
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) _____
PhD. Gabriel Barrientos

Fecha de aprobación: Guatemala, noviembre de 2025.

Dedicatoria

A mi madre, ejemplo de superación y esfuerzo desde el inicio de mi vida; la razón de mi persistencia y el motor que ha impulsado cada uno de mis logros.

A mis abuelos, cuyo cariño y apoyo incondicional han dado forma a la persona que soy.

A mi hermano, con la esperanza de que alcance todas las metas que se proponga y encuentre en su hermano mayor el ejemplo y apoyo que necesite en cada etapa de su vida.

A la Fundación Juan Bautista Gutiérrez, por brindarme la oportunidad de desarrollar mi carrera académica más allá de lo que alguna vez imaginé posible.

A mis compañeros de universidad, que mantuvieron el optimismo año tras año, avanzando juntos sin dejar a nadie atrás.

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.

La implementación de esta herramienta incluye el procesamiento de contenidos educativos y el desarrollo de un asistente de IA adaptado a los contenidos seleccionados a través de una aplicación móvil interactiva. Se incluye la validación de la fidelidad de la herramienta a través de comparaciones directas con el material proporcionado. Como resultado, se obtuvo una solución funcional, innovadora y escalable que valida el potencial de la IA para ser implementada en proyectos que busquen 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 inequalities, economic and technological. In countries like Guatemala, where a wide gap persists Educational gap, AI can become a key tool for 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 curricula, are often of moral values that, although included in educational programs, are often addressed in a theoretical way and disconnected from social reality. It is here that The importance of a tool that provides support to Students in reflecting on their role as citizens and in practicing Values such as respect, empathy, and responsibility.

The implementation of this tool includes content processing. educational content processing and the development of an AI assistant tailored to the content selected thru an interactive mobile app. The following is included: Validation of the tool's fidelity thru comparisons. direct comparisons with the provided material. As a result, a solution was obtained. Functional, innovative, and scalable, it validates the potential of AI to be implemented in projects aimed at contributing significantly to strengthening values education in environments with limited resources.

Índice

Dedicatoria	III
Resumen	IV
Abstract	V
Lista de cuadros	IX
Lista de algoritmos	X
Lista de figuras	XI
1. Abreviaturas	XII
2. Alcances	XIV
3. Introducción	1
4. Justificación	3
5. Objetivos	5
5.1. Objetivo general	5
5.2. Objetivos específicos	5
6. Marco teórico	6
6.1. Educación ciudadana y valores	7
6.1.1. Educación en valores	7
6.1.2. Formación ciudadana	7
6.1.3. Competencias cívicas fundamentales	8
6.1.4. Educación moral	9
6.2. Aprendizaje informal y brecha educativa	10
6.2.1. Educación informal	10
6.2.2. Autoformación guiada	10
6.2.3. Brecha educativa y tecnológica	10

6.2.4. Tecnología como herramienta de inclusión educativa	11
6.3. Fundamentos de Inteligencia Artificial en Educación	11
6.3.1. Inteligencia Artificial aplicada a la educación	11
6.3.2. Alucinación en Inteligencia Artificial	11
6.3.3. Modelos Grandes de Lenguaje	12
6.3.4. Arquitectura y funcionamiento de sistemas RAG	12
6.3.5. Bases de datos vectoriales y búsqueda semántica	15
6.3.6. Ingeniería de peticiones y diseño de instrucciones	16
6.3.7. Tutoría personalizada con IA	17
6.3.8. Método socrático aplicado a entornos digitales	17
6.3.9. Métricas de evaluación de chatbots educativos	18
6.3.10. Evaluación de calidad de respuestas en sistemas RAG	18
6.3.11. Congruencia y fundamentación en respuestas educativas	19
6.3.12. Validación técnica sin usuarios finales	19
6.4. Ética y Responsabilidad en la IA Educativa	20
6.4.1. Principios éticos fundamentales en la IA	20
6.4.2. Sesgos algorítmicos y culturales en contextos latinoamericanos	20
6.4.3. Transparencia y explicabilidad en sistemas inteligentes	20
6.4.4. Responsabilidad ante respuestas incorrectas o inadecuadas	21
6.4.5. Privacidad y seguridad en aplicaciones educativas para menores	21
6.4.6. Supervisión pedagógica en sistemas automatizados	21
6.5. Aprendizaje Móvil en Contextos de Recursos Limitados	22
6.5.1. Panorama del aprendizaje móvil en Guatemala y Centroamérica	22
6.5.2. Diseño de experiencias móviles para usuarios con baja alfabetización digital	22
6.5.3. Consideraciones de conectividad intermitente y consumo de datos	22
6.5.4. Aplicaciones móviles en la educación	23
6.6. Tecnologías de Implementación	23
6.6.1. Interfaz de Programación de Aplicaciones (<i>Application Programming Interface - API</i>)	23
6.6.2. Pinecone	23
6.6.3. PostgreSQL	23
6.6.4. NotebookLM	24
6.6.5. Postman	24
6.6.6. Servicio de Almacenamiento Simple (<i>Simple Storage Service o S3</i>)	24
6.6.7. Arquitectura cliente-servidor (<i>frontend/backend</i>)	25
6.6.8. Códigos de estado HTTP	25
6.6.9. Ping	26
6.6.10. Marcos de desarrollo móvil: Kotlin y ecosistema Android	26
6.6.11. Bases de datos vectoriales y su contraste con bases relacionales	33
6.6.12. APIs de IA: integración de modelos conversacionales	34
7. Metodología	35
7.1. <i>Sprint 1</i> : Identificación del perfil de usuario objetivo	36
7.1.1. Ejecución	36
7.2. <i>Sprint 2</i> : Recolección y procesamiento del contenido educativo inicial	37
7.2.1. Ejecución	38
7.3. <i>Sprint 3</i> : Construcción e implementación del servidor	42

7.3.1. Ejecución	43
7.4. <i>Sprint 4:</i> Desarrollo de la interfaz móvil en Kotlin	49
7.4.1. Ejecución	49
7.5. <i>Sprint 5:</i> Pruebas y validación	51
7.5.1. Ejecución	52
7.6. <i>Sprint 6:</i> Documentación y presentación	54
7.6.1. Ejecución	54
8. Resultados	56
8.1. Definición de usuario objetivo (Persona)	56
8.2. Corpus vectorizado	58
8.3. Interfaz móvil en Kotlin (Cliente)	59
8.3.1. Flujo de interacción	59
8.3.2. Funcionalidades Comunes	60
8.3.3. Funcionalidades del Administrador	61
8.4. Pruebas y Validación	62
8.4.1. Métricas Encontradas	63
9. Discusión	64
10. Conclusiones	69
11. Recomendaciones	71
12. Bibliografía	72
13. Anexos	82

Lista de cuadros

1.	Resumen de Contraste: Bases de Datos Relacionales vs. Vectoriales	34
2.	Estructura de los <i>Sprints</i>	35
3.	Métricas de desempeño del sistema	63
A1.	Resultados de las consultas y controles con sus tiempos de respuesta, éxito y congruencia.	82
A2.	Comparación entre referencia esperada y referencia obtenida	88

Lista de algoritmos

1.	Proceso de limpieza y normalización profunda de texto	39
2.	Estandarización de títulos y numeración en texto	39
3.	Validación de integridad de texto	40
4.	Vectorización de fragmentos	41
5.	Controlador de solicitudes	44
6.	Validación de token	44
7.	Flujo completo de procesamiento de preguntas	48
8.	Flujo básico de un Modelo de Vista	50

Lista de figuras

1.	Vista general del enfoque aplicado en el artículo de Lewis et al. [40]	13
2.	Flujo de procesamiento de documentos para generación del corpus.	42
3.	Arquitectura MVC (Modelo, Vista, Controlador).	43
4.	Diagrama entidad-relación (ER) de la base de datos relacional.	45
5.	Flujo de procesamiento de preguntas mediante RAG.	47
6.	Arquitectura MVVM (Modelo, Vista, Modelo de Vista).	49
7.	Ficha de Perfil Persona con datos básicos del usuario objetivo.	57
8.	Vista del índice vectorial en Pinecone con las representaciones numéricas (<i>embeddings</i>) cargadas.	58
9.	Interfaz de la aplicación móvil, opciones generales.	61
10.	Interfaz de la aplicación móvil, opciones de administrador.	62

CAPÍTULO 1

Abreviaturas

- ACID** Atomicity, Consistency, Isolation and Durability (Atomicidad, Consistencia, Aislamiento y Durabilidad). xi
- API** Application Programming Interface (Interfaz de Programación de Aplicaciones). xi
- AWS** Amazon Web Services (Servicios Web de Amazon). xi
- CRUD** Create, Read, Update, Delete (Crear, Leer, Actualizar, Eliminar). xi
- DI** Dependency Injection (Inyección de dependencias). xi
- ER** Entidad-Relación. xi
- HTML** HyperText Markup Language (Lenguaje de Marcado de HiperTexto). xi
- HTTP** HyperText Transfer Protocol (Protocolo de Transferencia de HiperTexto). xi
- IA** Inteligencia Artificial. xi
- ICMP** Internet Control Message Protocol (Protocolo de Mensajes de Control de Internet). xi
- JSON** JavaScript Object Notation (Notación de Objetos de JavaScript). xi
- JWT** JSON Web Token (Token Web en formato JSON). xi
- LLM** Large Language Model (Modelo Grande de Lenguaje). xi
- MVC** Model-View-Controller (Modelo-Vista-Controlador). xi
- MVP** Minimum Viable Product (Producto Mínimo Viable). xi
- MVVM** Model-View-ViewModel (Modelo-Vista-Modelo de Vista). xi

- NFKD** Normalization Form Compatibility Decomposition (Descomposición de Compatibilidad de Formas de Normalización). xi
- NLP** Natural Language Processing (Procesamiento de Lenguaje Natural). xi
- npm** Node Package Manager (Gestor de Paquetes de Node). xi
- OCR** Optical Character Recognition (Reconocimiento Óptico de Caracteres). xi
- RAG** Retrieval-Augmented Generation (Generación Mejorada por Recuperación). xi
- RDS** Relational Database Service (Servicio de Base de Datos Relacional). xi
- S3** Simple Storage Service (Servicio de Almacenamiento Sencillo). xi
- SMS** Short Message Service (Servicio de Mensajes Cortos). xi
- UI** User Interface (Interfaz de Usuario). xi
- UX** User Experience (Experiencia de Usuario). xi
- XML** Extensible Markup Language (Lenguaje de Marcado Extensible). xi

CAPÍTULO 2

Alcances

El proyecto *Compañero Digital: Ocho a Dieciocho* es una iniciativa que busca crear una herramienta de educación cívica basada en inteligencia artificial (IA) dirigida a la juventud guatemalteca. El propósito principal consiste en la construcción de una herramienta conversacional (*chatbot*) especializada, capaz de brindar orientación, información y acompañamiento sobre temas de educación cívica informal, accesible desde una aplicación móvil en cualquier momento y lugar. Como megaproyecto, el objetivo final busca **revitalizar la educación informal cívica en Guatemala** mediante tecnología moderna.

Bajo las expectativas funcionales del megaproyecto principal, el proyecto *Ciudadano Digital* constituye una base técnica inicial, centrada en demostrar la viabilidad de las herramientas tecnológicas contempladas para el desarrollo del producto final. Se busca definir la arquitectura base que permita el procesamiento del contenido educativo, la interacción con el usuario y un sistema independiente que pueda ser consumido por una aplicación móvil, pero a su vez esté disponible para cualquier otro tipo de implementación (por ejemplo, una página web). A su vez, se realiza el perfilamiento base de un ejemplo de usuario final objetivo, con el fin de mantener el enfoque principal del megaproyecto original. Cabe destacar que la implementación final requerirá de un volumen grande de documentos, es decir, un corpus amplio de información, para garantizar que el sistema pueda brindar respuestas completas y precisas.

Al tratarse meramente de una validación técnica, este proyecto no contempla pruebas de campo con usuarios reales. Las respuestas brindadas por el modelo se validan únicamente en comparación con los contenidos utilizados para la alimentación del conocimiento del mismo. Sin embargo, sí se valida la integración adecuada entre los distintos componentes tecnológicos que constituyen el producto final: modelo grande de lenguaje (LLM, por sus siglas en inglés), base de datos vectorial, base de datos relacional y aplicación móvil final.

Ciudadano Digital se centra únicamente en la construcción del Producto Mínimo Viable (MVP, por sus siglas en inglés) necesario para comprobar que la arquitectura propuesta puede operar de manera funcional y coherente. Se evalúa el procesamiento del corpus, el flujo de generación mejorada por recuperación (RAG, por sus siglas en inglés) y la conexión entre

la aplicación móvil y el servidor desarrollado, sin abordar aún la implementación completa de los componentes pedagógicos, escalabilidad institucional ni las funciones avanzadas previstas para el proyecto final.

CAPÍTULO 3

Introducción

La formación en valores y ciudadanía constituye un componente esencial en la construcción de sociedades democráticas, inclusivas y participativas. A través de ella, la población desarrolla 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 grandes de lenguaje (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 grande de lenguaje, conectado a una base de datos vectorial con materiales educativos, éticos y contextuales, genera respuestas personalizadas y fundamentadas que orientan la reflexión. No se busca sustituir al docente, sino complementar su labor mediante un acompañamiento continuo que promueva la autonomía moral, la empatía y la responsabilidad social a través de un diseño que prioriza la accesibilidad y la escalabilidad.

Si bien el alcance de este proyecto no contempla la realización de pruebas de campo con usuarios reales de la aplicación final, se buscó demostrar la viabilidad técnica y conceptual del mismo. Se propone como una base técnica sobre la cual se puedan implementar herramientas más complejas y especializadas, con el apoyo pertinente de expertos en el área deseada y la retroalimentación continua de usuarios reales.

CAPÍTULO 4

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 grandes de lenguaje (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 de generación mejorada por recuperación permite que estos sistemas fundamenten sus respuestas en documentos verificados, lo que garantiza 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, el enfoque de *Ciudadano Digital* está alineado con cubrir los espacios de la educación no cubiertos por el sistema educativo estructurado del país. De esta manera, los estudiantes pueden encontrar en la herramienta un motivante independiente que les otorgue un acompañamiento adicional en su vida diaria, alineado con la búsqueda de la interpretación y aplicación individual de conocimientos teóricos recibidos en el aula.

Finalmente, la elección de desarrollar la aplicación únicamente para el sistema operativo Android responde a criterios de accesibilidad y equidad tecnológica. Diversos estudios muestran que Android posee una presencia dominante en países de ingresos bajos y medios debido a la amplia disponibilidad de dispositivos de bajo costo, posibilitada por su carácter de código abierto y la diversidad de fabricantes [18]. En América Latina, esta tendencia es aún más marcada: los teléfonos Android representan la mayoría del mercado móvil y constituyen, para muchos usuarios, la forma principal de acceso a internet y a recursos digitales educativos [19]. Esta predominancia es especialmente relevante en comunidades con recursos limitados, donde los dispositivos Android suelen ser significativamente más asequibles que alternativas como iOS, facilitando así la adopción de soluciones tecnológicas inclusivas. Centrar el desarrollo a dispositivos bajo el sistema operativo Android, maximiza el alcance potencial del proyecto y asegura que su impacto educativo se oriente a los sectores que más requieren herramientas accesibles para el aprendizaje y la formación ciudadana.

CAPÍTULO 5

Objetivos

5.1. Objetivo general

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

5.2. Objetivos específicos

- Determinar si la aplicación de LLM pre-entrenados se puede adaptar a un ambiente de aprendizaje informal.
- 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 6

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 [20, 21].

Desde esta convergencia entre tecnología y formación ética, la literatura reciente destaca el potencial de los modelos grandes de lenguaje (LLMs, por sus siglas en inglés) para generar entornos conversacionales que promuevan la reflexión moral y la toma de decisiones fundamentadas [22, 23]. 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.

6.1. 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 [24, 25]. 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 [26].

6.1.1. 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 [27].

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 [27].

6.1.2. 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. [28]

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 [28].

6.1.3. 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 [29].

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 [29].

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 [29].

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. [29]

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 [29].

6.1.4. 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 [30].

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 [30].

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 [30].

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 [30].

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. [30]

6.2. 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 [31, 32].

6.2.1. 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. [32]

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 [32].

6.2.2. 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 [32].

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 [32].

6.2.3. 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, 33].

6.2.4. 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, 34].

6.3. 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 [35, 36].

6.3.1. 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, 21].

6.3.2. Alucinación en Inteligencia Artificial

Este concepto se refiere a la generación de información incorrecta, falsa o inconsistente por parte de modelos de lenguaje, incluso cuando estos parecen confiables y coherentes. Este fenómeno ocurre principalmente debido a la forma en que los modelos aprenden a partir de grandes volúmenes de datos: no comprenden la realidad de manera literal, sino que predicen la siguiente palabra más probable según patrones estadísticos aprendidos. Como resultado, pueden combinar información de manera inapropiada, inferir hechos inexistentes o extrapolar contenido de forma errónea. La alucinación es especialmente crítica en aplicaciones donde la precisión es esencial, como la educación, la medicina o la generación de informes, ya que puede inducir a errores o desinformación si no se detecta y corrige adecuadamente [37].

6.3.3. Modelos Grandes de Lenguaje

Los modelos grandes de lenguaje (*Large Language Models* o LLMs) 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 [38, 39].

6.3.4. Arquitectura y funcionamiento de sistemas RAG

La Generación Mejorada por Recuperación (RAG, por sus siglas en inglés) combina modelos grandes de lenguaje (LLMs) con motores de recuperación de información, 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*» por Lewis et al, 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. La generación mejorada por recuperación amplía las capacidades de tutoría digital al integrar conocimiento externo con generación de lenguaje natural [40, 41].

Representaciones numéricas (*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 comparan y recuperen información de manera eficiente, lo que permite medir la similitud entre conceptos y facilita búsquedas semánticas. En educación, estas representaciones numéricas permiten vincular preguntas de los estudiantes con contenidos relevantes, apoyando la personalización del aprendizaje [42, 43].

Arquitectura de Generación Mejorada por Recuperación

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 representaciones numéricas (*embeddings*) 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 de Lewis et al, se presenta una figura (reproducida en la Figura 1) que 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.

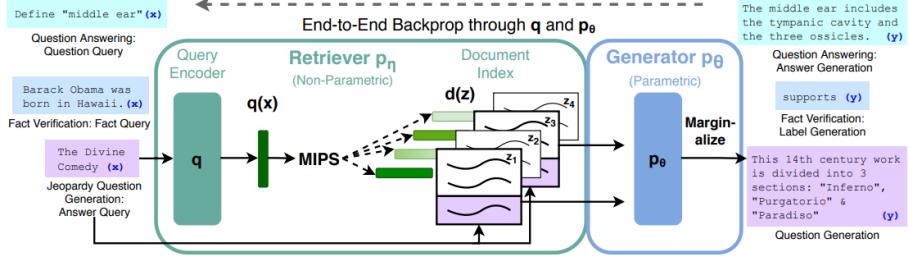


Figura 1: Vista general del enfoque aplicado en el artículo de Lewis et al. [40]

La figura 1 ilustra la arquitectura general de un sistema de recuperación y generación entrenado de extremo a extremo. El proceso inicia con una **consulta de entrada** x , por ejemplo, solicitar la respuesta a una pregunta. Esta consulta es procesada por un **codificador de consultas (Query Encoder)**, el cual transforma x en una representación densa $q(x)$.

A continuación, esta representación es enviada al **módulo recuperador no paramétrico (p_η)**, el cual opera sobre un índice de documentos. El recuperador utiliza *Maximum Inner Product Search (MIPS)* para identificar los documentos cuyas representaciones $d(z)$ presentan mayor similitud con el vector de consulta. El índice contiene múltiples documentos representados en forma vectorial y el recuperador selecciona los más relevantes, denotados como z_1, z_2, z_3, \dots

Los documentos seleccionados se redireccionan hacia el **generador paramétrico (p_θ)**, el cual recibe tanto la representación de la consulta como la de los documentos recuperados. Este generador produce una distribución sobre posibles salidas, es decir, posibles respuestas a la pregunta solicitada.

