2023.
- [8] E. Levy y V. Robano, *How AI can support teachers in Latin America*, <https://www.brookings.edu/articles/how-ai-can-support-teachers-in-latin-america/>, 2025.
- [9] UNESCO, *AI and education: guidance for policy-makers*, <https://unesdoc.unesco.org/ark:/48223/pf0000376709>, París, 2021.

- [10] Frontiers, “The use of ChatGPT in teaching and learning: a systematic review through SWOT analysis approach,” *Frontiers in Education*, vol. 9, pág. 17, 2024. DOI: 10.3389/feduc.2024.1328769. dirección: <https://www.frontiersin.org/journals/education/articles/10.3389/feduc.2024.1328769/full>.
- [11] R. Tulsiani, *ChatGPT And The Future Of Personalized Learning In Higher Education*, <https://elearningindustry.com/chatgpt-and-the-future-of-personalized-learning-in-higher-education>, 2024.
- [12] E. Elstad, “AI in Education: Rationale, Principles, and Instructional Implications,” *arXiv preprint*, vol. 2412.12116, 2024. dirección: <https://arxiv.org/abs/2412.12116>.
- [13] F. in Education, “The role of artificial intelligence in education among college instructors,” *Frontiers in Education*, vol. 10, pág. 1560074, 2025. dirección: <https://www.frontiersin.org/journals/education/articles/10.3389/feduc.2025.1560074/full>.
- [14] Ministerio de Educación de Guatemala (MINEDUC), *Acuerdo Ministerial MINEDUC 2810-2023*, https://leyes.infile.com/index.php?id=181&id_publicacion=87539, Guatemala, 2023.
- [15] H. Qin, *Transforming Education with Large Language Models: Opportunities, Challenges, and Ethical Considerations*, https://www.researchgate.net/publication/382825702_Transforming_Education_with_Large_Language_Models_Opportunities_Challenges_and_Ethical_Considerations, 2024.
- [16] D. Córdova, *AI-Powered Educational Agents: Opportunities, Innovations, and Ethical Challenges*, <https://www.mdpi.com/2078-2489/16/6/469>, 2025.
- [17] Z. Levonian et al., “Designing Safe and Relevant Generative Chats for Math Learning in Intelligent Tutoring Systems,” *Journal of Educational Data Mining*, 2025. dirección: <https://jedm.educationaldatamining.org/index.php/JEDM/article/download/840/238>.
- [18] B. I. for Kids, *Challenges and opportunities in the European year of digital citizenship education*, <https://better-internet-for-kids.europa.eu/en/news/challenges-and-opportunities-european-year-digital-citizenship-education>, Accedido el 17 de octubre de 2025, 2024.
- [19] J. Carter, S. Liu y M. Huang, “Ethical implications of ChatGPT and other large language models in academia,” *Frontiers in Artificial Intelligence*, vol. 7, pág. 1615761, 2024. DOI: 10.3389/frai.2025.1615761.
- [20] J. Seibt y C. Vestergaard, “Know Thyself, Improve Thyself: Personalized LLMs for Self-Knowledge and Moral Enhancement,” *Science and Engineering Ethics*, vol. 30, págs. 1-17, 2024. DOI: 10.1007/s11948-024-00518-9.
- [21] F. in Psychology, “Optimizing academic engagement and mental health through AI: an experimental study on LLM integration in higher education,” *Frontiers in Psychology*, vol. 16, pág. 1641212, 2025. DOI: 10.3389/fpsyg.2025.1641212.
- [22] UNESCO, *Global Citizenship Education: Topics and Learning Objectives*. Paris, France: UNESCO Publishing, 2015. DOI: 10.54675/DRHC3544. dirección: <https://unesdoc.unesco.org/ark:/48223/pf0000232993>.

- [23] W. Schulz, D. Kerr, B. Losito, J. Ainley y J. Fraillon, *ICCS 2009 International Report: Civic Knowledge, Attitudes, and Engagement Among Lower-secondary Students in 38 Countries*. Amsterdam, Netherlands: IEA, 2010, ISBN: 978-90-79549-07-8. dirección: https://www.iea.nl/sites/default/files/2019-04/ICCS_2009_International_Report.pdf.
