



**Desarrollo de un asesor normativo mediante inteligencia artificial para el
reglamento estudiantil del Politécnico Grancolombiano**

Autor:

Jeison Steven Angel Castro

Director:

Gabriel Eduardo Ávila Buitrago

Ingeniería de Sistemas
Facultad de Ingeniería, Diseño e Innovación
Institución Universitaria Politécnico Grancolombiano
Bogotá D.C., Colombia
2024

Agradecimientos

A mi madre, padre y hermano, quiero agradecerles su apoyo incondicional y confianza en mí. Su amor y comprensión me dieron la fuerza para superar los momentos difíciles y seguir adelante en este camino. Gracias por estar siempre a mi lado y por creer en mí.

Agradezco profundamente a mi director de proyecto de grado, por su invaluable acompañamiento, apoyo y consejos durante todo el proceso de investigación y elaboración de esta tesis. Su conocimiento, experiencia y dedicación fueron fundamentales para el desarrollo de este trabajo.

Por último, no puedo dejar de agradecerme a mí mismo por mantener la fortaleza frente a las adversidades y por siempre encontrar la motivación necesaria para seguir mejorando y aprendiendo de la vida. Este trabajo es un reflejo de mi esfuerzo y dedicación.

A todas las personas que ayudaron y participaron de este proceso, infinitas gracias.

Índice

Resumen del Proyecto:	6
1. TÍTULO DE LA PROPUESTA:	7
2. INTRODUCCIÓN.....	8
3. PLANTEAMIENTO DEL PROBLEMA:	10
4. OBJETIVOS.....	11
Objetivos específicos	11
5. JUSTIFICACIÓN.....	12
6. ALCANCE.....	13
7. MARCO TEÓRICO	14
Marco Teórico	14
Estado del Arte:.....	16
8. METODOLOGÍA.....	22
Análisis preliminar	22
Árbol de problemas	22
Encuestas.....	23
Definición de tecnologías.....	24
Actividades y cronograma	25
Diseño de la solución.....	27
Requerimientos	27
Casos de uso	28
Arquitectura.....	28
Base de datos.....	29
Diseño de interfaz	31
Desarrollo de la solución	31
Prototipado y experimentación.....	31
Entrenamiento e implementación de modelo	37
Desarrollo de módulo de procesamiento de texto	40
Desarrollo de backend	41
Desarrollo de frontend	42
Pruebas y finalización.....	45
9. RESULTADOS	46
Conclusiones	48
Trabajo futuro.....	49

10.	BIBLIOGRAFÍA	50
11.	ANEXOS	53

Ilustraciones y tablas

Ilustración 1	Árbol de problemas.....	23
Ilustración 2	Cronograma de actividades en diagrama de Gantt	27
Ilustración 3	Diagrama de casos de uso.....	28
Ilustración 4	Diagrama de arquitectura.....	29
Ilustración 5	Estructura de colecciones	30
Ilustración 6	Diseño página principal.....	31
Ilustración 7	Diseño página de resultados	31
Ilustración 8	Pseudocódigo buscador de palabras clave.....	33
Ilustración 9	Pseudocódigo Análisis de similaridad	34
Ilustración 10	Pseudocódigo Análisis de similaridad con BERT.....	35
Tabla 1	Tabla de criterios de tecnologías	36
Ilustración 11	Generación de información para dataset con ChatGPT y Claude.ai	38
Ilustración 12	Dataset de situaciones y clases	40
Ilustración 13	Estructura de APIs	42
Ilustración 14	Pantallazo aplicación frontend Home	43
Ilustración 15	Pantallazo aplicación frontend Búsqueda por reglamentos.....	44
Tabla 2	Evaluación de la aplicación de acuerdo con estudiantes	47

Resumen del Proyecto:

En el presente documento se hace una propuesta de solución a problemáticas relacionadas con la dificultad en la comprensión, falta de conocimiento y acceso a los reglamentos institucionales por parte de la comunidad estudiantil del Politécnico Grancolombiano. La existencia de esta situación se corroboró mediante una encuesta a miembros de la comunidad académica, en la cual se evidenció un bajo nivel de familiaridad con las normas y una percepción negativa hacia ellas. Para abordar esta problemática se desarrolló una aplicación web de asesoramiento normativo utilizando inteligencia artificial. Esta aplicación permite a los usuarios obtener respuestas relacionadas con situaciones o preguntas específicas y provee un acceso a la información actualizada del reglamento académico. Para el desarrollo se utilizaron técnicas de procesamiento de lenguaje natural (PLN) y clasificación de texto con redes neuronales, aprovechando modelos de lenguaje grandes (LLM) como BERT. Se realizó un proceso de selección y entrenamiento de la herramienta con información relacionada con el reglamento académico, para construir módulos que permitan manejar y presentar dicha información. Se espera que la aplicación mejore el acceso y comprensión de las normas institucionales, empoderando a los estudiantes para resolver sus dudas de manera autónoma y eficiente, lo cual no solo optimizará los recursos institucionales, sino que también fomentará una cultura educativa más colaborativa y autónoma. Para evaluar el funcionamiento de la aplicación se utilizó un instrumento de recolección de información con un grupo de estudiantes, quienes probaron la herramienta y evaluaron criterios como usabilidad, rendimiento y precisión.

Palabras clave:

Asesor Normativo, Redes Neuronales, Procesamiento de lenguaje natural

1. TÍTULO DE LA PROPUESTA:

Desarrollo de un asesor normativo mediante inteligencia artificial para el reglamento estudiantil del Politécnico Grancolombiano

2. INTRODUCCIÓN

En el contexto general de nuestro país el desconocimiento de las normas o leyes es un problema común que es visible en la sociedad actual, a pesar de que los recursos como libros y páginas web acerca de leyes, derechos y reglamentos están al alcance de la gran mayoría de la población, se nota de manera contundente el desconocimiento de gran parte de la población acerca de sus derechos, deberes y procedimientos que rigen las instituciones, llegando así al punto en que suceden situaciones particulares que violentan las normas y las personas involucradas no hacen uso de sus derechos legales para resolver estos inconvenientes. En este contexto, el Politécnico Grancolombiano no es una excepción, la opinión de los estudiantes respecto a los reglamentos y normas indica que a pesar de que estos documentos están al alcance de la comunidad, su extensión y complejidad provocan desaliento al momento de leerlos o intentar comprenderlos, lo cual puede desembocar en que los estudiantes recurran al personal administrativo para resolver sus inquietudes, lo cual en ocasiones representa una carga innecesaria para el personal y vuelve un poco más lentos los procesos administrativos.

Hasta la fecha, variedad de investigadores han dedicado sus esfuerzos a crear herramientas que ayuden a solventar el problema del desconocimiento de las normas y el no acceso a estas mismas, para ello se han creado distintos tipos de aplicaciones, desde chatbots jurídicos con inteligencia artificial a buscadores que utilizan técnicas de procesamiento del lenguaje natural.

Esta investigación, se realiza por el interés en brindar una solución innovadora y útil que ayude a mejorar la problemática evidenciada en la universidad, además con base en las tendencias de los últimos años por automatizar procesos y tareas, las tecnologías que utilizan inteligencia artificial son de llamativo interés para elaborar una solución.

El principal objetivo de este proyecto es desarrollar una aplicación web que utilice inteligencia artificial para asesorar a los estudiantes del Politécnico Grancolombiano sobre el reglamento académico. Para su desarrollo se desglosa en pequeñas metas: seleccionar un sistema de procesamiento de lenguaje natural y modelos de inteligencia artificial adecuados para el proyecto, entrenar la herramienta con información relacionada con el reglamento académico, proveer una interfaz de usuario intuitiva y fácil de usar, y evaluar la usabilidad y la experiencia de usuario de la aplicación mediante pruebas con usuarios finales.

La metodología implementada para alcanzar los objetivos mencionados se compone de varias etapas clave. En primer lugar, se realizará un análisis preliminar que permite identificar las necesidades específicas de los usuarios y abordar las causales y efectos del problema a tratar. Se llevarán a cabo encuestas para recolectar datos y comprender mejor las percepciones y necesidades de los estudiantes. El diseño de la solución

abarcará la definición de los requerimientos de la aplicación, casos de uso, arquitectura del sistema, base de datos y diseño de interfaz de usuario. La implementación del proyecto se llevará a cabo en varias fases, empezando con el prototipado, el entrenamiento del modelo de inteligencia artificial, el desarrollo del módulo de procesamiento de texto, backend y frontend. Cada fase será sometida a pruebas para asegurar su correcto funcionamiento. Finalmente, se realizarán pruebas con usuarios finales para evaluar la usabilidad y la experiencia de usuario de la aplicación. Estas pruebas permitirán identificar la percepción de los estudiantes ante la aplicación y su impacto en los procesos.

3. PLANTEAMIENTO DEL PROBLEMA:

El Politécnico Grancolombiano como institución de educación superior, tiene un conjunto de normas y reglamentos que regulan la convivencia y el funcionamiento académico. Con base en esto, se han implementado una serie de procedimientos y herramientas que ayudan a la comunidad a resolver los problemas que suceden a menudo en su vida universitaria. Sin embargo, algunos miembros de la comunidad manifiestan su insatisfacción al respecto.

En la búsqueda de contemplar la situación y analizar más detalladamente el punto de vista de la comunidad del Politécnico, como objeto de investigación se realiza una encuesta a una toma muestral de 54 individuos que hacen parte de la comunidad de la institución. Donde se evidencia cierta insatisfacción de los encuestados dado que las encuestas dejan las siguientes afirmaciones:

- Los estudiantes tienen un bajo nivel de familiaridad con las normas y reglamentos, y tienen una opinión negativa sobre ellos
- Los estudiantes no leen los reglamentos por considerarlos extensos, complejos o poco relevantes, a pesar de que afirman que es fácil acceder a los recursos académicos de reglamentos y políticas.
- La gran mayoría de los encuestados están de acuerdo y usarían (En un caso hipotético) una aplicación que les ayude a guiarlos con temas de normas y procedimientos para una situación específica que les esté sucediendo.

A partir estas afirmaciones, se puede analizar que, al no conocer las normas, los individuos pueden tomar decisiones erróneas para resolver los inconvenientes que se presentan en la universidad, sin tener un respaldo normativo. Esta falta de conocimiento también se traduce en un mayor número de consultas y solicitudes de información al personal administrativo, lo que consume tiempo y recursos que podrían utilizarse para realizar a otras actividades.

4. OBJETIVOS

Mejorar el acceso y comprensión de la información acerca de procedimientos y normas de la institución a través de una aplicación de asesoramiento normativo utilizando inteligencia artificial.

Objetivos específicos

- a. Diseñar un sistema de procesamiento de lenguaje natural usando modelos de inteligencia artificial que permitan clasificar textos según un entrenamiento para comprender las consultas y situaciones planteadas por los usuarios de la aplicación.
- b. Entrenar la herramienta diseñada con la información de preguntas y situaciones relacionadas con el contenido de las normas del reglamento académico.
- c. Proveer una interfaz de usuario que facilite la navegación y el acceso a la información brindada por la herramienta.
- d. Evaluar el funcionamiento, la experiencia de usuario y la usabilidad de la solución desarrollada con base en la opinión de usuarios potenciales.

5. JUSTIFICACIÓN

En el contexto delineado en el planteamiento del problema, la necesidad de abordar la falta de conocimiento normativo y las dificultades para acceder a los reglamentos institucionales es evidente. Por tanto, el desarrollo de una aplicación que facilite el acceso y comprensión de esta información se presenta como una solución innovadora y práctica.

Esta aplicación proporcionará orientación en el marco normativo del Politécnico Grancolombiano para situaciones de acuerdo con las normas establecidas en el reglamento, teniendo un impacto a nivel educativo que permita empoderar a los estudiantes para resolver sus dudas de manera autónoma y eficiente con base en las reglas establecidas. Además de evitar los inconvenientes asociados con el desconocimiento normativo y la dependencia excesiva del personal. Se espera mejorar la convivencia y el funcionamiento académico del Politécnico Grancolombiano, optimizando los recursos institucionales y fomentando una cultura educativa más colaborativa y autónoma.

Debido al interés propio en las tecnologías que usan inteligencia artificial para optimizar procesos y tareas, es de significativo valor planear e implementar una solución innovadora que pueda mejorar el acceso a la información de los reglamentos haciendo uso de tecnologías que implementan inteligencia artificial.

Además, es importante resaltar que la integración de herramientas de inteligencia artificial en la aplicación ofrece beneficios potenciales que justifican su implementación en el proyecto. Las herramientas con IA pueden procesar grandes volúmenes de información, lo cual facilita tareas como la búsqueda de palabras claves en textos y el procesamiento del lenguaje natural, los cuales son factores importantes para la construcción de un módulo de procesamiento de texto, el cual se proyecta como el componente principal de la aplicación.

6. ALCANCE

La solución por desarrollar se lleva a cabo bajo el contexto de un trabajo de grado y se limitará a:

Proveer una aplicación web que actúe como asesor normativo frente a una situación específica. La aplicación brindará una respuesta indicando las normas que están relacionadas con la situación expuesta, además permitirá ver la información actualizada de las normas del reglamento académico vigente a la fecha del Politécnico Grancolombiano.

La aplicación se ejecutará en ambiente local, en el cual se mostrará la aplicación y será objeto para la ejecución de las pruebas con usuarios finales.

Además, se tienen en cuenta las siguientes limitantes para el desarrollo de la solución: Debido al análisis de los recursos que se tienen disponibles para el desarrollo e investigación y la alta demanda de recursos de hardware que requieren las aplicaciones con inteligencia artificial se opta por presentar una solución utilizando herramientas de código abierto que en promedio consuman pocos requerimientos computacionales, las cuales en conjunto puedan brindar una solución precisa y útil para resolver el problema expuesto.

Asimismo, al analizar la alta demanda de información que requieren las herramientas a utilizar para lograr una precisión considerable, se limita el alcance del entrenamiento a situaciones que hagan referencia únicamente al reglamento académico.

7. MARCO TEÓRICO

Marco Teórico

- **PLN:**

El Procesamiento de Lenguaje Natural (PLN), o Natural Language Processing (NLP) en inglés, es una subdisciplina de la inteligencia artificial que se centra en el análisis del lenguaje humano. Se divide un conjunto de técnicas computacionales que permiten a las máquinas comprender, interpretar y generar el lenguaje natural de los seres humanos tanto en forma de texto como de voz.

Según la afirmación "El Procesamiento del Lenguaje Natural (PLN) es un subcampo de la Inteligencia Artificial (IA) que se ocupa de la interacción entre humanos y ordenadores en lenguaje natural.", Vanita (2024), esta disciplina abarca una amplia variedad de tareas como el análisis sintáctico y semántico del texto, el reconocimiento de entidades, la traducción automática, el análisis de sentimientos y la generación de texto.

Este enfoque del PLN ha sido impulsado por técnicas de aprendizaje profundo que han sido de gran ayuda para mejorar la precisión y eficiencia de estos sistemas. Debido a esto, ha dejado una gran huella y creó los cimientos de una tendencia por automatizar y crear herramientas que entiendan el lenguaje humano y puedan resolver problemas, generar texto, incluso que puedan tener la capacidad de razonar simulando el funcionamiento del cerebro humano, lo cual cambiará la forma en que los humanos interactúan con las computadoras.

