

UNIVERSIDAD CENTROAMERICANA
JOSÉ SIMEÓN CAÑAS



IMPLEMENTACIÓN DE PROTOTIPO PARA “UCACHAT”

TRABAJO DE GRADUACIÓN PREPARADO PARA LA
FACULTAD DE INGENIERÍA Y ARQUITECTURA

PARA OPTAR AL GRADO DE
INGENIERO(A) INFORMÁTICO(A)

POR:

WALTER RAFAEL MORALES HENRIQUEZ
MONICA ALEJANDRA VEGA FLORES
OMAR ALFREDO VASQUEZ ESCAMILLA
RENE ARMANDO FLORES CORTEZ

MES Y AÑO DE GRADUACIÓN
ANTIGUO CUSCATLÁN, EL SALVADOR, C.A.

RECTOR
NOMBRE DEL RECTOR

SECRETARIO GENERAL
NOMBRE DEL SECRETARIO O SECRETARIA GENERAL

DECANO DE LA FACULTAD DE INGENIERÍA Y ARQUITECTURA
NOMBRE DEL DECANO DE LA FACULTAD DE INGENIERÍA Y
ARQUITECTURA

DIRECTOR DE LA CARRERA DE INGENIERÍA INFORMÁTICA
NOMBRE DEL DIRECTOR DE LA CARRERA

DIRECTOR DEL TRABAJO
NOMBRE DEL DIRECTOR DEL TRABAJO DE GRADUACIÓN

LECTOR
NOMBRE DEL LECTOR O LECTORA

RESUMEN

La comunidad estudiantil de la Universidad Centroamericana José Simeón Cañas (UCA) enfrenta el desafío de acceder a información institucional, la cual se encuentra dispersa en múltiples fuentes y formatos. Esta descentralización genera ineficiencias, pérdida de tiempo y desinformación al realizar consultas sobre procesos académicos, administrativos y servicios universitarios. El presente proyecto aborda esta problemática mediante la implementación de un prototipo de asistente conversacional denominado “UCAchat”.

La solución se basa en una arquitectura de Retrieval-Augmented Generation (RAG), que potencia un Large Language Model (LLM) con una base de conocimiento específica, construida a partir de documentos públicos de la universidad. La metodología propuesta comprende tres fases: una investigación inicial para recopilar documentos y determinar las necesidades de los estudiantes a través de encuestas; el diseño y desarrollo del prototipo RAG y su interfaz web; y una fase final de pruebas y evaluación con usuarios para medir su eficacia y pertinencia. Se espera que el prototipo demuestre la viabilidad de centralizar la información institucional y mejorar significativamente la experiencia estudiantil.

ÍNDICE

RESUMEN	i
ÍNDICE DE FIGURAS	v
ÍNDICE DE TABLAS	vii
SIGLAS	ix
ABREVIATURAS.....	xi
NOMENCLATURA	xiii
CAPÍTULO 1. INTRODUCCIÓN	1
1.1 Planteamiento del problema	1
1.2 Antecedentes.....	1
1.3 Alcances y limitaciones	2
1.3.1 Alcances.....	2
1.3.2 Limitaciones.....	2
1.4 Objetivos.....	3
1.4.1 Objetivo general	3
1.4.2 Objetivos específicos	3
CAPÍTULO 2. ESTRUCTURA DE LOS CAPÍTULOS	5
CAPÍTULO 3. MARCO TEÓRICO	7
3.1 Orígenes de la Inteligencia Artificial (IA)	7
3.2 Procesamiento de Lenguaje Natural (PLN)	8
3.2.1 Ventaja de Procesamiento de Lenguaje Natural	8
3.2.2 Desventaja de Procesamiento de Lenguaje Natura.....	8
3.3 ¿De dónde nace el chatbot?	8
3.4 Large Language Models (LLMs)	9
3.5 Alucinaciones en Large Language Models.....	9
3.5.1 Retrieval-Augmented Generation (RAG)	10
3.5.2 ¿Que es Fine-Tuning?.....	10
3.5.3 Graphics Processing Units (GPUs)	10
3.5.4 Retrieval-Augmented Generation o Fine-Tuning.....	11
3.6 ¿Qué son los Embeddings?	12
3.7 Bases de datos vectoriales	12
3.8 ¿Qué es Ollama?.....	13
3.9 Métricas de evaluación para sistemas Retrieval-Augmented Generation	14
3.10 Consideraciones éticas y limitaciones	15
CAPÍTULO 4. METODOLOGÍA	17
4.1 Metodología	17
4.1.1 Fase I: Investigación y recopilación de datos.....	17
4.1.2 Fase II: Diseño y desarrollo del prototipo.....	17

4.1.3 Fase III: Pruebas y evaluación	17
CAPÍTULO 5. PRESENTACIÓN, ANÁLISIS E INTERPRETACIÓN DE RESULTADOS	19
5.1 Año de carrera y facultad	19
5.1.1 Análisis de la muestra.....	19
CAPÍTULO 6. FORMATO DE LOS TRABAJOS	21
CAPÍTULO 7. CONCLUSIONES Y RECOMENDACIONES	25
7.1 Conclusiones.....	25
7.2 Recomendaciones	25
GLOSARIO	27
REFERENCIAS	29

ANEXOS

ANEXO A. anexos

ÍNDICE DE FIGURAS

Figura 4.1 Cronograma de actividades del proyecto.....	18
Figura 5.1 Año de carrera y Facultad del gráfico.....	19

ÍNDICE DE TABLAS

Tabla 5.1 Año de carrera de los encuestados	19
Tabla 5.2 Facultad de los encuestados.....	19

SIGLAS

Siglas: En esta sección se deben escribir las siglas que se utilizan en el trabajo. Una sigla es una palabra conformada por las iniciales de un nombre o expresión compuesta por varias palabras, por ejemplo ONU (Organización de las Naciones Unidas).

ABREVIATURAS

Abreviaturas: En esta sección se deben escribir las abreviaturas que se utilizan en el trabajo. Las abreviaturas son representaciones reducidas de una palabra, por ejemplo Sr.(Señor), no se debe abusar de ellas y no se deben colocar las unidades de medida.

NOMENCLATURA

Nomenclaturas: En esta sección se debe escribir la nomenclatura que se utiliza en el trabajo. La nomenclatura son los diferentes símbolos que se utilizan en un ámbito específico de las ciencias, por ejemplo en el área de química H_2O (formula del agua) o en física V (Voltaje).