Finalmente, el sistema realiza la **marginalización** sobre todos los documentos recuperados, combinando las contribuciones de cada uno de ellos para producir la salida final y . Esta salida es la que toma la forma de la respuesta completa a la pregunta original.

2. Fase de inferencia: Al recibir una consulta:

- Se genera una representación numérica (*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 una petición contextualizada que combina la pregunta original con los fragmentos recuperados.
- El LLM genera una respuesta basándose exclusivamente en el contexto dado o tomándolo como guía (el enfoque depende de la estructura utilizada para construir la petición) 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 artículo de Lewis et al, se menciona la diferencia de utilizar una **RAG por secuencia** (el modelo selecciona un único documento sobre el cual basará su respuesta) frente al enfoque de **RAG por 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 [44]:

- **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.
- **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.
- **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.

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.

Ventajas de RAG en educación

La implementación de sistemas de generación mejorada por recuperación en enfoques educativos, brinda varias ventajas orientadas al uso de modelos de inteligencia artificial [45]:

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

- 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.
- **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.
- 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.

Desafíos conocidos

A pesar de sus beneficios, la implementación de generación mejorada por recuperación también presenta ciertos retos a cubrir en proyectos educativos de alto impacto [46]:

- **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.
- **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 deje de tener sentido, o bien, que una consulta que sí está relacionada con el contenido del corpus no pueda ser respondida.
- **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 la petición construida también cuenta con limitaciones de longitud y, por lo tanto, del nivel de especificación y claridad exigido al modelo.
- **Dificultad para sintetizar información de múltiples fragmentos dispersos** Al combinar varios fragmentos en una sola petición, si no se ha seleccionado el corpus cuidadosamente, se puede incurrir en contradicciones o redundancia en las consultas.

6.3.5. Bases de datos vectoriales y búsqueda semántica

Las bases de datos vectoriales permiten almacenar y consultar representaciones numéricas (*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 [47, 48].

6.3.6. Ingeniería de peticiones y diseño de instrucciones

La ingeniería de peticiones consiste en el diseño de instrucciones efectivas para guiar el comportamiento de modelos de lenguaje hacia objetivos específicos. [49] 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 una petición efectiva

Al construir una petición que será recibida por un modelo grande de lenguaje (LLM), se deben tomar en cuenta varios componentes que, en conjunto, serán los que definan la interacción entre el modelo y el usuario. El proceso de ingeniería de peticiones consiste en optimizar el lenguaje utilizado en la instrucción, enfocado en guiar el comportamiento de los modelos de lenguaje y obtener el mejor desempeño posible. Esto implica entender que los modelos procesan las instrucciones como programas en lenguaje natural, por lo que la precisión y coherencia de las respuestas se ven afectadas directamente por la estructura y redacción de cada parte de la petición. [50]

Los componentes más comunes son:

- **Rol (Persona):** Si bien es un componente ampliamente discutido, corresponde a la sección de la petición que permite asignar un rol específico a la inteligencia artificial (IA). A través de la definición de un rol específico, se determina el tono de las respuestas generadas y los límites que debe cumplir en su interacción con el usuario.
- **Información Adicional:** Comunmente identificado también como el **contexto** de la petición. Se refiere a todo el contenido adicional que la IA debe tomar en cuenta para generar la respuesta, ya sea interacciones previas, o bien, el contenido que delimita la respuesta final (por ejemplo, a través de un sistema de generación mejorada por recuperación).
- **Directiva:** Define la intención central de la petición. Aquí puede ir la pregunta, solicitud de información o generación de contenido multimedia (si aplica).
- **Formato de salida:** Comunmente, es deseable que la IA generativa sea capaz de devolver la información solicitada en un formato o patrón específico. Este aspecto también incluye las **instrucciones de estilo**, las cuales son un tipo de formato de salida utilizado para modificar la salida estilísticamente en lugar de estructuralmente (por ejemplo, solicitar una respuesta apropiada para una persona de edad específica).
- **Ejemplos (opcional):** Los ejemplos, también conocidos como **tomas**, actúan como demostraciones explícitas que guían a la IA para lograr un resultado específico para la tarea indicada. Este aspecto es especialmente útil en técnicas como el aprendizaje en contexto.

Estrategias en contextos educativos

Con el objetivo de enfocar el diseño de peticiones 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 [51].

Algunas de las estrategias más comunes son:

- ***Chain-of-thought prompting* (Petición de Cadena de Pensamiento):** Solicita al modelo que describa el razonamiento que utilizó para responder la pregunta, se exige el paso a paso de cómo llegó hasta la respuesta brindada.
- ***Socratic prompting* (Petición Socrática):** 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.
- ***Constitutional AI* (IA Constitucional):** Incorpora principios éticos en las instrucciones, por ejemplo, la omisión de palabras o temas sensibles.
- ***Retrieval-aware prompting* (Petición Consciente de Recuperación):** 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.

6.3.7. 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 [36, 52].

6.3.8. 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 [36, 53].

6.3.9. 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. [54]

Métricas de calidad de respuesta

En el diseño y evaluación de sistemas conversacionales, la calidad de la respuesta se relaciona con la capacidad de la herramienta para comprender la intención del usuario, mantener la coherencia del diálogo y producir respuestas naturales y adecuadas. La efectividad de un chatbot depende de componentes como el análisis de mensajes, la gestión del diálogo y la generación del lenguaje natural, los cuales influyen directamente en la calidad percibida por el usuario. [55]

Varias de las métricas más utilizadas son:

- **Relevancia del contexto:** Evalúa qué tan pertinentes son los documentos recuperados para la consulta inicial. [56].
- **Fidelidad de la respuesta:** Mide si la salida generada está fundamentada en la evidencia recuperada, evitando alucinaciones.[56].
- **Relevancia de la respuesta:** Valora si la respuesta aborda adecuadamente la consulta del usuario [56].

Métricas de evaluación técnica

Para evaluar sistemas de RAG desde una perspectiva técnica, se consideran métricas que capturan el rendimiento operacional del sistema en escenarios prácticos.[57]

Las más comunes son:

- **Tasa de éxito:** Porcentaje de consultas en las que el sistema genera respuestas correctas y apropiadas. Esta métrica evalúa la efectividad global del sistema en tareas específicas, siendo fundamental para aplicaciones donde la precisión es crítica [58, 59].
- **Latencia:** Tiempo total transcurrido desde el envío de la consulta hasta la recepción de la respuesta completa, incluyendo las fases de recuperación y generación. Esta métrica es crucial para la experiencia de usuario en aplicaciones interactivas y sistemas en tiempo real [60].

6.3.10. Evaluación de calidad de respuestas en sistemas RAG

Los sistemas que implementan RAG presentan desafíos específicos de evaluación relacionados con la recuperación y síntesis de información. 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 [40].

Fidelidad al contenido base (Faithfulness)

En el contexto de sistemas RAG educativos, la fidelidad evalúa si la respuesta se limita estrictamente a la información contenida en los documentos recuperados, evitando alucinaciones o afirmaciones no sustentadas por el corpus proporcionado. [61]

Esta métrica es particularmente crítica en contextos educativos, donde la precisión fáctica y la trazabilidad de la información son requisitos fundamentales. Se mide comparando el contenido de las respuestas generadas con el material documentado en el corpus base, verificando que toda afirmación esté respaldada por los documentos fuente. [61]

Relevancia de la respuesta (Answer Relevance)

La 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. Esta métrica evalúa la pertinencia contextual de la respuesta respecto a la consulta original. [62]

En sistemas educativos, una respuesta puede ser técnicamente correcta y basada en el contenido, pero si no aborda adecuadamente la pregunta del usuario, pierde su valor pedagógico. [62]

6.3.11. Congruencia y fundamentación en respuestas educativas

En contextos educativos, además de la precisión técnica, se espera que los sistemas basados en modelos de lenguaje mantengan consistencia con el conocimiento verificado y respondan de manera fundamentada. En este sentido, la **congruencia fáctica** puede entenderse como el grado en que las respuestas del sistema son coherentes con los hechos documentados en una base de conocimiento estructurada, evitando errores o afirmaciones sin evidencia empírica [63].

El **nivel de congruencia fáctica** puede calcularse como el porcentaje de consultas en las que el sistema responde correctamente de acuerdo con el conocimiento base. Esta medida complementa las métricas tradicionales de exactitud o fidelidad al evaluar específicamente la capacidad del sistema para mantenerse dentro de los límites del conocimiento validado, evitando desviaciones o **alucinaciones** que puedan inducir a error en entornos educativos [63].

6.3.12. Validación técnica sin usuarios finales

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. [64]

Este enfoque de validación se basa en la definición de conjuntos de prueba estructurados que incluyen tanto preguntas dentro del dominio esperado como preguntas de control fuera del alcance del sistema. La validación basada en el corpus verifica que las respuestas provengan efectivamente de los materiales fuente y que no contengan información ajena al contenido educativo. [35]

Si bien este tipo de validación no mide el impacto pedagógico real ni el aprendizaje logrado, permite demostrar la factibilidad técnica y coherencia funcional del sistema antes de proceder a estudios con usuarios reales. [35, 52, 64]

6.4. Ética y Responsabilidad en la IA Educativa

La ética en la IA educativa aborda la responsabilidad en el diseño, implementación y uso de sistemas inteligentes en contextos de aprendizaje, lo cual 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 [65, 66].

6.4.1. Principios éticos fundamentales en la IA

Los principios éticos fundamentales en la 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 [67, 68].

6.4.2. 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 conjuntos de datos y pruebas de equidad en los resultados. La prevención de sesgos asegura que todos los estudiantes reciban oportunidades de aprendizaje justas y equitativas [69, 70].

6.4.3. 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 [71, 72].

6.4.4. 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 [67, 68].

La 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 [67, 68].

6.4.5. 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 constituye un activo sensible que debe ser gestionado bajo principios de minimización de datos, consentimiento informado y almacenamiento seguro [65, 66].

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 [65, 66].

6.4.6. 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 [35, 64].

6.5. 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 [73, 74].

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. [74]

6.5.1. 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].

6.5.2. 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 [75].

6.5.3. 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 [74, 76, 77].

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 [78].

6.5.4. 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 [79, 80].

6.6. 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 complemente el aprendizaje sin reemplazar la reflexión y el esfuerzo personal [81, 82].

6.6.1. Interfaz de Programación de Aplicaciones (*Application Programming Interface - API*)

Las **APIs** permiten que dos sistemas de software se comuniquen entre sí mediante un conjunto definido de reglas, conocidos como *endpoints* o **puntos de conexión**. Según la definición clásica de Fielding y Taylor, una API proporciona un mecanismo estandarizado para que aplicaciones distintas intercambien datos o funcionalidades sin necesidad de conocer la implementación interna de cada sistema [83]. En el ámbito de la inteligencia artificial, las APIs ofrecen acceso directo a modelos avanzados sin requerir que el desarrollador entrene o despliegue modelos por su cuenta.

6.6.2. Pinecone

Pinecone es una base de datos vectorial administrada, en la nube y diseñada para búsquedas semánticas a gran escala. Permite almacenar, indexar y consultar representaciones numéricas (*embeddings*) de alta dimensionalidad con latencias muy bajas. Su arquitectura está optimizada para tareas como generación mejorada por recuperación (RAG), sistemas de recomendación, clasificación semántica y búsqueda inteligente. Proporciona escalabilidad automática, indexación eficiente y una API sencilla para integración en aplicaciones de aprendizaje de máquinas (*machine learning*) [84].

6.6.3. PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos relacional de código abierto; reconocido por su robustez, flexibilidad y estricto cumplimiento de estándares SQL. Está

diseñado para manejar cargas de trabajo complejas, consultas avanzadas y grandes volúmenes de datos, a la vez que ofrece características como transacciones ACID (siglas en inglés para Atomicidad, Consistencia, Aislamiento y Durabilidad), extensibilidad mediante funciones personalizadas, índices especializados y soporte para tipos de datos avanzados. Su arquitectura orientada a la confiabilidad y la integridad de los datos lo convierte en una opción preferida en aplicaciones empresariales, científicas y de software moderno que requieren estabilidad y alto rendimiento [85].

6.6.4. NotebookLM

NotebookLM es una herramienta de inteligencia artificial que permite a los usuarios «conversar» con sus propios documentos mediante técnicas de modelos grandes de lenguaje. A diferencia de un motor de búsqueda, NotebookLM analiza los textos cargados por el usuario y construye una base de conocimiento personalizada. A partir de esto, puede generar resúmenes, responder preguntas complejas sobre el contenido, extraer ideas clave, organizar la información en guías de estudio, e incluso convertir textos extensos en resúmenes de audio o notas estructuradas. [86]

Esta herramienta resulta especialmente útil para estudiantes, investigadores o profesionales que manejan gran cantidad de información, ya que facilita la comprensión y navegación de textos largos o complejos, reduciendo el tiempo necesario para extraer lo esencial. Además, al apoyarse únicamente en los documentos proporcionados por el usuario, NotebookLM ayuda a mantener la precisión y trazabilidad de la información, lo que puede mitigar el riesgo de errores comunes en modelos de lenguaje no especializados. :contentReference[oaicite:2]index=2

6.6.5. Postman

Postman es una herramienta de desarrollo utilizada para diseñar, probar y documentar APIs. Ofrece una interfaz gráfica intuitiva para enviar solicitudes HTTP (siglas en inglés para Protocolo de Transferencia de HiperTexto), revisar respuestas, automatizar pruebas, organizar colecciones de puntos de conexión y colaborar en equipos de desarrollo. Permite trabajar con distintos métodos HTTP, autenticación, variables de entorno y generación automática de documentación. [87]

6.6.6. Servicio de Almacenamiento Simple (*Simple Storage Service* o S3)

S3 es un servicio de almacenamiento de objetos altamente escalable, ofrecido por los servicios web de Amazon (*Amazon Web Services* o AWS). Permite guardar y recuperar cualquier cantidad de datos desde cualquier lugar a través de Internet, proporcionando durabilidad de casi el 100 %, alta disponibilidad y control granular de acceso. S3 organiza los datos en contenedores llamados «*buckets*» y ofrece diversas clases de almacenamiento optimizadas para distintos niveles de costo y frecuencia de acceso. [88]

6.6.7. 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 la 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 cliente se encarga de la interfaz y la experiencia de usuario, mientras que el servidor gestiona la lógica, la seguridad y la integración con IA y bases de datos [89, 90].

Patrón de diseño MVC

El patrón de diseño **Modelo-Vista-Controlador** (*Model-View-Controller* o 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. [91]

En este patrón, el **Modelo** contiene los datos y la lógica de negocio; la **Vista** es responsable de mostrar la información al usuario; y el **Controlador** 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. [91]

El patrón MVC tuvo sus orígenes en el entorno *Smalltalk-80* y ha sido ampliamente adoptado en múltiples marcos (*frameworks*) modernos de desarrollo, tanto web como de escritorio, esto debido a su capacidad de estructurar aplicaciones complejas de forma eficiente [91].

NodeJS como plataforma para el desarrollo del lado del servidor

NodeJS es una plataforma de ejecución de JavaScript basada en un modelo orientado a eventos y operaciones no bloqueantes, lo que permite manejar múltiples conexiones concurrentes con alta eficiencia y baja latencia. Su extenso ecosistema de paquetes mediante npm (Gestor de Paquetes de Node, por sus siglas en inglés) facilita integrar funcionalidades para redes, bases de datos, autenticación y servicios web, convirtiéndolo en una opción ampliamente adoptada para construir APIs (Interfaces de Programación de Aplicaciones), aplicaciones en tiempo real y arquitecturas modernas de microservicios. [92]

6.6.8. Códigos de estado HTTP

Son respuestas numéricas estandarizadas que un servidor envía al cliente para indicar el resultado de una solicitud, realizada mediante el protocolo HTTP. Estos códigos permiten identificar si la petición fue exitosa, si requiere acciones adicionales, si hubo errores del cliente o del servidor, o si existen problemas de redirección. Se organizan en cinco categorías principales: respuestas informativas (de 100 a 199), exitosas (de 200 a 299), redirecciones

(de 300 a 399), errores del cliente (de 400 a 499) y errores del servidor (de 500 a 599). Su uso adecuado facilita la comunicación clara entre aplicaciones y servicios web, lo que permite un manejo correcto de errores y una interacción más confiable en arquitecturas distribuidas [93].

6.6.9. Ping

Esta es una herramienta de diagnóstico de red que permite verificar la conectividad entre dos dispositivos mediante el envío de mensajes ICMP (siglas en inglés para Protocolo de Mensajes de Control de Internet, protocolo utilizado para diagnosticar problemas de comunicación). Su propósito principal es determinar si un dispositivo es accesible, así como medir parámetros como el tiempo de ida y vuelta (RTT o *Round-Trip Time*) y la pérdida de paquetes. Ping es ampliamente utilizado para identificar problemas de red básicos, ya que ayuda a detectar fallos de comunicación, latencias inusuales o interrupciones en la ruta entre origen y destino. Su funcionamiento se basa en el protocolo ICMP, definido en los estándares fundamentales de Internet [94].

