Universidad Politécnica
de Madrid

**Escuela Técnica Superior de
Ingenieros Informáticos**

Máster en Data Science

Trabajo Fin de Máster

# Sistema Generador y Evaluador Automático de Cuestionarios con Control del Nivel de Dificultad en Español Basado en Modelos de Lenguaje de Gran Escala

Autor: Paul Andree Eyzaguirre Barreda

Madrid, Junio, 2024

Este Trabajo Fin de Máster se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid.

*Trabajo Fin de Máster*
*Máster en* Data Science

*Título:* Sistema Generador y Evaluador Automático de Cuestionarios con Control del Nivel de Dificultad en Español Basado en Modelos de Lenguaje de Gran Escala
Junio, 2024

*Autor:* Paul Andree Eyzaguirre Barreda
*Tutor:* Carlos Badenes-Olmedo
Departamento de Sistemas Informáticos
ETSI Informáticos
Universidad Politécnica de Madrid

# Resumen

En este Trabajo de fin de Máster (TFM) presentamos un sistema innovador de generación y evaluación automática de cuestionarios (SGEC) capaz de generar cuestionarios controlando el nivel de dificultad en preguntas de opción múltiple y preguntas abiertas. El proceso se realiza sin intervención humana mediante técnicas de procesamiento del lenguaje natural en combinación con modelos de lenguaje de gran escala y aprendizaje contextual.

La iniciativa surge ante la necesidad de optimizar la creación de cuestionarios educativos en idioma Castellano, tarea que tradicionalmente consume tiempo y es susceptible a la subjetividad evaluadora del docente. Nuestro sistema no solo genera cuestionarios adaptados a tres niveles de dificultad, sino que también permite la autoevaluación, promoviendo así la autonomía y adaptabilidad en el aprendizaje. El sistema, desarrollado como servicio online basado en modelos de lenguaje, aborda varios desafíos específicos del ámbito educativo de habla hispana. Entre estos se incluyen la generación de cuestionarios personalizados y la adaptación de la dificultad de las preguntas según el nivel de conocimiento de los estudiantes.

Los resultados de las encuestas realizadas para el sistema SGEC muestran una alineación adecuada con las categorías de dificultad propuestas. Además, los resultados de las evaluaciones muestran que el SGEC genera preguntas semánticamente correctas, sintácticamente precisas, y contextualmente relevantes. También se obtuvo una retroalimentación detallada y precisa sobre las respuestas de los estudiantes con el sistema de evaluación automática en preguntas abiertas.

Finalmente, se revisan algunas limitaciones tecnológicas y de integración, y se proponen nuevas líneas de investigación para la mejora del sistema presentado.

# Abstract

In this Master's Thesis (TFM) we present an innovative system for the automatic generation and evaluation of questionnaires (SGEC) capable of generating questionnaires controlling the level of difficulty in multiple choice questions and open questions. The process is performed without human intervention using natural language processing techniques in combination with large language models and fine tuning at the instruction level.

The initiative arises from the need to optimize the creation of educational questionnaires, a task that traditionally consumes time and is susceptible to the teacher's evaluative subjectivity. Our system not only generates questionnaires adapted to three levels of difficulty, but also allows self-evaluation, thus promoting autonomy and adaptability in learning. The system, developed as an online service based on language models, addresses several challenges in the Spanish-speaking educational field. These include generating personalized quizzes and adapting the difficulty of questions according to the level of knowledge of the students.

The results of the surveys carried out for the SGEC system show adequate alignment with the proposed difficulty categories. Furthermore, the results of the evaluations show that the SGEC generates questions that are semantically correct, syntactically precise, and contextually relevant. Detailed and precise feedback on the students' responses was also obtained with the automatic evaluation tool for open questions.

Finally, some technological and integration limitations are reviewed, and new lines of research are proposed to improve the presented system.

# Tabla de contenidos

# Capítulo 1

# Introducción

La educación es uno de los pilares fundamentales para el desarrollo individual y colectivo de las sociedades. En este contexto, las evaluaciones juegan un papel crucial, ya que permiten medir el progreso de los estudiantes y ajustar los métodos de enseñanza para maximizar el aprendizaje. Sin embargo, la creación manual de cuestionarios y evaluaciones es una tarea que llega a consumir una gran cantidad de tiempo del instructor, especialmente cuando se requiere un gran número de preguntas. Aquí es donde la automatización de evaluaciones mediante tecnologías avanzadas de Procesamiento de Lenguaje Natural (PLN) e Inteligencia Artificial (IA) adquieren una relevancia especial.

La promoción del autoaprendizaje constituye unas de las principales motivaciones de este trabajo fin de Máster. Permitir que los estudiantes puedan crear cuestionarios personalizados, no solo fomenta una mayor autonomía en el proceso educativo, sino que es esencial para proporcionar una educación adaptativa que se ajuste a las necesidades específicas de cada alumno, potenciando así la calidad del aprendizaje y la retención de conocimientos.

Por una parte, las evaluaciones basadas en preguntas de opción múltiple, han demostrado ser herramientas eficaces para el aprendizaje y la retención de información [45]. Sin embargo, la dificultad en este tipo de preguntas generalmente varían dependiendo del criterio del profesor, lo que introduce sesgos y variaciones en la estandarización o clasificación de niveles de dificultad. Por otra parte, en el caso de las preguntas abiertas, se ha demostrado que permiten una evaluación más profunda de la comprensión de los estudiantes, pero su corrección es aún más laboriosa, debido a que el instructor debe revisar cada respuesta individualmente y proporcionar retroalimentación detallada, lo cual demanda un esfuerzo y tiempo considerable.

En respuesta a estas necesidades, este trabajo busca innovar la forma en que se crean y aplican las evaluaciones en contextos educativos combinando un enfoque pedagógico basado en la Taxonomía de Bloom para incorporar dificultad en los cuestionarios y un enfoque basado en modelos de lenguaje de gran escala (LLM) para la generación de los mismos.

## 1.1. Estructura del documento

En el capítulo 1 se introduce brevemente las necesidades y los problemas encontrados, además de la motivación de este trabajo.

En el capítulo 2 se introducen los conceptos y tecnologías del estado del arte que van a ser empleados en este proyecto. En este capítulo, además, se da especial atención a la Taxonomía de Bloom y su adaptación en el contexto de preguntas para la implementación de dificultad.

En el capítulo 3 se describe la propuesta de valor y las capacidades del sistema desarrollado. Así también, se formulan los objetivos principales que persigue este trabajo.

El capítulo 4 analiza una descripción detallada de la arquitectura del Sistema de Generación y Evaluación de Cuestionarios (SGEC). Se comienza describiendo el modelo de dominio donde capturamos los detalles de los conceptos, roles y escenarios de casos de uso. Posteriormente, se describen los componentes de la arquitectura del sistema y se proporciona información sobre nuestras decisiones de diseño y despliegue de la aplicación en su primera versión.

En el capítulo 5 se comenta la metodología del proyecto. Se explica cómo se generan las preguntas (abiertas y de opción múltiple) y cómo se evalúa automáticamente las respuestas.

El capítulo 6 está dedicado a detallar el proceso de evaluación que se llevó a cabo. Especial énfasis se hace a la teoría de respuesta al ítem y modelo Rasch, el cuál es el método usado para cuantificar la dificultad en las preguntas. Además, se describe el análisis sintáctico de preguntas y una evaluación en términos de precisión del sistema evaluador de respuestas. Consecuentemente, en el capítulo 7 se muestran los resultados obtenidos de dos encuestas realizadas con preguntas abiertas y preguntas de opción múltiple, las estimaciones del modelo Rasch, los resultados del análisis sintáctico de preguntas y la evaluación de la precisión de respuestas del sistema SGEC.

Finalmente, en el capítulo 8 se detallan las conclusiones a partir de los resultados obtenidos y se plantean diferentes líneas de investigación para trabajos futuros.

# Capítulo 2

# Contexto

## 2.1. Preguntas abiertas y de opción múltiple

Si bien se ha visto en el tiempo que la generación de preguntas no ha sido tan popular como la tarea relacionada de responder a preguntas, se ha encontrado un aumento constante en el número de publicaciones en esta área durante los últimos años [3] [26].

Una percepción generalizada sobre las preguntas de tipo opción múltiple, es que únicamente pueden evaluar niveles de conocimiento básicos, mientras que las preguntas de rellenar espacios o de respuesta abierta son necesarios para evaluar niveles superiores de conocimiento. Sin embargo, diferentes publicaciones han mostrado que las preguntas de opción múltiple también pueden evaluar niveles cognitivos superiores [36] [10]. En este trabajo nos enfocamos solamente a las preguntas de respuesta abierta y las preguntas de opción múltiple. En este sentido, se hace un breve repaso del estado del arte de estas categorías de preguntas.

#### Preguntas Abiertas

Para crear preguntas de este tipo, se ha utilizado tradicionalmente reglas y plantillas, como se describe en Lindberg et al., 2013. [32] [22], y Labutov, et al., 2015. [27]. Sin embargo, esta aproximación no resulta ser práctica para situaciones del mundo real debido a dos motivos. En primer lugar, no capturan la complejidad semántica de los textos de manera efectiva, lo que se traduce a menudo en preguntas generadas poco relevantes para el contexto dado o con perdida de información importante. En segundo lugar, la falta de capacidad de aprender y adaptarse automáticamente a nuevos dominios o patrones en el texto, lo que la hace menos flexible y escalable. Estos datos han sido expuestos y descritos en un trabajo sobre la generación neuronal de preguntas para la comprensión lectora, Du, Shao, y Cardie, 2017 [15]

Las limitaciones expuestas en el enfoque basado en reglas y plantillas se ha visto superado con las redes neuronales profundas, el uso de arquitecturas recurrentes tipo codificador-decodificador [6, 14, 16, 39, 58, 62] y la aparición de Transformers [12, 28, 30, 33].

Se observó en múltiples trabajos en los que usaron arquitecturas transformers y similares, como Alberti et al., 2019 [2] [20], en donde la tarea de generación de una pregunta se la formulaba a partir de una respuesta objetiva y un documento como entrada. En ese sentido, datasets como

SQuAD1.1 [43] y NewsQA [51] han sido muy utilizados como entrenamiento previo de los modelos para capturar la relación de preguntas y respuestas. Este enfoque, sin embargo, queda obsoleto si planteamos el caso de preguntas con diferentes niveles de dificultad. Especialmente por la falta y el elevado costo para crear datasets de entrenamiento de preguntas en castellano etiquetadas con algún tipo de nivel de dificultad.

Finalmente, el trabajo realizado en Elkins, et. al. (2024) [18], demostró que los modelos de lenguaje de gran escala pueden generar satisfactoriamente cuestionarios a partir de un contexto dado y de calidad comparable a uno escrito a mano por un profesor.