CAPÍTULO 1. INTRODUCCIÓN

1.1 Planteamiento del problema

Los estudiantes de la Universidad Centroamericana José Simeón Cañas, y las personas interesadas en estudiar en esta universidad, deben realizar a lo largo de su carrera diferentes procesos, tanto administrativos como educativos, sociales y recreativos. La cantidad de información existente impide su centralización, por lo cual muchos estudiantes desconocen los lineamientos que deben seguir, los lugares a los que deben acercarse y, en general, dónde encontrar la información que necesitan.

Esto provoca que los estudiantes enfrenten dificultades para acceder de manera rápida y eficiente a datos relevantes como horarios de la cafetería, reglamentos institucionales, mallas curriculares, servicios disponibles dentro de la institución, así como opciones de alimentación cercanas al campus. La falta de un sistema centralizado de consulta genera pérdida de tiempo, desinformación y, en algunos casos, desmotivación en la comunidad universitaria.

Ante esta situación, surge la necesidad de contar con una herramienta digital que centre y organice la información más relevante en un solo lugar, de manera accesible, práctica y actualizada.

1.2 Antecedentes

Chatbot para la educación: un asistente conversacional sobre inteligencia artificial.

Los autores Gil, F., Moraes, A. y Tift, W. de la Universidad de la República (Uruguay) en su trabajo de graduación para optar por el título de Ingeniero en Computación.

Colibri, un chatbot educativo de código abierto diseñado para apoyar a docentes en la enseñanza y comprensión de la inteligencia artificial generativa. El sistema se entrenó con fuentes confiables, integrando funcionalidades que permiten responder preguntas, brindar ejemplos y recomendar lecturas sobre IA. Además, buscó fomentar un aprendizaje autónomo mediante una interfaz conversacional intuitiva. Este antecedente resulta relevante porque muestra cómo los chatbots pueden ser empleados en el ámbito educativo para transmitir conocimiento especializado de manera clara y accesible Gil et al., 2025.

Chatbot en ámbitos académicos. Geekoders, 2019

Los autores Geekoders de la Universidad de El Salvador, sede en Santa Tecla, en su proyecto de graduación para optar por el título de Ingeniero en Software.

Un asistente virtual conversacional accesible a través de Facebook Messenger. El chatbot fue pro-

gramado con 38 intenciones específicas para atender consultas frecuentes de los estudiantes, como inscripciones, horarios, constancias y procesos administrativos. Durante su primera semana de uso, mostró resultados positivos en la cantidad de interacciones realizadas por los alumnos. Este antecedente es importante porque ejemplifica la aplicación práctica de chatbots en contextos universitarios para mejorar la comunicación institucional y dar respuestas rápidas a dudas comunes Geekoders, 2019 .

Chatbot for communicating with university students in emergency situations.

Los autores Balderas, A., García-Mena, R. F., Huerta, M., Mora, N., & Dodero, J. M. de la Universidad de Cádiz en su trabajo de graduación por optar por el título de Ingeniero en Informática.

Un chatbot desarrollado para atender a estudiantes universitarios durante situaciones de emergencia, como la pandemia de COVID-19. El sistema fue diseñado con Dialogflow y entrenado para manejar preguntas frecuentes relacionadas con servicios de apoyo psicológico, trámites académicos y acceso a recursos institucionales. Fue evaluado con estudiantes, docentes y personal administrativo, obteniendo resultados positivos en facilidad de uso, rapidez y claridad en las respuestas Balderas et al., 2023.

1.3 Alcances y limitaciones

1.3.1 Alcances

- La aplicación será accesible a través de una página web, permitiendo a los estudiantes y demás usuarios consultar información y resolver dudas relacionadas con la universidad de manera centralizada y práctica.
- Contará con una interfaz de usuario intuitiva, simple y accesible, diseñada para facilitar la navegación y garantizar que los usuarios, independientemente de su nivel de experiencia tecnológica, puedan interactuar con el sistema sin dificultad.
- La aplicación ofrecerá respuestas rápidas y contextualizadas, reduciendo la necesidad de que los estudiantes acudan a múltiples fuentes de información dispersas.
- Se incluirá un diseño adaptativo (responsive), que permitirá el acceso tanto desde computadoras de escritorio como desde dispositivos móviles.

1.3.2 Limitaciones

La aplicación estará limitada a contestar preguntas relacionadas al contexto de la universidad y sus extensiones, se evitará en la medida de lo posible dar seguimiento a información que esté fuera de estos límites.

1.4 Objetivos

1.4.1 Objetivo general

Implementar un chat con integración de inteligencia artificial para proveer información de manera centralizada acerca de la Universidad Centroamericana José Simeón Cañas y sus procesos.

1.4.2 Objetivos específicos

- Identificar los procesos más comunes por los cuales los estudiantes realizan consultas de información, con el fin de recopilar y clasificar sus necesidades principales.
- Elaborar una base de conocimiento estructurada que contenga la información académica y administrativa más consultada por los estudiantes.
- Analizar diferentes alternativas de modelos de inteligencia artificial que permitan un equilibrio entre eficiencia y uso de recursos.
- Integrar la base de conocimiento con un modelo de inteligencia artificial que responda de manera clara y precisa a las consultas estudiantiles.

CAPÍTULO 2. ESTRUCTURA DE LOS CAPÍTULOS

Los capítulos del Trabajo de Graduación con el tema de Implementación de prototipo para UCachat están estructurados de la siguiente manera:

1. Marco teórico
2. Metodología
3. Presentación, análisis e interpretación de resultados

CAPÍTULO 3. MARCO TEÓRICO

3.1 Orígenes de la Inteligencia Artificial (IA)

La inteligencia artificial (IA) es una tecnología que ha experimentado un avance espectacular en poco tiempo, gracias a la combinación de factores como el *big data*, el *blockchain*, la nube, el internet de las cosas, la robótica y la realidad virtual. Aunque la IA no es una invención reciente, ya que sus orígenes se remontan a hace más de 50 años, su impacto actual es enorme y afecta a casi todos los ámbitos de la vida. Una conjunción de factores, como los avances en la potencia informática, la disponibilidad de enormes cantidades de datos y nuevos algoritmos, ha permitido que se produzcan grandes logros en los sistemas de IA en los últimos años.