6.6.10. Marcos 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 marcos de IA para educación digital [95, 96].

Patrón de diseño MVVM

El patrón de diseño **Modelo-Vista-Modelo de Vista** (*Model-View-ViewModel* o 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 mantenibilidad del código. [97]

En este patrón, el **Modelo** encapsula los datos y reglas de negocio; la **Vista** representa la interfaz de usuario; y el **Modelo de Vista** 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. [97]

Actividades (*Activities*)

En Android, una Actividad o *Activity* es uno de los componentes fundamentales de una aplicación y representa una única pantalla con la que el usuario puede interactuar. Cada Actividad administra su propio ciclo de vida, el cual incluye estados como creación, inicio, pausa, reanudación y destrucción. Son el componente encargado de renderizar la interfaz de usuario, manejar eventos y coordinar la navegación hacia otras partes de la aplicación. Las

Activities actúan como puntos de entrada principales y permiten organizar la aplicación de forma modular, definiendo flujos independientes dentro del sistema. [98]

Fragments (*Fragmentos*)

En Android, un Fragmento o *Fragment* es un componente modular de la interfaz de usuario que representa una parte reutilizable de una Actividad. Cada Fragmento posee su propio ciclo de vida, su propia lógica y su propio diseño, pero siempre existe dentro del contexto de una Actividad que lo hospeda. Los Fragmentos permiten construir interfaces flexibles y adaptables, especialmente en pantallas grandes, dividiendo una Actividad en múltiples secciones que pueden combinarse dinámicamente, reemplazarse o reutilizarse en diversos flujos de navegación. Su uso facilita el diseño responsivo, la reutilización de componentes y la separación de responsabilidades dentro de una aplicación Android. [99]

Android Jetpack

Android Jetpack es un conjunto de bibliotecas, herramientas y componentes de arquitectura diseñados para facilitar el desarrollo de aplicaciones Android modernas, robustas y mantenibles. Organizado en cuatro pilares: Arquitectura (*Architecture*), Interfaz de Usuario (*User Interface* o *UI*), Comportamiento (*Behavior*) y base (*Foundation*). Jetpack proporciona soluciones listas para usar que simplifican tareas comunes como manejo del ciclo de vida, persistencia de datos, navegación, inyección de dependencias y diseño de interfaces reactivas. Sus componentes son modulares, retrocompatibles y siguen prácticas recomendadas, lo que permite a los desarrolladores construir aplicaciones más escalables, seguras y con menos código repetitivo. [100]

AppCompat

AppCompat es una librería de compatibilidad de Android Jetpack que permite utilizar componentes modernos de la interfaz de usuario en versiones antiguas del sistema operativo. Proporciona implementaciones retrocompatibles de elementos clave y comportamientos visuales actualizados, lo que garantiza una apariencia coherente y funcionalidades recientes incluso en dispositivos con versiones anteriores del sistema operativo. AppCompat es esencial para mantener compatibilidad retroactiva y asegurar que una misma aplicación funcione de forma uniforme en una amplia gama de dispositivos Android. [101]

Componente de Navegación (*Navigation Component*)

Este es una parte de Android Jetpack diseñada para gestionar la navegación dentro de una aplicación Android de forma estructurada, segura y declarativa. Permite definir en un único gráfico (llamado gráfico de navegación o *Navigation Graph*) todos los destinos y acciones de navegación, facilitando el manejo de transiciones entre Fragmentos, Actividades y pantallas desplegables. Ofrece funcionalidades como animaciones de transición, paso de

argumentos tipados (*SafeArgs*), y control de la pila de navegación (*back stack*) sin necesidad de manipular transiciones manuales entre fragmentos. Esto reduce errores comunes, simplifica el código y promueve arquitecturas más limpias y mantenibles. [102]

NavHostFragment

Es un contenedor especializado del Componente de Navegación de Android Jetpack que actúa como el anfitrión donde se muestran los destinos definidos en el gráfico de navegación. Es el componente responsable de gestionar los cambios de pantalla al navegar entre Fragmentos, controlar automáticamente el ciclo de vida de cada destino, la pila de navegación, las transiciones y la integración con el controlador de navegación (*NavController*). En esencia, sirve como el punto central que conecta la interfaz de usuario con la lógica de navegación declarada, permitiendo manejar flujos complejos sin necesidad de administrar manualmente transacciones de fragmentos. [103]

NavigationFragment

Al utilizar Android Jetpack, este suele ser el término para definir un fragmento que funciona como destino dentro del Componente de Navegación, aunque no existe una clase oficial con ese nombre en la librería. En la práctica, un *NavigationFragment* es cualquier fragmento diseñado para integrarse con el gráfico de navegación y ser gestionado por un *NavHostFragment*. Estos fragmentos participan en flujos de navegación declarativos, reciben argumentos tipados (por medio de *SafeArgs*), controlan su propio ciclo de vida y pueden iniciar acciones de navegación mediante un *NavController*. [102]

NavController

Este es el componente central del Componente de Navegación de Android Jetpack, encargado de orquestar las acciones de navegación dentro de una aplicación. Actúa como intermediario entre el *NavHostFragment* y el gráfico de navegación, ya que interpreta las acciones declaradas en el gráfico y ejecuta las transiciones correspondientes entre destinos. Además, administra la pila de navegación, permite navegar hacia destinos específicos, manejar argumentos tipados (*SafeArgs*) y coordinar animaciones o comportamientos especiales definidos en la navegación. [104]

SafeArgs

Este es un complemento del Componente de Navegación de Android Jetpack, el cual permite pasar argumentos entre destinos de manera tipada y segura en tiempo de compilación. Genera clases y métodos automáticamente a partir del gráfico de navegación, lo que evita errores comunes asociados al uso de paquetes (*Bundle*) manuales y proporciona una API clara. Con *SafeArgs*, se permite el envío de datos entre Fragmentos y Actividades usando objetos generados, lo que garantiza que los tipos coincidan, que los argumentos requeridos estén presentes y que se reduzca significativamente el riesgo de fallos en la navegación. [105]

Serializador (*Serializable*)

Es una interfaz de Java que permite convertir un objeto en una secuencia de *bytes* con el fin de almacenarlo o transmitirlo. En Android, aunque es compatible, no es el método más eficiente para pasar datos entre componentes, ya que su rendimiento es inferior al de *Parcelable*. Su principal ventaja es la simplicidad: basta con implementar la interfaz para habilitar la serialización automática del objeto. [106]

Parcelador (*Parcelable*)

Es una interfaz específica de Android diseñada para serializar objetos de manera más rápida y eficiente que el Serializador (*Serializable*). Permite descomponer un objeto en un paquete llamado *Parcel* para transferirlo entre Actividades, Fragmentos o servicios. Requiere implementar métodos explícitos para escribir y reconstruir el objeto, lo que reduce la sobrecarga y mejora el rendimiento, especialmente en dispositivos móviles donde la optimización es clave. [107]

Paquete (*Bundle*)

En Android, un *Bundle* es una estructura de datos de tipo llave-valor, utilizada para almacenar y transferir información entre componentes del sistema. Está optimizado para manejar tipos de datos primitivos y objetos que implementan *Parcelable* o *Serializable*. Los *Bundles* se emplean comúnmente para pasar argumentos en la navegación, preservar el estado durante cambios de configuración y enviar datos en eventos del ciclo de vida. Constituyen un mecanismo ligero y eficiente que permite empaquetar información de forma temporal dentro de la arquitectura de componentes de Android. [108]

Material Design

Corresponde a un sistema de diseño creado por **Google**, el cual proporciona una guía integral para crear interfaces de usuario consistentes, accesibles y visualmente atractivas. Se basa en principios como la metáfora del material, animaciones significativas, jerarquías claras y uso intencional de colores y tipografías. En Android, sus componentes están disponibles a través de la librería *Material Components for Android*, la cual ofrece *widgets* o complementos modernos, patrones de navegación y estilos personalizables para construir experiencias coherentes en todo el ecosistema Android. [109]

Lenguaje de Marcado Extensible (*Extensible Markup Language - XML*)

Este es un lenguaje diseñado para almacenar y transportar datos de forma estructurada y legible tanto para humanos como para máquinas. A diferencia de HTML (*HyperText Markup Language* o Lenguaje de Marcado de Hipertexto), su propósito no es describir la presentación, sino definir la estructura y el significado de la información mediante etiquetas

personalizables. XML es extensible, jerárquico y basado en texto, lo que lo convierte en un estándar ampliamente utilizado para intercambio de datos, configuraciones y definición de documentos. En Android, XML se utiliza para describir interfaces de usuario, recursos de diseño, valores, animaciones y configuraciones del sistema. [110]

Plantillas con XML

En Android, las plantillas con XML se refieren a la definición de interfaces de usuario mediante archivos XML que describen la estructura, diseño y atributos visuales de cada pantalla o componente. Estos archivos especifican elementos como elementos de texto, botones, listas, así como sus propiedades (márgenes, tamaños, colores, disposiciones, etc.). El uso de XML permite separar la lógica del diseño, mantener una arquitectura más limpia y reutilizar componentes de interfaz. **Android Studio** proporciona herramientas visuales y plantillas predefinidas que facilitan su creación y edición. [111]

ConstraintLayout

ConstraintLayout es un administrador de diseño (*layout manager*) de Android altamente flexible que permite posicionar y organizar vistas mediante restricciones. Estas restricciones definen relaciones entre vistas o entre una vista y su contenedor, lo que permite crear interfaces complejas y responsivas sin necesidad de anidar múltiples plantillas. Ofrece un excelente rendimiento y herramientas visuales como el *Layout Editor* (editor de plantilla), que facilitan la construcción de diseños adaptados a distintas resoluciones y orientaciones de pantalla. [112]

Glide

Se trata de una librería de carga y manejo de imágenes para Android que facilita la descarga, almacenamiento en caché y visualización eficiente de imágenes, ya sea desde la web, recursos locales o almacenamiento interno. Está optimizada para rendimiento y uso de memoria, para ofrecer integración con listas, animaciones suaves y soporte para imágenes GIF (siglas en inglés para Formato de Intercambio de Gráficos). Su API simple permite transformar imágenes, redimensionarlas y cargarlas de forma asíncrona, por lo que es una herramienta ampliamente utilizada en aplicaciones modernas. [113]

Base de Datos *Room*

Room es una biblioteca de persistencia de datos de Android Jetpack que proporciona una capa de abstracción sobre *SQLite*, lo que facilita el uso de bases de datos locales de forma segura y eficiente. Ofrece validación en tiempo de compilación para consultas SQL (siglas en inglés para Lenguaje de Consulta Estructurada), integración fluida con datos observables (*LiveData*), mapeo automático de entidades, y un ORM (siglas en inglés para Mapeo Objeto-Relacional) ligero que reduce el código repetitivo. *Room* promueve buenas

prácticas al asegurar operaciones seguras en hilos separados y facilitar migraciones y manejo estructurado de datos persistentes. [114]

Mapeo Objeto-Relacional (*Object-Relational Mapping*)

Esta es una técnica que permite relacionar objetos de un lenguaje de programación orientado a objetos con tablas de una base de datos relacional, lo que automatiza la conversión entre ambos modelos. Esto facilita el acceso y la manipulación de datos sin necesidad de escribir consultas SQL manuales, a la vez que reduce errores, mejora la mantenibilidad y promueve un diseño más limpio al desacoplar la lógica de negocio de la capa de persistencia. [115]

Preferencias Compartidas (*SharedPreferences*)

Es un mecanismo ligero de almacenamiento llave-valor en Android, utilizado para guardar datos persistentes simples como configuraciones, preferencias del usuario o estados pequeños. La información se almacena en archivos privados XML de la aplicación y puede leerse o escribirse de manera rápida. Es ideal para guardar datos que no requieren una base de datos completa, como cadenas de texto o valores numéricos simples. [116]

ThreeTen

Es una adaptación para Android de la biblioteca *java.time* introducida en Java 8. Provee un sistema moderno y robusto para manejar fechas, tiempos, zonas horarias y duraciones con mayor precisión y seguridad que las clases tradicionales. La versión más comúnmente utilizada en Android es *ThreeTenABP (Android Backport)*, la cual permite usar la API de tiempo nativa en dispositivos con versiones antiguas de Android. [117]

Retrofit

Retrofit es una librería de cliente HTTP para Android y Java que permite consumir *APIs REST* (siglas en inglés para Transferencia de Estado Representacional) de forma sencilla. Soporta conversión transparente de JSON (siglas en inglés para Notación de Objetos de JavaScript) a objetos Kotlin/Java, manejo asíncrono con corrutinas, interceptores, autenticación y manejo de errores. Es una herramienta fundamental para aplicaciones móviles que interactúan con servicios web. [118]

OkHTTP

Es una librería de cliente HTTP que permite realizar solicitudes de red eficientes y confiables en Android y Java. Ofrece características avanzadas como conexión persistente, enrutamiento transparente, manejo automático de compresión y un potente sistema de interceptores para personalizar solicitudes y respuestas. Se integra comúnmente con *Retrofit*.

para proporcionar la capa de transporte sobre la cual se ejecutan las llamadas HTTP en Android. [119]

Transferencia de Estado Representacional (*Representational State Transfer* o REST)

Es un estilo arquitectónico para el diseño de servicios web basado en principios como el uso uniforme de recursos, operaciones estándar HTTP (GET, POST, PUT, DELETE), ausencia de estado (*statelessness*), y representaciones intercambiables de información (JSON, XML, etc.). Los servicios *RESTful* buscan ser simples, escalables y fácilmente consumibles por clientes diversos. Es uno de los enfoques más utilizados para APIs modernas debido a su flexibilidad y compatibilidad multiplataforma. [120]

Datos Observables (*LiveData*)

Corresponde a una clase observable de Android Jetpack diseñada para contener y notificar datos de manera reactiva y consciente del ciclo de vida. Solo notifica cambios a componentes activos, lo que evita fugas de memoria y errores derivados de actualizaciones fuera del ciclo adecuado. Se utiliza comúnmente junto con el modelo de vista (*ViewModel*) para mantener datos persistentes ante rotaciones y cambios de configuración. [121]

Flujo de Estado *StateFlow*

Es un flujo observable y orientado al estado, provisto por las corrutinas de Kotlin. Representa un valor de estado que siempre está disponible y que emite actualizaciones a todos los suscriptores. A diferencia de los datos observables (*LiveData*), el Flujo de Estado forma parte de la librería de corrutinas y funciona en cualquier capa de la arquitectura, no solo en la capa de UI (siglas en inglés para Interfaz de Usuario). Es ampliamente utilizado en patrones como MVVM (Modelo-Vista-Modelo de Vista) para exponer un estado reactivo de forma concurrente y segura. [122]

Alcance de Modelo de Vista (*ViewModelScope*)

Es un alcance (*scope*) de corrutinas provisto por Android Jetpack que permite ejecutar tareas asincrónicas dentro de un Modelo de Vista. Todas las corrutinas lanzadas en este alcance se cancelan automáticamente cuando el Modelo de Vista es destruido, lo que evita fugas de memoria y simplifica la gestión del ciclo de vida. Es ideal para operaciones como llamadas a API, acceso a bases de datos o procesamiento de datos que deben sobrevivir a cambios de configuración. [123]

Alcance de Corrutina (*CoroutineScope*)

Este es un componente fundamental de las corrutinas de Kotlin que define el contexto en el cual se ejecutan las mismas, lo que incluye su ciclo de vida y reglas de cancelación. Permite estructurar tareas asincrónicas de manera organizada, para que se garantice que todas las corrutinas lanzadas dentro del mismo alcance puedan cancelarse de forma conjunta. Su uso es esencial para evitar fugas de memoria y para mantener controlado el comportamiento asincrónico de una aplicación, especialmente en arquitecturas basadas en Modelos de Vista, Actividades o servicios. [124]

Inyección de Dependencias (*Dependency Injection* o DI)

Consiste en un patrón de diseño que consiste en proporcionar a un objeto las dependencias que necesita en lugar de que él mismo las construya. Este enfoque fomenta la modularidad, la reutilización de código, las pruebas unitarias y la separación de responsabilidades. En Android, la DI permite administrar servicios como repositorios, fuentes de datos, controladores de red o Modelos de Vista de forma centralizada y escalable, lo que reduce el acoplamiento y facilita el mantenimiento a largo plazo de la aplicación. [125]

Hilt

Este es un marco oficial de Android basado en *Dagger* que proporciona una solución estandarizada para la inyección de dependencias. Simplifica la configuración de DI mediante anotaciones, ciclos de vida integrados y módulos predefinidos para componentes clave del sistema. Gracias a su integración con Jetpack, *Hilt* reduce significativamente el código repetitivo respecto al uso de *Dagger* puro, facilita las pruebas y asegura una inicialización eficiente de dependencias en toda la aplicación. [126]

Dagger

Es un marco de inyección de dependencias para Java y Android que genera código estático en tiempo de compilación. Esto lo hace extremadamente eficiente en rendimiento y seguro en términos de tipos, lo que evita la reflexión y reduce el costo en tiempo de ejecución. *Dagger* organiza las dependencias mediante módulos y componentes, esto permite crear grafos complejos de objetos administrados automáticamente. Aunque su configuración puede ser detallada, ofrece un control fino sobre la construcción y el ciclo de vida de las dependencias, lo que representa una solución robusta para aplicaciones grandes y escalables. [127]

6.6.11. Bases de datos vectoriales y su contraste con bases relacionales

Las bases de datos vectoriales almacenan representaciones numéricas (*embeddings*) de información, permitiendo búsquedas semánticas rápidas y precisas. En cambio, las bases

de datos relacionales 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 relacionales 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 [47, 128]. El cuadro 1 resume el contraste entre ambos conceptos.

Cuadro 1: Resumen de Contraste: Bases de Datos Relacionales vs. Vectoriales

Característica	Base de Datos Relacional	Base de Datos Vectorial
Elemento Almacenado	Filas y Columnas (<i>Tablas</i>)	Representaciones numéricas (vectores o <i>Embeddings</i>)
Organización	Estructurada, con esquema fijo	Densamente Numérica, sin esquema fijo
Tipo de Búsqueda	Exacta, basada en valores (SQL)	Por Similitud, basada en significado
Aplicación Clave	Gestión (Usuarios, Mensajes, Documentos)	Recuperación Semántica (Contenido)

6.6.12. 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 [129, 130].

CAPÍTULO 7

Metodología

El desarrollo del proyecto *Ciudadano Digital* se llevó a cabo bajo el marco de trabajo SCRUM, un enfoque ágil que permite la entrega incremental de productos funcionales mediante ciclos cortos de desarrollo (*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 orientado a la obtención progresiva de un prototipo funcional y validado de la aplicación (Cuadro 2). Cada *sprint* tuvo una duración de entre tres y cuatro semanas, ajustándose según la complejidad técnica y la carga académica del periodo correspondiente.

Sprint	Meta Principal	Duración
<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 servidor.	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 2: Estructura de los *sprints* del proyecto, incluyendo la meta principal y duración de cada uno.

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

- **Planificación (Planeación del Sprint):** se definieron los objetivos y alcance del *sprint*, así como las tareas específicas necesarias para cumplir la meta establecida.
- **Desarrollo (Ejecución del Sprint):** se ejecutaron las tareas asignadas con enfoque en la funcionalidad incremental, priorizando siempre la obtención de resultados medibles.
- **Revisión (Revisión del Sprint):** 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 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.

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

Duración: 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.

7.1.1. 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 [5, 131, 132].

- Análisis de estudios internacionales de organismos como UNESCO (Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura) y CIEN (Centro de Investigaciones Económicas Nacionales) sobre hábitos digitales, desigualdad educativa y desarrollo de competencias ciudadanas en adolescentes y jóvenes [6, 7, 75].

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:** las metas que el usuario quisiera conseguir 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.

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

Duración: 4 semanas

Este *sprint* se centró en recopilar, procesar y estructurar el contenido educativo inicial que alimenta inicialmente al asistente virtual de inteligencia artificial. El fin de este proceso fue garantizar que el sistema pudiera generar respuestas precisas y contextualizadas sobre formación ciudadana y valores morales, basándose en información confiable y organizada de manera semántica.

7.2.1. Ejecución

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. También se recolectaron 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), la UNESCO (Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura) o universidades extranjeras. Mediante este contenido, se buscó alimentar aún más el conocimiento teórico del asistente, así como diversificar las fuentes de información a contextos internacionales.
- **Estudios complementarios:** Además de las fuentes mencionadas, también se incluyó la recopilación de libros educativos de entidades independientes (tales como IGER, Instituto Guatemalteco de Educación Radifónica) así como de autores externos. [133]
- **Registro de metadatos:** cada documento fue almacenado en un contenedor tipo **Amazon S3** (*Simple Storage Service*). Los metadatos asociados a cada documento (título, autor, año de publicación), así como la ruta de almacenamiento relativa dentro del contenedor, fueron almacenados en la base de datos vectorial, según 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 la trazabilidad desde la fuente original hasta el fragmento vectorizado. Cabe aclarar que el uso de variables en inglés corresponde a buenas prácticas de programación, con el fin de que el código pueda ser comprendido a nivel global en caso de ser necesario [134].

2. Curación y digitalización

- **Conversión de documentos:** los archivos se transformaron a texto plano (`.txt`) con codificación UTF-8 mediante herramientas como **NFKD** (siglas en inglés para Descomposición de Compatibilidad de Formas de Normalización) o **Tesseract OCR**. Esta última extrae de forma automática el texto reconocible de imágenes o documentos PDF escaneados (Reconocimiento Óptico de Caracteres).
- **Limpieza y normalización:** se eliminaron saltos de línea innecesarios, espacios vacíos múltiples y caracteres especiales, a través del Algoritmo 1.