**Preguntas de opción múltiple**

En el contexto de preguntas de opción múltiple, este tipo de preguntas se generan a partir de un contexto y se componen por una pregunta y un número de respuestas plausibles, las cuales son denominadas en este y otros trabajos "distractores".

Un trabajo reciente que demostró buenos resultados en la creación de preguntas de opción múltiple (a pesar de no implementar niveles de dificultad), creó un modelo usando el transformer T5 con ajuste fino sobre el dataset SQuAD [20]. Cabe mencionar también que algunos trabajos previos dedicaron esfuerzos solo a la generación de distractores, tal como se describe en Gao et al., 2018 [19], donde se intentó generar distractores más largos y coherentes, prestando mayor atención a la estructura sintáctica de una oración.

**Sistemas de generación automática de preguntas y respuestas**

A la fecha se han desarrollado algunos sistemas parecidos al nuestro para la generación de preguntas. Web-Experimenter [23] genera preguntas al estilo de "completa el espacio vacío" (Cloze question) para pruebas de competencia en inglés. AnswerQuest [46] está enfocado a la compresión lectora y genera solamente preguntas. SQUASH [25] descompone artículos más grandes en párrafos y genera una pregunta de comprensión de texto para cada uno. Sin embargo, aquellos sistemas carecen de la capacidad para generar distractores y trabajar en idioma Castellano. También se han encontrado servicios en línea adaptados para profesores. Por ejemplo, Quillionz [41] toma textos educativos más largos y genera preguntas según un dominio seleccionado por el usuario, mientras que Questgen [40] puede trabajar con textos de hasta 500 palabras de longitud. Si bien estos sistemas ofrecen recomendaciones útiles de preguntas, también requieren licencias de paga. Por su contraparte, un sistema gratuito y de código abierto es Leaf [52], el cual se enfoca en generar preguntas de opción múltiple a partir de textos. Sin embargo, carece del control de dificultad.

En resumen, podemos decir que todos los trabajos revisados y mencionados previamente han sido optimizados y desarrollados para la generación de preguntas en idioma Inglés. En nuestra búsqueda, además, no se encontraron sistemas que controlen el nivel de dificultad en la generación de preguntas, ni tampoco sistemas para la evaluación automática a preguntas abiertas.

## 2.2. Métricas de evaluación de cuestionarios

Dado que la evaluación humana de un cuestionario suele ser siempre confiable y precisa, su coste y el tiempo que requiere la hacen menos práctica para las evaluaciones a gran escala o en el caso en que un estudiante quiera autoevaluarse. Por este motivo, se propone también la implementación de un componente en el sistema SGEC para la evaluación automática de respuestas de preguntas

abiertas.

Para su implementación, se han estudiado en primer lugar las métricas usadas comúnmente en trabajos previos de evaluación automática, tal como BLEU [37], ROUGE [31] y METEOR [29]. Se ha encontrado que dichas métricas, en su mayoría, solo comparan los resultados generados por el sistema con una respuesta o texto de referencia y miden el solapamiento léxico [37] [31] o la similitud semántica [61]. En este sentido, estas métricas basadas en referencias padecen de muchos inconvenientes, entre los más importantes se destacan los siguientes:

1. Algunos estudios han observado una correlación ineficiente de las puntuaciones otorgadas automáticamente con la evaluación humana (Novikova et al., 2017 [34] y Dhingra et al., 2019 [13]).

2. Las métricas como BLEU para la traducción automática y ROUGE para el resumen se centran en medir el solapamiento léxico entre los resultados generados y una referencia, ya que están diseñadas para comparar la similitud entre cadenas de texto basadas en la coincidencia de palabras o n-gramas. Por tanto, carecen la capacidad de tomar en cuenta la semántica subyacente en respuestas.
   En la práctica, los textos generados que difieran de las referencias reciben una puntuación baja, inclusive si son correctas.

3. BLEU, ROUGE y METEOR son sensibles a la longitud del texto. Es decir, pueden favorecer respuestas más cortas sobre respuestas más largas debido a la manera en que ponderan la coincidencia de n-gramas.

4. No toman en cuenta la estructura gramatical o la coherencia de las respuestas.

Por tanto, se puede concluir que las métricas mencionadas anteriormente no son suficientemente precisas ni sensibles para evaluar respuestas a preguntas de manera completa. Un enfoque alternativo y reciente son las métricas aprendidas. Las métricas aprendidas son modelos que se entrenan para emular los juicios humanos sobre la calidad de las respuestas generadas por sistemas de procesamiento de lenguaje natural [47]. En lugar de depender de reglas predefinidas o coincidencias de palabras, estas métricas aprenden a evaluar la calidad de las respuestas a través del aprendizaje contextual, dentro del cual existen múltiples técnicas como zero shot y few shot [57] [21], como se describe en la siguiente sección. Empero, en términos generales, para crear una métrica adaptada para la evaluación automática de respuestas, se necesita presentar al LLM instrucciones en lenguaje natural con los parámetros de evaluación, limitaciones de puntuación y posibles casos o ejemplos.

## 2.3. Modelos de lenguaje preentrenados

Los modelos de lenguaje de gran escala o también en inglés conocido como “Large Lenguaje Models” (LLM), son una clase de modelos de aprendizaje profundo basados en el uso de inteligencia artificial para comprender, generar y manipular lenguaje humano. Los LLM se distinguen por contener un gran número de parámetros que generalmente se encuentra en el rango de los billones. Estos parámetros son ajustados durante el proceso de entrenamiento para aprender a realizar diversas tareas de procesamiento de lenguaje natural (PLN) como traducción, resumen, análisis de

sentimientos, generación de texto, etc. Dicho entrenamiento de un LLM se lleva a cabo utilizando grandes cantidades de datos textuales provenientes de diversas fuentes como libros, artículos, sitios web, y otros tipos de contenido escrito. Este entrenamiento se realiza utilizando técnicas de pre entrenamiento generativo, comúnmente con el objetivo de la predicción del siguiente token (palabra), lo que significa que aprenden a predecir una posible continuación de un texto de entrada inicial a partir de una distribución de probabilidad sobre el vocabulario preentrenado. Modelos como T5 [19] o GPT-3 han demostrado que a mayor cantidad de parámetros del LLM mejor la calidad, coherencia y corrección en la generación de texto. Por lo que en los LLM open source, se esperaría un mejor rendimiento con versiones más grandes.

En consonancia con el entrenamiento de los LLMs para la predicción del siguiente token, el enfoque adoptado para la generación de preguntas debe proporcionar a un LLM un input textual, llamado "**prompt**" para que el modelo lo procese y genere texto. Diseñar este prompt para generar una respuesta precisa u objetiva, puede ser una tarea difícil, lo que ha resultado en una nueva dirección de investigación llamada Ingeniería de Prompts.

Por tanto, los LLM en su mayoría funcionan a partir de prompts, los cuales pueden contener una serie de instrucciones y parámetros que determinan cómo procesar y generar texto. Ollama, una plataforma de inteligencia artificial que facilita la implementación y uso de modelos de lenguaje, permite que la declaración y definición de instrucciones (prompts) se guarden explícitamente en un archivo del modelo (modelfile). Este archivo permite que el modelo siga indicaciones en lenguaje humano para generar texto de manera precisa.

Para permitir que los LLM aprendan de las instrucciones en lenguaje natural y completen tareas del mundo real, se han investigado diferentes técnicas de ajuste fino a nivel de instrucciones, las cuales han mostrado impresionantes capacidades de generalización. Esta tendencia es conocida como aprendizaje contextual o "In-Context Learning" (ICL) [9], la cual se basa en la capacidad de los LLM para aprender y realizar tareas específicas basándose únicamente en el contexto o indicaciones proporcionado en lenguaje natural y sin necesidad de modificar los pesos del modelo a través del entrenamiento tradicional.

La interacción efectiva con modelos de lenguaje preentrenados requiere, por tanto, la elección de alguna técnica de aprendizaje contextual (ICL) para lograr una buena precisión y utilidad de las respuestas del modelo. Algunas de las técnicas más exploradas y relevantes a este proyecto son:

- **Aprendizaje con cero muestras (Zero-shot)**: La técnica de Zero-shot [56] se refiere a la capacidad de un modelo para realizar tareas sin haber sido específicamente entrenado en dichas tareas. Esto implica que el modelo puede realizar tareas (por ejemplo generar preguntas) basadas solo en una instrucción y sin ejemplos específicos de entrenamiento para la tarea en cuestión.
- **Aprendizaje con pocas muestras (Few-shot)**: La técnica de Few-shot o aprendizaje con pocas muestras, implica proporcionar al modelo algunos ejemplos (a menudo entre 1 y 10) de la tarea específica antes de generar una respuesta o realizar la tarea. Esto ayuda al modelo a entender mejor el contexto y los requisitos de la tarea. Su uso ha demostrado lograr un mejor rendimiento en las respuestas del LLM [8].
- **Cadena de pensamientos (CoT)**: Introducido en Wei et al. (2022) [56], la técnica de Chain of Thought (CoT) descompone el razonamiento complejo en una serie de pasos intermedios más simples, que pueden ser resueltos secuencialmente. Esto mejora la capacidad del modelo para manejar tareas que requieren un razonamiento más detallado y estructurado. CoT

puede ser combinado con few-shot para obtener mejores resultados en tareas que requieran un razonamiento complejo antes de responder.

- **Tree of Thoughts (ToT)**: El Árbol de Pensamientos o Tree of Thoughts (ToT), es una técnica propuesta en Yao et al. (2023) [59], la cual generaliza las indicaciones de la cadena de pensamientos y fomenta la exploración de pensamientos que sirven como pasos intermedios para la resolución de problemas. ToT mantiene un árbol de pensamientos, donde los pensamientos representan secuencias de lenguaje coherentes que sirven como pasos intermedios hacia la resolución de un problema. Este enfoque permite a un LLM autoevaluar el progreso a través de pensamientos intermedios realizados para resolver un problema mediante un proceso de razonamiento deliberado.

- **ReAct**: ReAct (Reasoning and Acting) es una técnica presentada por Yao et al., 2022 [60] en la que el modelo de lenguaje no solo genera texto basado en las instrucciones, sino que también realiza acciones basadas en el razonamiento y el uso de herramientas externas como bases de datos o páginas web para recuperar información adicional que produzca respuestas fiables y fácticas.

- **Self Instruction Tunning**: Esta técnica, publicada en (Peng et al., 2023) [38], describe el ajuste de un LLM a partir de datos sintéticos generados por otro LLM de última generación como (GPT4). En este contexto, los Llm de última generación son denominados comúnmente "maestros" y son instruidos con técnicas de prompting para seguir una tarea específica, y de esta manera ajustar a otro LLM.