- **Clasificación de texto con redes neuronales:**

Las redes neuronales son modelos computacionales que imitan el comportamiento de las redes neuronales en el cerebro humano y se usan para procesar información y realizar tareas específicas a través del aprendizaje automático. A partir de esto, surgió la idea de utilizar estos algoritmos para abordar la clasificación de texto, la cual se enfoca en usar modelos de redes neuronales recurrentes y convolucionales con el objetivo de clasificar grandes volúmenes de datos textuales en múltiples clases o categorías. Esta técnica utiliza representaciones de palabras y unidades recurrentes para comprender la información contextual en los textos. Mannadiar, N., & Gürsoy, K. (2019) exponen que "la clasificación de textos es una técnica básica para el manejo y la organización de datos textuales" y "ha sido utilizada con cierto nivel de éxito en áreas de gran avance como la creación de resúmenes, el diagnóstico médico, la clasificación de noticias, el diagnóstico de spam y el análisis de sentimientos." Estas redes neuronales pueden aprender de los mismos datos textuales y ajustar sus parámetros para mejorar la clasificación de la precisión al emplear técnicas que incrustan palabras en un espacio semántico y capas de max-pooling (proceso de discretización basado en muestras) para identificar las características más importantes en los textos. Lo cual resulta mejor que la implementación de

modelos tradicionales al comprender de una forma más detallada el contexto que rodea a las palabras y las frases dentro de una oración.

- **LLM:**

Un Modelo de Lenguaje Grande, Large Language Model (LLM) en inglés, es un sistema avanzado de inteligencia artificial que puede manejar y generar texto coherentemente simulando el entendimiento del lenguaje humano. Estos modelos están basados en arquitecturas que permiten el procesamiento en paralelo de secuencias de texto y que permiten capturar las relaciones contextuales entre palabras y frases. Para ello, estos modelos están entrenados con datasets (término que se refiere a colecciones estructuradas de información) con grandes cantidades de datos que normalmente oscilan en el rango de los gigabytes o terabytes, lo cual facilita el "aprendizaje de representaciones lingüísticas profundas y generalizables" (Naveed, H., Khan, A.U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Barnes, N., & Mian, A.S., 2023). Además, estos modelos usan un preentrenamiento auto-supervisado seguido de ajustes para tareas específicas que han mejorado su capacidad de generación de texto y su rendimiento en una amplia variedad de aplicaciones, desde la traducción automática, hasta la generación de textos creativos. De hecho, estos modelos muestran capacidades de razonamiento, planificación y toma de decisiones que son sorprendentes, y por lo tanto útiles en campos distintos como la robótica, la interacción entre humano-computadora y la automatización de procesos.

- **BERT:**

El modelo de representación del lenguaje BERT (Bidirectional Encoder Representations from Transformers) desarrollado por Devlin et al. en Google AI Language. BERT se destaca por ser un modelo de representación del lenguaje bidireccional, lo que significa que preentrena representaciones profundas bidireccionales a partir de texto no etiquetado, lo cual condiciona simultáneamente el contexto izquierdo y derecho en todas las capas. Esto permite hacer una comprensión más detallada y completa del contexto de una palabra en una oración, así como lo menciona Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019) "BERT está diseñado para preentrenar representaciones bidireccionales profundas a partir de texto sin etiquetar condicionando conjuntamente el contexto izquierdo y derecho en todas las capas".

Para entrenar este modelo se realizan dos fases principales:

En primera instancia un preentrenamiento con grandes cantidades de texto sin etiquetar haciendo una tarea de modelo de lenguaje enmascarado (MLM) que entrena al modelo para predecir tokens ocultos en el texto y una tarea de predicción de la siguiente oración que ayuda al modelo a entender la relación entre dos frases.

Luego, se realiza la fase de Fine-Tuning para configurar la capacidad del modelo para realizar una amplia gama de tareas de procesamiento del lenguaje natural

como responder preguntas y deducción del lenguaje. Para ello, se agrega una capa de salida adicional al modelo preentrenado utilizando datos etiquetados relacionados para cumplir una tarea específica.

El fine-tuning (ajuste fino) es un proceso en el aprendizaje profundo donde se toma un modelo preentrenado, que ya ha sido entrenado en un conjunto de datos grandes y se le aplican ajustes adicionales para adaptarlo a una tarea específica. Para ello se utiliza un conjunto de datos más pequeño y etiquetado que es relevante para la tarea deseada. Esta técnica permite que el modelo transfiera el conocimiento adquirido durante el preentrenamiento a la nueva tarea, lo que generalmente resulta en un mejor rendimiento que entrenar un modelo desde cero para un fin específico. Lo que ha demostrado ser extremadamente eficaz en diversas tareas como la respuesta a preguntas, el reconocimiento de entidades nombradas y la inferencia de relaciones semánticas. Como menciona Church, K.W., Chen, Z., & Ma, Y. (2021) "El ajuste fino se ha popularizado recientemente, en gran parte porque funciona muy bien en muchos de nuestros puntos de referencia más populares", uno de los principales beneficios es su capacidad para combinar datos etiquetados y no etiquetados para optimizar el uso de recursos disponibles. Debido a que los datos etiquetados suelen ser difíciles de obtener, esta técnica resulta sobresaliente para maximizar la eficiencia y efectividad del modelo, utilizando únicamente una pequeña cantidad de datos etiquetados complementados con grandes conjuntos de datos no etiquetados.

Para el modelo, permite que este se vaya adaptando el conocimiento de un LLM robusto para enfatizarlo en tareas específicas, lo que posibilita generar nuevas predicciones con base en un conjunto de datos etiquetado más pequeño, esto permite que BERT se adapte diferentes tareas con unos cuantos parámetros adicionales y sin modificar significativamente la arquitectura del modelo.

Este modelo configurado con las técnicas mencionadas anteriormente es bastante potente, lo cual lo ha posicionado como un punto de referencia para la ejecución de tareas de NLP al proporcionar un arquitectura poderosa, flexible y efectiva que permite un mejor entendimiento del contexto bidireccional del lenguaje, lo que le da un plus sobre modelos antiguos como ELMo y OpenIA GPT, que utilizan enfoques basados en características o de ajuste fino. Además, al comparar este modelo con otros sistemas en todas las tareas, BERT presenta una mejora sustancial en el rendimiento promedio comparándolos con las técnicas utilizadas anteriormente.

Estado del Arte:

En los últimos años, la implementación de tecnologías basadas en la inteligencia artificial se ha convertido en un factor clave en la optimización y mejora de los

procesos que habitualmente se manejan en varias industrias. La IA es capaz de analizar grandes cantidades de datos, identificar patrones complejos y generar su propio aprendizaje, lo que le permite optimizar procesos para realizar tareas repetitivas y difíciles.

Ya que esta investigación está enfocada hacia el ámbito educativo y normativo, es de considerable importancia revisar un poco acerca de cómo en los últimos años se han implementado las tecnologías basadas o desarrolladas con inteligencia artificial para mejorar procesos y dar soluciones a los problemas que surgen en los sectores de la educación y en temas relacionados con normativas. Se realiza una búsqueda exhaustiva de documentos, pero no se encontraron herramientas con inteligencia artificial referentes al tema, debido a esto, en esta sección se exponen artículos relacionados con implementaciones de herramientas que usan inteligencia artificial para temas de asesoría legal.

Uno de los casos analizados es el que describen Nataraj, P., Devaraj, Teja, R., Kumar, M., & Gangrade, A. (2023) en su documento *Development of a Legal Document AI-Chatbot*, donde presentan el desarrollo de un chatbot de inteligencia artificial para la documentación legal como una solución prometedoras para agilizar el manejo y comprensión de documentos con asuntos legales. Para desarrollar esta solución plantean tres módulos, los cuales se componen de; una aplicación Android desarrollada en Kotlin y XML, que proporciona una interfaz que permite al usuario realizar preguntas acerca de documentos legales. Un módulo que implementa la librería Langchain que funciona como un marco de procesamiento de lenguaje natural utilizando el LLM de ChatGPT desarrollado por OpenAI, el cual se utiliza para comprender el contexto de las consultas del usuario. Por último, una aplicación desarrollada con Python y el framework Flask que actúa como el backend para procesar las consultas del usuario a través de solicitudes POST.

Los autores destacan la elección de Flask como principal tecnología para la aplicación backend debido a su naturaleza ligera y flexible, soporte para API RESTful, integración con Python (el lenguaje en el que está escrito Langchain) y facilidad de escalabilidad y alojamiento. Además de esto describen detalladamente los módulos implementados donde se evidencia un acople y manejo de las herramientas bastante sencillo y escalable. Como conclusión, los autores destacan el potencial futuro del proyecto para convertirse en un chatbot estándar de la industria, con posibles mejoras en características como el entrenamiento de IA, la carga de documentos en la aplicación Android y el límite de tokens de consulta. Además, mencionan la asertividad de las respuestas de la aplicación y su capacidad para responder preguntas relacionadas con los temas legales dentro del contexto de la constitución india.

En el mismo ámbito de los temas legales, Morgan, J., Paiement, A., Seisenberger, M., Williams, J., & Wyner, A.Z. (2018) exponen en su documento *A Chatbot Framework for the Children's Legal Centre*, una alternativa innovadora para ayudar a los niños a acceder a

sus derechos legales, integrando técnicas de aprendizaje automático, un grafo de diálogo y la extracción de información. La solución propuesta es un chatbot que sirve como una interfaz de comunicación entre los niños y un asesor legal, lo que permite que la información legal esté fácilmente disponible para los jóvenes a través de una conversación. La herramienta identifica en primera instancia las circunstancias legales y las partes, de lo cual genera un caso para un asesor legal. Al comienzo de la conversación, se crea un grafo de diálogo para rastrear los intercambios de mensajes y así identificar el espacio en el que se desarrolla la conversación. El chatbot procesa las respuestas del usuario usando un modelo de red neuronal que clasifica los mensajes basados en su acto de habla y tipo de legal, lo que permite identificar el contexto del caso legal. La herramienta también fue diseñada para que pueda reconocer y extraer entidades nombradas, como nombres, ubicaciones y momentos, sin tener que preguntar explícitamente por ellas. Este proceso se realiza mediante el uso de expresiones regulares para las entradas que estén correctamente formateadas como direcciones de correo electrónico. Después de todo el proceso de análisis y de extracción de información, el chatbot genera un informe para el asesor legal, que incluye la información recopilada durante la interacción. Este enfoque permite a los niños que no tienen un conocimiento amplio de temas acerca de sus derechos o de documentación legal, poder expresar sus problemas en lenguaje cotidiano y recibir asistencia legal de manera más accesible y comprensible.

En cuanto a la medición del rendimiento y facilidad de uso de la aplicación realizan encuestas a un grupo de niños que usaron el chatbot para crear un caso. Para ello utilizan variables como la facilidad de uso, rendimiento de interacción, cortesía y respuestas, comprensión percibida y uso futuro. Estas variables son evaluadas con un puntaje que se asigna a cada pregunta de la encuesta que está relacionada con la variable.

Los autores dejan como conclusión que desarrollaron una aplicación que verdaderamente mejora el acceso de los niños a un asesor legal y el conocimiento de sus derechos legales. Además, resaltan la importancia del método implementado para el aprendizaje automático, ya que le brinda la capacidad a la aplicación de realizar predicciones cada vez más precisas para definir el acto de habla y el tipo jurídico que describe el usuario.

Estas aplicaciones de inteligencia artificial dejan un gran aporte para los investigadores e incentivan a mejorar los procesos y técnicas que se han usado para desarrollar las soluciones ejecutadas. De esa manera, se han venido utilizado nuevas herramientas para la implementación de chatbots o aplicaciones de asesoramiento.

En esta misma área Cui, J., Li, Z., Yan, Y., Chen, B., & Yuan, L. (2023) exponen en su artículo la elaboración de un asistente legal colaborativo basado en múltiples agentes y un modelo de mezcla de expertos (MoE) integrado con grafos de conocimiento. Su principal objetivo fue desarrollar una herramienta precisa que manejara de forma

coherente y eficaz las conversaciones que tratara, para ello utilizaron diferentes técnicas para mejorar y potenciar el poder y conocimiento de un LLM con su modelo MoE, el cual se basa en aumentar el espacio de parámetros para permitir que diferentes expertos se encarguen de tareas de asesoramiento específicas, lo cual mejora considerablemente la precisión de la consulta legal. Además, para su entrenamiento crearon un conjunto de datos legales de alta calidad con múltiples evaluaciones e integraron los conocimientos similares en un grafo de conocimiento, lo cual garantizó la precisión de los datos. Y para dar una estructura más flexible y escalable a su modelo implementaron procedimientos operativos estandarizados que se basan en los flujos de trabajo que se manejan en las firmas de abogados, lo que permite implementar el modelo multi-agente que permite la estandarización de roles para desarrollar tareas específicas como la recolección de información, investigación de materiales, asesoramiento legal y redacción de informes.

Para evaluar el modelo propuesto, se enfocaron en dos puntos de referencia y lo compararon con modelos ya existentes donde se evaluaron criterios como la comprensión y dominio de los sistemas legales, regulaciones, jurisprudencia y procedimientos legales. El primer punto de referencia se resume en la elaboración de pruebas en LawBench42, una herramienta de evaluación integral basada en el sistema legal chino, la cual evalúa tres criterios cognitivos: memoria, comprensión y aplicación de conocimientos jurídicos. Los resultados de estas pruebas fueron satisfactorios, dejando un puntaje promedio superior al modelo GPT-4, lo cual resalta la efectividad del modelo en diferentes niveles cognitivos.

En segunda instancia, evaluaron su modelo aplicándolo en el examen de calificación unificado para profesionales jurídicos de China, el cual tiene como objetivo evaluar la comprensión y capacidad de aplicación de los conceptos, principios y disposiciones legales para los LLM. La puntuación de esta prueba fue sobresaliente, ya que superó notablemente a los demás modelos evaluados, lo cual dejó evidenciada la capacidad del modelo para manejar preguntas de tipo legal.

Según las evaluaciones y métodos implementados, este modelo se destaca por ser preciso y eficaz para resolver problemas o situaciones basándose en el conocimiento del marco legal. Además, los autores resaltan los desafíos futuros para mejorar su aplicación puntualizando aspectos como la protección de la privacidad, la retención de conocimientos para que los usuarios puedan retomar el contexto del chat sin tener que volver a ingresar la información relevante, la búsqueda por mejorar el consumo de recursos computacionales, debido a que el modelo no es del todo rápido cuando hay consultas en simultáneo de diferentes usuarios, para lo cual plantean el uso de técnicas de cuantificación o destilación para hacer un modelo más pequeño que también pueda ser usado en dispositivos personales.