Algoritmo 1 Proceso de limpieza y normalización profunda de texto

```
1: Procedimiento LimpiarTextotexto
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:   Retornar texto
9: Fin del procedimiento
```

- **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: Procedimiento EstandarizarFormatotexto
2:   texto ← ReemplazarMarkdownConTitulo(texto)
3:   texto ← ReemplazarNumeracionConTitulo(texto)
4:   texto ← ConvertirTitulosMayusculas(texto)
5:   texto ← UniformarNumeracion(texto)
6:   texto ← EliminarEspaciosExtremos(texto)
7:   Retornar texto
8: Fin del procedimiento
```

- **Validación de integridad:** se verificó que los textos conservaran coherencia y completitud al eliminar duplicados o secciones ilegibles. Se siguió el flujo descrito en el Algoritmo 3.

Algoritmo 3 Validación de integridad de texto

```
1: Procedimiento ValidarIntegridadtexto
2:   líneas ← DividirEnLineas(texto)
3:   líneas_limpias ← ListaVacía()
4:   Para cada línea en líneas hacer
5:     Si Longitud(Trim(linea)) < 3 entonces
6:       Continuar
7:     Fin Si
8:     caracteres_válidos ← ContarCaracteresAlfanumericosYEspacios(linea)
9:     Si caracteres_válidos / Max(Longitud(linea), 1) > 0.6 entonces
10:      Añadir(linea, líneas_limpias)
11:    Fin Si
12:  Fin Para
13:  líneas_sin_duplicados ← EliminarDuplicados(líneas_limpias)
14:  texto_limpio ← UnirLineas(líneas_sin_duplicados)
15:  Retornar texto_limpio
16: Fin del procedimiento
```

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*. Esta lista puede incrementarse 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. Se respetaron signos de puntuación, saltos de línea, listas y numeración de títulos con el fin de preservar la coherencia semántica.
- **Etiquetado y registro:** cada fragmento se asoció a una categoría temática y se describió con los siguientes metadatos: **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 representaciones numéricas:** cada fragmento fue procesado con el modelo *text-embedding-3-small* de OpenAI, lo que permitió generar vectores de 1536 dimensiones.
- **Normalización final:** la asignación del metadato *sha1* en cada vector permitió validar la ausencia de duplicados, lo que asegura la unicidad de cada vector en la base de datos vectorial.
- **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:** la representación numérica de cada fragmento se insertó junto con sus metadatos para permitir consultas semánticas eficientes. El proceso completo se muestra en el Algoritmo 4. Destaca la presencia de la variable **BATCH_SIZE**, la cual corresponde a una constante utilizada para definir el tamaño del paquete de vectores enviado a la base de datos. Esto busca evitar enviar vectores individuales que podrían ralentizar el proceso de almacenamiento y elevar los costos relacionados a la base de datos vectorial.

Algoritmo 4 Vectorización de fragmentos

```

1: Procedimiento VectorizarFragmentos(fragmentos, identificador, fuente, autor, año,  

   BATCH_SIZE)
2:   lote  $\leftarrow$  Lista Vacía()
3:   Para cada frag en fragmentos hacer
4:     sha1  $\leftarrow$  CalcularSHA1(frag)
5:     Si FragmentoYaIndexado(sha1) or EsVacio(frag) entonces
6:       Continuar
7:     Fin Si
8:     categoria  $\leftarrow$  ClasificarCategoria(frag)
9:     embedding  $\leftarrow$  GenerarEmbedding(modelo=«text-embedding-3-small», tex-  

   to=frag)
10:    metadatos  $\leftarrow$  CrearDiccionario({ «document_id»: identificador, «text»: frag,  

    «source»: fuente, «author»: autor, «year»: año, «category»: categoria, «sha1»: sha1,  

    «uploaded_at»: FechaHoraActual() })
11:    AñadirAlLote(lote, CrearVector(id=GenerarUUID(), valores=embedding, meta-  

   datos=metadatos))
12:    Si Longitud(lote)  $\geq$  BATCH_SIZE entonces
13:      RegistrarLote(lote, namespace=«ciudadanía»)
14:      lote  $\leftarrow$  Lista Vacía()
15:    Fin Si
16:  Fin Para
17:  Si lote no vacío entonces
18:    RegistrarLote(lote)
19:  Fin Si
20: Fin del procedimiento

```

- **Implementación de flujo para procesamiento de archivos** El sistema obtenido a través del proceso anterior, permitió procesar diversos formatos de archivo:
 - Documentos PDF (.pdf)
 - Documentos de Word (.doc, .docx)
 - Presentaciones (.ppt, .pptx)
 - Texto plano (.txt)
 - Markdown (.md)

- Imágenes y documentos escaneados (mediante Tesseract OCR)

La Figura 2 ilustra el flujo completo de procesamiento:

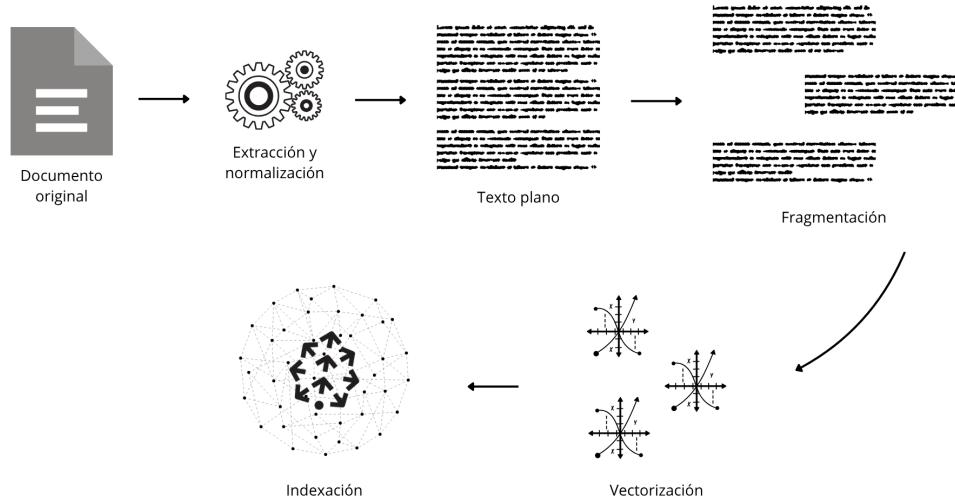


Figura 2: Flujo de procesamiento de documentos para generación del corpus.

El proceso se dividió en dos etapas principales:

- Extracción y normalización del contenido:** En esta etapa se realizó la lectura y extracción de texto desde los formatos soportados. El flujo incluyó la limpieza del texto, la estandarización del formato (identificación de títulos, encabezados y listas para preservar la semántica), la validación de la integridad (omisión de duplicados, normalización de caracteres) y la fragmentación semántica. El producto final resultante está compuesto por fragmentos de texto plano autocontenido.
- Generación de representaciones numéricas e indexación vectorial:** Cada fragmento de texto se transformó en un vector mediante el modelo de OpenAI `text-embedding-3-small`. A cada representación numérica de texto, se le asociaron metadatos relevantes (identificador del documento original, título, categoría temática, texto original, autor y año de publicación). Posteriormente, se realiza una operación de *upsert* (es decir, insertar o actualizar el registro según corresponda) en el índice vectorial de Pinecone. Simultáneamente, a través del identificador del documento original, se registró en la base de datos relacional la trazabilidad entre el vector y el archivo almacenado en un contenedor de Amazon S3.

7.3. *Sprint 3: Construcción e implementación del servidor*

Duración: 4 semanas

Este *sprint* se enfocó en el diseño, construcción e implementación de la arquitectura del

servidor del asistente virtual. Se buscó garantizar la integración de bases de datos relacionales y vectoriales, así como establecer la comunicación segura y eficiente con el modelo grande de lenguaje (LLM, por sus siglas en inglés) mediante un flujo de generación mejorada por recuperación. 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.

7.3.1. Ejecución

1. Diseño de arquitectura

- Se adoptó el patrón de diseño **Modelo–Vista–Controlador (MVC)**, mediante la estructura de la Figura 3. Esta arquitectura permitió separar las responsabilidades del sistema, lo que buscó facilitar el mantenimiento y escalabilidad.

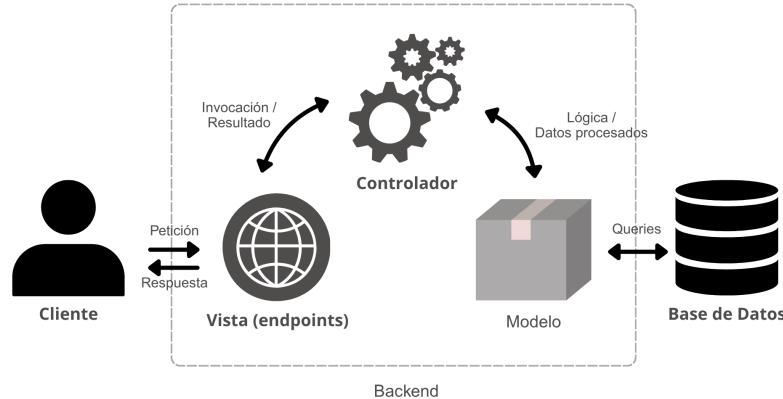


Figura 3: Arquitectura MVC (Modelo, Vista, Controlador).

■ Modelos:

- Representan entidades del sistema: usuarios, chats, mensajes, sesiones, documentos, categorías.
- Cada modelo incluyó operaciones de creación, lectura, actualización y eliminación de registros (CRUD, por sus siglas en inglés) que se ejecutan directamente sobre la base de datos 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.

- **Vistas:** En la implementación realizada del modelo MVC (Modelo-Vista-Controlador), las vistas corresponden a los puntos de conexión expuestos por el API, también llamados rutas o *endpoints*. A través de la consulta a estos puntos de conexión, se consiguió acceso a las funciones definidas por el servidor; como registro de usuarios, inicio de sesión, listado de chats, envío de preguntas, entre otros.

- **Controladores:** Su diseño buscó gestionar 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: Procedimiento Controladorrequest
2:   Validar(request.datos)
3:   resultado ← modelo.operacion(request)
4:   devolver(resultado)
5: Fin del procedimiento

```

- **Módulos auxiliares:**

- **Middlewares (Software intermedio):** fueron diseñadas como funciones ejecutadas antes de realizar la acción principal de cada punto de conexión. Se encargan de validar la autenticación y seguridad antes de pasar al controlador. El Algoritmo 6 muestra un ejemplo de middleware para validar tokens JWT (JSON Web Token).

Algoritmo 6 Validación de token

```

1: Procedimiento ValidarTokenRequestrequest
2:   Si not ValidarToken(request.token) entonces
3:     DevolverError(401, «Token inválido»)
4:   Sino
5:     Continuar(request)
6:   Fin Si
7: Fin del procedimiento

```

- **Helpers (ayudantes):** funciones auxiliares reutilizadas a lo largo del código, como:
 - encryptarContraseña(contraseña)
 - generarToken(usuarioID)
 - formatearFecha(fecha)

2. Diseño y construcción de bases de datos

Base de datos relacional

Levantada en PostgreSQL, fue la encargada de almacenar la información estructurada del sistema; como usuarios, chats, mensajes y sesiones, de manera que se mantuvo la persistencia de datos y se facilitó la gestión de las interacciones del usuario con el asistente virtual. La Figura 4 muestra el diagrama entidad-relación (ER) de la base de datos relacional diseñada.

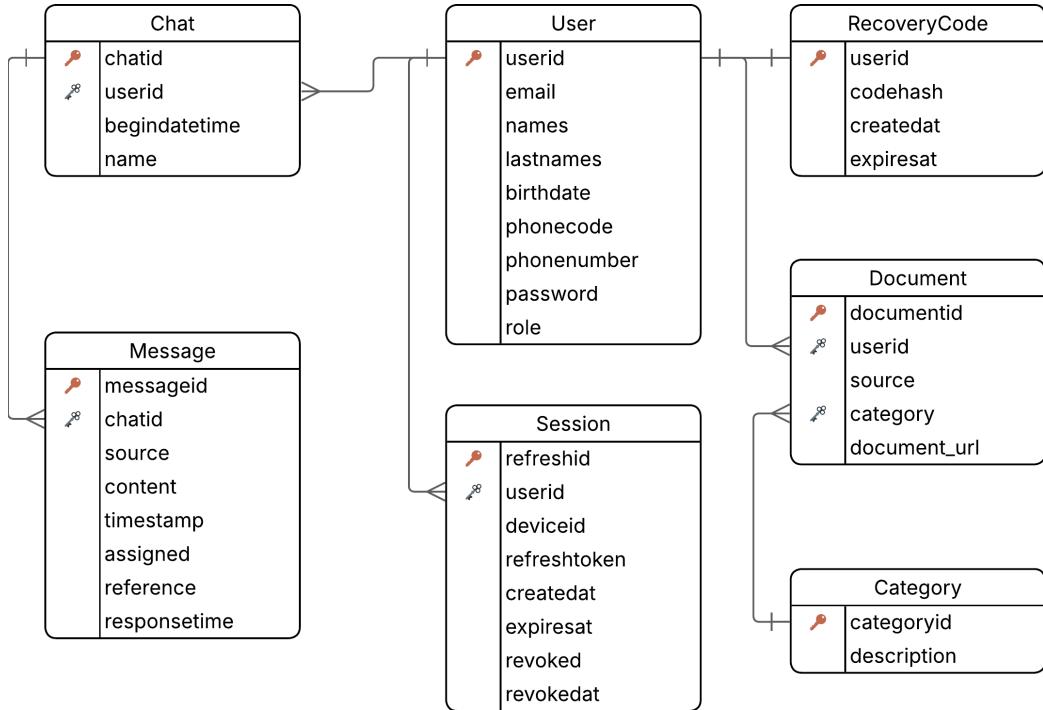


Figura 4: Diagrama entidad-relación (ER) de la base de datos relacional.

- 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 de generación mejorada por recuperación), se implementó una base de datos vectorial utilizando Pinecone. Esta base almacena las representaciones numéricas generadas previamente a partir de los fragmentos de texto, lo que permitió generar consultas rápidas y precisas basadas en similitud semántica.

- Motor: Pinecone.

- Contenido: representaciones numéricas de fragmentos de documentos con metadatos (fuente, categoría, documento, bloque, relevancia).
- Obtención de los fragmentos más relevantes en relación al vector generado a partir de la pregunta realizada. Esto lo realiza Pinecone de forma automática, según el método de similitud configurado al momento de crear el índice. En este caso, se indicó que se utilice la métrica de **similitud coseno**, 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:

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

En esta etapa del desarrollo se definió un umbral de **0.4**, lo que indica que, para que una representación numérica devuelta por Pinecone sea tomada como válida, esta debe tener una coincidencia de, por lo menos, el 40 % con la representación numérica de la pregunta. Esta barrera puede variar a medida que se enriquece el corpus, ya que a mayor variedad hay más oportunidades de similitud entre vectores.

3. Implementación del modelo LLM y flujo RAG

Se implementó un flujo de generación mejorada por recuperación que permitió al asistente virtual generar respuestas fundamentadas en los contenidos educativos previamente indexados. Este flujo integró los siguientes componentes:

- API en NodeJS:** Actúa como servidor principal al orquestar la comunicación entre la aplicación móvil y los servicios de procesamiento de lenguaje natural.
- Microservicio en Python:** Gestiona la interacción con el modelo LLM de OpenAI para generar respuestas y procesar documentos.
- API de OpenAI:** Se utilizó para la generación de representaciones numéricas (*embeddings*) y respuestas contextualizadas.
- Base de datos vectorial (Pinecone):** Almacena y permite la recuperación eficiente de las representaciones numéricas generadas a partir de los contenidos educativos.
- Base de datos relacional (PostgreSQL):** Gestiona usuarios, mensajes, sesiones e información de los documentos procesados, asegurando la trazabilidad desde los vectores hasta el documento original.

El flujo completo del proceso RAG, desde la recepción de una pregunta en la aplicación móvil hasta la devolución de la respuesta, se ilustra en la Figura 5.

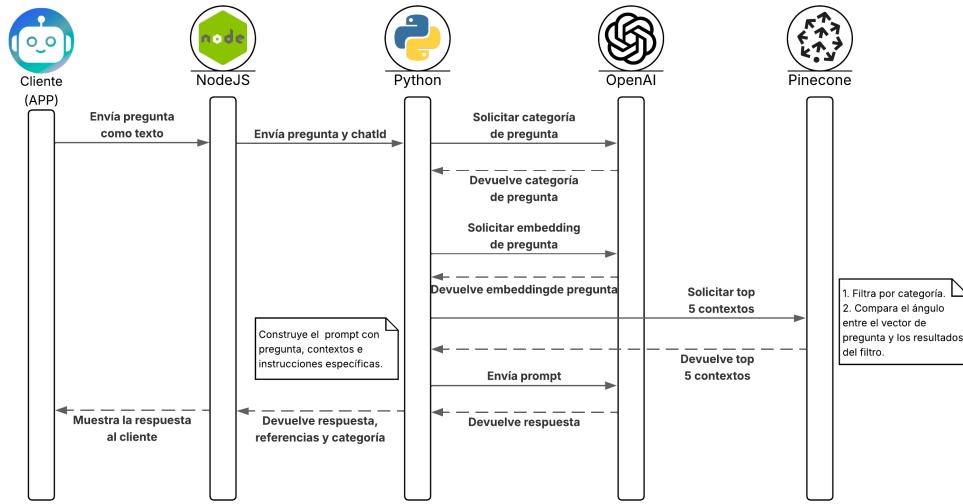


Figura 5: Flujo de procesamiento de preguntas mediante RAG.

Arquitectura de almacenamiento distribuido

El almacenamiento de información a lo largo del sistema se distribuyó en tres elementos principales que trabajan de forma coordinada:

- **Base de datos relacional (PostgreSQL):** se definió como el gestor principal de los datos estructurados del sistema; almacena información de usuarios, chats, mensajes, sesiones, códigos de recuperación de contraseñas y metadatos básicos de los documentos procesados.
- **Base de datos vectorial (Pinecone):** se designó como encargada de almacenar las representaciones numéricas generadas a partir de los documentos seleccionados. Constituye la base del funcionamiento RAG al permitir recuperar el contexto necesario según la pregunta realizada, de manera que el modelo LLM pueda generar respuestas fundamentadas.
- **Almacenamiento de documentos originales (AWS S3):** implementado para almacenar todos los documentos originales cargados al sistema. Esto permite realizar la consulta de estos documentos por parte usuarios con rol de **Administrador**.

Esta arquitectura distribuida buscó garantizar la trazabilidad completa desde cada vector indexado hasta su documento fuente, mientras mantuvo la eficiencia en las búsquedas semánticas y la integridad de los datos originales.

Flujo detallado de procesamiento de consultas

El microservicio de Python gestionó el flujo RAG completo al coordinar la interacción entre la base vectorial y el modelo grande de lenguaje, esto con el fin de generar respuestas contextualizadas y fundamentadas. El proceso se dividió en tres etapas principales:

- 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 una representación numérica (*embedding*) del texto de la pregunta utilizando el modelo *text-embedding-3-small* de OpenAI, transformando la información textual en un vector semántico de 1536 dimensiones.
 - 3) Se realiza una consulta al índice de Pinecone con la representación numérica generada, recuperando los cinco fragmentos más relevantes del corpus vectorizado mediante similitud coseno, 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 los metadatos de los vectores y se combina con la pregunta original, construyendo una petición que resume la información relevante y plantea la consulta al modelo grande de lenguaje.
 - 6) Envía la petición al modelo LLM de OpenAI para generar la respuesta contextualizada.
 - 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 completo de interacciones.

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

Algoritmo 7 Flujo completo de procesamiento de preguntas

```

1: Procedimiento ProcesarPreguntaUsuario
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: Fin del procedimiento

```

7.4. Sprint 4: Desarrollo de la interfaz móvil en Kotlin

Duración: 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. Esto permitió una interacción eficiente con el asistente al integrar las funcionalidades proporcionadas por el servidor a través de servicios puntos de conexión (*endpoints*). El diseño de la arquitectura se basó en el patrón de diseño MVVM (Modelo-Vista-Modelo de Vista), 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.

7.4.1. Ejecución

1. Diseño de arquitectura

- Se adoptó el patrón de diseño **Modelo-Vista-Modelo de Vista (MVVM, por sus siglas en inglés)** al seguir la estructura de la Figura 6. Esta arquitectura permitió separar las responsabilidades del sistema, con el fin de facilitar el mantenimiento y escalabilidad. Mediante este patrón de diseño se buscó lograr una comunicación reactiva entre la interfaz de usuario y las fuentes de datos, a través del uso de componentes de arquitectura de Android como Datos Observables (*LiveData*), Vista de modelo (*ViewModel*) y Repositorios (*Repositories*).