Esta capacidad de aprendizaje contextual es particularmente relevante en el contexto de LLM de código abierto, tal como es el LLM Llama3, desarrollado por Meta [50], y el cual es usado en este trabajo. Llama 3 se destaca entre sus competidores por su precisión y su entrenamiento con 15 billones de tokens, permitiendo a desarrolladores e investigadores experimentar con modelos avanzados de lenguaje sin las restricciones de soluciones propietarias. Llama 3 está construida sobre una arquitectura Transformer sin codificador, las variantes del modelo vienen en dos tamaños: 8B y 70B, que indican 8.000 millones y 70.000 millones de parámetros respectivamente. También ofrece tres versiones, las cuales son: Base, Instruct y Chat. Base se refiere a la versión estándar preentrenada. Instruct y Chat son versiones afinadas y optimizadas mediante entrenamiento adicional con datos relevantes para aplicaciones de tipo preguntas-respuestas o instrucciones procedimentales.

Para complementar la optimización en la generación de texto de un LLM (en concreto en este trabajo Llama 3), se puede aplicar la técnica de "chunking" o división de texto [42]. El chunking implica dividir el texto en fragmentos más manejables, lo que facilita al modelo procesar grandes volúmenes de información. Aquí es donde la división semántica toma importancia, ya que el chunking solo toma en cuenta una cantidad de tokens para la división texto y por tanto los fragmentos podrían perder continuidad y coherencia. Por este motivo, la división semántica de texto organiza un documento en segmentos lógicamente coherentes, permitiendo al modelo mantener el contexto y la relevancia en cada parte del texto. Estas técnicas son especialmente útiles en el contexto de ICL, ya que permiten presentar ejemplos y tareas de manera estructurada y comprensible dentro de los límites del model-file o prompt. Al combinar in-context learning con chunking y división semántica, un LLM puede manejar eficientemente tareas complejas, proporcionando respuestas precisas y contextualizadas que reflejan una comprensión profunda del contenido.

## 2.4. Taxonomía de Bloom

La Taxonomía de Bloom (Bloom, 1956), revisada por Anderson y Krathwohl el año 2014 [4], es una conocida herramienta en el mundo educativo para redactar preguntas de examen y evaluar actividades de aprendizaje, así como para interpretar habilidades de los estudiantes. La Taxonomía de Bloom tiene dos dimensiones independientes, la dimensión del proceso cognitivo y la dimensión del conocimiento.

La dimensión cognitiva describe qué tipo de razonamiento es necesario para completar una tarea (responder a preguntas en nuestro caso), y propone seis niveles jerárquicos y acumulativos de aprendizaje (**Figura 2.1**). En general, los niveles inferiores requieren de menos esfuerzo porque se centran más en el recuerdo de la información, mientras que los niveles cognitivos superiores no, ya que implican la deconstrucción y construcción de conceptos.

El propósito de cada uno de los seis niveles cognitivos se describen a continuación:

1. **Recordar:** Este nivel implica la capacidad de recordar hechos, términos y conceptos básicos sin necesidad de entenderlos en profundidad y utiliza conocimientos que el alumno puede recuperar en el largo plazo. Ejemplos: recordar fechas históricas, definiciones clave, nombres de personas importantes.

2. **Entender:** Este nivel se centra en el entendimiento demostrativo básico de ideas, conceptos y hechos. Ejemplos: interpretar un texto, resumir un artículo, parafrasear una idea.

3. **Aplicar:** Este nivel consiste en utilizar todo el conocimiento aprendido para resolver problemas en situaciones diferentes a las del contexto de aprendizaje. Ejemplos: aplicar una fórmula matemática para resolver un problema, implementar un método aprendido en un experimento científico

4. **Analizar:** Este nivel implica la descomposición en partes de un problema, percibiendo el significado de cada una de las partes en relación con el conjunto, y cómo se relacionan unas con otras. El alumno debe ser capaz de identificar causas y motivos. Ejemplos: comparar diferentes puntos de vista o contrastar teorías.

5. **Evaluar:** Este nivel se centra en la capacidad de elaborar juicios sobre informaciones, ideas o calidad de un trabajo de acuerdo con una serie de criterios preestablecidos. Ejemplos: Criticar un diseño experimental o justificar una decisión.

6. **Crear:** Crear es el nivel de mayor dificultad en la pirámide de Bloom. El alumno debe ser capaz de generar, planificar, modificar y producir para formar una idea o solución coherente, ya sea creando un patrón nuevo o bien modificando uno existente. Ejemplos: programar, innovar un proyecto, diseñar un experimento.

La segunda dimensión de la Taxonomía de Bloom revisada se centra en el tipo de conocimiento de la tarea (**Figure 2.2**). El tipo de conocimiento puede ser:

- **Factual:** Se refiere a los hechos, la terminología, los detalles o los elementos esenciales que los estudiantes deben conocer o con los que deben estar familiarizados para comprender una disciplina o resolver un problema.

- **Conceptual:** Se refiere al conocimiento de clasificaciones, principios, generalizaciones, teorías, modelos o estructuras pertinentes a un área disciplinar concreta.

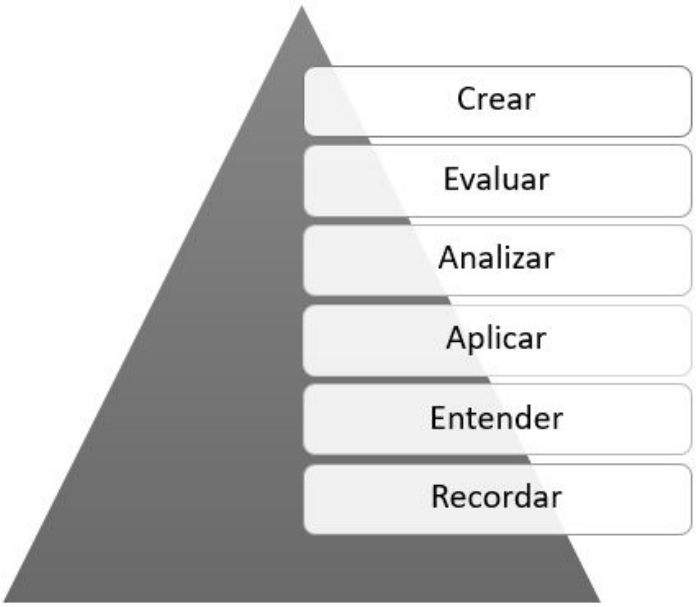



Figura 2.1: ***Dimensión Cognitiva de la Taxonomía de Bloom revisada por Anderson y Krathwohl [4].*** *Cada nivel de la jerarquía representa un proceso cognitivo.*

- **Procedimental:** Se refiere a la información o los conocimientos que ayudan a los estudiantes a usar en la práctica algo específico de una asignatura o área de estudio. También se refiere a métodos de indagación, destrezas, algoritmos, técnicas y metodologías particulares.
- **Metacognitivo:** Es el conocimiento estratégico o reflexivo sobre cómo proceder para resolver problemas, tareas cognitivas, incluyendo el conocimiento contextual y el conocimiento de uno mismo.

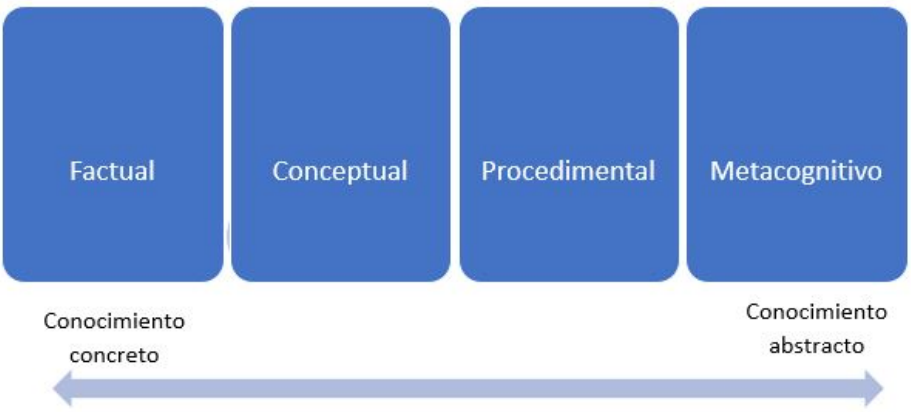



Figura 2.2: ***Dimensión del Conocimiento de la Taxonomía de Bloom revisada por Anderson y Krathwohl [4].*** *Cada tipo de conocimiento implica un razonamiento más o menos abstracto.*

Elkins, et. al. (2024) [18] incluyó la taxonomía de Bloom para mejorar la calidad de preguntas, demostrando en uno de sus experimentos realizados que los profesores que participaron en su encuesta prefirieron fuertemente escribir cuestionarios con la ayuda de un LLM controlando los niveles de Bloom.

En este sentido, se ha visto por conveniente utilizar la taxonomía de Bloom revisada [7], para formular preguntas, debido a su clasificación de las dimensiones del conocimiento y los procesos cognitivos, los cuales han sido ampliamente usados para actividades de aprendizaje y de evaluación.

## 2.5. Relación entre la taxonomía de Bloom y niveles de dificultad

En la literatura, la dificultad de preguntas de opción múltiple se ha alineado con éxito con la dimensión del proceso cognitivo de la taxonomía de Bloom (Tiemeier et al. (2011) [49] y Kim et al. (2012) [24]). Una importante investigación relacionada a la generación de preguntas con dos niveles de dificultad, evaluó empíricamente el poder predictivo de las dos dimensiones de Bloom para estimar la dificultad de las preguntas [35]. Esta investigación concluyó demostrando la relación entre los niveles de la jerarquía de Bloom y la dificultad de preguntas. Las preguntas situadas en niveles bajos de la jerarquía de Bloom demostraron ser más fáciles a las preguntas situadas más arriba en la jerarquía de Bloom, las cuales resultaron ser más demandantes e implicaban un análisis sintáctico y semántico más amplio.

Anderson et. al. (2014) [4], recomienda explícitamente que los usuarios de la taxonomía de Bloom infieran los niveles de las dimensiones a partir del enunciado de las tareas, que en el caso de este trabajo son preguntas. Por ejemplo, verbos como “comparar” o “generalizar” se asocian al nivel Comprender, mientras que “identificar” o, más sencillamente, “nombrar” se asocian al nivel Recordar.

En este sentido, una investigación realizada por Stanny (2016) [55], identificó los verbos de la taxonomía de Bloom a partir de 30 recopilaciones publicadas en sitios web. Este trabajo tuvo en cuenta los verbos y la frecuencia con la que se incluían en las tablas de verbos disponibles en línea. El procedimiento consistió en tomar en cuenta un verbo si se encontraba en un nivel de Bloom en 10 o más tablas de las 30 fuentes incluidas en el estudio. El resultado de dicha investigación en idioma Inglés, consiguió 433 verbos únicos, de los cuales 236 (54,5 %) aparecían en una sola categoría. Los 197 verbos restantes (45,5 %) aparecían en dos a seis categorías distribuidos de la siguiente manera: 108 verbos (24,9 %) aparecieron en dos categorías, 41 verbos (9,5 %) aparecieron en tres categorías, 30 verbos (6,9 %) aparecieron en cuatro categorías, 15 verbos (3,5 %) aparecieron en cinco categorías, y 3 verbos (elegir, relacionar, seleccionar) aparecieron en las seis categorías.