- [24] T. Bentley y D. Broady, “Education for Democratic Citizenship: Developing Skills and Competences in Students,” *European Journal of Education*, vol. 53, págs. 181-196, 2018. DOI: 10.1111/ejed.12271.
- [25] T. Lovat y R. Toomey, *Values Education and Quality Teaching: The Double Helix Effect*. Springer, 2009, págs. xi-xiv, ISBN: 978-1-4020-9962-5. DOI: 10.1007/978-1-4020-9962-5. dirección: <https://link.springer.com/book/10.1007/978-1-4020-9962-5>.
- [26] J. Crittenden y P. Levine, *Civic Education*, <https://plato.stanford.edu/entries/civic-education/>, 2007.
- [27] M. D. Barrett, *Competencies for Democratic Culture: Living Together as Equals in Culturally Diverse Democratic Societies*. Strasbourg, France: Council of Europe Publishing, 2016, ISBN: 978-92-877-1823-7. dirección: <https://rm.coe.int/16806ccc07>.
- [28] M. Hand, *A Theory of Moral Education*. Routledge, 2017, págs. 46-60, ISBN: 9781315708508. DOI: 10.4324/9781315708508. dirección: <https://www.taylorfrancis.com/books/mono/10.4324/9781315708508/theory-moral-education-michael-hand>.
- [29] P. H. Coombs, *The World Educational Crisis: A Systems Analysis*. New York, USA: Oxford University Press, 1968. dirección: <https://unesdoc.unesco.org/ark:/48223/pf0000075799>.
- [30] S. Mills y P. Kraftl, *Informal Education, Childhood and Youth: Geographies, Histories, Practices*. Springer, 2014, págs. 3-4, ISBN: 978-1-137-02773-3. DOI: 10.1057/9781137027733. dirección: <https://link.springer.com/book/10.1057/9781137027733>.
- [31] J. Van Dijk, “The Deepening Divide: Inequality in the Information Society,” *Sage Publications*, 2005. dirección: <https://sk.sagepub.com/book/mono/the-deepening-divide/toc>.
- [32] M. Teräs, “Education and technology: Key issues and debates,” *International Review of Education*, vol. 68, n.º 4, págs. 635-636, 2022. DOI: 10.1007/s11159-022-09971-9. dirección: <https://doi.org/10.1007/s11159-022-09971-9>.
- [33] W. Holmes, M. Bialik y C. Fadel, “Artificial Intelligence in Education: Promises and Implications for Teaching and Learning,” *Center for Curriculum Redesign*, 2019. dirección: <https://curriculumredesign.org/wp-content/uploads/AIED-Book-Excerpt-CCR.pdf>.
- [34] B. P. Woolf, *Building Intelligent Interactive Tutors: Student-Centered Strategies for Revolutionizing E-Learning*. Boston, MA, USA: Morgan Kaufmann, 2009, ISBN: 978-0-12-373594-2. dirección: https://booksite.elsevier.com/samplechapters/9780123735942/Sample_Chapters/01~Front_Matter.pdf.
- [35] T. e. a. Brown, “Language Models are Few-Shot Learners,” *Advances in Neural Information Processing Systems*, vol. 33, págs. 1877-1901, 2020. dirección: <https://arxiv.org/abs/2005.14165>.

- [36] C. e. a. Raffel, “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer,” *Journal of Machine Learning Research*, vol. 21, págs. 1-67, 2020. dirección: <https://arxiv.org/abs/1910.10683>.
- [37] P. e. a. Lewis, “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” *Advances in Neural Information Processing Systems*, vol. 33, págs. 9459-9474, 2020. dirección: <https://arxiv.org/abs/2005.11401>.
- [38] U. e. a. Khandelwal, “Generalization through Memorization: Nearest Neighbor Language Models,” *ICLR*, 2020. dirección: <https://arxiv.org/abs/1911.00172>.
- [39] Z. Wang et al., “Document Segmentation Matters for Retrieval-Augmented Generation,” en *Findings of the Association for Computational Linguistics: ACL 2025*, W. Che, J. Nabende, E. Shutova y M. T. Pilehvar, eds., Vienna, Austria: Association for Computational Linguistics, jul. de 2025, págs. 8063-8075, ISBN: 979-8-89176-256-5. DOI: 10.18653/v1/2025.findings-acl.422. dirección: <https://aclanthology.org/2025.findings-acl.422/>.