Algo similar a esto se realizó en Canadá, donde se evidenció una problemática social debido al alto costo de los servicios legales, lo cual excluye a personas vulnerables y las

presiona a representarse a sí mismas en los tribunales. Para solventar este problema Queudot, M., Charton, E., & Meurs, M. (2020), presentan dos modelos de chatbots para proporcionar información legal en su artículo *Improving Access to Justice with Legal Chatbots*. Este artículo destaca la falta de representación de la comunidad como un factor significativo en los problemas de acceso a la justicia, donde los litigantes no representados enfrentan desafíos adicionales al presentar demandas. El proyecto para crear una solución a la problemática mencionada es nombrado LegalIA, este involucra a investigadores de varias universidades en Montreal, con el objetivo de desarrollar un chatbot para ayudar a actores legales como abogados, jueces y litigantes. Para ello estructuran y entrenan los chatbots con una metodología de 6 etapas; en primera instancia, realizan una recopilación de datos obtenidas de fuentes oficiales organizando la información en forma de preguntas frecuentes. Luego, la segunda etapa tiene como objetivo la preparación de los datos donde crean conjuntos de datos etiquetados. Con estos datos ya procesados se da paso a la tercera fase, el diseño del chatbot, los cuales se configuran para proporcionar información legal a través de una interfaz conversacional, los cuales están basados en sistemas de recuperación de información y utilizan algoritmos de clasificación para identificar la intención del usuario. Como cuarto paso, se entrenan los modelos de los chatbots usando técnicas de aprendizaje automático supervisado, utilizando algoritmos como StarSpace y BERT para clasificar las intenciones de los usuarios y representar las conversaciones, utilizando el conocimiento de modelos preentrenados. En la quinta fase, se evalúan los chatbots utilizando métricas de precisión, recall y f1, con un conjunto de datos de validación y se recopilaban interacciones reales con usuarios para mejorar la afinidad de los conjuntos de datos de prueba. Por último, integran documentos de jurisprudencia en el chatbot para proporcionar un contexto adicional y mejorar las respuestas en casos donde no hay una solución clara.

Con base en lo anteriormente expuesto, el artículo presenta una contribución significativa al problema del acceso a la justicia mediante el uso de tecnología de inteligencia artificial para facilitar la información legal y el acceso a los servicios legales para todos los individuos de la comunidad.

En el mismo contexto de la implementación de tecnologías de inteligencia artificial, Han, H. (2023) de la universidad de Alabama, presenta en su documento *Potential Benefits of Employing Large Language Models in Research in Moral Education and Development*, cómo los LLMs que han sido entrenados con grandes conjuntos de datos de lenguaje pueden contribuir en la investigación de la educación y el desarrollo moral. Los LLMs tienen características que emulan el pensamiento humano como el aprendizaje en contexto y la cadena de pensamiento, las cuales eran difíciles de implementar en modelos anteriores. El autor propone que estos modelos podrían asistir en la investigación de la educación moral, a pesar de las limitaciones tecnológicas actuales. Para comprobar su teoría, pone a prueba la capacidad de ChatGPT para razonar moralmente con situaciones éticas y concluye que su desempeño se iguala al de estudiantes universitarios. Además, realiza un experimento que sugiere que los LLMs

podrían aumentar su comprensión moral al ser entrenados con historias con ejemplares morales de una forma similar a como se hace con los humanos.

De igual forma, indaga si los modelos pueden actualizar su razonamiento y decisiones morales basándose en información contextual, usando como ejemplo la Carta desde la Cárcel de Birmingham escrita por Martin Luther King Jr. Los resultados su investigación indican que los modelos utilizados pueden aprender lecciones morales y aplicarlas a nuevos contextos, lo que implica que poseen capacidades avanzadas de aprendizaje y razonamiento.

Basándose en sus pruebas y experimentos, concluye que los LLMs pueden ser herramientas que ayudarán a mejorar la educación moral proporcionando datos sobre cómo el aprendizaje en contextos morales influye en el funcionamiento moral de las personas. Además, estos modelos pueden ayudar a predecir los impactos de los materiales educativos en procesos cognitivos, emocionales y motivacionales antes de ser probados con los humanos. No obstante, también menciona que no se puede simular perfectamente la forma de pensar y el comportamiento humano a través de estas tecnologías, por lo que recomienda tener precaución en su uso como modelos de simulación.

8. METODOLOGÍA

Para la elaboración de este proyecto se plantean 4 fases, comenzando con un análisis preliminar para establecer los causales y efectos del problema a tratar, y recolectar la información necesaria para plantear posibles soluciones. Siguiendo con el diseño de la solución estableciendo los requerimientos, arquitectura y herramientas que serán usadas en la elaboración de la solución. Por consiguiente, se expondrá el desarrollo de la solución que abarca la descripción de los módulos desarrollados en la aplicación, además de la información pertinente de los experimentos y prototipos previos que se desarrollaron para luego ser implementados dentro de la herramienta propuesta. Por último, se exponen las pruebas y resultados obtenidos que denotan la calificación en base a los criterios establecidos para medir el impacto de la aplicación en los estudiantes de la universidad.

Análisis preliminar

En primer lugar, se realiza un análisis para puntualizar la problemática, analizar su causales y efectos que están sucediendo en la comunidad del Politécnico Grancolombiano, para ello se realizan una serie de técnicas que permiten identificar las opiniones de algunos individuos de la población, el impacto del problema, el planteamiento de cómo se abordará el desarrollo la solución y la planeación de las actividades y lapsos de tiempo para el desarrollo del proyecto.

Árbol de problemas

Para abordar el problema de una forma más directa se cree pertinente la elaboración de un árbol de problemas indicando el inconveniente principal y visualizar de manera clara las causas y efectos relacionados.

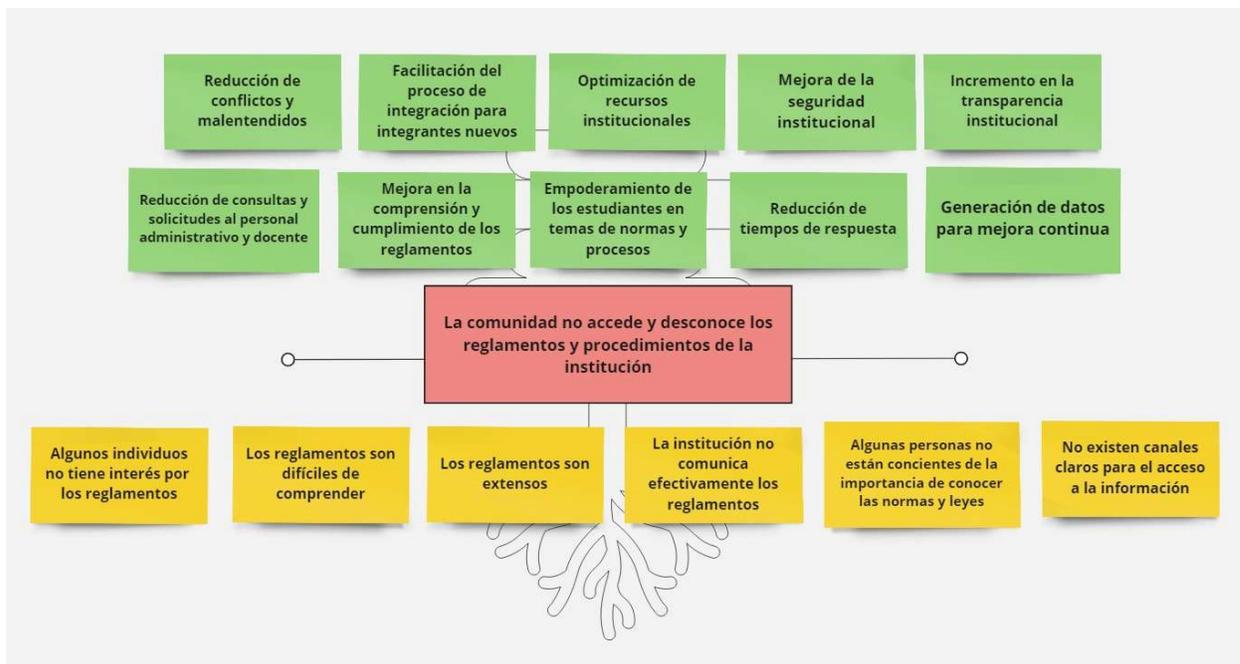


Ilustración 1 Árbol de problemas
Fuente: Elaboración propia

Según el diagrama ilustrado se puede concluir que la falta de acceso a la información sobre los reglamentos de la institución es un desafío complejo que afecta la experiencia educativa de los estudiantes. Los problemas identificados, que van desde la falta de conocimiento y dificultad de acceso hasta el desinterés y la falta de comunicación, destacan la necesidad de abordar esta cuestión de manera integral.

Debido a esto, se pueden plantear soluciones como; optimizar y mejorar el acceso la información mediante el uso de una herramienta tecnológica, fomentar el conocimiento e importancia de las normas y procedimientos mediante campañas estudiantiles, mejorar la comunicación de los reglamentos redactando de una manera más sencilla su estructura, implementar un asesor virtual para ayudar a los estudiantes a comprender los reglamentos o crear una oficina virtual de atención al estudiante para uso exclusivo en temas normativos.

De esta manera, sería ideal que la institución se comprometa con promover la transparencia y la accesibilidad de la información de temas normativos, implementando medidas para garantizar que los estudiantes puedan acceder y comprender los reglamentos de manera efectiva.

Encuestas

Para definir las necesidades e identificar la problemática evidenciada en el Politécnico Grancolombiano, se realiza una encuesta a un grupo de 54 miembros de la universidad

(Estudiantes, profesores y personal administrativo) con preguntas relacionadas acerca de los reglamentos de la institución, las cuales se pueden evidenciar en el [Anexo 1](#).

Con base en el análisis de los datos recolectados se evidencian conclusiones representativas como:

- Ninguno de los estudiantes encuestados afirma haber leído completamente los reglamentos
- El 70% de los encuestados afirman estar poco y nada familiarizados con los reglamentos.
- El 66.6% de los encuestados tienen una opinión negativa acerca de los reglamentos.
- El 63% de los encuestados afirma que es fácil encontrar información acerca de las normas, reglas y procedimientos.
- El 91% de los encuestados utilizaría una herramienta para entender de forma más sencilla las normas y que indique cuáles son los procedimientos que se deben seguir según una situación específica.

Definición de tecnologías

Con base en la literatura revisada, el análisis de los recursos destinados para esta investigación y desarrollo, y la experticia sobre el área, se plantea usar las siguientes tecnologías para el desarrollo de cada módulo de la aplicación:

- **Módulo frontend:** El módulo frontend constará de alrededor de 6 páginas aproximadamente, así que por facilidad de uso y por ser una librería liviana se usará React.js para desarrollar los componentes y elementos gráficos que se presentarán al usuario.
- **Módulo backend:** Para el backend de la aplicación se requiere un procesamiento rápido y compatible con las tecnologías a utilizar en el módulo de procesamiento de texto, debido a esto se utilizará Python y el microframework Flask, ya que brinda herramientas de fácil uso para la elaboración de APIs y peticiones HTTP.
- **Módulo de procesamiento de texto:** El módulo de procesamiento de texto requiere una gran capacidad para resolver tareas complejas relacionadas con procesamiento de datos y texto, para ello se elige Python, debido a que es un lenguaje muy versátil que cuenta con librerías optimizadas para el manejo de datos y procesamiento de lenguaje natural.

Para el desarrollo del módulo de procesamiento de texto se plantean las siguientes técnicas para conseguir los resultados esperados, la elección del método irá ligada al análisis de las conclusiones que surjan al ejecutar la fase del desarrollo de prototipos y experimentos:

Implementación de LLM re-entrenado con situaciones y respuestas en el mismo formato <Input-Output> generadas por chatbots con inteligencia artificial, utilizando técnicas de *cuantización* y *fine-tuning*. Lo cual generará respuestas con el formato <Solución-Normas>.

Uso de herramientas para modelar y configurar redes neuronales (Sciki-learn) para clasificación de texto, las cuales serán entrenadas con situaciones generadas por chatbots con inteligencia artificial en el formato <Clase;Situación>, la cual funciona analizando como parámetro de entrada una situación y resultando en la predicción de la clase a la cual pertenece o a la cual más se asemeja. Esta predicción se utilizará hacer un filtrado de las normas guardadas en formato *JSON* y previamente clasificadas mediante etiquetas, la cual provee como solución la lista de las normas relacionadas con la situación.

Implementación de librerías y tecnologías para procesamiento de lenguaje natural (spaCy y el modelo BERT) para analizar y vectorizar la información de la situación que viene como parámetro de entrada, para luego utilizar métodos que analicen variables como la similaridad entre el texto entrante y el contenido de cada norma, y la búsqueda de palabras clave entre ambos textos. Esto da como resultado la selección de las normas que coinciden con la situación ingresada.

- **Base de datos:** Para la gestión de los datos que se van a manejar dentro de la aplicación se selecciona *MongoDB* como motor de bases de datos debido a su estructura sencilla y rendimiento.

Actividades y cronograma

Para abordar el desarrollo de la solución planteada, se elabora un cronograma para organizar los tiempos y duración de las actividades a desarrollar, agrupando las tareas en cinco entregables tangibles que están anexados a este documento:

- **Entregable 1 - Análisis del problema y diseño de la solución:**
El primer entregable se compone de las actividades enfocadas a el análisis preliminar para identificar y estudiar el problema. Así como el diseño y planteamiento de requerimientos, elaboración de diagramas, definición de tecnologías y prototipado de las mismas. [Anexo 1](#)
- **Entregable 2 - Desarrollo de la solución (Módulo de procesamiento y análisis de texto):**
Para el entregable 2 se espera la ejecución y finalización de las tareas encaminadas al desarrollo del módulo de procesamiento y análisis de texto, que

se compone de actividades como; el entrenamiento y parametrización del módulo, su implementación en el contexto de la solución y la documentación respectiva de las pruebas.

- **Entregable 3 – Desarrollo de la solución (Módulo backend, módulo de buscador)**

El tercer entregable está compuesto por las tareas que hacen parte del desarrollo del módulo de backend y su integración con el sistema de procesamiento y análisis de texto, y el desarrollo del módulo de frontend de tipo buscador. También hace parte del entregable la documentación respectiva de las pruebas realizadas a los módulos.

Este ítem junto con el entregable 2 se almacenan de forma pública en dos repositorio de código:

FRONTEND:

https://github.com/jangel347/policonsejero_frontend

BACKEND:

https://github.com/jangel347/policonsejero_processor

- **Entregable 4 – Documentación pruebas:**

El entregable 4 se basa en la ejecución y documentación de pruebas para certificar la calidad y eficacia del aplicativo, se compone de la documentación unificada de todas las pruebas realizadas a cada módulo de la aplicación. Anexo 2

- **Entregable 5 – Documentación científica:**

Por último, se establecen el entregable requerido para la elaboración del documento científico que explica los detalles de la elaboración de la solución. El cual corresponde a la elaboración de este documento.