Los investigadores establecieron un 75 % de acuerdo como el criterio umbral para mantener un verbo en una columna. La tabla que utilizamos en este proyecto (traducida) se reduce a la versión más conservadora, con verbos que han sido nominados por 10 o más veces para una categoría Bloom. Un total de 104 verbos se muestra en la **Figura 2.3**.

| Recordar | f | Entender | f | Aplicar | f | Analizar | f | Evaluar | f | Crear | f |
|---|---|---|---|---|---|---|---|---|---|---|---|
| citar | 17 | clasificar | 18 | aplicar | 19 | analizar | 24 | evaluar | 22 | organizar | 22 |
| definir | 21 | comparar | 21 | calcular | 19 | apreciar | 11 | argumentar | 17 | ensamblar | 14 |
| describir | 14 | convertir | 13 | elegir | 24 | categorizar | 12 | valorar | 17 | combinar | 14 |
| identificar | 20 | defender | 16 | demostrar | 22 | clasificar | 13 | elegir | 30 | componer | 14 |
| etiquetar | 19 | describir | 22 | construir | 20 | comparar | 20 | concluir | 20 | construir | 29 |
| listar | 20 | discutir | 21 | dramatizar | 16 | contrastar | 15 | criticar | 22 | crear | 29 |
| ubicar | 14 | distinguir | 14 | emplear | 21 | criticar | 13 | criticar | 12 | diseñar | 24 |
| coincidir | 13 | estimar | 12 | ilustrar | 16 | diagramar | 18 | defender | 17 | desarrollar | 19 |
| memorizar | 14 | explicar | 18 | interpretar | 12 | diferenciar | 21 | estimar | 12 | idear | 20 |
| nombrar | 20 | expresar | 17 | manipular | 17 | discriminar | 14 | evaluar | 19 | formular | 16 |
| delinear | 16 | extender | 11 | modificar | 19 | distinguir | 13 | juzgar | 19 | generar | 19 |
| recordar | 20 | generalizar | 12 | operar | 16 | dividir | 16 | manejar | 19 | inventar | 15 |
| recitar | 16 | inferir | 15 | practicar | 21 | examinar | 16 | preparar | 19 | modificar | 15 |
| reconocer | 14 | interpretar | 17 | preparar | 16 | inferir | 14 | reajustar | 16 | ordenar | 19 |
| grabar | 16 | ubicar | 12 | producir | 22 | delinear | 15 | reconciliar | 16 | planificar | 22 |
| relacionar | 14 | parafrasear | 10 | relatar | 19 | señalar | 19 | cuestionar | 22 | preparar | 18 |
| repetir | 20 | predecir | 16 | programar | 13 | cuestionar | 19 | seleccionar | 20 | calificar | 21 |
| reproducir | 11 | reconocer | 13 | mostrar | 20 | relacionar | 19 | sintetizar | 16 | revisar | 21 |
| seleccionar | 16 | reportar | 16 | esbozar | 13 | seleccionar | 22 | probar | 14 | escribir | 17 |
| declarar | 23 | reiterar | 16 | resolver | 19 | subdividir | 16 | | | | |
| | | revisar | 20 | utilizar | 25 | probar | 14 | | | | |
| | | reescribir | 20 | | | | | | | | |
| resumir | 20 | | | | | | | | | traducir | 21 |

Figura 2.3: ***Lista indicativa de verbos asociada a los 6 niveles cognitivos de Bloom.*** *104 verbos que han sido nominados en el estudio por 10 o más veces para una categoría Bloom (f).*

Finalmente, para agrupar la taxonomía de Bloom en un número determinado de niveles de dificultad, nos hemos basado en el estudio realizado en Vachev et al., 2017 [54]. Este estudio analizó 13,189 enunciados de exámenes de asignaturas de grado y posgrado con etiquetas de clasificaciones de la taxonomía de Bloom proporcionadas por el comité docente de una universidad Australiana. De estos enunciados, 8,115 eran de asignaturas de grado y 5,074 de posgrado. La distribución de los resultados de aprendizaje por niveles de Bloom mostró que en asignaturas de grado había más ejemplos del nivel Aplicar que cualquier otra categoría (24.6 %, N=1996), mientras que en posgrado predominaban las de los niveles Crear (31.3 %, N=1598) y Evaluar (30.2 %, N=1533). Sin embargo, se encontró dificultades para diferenciar entre los niveles de Recordar y Comprender, ya que una proporción significativa de enunciados del nivel Recordar se identificaron como del nivel Comprender (55.6 %, N=80). En la **Figura 2.4** se muestra la distribución de los resultados del aprendizaje por niveles de la taxonomía de Bloom de este estudio [54].

| Clasificación de Bloom | Pregrado (%) | Pregrado (N) | Posgrado (%) | Posgrado (N) |
|---|---|---|---|---|
| Recordar | 1.36 | 110 | 0.67 | 34 |
| Entender | 11.61 | 942 | 5.22 | 265 |
| Aplicar | 24.60 | 1996 | 15.16 | 769 |
| Analizar | 19.57 | 1588 | 17.42 | 884 |
| Evaluar | 21.84 | 1772 | 30.21 | 1533 |
| Crear | 21.04 | 1707 | 31.32 | 1589 |

Figura 2.4: ***Distribución de los niveles de Bloom para las asignaturas de grado y posgrado del estudio realizado en Vachev et al., 2017 [54].***

En resumen y partiendo de lo expuesto anteriormente sobre la taxonomía de Bloom, nuestra hipótesis establece que el grado de dificultad de una pregunta está relacionada con los niveles de la taxonomía de Bloom siempre que se la haya usado para la construcción de dicha pregunta. En concreto, para el sistema SGEC el nivel Recordar se corresponde a la categoría fácil. Similar-

mente, los niveles de Entender y Aplicar a la categoría intermedia. Y finalmente las categorías de Evaluar y Analizar a la categoría difícil. Adicionalmente, para asegurar el nivel de dificultad propuesto, se requerirá que la frase verbal describa el proceso cognitivo previsto y que cada nivel de la dimensión de conocimiento sea asociado a una categoría de dificultad propuesta.

# Capítulo 3

# Propuesta

Para abordar el problema y necesidad de la creación y evaluación de preguntas, se propone la implementación de un Sistema de Generación y Evaluación Automática de cuestionarios educativos (SGEC) a partir de un material docente en formato de texto.

Este sistema generador de preguntas controla la dificultad de las preguntas en tres niveles (fácil, intermedio y difícil). La aplicación está diseñada para permitir al estudiante seleccionar el número deseado de preguntas, la dificultad y el tipo de preguntas en las que quiera examinarse (preguntas abiertas y de opción múltiple) pudiendo también seleccionar el porcentaje de la cantidad de cada tipo de pregunta en un cuestionario. Además de la generación de cuestionarios, se propone la evaluación automática de respuestas para las preguntas abiertas (**Anexo 3**). De esta manera logramos implementar un aprendizaje adaptativo completo que se ajusta a la situación en el proceso de aprendizaje de cada estudiante.

Cabe mencionar además que este trabajo ha contribuido al proyecto de innovación educativa (Código: IE24.6109) [1] realizado en la Universidad Politécnica de Madrid, en el cual se busca transformar la experiencia de evaluación, ofreciendo a los estudiantes un enfoque más dinámico y personalizado para medir y progresar en su comprensión y dominio de los contenidos impartidos en cualquier asignatura.

En este contexto, los estudiantes tendrán la posibilidad de realizar evaluaciones más frecuentes, proporcionándoles no solo una herramienta de medición, sino también una oportunidad continua para reflexionar sobre su aprendizaje, ya que junto con los resultados, el sistema SGEC proporciona una explicación con extractos de contenido de la respuesta correcta. Al proporcionar retroalimentación instantánea y detallada sobre cada respuesta, este enfoque proactivo contribuye a un aprendizaje más personalizado y autónomo, lo que en última instancia, esperamos que se traduzca en una mejora significativa de los resultados académicos.

## 3.1. Sistema SGEC

En este trabajo, nos hemos enfocado en el uso de modelos de lenguaje de gran escala de código abierto para la generación de preguntas, con el fin de diseñar un sistema de libre acceso y fácil disponibilidad.

En este sentido, Llama 3 [50] ha destacado significativamente en comparación con competidores como Gemma 7B de Google, Mistral 7B de Mistral y Claude 3 Sonnet de Anthropic, demostrando excelentes resultados en dos pruebas críticas: 1) Comprensión lingüística multitarea masiva (MMLU) y 2) Respuesta a preguntas de propósito general (GPQA). Según estadísticas publicadas por Meta, el modelo Llama 3 70B supera a estos modelos en ambas pruebas, que son fundamentales para nuestra tarea de generación de preguntas. Por tanto, se ha visto conveniente para la primera versión del sistema SGEC el uso de Llama3 8B.

Para abordar los desafíos de procesamiento asociados a este LLM, hemos usado el software Ollama. Este software permite la ejecución de modelos de lenguaje de gran escala cuantizados directamente desde una máquina local, prescindiendo de requisitos de RAM y GPU. En particular, hemos utilizado la versión conversacional cuantizada a 4 bits de Llama 3 8B.

El sistema SGEC ha sido diseñado como una aplicación web basada en el uso del lenguaje de programación Python y técnicas de aprendizaje contextual o ïn-context learning"(ICL) para instruir al modelo de lenguaje Llama 3.

Esta elección se justifica no solo por las capacidades superiores del LLM Llama 3 en las pruebas MMLU y GPQA, sino también por su enfoque en el aprendizaje contextual, que es crucial para la generación y evaluación de preguntas. Además, el uso de Ollama para la implementación local del modelo asegura una mayor accesibilidad en comparación con la versión original que requiere grandes recursos de procesamiento.

## 3.2. Objetivos

En este trabajo se pretende dar especial atención y prioridad a la evaluación del rendimiento del modelo generativo en las tareas de: 1) Generación de preguntas controlando el nivel de dificultad, 2) Control de respuestas de opción múltiple (distractores), y 3) Evaluación automática de respuestas. Por tanto, nuestros objetivos son los siguientes:

1. Incorporar la dimensión cognitiva y de conocimiento de la taxonomía de Bloom en la generación de cuestionarios mediante técnicas de prompting y few-shot learning.

2. Medir la dificultad en preguntas generadas automáticamente según los tres niveles de dificultad propuestos en este trabajo, mediante el modelo Rasch.

3. Analizar la correspondencia léxica entre las preguntas generadas para cada nivel de dificultad con los niveles de la taxonomía de Bloom.