- [40] S. Gupta, R. Ranjan y S. N. Singh, *A Comprehensive Survey of Retrieval-Augmented Generation (RAG): Evolution, Current Landscape and Future Directions*, 2024. arXiv: 2410.12837 [cs.CL]. dirección: <https://arxiv.org/abs/2410.12837>.
- [41] T. Zheng, W. Li, J. Bai, W. Wang e Y. Song, *KnowShiftQA: How Robust are RAG Systems when Textbook Knowledge Shifts in K-12 Education?* 2025. arXiv: 2412.08985 [cs.CL]. dirección: <https://arxiv.org/abs/2412.08985>.
- [42] T. e. a. Mikolov, “Efficient Estimation of Word Representations in Vector Space,” en *Proceedings of ICLR*, 2013. dirección: <https://arxiv.org/abs/1301.3781>.
- [43] Q. Le y T. Mikolov, “Distributed Representations of Sentences and Documents,” en *Proceedings of ICML*, 2014, págs. 1188-1196. dirección: <https://arxiv.org/abs/1405.4053>.
- [44] J. e. a. Johnson, “Billion-Scale Similarity Search with GPUs,” en *IEEE Transactions on Big Data*, vol. 7, 2019, págs. 535-547. DOI: 10.48550/arXiv.1702.08734. dirección: <https://arxiv.org/abs/1702.08734>.
- [45] Y. Han, C. Liu y P. Wang, “A Comprehensive Survey on Vector Database: Storage and Retrieval Technique, Challenge,” *arXiv preprint arXiv:2310.11703*, 2023. dirección: <https://arxiv.org/abs/2310.11703>.
- [46] J. White et al., *A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT*, 2023. arXiv: 2302.11382 [cs.SE]. dirección: <https://arxiv.org/abs/2302.11382>.
- [47] Y. Zhou et al., *Large Language Models Are Human-Level Prompt Engineers*, 2023. arXiv: 2211.01910 [cs.LG]. dirección: <https://arxiv.org/abs/2211.01910>.
- [48] J. Wei et al., “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models,” en *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho y A. Oh, eds., vol. 35, Curran Associates, Inc., 2022, págs. 24 824-24 837. dirección: https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

- [49] O. Zawacki-Richter, V. I. Marín, M. Bond y F. Gouverneur, “Systematic review of research on artificial intelligence applications in higher education – where are the educators?” *International Journal of Educational Technology in Higher Education*, vol. 16, n.º 1, pág. 39, 2019. DOI: 10.1186/s41239-019-0171-0. dirección: <https://educationaltechnologyjournal.springeropen.com/articles/10.1186/s41239-019-0171-0>.
- [50] L. A. Favero, J. A. Pérez-Ortiz, T. Käser y N. Oliver, “Enhancing Critical Thinking in Education by means of a Socratic Chatbot,” en *International Workshop on AI in Education and Educational Research (ECAI 2024)*, Springer, 2024. DOI: 10.1007/978-3-031-93409-4_2.
- [51] S. Wollny, J. Schneider, D. Di Mitri, J. Weidlich, M. Rittberger y H. Drachsler, “Are We There Yet? - A Systematic Literature Review on Chatbots in Education,” *Frontiers in Artificial Intelligence*, vol. Volume 4 - 2021, 2021, ISSN: 2624-8212. DOI: 10.3389/frai.2021.654924. dirección: <https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2021.654924>.
- [52] E. Adamopoulou y L. Moussiades, “Chatbots: History, technology, and applications,” *Machine Learning with Applications*, vol. 2, pág. 100 006, 2020, ISSN: 2666-8270. DOI: <https://doi.org/10.1016/j.mlwa.2020.100006>. dirección: <https://www.sciencedirect.com/science/article/pii/S2666827020300062>.
- [53] K. Shuster, S. Poff, M. Chen, D. Kiela y J. Weston, *Retrieval Augmentation Reduces Hallucination in Conversation*, 2021. arXiv: 2104.07567 [cs.CL]. dirección: <https://arxiv.org/abs/2104.07567>.