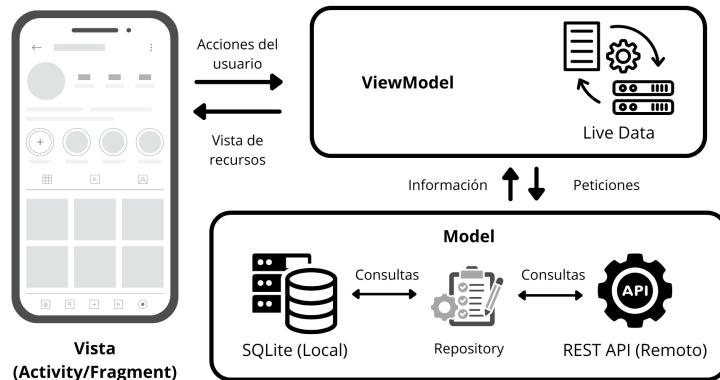


Figura 6: Arquitectura MVVM (Modelo, Vista, Modelo de Vista).

■ Vista:

- Se implementó a través de Actividades y Fragmentos organizados modularmente.
- Cada fragmento representó una sección funcional del sistema (por ejemplo: inicio de sesión, chat, documentos, perfil).

- Se aplicó el patrón de navegación basado en las librerías de Android *NavController* y *SafeArgs*, lo que busca garantizar transiciones seguras y controladas entre vistas.
- Se usó componentes de diseño basados en *Material Design 3*, el estandar de diseño de Google, para mantener consistencia visual, accesibilidad y adaptabilidad en distintos tamaños de pantalla [109].
- Cada Fragmento o Actividad contó con su propia plantilla, definida en formato XML (Lenguaje de Marcado Extensible, por sus siglas en inglés), que especifica la disposición de los elementos visuales.

■ **Modelo de Vista:**

- Fue definido como intermediario entre la vista y las fuentes de datos, encargado de manejar la lógica de presentación.
- Se diseñó para emplear Datos Observables (*LiveData*) y Flujos de Estado (*StateFlow*) para notificar automáticamente a la vista sobre cambios en los datos.
- Encapsuló la interacción con los repositorios para garantizar que la vista permanezca libre de lógica de negocio.
- El Algoritmo 8 ilustra el flujo básico de un Modelo de Vista típico.

Algoritmo 8 Flujo básico de un Modelo de Vista

```

1: Procedimiento ObtenerMensajeschatId
2:   emitar(Estado.Cargando)
3:   datos ← repositorio.obtenerMensajes(chatId)
4:   emitar(Estado.Exitoso(datos))
5: Fin del procedimiento

```

■ **Modelo (Repositorio y Fuentes de Datos):**

- Los repositorios centralizaron el acceso a las fuentes de datos, tanto locales como remotas.
- Se definieron dos capas de origen de datos:
 - a) **Datos Locales:** implementados con la base de datos de Room, definido para gestionar entidades como usuarios, mensajes, chats y documentos. Esta capa permitió 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 requieren comunicación con internet.
 - b) **Datos Remotos:** implementados mediante *Retrofit* y *OkHttp*, se utilizó para consumir los puntos de conexión (*endpoints*) del servidor desarrollado en el *sprint* anterior.
- Los repositorios se encargan de determinar 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, lo que simplificó la creación de instancias de Modelos de Vistas, repositorios y servicios. Este

enfoque buscó garantizar el bajo acoplamiento entre componentes y favorecer 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 el Fragmento de Navegación (*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 usó el alcance del modelo de vista (*ViewModelScope*) y el alcance de coroutines (*CoroutineScope*) para ejecutar tareas asíncronas sin bloquear la interfaz.
- Se implementó almacenamiento persistente mediante Preferencias Compartidas (*SharedPreferences*) y Room, buscando 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 permitió 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. Integración con el servidor y servicios de Python

- Se verificó de forma directa el acceso del cliente al consumo de puntos de conexión del servidor para autenticación, gestión de sesiones, envío de preguntas y recuperación de respuestas.
- Se validó en la aplicación el procesamiento de respuestas JSON, conversión y renderizado de manera clara y comprensible.
- Se definió el manejo de errores y reconexión ante fallos de red, lo que buscó asegurar robustez en la experiencia de usuario.

7.5. *Sprint 5: Pruebas y validación*

Duración: 3 semanas

Este *sprint* se centró en validar el funcionamiento integral del sistema para asegurar la correcta interacción entre la aplicación móvil, el servidor, 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 mediante la metodología y métricas propuestas en las secciones 6.3.9 y 6.3.10 de este documento.

7.5.1. Ejecución

1. Pruebas funcionales del sistema

Las pruebas funcionales se llevaron a cabo al seguir un proceso estructurado compuesto por cinco fases: configuración del entorno, verificación de comunicación entre módulos, pruebas de puntos de conexión, verificación de operaciones CRUD y validación de integridad entre fuentes de datos.

- **1. Configuración del entorno de prueba.** Se desplegó el servidor en un entorno local sobre NodeJS mediante el uso de las mismas variables de entorno del entorno de producción. La base de datos relacional (PostgreSQL) se inicializó con la estructura definitiva del proyecto y una carga mínima de datos para pruebas. En el caso de la base de datos vectorial, Pinecone se configuró con el índice previamente alimentado; mientras que al contenedor S3 ya habían sido cargados los documentos del corpus inicial.

Para que la aplicación móvil tuviera acceso al servidor local, el equipo en el que se levantó el servidor fue habilitado como punto de conexión inalámbrico. De esta manera, el dispositivo Android de pruebas pudo conectarse a la misma red y utilizar la dirección IP de la computadora como la IP del servidor.

- **2. Verificación de comunicación entre módulos.** Para asegurar que cada componente podía comunicarse con el siguiente, se realizaron solicitudes controladas del tipo «ping».

- **Desde la aplicación móvil al servidor:** envío de solicitudes HTTP simulando inicio de sesión y envío de mensajes.
- **Del servidor a la base de datos relacional:** ejecución de consultas directas de lectura y escritura.
- **Del servidor a la base de datos vectorial:** inserción y recuperación de vectores de prueba.
- **Del servidor al contenedor S3:** subida, lectura y eliminación de documentos.

Se consideró exitosa la comunicación cuando cada módulo pudo responder con el código de estado esperado y en el formato establecido.

- **3. Pruebas de puntos de conexión mediante Postman.** Se creó una colección de Postman con los puntos de conexión del sistema, agrupados por funcionalidad: /autenticación, /mensajes, /historial, /documentos. Para cada punto de conexión se diseñaron casos de prueba que contemplaron:

- Parámetros válidos e inválidos.
- Códigos de estado esperados.
- Verificación de restricciones de acceso (por rol o por token).
- Validación de formatos de respuesta (JSON).
- Manejo de errores (HTTP 400 y HTTP 500).

La prueba se consideró exitosa si la respuesta coincidía con el comportamiento esperado para cada punto de conexión definido en el servidor.

- **4. Pruebas CRUD en las bases de datos.** Se verificaron las operaciones de creación, lectura, actualización y eliminación tanto en PostgreSQL como en Pinecone. Por cada operación CRUD realizada desde el cliente, se confirmó:

- Que la operación se reflejara correctamente en la base de datos relacional.
- Que el vector correspondiente se actualizara o eliminara en Pinecone.
- Que en caso de involucrar documentos, los cambios se reflejaran también en el contenedor S3.

Se revisó manualmente cada operación mediante consultas SQL directas y mediante la consola de Pinecone.

- **5. Validación de integridad entre fuentes de datos.** Para comprobar la consistencia entre las tres fuentes de datos (PostgreSQL, Pinecone y contenedor S3), se ejecutaron casos en los que:

- **Se creaba un documento desde la aplicación:** En el resultado esperado, este documento debía aparecer en la base relacional, almacenarse en S3 y generarse su vector en Pinecone.
- **Se eliminaba un documento** En el resultado esperado, el documento debía eliminarse de los tres sistemas.

Se consideró exitosa la prueba cuando los cambios se replicaban de forma consistente en todas las fuentes, sin incongruencias.

2. Pruebas de calidad y confiabilidad de las respuestas

- **1. Construcción del conjunto de pruebas.** Se elaboró una lista de 55 preguntas:
 - 45 relacionadas con el corpus, extraídas con herramientas comerciales de Inteligencia Artificial, como **NotebookLLM**.
 - 10 preguntas de control, no asociadas al corpus, para evaluar la **no alucinación**.

Cada pregunta fue almacenada en una hoja de cálculo, junto con las referencias que se esperaba que el modelo consultara para responder cada pregunta.

- **2. Ejecución manual de cada consulta.** Se envió cada pregunta al sistema desde la aplicación móvil, asegurando que:
 - el historial de chat estuviera limpio antes de cada prueba,
 - la pregunta se procesara en un nuevo mensaje,
 - se registraran:
 - el tiempo de respuesta,
 - la respuesta generada,
 - las referencias devueltas por el sistema.

Para calcular el tiempo de respuesta, el sistema inicia un contador justo antes de realizar la llamada al servicio de Python, y lo detiene hasta que se devuelve la respuesta final al cliente.

- **3. Validación de alineación con el corpus.** Cada respuesta obtenida por el sistema, fue clasificada como:
 - **Exitosa:** si el sistema fue capaz de responderla.
 - **Congruente:** si el comportamiento esperado se cumplió. Si la pregunta era de control, el comportamiento esperado fue que el sistema no pudiera responder.

Si el modelo respondía cuando no debía hacerlo, o viceversa, se consideraba como **incongruente**

- **4. Verificación de referencias.** Para cada respuesta obtenida del sistema, se compararon las referencias consultadas contra las referencias esperadas, catalogando cada coincidencia como:
 - **Exacta:** si las referencias esperadas y las consultadas por el modelo coinciden por completo.
 - **Parcial:** si las referencias consultadas incluyen algunas de las referencias esperadas, pero también el modelo hizo uso de referencias adicionales.
 - **Nula:** si las referencias utilizadas difieren completamente de las referencias esperadas.
- **5. Medición de latencia.** Para cada una de las 55 consultas se registró el tiempo exacto entre solicitud y respuesta. Posteriormente, se calculó:
 - tiempo mínimo,
 - tiempo máximo,
 - tiempo promedio,
 - desviación estándar entre tiempos.

Este proceso permite que cualquier evaluador pueda reproducir la medición bajo las mismas condiciones controladas.

7.6. *Sprint 6: Documentación y presentación*

Duración: 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.

7.6.1. Ejecución

1. Se elaboró el informe final por medio de

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

2. Se preparó la presentación final, que incluyó

- El desarrollo del material visual que resumió el proyecto, así como diagramas de arquitectura, capturas de pantalla del prototipo móvil, flujo de interacción y ejemplos de uso del asistente.
- La elaboración de una presentación estructurada para explicar el proceso de desarrollo, resultados obtenidos y funcionalidades del sistema.
- El ensayo de la presentación y ajuste de contenido para garantizar claridad, concisión y relevancia para el público evaluador.

3. Se trasladó y entregó toda la documentación al cliente, a través de

- La consolidación de repositorios de código, documentos de investigación, fichas de usuario, diagramas de arquitectura y demás materiales generados.
- La entrega formal de toda la información consolidada a la Fundación de Scouts de Guatemala, con el objetivo de asegurar el acceso a todos los recursos para pruebas, mantenimiento y futuras actualizaciones.
- El registro de la entrega, que incluye el inventario de archivos, la versión final de toda la documentación generada y la evidencia de disponibilidad del producto para pruebas finales.

CAPÍTULO 8

Resultados

Esta fase del proyecto *Ciudadano Digital* culminó con la implementación de un prototipo funcional de un asistente virtual para la educación informal en ciudadanía y valores morales. El sistema integra de manera coherente componentes de procesamiento de lenguaje natural (NLP, por sus siglas en inglés), recuperación semántica de contexto y una interfaz nativa móvil para Android. Si bien los resultados no constituyen una validación empírica del impacto educativo, al no contar con pruebas de campo reales con usuarios finales, esta primera versión demostró la viabilidad técnica y conceptual del proyecto. Asimismo, se verificó su alineación con el marco teórico y los objetivos específicos, destacando los logros técnicos y de diseño alcanzados durante los sprints definidos.

8.1. Definición de usuario objetivo (Persona)

Durante el primer *sprint* se elaboró una ficha técnica detallada del **perfil Persona**, ilustrada en la Figura 7.

ANA LÓPEZ	
	
EDAD	14 años
SEXO	Mujer
PAÍS	Guatemala
EDUCACIÓN	Secundaria
UBICACIÓN	Rural
ETNIA	Maya
TECNOLOGÍA	Ocasional
Frase representativa <i>"Quiero entender cómo puedo ayudar a mi comunidad y a mí país, y que haya guías claras y rápidas para aprender, sin tener que leer documentos enormes y aburridos."</i>	
Biografía <p>Actualmente cursa el nivel de educación básico en una institución pública y se identifica como parte del pueblo maya al pertenecer a una familia k'iche'. Accede a internet principalmente por medio de recargas en su teléfono celular, con un gasto mensual promedio de entre Q1 y Q50. Su contexto educativo y tecnológico presenta limitaciones de acceso a recursos formales.</p>	
Motivaciones <p>Ana desea comprender mejor sus derechos y responsabilidades como ciudadana, así como contar con recursos accesibles que le permitan formarse en valores éticos y ciudadanía digital. Le motiva poder utilizar su teléfono móvil para aprender de manera sencilla y rápida, ya que reconoce la importancia de involucrarse en su comunidad y aportar al bienestar colectivo.</p>	
Frustraciones <ul style="list-style-type: none"> La educación cívica en su escuela es poco atractiva y limita su interés. Tiene dificultades para acceder a materiales digitales por la conectividad limitada y el costo de recargas. percibe que la información en internet no siempre es confiable ni adaptada a su contexto cultural. Considera que las oportunidades educativas en su comunidad son de menor calidad que en zonas urbanas. 	
Necesidades Principales <ul style="list-style-type: none"> Contenidos breves y atractivos que faciliten la comprensión de temas de ciudadanía y valores éticos. Información confiable y relevante que pueda relacionar con su vida diaria. Herramientas digitales ligeras que funcionen en condiciones de baja conectividad y con costos reducidos de datos. 	
Objetivos <ul style="list-style-type: none"> Acceder a información clara y confiable sobre derechos humanos, deberes ciudadanos y valores éticos. Mejorar sus competencias cívicas para participar en actividades comunitarias. Utilizar herramientas digitales que se adapten a sus limitaciones de conectividad y recursos económicos. 	
Comportamiento digital <p>Ana utiliza el celular de manera ocasional para acceder a redes sociales y buscar información para sus tareas escolares. Prefiere contenidos visuales y cortos, como videos explicativos, y suele descargar materiales cuando tiene conexión disponible para revisarlos después sin gastar datos.</p>	

Referencias: Compendio Estadístico con enfoque de JUVENTUD (INE, 2023), Educación y desarrollo de competencias digitales en América Latina y el Caribe (CEPAL, 2025), Desafíos y oportunidades para Guatemala: hacia una agenda de futuro (PNUD, 2022), Desigualdad en Centroamérica, México y el Caribe: Análisis de brechas y recomendaciones (CEPAL, 2023), Desarrollo de habilidades digitales en América Latina y el Caribe: ¿Cómo aumentar el uso significativo de la

Figura 7: Ficha de **Perfil Persona** con datos básicos del usuario objetivo.

Al identificar las características más comunes entre la población guatemalteca en edad escolar, se definieron los parámetros necesarios para conformar un ejemplo del usuario objetivo del proyecto. Estas características incluyen:

- Edad promedio: 13-17 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.

A partir de este perfil se definieron los siguientes aspectos centrales de la aplicación:

- **Plataforma:** En esta fase, el desarrollo se concentró en el sistema operativo Android. Se obtuvo compatibilidad para dispositivos de este sistema operativo a partir de la versión **Android 7**.

- **Enfoque:** La interacción pregunta-respuesta se diseñó bajo la filosofía del método socrático, lo que permitió que en cada interacción con el asistente se obtenga también una lista de preguntas sugeridas.
- **Diseño:** La interfaz sigue los principios de *Material Design*, esto al priorizar un diseño minimalista y enfocado en la funcionalidad del sistema

8.2. Corpus vectorizado

El sistema definido en los *sprints* 2 y 3 permitió el procesamiento y gestión de los documentos con los que se alimenta el modelo. Con ello, se tomaron los documentos seleccionados al inicio del *sprint* 2, se sometieron a procesamiento, limpieza e indexación, todo a través de solicitudes realizadas desde el cliente, validando la correcta integración de sistemas internos y cliente-servidor.

Las pruebas realizadas verificaron la sincronización entre todas las fuentes de datos a través del servidor, asegurando que las consultas se asociaran con el contenido adecuado. La Figura 8 muestra el índice vectorial en Pinecone después de cargar las representaciones numéricas (*embeddings*) de los fragmentos de texto.

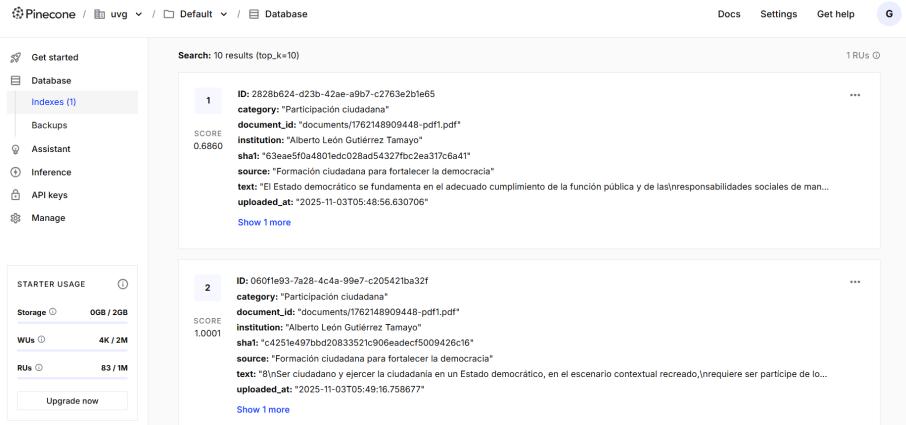


Figura 8: Vista del índice vectorial en Pinecone con las representaciones numéricas (*embeddings*) cargadas.

A través de este proceso, se obtuvo:

- Una base documental curada y segmentada en categorías temáticas.
- Representaciones numéricas (*embeddings*) generadas 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.

8.3. Interfaz móvil en Kotlin (Cliente)

Durante el cuarto *sprint* se desarrolló la aplicación móvil nativa para Android utilizando Kotlin y el patrón de arquitectura MVVM (Modelo-Vista-Modelo de Vista). Esta estructura permitió mantener una organización clara del código, lo que facilita la escalabilidad y el mantenimiento. Como resultado, se obtuvo una interfaz intuitiva y accesible para cualquier usuario, incluso sin experiencia previa en herramientas similares.

8.3.1. Flujo de interacción

El flujo de interacción del usuario se compone de tres actividades principales y un conjunto de fragmentos especializados que gestionan las distintas funciones de la aplicación.

▪ Actividades

- **SplashActivity:** Es la primera vista al abrir la aplicación. Aunque únicamente muestra el logotipo del proyecto, se encarga de verificar si existe una sesión activa mediante el token JWT almacenado en las Preferencias Compartidas (*SharedPreferences*). Si el token es válido, redirige directamente a la vista principal, evitando que el usuario deba iniciar sesión cada vez que abre la aplicación.
- **UnloggedActivity:** Agrupa los fragmentos relacionados con la autenticación (*LoginFragment*, *RegisterFragment* y *RecoverPasswordFragment*). Cada fragmento aplica sus propias validaciones y coordina la comunicación con el servidor para garantizar un ingreso seguro al sistema.
- **MainActivity:** Es el núcleo de la aplicación una vez el usuario ha sido autenticado. Aloja el *NavHostFragment*, componente encargado de gestionar la navegación entre los fragmentos funcionales; como chat, perfil de usuario y visualización de documentos. Además, incorpora un menú lateral que permite acceder rápidamente al historial de conversaciones.