ENTREGABLES	ACTIVIDADES	TIEMPO DE DURACIÓN																											
		#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16	#17	#18	#19	#20	#21	#22	#23	#24	#25	#26	#27	#28
		FEBRERO				MARZO				ABRIL				MAYO				JUNIO				JULIO							
Entregable 1 - Análisis del problema y diseño de la solución	Recopilación y análisis de información (encuestas y análisis del problema)	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
	Definición de tecnologías, diseño de casos de uso, diagramas de flujo y mockups																												
	Diseño de arquitectura y base de datos																												
	Prototipado de tecnologías																												
Entregable 2 - Desarrollo de la solución (Módulo de procesamiento y análisis de texto)	Entrenamiento y configuración de sistema de procesamiento de lenguaje natural para analizar textos																												
	Implementación módulo de procesamiento de texto																												
	Pruebas módulo de procesamiento y análisis de texto																												
Entregable 3 - Desarrollo de la solución (Módulo backend y módulo de buscador)	Desarrollo de backend																												
	Desarrollo de módulo frontend buscador																												
	Pruebas de integración de módulos																												
Entregable 4 - Documentación de pruebas	Pruebas generales																												
Entregable 5 - Documentación científica:	Desarrollo de documento de anteproyecto																												
	Desarrollo de documento tesis																												

Ilustración 2 Cronograma de actividades en diagrama de Gantt

Fuente: Elaboración propia

Diseño de la solución

Requerimientos

Con base en la problemática evidenciada y la propuesta de una aplicación de asesoramiento normativo para resolver situaciones específicas y mostrar las normas que están involucradas, se plantean los siguientes requerimientos:

Requerimientos funcionales

- Se debe permitir la consulta de información por medio de la descripción de una situación o pregunta específica.
- La aplicación debe analizar la situación o pregunta, dar una respuesta y mostrar qué reglamentos o normas están siendo vulnerados según la información actualizada de los reglamentos y políticas de la universidad.
- La aplicación debe mostrar la información completa y verídica acerca de las normas, reglamentos y políticas de la universidad.
- La aplicación debe permitir al usuario evaluar el nivel de satisfacción y si fueron útiles las respuestas dadas.
- Debe permitir visitar los reglamentos y políticas de la página oficial de la institución.
- La aplicación mostrará en un apartado tips y tutoriales para el uso de la herramienta.

Requerimientos no funcionales

- La interfaz de la aplicación debe ser intuitiva y fácil de usar
- La interfaz de la aplicación debe ser clara y fácil de comprender
- La interfaz de la aplicación debe estar diseñada de forma responsive.

Requerimientos técnicos

- La aplicación debe estar disponible para su uso mediante cualquier navegador web en un ambiente local.

Casos de uso

Para el desarrollo de la aplicación se plantean dos roles (usuario y administrador) para poder navegar y ejecutar las funciones de la aplicación. Dichas funciones se describen a continuación:

Formatos de casos de uso

Los formatos de casos de uso correspondientes se evidencian en el [Anexo 2](#).

Diagrama de casos de uso

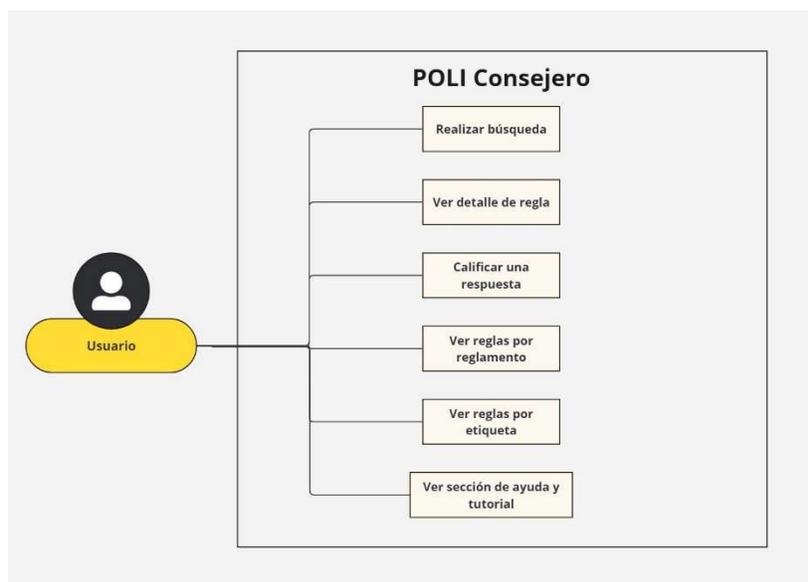


Ilustración 3 Diagrama de casos de uso
Fuente: Elaboración propia

Arquitectura

Diagrama de arquitectura

Para la elaboración de la aplicación, se plantea una arquitectura sencilla que se compone de dos componentes principales: el módulo frontend y el módulo backend.

El módulo frontend contiene toda la lógica de presentación de la aplicación. Su función principal es mostrar al usuario la información o los resultados correspondientes. Para

ello, se comunica con el módulo backend mediante peticiones HTTP consumiendo las APIs expuestas.

El módulo backend se encarga de exponer los servicios necesarios y de ejecutar la lógica de procesamiento de texto. Este módulo gestiona la conexión y comunicación con la base de datos para obtener la información requerida.

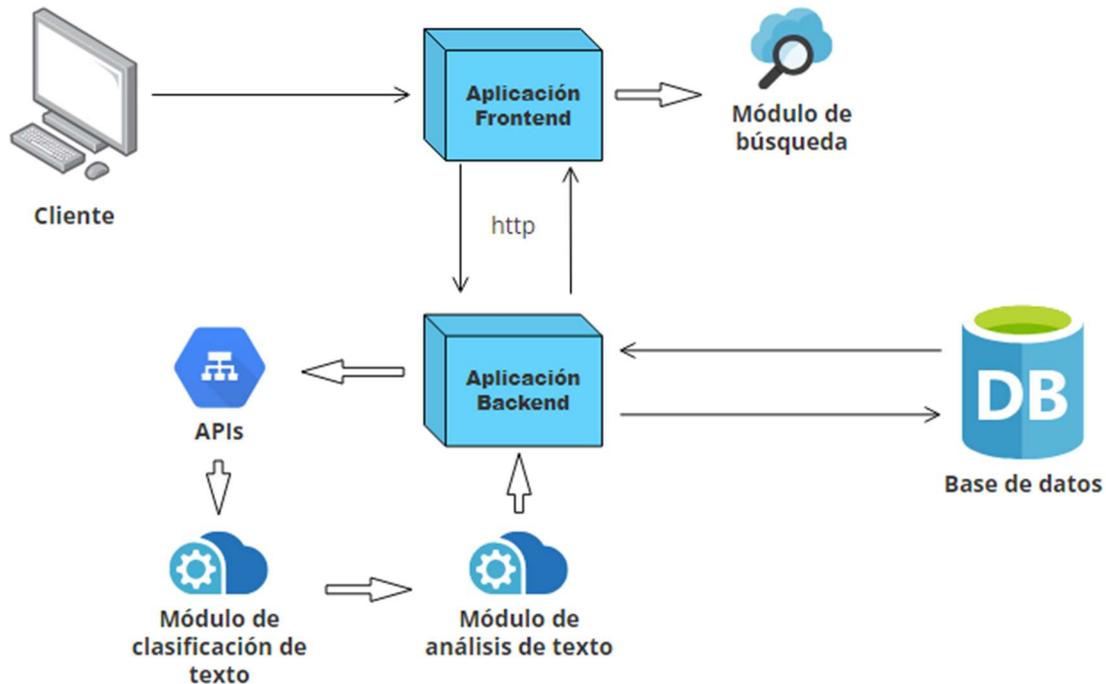


Ilustración 4 Diagrama de arquitectura

Fuente: Elaboración propia

Base de datos

Para la organización y persistencia de los datos se definen las siguientes colecciones:

Regulaciones:

Estructura que guarda la información de los reglamentos.

- **nombre:** Describe el nombre del reglamento

Rules:

Colección destinada para guardar los datos relacionados con las normas.

- **nombre:** Describe el nombre de la norma.
- **reglamento:** Describe el nombre del reglamento.
- **contenido:** Guarda el contenido de la norma.
- **etiquetas:** Guarda una lista con el nombre de las etiquetas relacionadas con la norma.

- **paginas:** Contiene una lista de las páginas del reglamento donde se puede ver la norma.

Tags:

Estructura que guarda la información de las etiquetas.

- **nombre:** Describe el nombre de la etiqueta.
- **respuesta:** Contiene una respuesta predeterminada para la etiqueta.

Stadistics:

Estructura destinada para guardar los datos de las encuestas para la evaluación de la aplicación.

- **usabilidad:** Describe la usabilidad del aplicativo en una escala de 1 a 5.
- **rendimiento:** Indica la usabilidad del aplicativo en una escala de 1 a 5.
- **precision:** Describe la precisión de la respuesta y normas en una escala de 1 a 5.
- **utilidad:** Denota qué tan útil fue el aplicativo en una escala de 1 a 5.
- **amigable:** Describe qué tan amigable es la interfaz de la aplicación en una escala de 1 a 5.
- **uso_futuro:** Indica si el usuario usaría nuevamente la aplicación.

Además de las características mencionadas para cada colección, todas las colecciones tienen el campo `_id` que actúa como identificador único para cada registro de cada colección.

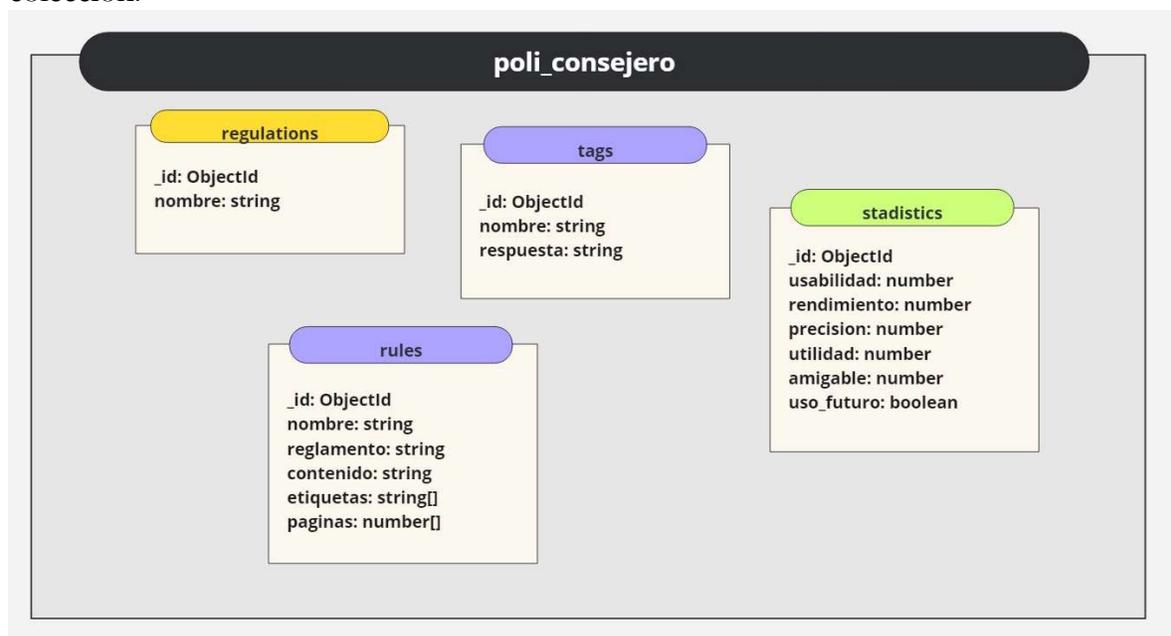


Ilustración 5 Estructura de colecciones

Fuente: Elaboración propia

Diseño de interfaz

Mockups

Para la maquetación de la interfaz que se entregará de cara al usuario se realizan maquetas sencillas que servirán de guía para posicionar los elementos dentro de la página web. La información completa de los mockups se encuentra en el Anexo 2.

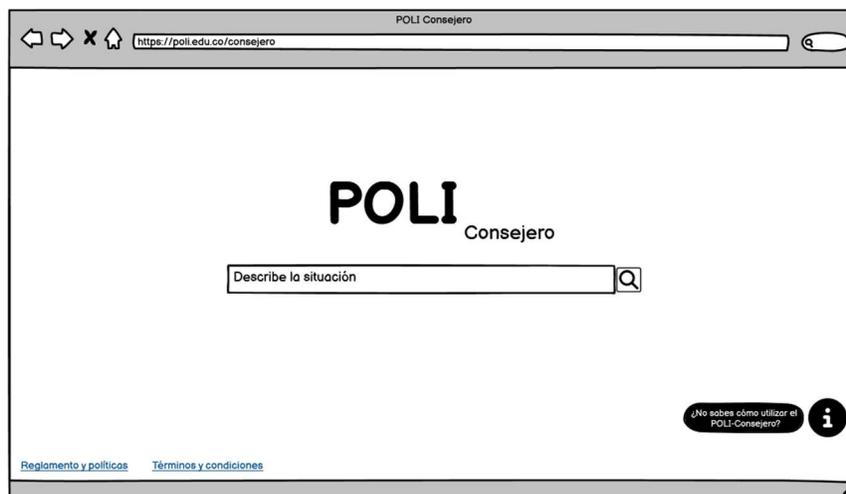


Ilustración 6 Diseño página principal

Fuente: Elaboración propia

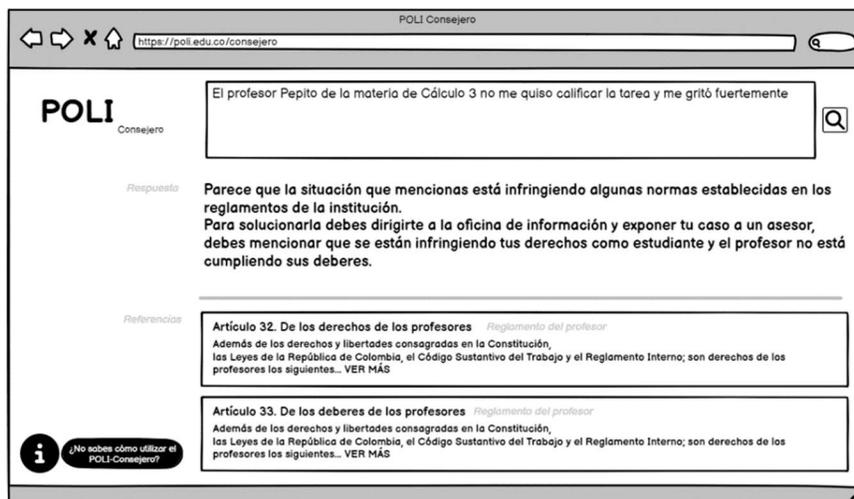


Ilustración 7 Diseño página de resultados

Fuente: Elaboración propia

Desarrollo de la solución

Prototipado y experimentación

Para poder tener una mejor comprensión de las herramientas a utilizar se realizan prototipos sencillos que cubran de manera simple las necesidades básicas del proyecto.

Buscador de coincidencias con SpaCy en Python

Se realiza un ejercicio sencillo para identificar si un conjunto de palabras clave extraídas de un texto inicial se encuentran presentes en un segundo texto. Para esto, se emplea una herramienta que facilita el procesamiento del lenguaje natural.

Este módulo consta de tres funciones principales que se encargan de procesar las características de los textos a evaluar. En primera instancia, la función *get_keywords* se encarga de limpiar el texto de signos de puntuación o palabras no relevantes y recolecta las palabras principales del texto ingresado, para lo cual utiliza una función de la librería que extrae las palabras más comunes en una cadena de texto, dicha función devuelve una lista con las palabras clave del texto entrante.