- [54] Y. Bang et al., *A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity*, 2023. arXiv: 2302.04023 [cs.CL]. dirección: <https://arxiv.org/abs/2302.04023>.
- [55] R. Luckin y W. Holmes, *Intelligence Unleashed: An Argument for AI in Education*. London, UK: Pearson Education, 2016. dirección: https://www.researchgate.net/publication/299561597_Intelligence_Unleashed_An_argument_for_AI_in_Education.
- [56] Z. Li, J. Zhang, Z. Fei, Y. Feng y J. Zhou, “Addressing Inquiries about History: An Efficient and Practical Framework for Evaluating Open-domain Chatbot Consistency,” en *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021. dirección: <https://aclanthology.org/2021.findings-acl.91.pdf>.
- [57] N. Selwyn, *Should Robots Replace Teachers? AI and the Future of Education*. Cambridge, UK: Polity Press, 2019. dirección: <https://research.monash.edu/en/publications/should-robots-replace-teachers-ai-and-the-future-of-education>.
- [58] B. Williamson, “The social life of AI in education,” *International Journal of Artificial Intelligence in Education*, vol. 34, n.º 1, págs. 97-104, 2023. DOI: 10.1007/s40593-023-00342-5. dirección: <https://doi.org/10.1007/s40593-023-00342-5>.
- [59] A. e. a. Jobin, “The Global Landscape of AI Ethics Guidelines,” *Nature Machine Intelligence*, vol. 1, págs. 389-399, 2019. DOI: 10.1038/s42256-019-0088-2.
- [60] L. e. a. Floridi, “AI4People—An Ethical Framework for a Good AI Society: Opportunities, Risks, Principles, and Recommendations,” *Minds and Machines*, vol. 28, págs. 689-707, 2018. DOI: 10.1007/s11023-018-9482-5.

- [61] N. e. a. Mehrabi, “A Survey on Bias and Fairness in Machine Learning,” *ACM Computing Surveys*, vol. 54, n.º 6, págs. 1-35, 2019. DOI: 10.48550/arXiv.1908.09635. dirección: <https://arxiv.org/abs/1908.09635>.
- [62] R. Binns, “Fairness in Machine Learning: Lessons from Political Philosophy,” en *Proceedings of the 2018 Conference on Fairness, Accountability, and Transparency*, 2018, págs. 149-159. DOI: 10.48550/arXiv.1712.03586. dirección: <https://arxiv.org/abs/1712.03586>.
- [63] F. Doshi-Velez y B. Kim, “Towards a Rigorous Science of Interpretable Machine Learning,” *arXiv preprint arXiv:1702.08608*, 2017. DOI: 10.48550/arXiv.1702.08608. dirección: <https://arxiv.org/abs/1702.08608>.
- [64] Z. C. Lipton, “The Mythos of Model Interpretability,” *Communications of the ACM*, vol. 61, n.º 10, págs. 36-43, 2018. DOI: 10.1145/3233231.
- [65] J. Traxler, “Defining, Discussing, and Evaluating Mobile Learning: The Moving Finger Writes and Having Writ...,” en *International Review of Research in Open and Distributed Learning*, 2007. dirección: <https://www.irrodl.org/index.php/irrodl/article/view/346>.
- [66] UNESCO, *Policy Guidelines for Mobile Learning*, 2013. dirección: <https://unesdoc.unesco.org/ark:/48223/pf0000219641>.
- [67] International Commission on the Futures of Education, “Reimagining Our Futures Together: A New Social Contract for Education,” UNESCO, Paris, inf. téc., 2021, Accedido: 2025-11-04, pág. 186. dirección: <https://unesdoc.unesco.org/ark:/48223/pf0000379707>.
- [68] S. Shrestha et al., “Offline Mobile Learning: Open Platforms for ICT4D,” en *2010 IEEE International Conference on Advanced Learning Technologies (ICALT)*, Disponible en IEEE/Computer Society; resumen en ResearchGate. Accedido: 2025-11-04, 2010. dirección: <https://www.computer.org/csdl/proceedings-article/icalt/2010/4055a703/120mNB1NVOA>.