No hay una definición clara de estos sistemas que goce de amplio consenso, porque, de una parte, la IA está sometida a las variaciones que se produzcan como consecuencia de los avances tecnológicos, de forma que no responde a algo estático, sino que es producto de una tecnología disruptiva que, además, se desarrolla a pasos acelerados. De otra parte, no existe un consenso entre los agentes implicados sobre lo que son los sistemas de IA. Sin embargo, con el objeto de regular la inteligencia artificial (IA) de manera efectiva, es necesaria una comprensión común de lo que se entiende por “inteligencia artificial”. Con ello se pretende dar respuesta al interrogante de qué es lo que se quiere regular y por qué, así como identificar cuáles son los aspectos que se consideran “peligrosos” y que deben ser objeto de regulación. No todos los sistemas de IA son motivo de preocupación, ya que no todos pueden afectar al régimen de derechos.

El término IA, *Inteligencia Artificial*, fue usado por primera vez por John McCarthy (1956) para referirse a “la ciencia y la ingeniería de crear máquinas inteligentes, especialmente programas de computación inteligentes”. Los sistemas de IA son capaces de adaptar su comportamiento, analizar los efectos de acciones previas y trabajar de manera autónoma. Son tecnologías de procesamiento de la información que integran modelos y algoritmos con capacidad para aprender y realizar tareas cognitivas, dando lugar a resultados como la predicción y la adopción de decisiones en entornos materiales y virtuales.

La inteligencia artificial se relaciona de forma clara con el *big data*. Lo necesita para desarrollar sus funcionalidades, ya que se nutre de la gran cantidad de datos recopilados para entrenar modelos de aprendizaje automático y tomar decisiones basadas en patrones y correlaciones. Esta sinergia permite a la inteligencia artificial realizar tareas como el procesamiento de lenguaje natural, la visión por computadora y la toma de decisiones predictivas con un alto grado de precisión Cotino Hueso, 2017. La tecnología *blockchain* también desempeña un papel importante en este ecosistema Merchán Murillo, 2019. Al aprovechar la seguridad y la inmutabilidad que proporciona la cadena de bloques, tanto el *big data* como la inteligencia artificial pueden garantizar la integridad de los datos, lo que es

esencial en aplicaciones críticas como la autenticación de identidades digitales y la gestión de claves criptográficas, fortaleciendo así la seguridad en las comunicaciones y transacciones digitales.

3.2 Procesamiento de Lenguaje Natural (PLN)

Una de las tareas fundamentales de la Inteligencia Artificial (IA) es la manipulación de lenguajes naturales usando herramientas de computación, en esta, los lenguajes de programación juegan un papel importante, ya que forman el enlace necesario entre los lenguajes naturales y su manipulación por una máquina. El PLN consiste en utilizar un lenguaje natural para comunicarnos con la computadora, debiendo ésta entender las oraciones que le sean proporcionadas, el uso de estos lenguajes naturales facilita el desarrollo de programas que realicen tareas relacionadas con el lenguaje o bien, desarrollar modelos que ayuden a comprender los mecanismos humanos relacionados con el lenguaje.

El uso del lenguaje natural (LN) en la comunicación hombre-máquina presenta a la vez una ventaja y un obstáculo con respecto a otros medios de comunicación.

3.2.1 Ventaja de Procesamiento de Lenguaje Natural

Vásquez, Quispe y Huayna (2009) mencionan que, por un lado, es una ventaja, en la medida en que el locutor no tiene que esforzarse para aprender el medio de comunicación a diferencia de otros medios de interacción como lo son los lenguajes de comando o las interfaces gráficas.

3.2.2 Desventaja de Procesamiento de Lenguaje Natural

Su uso también presenta limitaciones porque la computadora tiene una limitada comprensión del lenguaje. Por ejemplo, el usuario no puede hablar sobreentendidos, ni introducir nuevas palabras, ni construir sentidos derivados, tareas que se realizan espontánea mente cuando se utiliza el lenguaje natural. Realmente, lo que constituye en ventaja para la comunicación humana se convierte en problema a la hora de un tratamiento computacional, ya que implican conocimiento y procesos de razonamiento que aún no sabemos ni cómo caracterizarlos ni cómo formalizarlos

3.3 ¿De dónde nace el chatbot?

El primer sistema informático de procesamiento de lenguaje natural fue creado en el Instituto de Tecnología de Massachusetts (MIT por sus siglas en inglés) por Joseph Weizenbaum en 1966. El nombre de este bot conversacional fue ELIZA y su finalidad era simular a una psicoterapeuta (Khan & Das, 2018). Desde aquel entonces se empezaron a crear proyectos con funcionalidades similares. En el año 2011, también se marca trascendencia con el lanzamiento de Siri, el asistente de Apple, el cual ayuda a realizar tareas personales para los usuarios del sistema operativo iOS mediante el

uso de una interfaz de lenguaje natural controlada por voz (Reehal, 2016). Los chatbots son un tipo de asistente virtual, generalmente entrenados para fines específicos, es decir, la información o tareas que realizan se encuentran delimitadas por la entidad que lo entrena y eso es lo que marca la diferencia con los asistentes virtuales de tipo personal como lo son Siri de Apple, Google Assistant de Google, Alexa de Amazon, Cortana de Windows, entre otros. Según (Galitsky, 2019), dentro de su libro *Developing Enterprise Chatbots: Learning Linguistic Structures* describe que un chatbot es “un sistema informático que funciona como una interfaz entre los usuarios humanos y las aplicaciones de software, utilizando el lenguaje natural hablado o escrito como medio principal de comunicación”. Debido al auge de la Inteligencia Artificial en nuestro tiempo, existen decenas de proveedores de la nube que dentro de los servicios de IA que poseen, proporcionan plataformas que permiten crear interfaces conversacionales.

3.4 Large Language Models (LLMs)

Large Language Models o LLMs, son sistemas de inteligencia artificial entrenados con vastas cantidades de datos de texto para comprender, generar y manipular el lenguaje humano. Estos modelos, basados en arquitecturas de redes neuronales como los Transformers (Rawte, Sheth & Das, 2023), que utilizan mecanismos de atención para procesar y relacionar palabras dentro de un contexto, permitiendo al modelo capturar dependencias a largo plazo en el texto. Gracias a esta arquitectura, los LLMs han demostrado una capacidad sobresaliente para realizar una amplia gama de tareas. Sin embargo, su conocimiento es inherentemente generalista y limitado a la información contenida en sus datos de entrenamiento. Al enfrentarse a consultas altamente especializadas o que requieren información muy reciente, los LLMs pueden generar respuestas plausibles pero incorrectas, un fenómeno conocido como ”alucinación” (Ji et al., 2023). Es por ello que, para aplicaciones en dominios específicos como el universitario, se requieren técnicas que anclen sus capacidades a una base de conocimiento controlada.