▪ Fragmentos

- **LoginFragment:** Permite al usuario ingresar sus credenciales y valida su formato y autenticidad.
- **RegisterFragment:** Permite crear una cuenta nueva con validaciones de integridad y unicidad. El correo electrónico es indispensable para recuperación de contraseña y notificaciones de ingreso o eliminación de documentos.
- **SendRecoveryFragment:** Sigue el envío del código de recuperación al correo registrado, el cual solo se envía si se encuentra una cuenta asociada al mismo.
- **VerifyCodeFragment:** Valida el código de recuperación ingresado.
- **ResetPasswordFragment:** Permite establecer una nueva contraseña tras validar el código enviado.
- **ChatFragment:** Facilita la interacción con el asistente virtual, ya que muestra el historial de mensajes y permite enviar nuevas consultas.

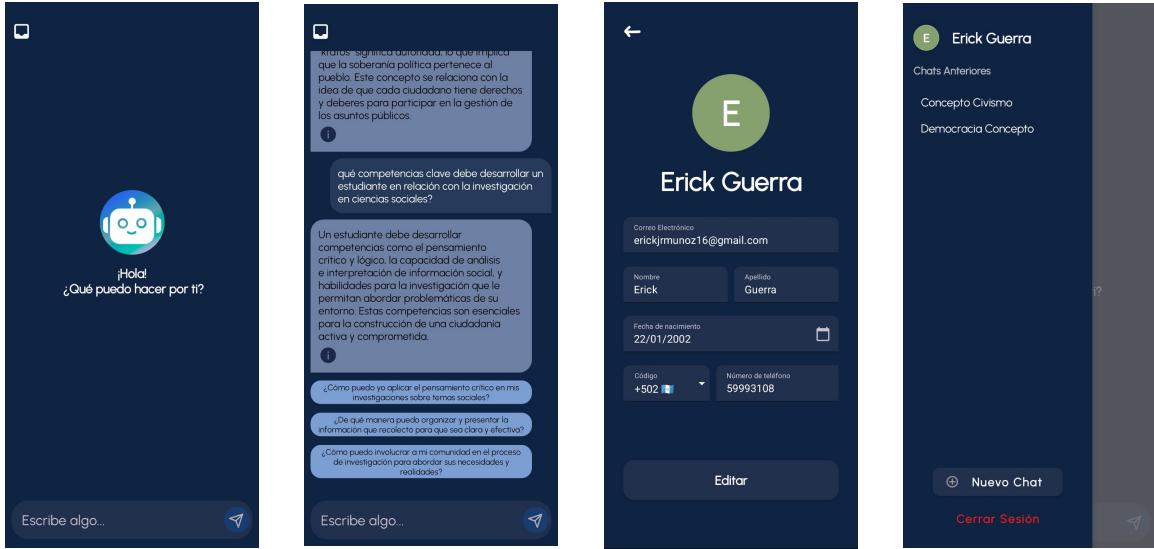
- ***ProfileFragment***: Permite visualizar y editar la información personal del usuario, exceptuando la contraseña, que solo puede modificarse mediante el proceso de recuperación.
- ***DocumentsFragment***: Muestra la lista de documentos cargados y permite su gestión.
- La navegación entre fragmentos se implementó mediante el componente de navegación de Android (*Navigation Component*), lo cual garantiza transiciones coherentes y control sobre la pila de navegación.
- Se integraron indicadores de carga y estado de conexión para brindar retroalimentación inmediata durante las consultas al LLM.

8.3.2. Funcionalidades Comunes

La aplicación distingue entre dos roles de usuario: **Usuario** y **Administrador**, definidos en el campo «**role**» de la base de datos. Sin embargo, ambos comparten un conjunto de funcionalidades esenciales:

- **Conversaciones**: Permite crear chats, enviar mensajes y recibir respuestas del modelo.
- **Perfil de Usuario**: Los usuarios pueden visualizar y actualizar su información personal (nombre, apellidos, fecha de nacimiento, correo y teléfono).
- **Cerrar Sesión**: Se puede cerrar la sesión manualmente o esperar a que expire automáticamente tras una hora de inactividad.

La Figura 9 muestra las pantallas principales disponibles para todos los usuarios.



(a) Pantalla de chat con asistente virtual.

(b) Pantalla de chat con mensajes.

(c) Vista de perfil del usuario.

(d) Panel lateral general.

Figura 9: Interfaz de la aplicación móvil, opciones generales.

8.3.3. Funcionalidades del Administrador

Por otro lado, existen funciones exclusivas para el rol **Administrador**, orientadas al control del corpus documental usado por el modelo. Estas funciones incluyen:

- **Ver documentos:** Acceso al listado de archivos cargados y opción de abrir el documento original.
- **Añadir documentos:** Permite registrar nuevos archivos completando un formulario con información básica (Nombre, Autor, Año, Rango de edades). La carga y procesamiento del documento se realiza en segundo plano y se notifica por correo al finalizar.
- **Eliminar documentos:** Permite remover documentos del corpus. La eliminación también se procesa en segundo plano y el administrador recibe una notificación al completarse.

La Figura 10 muestra las interfaces destinadas exclusivamente a administradores.



(a) Panel lateral con acceso a documentos.

(b) Pantalla de gestión de documentos.

(c) Formulario para ingreso de documentos.

Figura 10: Interfaz de la aplicación móvil, opciones de administrador.

8.4. Pruebas y Validación

La implementación del prototipo concluyó con una evaluación exhaustiva del sistema integrado. Se realizaron pruebas a los puntos de conexión mediante Postman, los cuales cubren la autenticación, recuperación de contraseña, envío de mensajes, historial de chats y administración de documentos.

En el Anexo A1, se presenta una serie de 45 preguntas relacionadas al tema central del proyecto, abarcando conceptos de ciudadanía, formación ciudadana y valores morales. Adicionalmente, se realizaron 10 preguntas de control (no relacionadas con el tema de civismo ni con los documentos que conforman el corpus), diseñadas para evaluar la **no alucinación** del modelo; es decir, que no responda cuando no deba hacerlo.

Para cada pregunta, independientemente de si era de control o no, se tomó el tiempo de respuesta, así como las referencias devueltas por el sistema (Anexo A2). La columna **Exitosa** se refiere a si la pregunta fue o no fue respondida por el modelo, mientras que la columna **Congruente** establece la congruencia según el tipo de consulta realizada:

- **Consulta:** Estas se refieren a preguntas que el modelo **sí debería responder** ya que están basadas estrictamente en el contenido del corpus proporcionado.
- **Control:** Se trata de preguntas deliberadamente fuera del contexto del sistema, las cuales el modelo **no debería responder**. En este caso, la congruencia es positiva cuando el modelo efectivamente se niega a responder por estar fuera de su alcance y propósito.

Como se mencionó, el Anexo A2 muestra la relación entre los documentos que se espera

que el modelo seleccione para responder cada pregunta, frente a las referencias que finalmente utilizó para extraer el contexto necesario. Se obtienen los siguientes resultados para las 3 clasificaciones descritas previamente en la sección 7.5.1 de este documento:

- **Coincidencia exacta:** Se evidencia en las filas 13, 16, 17, 22, 23, 34, 39, 42, 46, 48 y 53. Destaca el caso mostrado en la fila 13, puesto que si bien las referencias utilizadas son las mismas que las esperadas, difiere el orden de prioridad interpretado por el modelo.
- **Coincidencia parcial:** Se evidencia este comportamiento en las filas 1, 7, 11, 14, 18, 26, 28, 33, 36, 38, 41, 43, 44, 47, 49, 51, 52 y 55.
- **Coincidencia nula:** Se evidencia en las filas 3, 6, 8, 12, 19 y 31.

8.4.1. Métricas Encontradas

A partir de los resultados, se calcularon las siguientes métricas relacionadas con el desempeño del sistema:

En primer lugar, el Anexo A1 muestra que, de 45 consultas reales (excluyendo las de control), 35 fueron respondidas exitosamente, lo que representa una tasa de éxito global del 77.78 %. Para las consultas de control, ninguna fue respondida, lo que resulta en una tasa de éxito del 100 % para este tipo.

Respecto a la congruencia, de las 55 consultas totales, 45 mostraron un comportamiento congruente con lo establecido, lo que equivale a una congruencia fáctica global del 81.81 %.

Para la latencia, el tiempo de respuesta mínimo fue de 3.303 segundos y el máximo de 19.121 segundos, con una desviación estándar aproximada de 2.96. El tiempo de respuesta promedio fue de 7.8184 segundos. La comparación de estos datos con modelos comerciales será abordada en la sección de discusión.

Analizando el Anexo A2, de las 35 consultas respondidas correctamente, la distribución por tipo de coincidencia fue:

- Coincidencia Exacta: 11 (31.43 %)
- Coincidencia Parcial: 18 (51.43 %)
- Coincidencia Nula: 6 (17.14 %)

Las métricas calculadas se resumen en el Cuadro 3.

Cuadro 3: Métricas de desempeño del sistema

Métrica	Valor
Tasa de éxito global	77.78 %
Congruencia fáctica	81.81 %
Latencia promedio	7.8184

CAPÍTULO 9

Discusión

El desarrollo del proyecto *Ciudadano Digital* permitió obtener un prototipo funcional de un 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 generación mejorada por recuperación (RAG, por sus siglas en inglés) con modelos grandes de lenguaje (LLM, por sus siglas en inglés) 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.

Como se mencionó al inicio de este documento, el desarrollo de *Ciudadano Digital* contempla la base técnica que buscó demostrar la viabilidad tecnológica de las herramientas para conformar un sistema de generación mejorada por recuperación efectivo. Para este punto, como se mencionó en la justificación, se optó por centrar el funcionamiento de la aplicación para su disponibilidad en dispositivos bajo el sistema operativo Android. Esta decisión responde a criterios de accesibilidad y equidad tecnológica, puesto que los dispositivos bajo este sistema operativo representan la mayoría del mercado móvil en regiones marginadas, por lo que se alinea con el público objetivo identificado. El diseño implementado para la aplicación enfatiza la legibilidad, los contrastes adecuados, los controles táctiles amplios y los flujos de navegación sencillos, todo ello a través de los principios dictados por el sistema de diseño de Google, *Material Design*.

Por otro lado, la interacción pregunta-respuesta diseñada bajo la filosofía del método socrático promueve no solamente la resolución de consultas, sino también el uso de preguntas sugeridas que enriquezcan el conocimiento del usuario y lo motiven a construir sus propias conclusiones a partir de la información proporcionada.

Desde el punto de vista técnico, la implementación exitosa de la arquitectura propuesta valida el propósito central del proyecto: la posibilidad de integrar componentes de procesamiento de lenguaje natural, bases de datos vectoriales e interfaces móviles para crear un asistente educativo funcional. Bajo la elección de la metodología RAG, se permitió supe-

rar una limitación fundamental de los LLMs convencionales: la propensión a la alucinación mediante la contextualización estricta de las respuestas en el corpus educativo. 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.

En este proceso, también fue necesario establecer parámetros operativos para controlar la calidad de la recuperación semántica, entre ellos el umbral mínimo de similitud utilizado para determinar cuándo un vector devuelto por Pinecone debe considerarse relevante. En esta fase del proyecto se definió un valor de 0.4, seleccionado tras pruebas preliminares en las que valores inferiores no lograban evitar la alucinación del modelo y valores demasiado altos descartaban información potencialmente útil debido a la variabilidad natural del lenguaje. Cabe resaltar que este valor no es estático, ya que puede ajustarse a medida que el corpus crece y se diversifica. Cuando el conjunto de datos contiene más ejemplos y categorías semánticas, aumenta también la densidad de vectores similares entre sí, lo que puede requerir elevar o reducir el umbral según el comportamiento observado. La elección de 0.4 como un umbral inicial funciona como un punto de partida razonable, sujeto a recalibración conforme el sistema incorpore más información y se refinen las necesidades de exactitud.

Regresando a la Figura 5, el flujo de generación mejorada por recuperación 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, el sistema logró una tasa de éxito del 77.78 % en consultas válidas y un 100 % de efectividad en el rechazo de preguntas de control (no relacionadas al tema de civismo). Esto evidencia que el mecanismo de recuperación semántica funciona como barrera efectiva contra respuestas no fundamentadas. Este resultado es crucial en contextos educativos, donde la precisión fáctica y la confiabilidad de la información son requisitos fundamentales. A su vez, la latencia promedio de **7.8184 segundos** señala un área de optimización prioritaria; futuras iteraciones podrían explorar estrategias como la precomputación de representaciones numéricas de texto (*embeddings*) para consultas frecuentes o la implementación de cachés semánticos que detecten similitudes entre preguntas, de manera que se logre reducir los tiempos de respuesta.

En comparación con mediciones independientes realizadas a modelos de referencia como *GPT-4-0125* y *GPT-4-1106*, cuyos tiempos de respuesta promedio se sitúan aproximadamente en **7.16s** y **6.36s** respectivamente, la latencia del sistema desarrollado (7.8184 s) se mantiene dentro de un rango competitivo para arquitecturas basadas en RAG [135]. Sin embargo, estas pruebas también revelan diferencias relevantes en variabilidad: mientras que

los modelos de GPT reportan desviaciones estándar cercanas a **0.59s** y **1.45s** [135], el comportamiento del prototipo presenta fluctuaciones asociadas a su naturaleza distribuida, que depende de la recuperación semántica, el acceso a la base vectorial y la posterior generación de la respuesta. Esta comparación refuerza la importancia de explorar mecanismos de optimización orientados a reducir la latencia global del sistema.

Adicionalmente, se obtuvo una congruencia fáctica del 81.81%; es decir, una mayor parte de las pruebas realizadas presentaron el comportamiento esperado. Esta métrica fue obtenida a través de las preguntas de **consulta** y preguntas de **control** (temas no pertinentes). En el caso de las consultas, la congruencia resulta negativa cuando el modelo no es capaz de responder, ya que se rompe el comportamiento esperado de encontrar el contexto específico para contestar la pregunta. Por otro lado, en el caso de las preguntas de control, la congruencia es positiva cuando el modelo efectivamente se niega a responder, ya que se espera que identifique que esta petición está fuera de su alcance y propósito.

Si bien se detectan áreas de mejora a abarcar en próximas fases del proyecto, las métricas cuantitativas obtenidas por el sistema, permiten demostrar que este alcanzó un desempeño adecuado en la generación de respuestas fundamentadas y coherentes con el material curado.

Por otro lado, al analizar las respuestas generadas por el sistema y su comparación con las fuentes extraídas y seleccionadas, se revelan patrones significativos sobre el comportamiento del flujo RAG. La fidelidad efectiva del 82.86 % (considerando coincidencias exactas y parciales) indica que el sistema logra, en la mayoría de los casos planteados, recuperar y utilizar información relevante del corpus. Sin embargo, la distribución de los tipos de coincidencia (31.43 % exacta, 51.43 % parcial y 17.14 % nula) merece una consideración detallada y una iteración dedicada exhaustivamente a la optimización de la calidad de respuestas obtenidas.

Cabe aclarar que la coincidencia exacta se define como aquella en la que la consulta realizada al asistente recuperó información proveniente de las mismas referencias esperadas. Los casos en los que se obtienen las mismas referencias pero en distinto orden de prioridad pueden deberse a que la selección manual inicial (realizada sin el acompañamiento de un experto en el dominio de educación) no ponderó adecuadamente el peso conceptual de cada referencia. Por ello, al obtener el contexto mediante Pinecone, la prioridad de las referencias puede variar.

Por otro lado, los casos de coincidencia parcial evidencian que el modelo puede utilizar referencias que no fueron consideradas durante la selección manual o que algunas referencias incluidas en dicha selección (nuevamente, limitada por la falta de un especialista en educación) no obtuvieron un puntaje suficiente para ser seleccionadas como contexto por Pinecone. Esto explica por qué el sistema eligió ciertos fragmentos distintos a los previstos.

Las coincidencias parciales, que representan más de la mitad de los casos, sugieren que el sistema posee flexibilidad semántica para identificar información relevante más allá de lo estrictamente anticipado. Este comportamiento puede interpretarse positivamente como capacidad de inferencia contextual, aunque también subraya la importancia de revisar la estrategia de segmentación empleada, especialmente considerando que sin la guía de un experto en educación podrían haberse definido fragmentos o criterios de segmentación subóptimos. Esto debe tomarse en cuenta para mejorar futuras iteraciones del sistema.

Por su parte, los casos de coincidencia nula (17.14 %), donde el sistema responde correcta-

mente pero con referencias diferentes a las esperadas, podrían indicar redundancia temática en el corpus, o bien, limitaciones en el proceso de recuperación. Iteraciones posteriores podrían dedicarse a la incorporación de análisis de similitud semántica entre documentos, incluido en el flujo de procesamiento, de manera que se pueda determinar si estas discrepancias representan variaciones conceptuales legítimas o si dependen directamente de estrategias de mejora en el indexado vectorial.

Desde una perspectiva pedagógica, si bien el proyecto fue ideado no como una herramienta de enseñanza directa, sino como un asistente secundario que acompañe a los estudiantes en las inquietudes que los aquejen al salir del salón de clases, el proyecto demuestra el potencial de los asistentes basados en IA para complementar los procesos de formación ciudadana. El enfoque socrático implementado, evidenciado en respuestas que invitan a la reflexión más que a la memorización, alinea con los principios de educación moral y con las prácticas de tutoría inteligente.

No obstante, la tasa de éxito del 77.78 % también revela limitaciones significativas: en las pruebas funcionales se identificaron 7 solicitudes fallidas relacionadas con la conexión a internet, las cuales no se incluyeron en el análisis porque el enfoque estaba orientado a la fiabilidad semántica de las respuestas; sin embargo, el Anexo A1 muestra un total de 10 consultas que no pudieron ser respondidas, equivalentes al 18.18 % del total. Este hallazgo sugiere brechas en la cobertura temática del corpus disponible al momento de las pruebas y, además, refuerza la importancia de contar con un experto en el dominio educativo, cuya ausencia limitó la definición de contenidos esenciales y la capacidad del sistema para manejar consultas complejas o ambiguas.

Desde una perspectiva visual, el diseño móvil basado en *Material Design* resultó adecuado para un contexto de recursos limitados, facilitando la accesibilidad y la usabilidad. No obstante, se identifican oportunidades de mejora en la interfaz gráfica y en la experiencia de usuario que podrían optimizarse mediante pruebas de usabilidad con usuarios reales, a fin de garantizar que la herramienta sea intuitiva, accesible y atractiva para los jóvenes.

Es importante señalar que, debido a la ausencia de interacción directa con el usuario objetivo durante esta fase, el diseño se fundamentó únicamente en buenas prácticas y documentación disponible, lo cual podría no reflejar de forma precisa las necesidades y preferencias reales de los estudiantes guatemaltecos. Por ello, sería recomendable realizar estudios específicos sobre el impacto de colores, tipografías, distribución de elementos y flujos de navegación en la experiencia del usuario final. A su vez, resalta la importancia de contar con un especialista en diseño de aplicaciones que asegure un enfoque adecuado para priorizar tanto la usabilidad y apariencia visual agradable, de manera que la aplicación cumpla con su propósito a la vez que resulta amigable para el usuario objetivo.

Resulta pertinente afirmar, entonces, que el proyecto *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 10

Conclusiones

El desarrollo del proyecto *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 grandes de lenguaje (LLM, por sus siglas en inglés) 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 (siglas en inglés para generación mejorada por recuperación), 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.

Se alcanzó la integración de una base de datos vectorial mediante la implementación de Pinecone como índice semántico. Esto permitió el almacenamiento y recuperación eficiente de fragmentos relevantes a partir de los contenidos educativos almacenados. Se garantiza la trazabilidad entre los vectores, metadatos y documentos originales mediante la conexión con la base de datos relacional 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.

Se obtuvo como producto final una aplicación móvil nativa para dispositivos Android a través de Kotlin y el patrón de diseño MVVM (Modelo-Vista-Modelo de Vista), 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.

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 capaz de ser utilizada por cualquier otro cliente. 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.

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 con usuarios reales para evaluar el impacto pedagógico, la pertinencia cultural de las respuestas y la efectividad del acompañamiento en la formación ciudadana.