4. Evaluar la eficacia del ajuste fino de instrucciones para la personalización de preguntas y control de distractores.

5. Evaluar cuantitativamente la similitud semántica de distractores y demostrar su efecto en la generación de dificultad.

# Capítulo 4

# Arquitectura del Sistema

Esta sección detalla la arquitectura del Sistema de Generación y Evaluación de Cuestionarios (SGEC). La sección 4.1 describe el modelo de dominio, donde capturamos los detalles de los conceptos, roles y escenarios de casos de uso. La sección 4.2 describe la arquitectura detallada del sistema y proporciona información sobre nuestras decisiones de diseño.

## 4.1. Modelo de Dominio

El propósito del modelo de dominio es promover la comprensión común de terminologías entre todos las partes interesadas. El modelo de dominio también describe los escenarios de casos de uso soportados por el sistema.

- **Fuente de Conocimiento.** La fuente de conocimiento se refiere a la entidad a partir de la cual se puede generar un conjunto de preguntas. La entidad puede ser un texto o un documento en formato PDF. La fuente de conocimiento es la entrada principal del sistema, y es proporcionada por el usuario. Las fuentes de conocimiento se pueden obtener de diferentes lugares, como por ejemplo, material docente, transcripciones de videos de conferencias, capítulo de un libro o texto de un sitio web. Cabe resaltar que la calidad de las preguntas generadas depende en gran medida de la estructura y contenido de las fuentes de conocimiento proporcionadas.
- **Pregunta de Opción Múltiple** Nuestra pregunta de opción múltiple consta de dos partes, una pregunta principal y un conjunto de 4 posibles respuestas que distraen al usuario. Solo una opción entre las posibles respuestas es correcta. Los usuarios responden indicando como máximo una opción.
- **Distractores.** Se denomina distractor a cada respuesta alternativa plausible pero incorrecta.
- **Pregunta Abierta.** Una pregunta abierta requiere que el usuario responda en un formato de texto abierto, expresando su conocimiento o comprensión sobre el tema. La respuesta a este tipo de preguntas no se limita a un conjunto de opciones.
- **Cuestionario.** El cuestionario estará compuesto por preguntas abiertas o de opción múltiple. Además, un cuestionario tendrá un nivel de dificultad asociado, pudiendo ser fácil, intermedio o difícil, y estará asociado a solo una fuente de conocimiento.

- **Evaluación Automática**. La evaluación automática es una herramienta que no requiere intervención humana y que cumple el rol de instructor para dar una retroalimentación y puntaje inmediato a una respuesta.

### 4.1.1. Escenarios de casos de uso

El sistema de Generación y Evaluación de Cuestionarios (SGEC) admite los siguientes escenarios;

1. Un usuario puede cargar un documento o texto al sistema y obtener un cuestionario generado automáticamente como respuesta del sistema y listo para imprimir.
2. Un usuario puede definir el nivel de dificultad y definir el porcentaje del tipo de preguntas del cuestionario.
3. En una situación en la que el usuario no está satisfecho con un cuestionario, podrá generarlo nuevamente para recibir modificaciones en las preguntas.
4. Un usuario puede autoevaluarse a partir de un cuestionario generado y recibir una calificación inmediata junto con la explicación de las respuestas correctas.

## 4.2. Diseño del Sistema

El diseño del sistema describe la descomposición lógica del sistema, el flujo de interacción entre componentes y las tecnologías utilizadas para el desarrollo de los componentes.

### 4.2.1. Diagrama de contexto de sistema

El diagrama de contexto de Sistema describe la interacción entre el SGEC y entidades externas. El ecosistema del SGEC comprende los siguientes componentes:


Figura 4.1: ***Diagrama de Contexto del Sistema SGEC.***

- **Usuario.** Entidad que envía peticiones al sistema SGEC a través de la aplicación web.
- **Sistema de Generación y Evaluación de Cuestionarios (SGEC).** El componente SGEC representa y define los límites de nuestro sistema. El SGEC encapsula los componentes lógicos, tecnologías y decisiones de diseño tomadas durante el diseño del sistema. El SGEC interactúa con la interfaz de usuario a través de una API REST y proporciona servicios que abarcan todos los escenarios de casos de uso descritos anteriormente. También interactúa mediante una API REST con el LLM para la generación y evaluación de preguntas.
- **LLM - Llama 3.** El modelo de lenguaje preentrenado Llama 3 8B es una dependencia del sistema SGEC, la cual está alojada en un servidor externo para facilitar y acelerar los

tiempos de generación y evaluación de preguntas. La comunicación entre el SGEC y Llama3 se realiza a través de una API REST.

### 4.2.2. Vista lógica del sistema

Esta sección describe los componentes y los estilos de arquitectura adoptados en este proyecto.

El SGEC utiliza una arquitectura cliente-servidor, Este enfoque permite gestionar de manera centralizada numerosas solicitudes de múltiples usuarios realizadas mediante la aplicación web. La comunicación entre SGEC y los usuarios se facilita mediante una arquitectura REST (Representational State Transfer),la cual utiliza protocolos estándar de HTTP/HTTPS para permitir interacciones entre clientes y servidores.

El SGEC se descompone en 3 componentes: Interfaz Web (frontend), Interfaz de programación de aplicaciones (API), Parseador de documentos y el Modelo generativo de lenguaje.

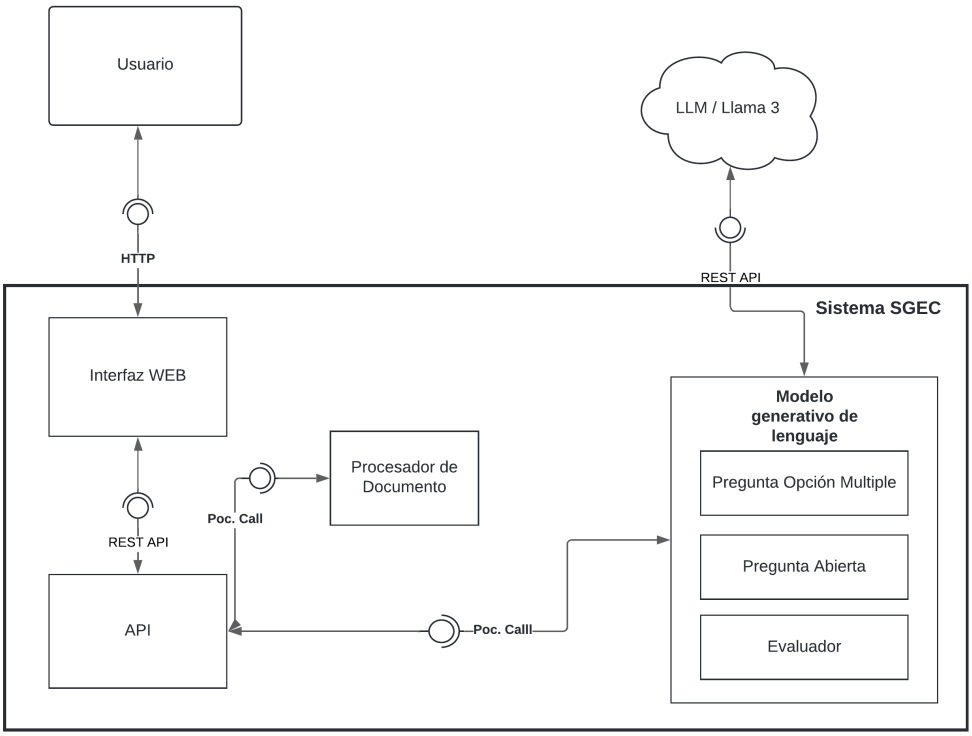



Figura 4.2: ***Vista lógica del sistema SGEC.***

1. **Interfaz Web**
   El servidor web aloja el frontend, proporcionando una interfaz gráfica para los usuarios. Este frontend se ha desarrollado con JavaScript, CSS y HTML, y se ejecuta sobre Flask, un microframework de Python. Por otro lado, WTForms se emplea para validar los parámetros ingresados por el usuario, asegurando la corrección de los datos antes de ser enviados.

   Además, Flask-Session en su versión de almacenamiento local se utiliza para guardar tem-

poralmente las preguntas y respuestas solicitadas por el usuario y proporcionadas por el servidor de aplicaciones. Esta configuración permite una interfaz dinámica y responsiva, facilitando una interacción eficiente con el sistema de generación de cuestionarios.

2. **API**
   La API actúa como intermediario entre el modelo generativo de lenguaje y la interfaz web. Expone los endpoints de la API REST que la interfaz web utiliza para que los usuarios puedan acceder al servicio. Para este fin, utilizamos FastAPI y Uvicorn. FastAPI proporciona las herramientas y la estructura para construir APIs RESTful, mientras que Uvicorn es un servidor ASGI de alto rendimiento que se usa para desplegar aplicaciones FastAPI.

   Los usuarios realizan solicitudes a través de la interfaz web, las cuales llegan a la API mediante endpoints REST. La API recibe y valida las solicitudes para luego invocar los componentes necesarios según la tarea que se esté realizando, como el parseador de documentos o el modelo generativo de lenguaje. Una vez que los componentes han completado sus procesos, el resultado se devuelve y se muestra en la interfaz web.

3. **Parseador de Documentos**
   Este componente de procesamiento de documentos es responsable de extraer semánticamente el texto del documento recibido como entrada del usuario. Para ello se utilizan representaciones vectoriales (embeedings) y la distancia del coseno para extraer conjuntos de oraciones semánticamente similares. En esta primera versión de la aplicación SGEC, el procesador de documentos admite la extracción desde formato de documentos PDF mediante el uso de la librería pyPdf. El código fue implementado en Python 3.11.4 y el proceso completo se describe en la sección 5.

4. **Modelo generativo de lenguaje**

   El componente de generadores encapsula la lógica para crear preguntas y evaluar respuestas. El componente de modelo generativo de lenguaje incluye los módulos de preguntas de opción múltiple, de preguntas abiertas y el módulo de evaluación de respuestas. Este componente se comunica con el modelo Llama 3 a través del software Ollama, el cual expone la versión Llama3-8B cuantizada a 4 bits sin necesidad de GPU. Los modelos de Llama 3 se utiliza como parte de LLM-as-a-Service, accedido mediante llamadas HTTP.

### 4.2.3. Vista de procesos

La vista de procesos describe el flujo lógico de comunicación e interacción entre los distintos componentes del sistema. Una vez se haya inicializado el sistema SGE, el servicio estará listo para aceptar solicitudes desde la interfaz WEB. El usuario podrá realizar una llamada al sistema SGEC a través de la interfaz Web o directamente a través de una API para generar o evaluar un cuestionario. La solicitud para generar un cuestionario consta de una fuente de conocimiento (texto/documento), tipo de preguntas, número de preguntas y el nivel de dificultad. La solicitud de evaluar un cuestionario, consta de una fuente de conocimiento, una pregunta y una respuesta.