3.5 Alucinaciones en Large Language Models

Large Language Models (LLMs), a pesar de sus capacidades para procesar y generar lenguaje natural, presentan una desventaja coloquialmente conocida como “alucinaciones”. Este término hace referencia a la tendencia de los modelos a generar información que no es correcta, sin sentido, o que no tiene fundamentos en datos relacionados al entrenamiento, pero que es presentada con un alto grado de confianza y fluidez (Ji et al., 2023; Rawte, Sheth & Das, 2023).

Las alucinaciones no implican un error de programación, sino una consecuencia directa de la arquitectura y objetivo de los LLMs. Estos modelos son sistemas probabilísticos diseñados para predecir la siguiente cadena más probable de una secuencia. Al enfrentarse a una pregunta cuya respuesta no está claramente representada en sus datos de entrenamiento, el modelo puede “inventar” una respuesta

que es estadísticamente plausible en su estructura lingüística, pero factualmente incorrecta (Ji et al., 2023; Rawte, Sheth & Das, 2023).

Acorde a Ji et al. (2023), existen dos categorías principales de alucinaciones: la alucinación intrínseca, donde el modelo contradice el conocimiento presente y la fuente, y la alucinación extrínseca, donde genera información que no puede ser verificada a partir de la fuente.

3.5.1 Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) es una arquitectura avanzada de inteligencia artificial diseñada para superar las limitaciones de los LLMs. Propuesta por Lewis et al. (2020), la técnica RAG enriquece el proceso de generación de respuesta de un LLM al permitirle acceder a una base de conocimiento externa en tiempo real. El proceso funciona en dos etapas principales:

1. Recuperación: Ante una pregunta del usuario, el sistema primero busca y recupera los fragmentos de información más relevantes de una base de datos documental previamente establecida.
2. Generación: A continuación, esta información recuperada se entrega al LLM junto con la pregunta original. El LLM utiliza este contexto adicional y fáctico para generar una respuesta precisa, coherente y fundamentada en los documentos de la fuente (Databricks, 2023; LlamaIndex, 2024).

3.5.2 ¿Qué es Fine-Tuning?

El Fine-Tuning es una técnica fundamental en el aprendizaje transferencial para modelos de lenguaje. Consiste en tomar un modelo preentrenado genérico (que ya ha aprendido patrones lingüísticos generales de un vasto corpus de texto) y especializarlo para una tarea, dominio o estilo particular. Esto se logra mediante un ciclo adicional de entrenamiento, pero utilizando un conjunto de datos mucho más pequeño, específico y etiquetado para el nuevo objetivo. Durante este proceso, se ajustan ligeramente los pesos (parámetros) internos de la red neuronal, lo que permite que el modelo "internalice" el nuevo conocimiento y se adapte a contextos específicos, como jerga técnica, formatos de respuesta particulares o información corporativa privada (Howard & Ruder, 2018). Aunque es muy potente para la especialización, este método crea una versión estática del modelo que no puede actualizar su conocimiento sin un costoso proceso de re-entrenamiento.

3.5.3 Graphics Processing Units (GPUs)

Graphics processing units (GPUs) alimentan las supercomputadoras más rápidas de la actualidad, son la plataforma dominante para el aprendizaje profundo y proporcionan la inteligencia para dispositivos que van desde automóviles autónomos hasta robots y cámaras inteligentes. También generan

imágenes fotorrealistas a velocidades de fotogramas en tiempo real. Las GPUs han evolucionado agregando funciones para admitir nuevos casos de uso. La combinación de programabilidad y rendimiento de punto flotante hizo que las GPUs fueran atractivas para ejecutar aplicaciones científicas. Los científicos encontraron formas de usar las primeras GPUs programables al convertir sus cálculos como sombreadores de vértices y fragmentos. Las GPUs evolucionaron para satisfacer las necesidades de los usuarios científicos al agregar hardware para una programación más simple, aritmética de punto flotante de doble precisión y resistencia.

3.5.4 Retrieval-Augmented Generation o Fine-Tuning

Para adaptar un LLM a un dominio de conocimiento determinado, existen dos enfoques principales: el Ajuste Fino (Fine-Tuning) y la Generación Aumentada por Recuperación (Retrieval-Augmented Generation). Aunque ambos buscan mejorar el rendimiento del modelo, operan de maneras distintas.

El Fine-Tuning es un proceso que, partiendo de un LLM preentrenado, continúa su entrenamiento con un conjunto de datos más pequeño y específico (Radford et al., 2018). Este proceso, popularizado por investigadores de OpenAI y otros laboratorios, ajusta los pesos internos (parámetros) del modelo para especializarlo en el estilo, tono y contenido de los nuevos datos. El resultado es un modelo que "conoce" la información de forma inherente. Sin embargo, este enfoque presenta desventajas significativas: es computacionalmente costoso, requiere grandes cantidades de datos de alta calidad en formato de pregunta-respuesta, y si la información cambia (por ejemplo, una nueva normativa académica), el modelo debe ser re-entrenado por completo (Databricks, 2023).

Por otro lado, RAG no modifica los pesos del LLM. En su lugar, conecta el modelo a una base de conocimiento externa (Lewis et al., 2020). Cuando llega una consulta, RAG primero recupera información relevante de esta base de datos y luego la proporciona al LLM como contexto para generar la respuesta. Esta aproximación, desarrollada inicialmente por investigadores de Facebook AI Research (FAIR), ofrece ventajas clave para este caso de uso (Lewis et al., 2020):

1. Actualización sencilla: Para actualizar el conocimiento, basta con añadir, modificar o eliminar documentos de la base de datos, sin necesidad de re-entrenar el modelo.
2. Transparencia y verificabilidad: Las respuestas están basadas directamente en los documentos recuperados, lo que permite citar las fuentes y reduce drásticamente las alucinaciones (Gao et al., 2023).
3. Eficiencia de costos: Es mucho menos costoso que el Fine-Tuning, ya que no requiere un entrenamiento intensivo con GPUs (Pinecone, 2023).

En resumen, mientras que el fine-tuning enseña al modelo "cómo hablar" sobre un tema, RAG le da al modelo "algo sobre qué leer" antes de hablar. Para un sistema que debe garantizar la precisión y

mantenerse actualizado con información factual, RAG es la arquitectura superior (Pinecone, 2023).

3.6 ¿Qué son los Embeddings?

La fase de "recuperación" en la arquitectura RAG tiene paso gracias a una técnica de PLN llamada Embeddings (incrustaciones o vectores de palabras). Un embedding es una representación numérica de un texto (ya sea una palabra, una oración o un documento completo) en forma de un vector de números de punto flotante en un espacio multidimensional (TensorFlow, s.f.).