Por consiguiente, la función *match_filter_prepare* prepara un objeto para emparejar frases utilizando las palabras clave obtenidas anteriormente, añadiéndolas a un objeto de buscador proporcionado por la librería para que pueda encontrar coincidencias en otro texto. Posteriormente, una tercera función *match_filter* utiliza el buscador para verificar las coincidencias en el segundo texto. Si más del 70% de las palabras clave se encuentran en el segundo texto, se considera que hay una coincidencia significativa; en caso contrario, no la hay. Finalmente, para la ejecución del módulo se definen textos de entrada que se procesan para crear los documentos correspondientes, se prepara el buscador con el primer documento y se buscan coincidencias en el segundo. El resultado se imprime indicando si existe una coincidencia significativa entre las palabras clave del primer texto y el contenido del segundo.

Analizando los experimentos realizados con este ejercicio, se puede concluir que el módulo permite identificar de manera eficiente la relevancia de un conjunto de palabras clave en un texto objetivo, lo cual se puede considerar como una herramienta útil para buscar palabras clave dentro del contenido que hace parte de cada una de las normas del reglamento.

```

1 Algoritmo buscador_palabras_clave
2   Función obtener_palabras_clave(documento, cantidad)
3     Filtrar y contar palabras frecuentes
4     Devolver palabras clave comunes
5   Fin Función
6
7   Función preparar_buscador(documento)
8     Crear buscador
9     Obtener y añadir palabras clave
10    Devolver buscador
11  Fin Función
12
13  Función verificar_coincidencias(buscador, documento)
14    Buscar coincidencias en documento
15    Si coincidencias son suficientes
16      Devolver Verdadero
17    Devolver Falso
18  Fin Función
19
20  Texto situación
21  Documento documento_situación = procesar
22  Texto regla
23  Documento documento_regla = procesar
24
25  Buscador = preparar_buscador(documento_situación)
26  Resultado = verificar_coincidencias(buscador, documento_regla)
27  Imprimir resultado
28 FinAlgoritmo

```

Ilustración 8 Pseudocódigo buscador de palabras clave
Fuente: Elaboración propia

Análisis de similitud con SpaCy en Python

Este ejercicio evalúa la similitud entre dos textos, la cual se mide a través de los vectores de palabras de los textos proporcionados y se determina en función de un umbral predefinido.

En primera instancia, se define una función principal que toma dos parámetros y calcula la similitud entre ellos utilizando una función de la librería. Si la similitud supera el 50% se consideran similares y se devuelve un valor positivo con el grado de similitud. De lo contrario, se devuelve un valor negativo y el grado de similitud. Luego, para la ejecución del módulo se definen los textos de entrada y se procesan para crear los documentos correspondientes. La función de similitud se usa para calcular la similitud entre ambos documentos, el resultado se imprime indicando si los textos son similares y mostrando el valor de similitud evaluada.

Se realizan experimentos con este ejercicio y se concluye que es una herramienta útil para medir la similitud entre dos textos, lo que puede ser beneficioso para buscar textos relevantes que se asemejen. Llevándolo al contexto de esta investigación, puede ser de utilidad para buscar la similitud entre una situación o una pregunta, con el contenido de una norma del reglamento.

```

Algoritmo analisis_similaridad
  Función filtro_similitud(documento_regla, documento_situación)
    Calcular similitud entre los documentos
    Si similitud > 0.5
      Imprimir "Similar" y valor de similitud
      Devolver Verdadero y valor de similitud
    Si no
      Imprimir "No similar" y valor de similitud
      Devolver Falso y valor de similitud
    Fin Función

  Texto situación
  Texto regla
  Documento documento_regla = procesar texto regla
  Documento documento_situación = procesar texto situación

  Resultado = filtro_similitud(documento_regla, documento_situación)
  Imprimir resultado
FinAlgoritmo

```

Ilustración 9 Pseudocódigo Análisis de similaridad
Fuente: Elaboración propia

Análisis de similaridad con BERT en Python

Este módulo está diseñado para calcular la similitud entre dos textos utilizando técnicas de procesamiento del lenguaje natural y modelos de aprendizaje automático. Se comienza cargando un modelo pre-entrenado de procesamiento del lenguaje natural específico para el idioma español, el cual tiene la capacidad de realizar diferentes tareas de análisis de texto, tales como la lematización y la eliminación de palabras irrelevantes, lo cual es importante para una comparación precisa entre dos textos.

Por consiguiente, se realiza el preprocesamiento de los textos, en primera instancia se limpian y normalizan. Este proceso se basa en convertir todo el texto a minúsculas para asegurar la uniformidad, eliminar los signos de puntuación y los números que no aportan significado semántico, y reducir las palabras a su forma base mediante la lematización. Además, se eliminan las palabras comunes y poco significativas que no contribuyen al contenido semántico del texto. Esta etapa de preprocesamiento es fundamental para preparar los textos de manera que su comparación sea más precisa y significativa.

En el siguiente paso, los textos se transforman en vectores utilizando un modelo de lenguaje. Con esta conversión, este modelo es capaz de extraer el significado semántico del texto convirtiendo cada fragmento en un vector multidimensional. Estos vectores son utilizados para el proceso de comparación, para ello se emplea una medida matemática conocida como la similitud del coseno, que compara las orientaciones de los vectores en espacios multidimensionales. Como resultado, esta comparación y análisis arroja una medida que indica cuánto se parecen los textos entre sí.

Esto permite evaluar de manera eficiente y precisa la relación semántica entre diferentes textos. Con esto, se puede concluir que esta herramienta es precisa y efectiva para hallar

una relación entre dos textos, lo cual se puede implementar para buscar la similitud entre un texto que describe una situación o pregunta, con el texto del contenido de una norma del reglamento.

```
Algoritmo analisis_similaridad_bert
Función preprocess(texto_):
    texto_ = convertir_a_minúsculas(texto_)
    texto_ = eliminar_puntuación(texto_)
    texto_ = eliminar_números(texto_)
    Documento documento_texto = procesar(texto_)
    palabras = lematizar_y_eliminar_palabras_irrelevantes(documento)
    retornar texto preprocessado(palabras)
FinFunción

Función obtener_embeddings_BERT(texto):
    entradas = tokenizar texto(tokenizer, texto)
    salidas = generar_embeddings(modelo, entradas)
    retornar media de los embeddings(salidas)
FinFunción

texto texto1
texto texto2
texto1_limpio = preprocess(texto1)
texto2_limpio = preprocess(texto2)

tokenizer = cargar_tokenizer_preentrenado('bert-base-spanish-wmm-uncased')
modelo = cargar_modelo_BERT_preentrenado('bert-base-spanish-wmm-uncased')
embedding_texto1 = obtener_embeddings_BERT(texto1_limpio)
embedding_texto2 = obtener_embeddings_BERT(texto2_limpio)
similitud_coseno = calcular_similitud_del_coseno(embedding_texto1, embedding_texto2)

imprimir "Similitud del coseno entre los textos (BERT):", similitud_coseno
FinAlgoritmo
```

Ilustración 10 Pseudocódigo Análisis de similaridad con BERT

Fuente: Elaboración propia

Predicción de clases con redes neuronales

Para este ejercicio se implementa un módulo para clasificar mensajes de texto como spam o ham (no spam) basado en un artículo de la empresa *Aryys* llamado *Clasificación de textos con Python y Jupyter Notebook*. Se utiliza un conjunto de datos en formato CSV con 5575 registros que contienen mensajes y sus etiquetas correspondientes.

Para iniciar, se carga un archivo CSV que hace el rol del dataset y se seleccionan dos columnas. Para luego hacer el conteo de la cantidad de casos que corresponden a cada etiqueta, lo cual permite identificar la distribución de los datos. Además, se realiza una limpieza de los mensajes eliminando signos de puntuación y convirtiendo los caracteres a minúsculas.

En el siguiente paso, los textos se convierten en vectores de características usando una herramienta que crea una representación binaria de la presencia de palabras. El conjunto de datos se divide en subconjuntos de entrenamiento y prueba, asegurando que la proporción de etiquetas sea la misma en ambos conjuntos para evitar sesgos.

El modelo SVM lineal se entrena con los datos de entrenamiento transformados con el objetivo de separar las clases. Luego, se evalúa su rendimiento en el conjunto de prueba calculando la precisión, que mide la proporción de predicciones correctas.

Finalmente, se transforman nuevas frases en vectores de características y se utilizan para hacer predicciones con el clasificador entrenado. Las predicciones se imprimen junto con las frases originales para interpretar los resultados. El módulo tiene buen rendimiento y precisión, pero podría mejorar con un entrenamiento más robusto, lo cual es útil para la creación de sistemas de clasificación de texto más complejos basándose en este modelo.

Conclusiones de prototipos y experimentos

Para medir los prototipos y experimentos realizados se lleva a cabo una tabla con la explicación de algunos criterios que se tuvieron en cuenta para seleccionar las herramientas utilizadas.

Tecnología	Precisión	Rendimiento	Facilidad de uso	Flexibilidad	Requerimientos computacionales
Redes neuronales	Alta para clasificación	Moderado	Requiere conocimiento avanzado	Alta	Alto
BERT	Muy alta para PLN	Lento sin GPU	Requiere conocimiento avanzado	Alta para NLP	Muy Alto
Spacy (Similaridad)	Alta en PLN	Rápido	Muy fácil	Alta para NLP	Bajos
Spacy (Buscador)	N/A	Muy rápido	Muy fácil	Alta para NLP	Bajos

Tabla 1 Tabla de criterios de tecnologías

Fuente: Elaboración propia

Las redes neuronales tienen una alta precisión en tareas de clasificación complejas, pueden ser un poco lentas y requerir bastantes recursos computacionales. Además, pueden ser muy flexibles para ser adaptadas a diferentes tipos de datos y problemas. En cuanto a su configuración y utilización se deben tener conocimientos profundos de aprendizaje automático y ajuste de hiperparámetros.

BERT, también brinda un alto nivel de precisión en tareas de similaridad semántica y comprensión del contexto, pero tiene una alta demanda de recursos computacionales, ya que los modelos requieren de un entrenamiento con grandes volúmenes de datos para alcanzar mayor precisión en sus predicciones. Por otra parte, la flexibilidad para tareas de NLP complejas específicamente para el manejo de texto es considerablemente mejor. Este modelo es complejo de implementar y usar por sí solo, pero existen herramientas como *Torch* y *Transformers* que facilitan su uso.

En cuanto a la librería SpaCy, es una herramienta rápida y eficiente para tareas de NLP, aunque no puede alcanzar la precisión de BERT en tareas complejas de similaridad, pero consume considerablemente muchos menos recursos computacionales. Es muy adaptable para tareas específicas de procesamiento del lenguaje natural y cuenta con

una suite de funciones, modelos preentrenados y herramientas listas para ser implementadas.

Por otra parte, se analizaron tecnologías más complejas como la implementación de LLMs como Mistral que no pudieron ser implementadas en el aplicativo debido a problemas que se obtuvieron debido a la alta demanda de recursos computacionales que requieren.

Con base en los prototipos desarrollados y el análisis de las tecnologías, se decide utilizar una combinación de los módulos desarrollados para construir un módulo de procesamiento de texto que contribuya al desarrollo preciso y viable de la solución, el cual tiene las siguientes características:

- Implementación de funciones para analizar las coincidencias de palabras claves extraídas de la situación entrante y evaluándolas en el contenido de las normas de los reglamentos seleccionados.
- Análisis de similaridad entre el texto de la situación entrante y el contenido de las normas de los reglamentos.
- Predicción de la clase de la situación o pregunta que se obtiene como parámetro de entrada haciendo uso de una red neuronal para clasificación de texto.

Entrenamiento e implementación de modelo

Generación de datos

Para lograr un óptimo resultado en las predicciones del módulo de procesamiento de texto, se hace necesario poblar un dataset con situaciones y su clase asociada con el formato <tipo;situación>. Para esto, se hace uso de herramientas que usan inteligencia para generar texto, como lo son ChatGPT y Claude.ai en sus versiones de no pago, utilizando *prompts* estructurados para la generación de información destinada a poblar el dataset.

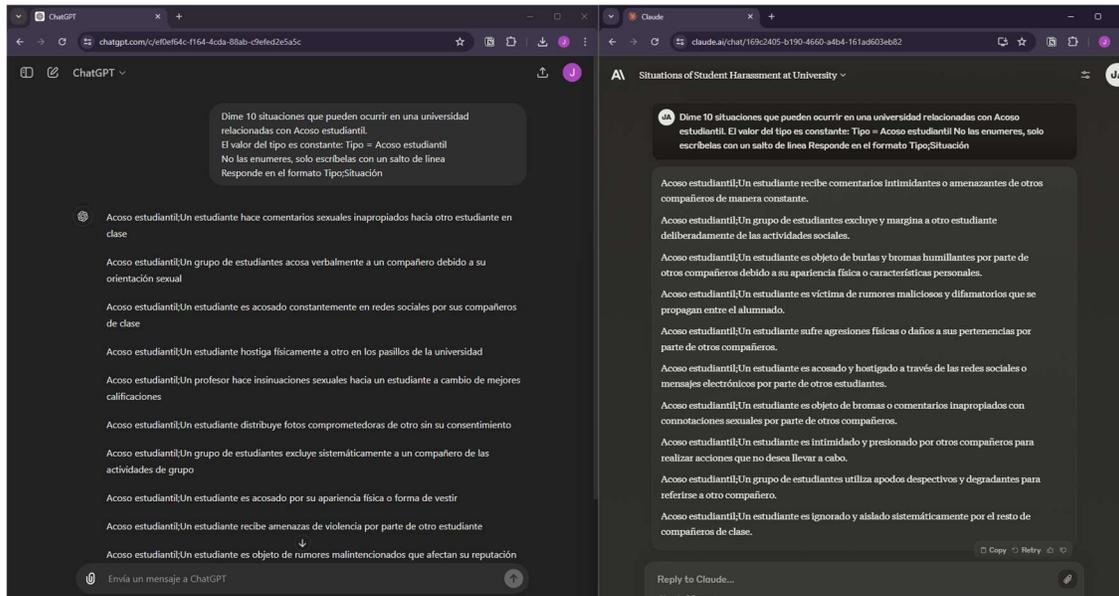
Utilizando de manera provechosa el poder de estos chatbots entrenados con LLMs para generación de texto, se implementa un *prompt* que solicita al chat 10 situaciones (para no sobrepasar la cantidad de tokens) relacionadas con un tema particular (variable), especificando que no se enumeren y que se escriban en el formato <tipo;situación>.

Dime 10 situaciones que pueden ocurrir en una universidad relacionadas con *<VARIABLE>*.

El valor del tipo es constante: Tipo = *<VARIABLE>*

No las enumeres, solo escríbelas con un salto de línea

Responde en formato *Tipo; Situación*



*Ilustración 11 Generación de información para dataset con ChatGPT y Claude.ai
Fuente: Elaboración propia*

Gracias a la ejecución de las aplicaciones utilizando los prompts planteados se consigue la información necesaria para poblar el dataset con 7783 registros de preguntas y situaciones relacionadas con el reglamento que serán utilizado para entrenar la red neuronal, en los cuales se incluyen situaciones y preguntas identificadas con su respectiva clase o etiqueta.