Una vez hecha la solicitud, la API comprobará la tarea que se ha solicitado y la exactitud de los datos. En el caso de generación de cuestionarios, la API invocará al parseador de documentos

para extraer el texto del documento y hacer una división semántica de todo el texto. Finalmente, el procesador de documentos devolverá partes del documento sin formato como respuestas. Cada división de texto recibido es introducido con el resto de entradas en el componente del modelo generador de lenguaje. Este devolverá en cualquier tarea una respuesta tipo JSON.

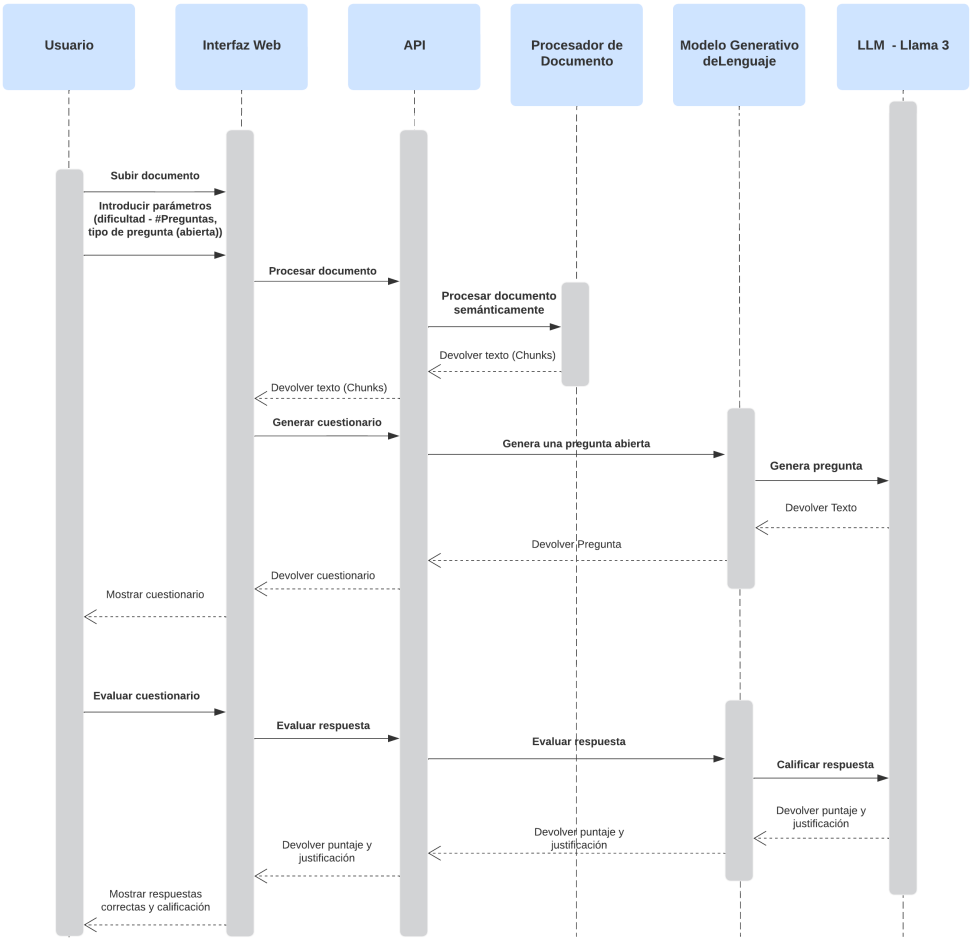



Figura 4.3: ***Vista de procesos SGEC cuando se selecciona preguntas abiertas.***

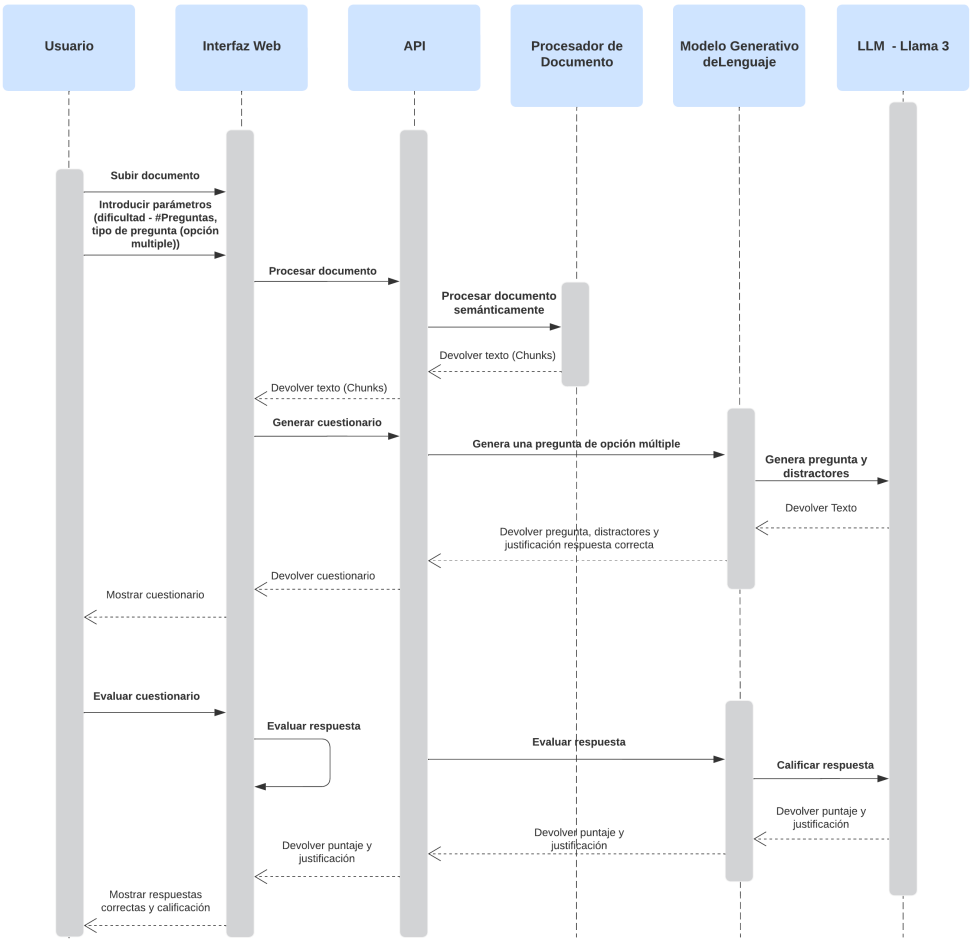



Figura 4.4: ***Vista de procesos SGEC cuando se selecciona preguntas de opción múltiple.***

### 4.2.4. Diagrama de flujo

El siguiente diagrama de flujo describe las acciones que se toman desde el momento en que el usuario realiza una solicitud hasta cuando el usuario responde las preguntas y se evalúa.

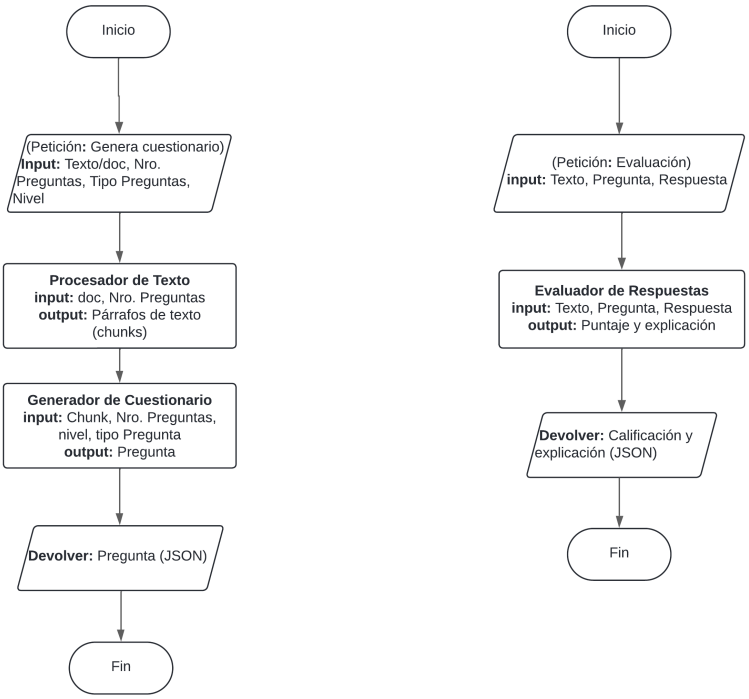



Figura 4.5: ***Diagrama de flujo SGEC.*** *El diagrama de la izquierda muestra el flujo en la generación de preguntas. El diagrama de la derecha muestra el flujo de la evaluación de respuestas a preguntas abiertas.*

### 4.2.5. Vista física

La vista física modela el entorno a nivel de hardware del sistema. Asigna los componentes lógicos al hardware en el que se ejecutan. Nuestro sistema se descompone en 3 unidades desplegables **Figura 4.6**. Estos son: el modelo de lenguaje a gran escala (Llama3), el servicio web (frontend), y el sistema SGEC (backend). Elegimos desplegar estos componentes en su propio entorno por separado para permitir que sean fácilmente escalables y modificables.

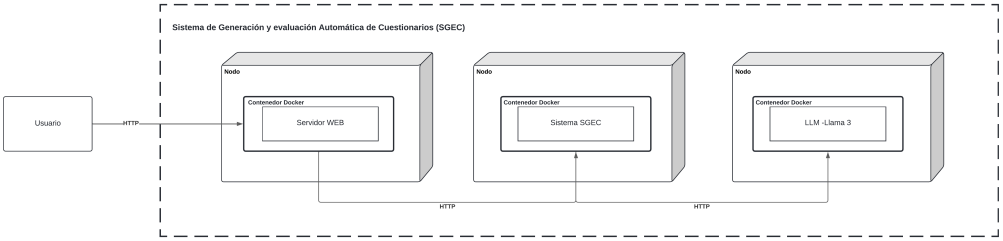



Figura 4.6: ***Vista Física SGEC.***

# Capítulo 5

# Generación de cuestionarios

En este capítulo se describe el procedimiento para la generación de cuestionarios controlando el nivel de dificultad y la evaluación automática del sistema SGEC. Cabe recordar que la velocidad de la generación de preguntas depende de la CPU, por lo que es recomendable usar GPU en entornos reales. Sin embargo, para permitir el uso, experimentación y despliegue local de esta aplicación se usó el software Ollama, el cual permite ejecutar modelos de lenguaje a gran escala cuantizados directamente desde una máquina local y sin requerimientos de GPU. La versión usada fue Llama3-8B cuantizada a 4 bits.

El repositorio de este proyecto se encuentra disponible en Github para su consulta detallada: SGEC [5].