El objetivo de los modelos de embeddings es capturar el significado semántico del texto. Esto significa que dos fragmentos de texto con significados parecidos, aunque usen palabras diferentes, tendrán vectores numéricos muy cercanos entre sí en ese espacio multidimensional (Mikolov et al., 2013).

En el contexto de RAG, el proceso funciona de la siguiente manera:

1. **Indexación:** Todos los documentos de la base de conocimiento (reglamentos, catálogos de materias, etc.) se dividen en fragmentos más pequeños (chunks). Cada chunk se pasa a través de un modelo de embeddings para generar un vector que captura su significado.
2. **Almacenamiento:** Estos vectores se almacenan en una base de datos especializada, conocida como base de datos vectorial.
3. **Búsqueda:** Cuando un usuario hace una pregunta, esa pregunta también se convierte en un vector utilizando el mismo modelo de embeddings. El sistema luego busca en la base de datos vectorial los vectores de los chunks que son más "cercanos" o "similares" al vector de la pregunta.

Esta búsqueda por similitud semántica, en lugar de por palabras clave, es lo que permite a RAG encontrar los fragmentos de información más relevantes para responder a la consulta del usuario, incluso si la pregunta no utiliza los mismos términos exactos que el documento original.

3.7 Bases de datos vectoriales

Una vez que los documentos de la base de conocimiento han sido convertidos en embeddings, es necesario un sistema para su almacenamiento y búsquedas de similitud de manera eficiente. Las bases de datos tradicionales, diseñadas para consultas estructuradas (SQL) o de texto simple, no son eficientes para esta tarea.

Una base de datos vectorial es un sistema de almacenamiento optimizado para guardar y consultar datos de alto alcance, como los embeddings. Su función principal es realizar búsquedas de vecinos más cercanos (Approximate Nearest Neighbor - ANN) a una velocidad extremadamente alta. En

lugar de comparar el vector de una consulta con cada uno de los millones de vectores almacenados (lo que sería computacionalmente inviable), utilizan algoritmos de indexación especializados, como Hierarchical Navigable Small World (HNSW). Hierarchical Navigable Small World es un algoritmo que organiza los vectores en una estructura de grafos jerárquica de múltiples capas. La búsqueda comienza en las capas superiores, que son más escasas y permiten navegación rápida para localizar la región aproximada donde se encuentran los vecinos más cercanos. Luego, el proceso desciende progresivamente a capas más densas hasta refinar la búsqueda en la capa inferior, que contiene todos los vectores. Esta arquitectura permite encontrar los vectores más similares de forma casi instantánea

Una base de datos vectorial es el componente que alberga el "cerebro" de conocimiento del sistema. Permite que, ante una pregunta de un estudiante, el sistema pueda:

1. Recibir el vector de la pregunta.
2. Consultar el índice de millones de vectores de documentos.
3. Devolver en milisegundos los K fragmentos de texto más relevantes.

Estos fragmentos recuperados son los que se inyectan en el contexto del LLM para que genere una respuesta precisa y fundamentada. Ejemplos de bases de datos vectoriales populares incluyen ChromaDB de código abierto, FAISS de Facebook AI, y servicios en la nube como Pinecone o Weaviate.

3.8 ¿Qué es Ollama?

La operatividad de los Large Language Models dentro de una arquitectura de software requiere un framework que gestione su ciclo de vida, desde la carga en memoria hasta la inferencia. En este contexto, surgen dos paradigmas de despliegue: el modelo como servicio (Model-as-a-Service, MaaS) a través de APIs de terceros y el despliegue local auto alojado (self-hosting). Para aplicaciones que manejan información sensible el enfoque de auto alojamiento es teóricamente superior (García, 2023).

Frameworks como Ollama representan un componente clave en la materialización de este paradigma. Su función es abstraer la complejidad asociada con la ejecución de modelos de código abierto, proporcionando un entorno de ejecución estandarizado y una interfaz de programación de aplicaciones (API) local (Ollama Documentation, 2024). La adopción de este enfoque ofrece ventajas fundamentales:

1. **Soberanía de datos:** al procesar las consultas íntegramente dentro de una infraestructura controlada, se garantiza que la información sensible nunca abandone el perímetro de seguridad de la institución. Esto es fundamental para el cumplimiento de normativas de protección de datos y para mantener la confidencialidad de la información (Ollama Documentation, 2024).
2. **Reducción de latencia:** la comunicación entre los componentes del sistema se realiza a través de la red local, eliminando la latencia inherente a las llamadas de API a través de internet. Esto

se traduce en tiempos de respuesta más rápidos y una mejor experiencia de usuario (Smith et al., 2023).

3. **Independencia y continuidad operativa:** el sistema se desvincula de la disponibilidad, políticas de uso y posibles cambios en los modelos o costos de proveedores externos. Esto asegura la continuidad del servicio y un control total sobre las versiones del modelo y su configuración (García, 2023).
4. **Optimización de costos a escala:** se transita de un modelo de costo operativo variable, basado en el consumo por token o por llamada, a un modelo de inversión en infraestructura (capital fijo). Para aplicaciones con un alto volumen de consultas, este enfoque puede resultar significativamente más económico a largo plazo (Chen & Patel, 2024).

Por lo tanto, la selección de un framework para el despliegue local no es meramente una decisión técnica, sino una elección arquitectónica estratégica que prioriza la seguridad, el rendimiento y la autonomía sobre la conveniencia de los servicios gestionados por terceros.

3.9 Métricas de evaluación para sistemas Retrieval-Augmented Generation

La evaluación de un sistema de Retrieval-Augmented Generation (RAG) es una tarea compleja que va más allá de medir la simple corrección de una respuesta. Si el sistema consta de dos componentes principales, el recuperador (retriever) y el generador (generator), una evaluación robusta debe analizar tanto el rendimiento de las partes como la calidad del resultado final (Schockaert et al., 2023).

El objetivo es cuantificar la capacidad del sistema para proporcionar respuestas que además de correctas, sean también fieles al contexto y relevantes para la pregunta del usuario.

Schockaert et al. (2023) han definido un conjunto de métricas especializadas, a menudo agrupadas en frameworks como RAGAS (Rag Assessment). Estas métricas clave incluyen:

1. Evaluación del módulo de recuperación:

- **Precisión del contexto (Context precision):** mide la relación señal-ruido del contexto recuperado. Evalúa si los fragmentos de texto obtenidos de la base de conocimiento son verdaderamente relevantes para la pregunta. Una baja precisión indica que el recuperador está introduciendo "ruido" que podría confundir al generador.
- **Recuperación del contexto (Context recall):** evalúa si el recuperador ha sido capaz de encontrar la información necesaria contenida en la base de datos para responder a la pregunta de manera exhaustiva.