Procesamiento de los datos

Con los información organizada en el dataset, se da paso a la fase de procesamiento de los datos donde se realiza una limpieza y unificación de los datos obtenidos para definir las clases que va a predecir la red neuronal.

Las clases que se trabajarán se identifican de acuerdo con los contextos que indican los capítulos y títulos del *reglamento académico*, de igual forma se le asigna una etiqueta con el mismo nombre de las normas que estén dentro de la sección del reglamento que describe la clase.

Clases y etiquetas:

- **Faltas:** Describen situaciones relacionadas con inconvenientes que violen las normas del reglamento. Se asigna a normas que hablen sobre la sanción o especificación de actos que son prohibidos.
- **Información:** Se refiere a preguntas puntuales que hacen referencia a temas del reglamento, como el objetivo, misión social o políticas, etc. Es asignada a normas que se traten de los temas mencionados.
- **Procesos:** Hace referencia a preguntas puntuales acerca de procesos que se manejan en la universidad, como cambio de modalidad, homologaciones o sanciones. De igual forma, es asociada a las normas que hablen puntualmente de los casos mencionados.
- **Marco legal:** Describe situaciones o preguntas que hacen referencia a la vulneración de los derechos o deberes de los estudiantes. Se asigna a normas que se relacionen con los temas mencionados.
- **Ingreso:** Designa situaciones o preguntas relacionadas con el ingreso o admisión a la universidad. Es asignada a normas que se traten de los temas mencionados.
- **Solicitudes:** Hace referencia a situaciones o preguntas acerca de solicitudes o trámites académicos. De igual forma, es asociada a las normas que hablen puntualmente de los casos mencionados.
- **Programas:** Se refiere a las preguntas o situaciones relacionadas con características de los programas académicos. Se asigna a las normas que coincidan con las características.

Para la unificación de las clases se realiza un proceso manual para la unificación de estas utilizando el editor de código *Visual Studio Code*, el cual permite seleccionar y editar líneas de texto en paralelo, haciendo que el proceso de unificación de clases sea más sencillo y rápido.

En general, se realiza una selección múltiple de los tipos de situaciones o preguntas correspondientes a cada valor que se utilizó en los *prompts*, para reemplazarlo por la etiqueta de la clase a la cual corresponde la situación o pregunta.

```
dataset.csv M X
dataset > dataset.csv
1 v1;v2
2 Faltas;Un estudiante recibe mensajes amenazantes y hostiles de otro estudiante a través de redes sociales.
3 Faltas;Un grupo de estudiantes sigue a otro estudiante por el campus, intimidándolo y haciendo comentarios ofensivos.
4 Faltas;Un profesor hace comentarios de naturaleza sexual hacia sus estudiantes durante las clases.
5 Faltas;Un estudiante es constantemente objeto de burlas y bromas humillantes por parte de sus compañeros de clase.
6 Faltas;Un miembro del personal de la universidad utiliza su posición para presionar a estudiantes a participar en actividades sexuales no
7 Faltas;Un estudiante es excluido deliberadamente de actividades académicas y sociales debido a su género o identidad de género.
8 Faltas;Un grupo de estudiantes difunde rumores falsos sobre otro estudiante con el objetivo de dañar su reputación.
9 Faltas;Un estudiante escribe mensajes ofensivos y discriminatorios en los baños o paredes de la universidad dirigidos a otros estudiantes
10 Faltas;Un estudiante es objeto de acoso sistemático y continuo por parte de un grupo de compañeros debido a su orientación sexual o ident
11 Faltas;Un miembro del personal realiza avances sexuales no deseados hacia un estudiante y utiliza su poder para amenazarlo si se niega a
12 Faltas;Un estudiante es objeto de exclusión y ostracismo por parte de sus compañeros de clase debido a su origen socioeconómico.
13 Faltas;Un grupo de estudiantes ridiculiza y menosprecia constantemente a otro estudiante por su religión o creencias espirituales.
14 Faltas;Un estudiante es objeto de intimidación y agresión física por parte de otros estudiantes debido a su apariencia física o discapaci
15 Faltas;Un profesor utiliza su autoridad para menospreciar y denigrar públicamente a un estudiante en frente de sus compañeros.
16 Faltas;Un estudiante recibe correos electrónicos anónimos con amenazas de violencia física si continúa participando en ciertas actividad
17 Faltas;Un grupo de estudiantes crea un grupo en línea donde comparten imágenes sexualmente explícitas de otros estudiantes sin su consent
18 Faltas;Un miembro del personal de la universidad realiza comentarios racistas y xenófobos hacia estudiantes internacionales.
19 Faltas;Un estudiante es objeto de acoso sistemático y recurrente por parte de un grupo de compañeros de clase debido a su condición de in
20 Faltas;Un profesor hace comentarios despectivos y discriminatorios hacia estudiantes con discapacidades, afectando su autoestima y desemp
21 Faltas;Un estudiante es víctima de chantaje emocional por parte de otro estudiante, quien amenaza con difundir información privada si no
22 Faltas;Un estudiante es objeto de acoso por parte de un grupo de compañeros debido a su preferencia política, recibiendo insultos y amena
23 Faltas;Un miembro del personal de la universidad difunde rumores falsos sobre un estudiante con el fin de desacreditarlo y marginarlo.
24 Faltas;Un grupo de estudiantes crea un ambiente hostil en línea donde se burlan y ridiculizan a otros estudiantes a través de memes y com
```

Ilustración 12 Dataset de situaciones y clases

Fuente: Elaboración propia

Desarrollo de módulo de procesamiento de texto

Para la fase inicial del desarrollo de la aplicación se realiza la lógica que se encarga de procesar el texto de entrada de las situaciones, para ello se decide empezar por el módulo más complejo que tendrá el aplicativo. Este se compone de dos módulos clave para el procesamiento de texto, que en conjunto permiten analizar, comparar y responder a situaciones textuales de manera eficiente y precisa.

Módulo de Clasificación de Texto

El segundo módulo implementa un clasificador de texto que gestiona todo el flujo de trabajo desde la lectura de datos hasta la predicción de nuevas frases. Este módulo está representado por la clase *Classifier*.

En primera instancia, se instancia un objeto de la clase *Classifier* y se le envía la ruta del del dataset. Luego, se ejecuta el método *train* que se encarga de leer el archivo y procesar los datos. En este proceso se realiza la extracción de las columnas que estén definidas en el archivo, las cuales se renombran a "label" (etiqueta) y "text" (texto). Además, se realiza una división del conjunto de datos separándolo en dos grupos, uno de entrenamiento y uno de prueba.

El texto se vectoriza utilizando técnicas que convierten las palabras en una matriz de frecuencias. Con esta matriz, se entrena un modelo de clasificación como un clasificador lineal. Tras el entrenamiento, el modelo se evalúa en el conjunto de datos de prueba, calculando la precisión y generando un reporte de clasificación que incluye métricas como la precisión, la recuperación y la puntuación F1.

Los componentes entrenados se almacenan como atributos en la clase principal, para luego usarlos en la función `generate_predict` que utiliza el vectorizador y su función `transform` para vectorizar la situación o pregunta, y luego este resultado se envía como parámetro a la función `predict` del componente correspondiente al clasificador, el cual genera la predicción de la etiqueta que más se asimila al texto recibido.

Módulo de Procesamiento de Situaciones Textuales

El primer módulo se enfoca en procesar situaciones textuales para determinar respuestas pertinentes basadas en un conjunto de reglas y etiquetas predefinidas. Este módulo está representado por la clase *SituationProcessor*. Esta clase se configura inicialmente con un modelo de lenguaje, un conjunto de reglas y un conjunto de etiquetas, los cuales son fundamentales para su funcionamiento.

La clase permite cargar una situación textual, normalizándola a minúsculas y preparándola para su procesamiento posterior. El método principal de esta clase, denominado `get_rules`, convierte la situación en un documento y utiliza un emparejador de frases para identificar coincidencias significativas entre la situación y las reglas preestablecidas. Este método también calcula la similitud entre el texto de la situación y el contenido de las reglas, incorporando aquellas que presentan una alta similitud o un número suficiente de coincidencias. Además, se considera la etiqueta de predicción para añadir las reglas pertinentes de acuerdo con esta etiqueta.

Para lograr estos objetivos, *SituationProcessor* incluye varios métodos auxiliares:

- **Cálculo de Similitud:** Es un método que compara la similitud entre textos para determinar la relevancia de las reglas.
- **Extracción de Palabras Clave:** Identifica las palabras clave en la situación o pregunta y devuelve una lista de estas.
- **Limpieza de Texto:** Elimina caracteres no deseados y normalizan el texto.

Con la ejecución de la clase se puede generar una respuesta basada en la etiqueta de predicción generada. Además, se depuran las reglas analizando la similaridad de cada una de ellas con la situación o pregunta ingresada, lo cual permite identificar las reglas que tengan una similaridad y contexto parecidos al del texto ingresado.

Desarrollo de backend

La aplicación backend es el componente donde se maneja la mayor parte de la lógica de datos de la aplicación, es el encargado de traer la información almacenada en la base de datos y realizar el procesamiento del texto de entrada.

Para recibir las peticiones de información se implementa una serie de endpoints con tareas específicas (*Ilustración 13*) las cuales son expuestas en un servidor local. Su

funcionamiento se basa en cargar primeramente los modelos y configuración que requieren las librerías (Spacy, Torch, Transformer), para luego crear la conexión a la base de datos y por último exponer los endpoint de la API en el puerto designado.

El endpoint encargado de la procesar la situación o pregunta ingresada por el usuario es *api/evaluate* el cual trabaja de la mano con el módulo de procesamiento de texto. Para la ejecución de esta lógica se crea una instancia de la clase *Classifier* la cual devuelve una predicción indicando la etiqueta a la cual pertenece el texto de entrada, con esta información cargada se procede a crear una instancia de la clase *SituationProcessor* y cargar los datos correspondientes al texto ingresado y la predicción anteriormente creada. Esta clase genera las reglas relacionadas y la respuesta a la pregunta o situación, las cuales son enviadas en el cuerpo de la respuesta.

Los demás endpoints se encargan de generar consultas o inserciones de información en la base de datos.

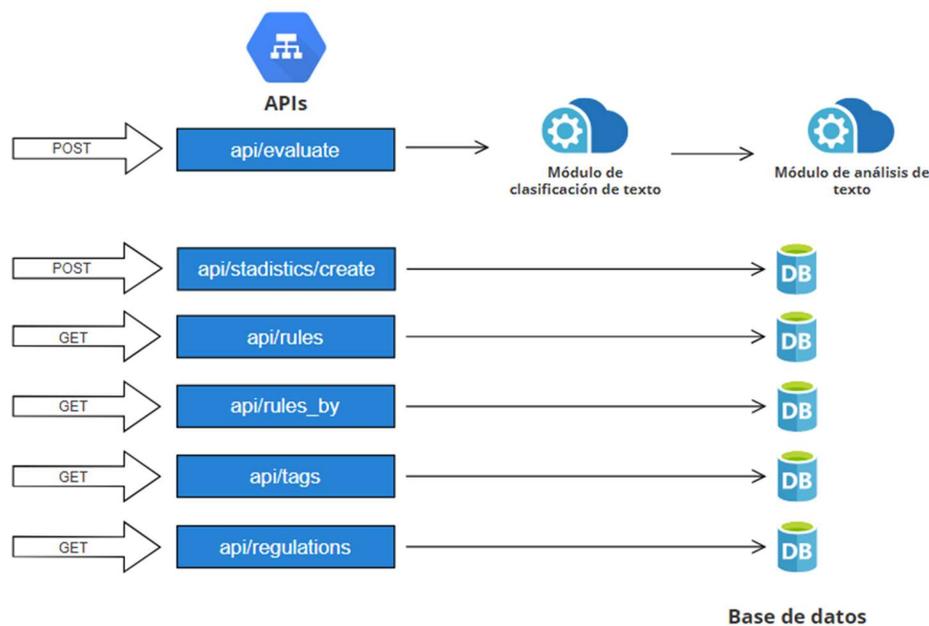


Ilustración 13 Estructura de APIs
Fuente: Elaboración propia

Desarrollo de frontend

La parte de interfaz de usuario de la aplicación se maneja desde la aplicación frontend, este módulo está construido para mostrar la información al usuario final y brindar una experiencia amigable. Para ello se crearon 5 vistas que permiten consultar la información de normas y reglamentos, al igual que realizar la búsqueda por situación o pregunta. Se implementa el modelo de aplicación de una sola página (SPA) para mostrar una sola página HTML y utilizar JavaScript para actualizar el contenido de la página de forma dinámica sin necesidad de recargar completamente el navegador.

Además de mostrar las vistas correspondientes para cada sección de la aplicación, se hace uso de un controlador, el cual es el encargado de comunicarse directamente con la aplicación backend para obtener la información correspondiente consumiendo los endpoints expuestos.

Esta estructura utilizada permite que la aplicación sea liviana y fácil de ejecutar en cualquier navegador web, solo accediendo desde la URL definida, que para este caso es el localhost en el puerto 3000.



Ilustración 14 Pantalla de aplicación frontend Home
Fuente: Elaboración propia



Ilustración 15 Pantalla de aplicación frontend Búsqueda por reglamentos
Fuente: Elaboración propia

Pruebas y finalización

Para verificar el funcionamiento de la aplicación se aplican una serie de pruebas a la plataforma, en primera instancia se verifica el funcionamiento de la integración de cada módulo para verificar las conexiones y que el transporte de la información sea el adecuado. Con esto se pasa a realizar la verificación de los requisitos funcionales y no funcionales a través de pruebas manuales realizadas en el ambiente local donde se ejecuta la aplicación. Por último, se realizan pruebas de aceptación con el usuario final (estudiantes) para verificar el funcionamiento de la aplicación y la evaluación de este mismo con base en criterios de usabilidad, rendimiento, precisión, utilidad, amigabilidad y uso futuro. (Anexo 4)

9. RESULTADOS

Para la evaluación de la solución desarrollada se crea una encuesta para puntuar los criterios establecidos para medir el impacto de la solución. Para ello, se toma como referencia el método realizado en la investigación documentada en el artículo *A Chatbot Framework for the Children's Legal Centre*, en el cual se realiza encuesta a 14 participantes que interactuaron con el chatbot. Las respuestas se clasifican del 1 (muy en desacuerdo/no) al 5 (muy de acuerdo/sí). Con base en esto, se crea una encuesta similar que ayudará a medir los criterios establecidos en el inicio del documento.

1. ¿Crees que el Poli-Consejero es una aplicación fácil de usar?
2. En términos de rendimiento, ¿Crees que la aplicación fue rápida en su respuesta?
3. ¿Crees que la precisión de la respuesta y normas relacionadas que brindó la aplicación fue buena?
4. ¿Crees que la aplicación es útil para mejorar ?¿Crees que el Poli-Consejero es útil para mejorar el acceso a las normas y reglamentos?
5. ¿Consideras que la aplicación tiene una interfaz de usuario amigable?
6. ¿Volverías a usar el Poli-Consejero en el futuro para resolver inquietudes o situaciones con respecto al reglamento?