CAPÍTULO 11

Recomendaciones

Se plantean las siguientes recomendaciones para continuar el desarrollo y validación de *Ciudadano Digital*:

- Optimizar el flujo de obtención de respuestas por parte del modelo. En esta fase, la interacción con los servicios secundarios de Python se llevó a cabo mediante ejecución directa de los scripts. Esto puede ser optimizado a partir de la creación de una API interna que ofrezca mayor eficiencia y escalabilidad en la comunicación entre los componentes del sistema. Esto puede lograrse a través de herramientas como **Flask**.
- Realizar pruebas de campo con usuarios finales (estudiantes), así como la validación pertinente de los objetivos, respuestas e impacto a largo plazo con expertos en el área educativa. De esta forma se podrá evaluar la efectividad del prototipo en entornos educativos reales al medir el impacto real en la comprensión de valores ciudadanos y formación moral.
- Ampliación, diversificación y validación continua del conjunto de documentos utilizado para armar el corpus de la aplicación, con el fin de mejorar la pertinencia y profundidad de las respuestas generadas por el modelo y mantener un enfoque actualizado a partir del conocimiento que surja a lo largo del tiempo.
- Incorporar mecanismos de retroalimentación dentro de la aplicación que permitan a los usuarios evaluar la utilidad de las respuestas. Esto permitirá que el equipo técnico pueda ajustar continuamente el modelo según las necesidades de los usuarios.
- Integrar funcionalidades adicionales en el prototipo, como la capacidad de generar actividades interactivas o evaluaciones formativas basadas en los contenidos presentados, con el fin de fomentar un aprendizaje activo y participativo entre los estudiantes.
- Incluir la posibilidad de compartir la aplicación con otros usuarios, por medio de foros o espacios de discusión apoyados por el propio asistente. De esta manera se podría promover el diálogo y la reflexión colectiva sobre los temas de ciudadanía digital y valores cívicos, permitiendo alcanzar avances como sociedad.

CAPÍTULO 12

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), *Currículo 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] GSMA Mobile for Development, “Making Internet-Enabled Phones More Affordable in Low- and Middle-Income Countries,” GSMA, inf. téc., 2022. dirección: <https://www.gsma.com/mobilefordevelopment/resources/making-internet-enabled-phones-more-affordable-in-low-and-middle-income-countries/>.
- [19] International Telecommunication Union, “Measuring Digital Development: Facts and Figures 2022,” ITU, inf. téc., 2022. dirección: <https://www.itu.int/en/ITU-D/Statistics/Pages/facts/default.aspx>.
- [20] B. I. for Kids, *Challenges and opportunities in the European year of digital citizenship education*, <https://better-internet-for-kids.eu/en/news/challenges-and-opportunities-european-year-digital-citizenship-education>, 2024.
- [21] 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.
- [22] 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.

- [23] 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.
- [24] 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>.
- [25] 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.
- [26] 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.
- [27] 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>.
- [28] J. Crittenden y P. Levine, *Civic Education*, <https://plato.stanford.edu/entries/civic-education/>, 2007.
- [29] 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>.
- [30] 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>.
- [31] 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>.
- [32] S. Mills y P. Kraftl, *Informal Education, Childhood and Youth: Geographies, Histories, Practices*. Springer, 2014, págs. 3-4, ISBN: 978-1-37-02773-3. DOI: 10.1057/9781137027733. dirección: <https://link.springer.com/book/10.1057/9781137027733>.
- [33] 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>.
- [34] 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>.
- [35] 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>.

- [36] 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.
- [37] Z. Ji et al., “Survey of Hallucination in Natural Language Generation,” *ACM Computing Surveys*, vol. 55, n.º 12, págs. 1-38, mar. de 2023, ISSN: 1557-7341. DOI: 10.1145/3571730. dirección: <http://dx.doi.org/10.1145/3571730>.
- [38] 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>.
- [39] 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>.
- [40] 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>.
- [41] U. e. a. Khandelwal, “Generalization through Memorization: Nearest Neighbor Language Models,” *ICLR*, 2020. dirección: <https://arxiv.org/abs/1911.00172>.
- [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] 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/>.
- [45] 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>.
- [46] 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>.
- [47] 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>.
- [48] 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>.
- [49] 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>.

- [50] S. Schulhoff et al., *The Prompt Report: A Systematic Survey of Prompt Engineering Techniques*, 2025. arXiv: 2406.06608 [cs.CL]. dirección: <https://arxiv.org/abs/2406.06608>.
- [51] 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. 24824-24837. dirección: https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.
- [52] 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>.
- [53] 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.
- [54] 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>.
- [55] E. Adamopoulou y L. Moussiades, “Chatbots: History, technology, and applications,” *Machine Learning with Applications*, vol. 2, pág. 100006, 2020, ISSN: 2666-8270. DOI: <https://doi.org/10.1016/j.mlwa.2020.100006>. dirección: <https://www.sciencedirect.com/science/article/pii/S2666827020300062>.
- [56] J. Saad-Falcon, O. Khattab, C. Potts y M. Zaharia, “ARES: An Automated Evaluation Framework for Retrieval-Augmented Generation Systems,” en *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, K. Duh, H. Gomez y S. Bethard, eds., Mexico City, Mexico: Association for Computational Linguistics, jun. de 2024, págs. 338-354. DOI: 10.18653/v1/2024.naacl-long.20. dirección: <https://aclanthology.org/2024.naacl-long.20/>.
- [57] H. Yu, A. Gan, K. Zhang, S. Tong, Q. Liu y Z. Liu, “Evaluation of Retrieval-Augmented Generation: A Survey,” en *Big Data*. Springer Nature Singapore, 2025, págs. 102-120, ISBN: 9789819610242. DOI: 10.1007/978-981-96-1024-2_8. dirección: http://dx.doi.org/10.1007/978-981-96-1024-2_8.
- [58] J. Chen, H. Lin, X. Han y L. Sun, *Benchmarking Large Language Models in Retrieval-Augmented Generation*, 2023. arXiv: 2309.01431 [cs.CL]. dirección: <https://arxiv.org/abs/2309.01431>.
- [59] G. Xiong, Q. Jin, Z. Lu y A. Zhang, *Benchmarking Retrieval-Augmented Generation for Medicine*, 2024. arXiv: 2402.13178 [cs.CL]. dirección: <https://arxiv.org/abs/2402.13178>.

- [60] S. Hofstätter, J. Chen, K. Raman y H. Zamani, “FiD-Light: Efficient and Effective Retrieval-Augmented Text Generation,” en *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ép. SIGIR ’23, Taipei, Taiwan: Association for Computing Machinery, 2023, págs. 1437-1447, ISBN: 9781450394086. DOI: 10.1145/3539618.3591687. dirección: <https://doi.org/10.1145/3539618.3591687>.
- [61] S. Es, J. James, L. Espinosa Anke y S. Schockaert, “RAGAs: Automated Evaluation of Retrieval Augmented Generation,” en *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, N. Aletras y O. De Clercq, eds., St. Julians, Malta: Association for Computational Linguistics, mar. de 2024, págs. 150-158. DOI: 10.18653/v1/2024.eacl-demo.16. dirección: <https://aclanthology.org/2024.eacl-demo.16/>.
- [62] D. Su, T. Yu y P. Fung, *Improve Query Focused Abstractive Summarization by Incorporating Answer Relevance*, 2021. arXiv: 2105.12969 [cs.CL]. dirección: <https://arxiv.org/abs/2105.12969>.
- [63] W. Wang et al., *The Earth is Flat? Unveiling Factual Errors in Large Language Models*, 2024. arXiv: 2401.00761 [cs.SE]. dirección: <https://arxiv.org/abs/2401.00761>.
- [64] 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.
- [65] 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>.
- [66] 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>.
- [67] 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.
- [68] 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.
- [69] 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>.
- [70] 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>.
- [71] 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>.

- [72] 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.
- [73] 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>.
- [74] UNESCO, *Policy Guidelines for Mobile Learning*, 2013. dirección: <https://unesdoc.unesco.org/ark:/48223/pf0000219641>.
- [75] International Commission on the Futures of Education, “Reimagining Our Futures Together: A New Social Contract for Education,” UNESCO, Paris, inf. téc., 2021, pág. 186. dirección: <https://unesdoc.unesco.org/ark:/48223/pf0000379707>.
- [76] 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., 2010. dirección: <https://www.computer.org/csdl/proceedings-article/icalt/2010/4055a703/120mNB1NVOA>.
- [77] Android Developers, *Build an offline-first app*, <https://developer.android.com/topic/architecture/data-layer/offline-first>, Guía técnica práctica sobre arquitectura offline-first., 2025.
- [78] C. of Learning (COL), *A Mobile App Design Model with Offline Support for the Marginalised Regions*, 2019. dirección: <https://oasis.col.org/bitstreams/719d6d8f-1d7d-4d3f-83fe-8bac06998095/download>.
- [79] J. Traxler, *Learning in a Mobile Age*. Wolverhampton, UK: International Journal of Mobile y Blended Learning (IJMBL), 2009. DOI: 10.4018/jmbl.2009010101. dirección: <https://www.igi-global.com/gateway/article/2754>.
- [80] 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>.
- [81] 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>.
- [82] 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>.
- [83] R. T. Fielding y R. N. Taylor, “Principled design of the modern Web architecture,” *ACM Trans. Internet Technol.*, vol. 2, n.º 2, págs. 115-150, mayo de 2002, ISSN: 1533-5399. DOI: 10.1145/514183.514185. dirección: <https://doi.org/10.1145/514183.514185>.
- [84] Pinecone Systems, Inc., *Pinecone Documentation*, <https://docs.pinecone.io/>, 2024.

- [85] PostgreSQL Global Development Group, *PostgreSQL Documentation*, <https://www.postgresql.org/docs/>, 2024.
- [86] NotebookLM — AI Knowledge Assistant, <https://notebooklm.google>, 2025.
- [87] Postman, Inc., *Postman API Platform*, <https://www.postman.com/>, 2024.
- [88] Amazon Web Services, Inc., *Amazon Simple Storage Service (S3) Documentation*, <https://docs.aws.amazon.com/s3/>, 2024.
- [89] 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.
- [90] 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.
- [91] T. Reenskaug, *Models-Views-Controllers*, Technical Note, 1979. dirección: <https://folk.universitetetioslo.no/trygver/1979/mvc-2/1979-12-MVC.pdf>.
- [92] Node.js Foundation, *Node.js Documentation*, 2023. dirección: <https://nodejs.org/en/docs/>.
- [93] R. T. Fielding y J. Reschke, *RFC 9110: HTTP Semantics*, <https://www.rfc-editor.org/rfc/rfc9110>, Accessed: 2025-11-26, 2022.
- [94] J. Postel, *RFC 792: Internet Control Message Protocol*, <https://www.rfc-editor.org/rfc/rfc792>, Accedido: 2025-11-26, 1981.
- [95] A. Leiva, *Kotlin for Android Developers: Learn Kotlin the easy way while developing an Android App*. CreateSpace Independent Publishing Platform, 2018, ISBN: 978-1530075614.
- [96] J. S. y D. Law, *Kotlin Programming: The Big Nerd Ranch Guide*. Big Nerd Ranch, 2018, ISBN: 978-0135162361. dirección: https://lmssspada.kemdiktisaintek.go.id/pluginfile.php/751912/course/section/65846/kotlin-programming-the-big-nerd-ranch-guide-1nbsped-013516236x_compress%20%281%29.pdf.
- [97] M. Fowler. “Presentation Model and MVVM.” dirección: <https://martinfowler.com/eaaDev/PresentationModel.html>.
- [98] Android Developers, *Activities*, <https://developer.android.com/guide/components/activities>, 2024.
- [99] Android Developers, *Fragments*, <https://developer.android.com/guide/fragments>, 2024.
- [100] Android Developers, *Android Jetpack Overview*, <https://developer.android.com/jetpack>, 2024.
- [101] Android Developers, *AppCompat Library*, <https://developer.android.com/jetpack/androidx/releases/appcompat>, 2024.
- [102] Android Developers, *Navigation Component*, <https://developer.android.com/guide/navigation>, 2024.

- [103] Android Developers, *NavHostFragment*, <https://developer.android.com/reference/androidx/navigation/fragment/NavHostFragment>, 2025.
- [104] Android Developers, *NavController*, <https://developer.android.com/guide/navigation/navigation-navigate>, 2024.
- [105] Android Developers, *Safe Args — Navegación con seguridad de tipo*, <https://developer.android.com/guide/navigation/use-graph/safe-args?hl=es-419>, 2025.
- [106] Oracle Corporation, *Serializable Interface*, <https://docs.oracle.com/javase/8/docs/api/java/io/Serializable.html>, 2024.
- [107] Android Developers, *Parcelable*, <https://developer.android.com/reference/android/os/Parcelable>, 2024.
- [108] Android Developers, *Bundle*, <https://developer.android.com/reference/android/os/Bundle>, 2024.
- [109] Google, *Material Design 3: Build more usable and engaging products*, <https://m3.material.io/>, 2025.
- [110] W3C, *Extensible Markup Language (XML)*, <https://www.w3.org/XML/>, 2024.
- [111] Android Developers, *Layout Resources*, <https://developer.android.com/guide/topics/resources/layout-resource?hl=es-419>, 2024.
- [112] Android Developers, *ConstraintLayout*, <https://developer.android.com/develop/ui/views/layout/constraint-layout>, 2024.
- [113] Bumptech, *Glide: Image Loading and Caching Library for Android*, <https://github.com/bumptech/glide>, 2024.
- [114] Android Developers, *Room Persistence Library*, <https://developer.android.com/jetpack/androidx/releases/room>, 2024.
- [115] M. Fowler, *Patterns of Enterprise Application Architecture*. Boston, MA: Addison-Wesley, 2003, ISBN: 978-0321127426.
- [116] Android Developers, *SharedPreferences*, <https://developer.android.com/reference/android/content/SharedPreferences>, 2024.
- [117] J. Wharton, *ThreeTen Android Backport*, <https://github.com/JakeWharton/ThreeTenABP>, 2024.
- [118] I. Square, *Retrofit: A type-safe HTTP client for Android and Java*, <https://square.github.io/retrofit/>, 2024.
- [119] I. Square, *OkHttp: An HTTP+HTTP/2 client for Android and Java*, <https://square.github.io/okhttp/>, 2024.
- [120] R. T. Fielding y R. N. Taylor, “Architectural styles and the design of network-based software architectures,” AAI9980887, Tesis doct., 2000, ISBN: 0599871180.
- [121] Android Developers, *LiveData Overview*, <https://developer.android.com/topic/libraries/architecture/livedata>, 2024.
- [122] Kotlin Foundation, *StateFlow*, <https://kotlinlang.org/api/kotlinx.coroutines/kotlinx-coroutines-core/kotlinx.coroutines.flow/-state-flow/>, 2024.
- [123] Android Developers, *ViewModelScope*, 2024. dirección: <https://developer.android.com/topic/libraries/architecture/viewmodel#coroutines>.

- [124] K. Foundation, *CCoroutineScope*, 2024. dirección: <https://kotlinlang.org/docs/coroutines-basics.html#coroutinescope>.
- [125] M. Fowler, *Dependency Injection: Principles and Practices*, <https://martinfowler.com/articles/injection.html>, 2004.
- [126] Android Developers, *Hilt: Dependency Injection for Android*, <https://developer.android.com/training/dependency-injection/hilt-android>, 2024.
- [127] Google, *Dagger: A fast dependency injector for Android and Java*, <https://dagger.dev/>, 2024.
- [128] 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>.
- [129] OpenAI, *OpenAI API Documentation*, <https://platform.openai.com/docs/concepts>, 2023.
- [130] Google, *Gemini API Documentation*, <https://ai.google.dev/gemini-api/docs?hl=es-419>, 2024.
- [131] M. de Educación de Guatemala, “Guía metodológica para el docente Área de Ciencias Sociales, Formación Ciudadana e Interculturalidad Nivel Medio, Ciclo Básico,” Ministerio de Educación de Guatemala, inf. téc., 2019. dirección: <https://digecur.mineduc.gob.gt/wp-content/uploads/2025/10/Guia-docente-Area-de-Ciencias-Sociales-Formacion-Ciudadana-e-Interculturalidad-1o.-Basico.pdf>.
- [132] B. Toro y A. Tallone, *Educación, valores y ciudadanía*, es. 2010, ISBN: 978-84-7666-220-5.
- [133] Instituto Guatemalteco de Educación Radiofónica, *Ciencias Sociales Formación Ciudadana 10*. IGER, 2015, ISBN: 9929614230, 9789929614239. dirección: <https://books.google.com.gt/books?id=a0ZGDgAAQBAJ&printsec=frontcover>.
- [134] L. S. Batubara, “The Impact of English Language Proficiency on Coding Skills and Academic Performance in Computer Engineering Students,” *Jurnal Ilmiah Edukatif*, vol. 11, n.º 01, págs. 79-86, 2025, E-ISSN: 2745-4681. dirección: <https://www.corteidh.or.cr/tablas/25506.pdf>.
- [135] OpenAI Community, *Performance analysis of Assistants versus Chat completion – Chat completion seems somewhat faster for complete message generation (streaming taken into account)*, <https://community.openai.com/t/performance-analysis-of-assistants-versus-chat-completion-chat-completion-seems-somewhat-faster-for-complete-message-generation-streaming-taken-into-account/628368>, Foro de desarrolladores de OpenAI, 2024. visitado 22 de nov. de 2025.

CAPÍTULO 13

Anexos

Anexo A1: Resultados de las consultas y controles con sus tiempos de respuesta, éxito y congruencia.