- [69] Android Developers, *Build an offline-first app*, <https://developer.android.com/topic/architecture/data-layer/offline-first>, Guía técnica práctica sobre arquitectura offline-first. Accedido: 2025-11-04, 2025.
- [70] C. of Learning (COL), *A Mobile App Design Model with Offline Support for the Marginalised Regions*, Caso de diseño / modelo aplicado a regiones marginadas; Accedido: 2025-11-04, 2019. dirección: <https://oasis.col.org/bitstreams/719d6d8f-1d7d-4d3f-83fe-8bac06998095/download>.
- [71] J. Traxler, *Learning in a Mobile Age*. Wolverhampton, UK: International Journal of Mobile y Blended Learning (IJMBL), 2009. DOI: 10.4018/jmb1.2009010101. dirección: <https://www.igi-global.com/gateway/article/2754>.
- [72] H. Crompton y D. Burke, “The use of mobile learning in higher education: A systematic review,” *Computers & Education*, vol. 123, págs. 53-64, 2018. DOI: 10.1016/j.compedu.2018.04.007. dirección: <https://doi.org/10.1016/j.compedu.2018.04.007>.

- [73] T. Bretag et al., “Contract cheating: a survey of Australian university students,” *Studies in Higher Education*, vol. 44, n.º 11, págs. 1837-1856, 2019. DOI: 10.1080/03075079.2018.1462788. dirección: <https://doi.org/10.1080/03075079.2018.1462788>.
- [74] S. E. Eaton, “Postplagiarism: Transdisciplinary ethics and integrity in the age of artificial intelligence and neurotechnology,” *International Journal for Educational Integrity*, vol. 19, n.º 1, págs. 1-10, 2023. DOI: 10.1007/s40979-023-00144-1. dirección: <https://doi.org/10.1007/s40979-023-00144-1>.
- [75] A. S. Tanenbaum y M. Van Steen, *Distributed Systems: Principles and Paradigms*, 2.^a ed. Upper Saddle River, NJ, USA: Pearson Prentice Hall, 2007. dirección: https://vowi.fsinf.at/images/b/bc/TU_Wien-Verteilte_Systeme_V0_%28G%C3%B6schka%29_-_Tannenbaum-distributed_systems_principles_and_paradigms_2nd_edition.pdf.
- [76] K. Hwang y D. Li, “Trusted Cloud Computing with Secure Resources and Data Coloring,” *IEEE Internet Computing*, vol. 14, n.º 5, págs. 14-22, 2010. DOI: 10.1109/MIC.2010.86. dirección: https://www.researchgate.net/publication/220491450_Li_D_Trusted_Cloud_Computing_with_Secure_Resources_and_Data_Coloring_IEEE_Internet_Computing_145_14-22.
- [77] T. Reenskaug, *Models-Views-Controllers*, Technical Note, Accedido: 2025-11-04, 1979. dirección: <https://folk.universitetetioslo.no/trygver/1979/mvc-2/1979-12-MVC.pdf>.
- [78] A. Leiva, *Kotlin for Android Developers: Learn Kotlin the easy way while developing an Android App*. CreateSpace Independent Publishing Platform, 2018, ISBN: 978-1530075614.
- [79] J. S. y D. Law, *Kotlin Programming: The Big Nerd Ranch Guide*. Big Nerd Ranch, 2018, ISBN: 978-0135162361. dirección: https://lmsspada.kemdiktisaintek.go.id/pluginfile.php/751912/course/section/65846/kotlin-programming-the-big-nerd-ranch-guide-1nbsped-013516236x_compress%20%281%29.pdf.
- [80] M. Fowler. “Presentation Model and MVVM.” Accedido: 2025-11-04. dirección: <https://martinfowler.com/eaaDev/PresentationModel.html>.
- [81] S. Chaudhuri y U. Dayal, “An Overview of Data Warehousing and OLAP Technology,” *ACM Sigmod Record*, 1997. dirección: <https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/sigrecord.pdf>.
- [82] OpenAI, *OpenAI API Documentation*, <https://platform.openai.com/docs/concepts>, 2023.
- [83] Google, *Gemini API Documentation*, <https://ai.google.dev/gemini-api/docs?hl=es-419>, 2024.