## 5.1. División semántica de texto

La división semántica de texto, o "chunking", es una técnica importante en el procesamiento de grandes volúmenes de información, especialmente en el contexto de los LLM. Esta sección se enfoca en cómo implementamos esta técnica para mejorar la eficiencia y coherencia de preguntas en nuestro sistema, particularmente al trabajar con la versión conversacional de Llama3.

Para la elección de la versión conversacional de Lama-3, se han evaluado a la vez la versión Instructivo y Conversacional. La versión conversacional destaca significativamente por su capacidad para entender y retener información, lo que facilita un intercambio dinámico y adaptativo, crucial para nuestro sistema que requiere una interacción continua con nuevos contextos (textos). En el contexto de preguntas y respuestas con contextos amplios, el modelo conversacional demostró una mayor eficacia en el manejo de información compleja y contextual en comparación con el modelo instructivo. Por estas razones, hemos optado por emplear la versión conversacional de Llama3, que se adapta mejor a nuestras necesidades específicas.

El modelo conversacional requiere de un input con un contexto para la generación de texto. Dado que estos modelos conversacionales aceptan un número limitado de tokens en los archivos de configuración del modelo (ModelFile), surge la necesidad de implementar la técnica de chunking. El chunking consiste en dividir grandes cantidades de texto en segmentos más pequeños, en el caso de Lama-3, menores a 8192 tokens (palabras). Esta técnica es crucial para manejar grandes volúmenes de información sin perder coherencia y precisión.

Nuestra estrategia de chunking va más allá de una simple división del texto basada en un número fijo de tokens. Hemos desarrollado un método más sofisticado y preciso, centrado en mantener la coherencia y relevancia semántica en cada fragmento. Este enfoque asegura que la información esencial no se diluya durante el proceso de división.

En la práctica, por ejemplo, cuando se trabaja con documentos de varias páginas, el componente "Parseador de Documentos´´ del sistema SGEC identifica párrafos o grupos de oraciones que son semánticamente similares y los agrupa en unidades de texto coherentes. Este enfoque garantiza que cada fragmento de texto mantenga información relevante, lo que maximiza la eficiencia y precisión en la generación de preguntas y respuestas. Este proceso es fundamental, ya que cada división del texto debe producir preguntas abiertas o de opción múltiple que sean claras y pertinentes.

Para lograr una fragmentación eficaz, utilizamos la representación vectorial de cadenas de texto, conocida como "embeddings". Los embeddings permiten capturar el significado semántico de las cadenas, facilitando el agrupamiento de fragmentos semánticamente similares. Mediante el uso de la distancia del coseno, podemos identificar y agrupar textos similares, asegurando que cada fragmento sea coherente y contextualmente relevante. Este enfoque avanzado nos permite mantener altos estándares de coherencia y relevancia en los textos generados, optimizando la interacción y experiencia del usuario con el modelo conversacional de Llama3.

El proceso de división semántica que realiza el sistema SGEC es el siguiente:

1. **División inicial del documento:** Primero, dividimos el documento por páginas. De cada página, extraemos los párrafos, asumiendo que un párrafo termina con uno o más saltos de línea (`\n \n+`). Luego, cada párrafo se divide en oraciones utilizando los caracteres "." "?" y "!" , que representan siempre la finalización de una oración. Las oraciones son almacenadas en una lista de diccionarios con las siguientes llaves: "sentence", "index" , "combined_Sentence".

2. **Combinación de oraciones:** En segundo lugar, combinamos cada oración con la oración que le precede y la que le sigue. Este enfoque tiene como objetivo capturar el contexto inmediato de cada oración.

   Es importante mencionar que ampliar el número de oraciones precedentes y sucesivas puede mejorar la división semántica en algunos casos. Por ello, creamos una función configurable que define el tamaño de ventana (bufferSize), indicando el número de oraciones anteriores y posteriores a tomar en cuenta. Nuestra implementación inicial ha sido conservadora, manteniendo el bufferSize = 1. (**Anexo 4.**).

3. **Vectorización de oraciones combinadas:** A continuación, obtenemos la representación vectorial de las oraciones combinadas (combinedSentences) utilizando OllamaEmbeddings y Llama3. Esta representación numérica es esencial para calcular las distancias entre los grupos de oraciones (**Figura 5.1**).

   Calculamos las distancias entre los grupos secuenciales de tres oraciones usando la distancia del coseno. Este valor numérico es un componente clave para determinar la similitud o disimilitud entre grupos de oraciones. Valores altos sugieren un cambio de tema o contexto, mientras que valores bajos o cercanos a la media indican continuidad semántica. El valor de la distancia del coseno entre oraciones se puede observar en la **Figura 5.2** donde mostramos un ejemplo de un documento de texto de 4 páginas y sus distancias de similitud entre grupos de oraciones.

4. **Identificación de puntos de ruptura:** Finalmente, identificamos puntos de ruptura entre los grupos formados. Como se puede observar en **Figura 5.2**, podemos ver secciones donde las distancias son menores y mayores.

   Nos enfocamos en encontrar valores atípicos dispersos, ya que representan diferencias significativas de significado y son buenos puntos de ruptura para la división del texto. Definimos un umbral de distancia a partir del cual consideraremos los valores atípicos. En nuestro caso, hemos fijado un punto de ruptura en el percentil 95.

   Cabe mencionar que ajustar el límite percentil puede modificar la cantidad de divisiones, lo cual es útil para adaptarse a diferentes necesidades, como generar un mayor o menor número de preguntas.

[ { **'sentence'**: La microbiota intestinal humana, una comunidad compleja de microorganismos que residen en el tracto gastrointestinal (...).
**'index'**: 0,
}
{ **'sentence'**:Este microbioma diverso incluye bacterias, arqueas, virus y hongos, predominando especies bacterianas como Firmicutes, (...).
**'index'**: 1,
**'combined_sentence'**: La microbiota intestinal humana, una comunidad compleja de microorganismos que residen en el tracto gastrointestinal (...).
Este microbioma diverso incluye bacterias, arqueas, virus y hongos, predominando especies bacterianas como Firmicutes, (...).
}
...
]

Figura 5.1: **Ejemplo de grupos seccuenciales de tres oraciones (Buffersize = 1).** Nótese en "Combined_Sentences" la oración que la precede (index 0), la actual (index 1), y la que la sucede (...). Esta secuencia continua con el resto de oraciones del documento.

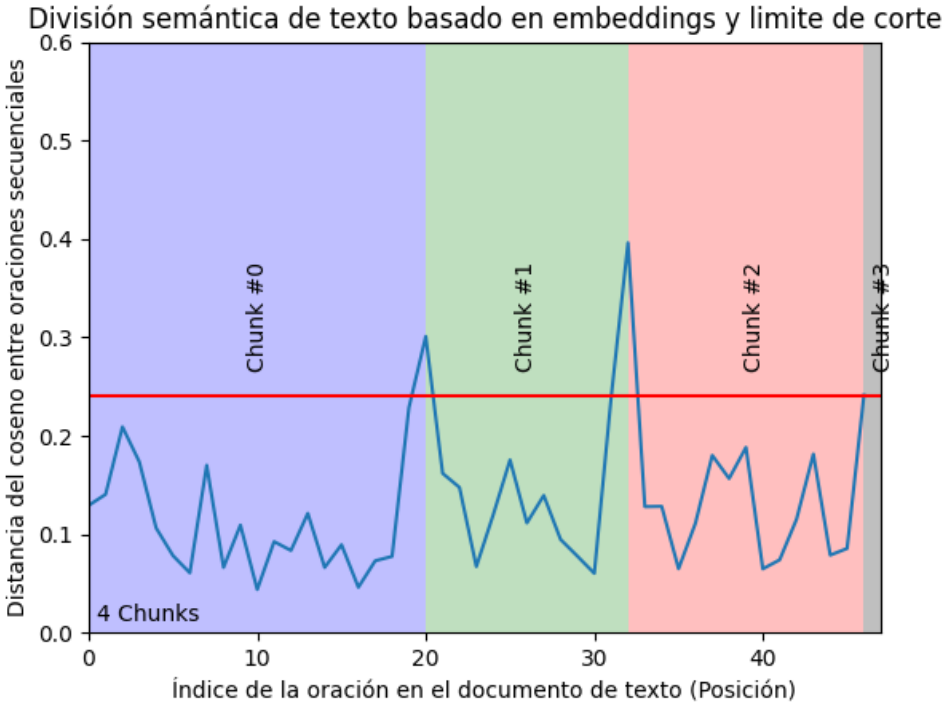



Figura 5.2: ***Distancias del coseno obtenidas a partir de embeddings.*** *La línea horizontal roja marca el percentil 95 y los "chunks" hacen referencia al conjunto de oraciones que siguen una semántica similar.*

## 5.2. Generación de preguntas con dificultad

Para crear cuestionarios mediante el sistema SGEC con tres niveles de dificultad, hemos agrupado y distribuido la taxonomía de Bloom revisada en tres nuevas categorías. Para este fin, se tomaron en cuenta solo la dimensión cognitiva de Recordar, Entender, Aplicar, Evaluar y Analizar **Figure 2.1**, y también la dimensión de conocimiento factual, Procedimental y Conceptual **Figure 2.2**. El motivo de excluir el nivel de Crear de la dimensión cognitiva, se debe a que por la naturaleza de las preguntas de opción múltiple, estas son respuestas cerradas que no toman en cuenta la creatividad de la respuesta del estudiante. En la dimensión de conocimiento, se excluyó el nivel Metacognitivo porque en preguntas de opción múltiple, las respuestas autorreflexivas del usuario no se las puede inferir desde el sistema con respuestas cerradas.

Por tanto, la agrupación y combinación de las dos dimensiones de Bloom resultó de la siguiente manera:

1. Para el nivel fácil, se usó el primer nivel de la dimensión cognitiva, Recordar. Por ser el nivel más básico y tomando en cuenta estudios previos [39], se lo situó al nivel Recordar en nuestro nivel 1 (fácil) junto con la dimensión de conocimiento Factual. Estas preguntas de examen podrían pedir al alumno que identifique o defina información factual de la fuente de conocimiento.

2. Para el nivel intermedio, se usaron los niveles de Entender y Aplicar de la dimensión cognitiva. Esta combinación además ha sido evaluada en Pado et al., 2017 [35], donde se demostró una diferencia significativa entre el nivel de Recordar y Entender en términos de dificultad. En este sentido y por situarse ambos en el siguiente nivel después del nivel Recordar en la escala Bloom, se situaron en el nivel SGEC 2 junto con el nivel procedimental y conceptual de la dimensión de conocimiento. Se espera que en estas preguntas los alumnos realicen un cálculo, describan un procedimiento, apliquen algún conocimiento en específico o interpreten información clave.