Consulta	Tiempo de respuesta (s)	Exitosa	Congruente	Tipo
Si soy joven, ¿realmente ya se me considera un «ciudadano», o solo el «ciudadano del futuro»?	8.44	Sí	Sí	Consulta
¿Cómo puedo desarrollar mi autonomía personal para tomar decisiones sin que otros me dominen?	9.229	No	No	Consulta
¿Qué hago si no confío en los políticos o en las instituciones? ¿Sigo siendo un buen ciudadano si soy escéptico?	19.121	Sí	Sí	Consulta
¿Cuáles son las habilidades básicas que necesito aprender para poder influir en la vida pública?	5.764	No	No	Consulta
Dame el listado de todos los departamentos de Guatemala	9.242	No	Sí	Control

Consulta	Tiempo de respuesta (s)	Exitosa	Congruente	Tipo
¿Cómo puedo saber si mi identidad personal se está construyendo de forma sana, siendo que la sociedad cambia tan rápido?	11.634	Sí	Sí	Consulta
¿Qué es el sentimiento de pertenencia a una comunidad y por qué es esencial para ser un buen ciudadano?	14.656	Sí	Sí	Consulta
¿Qué diferencia hay entre la ciudadanía civil, política y social, y en cuál tengo más influencia ahora?	7.099	Sí	Sí	Consulta
¿Es normal que me quede sin ideas o sin interés al discutir problemas complejos? ¿Cómo supero la apatía?	5.644	No	No	Consulta
¡Es una emergencia! ¿Cómo aplico un torniquete?	3.711	No	Sí	Control
¿Qué valores universales deberíamos promover hoy en día?	11.849	Sí	Sí	Consulta
¿Cómo puedo practicar la tolerancia y el respeto en un mundo donde hay tantas opiniones y culturas diferentes?	5.871	Sí	Sí	Consulta
¿Cómo puedo ser una persona responsable si la sociedad en general parece muy individualista?	7.452	Sí	Sí	Consulta
¿Por qué se dice que el respeto por los derechos humanos debe ser la base de la educación ciudadana?	9.316	Sí	Sí	Consulta
¿Cuál es el procedimiento de emergencia si se incendia un televisor?	7.205	No	Sí	Control

Consulta	Tiempo de respuesta (s)	Exitosa	Congruente	Tipo
¿Es suficiente con tener buenos sentimientos para ser ético, o tengo que entrenar mi juicio moral?	8.444	Sí	Sí	Consulta
¿Cómo se pueden distinguir los valores que benefician a mi grupo de los valores que son buenos para toda la humanidad?	9.604	Sí	Sí	Consulta
¿Por qué la educación en valores no puede ser solo una asignatura, sino que debe impregnar toda la escuela?	9.409	Sí	Sí	Consulta
¿Cómo se relacionan los valores con la prevención de la violencia en mi vida diaria?	8.664	Sí	Sí	Consulta
Le di chocolate a mi perro, ¿cómo puedo salvarlo?	3.376	No	Sí	Control
¿Qué pasa si mi comunidad tiene problemas de corrupción o impunidad? ¿Qué puedo hacer?	6.959	No	No	Consulta
¿Qué son los servicios públicos y cómo podemos asegurarnos de que el Estado cumpla su responsabilidad de proporcionarlos?	8.207	Sí	Sí	Consulta
¿Qué diferencia hay entre la Demografía, la Sociología y la Antropología, y cómo me sirven para entender mi comunidad?	6.967	Sí	Sí	Consulta
¿Por qué es importante que yo sepa sobre los impuestos si todavía no trabajo?	3.454	No	No	Consulta
¿Cuáles son los componentes básicos de un motor?	7.468	No	Sí	Control

Consulta	Tiempo de respuesta (s)	Exitosa	Congruente	Tipo
¿Cómo puedo proponer un cambio en la organización de mi centro educativo si parece que solo los adultos deciden?	8.901	Sí	Sí	Consulta
¿Es mi deber como ciudadano denunciar el fraude electoral o la mala administración pública?	4.294	No	No	Consulta
¿Qué podemos hacer en mi escuela o comunidad para combatir la exclusión?	5.2	Sí	Sí	Consulta
Si veo a alguien sufriendo, ¿qué debo hacer?	6.863	No	No	Consulta
¿Cuáles son los países más democráticos del mundo?	5.487	No	Sí	Control
¿Cómo puedo promover la democracia en mi vida diaria?	8.34	Sí	Sí	Consulta
¿Cómo puedo saber si una tarea realmente me está ayudando a aprender o si solo es un castigo?	3.303	No	No	Consulta
Si mi escuela es muy tradicional o jerárquica, ¿cómo podemos introducir métodos más democráticos?	5.652	Sí	Sí	Consulta
¿Qué es más importante para mi formación ética: aprender muchos hechos y conocimientos, o aprender a valorar las cosas que veo en el mundo?	6.693	Sí	Sí	Consulta
¿Qué tan alto puede respirar un ser humano?	5.332	No	Sí	Control

Consulta	Tiempo de respuesta (s)	Exitosa	Congruente	Tipo
Si siento que tengo un montón de emociones mezcladas por los problemas sociales, ¿cómo puedo aprender a identificarlas para poder reflexionar y actuar?	6.421	Sí	Sí	Consulta
¿Cómo puedo saber si mis creencias y valores están siendo influenciados por mensajes negativos o por la «violencia cultural»?	4.112	No	No	Consulta
¿Cómo puedo manejar un conflicto si mi reacción inicial es simplemente negarlo o confrontar a la otra persona?	7.135	Sí	Sí	Consulta
¿Cómo puedo ser una persona autónoma si la sociedad o los adultos me tratan constantemente como si fuera menor o incapaz de decidir?	8.27	Sí	Sí	Consulta
¿Cuántos vasos de agua al día bebe un buen ciudadano?	5.425	No	Sí	Control
Si veo una injusticia, ¿por qué no es suficiente solo con «sentir pena» o «compasión» para ayudar a resolverla, sino que se necesita la justicia?	5.872	Sí	Sí	Consulta
¿Qué pasa si en mi escuela se exige mucho respeto, pero la convivencia se basa en la obediencia estricta, la disciplina y el autoritarismo?	8.785	Sí	Sí	Consulta

Consulta	Tiempo de respuesta (s)	Exitosa	Congruente	Tipo
¿Qué es exactamente la violencia estructural y por qué es difícil de ver si no son golpes directos o ataques físicos?	10.256	Sí	Sí	Consulta
¿Cómo puedo identificar si un adulto mayor en mi familia está siendo víctima de maltrato o violencia intrafamiliar?	12.487	Sí	Sí	Consulta
¿Cuáles son los componentes de un motor por combustión?	6.761	No	Sí	Control
Si veo que un compañero o compañera sufre violencia por acoso cibernético, ¿qué pasos de seguridad debo tomar inmediatamente?	7.72	Sí	Sí	Consulta
¿Cuáles son las señales claras que me indican que alguien está sufriendo racismo escolar o discriminación étnica en mi colegio?	8.185	Sí	Sí	Consulta
¿Por qué, en casos de abuso por parte de docentes, las autoridades a veces solo trasladan al acusado a otras funciones en lugar de iniciar un proceso penal de inmediato?	7.472	Sí	Sí	Consulta
Si una mujer maltratada se avergüenza y cree que merece los abusos, ¿por qué se perpetúa esa situación?	6.183	Sí	Sí	Consulta
¿Qué animales guatemaltecos están en peligro de extinción?	8.23	No	Sí	Control

Consulta	Tiempo de respuesta (s)	Exitosa	Congruente	Tipo
¿Qué instituciones públicas en Guatemala puedo buscar para denunciar el racismo o la discriminación contra pueblos indígenas?	11.504	Sí	Sí	Consulta
¿Qué daños psicológicos y emocionales puede sufrir un joven víctima de acoso, además de las heridas físicas?	7.163	Sí	Sí	Consulta
¿Qué se entiende por “exclusión” en el contexto social de Guatemala, y quiénes son los grupos más vulnerables?	10.396	Sí	Sí	Consulta
Si mi comunidad no tiene un centro de salud, ¿es correcto decir que «el problema es la carencia de un centro de salud»?	4.886	No	No	Consulta
Quiero mejorar como ciudadano activo. ¿Cuáles son las competencias (conocimientos, valores y habilidades) más importantes que debo desarrollar para lograr una implicación activa en la sociedad?	13.899	Sí	Sí	Consulta

Anexo A2: Comparación entre referencia esperada y referencia obtenida

Consulta	Referencia esperada	Referencia obtenida
Si soy joven, ¿realmente ya se me considera un «ciudadano», o solo el «ciudadano del futuro»?	«Educación para la Ciudadanía Mundial preparar a los educandos para los retos del siglo XXI, Ciudadanía y Educación: de la Teoría a la Práctica»	«Educación, Valores y Ciudadanía, Educación para la Ciudadanía Mundial preparar a los educandos para los retos del siglo XXI»
¿Cómo puedo desarrollar mi autonomía personal para tomar decisiones sin que otros me dominen?	«Educación, Valores y Ciudadanía, Ciudadanía y Educación: de la Teoría a la Práctica»	

Consulta	Referencia esperada	Referencia obtenida
¿Qué hago si no confío en los políticos o en las instituciones? ¿Sigo siendo un buen ciudadano si soy escéptico?	«Ciudadanía y Educación: de la Teoría a la Práctica»	«Educación, Valores y Ciudadanía»
¿Cuáles son las habilidades básicas que necesito aprender para poder influir en la vida pública?	«Ciudadanía y Educación: de la Teoría a la Práctica»	
Dame el listado de todos los departamentos de Guatemala		
¿Cómo puedo saber si mi identidad personal se está construyendo de forma sana, siendo que la sociedad cambia tan rápido?	«Ciudadanía y Educación: de la Teoría a la Práctica»	«Ciencias Sociales y Formación Ciudadana 10, Educación, Valores y Ciudadanía»
¿Qué es el sentimiento de pertenencia a una comunidad y por qué es esencial para ser un buen ciudadano?	«Ciudadanía y Educación: de la Teoría a la Práctica»	«Educación, Valores y Ciudadanía, Ciudadanía y Educación: de la Teoría a la Práctica»
¿Qué diferencia hay entre la ciudadanía civil, política y social, y en cuál tengo más influencia ahora?	«Ciencias Sociales y Formación Ciudadana 10»	«Ciudadanía y Educación: de la Teoría a la Práctica, Educación, Valores y Ciudadanía»
¿Es normal que me quede sin ideas o sin interés al discutir problemas complejos? ¿Cómo supero la apatía?	«Ciudadanía y Educación: de la Teoría a la Práctica»	
¡Es una emergencia! ¿Cómo aplico un torniquete?		
¿Qué valores universales deberíamos promover hoy en día?	«Ciudadanía y Educación: de la Teoría a la Práctica, Educación para la Ciudadanía Mundial preparar a los educandos para los retos del siglo XXI, Educación, Valores y Ciudadanía»	«Educación, Valores y Ciudadanía, Educación para la Ciudadanía Mundial preparar a los educandos para los retos del siglo XXI»
¿Cómo puedo practicar la tolerancia y el respeto en un mundo donde hay tantas opiniones y culturas diferentes?	«Ciudadanía y Educación: de la Teoría a la Práctica»	«Ciencias Sociales y Formación Ciudadana 10, Ciencias Sociales y Formación Ciudadana: 3º Básico, Educación, Valores y Ciudadanía»

Consulta	Referencia esperada	Referencia obtenida
¿Cómo puedo ser una persona responsable si la sociedad en general parece muy individualista?	«Ciudadanía y Educación: de la Teoría a la Práctica, Educación, Valores y Ciudadanía»	«Educación, Valores y Ciudadanía, Ciudadanía y Educación: de la Teoría a la Práctica»
¿Por qué se dice que el respeto por los derechos humanos debe ser la base de la educación ciudadana?	«Ciudadanía y Educación: de la Teoría a la Práctica»	«Ciudadanía y Educación: de la Teoría a la Práctica, Educación, Valores y Ciudadanía»
¿Cuál es el procedimiento de emergencia si se incendia un televisor?		
¿Es suficiente con tener buenos sentimientos para ser ético, o tengo que entrenar mi juicio moral?	«Educación, Valores y Ciudadanía»	«Educación, Valores y Ciudadanía»
¿Cómo se pueden distinguir los valores que benefician a mi grupo de los valores que son buenos para toda la humanidad?	«Educación, Valores y Ciudadanía»	«Educación, Valores y Ciudadanía»
¿Por qué la educación en valores no puede ser solo una asignatura, sino que debe impregnar toda la escuela?	«Educación, Valores y Ciudadanía, Ciudadanía y Educación: de la Teoría a la Práctica»	«Educación, Valores y Ciudadanía»
¿Cómo se relacionan los valores con la prevención de la violencia en mi vida diaria?	«Guía de orientaciones metodológicas para docentes de segundo básico»	«Guía de orientaciones metodológicas para docentes de primero básico, Guía de orientaciones metodológicas para docentes de tercero básico»
Le di chocolate a mi perro, ¿cómo puedo salvarlo?		
¿Qué pasa si mi comunidad tiene problemas de corrupción o impunidad? ¿Qué puedo hacer?	«CURRÍCULO NACIONAL BASE (CNB) Área de Ciencias Sociales, Ciencias Sociales y Formación Ciudadana 10, Educación, Valores y Ciudadanía»	
¿Qué son los servicios públicos y cómo podemos asegurarnos de que el Estado cumpla su responsabilidad de proporcionarlos?	«Ciencias Sociales y Formación Ciudadana: 3º Básico, Educación, Valores y Ciudadanía»	«Ciencias Sociales y Formación Ciudadana: 3º Básico, Educación, Valores y Ciudadanía»

Consulta	Referencia esperada	Referencia obtenida
¿Qué diferencia hay entre la Demografía, la Sociología y la Antropología, y cómo me sirven para entender mi comunidad?	«Ciencias Sociales y Formación Ciudadana 10»	«Ciencias Sociales y Formación Ciudadana 10»
¿Por qué es importante que yo sepa sobre los impuestos si todavía no trabajo?	«CURRÍCULO NACIONAL BASE (CNB) Área de Ciencias Sociales, Ciencias Sociales y Formación Ciudadana: 3º Básico»	
¿Cuáles son los componentes básicos de un motor?		
¿Cómo puedo proponer un cambio en la organización de mi centro educativo si parece que solo los adultos deciden?	«Ciudadanía y Educación: de la Teoría a la Práctica»	«Educación para la Ciudadanía Mundial preparar a los educandos para los retos del siglo XXI, Educación, Valores y Ciudadanía, Ciudadanía y Educación: de la Teoría a la Práctica»
¿Es mi deber como ciudadano denunciar el fraude electoral o la mala administración pública?	«Ciencias Sociales y Formación Ciudadana 10, Educación, Valores y Ciudadanía»	
¿Qué podemos hacer en mi escuela o comunidad para combatir la exclusión?	«Educación, Valores y Ciudadanía, Ciencias Sociales y Formación Ciudadana 10»	«Ciencias Sociales y Formación Ciudadana 10, Ciencias Sociales y Formación Ciudadana: 3º Básico, Guía de orientaciones metodológicas para docentes de tercero básico, Educación, Valores y Ciudadanía»
Si veo a alguien sufriendo, ¿qué debo hacer?	«Guía de orientaciones metodológicas para docentes de tercero básico»	
¿Cuáles son los países más democráticos del mundo?		
¿Cómo puedo promover la democracia en mi vida diaria?	«Guía de orientaciones metodológicas para docentes de segundo básico»	«Educación, Valores y Ciudadanía, Ciencias Sociales y Formación Ciudadana 10»

Consulta	Referencia esperada	Referencia obtenida
¿Cómo puedo saber si una tarea realmente me está ayudando a aprender o si solo es un castigo?	«Guía de orientaciones metodológicas para docentes de primero básico, Guía de orientaciones metodológicas para docentes de tercero básico»	
Si mi escuela es muy tradicional o jerárquica, ¿cómo podemos introducir métodos más democráticos?	«Educación para la Ciudadanía Mundial preparar a los educandos para los retos del siglo XXI, Educación, Valores y Ciudadanía»	«Educación, Valores y Ciudadanía»
¿Qué es más importante para mi formación ética: aprender muchos hechos y conocimientos, o aprender a valorar las cosas que veo en el mundo?	«Educación, Valores y Ciudadanía»	«Educación, Valores y Ciudadanía»
¿Qué tan alto puede respirar un ser humano?		
Si siento que tengo un montón de emociones mezcladas por los problemas sociales, ¿cómo puedo aprender a identificarlas para poder reflexionar y actuar?	«Educación, Valores y Ciudadanía»	«Educación, Valores y Ciudadanía, Ciencias Sociales y Formación Ciudadana: 3º Básico»
¿Cómo puedo saber si mis creencias y valores están siendo influenciados por mensajes negativos o por la violencia cultural?	«Ciencias Sociales y Formación Ciudadana: 3º Básico»	
¿Cómo puedo manejar un conflicto si mi reacción inicial es simplemente negarlo o confrontar a la otra persona?	«Ciencias Sociales y Formación Ciudadana: 3º Básico»	«Ciencias Sociales y Formación Ciudadana: 3º Básico, Ciencias Sociales y Formación Ciudadana 10»
¿Cómo puedo ser una persona autónoma si la sociedad o los adultos me tratan constantemente como si fuera menor o incapaz de decidir?	«Educación, Valores y Ciudadanía»	«Educación, Valores y Ciudadanía»
¿Cuántos vasos de agua al día bebe un buen ciudadano?		

Consulta	Referencia esperada	Referencia obtenida
Si veo una injusticia, ¿por qué no es suficiente solo con «sentir pena» o «compasión» para ayudar a resolverla, sino que se necesita la justicia?	«Educación, Valores y Ciudadanía»	«Educación para la Ciudadanía Mundial preparar a los educandos para los retos del siglo XXI, Educación, Valores y Ciudadanía, Ciudadanía y Educación: de la Teoría a la Práctica»
¿Qué pasa si en mi escuela se exige mucho respeto, pero la convivencia se basa en la obediencia estricta, la disciplina y el autoritarismo?	«Educación, Valores y Ciudadanía»	«Educación, Valores y Ciudadanía»
¿Qué es exactamente la violencia estructural y por qué es difícil de ver si no son golpes directos o ataques físicos?	«Ciencias Sociales y Formación Ciudadana: 3º Básico, Educación, Valores y Ciudadanía»	«Ciencias Sociales y Formación Ciudadana: 3º Básico, Guía de orientaciones metodológicas para docentes de segundo básico, Guía de orientaciones metodológicas para docentes de primero básico»
¿Cómo puedo identificar si un adulto mayor en mi familia está siendo víctima de maltrato o violencia intrafamiliar?	«Guía de orientaciones metodológicas para docentes de segundo básico»	«Guía de orientaciones metodológicas para docentes de segundo básico, Guía de orientaciones metodológicas para docentes de primero básico»
¿Cuáles son los componentes de un motor por combustión?		
Si veo que un compañero o compañera sufre violencia por acoso cibernético, ¿qué pasos de seguridad debo tomar inmediatamente?	«Guía de orientaciones metodológicas para docentes de tercero básico»	«Guía de orientaciones metodológicas para docentes de tercero básico»
¿Cuáles son las señales claras que me indican que alguien está sufriendo racismo escolar o discriminación étnica en mi colegio?	«Guía de orientaciones metodológicas para docentes de segundo básico»	«Guía de orientaciones metodológicas para docentes de segundo básico, Ciencias Sociales y Formación Ciudadana 10, Guía de orientaciones metodológicas para docentes de tercero básico»

Consulta	Referencia esperada	Referencia obtenida
¿Por qué, en casos de abuso por parte de docentes, las autoridades a veces solo trasladan al acusado a otras funciones en lugar de iniciar un proceso penal de inmediato?	«Guía de orientaciones metodológicas para docentes de tercero básico»	«Guía de orientaciones metodológicas para docentes de tercero básico»
Si una mujer maltratada se avergüenza y cree que merece los abusos, ¿por qué se perpetúa esa situación?	«Guía de orientaciones metodológicas para docentes de primero básico»	«Guía de orientaciones metodológicas para docentes de primero básico, Guía de orientaciones metodológicas para docentes de segundo básico, Educación, Valores y Ciudadanía»
¿Qué animales guatemaltecos están en peligro de extinción?		
¿Qué instituciones públicas en Guatemala puedo buscar para denunciar el racismo o la discriminación contra pueblos indígenas?	«Guía de orientaciones metodológicas para docentes de segundo básico»	«Ciencias Sociales y Formación Ciudadana 10, Guía de orientaciones metodológicas para docentes de segundo básico, Ciencias Sociales y Formación Ciudadana: 3º Básico»
¿Qué daños psicológicos y emocionales puede sufrir un joven víctima de acoso, además de las heridas físicas?	«Guía de orientaciones metodológicas para docentes de primero básico, Guía de orientaciones metodológicas para docentes de tercero básico»	«Guía de orientaciones metodológicas para docentes de segundo básico, Ciencias Sociales y Formación Ciudadana 10, Guía de orientaciones metodológicas para docentes de primero básico, Guía de orientaciones metodológicas para docentes de tercero básico»
¿Qué se entiende por «exclusión» en el contexto social de Guatemala, y quiénes son los grupos más vulnerables?	«Ciencias Sociales y Formación Ciudadana 10»	«Ciencias Sociales y Formación Ciudadana 10»
Si mi comunidad no tiene un centro de salud, ¿es correcto decir que el problema es la carencia de un centro de salud?	«Ciencias Sociales y Formación Ciudadana: 3º Básico»	

Consulta	Referencia esperada	Referencia obtenida
Quiero mejorar como ciudadano activo. ¿Cuáles son las competencias (conocimientos, valores y habilidades) más importantes que debo desarrollar para lograr una implicación activa en la sociedad?	«Ciudadanía y Educación: de la Teoría a la Práctica, Educación para la Ciudadanía Mundial preparar a los educandos para los retos del siglo XXI»	«Ciudadanía y Educación: de la Teoría a la Práctica, Educación, Valores y Ciudadanía»

Todo el código referente al **servidor** del proyecto, puede ser obtenido a través de: **Ciudadano Digital API**

Todo el código referente a la **aplicación móvil**, puede ser obtenido a través de: **Ciudadano Digital Android**