2. Evaluación del módulo de generación:

- **Fidelidad (Faithfulness):** esta métrica es crucial para medir la mitigación de alucinaciones. Se evalúa en qué medida la respuesta generada se basa estrictamente en el contexto

proporcionado. Una respuesta con alta fidelidad no contiene información que contradiga o no esté presente en los fragmentos recuperados.

- **Relevancia de la respuesta (Answer relevancy):** mide cuán pertinente es la respuesta generada con respecto a la pregunta original del usuario. Una respuesta puede ser factualmente correcta y fiel al contexto, pero no abordar directamente la duda del usuario, lo que resultaría en una baja puntuación de relevancia.

Además de estas métricas cuantitativas, la evaluación cualitativa a través de encuestas de satisfacción del usuario es fundamental para medir el éxito global del chatbot en su contexto de aplicación real, ya que captura aspectos de la experiencia de usuario que las métricas automáticas no pueden medir.

3.10 Consideraciones éticas y limitaciones

La implementación de un sistema de inteligencia artificial en un entorno educativo, como un chatbot para consultas estudiantiles, conlleva responsabilidades éticas significativas y el reconocimiento de sus limitaciones inherentes, un principio ampliamente respaldado por marcos globales como el libro ChatGPT and Artificial Intelligence in higher education Quick start guide de la UNESCO (2023). Un análisis proactivo de estos factores es indispensable para un despliegue responsable.

1. **Sesgos en los datos y algoritmos:** la principal limitación del sistema es que su conocimiento se deriva exclusivamente de la base documental con la que es alimentado. Si estos documentos contienen información desactualizada, sesgos institucionales históricos o errores, el chatbot los perpetuará y presentará como hechos. Adicionalmente, el LLM subyacente, entrenado con datos masivos de internet, puede albergar sesgos sociales (de género, culturales, etc.) que podrían manifestarse sutilmente en el tono o la estructura de las respuestas, a pesar de las restricciones impuestas por RAG (Bender et al., 2021).
2. **Transparencia y explicabilidad:** es éticamente necesario que los usuarios sean conscientes en todo momento de que están interactuando con una IA y no con un ser humano. El sistema debe ser transparente sobre sus capacidades y limitaciones. La arquitectura RAG facilita parcialmente la explicabilidad al permitir la cita de las fuentes documentales, otorgando al usuario una vía para verificar la información.
3. **Privacidad de los datos del usuario:** aunque se opte por una arquitectura de despliegue local para proteger la soberanía de los datos, las consultas de los estudiantes pueden contener información personal o sensible. Se deben establecer políticas claras sobre el almacenamiento, anonimización y retención de los registros de conversaciones para proteger la privacidad del estudiantado y cumplir con la legislación de protección de datos personales.
4. **Margen de error y exceso de confianza:** existe el riesgo de que los usuarios desarrollen una confianza excesiva en las respuestas del chatbot, tratándolo como una fuente de verdad infalible. El sistema debe diseñarse para incluir descargos de responsabilidad claros, indicando que es una

herramienta de asistencia y que la información crítica debe ser confirmada con las autoridades académicas humanas correspondientes. Se debe proveer un canal de comunicación directo con personal administrativo para casos complejos o no contemplados en la base de conocimiento.

Reconocer estas limitaciones no disminuye el valor del proyecto, sino que enmarca su utilidad como un poderoso asistente de primer nivel, diseñado para complementar, y no para reemplazar, el juicio y la asistencia humana en el entorno universitario.

CAPÍTULO 4. METODOLOGÍA

4.1 Metodología

El desarrollo del prototipo “UCAchat” se llevará a cabo mediante una metodología estructurada en tres fases secuenciales, diseñadas para garantizar que el producto final responda a las necesidades reales de los usuarios y cumpla con los objetivos técnicos del proyecto.

4.1.1 Fase I: Investigación y recopilación de datos

Esta fase inicial se centra en comprender el problema y reunir los insumos necesarios para el sistema.

1. **Levantamiento de requerimientos del usuario:** Se diseñará y distribuirá una encuesta dirigida a la población estudiantil para identificar las dudas más frecuentes y las expectativas sobre un asistente virtual.
2. **Recopilación de información:** Se recolectarán documentos públicos y relevantes de la UCA (reglamento académico, guías de inscripción, mallas curriculares, etc.) que conformarán la base de conocimiento del sistema.

4.1.2 Fase II: Diseño y desarrollo del prototipo

En esta fase se construirán los componentes técnicos de la solución.

1. **Construcción de la base de conocimiento:** Los documentos recopilados serán procesados y estructurados para su uso en el sistema RAG.
2. **Implementación del sistema RAG:** Se seleccionará un LLM de base y se desarrollará la lógica de recuperación para conectarlo con la base de conocimiento, estará expuesto a través de un backend con arquitectura REST.
3. **Desarrollo de la interfaz de usuario (UI):** Se diseñará y programará una aplicación web que sirva como interfaz para el chatbot, con un diseño intuitivo y adaptativo.

4.1.3 Fase III: Pruebas y evaluación

La fase final consiste en validar el prototipo y recopilar retroalimentación para futuras mejoras.

1. **Prueba piloto:** Se seleccionará un grupo representativo de estudiantes para que interactúen con el prototipo “UCAchat”, en formato de focus group.
2. **Recopilación y análisis de retroalimentación:** Se recopilarán datos cuantitativos y cualitativos a través de observaciones y encuestas post-prueba.

3. **Informe de resultados:** Se analizarán los datos para evaluar el desempeño del prototipo, identificar y realizar posibles mejoras.