Para el desarrollo de las pruebas se realizan pruebas funcionales de la aplicación a 13 estudiantes aleatorios pertenecientes a la comunidad del Politécnico Grancolombiano. La ejecución de las pruebas se realiza en un ambiente de desarrollo local y cada individuo de la muestra hace uso del equipo utilizado para ejecutar la aplicación. Cada participante tiene el libre albedrío de utilizar la aplicación a su conciencia y entendimiento, para ello solo se le indica que lea las instrucciones de uso que brinda el módulo de ayuda de la aplicación. Paso siguiente, el estudiante deberá diligenciar la encuesta generada por la aplicación y enviar el resultado.

Las respuestas almacenadas fueron analizadas y se toma la puntuación promedio de todos los participantes (tabla 2). Con base en estos resultados es considerable afirmar que:

- La facilidad de uso muestra el grado de dificultad para interactuar con la aplicación, la cual brinda una interfaz sencilla.
- El rendimiento no es del todo satisfactorio, ya que las limitaciones computacionales impiden que la respuesta se genere en un periodo corto de tiempo.
- La precisión de la respuesta y las normas generadas es bastante buena, lo que indica que la situación o pregunta ingresada tiene una relación cercana con las normas y la respuesta entregada por la aplicación
- La utilidad muestra que es una herramienta práctica para mejorar el acceso a los reglamentos y normar según la opinión de los estudiantes.

- El criterio de amigabilidad muestra que la aplicación es bastante agradable para los usuarios.
- El uso futuro o repetitivo de la aplicación es considerable en gran medida por los estudiantes.

Criterio	Promedio
Usabilidad	4.15
Rendimiento	3.69
Precisión	4.08
Utilidad	4.62
Amigable	4.46
Uso futuro	4.69

Tabla 2 Evaluación de la aplicación de acuerdo con estudiantes
Fuente: Elaboración propia

Conclusiones

Teniendo en cuenta el proceso de implementación de la herramienta realizada se puede concluir varios puntos clave frente a este desarrollo. En primera instancia, el desarrollo de la aplicación confirmó el uso efectivo de tecnologías que utilizan inteligencia artificial para procesar y clasificar texto normativo, la capacidad de los modelos para entender el contexto de las situaciones o preguntas recibidas, y su efectividad para proporcionar respuestas relacionadas, demostró ser una solución viable para mejorar el acceso y comprensión de las normas del reglamento académico.

En el mismo contexto, las evaluaciones realizadas con usuarios finales revelaron que la aplicación es intuitiva y fácil de usar, y mejora significativamente la eficiencia en la resolución de dudas o inquietudes del ámbito normativo. Además, manifestaron una alta satisfacción con la interfaz y funcionabilidad de la aplicación, donde se destaca la precisión de las respuestas proporcionadas. La aplicación tiene el potencial de reducir la carga de trabajo del personal administrativo, ya que los estudiantes pudieron resolver sus consultas de manera autónoma.

Por otra parte, se puede concluir que la metodología implementada permitió identificar de manera precisa las necesidades y desafíos de los usuarios, lo cual fue fundamental para el desarrollo de la herramienta. Además, los prototipos iniciales permitieron identificar y resolver problemas de usabilidad, precisión y funcionalidad en las primeras etapas del desarrollo, el uso de tecnologías avanzadas de procesamiento de lenguaje natural y LLMs garantizó que la aplicación pudiera manejar consultas complejas con precisión.

Sin embargo, las pruebas también dejan áreas de mejora. Algunas consultas complejas no fueron respondidas con la precisión deseada lo que sugiere la necesidad de un entrenamiento adicional del modelo con un conjunto de datos más amplio y diverso. Así como el desarrollo de un método de programación que reduzca el tiempo del análisis de la situación y la generación de la respuesta, o la implementación de recursos de hardware más sofisticados y con más capacidad de procesamiento.

En cuanto a la arquitectura de la aplicación web, se evidencia como un diseño ligero y eficiente que soporta adecuadamente las necesidades del proyecto. La elección de una arquitectura modular que separa el frontend del backend y los componentes de procesamiento de lenguaje natural, ha permitido una implementación flexible y escalable.

Trabajo futuro

La realización de la herramienta, las pruebas realizadas y las conclusiones analizadas en este documento dejan resultados satisfactorios que indican el éxito del proyecto, sin embargo, se plantea una serie de mejoras a la aplicación para aumentar su rendimiento, velocidad y precisión.

Primero, para mejorar el rendimiento de la red neuronal se plantea aumentar significativamente el volumen del dataset. Esto permitirá al modelo de inteligencia artificial tener una mayor diversidad y cantidad de ejemplos para aprender, lo que incrementará su precisión y capacidad de manejar consultas más complejas.

Se plantea implementar programación paralela para reducir el tiempo de respuesta de la aplicación. El uso de esta técnica permitirá procesar múltiples consultas de manera simultánea, mejorando la eficiencia del sistema y proporcionando respuestas más rápidas a los usuarios.

La implementación de aprendizaje no supervisado para que el modelo pueda incrementar su conocimiento por sí solo con base en las situaciones que ingresan los usuarios al utilizar la aplicación. Sin embargo, este proceso requiere de la implementación del almacenamiento de las situaciones o preguntas ingresadas por los usuarios, por lo cual se debe tener suma precaución debido a que los usuarios pueden ingresar información privada o confidencial la cual si no se trata debidamente puede llegar a introducirse en el conocimiento del modelo, para ello se sugiere implementar técnicas de anonimización de datos y realizar un módulo que informe al usuario final sobre cómo será tratada la información que ingrese.

Por último, sería ideal que el Politécnico Grancolombiano evalúe la implementación de la herramienta desarrollada para mejorar los procesos de asesoramiento y comunicación con los estudiantes.

10. BIBLIOGRAFÍA

- Nataraj, P., Devaraj, Teja, R., Kumar, M., & Gangrade, A. (2023). Development of a Legal Document AI-Chatbot. ArXiv, abs/2311.12719.
- Queudot, M., Charton, E., & Meurs, M. (2020). Improving Access to Justice with Legal Chatbots. Stats.
- Cui, J., Li, Z., Yan, Y., Chen, B., & Yuan, L. (2023). Chatlaw: A Multi-Agent Collaborative Legal Assistant with Knowledge Graph Enhanced Mixture-of-Experts Large Language Model.
- Han, H. (2023). Potential Benefits of Employing Large Language Models in Research in Moral Education and Development. ArXiv, abs/2306.13805.
- Morgan, J., Paiement, A., Seisenberger, M., Williams, J., & Wyner, A.Z. (2018). A Chatbot Framework for the Children's Legal Centre. International Conference on Legal Knowledge and Information Systems.
- Ng, J., Haller, E., & Murray, A. (2022). The ethical chatbot: A viable solution to socio-legal issues. *Alternative Law Journal*, 47, 308 - 313.
<https://doi.org/10.1177/1037969X221113598>.
<https://researchers.cdu.edu.au/en/publications/the-ethical-chatbot-a-viable-solution-to-socio-legal-issues>
- Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. North American Chapter of the Association for Computational Linguistics.
- Abdulwahhab, R., Makhmari, H., & Battashi, S. (2015). An educational web application for academic advising. 2015 IEEE 8th GCC Conference & Exhibition, 1-6. <https://doi.org/10.1109/IEEEGCC.2015.7060084>.
- Aly, W., Eskaf, K., & Selim, A. (2017). Fuzzy mobile expert system for academic advising. 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), 1-5. <https://doi.org/10.1109/CCECE.2017.7946846>.
- Arsys (2020) Clasificación de textos con Python y Jupyter Notebooks [Código fuente]. Arsys <https://www.arsys.es/blog/clasificaciontextos-python-jupyternotebooks>
- Church, K.W., Chen, Z., & Ma, Y. (2021). Emerging trends: A gentle introduction to fine-tuning. *Natural Language Engineering*, 27, 763 - 778.
- Naveed, H., Khan, A.U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Barnes, N., & Mian, A.S. (2023). A Comprehensive Overview of Large Language Models. ArXiv, abs/2307.06435.
- Mannadiar, N., & Gürsoy, K. (2019). Neural Networks for Text Classification.
- Minace, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2020). Deep Learning--based Text Classification. *ACM Computing Surveys (CSUR)*, 54, 1 - 40.
- Попова, Е., Ророва, Е., Спицын, В., Спичун, В., Иванова, Ю., & Иванова, Ю. (2019). Using artificial neural networks to solve text classification problems. *GraphiCon'2019 Proceedings*. Volume 1.

- Sun, Z. (2023). A Short Survey of Viewing Large Language Models in Legal Aspect. ArXiv, abs/2303.09136.
- Hmeidi, I., Al-Ayyoub, M., Abdulla, N., Almodawar, A., Abooraig, R., & Mahyoub, N. (2015). Automatic Arabic text categorization: A comprehensive comparative study. *Journal of Information Science*, 41, 114 - 124.
- Zhang, X., Zhao, J.J., & LeCun, Y. (2015). Character-level Convolutional Networks for Text Classification. *Neural Information Processing Systems*.
- Liddy, E.D., Paik, W., & Yu, E.S. (1994). Text categorization for multiple users based on semantic features from a machine-readable dictionary. *ACM Trans. Inf. Syst.*, 12, 278-295.
- Kamruzzaman, S. (2010). Text Classification using Artificial Intelligence. ArXiv, abs/1009.4964.
- Howard, J., & Ruder, S. (2018). Universal Language Model Fine-tuning for Text Classification. *Annual Meeting of the Association for Computational Linguistics*.
- Qasem, R., Tantour, B., & Maree, M. (2023). Towards the Exploitation of LLM-based Chatbot for Providing Legal Support to Palestinian Cooperatives. *ArXiv, abs/2306.05827*.
<https://www.mckinsey.com/capabilities/quantumblack/our-insights/global-survey-the-state-of-ai-in-2020?cid=other-soc-lkn-mip-mck-oth---&sid=4255306818&linkId=105722332>
- Minace, S., Cambria, E., & Gao, J. (2020). Deep Learning--based Text Classification. *ACM Computing Surveys (CSUR)*, 54, 1 - 40.
- Dande, A.A., & Pund, D.M. (2023). A Review Study on Applications of Natural Language Processing. *International Journal of Scientific Research in Science, Engineering and Technology*.
- Vanita (2024). An Extant of Natural Language Processing. *International Journal For Multidisciplinary Research*.
- Yadav, R.K., Madaan, A., & Janu (2024). Comprehensive analysis of natural language processing. *Global Journal of Engineering and Technology Advances*.
- Cambria, E., & White, B. (2014). Jumping NLP Curves: A Review of Natural Language Processing Research [Review Article]. *IEEE Computational Intelligence Magazine*, 9, 48-57.
- G. Suganya, R. Porkodi (2020). Prediction and Analysis of Extracting Relations using Spacy Model. *International Journal of Recent Technology and Engineering*.
- Yao, L., Mao, C., & Luo, Y. (2018). Graph Convolutional Networks for Text Classification. ArXiv, abs/1809.05679.
- Delaforge, A., Azé, J., Bringay, S., Mollevi, C., Sallaberry, A., & Servajean, M. (2022). EBBE-Text: Explaining Neural Networks by Exploring Text Classification Decision Boundaries. *IEEE Transactions on Visualization and Computer Graphics*, 29, 4154-4171.

- Gutiérrez, F., Seipp, K., Ochoa, X., Chiluzza, K., Laet, T.D., & Verbert, K. (2020). LADA: A learning analytics dashboard for academic advising. *Comput. Hum. Behav.*, 107, 105826.
- Alkhoori, A., Kuhail, M.A., & Alkhoori, A. (2020). UniBud: A Virtual Academic Adviser. 2020 12th Annual Undergraduate Research Conference on Applied Computing (URC), 1-4.

11. ANEXOS

- Anexo 1. Entregable 1 Análisis del problema y diseño de la solución

Encuestas:

ID	Hora de inicio	Hora de finalización	Rol dentro de la encuesta	Carrera estudiada	Universidad	Rango de edad	Lo estás con los reglamentos	Lo entiendes hasta más o menos	Capacidad para resolver	Símbolos de información	¿Sobre las reglas, normas o procedimientos?	¿Deberían explicarse más a los estudiantes las inducciones el manejo y la actualización del reglamento?	¿Formas, indicando cuáles?	
1	2-23-24 15:43:10	2-23-24 15:45:40	Estudiante		0	1	De 18 a 23 años	Poco familiarizado, he leído algunas partes	No es del todo claro, no comprendí algunas cosas	No	De 1 a 5 veces	Un poco complicado	Deberían explicarse más a los estudiantes las inducciones el manejo y la actualización del reglamento.	Si
5	2-23-24 15:47:10	2-23-24 15:49:32	Estudiante	Adm hotelera y gastronomía		2	De 18 a 23 años	Poco familiarizado, he leído algunas partes	No es del todo claro, no comprendí algunas cosas	No	0 veces	Un poco complicado	Deberían explicarse más a los estudiantes las inducciones el manejo y la actualización del reglamento.	Si
7	2-23-24 15:49:03	2-23-24 15:51:15	Estudiante	Ingeniería de Telecomunicaciones		1	De 18 a 23 años	No estoy familiarizado, no lo he leído	Muy confuso, no entendí nada	No	0 veces	Fácil	Brindar las reglas, normas de una manera sencilla	Si
10	2-23-24 15:53:41	2-23-24 15:58:29	Estudiante	Ingeniería de Sistemas		0.5	De 18 a 23 años	Poco familiarizado, he leído algunas partes	Muy claro y fácil de entender	Si	0 veces	Fácil	Ninguno	Si
14	2-23-24 15:58:05	2-23-24 15:59:59	Estudiante	Ingeniería Industrial		1	De 18 a 23 años	Poco familiarizado, he leído algunas partes	Muy claro y fácil de entender	Si	De 1 a 5 veces	Muy fácil	No	Si
17	2-23-24 16:05:20	2-23-24 16:06:54	Estudiante	Comunicación social		1	De 18 a 23 años	No estoy familiarizado, no lo he leído	No es del todo claro, no comprendí algunas cosas	No	De 1 a 5 veces	Un poco complicado	Mas visibilidad	Si
19	2-23-24 16:09:18	2-23-24 16:10:20	Estudiante	Mercadeo y publicidad		2	De 18 a 23 años	No estoy familiarizado, no lo he leído	No es del todo claro, no comprendí algunas cosas	Si	0 veces	Un poco complicado	Ninguna	Si
20	2-23-24 16:12:52	2-23-24 16:14:57	Administrativo			1	De 24 a 30 años	Poco familiarizado, he leído algunas partes	No es del todo claro, no comprendí algunas cosas	No	0 veces	Fácil	Deberían publicar y hacer más visible el reglamento en páginas videos	Si
24	2-23-24 16:17:31	2-23-24 16:32:00	Administrativo			1	De 24 a 30 años	Muy familiarizado, lo he leído completamente	Muy claro y fácil de entender	Si	+10 veces	Fácil	No	Si
26	2-23-24 17:51:11	2-23-24 17:52:39	Docente			13	De 31 a 40 años	Bastante familiarizado, lo he leído en su mayor parte	Muy claro y fácil de entender	Si	0 veces	Fácil	Ninguna	Si
33	2-26-24 11:09:14	2-26-24 11:13:41	Docente			9	Mayor de 40 años	Bastante familiarizado, lo he leído en su mayor parte	Muy claro y fácil de entender	Si	0 veces	Muy fácil	A veces están desactualizados o no se sabe cuál es el que se aplica directamente en el momento de consultar.	Si
36	2-26-24 18:33:09	2-26-24 18:35:30	Docente			5	Mayor de 40 años	Bastante familiarizado, lo he leído en su mayor parte	Muy claro y fácil de entender	Si	0 veces	Fácil	No	No es de mi importancia
37	2-26-24 18:35:13	2-26-24 18:36:11	Docente			9	Mayor de 40 años	Bastante familiarizado, lo he leído en su mayor parte	Muy claro y fácil de entender	Si	0 veces	Muy fácil	Ninguna	Si
38	2-26-24 18:39:53	2-26-24 18:41:08	Estudiante	Ingeniería en sistemas		3	De 18 a 23 años	Poco familiarizado, he leído algunas partes	Muy claro y fácil de entender	No	De 6 a 10 veces	Un poco complicado	Trabajan a conveniencia de la institución y no velan por el bienestar del estudiante.	Si
39	2-26-24 18:39:52	2-26-24 18:42:01	Estudiante	Ingeniería de sistemas		2	De 18 a 23 años	Poco familiarizado, he leído algunas partes	No es del todo claro, no comprendí algunas cosas	Si	De 1 a 5 veces	Fácil	sería más rápido y eficiente las respuestas de los docentes.	Si
44	2-26-24 18:45:07	2-26-24 18:47:06	Estudiante	Ingeniería de sistemas		5	De 18 a 23 años	Poco familiarizado, he leído algunas partes	No es del todo claro, no comprendí algunas cosas	Si	0 veces	Un poco complicado	No	Si
45	2-26-24 18:44:06	2-26-24 18:47:46	Estudiante	Ingeniería de Sistemas		2.5	De 18 a 23 años	No estoy familiarizado, no lo he leído.	Muy confuso, no entendí nada	Si	0 veces	Un poco complicado	sería una manera fácil de consultar algo en específico, que sea clara.	Si
49	2-29-24 11:14:22	2-29-24 11:25:57	Docente			9	De 31 a 40 años	Bastante familiarizado, lo he leído en su mayor parte	No es del todo claro, no comprendí algunas cosas	Si	0 veces	Muy fácil	ninguno	Si
50	2-29-24 12:04:57	2-29-24 12:05:56	Docente			2	Mayor de 40 años	Bastante familiarizado, lo he leído en su mayor parte	Muy claro y fácil de entender	Si	0 veces	Muy fácil	Ninguno	Si
51	2-29-24 20:14:11	2-29-24 20:17:01	Docente			12	Mayor de 40 años	Bastante familiarizado, lo he leído en su mayor parte	Muy claro y fácil de entender	Si	0 veces	Fácil	Hay reglamentos generales como el académico, de profesores, que son de uso común y se difunden bastante, otras normas y procedimientos más específicos pueden ser desconocidos, y sería interesante un catálogo de para conocer que está plenamente reglamentado. Existe Kawak pero algo en formato gráfico puede resultar interesante.	Si
52	3-1-24 9:10:13	3-1-24 9:12:54	Docente			6	Mayor de 40 años	Bastante familiarizado, lo he leído en su mayor parte	No es del todo claro, no comprendí algunas cosas	Si	0 veces	Un poco complicado	El manejo de los casos no se acomoda en ocasiones con los reglamentos. La institución podría tener un repositorio de preguntas frecuentes o un foro para socializar inquietudes.	Si
53	3-1-24 11:32:12	3-1-24 11:34:06	Docente			6	De 31 a 40 años	Bastante familiarizado, lo he leído en su mayor parte	Muy claro y fácil de entender	Si	0 veces	Muy fácil	Ninguno	Si
54	3-2-24 8:19:01	3-2-24 8:20:40	Docente			5	De 31 a 40 años	Bastante familiarizado, lo he leído en su mayor parte	Muy claro y fácil de entender	Si	0 veces	Fácil	No	Si

Casos de uso:

Identificador	CU-U-001	
Nombre	Realizar búsqueda	
Descripción	El usuario ingresa una situación específica o palabra clave en la caja de búsqueda y da click en el botón consultar	
Rol	Usuario	
Pre-Condiciones	El usuario debe haber aceptado los términos y condiciones (CU-U-001)	
Secuencia	Paso	Acción
	1	El usuario escribe una situación o palabra clave en la caja de búsqueda
	2	La aplicación muestra un preloader
	3	La aplicación muestra el resultado de la búsqueda
Post-Condiciones	La aplicación analiza la situación y devuelve las reglas relacionadas con la situación.	
Excepciones	Paso	Acción
	3.1	El sistema muestra un error si no hay conexión a internet

Identificador	CU-U-002	
Nombre	Calificar una respuesta	
Descripción	El usuario da click en el botón "¿Fue útil esta respuesta?", diligencia la encuesta y da click en el botón enviar.	
Rol	Usuario	
Pre-Condiciones	El usuario debe haber consultado una situación (CU-U-002)	
Secuencia	Paso	Acción
	1	El usuario da click en el botón "¿Fue útil esta respuesta?"
	2	El sistema muestra un pop-up con una encuesta de satisfacción.
	3	El usuario diligencia los campos de la encuesta.
	4	El usuario da click en el botón "Enviar".
	5	El sistema muestra un mensaje indicando que la encuesta fue registrada satisfactoriamente.
Post-Condiciones		
Excepciones	Paso	Acción
	5.1	El sistema muestra un error si no hay conexión a internet
	4.1	El usuario cierra el pop-up de la encuesta

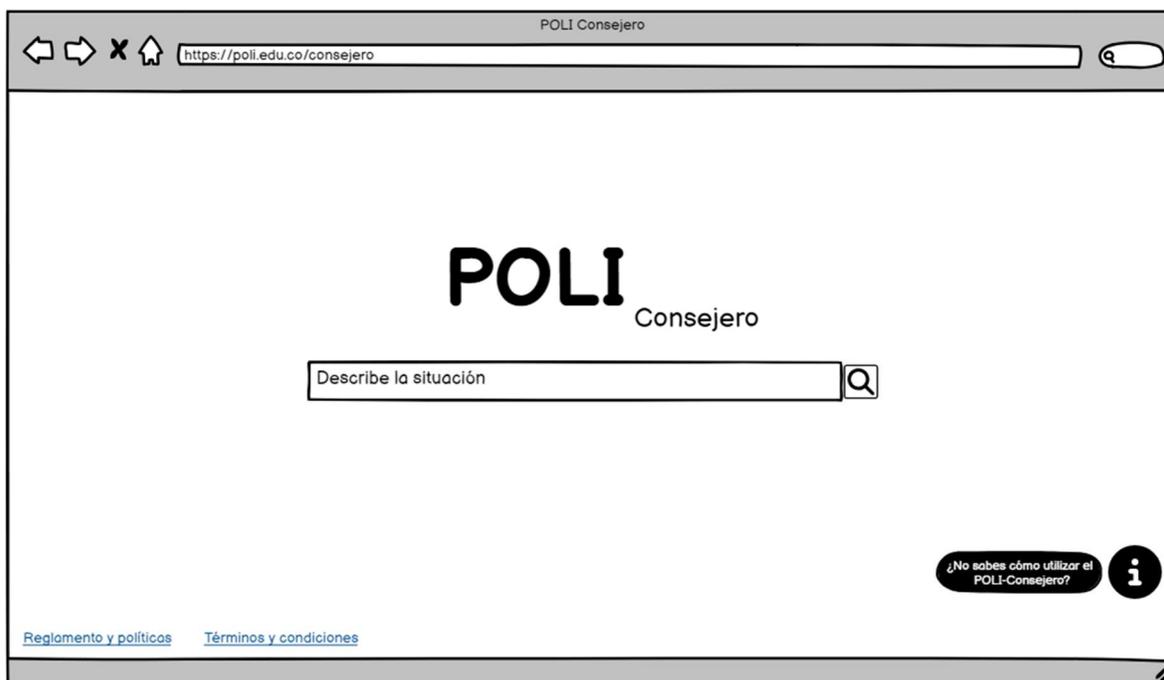
Identificador	CU-U-003	
Nombre	Ver reglamento	
Descripción	El usuario da click en el botón de un reglamento	
Rol	Usuario	
Pre-Condiciones	* El usuario debe haber ingresado a la aplicación * El usuario debe haber consultado una situación (CU-U-002).	
Secuencia	Paso	Acción
	1	El usuario da click en el botón de algún reglamento
	2	La aplicación muestra una vista donde muestra las reglas pertenecientes al reglamento seleccionado.
Post-Condiciones	El sistema muestra una vista con las reglas y etiquetas relacionadas al reglamento.	
Excepciones	Paso	Acción
	1.1	El sistema muestra un error si no hay conexión a internet

Identificador	CU-U-004	
Nombre	Ver reglas por etiqueta	
Descripción	El usuario da click en el botón de una etiqueta	
Rol	Usuario	
Pre-Condiciones	* El usuario debe haber consultado una situación (CU-U-002). * El usuario debe haber entrado a la vista de un reglamento (CU-U-006).	
Secuencia	Paso	Acción
	1	El usuario da click en el botón de una etiqueta.
	2	La aplicación muestra una vista donde muestra las reglas relacionadas con la etiqueta seleccionada.
Post-Condiciones	El sistema muestra una vista con las reglas y etiquetas relacionadas a la etiqueta.	
Excepciones	Paso	Acción
	1.1	El sistema muestra un error si no hay conexión a internet

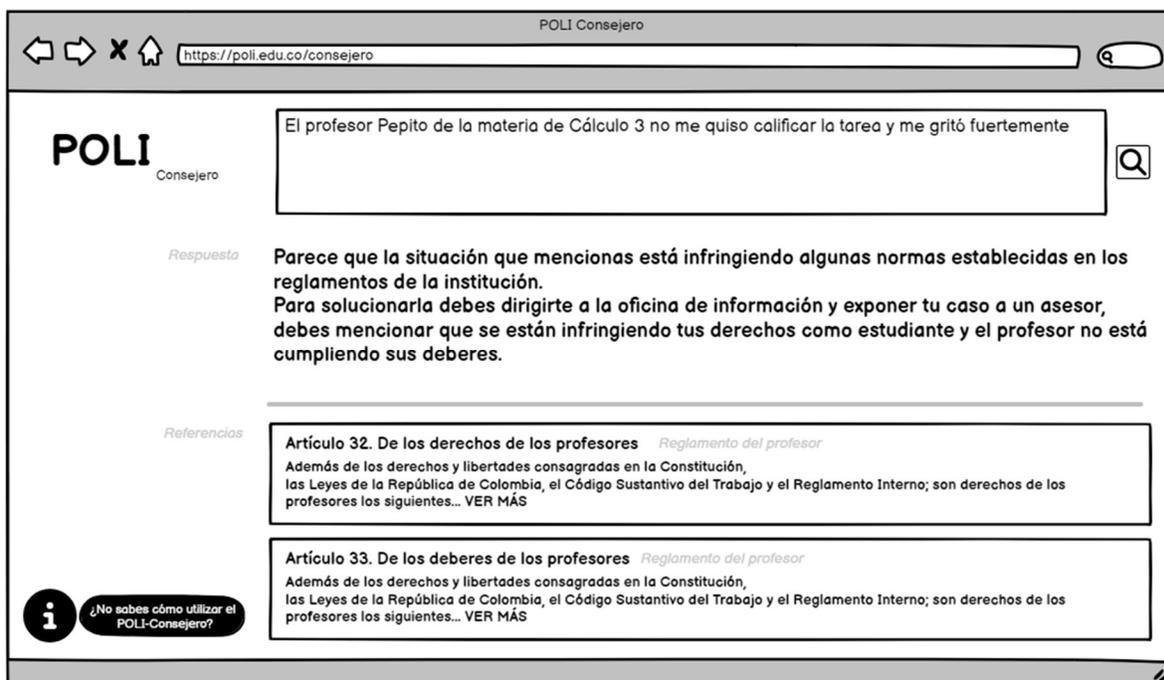
Identificador	CU-U-005	
Nombre	Ver sección de ayuda y tutorial	
Descripción	El usuario da click en el botón ayuda o en el botón "¿No sabes cómo utilizar el PoliConsejero?"	
Rol	Usuario	
Pre-Condiciones	El usuario debe haber ingresado a la aplicación	
Secuencia	Paso	Acción
	1	El usuario da click en el botón ayuda o en el botón "¿No sabes cómo utilizar el PoliConsejero?".
	2	La aplicación muestra una vista donde muestra unos consejos y tutoriales para utilizar la aplicación.
Post-Condiciones	El sistema muestra una vista con consejos y tutoriales para utilizar la aplicación.	
Excepciones	Paso	Acción
	1.1	El sistema muestra un error si no hay conexión a internet

Mockups

Inicio



Búsqueda por situación o pregunta



Búsqueda por etiquetas

POLI Consejero

https://poli.edu.co/consejero

Etiquetas

Información Faltas Procesos

Información

Artículo 32. De los derechos de los profesores *Reglamento del profesor*

Además de los derechos y libertades consagradas en la Constitución, las Leyes de la República de Colombia, el Código Sustantivo del Trabajo y el Reglamento Interno; son derechos de los profesores los siguientes... VER MÁS

Información

Artículo 33. De los deberes de los profesores *Reglamento del profesor*

Además de los derechos y libertades consagradas en la Constitución, las Leyes de la República de Colombia, el Código Sustantivo del Trabajo y el Reglamento Interno; son deberes de los profesores los siguientes... VER MÁS

Información

¿No sabes cómo utilizar el POLI-Consejero?



Búsqueda por reglamento

POLI Consejero

https://poli.edu.co/consejero

Reglamentos

Reglamento académico Reglamento del profesor

Reglamento del profesor

Artículo 32. De los derechos de los profesores *Reglamento del profesor*

Además de los derechos y libertades consagradas en la Constitución, las Leyes de la República de Colombia, el Código Sustantivo del Trabajo y el Reglamento Interno; son derechos de los profesores los siguientes... VER MÁS

Información

Artículo 32. De los derechos de los profesores *Reglamento del profesor*

Además de los derechos y libertades consagradas en la Constitución, las Leyes de la República de Colombia, el Código Sustantivo del Trabajo y el Reglamento Interno; son deberes de los profesores los siguientes... VER MÁS

Información

¿No sabes cómo utilizar el POLI-Consejero?



- Anexo 2. Entregable 4 Documentación pruebas

Pruebas de medición:

usabilidad	rendimiento	precision	utilidad	amigable	uso_futuro
4	3	4	5	5	5
5	5	5	4	5	5
3	3	4	4	5	5
4	3	4	5	5	5
4	3	4	5	5	5
5	5	4	5	4	5
5	4	5	4	4	5
4	5	4	5	4	5
3	2	2	3	3	1
4	4	4	5	4	5
4	3	5	5	4	5
4	3	4	5	5	5
5	5	4	5	5	5
4,15	3,69	4,08	4,62	4,46	4,69