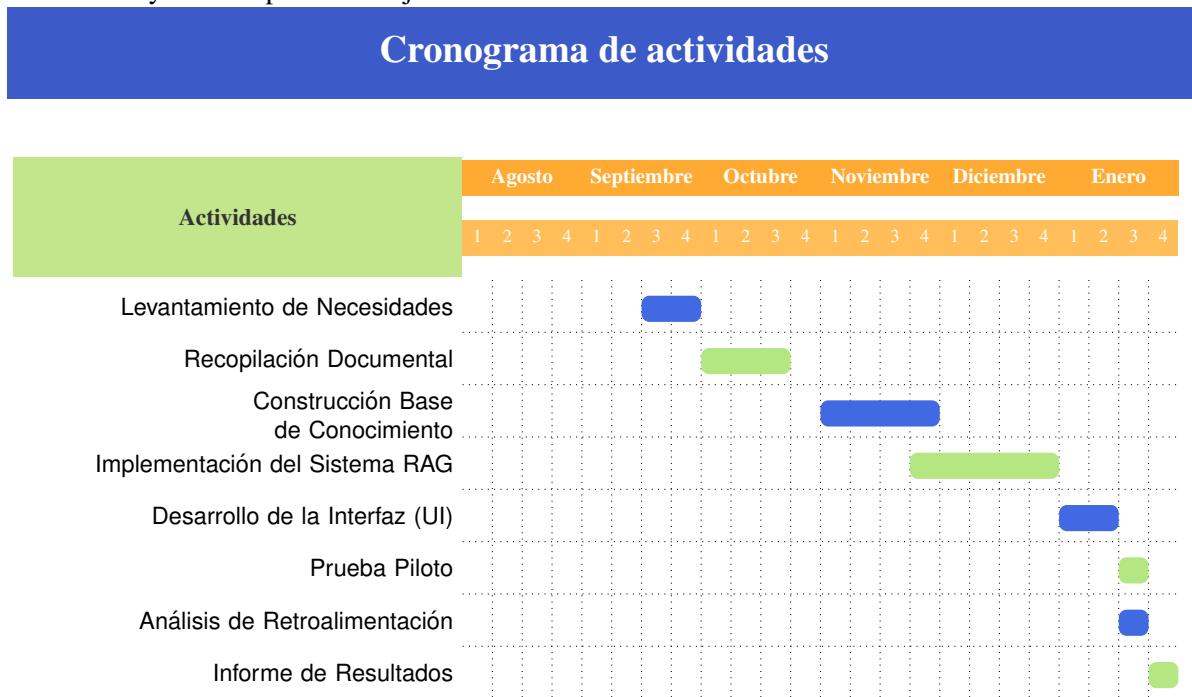


Figura 4.1 Cronograma de actividades del proyecto.

CAPÍTULO 5. PRESENTACIÓN, ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

En esta sección se presentan los resultados obtenidos de la encuesta realizada a estudiantes de 1er año a 5to año de Ingeniería y Arquitectura. Este capítulo proporciona los datos necesarios para guiar el diseño de UCACChat.

5.1 Año de carrera y facultad

Las primeras dos preguntas buscaban identificar en qué año de la carrera se encuentran los encuestados y la facultad a la que pertenecen.

Tipo	Año	Cantidad
Opción Cerrada	Primero	31
	Segundo	9
	Tercero	12
	Cuarto	18
	Quinto	17
Total		87

Tabla 5.1 Año de carrera de los encuestados

Tipo	Facultad	Cantidad
Opción Cerrada	Ingeniería y Arquitectura	87
	Ciencias Sociales y Humanidades	0
	Ciencias Económicas y Empresariales	0
Total		87

Tabla 5.2 Facultad de los encuestados

5.1.1 Análisis de la muestra

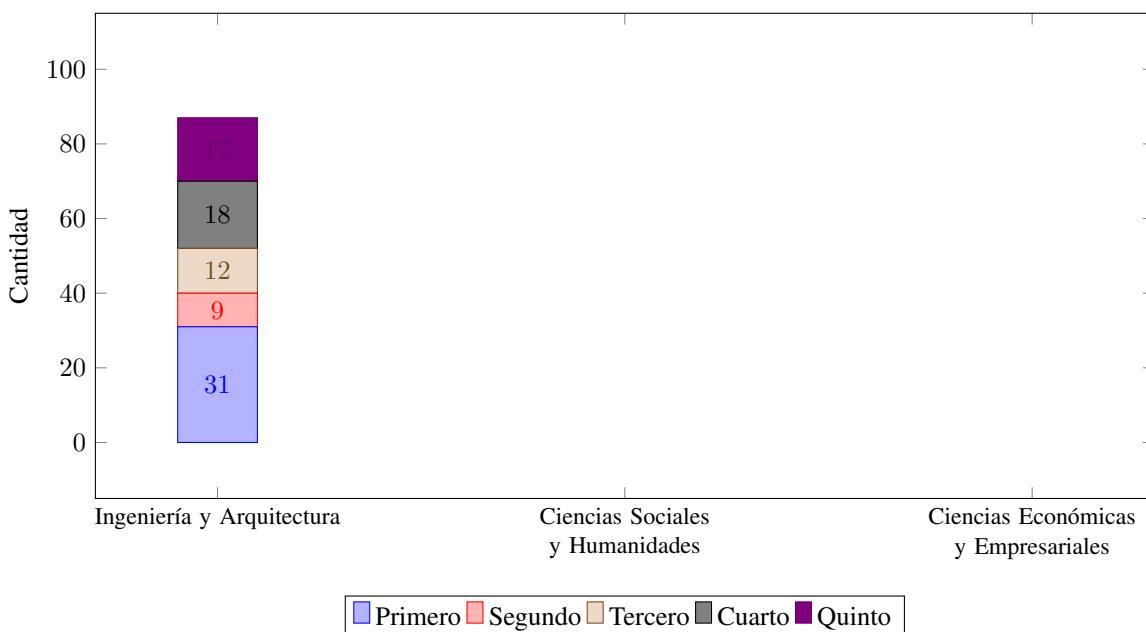


Figura 5.1 Año de carrera y Facultad del gráfico

La composición de la muestra se alineo bien con el objetivo del estudio, ya que la mayor parte de los que respondieron la encuesta fueron estudiantes de primer año (35.63%), todos pertenecientes a la Facultad de Ingeniería y Arquitectura (100%). Queremos que UCACchat está especialmente orientado a apoyar a estudiantes de nuevo ingreso, quienes enfrentan mayor incertidumbre respecto a procesos institucionales clave como retiro de materias, diferidos, inscripción y ubicación de información académica. Dado que estos estudiantes son quienes más requieren orientación en su transición universitaria, la muestra resultante es coherente y adecuada para los fines de la investigación, garantizando que las conclusiones reflejen las necesidades de los estudiantes que más se beneficiará con la implementación de UCACchat.

CAPÍTULO 6. FORMATO DE LOS TRABAJOS

A continuación, se muestran las características principales del formato de trabajos de graduación creado por la facultad de ingeniería y arquitectura de la UCA.

- **Partes del documento:**

- Primera portada.
- Segunda portada.
- Agradecimientos (opcional).
- Dedicatorias (opcional).
- Resumen.
- Índice.
- Índice de figuras.
- Índice de tablas.
- Siglas.
- Abreviaturas.
- Nomenclatura.
- Capítulos.
- Glosario (opcional).
- Referencias.
- Anexos.

- **Formato general:**

- Pagina tamaño carta.
- Margen de 2.5cm en todos lados.
- Borde de encuadernación de 1cm.
- Fuente Times New Roman 11pt.
- Interlineado 1.5pt.
- Espacio extra entre párrafos 0.
- Línea vacía entre párrafos.
- Todas las secciones deben comenzar en página impar.
- En todas las secciones que llevan título, el título se coloca al inicio de la página, en negrita, en mayúscula y centrado horizontalmente.
- Debe haber separación de una línea vacía entre el título y el texto.

- **Portadas:**

- Texto centrado horizontalmente.
- Espacio igual entre los párrafos para que el texto ocupe todo el espacio disponible.
- Texto en mayúscula.
- Fuente tamaño 14pt.
- Logo de la primera portada 2.5cm de alto.

- En la primera portada los nombres de los integrantes van ordenados en orden alfabético basados en el primer apellido.

- **Agradecimientos:**

- Extensión máxima una página.

- **Dedicatoria:**

- Una por cada integrante.
 - Extensión máxima una página.
 - Debe llevar el nombre correspondiente del integrante al final de la página y alineado al lado derecho.

- **Resumen:**

- Extensión de una a tres páginas.

- **Índice:**

- El espacio horizontal entre el nombre de las secciones y el número de página debe estar lleno de puntos.
 - Se permite hasta tres niveles de título.
 - Los títulos de las secciones principales van en mayúscula.
 - Al final se deben colocarlos anexos.

- **Índice de figuras e índice de tablas:**

- Cada entrada del índice de figuras debe iniciar con la palabra “Figura” seguido del número de capítulo y el número correlativo de la figura.
 - Cada entrada del índice de tablas debe iniciar con la palabra “Tabla” seguido del número de capítulo y el número correlativo de la tabla.

- **Siglas, abreviaturas y nomenclatura:**

- Van separados en dos columnas, en la columna izquierda se coloca la sigla o palabra seguido de dos puntos y en la columna derecha se coloca la definición.

- **Figuras:**

- Las figuras incluyen gráficos, diagramas, fotos, etc.
 - El epígrafe va en la parte inferior centrado horizontalmente.
 - Al inicio va la palabra “Figura” seguido del número de capítulo y número correlativo.
 - En el epígrafe continuo a la descripción de la figura va la fuente.
 - La fuente sigue el siguiente formato, “Fuente: [apellido autor, año]”.
 - Si la figura es de elaboración propia se coloca, “Fuente: [Elaboración propia]”.
 - Si la figura ha sido adaptada de alguna parte se coloca, “Adaptado de: [apellido autor, año]”.
 - Las figuras se pueden colocar con orientación horizontal, en ese caso siempre de izquierda a derecha.
 - Si la figura tiene orientación horizontal, el epígrafe se coloca en el lado derecho de la página.

- **Tabla:**

- El epígrafe va en la parte superior, centrado horizontalmente.
 - Al inicio va la palabra “Tabla” seguido del número de capítulo y número correlativo.
 - La fuente se coloca en la parte inferior, centrada horizontalmente.
 - La fuente sigue el siguiente formato, “Fuente: [apellido autor, año]”.
 - Si la tabla es de elaboración propia se coloca, “Fuente: [Elaboración propia]”.
 - Si la figura ha sido adaptada de alguna parte se coloca, “Adaptado de: [apellido autor, año]”.
 - Las tablas se pueden colocar con orientación horizontal, en ese caso siempre de izquierda a derecha.
 - Si la tabla tiene orientación horizontal, el epígrafe se coloca en el lado izquierdo de la página.
- **Ecuaciones:**
 - Las ecuaciones deben estar numeradas de la siguiente forma “(Ec.1.1)”.
 - **Glosario:**
 - Tiene el mismo formato que las siglas, abreviaturas y nomenclatura.
 - **Referencia:**
 - El formato de la referencia es en formato APA.
 - No se coloca sangría en las líneas de cada una de las referencias.
 - Las referencias van separadas horizontalmente por una línea vacía.
 - **Portada anexos:**
 - Los anexos se enumeran utilizando letras.
 - La portada de los anexos lleva dos partes, lleva la palabra “ANEXO” en mayúscula seguido de la letra correspondiente, el tamaño de fuente de esta parte es 20pt.
 - El título del anexo en mayúscula, el tamaño de fuente de esta parte es 16pt.
 - Todo el texto va centrado vertical y horizontalmente.

CAPÍTULO 7. CONCLUSIONES Y RECOMENDACIONES

7.1 Conclusiones

En esta sección se escriben las conclusiones a las que se ha llegado luego de realizar el trabajo, las conclusiones deben concordar con los objetivos planteados anteriormente.

7.2 Recomendaciones

En esta sección, se presentan las recomendaciones hechas por el grupo de trabajo para las personas que se beneficiarán del estudio o proyecto realizado.

GLOSARIO

Glosario: En esta sección, se incluyen términos utilizados en el documento cuya definición podría no ser conocida por el lector, especialmente aquellos de carácter técnico. Esta sección es opcional, como por ejemplo: BackEnd (Parte de una aplicación web que gestiona la lógica, el procesamiento de datos y la comunicación con el servidor).

REFERENCIAS

- Balderas, A., García-Mena, R. F., Huerta, M., Mora, N., & Dodero, J. M. (2023). Chatbot for communicating with university students in emergency situations. *Heliyon*, (9), 9.
- Cotino Hueso, L. (2017). Inteligencia artificial y derechos fundamentales: riesgos y oportunidades. *Revista General de Derecho Administrativo*, (46). https://www.iustel.com/v2/revistas/detalle_revista.asp?id_noticia=418061
- Geekoders. (2019). *Chatbot en ámbitos académicos*. https://geekoders.com/chatbot-en-ambitos-academicos/?utm_source=chatgpt.com
- Gil, F., Moraes, A., & Tift, W. (2025). *Chatbot para la educación: un asistente conversacional sobre inteligencia artificial* [Tesis de grado]. Universidad de la República, Uruguay.
- Merchán Murillo, A. (2019). Blockchain y la inteligencia artificial: sinergias y desafíos. *Revista Colombiana de Tecnologías de la Información*, 21(2), 45-59.

ANEXO A
ANEXOS

En los anexos se colocan los documentos, gráficos, tablas, imágenes u otros materiales complementarios que proporcionan información adicional relevante, pero que no forman parte del cuerpo principal del trabajo.