3. Finalmente, para el nivel difícil se usaron los niveles de Analizar y Evaluar de la dimensión cognitiva. Las preguntas del nivel Analizar y Evaluar comparten una similitud según Qi et al. (2020). Dado que ambos niveles son los dos últimos más altos en la escala de Bloom, se los situó en el nivel SGEC 3 (difícil) junto con la dimensión de conocimiento conceptual. Se espera que en el caso de una pregunta de Analizar, los alumnos descompongan la información para justificar su estructura y relación o sinteticen la información antes de responder, mientras que en una pregunta de Evaluar, se espera que hagan juicios sobre el valor de las ideas o conceptos para un propósito dado. En resumen, este nivel requiere que los estudiantes critiquen y justifiquen decisiones, basándose en criterios y estándares específicos.

| | |
|---|---|
| **PREGUNTA** | ¿En qué año se publicó un estudio en la revista médica "The Lancet" que relacionaba la vacuna de la triple vírica con el autismo? |
| **D. Cognitiva** | Recordar |
| **D. Conocimiento** | Factual |
| **Nivel SGEC** | Fácil |
| **Estimación RASCH** | -0.71 |

Cuadro 5.1: **Ejemplo de pregunta con dificultad 1 (fácil), generada a partir de la taxonomía Bloom agrupada.** Nótese el requerimiento implícito factual (año) y la tarea implícita de recordar información específica (dimensión cognitiva Recordar).

El SGEC incorporó mediante propmts y aprendizaje contextual estas nuevas directrices para la generación de preguntas con dificultad. Nuestro objetivo se centra en primer lugar en explotar la técnica de fewshot. En las instrucciones, por tanto, se incluyó: (1) el nivel cognitivo de Bloom, (2) el nivel de conocimiento de Bloom, (3) verbos indicativos de cada nivel de Bloom y (3) fewshot learning. Los prompts se muestran en las secciones 5.2.1 y 5.2.2.

Cabe mencionar, por último, que un desafío encontrado con los LLM open source es el forzar el tipo de respuesta de los modelos en formato JSON. Como solución a esto, hemos implementado una función en Python desde el Servidor APP, la cual itera por un máximo de 3 veces la generación de la misma pregunta en caso de no estar en formato JSON. Esta implementación ha sido en comparación más fácil, efectiva y compatible que con la solución dada por la librería Pydantic.

### 5.2.1. Preguntas de opción múltiple

Para evitar verbosidad en las figuras que se presentan a continuación, la **Figura 5.3** muestra la estructura de la cabecera que todos los archivos de configuración del modelo (ModelFile) tienen, tanto para preguntas abiertas y de opción múltiple.

Por ejemplo, en la **Figura 5.3** se definen los parámetros como la temperatura y "Topk Sampling", los cuales son fundamentales para controlar la aleatoriedad y la coherencia en la generación de texto. El valor de 0.1 en la temperatura ajusta la suavidad de la distribución de probabilidades de las palabras siguientes de manera más coherente y predecible. Mientras que Top-k Sampling o num_keep24 limita la selección de la siguiente palabra a las 24 opciones con mayores probabilidades, equilibrando la diversidad y la calidad del texto generado.

```
FROM llama3:latest
TEMPLATE {{ if .System }} <|start_header_id|>system
<|end_header_id|>
{{.System}} <|eot_id|>{{end}} {{if .Prompt}}
<|start_header_id|>user <|end_header_id|>
{{.Prompt}} <|eot_id|>{{end}}
<|start_header_id|>assistant <|end_header_id|>
{{.Response}} <|eot_id|>
PARAMETER num_keep 24
PARAMETER stop <|start_header_id|>
PARAMETER stop <|end_header_id|>
PARAMETER stop <|eot_id|>
PARAMETER temperature 0.1
```

Figura 5.3: Cabecera del archivo Modelfile para el ajuste fino a nivel de instrucciones. (Parte 1).

```
.SYSTEM """ Eres un asistente en Castellano que genera preguntas y
respuestas usando la Taxonomía de Bloom. La taxonomía de Bloom es una
estructura conceptual que clasifica los procesos cognitivos en seis
niveles jerárquicos (recordar, entender, aplicar, analizar, evaluar,
crear) y también incorpora una dimensión del tipo de conocimiento
(factual, conceptual, procedimental, metacognitivo). Para esta
tarea deberás usar el tipo de conocimiento 'conceptual' y el nivel
cognitivo de 'analizar' o 'evaluar'.

Utiliza sólo la información del contexto que recibirás del usuario
para generar una pregunta y cuatro opciones de respuestas. Solo una
respuesta deberá ser correcta. En total deberás generar una pregunta
y cuatro opciones de respuestas: 'OPCION 1' 'OPCION 2' 'OPCION 3' y
'OPCION 4'. La respuesta correcta siempre deberá estar en la 'OPCION
4'. La 'OPCION 1' 'OPCION 2' y 'OPCION 3' deben ser respuestas
incorrectas.
La pregunta debe ser de nivel difícil. Para generar una pregunta
de nivel difícil la pregunta debe estar basada en el nivel de
'analizar' o 'evaluar' de la taxonomía Bloom y el tipo de pregunta
debe ser 'conceptual'. Las preguntas de tipo conceptual se refieren
a la comprensión de conceptos, principios y teorías que mencione
el texto, además implica la capacidad de organizar y relacionar
ideas, identificar patrones y comprender las relaciones entre
diferentes conceptos. La pregunta puede ser de comparar y contrastar
ideas, argumentar puntos de vista o tomar decisiones basadas
en la información disponible del texto. Es necesario que uses
alguno de los siguientes verbos en el enunciado de la pregunta:
Analizar, categorizar, clasificar, comparar, criticar, diferenciar,
distinguir, examinar, Evaluar, argumentar, concluir, resumir,
estimar, sintetizar.
La pregunta y las respuestas deben tener menos de 17 palabras. Es
obligatorio que todo esté en Castellano.
Utiliza los siguientes ejemplos como referencia de preguntas de nivel
difícil:
¿Qué criterio se usa para evaluar las implicaciones éticas y legales
de la utilización de técnicas de hacking ético para probar la
seguridad de sistemas informáticos?

...

¿Cómo puedes comparar la complejidad del conjunto de instrucciones y
la eficiencia energética en sistemas informáticos modernos?
Analice críticamente las evidencias paleontológicas, genéticas y
anatómicas que respaldan la teoría de la evolución por selección
natural de Darwin. """
```

Figura 5.4: Ejemplo de ajuste fino de instrucciones para el nivel SGEC 3 (difícil), parte 2. Los demás ejemplos (...) se omiten en la figura por cuestiones de espacio.

### 5.2.2. Preguntas abiertas

A continuación se muestra como ejemplo, el archivo de configuración para la generación de preguntas abiertas de nivel fácil.

```
.SYSTEM """ Eres un asistente en Castellano que genera preguntas y
respuestas usando la Taxonomía de Bloom. La taxonomía de Bloom es una
estructura conceptual que clasifica los procesos cognitivos en seis
niveles jerárquicos (recordar, entender, aplicar, analizar, evaluar,
crear) y también incorpora una dimensión del tipo de conocimiento
(factual, conceptual, procedimental, metacognitivo). Para esta tarea
deberás usar solamente el tipo de conocimiento 'factual' y el nivel
cognitivo de 'recordar'.
Utiliza sólo la información del contexto que recibirás del usuario
para generar una pregunta. El nivel de dificultad de la pregunta
debe ser de nivel fácil. Para generar una pregunta de nivel fácil
la pregunta debe estar basada en el nivel 'recordar' de la taxonomía
Bloom y el tipo de pregunta debe ser factual. La pregunta factual
tiene como respuesta algún hecho, el nombre de una persona, o de una
localidad, la extensión o longitud de un objeto o el día en el cual
sucedió un evento. El nivel de recordar busca recuperar información
básica del texto. El objetivo es crear una pregunta o instrucción
que implique recordar información básica del texto, como hechos,
términos o conceptos, definiciones, identificar elementos o listar
características. Es necesario que uses alguno de los siguientes
verbos en el enunciado de la pregunta: Citar, definir, describir,
identificar, etiquetar, enumerar, nombrar, mencionar, reconocer,
relacionar, repetir.

La pregunta y las respuestas deben tener menos de 10 palabras. Es
obligatorio que todo esté en Castellano. Utiliza los siguientes
ejemplos como referencia de preguntas de nivel fácil:

Ejemplo definiciones de términos: ¿Cuál es la definición de la
palabra filantropía?, ¿Cómo se define una red neuronal recurrente?,
¿Qué significa ADN?, ¿Cuál era la longitud del muro de Berlín?.

Ejemplo recuperación de datos: ¿Cuántos planetas hay en nuestro
sistema solar?, ¿Cuál es el río más grande del mundo?, ¿Qué evento
histórico marcó un antes y después durante la revolución francesa?

Ejemplo de lista de características: Enumera los estados de la
materia. ¿Puedes listar algunos beneficios del ejercicio regular?,
¿Cuáles son las cinco principales atracciones turísticas de París?
"""
```

Figura 5.5: Ejemplo del ajuste fino de intrucciones para el nivel propuesto 1 (fácil).

## 5.3. Evaluación automática de respuestas abiertas

Finalmente, en nuestro sistema que genera preguntas abiertas, hemos implementado un componente adicional de evaluación automática de respuestas abiertas. El proceso de evaluación consiste en proporcionar al modelo tanto la pregunta generada como la respuesta del estudiante, permitiendo que el modelo analice la coherencia, precisión y relevancia de la respuesta en relación con la pregunta. En esta versión del sistema SGEC, hemos implementado una escala de puntuación simple y efectiva. Utilizando el modelo Llama3, las respuestas de los estudiantes se puntúan en una escala de tres niveles: 0, 0.5 y 1. Una respuesta recibe una puntuación de 0 si está incorrecta, 0.5 si es suficientemente correcta y 1 si es completamente correcta. Este enfoque permite una evaluación clara y directa, aunque presenta desafíos, como asegurar la precisión y consistencia en la

puntuación. Además, es esencial que el modelo pueda discernir adecuadamente los matices en las respuestas para evitar evaluaciones injustas y garantizar que las puntuaciones reflejen fielmente la calidad de las respuestas dadas.

A continuación se presenta las instrucciones que se usó para implementar el componente de evaluación automática.

```
.SYSTEM """ Eres un profesor de universidad y tu tarea es evaluar la
respuestas de un estudiante. Deberás usar como referencia la pregunta
original y el texto que se te de para evaluar cada respuesta. Asigna
un puntaje basado en la precisión y completitud de la respuesta.
Puntaje 0: en caso que la respuesta sea irrelevante, sea incorrecta,
esté vacía. También se dara un puntaje de 0 si el estudiante responde
únicamente con la palabra 'si', 'no' o 'no sé'. Puntaje 0.5: en caso
que el estudiante mencione en su respuesta partes o fragmentos del
texto correctamente y sea coherente. Puntaje 1: Cuando la respuesta
es totalmente o casi correcta.
El puntaje deberá estar en la variable 'PUNTAJE' y la justificación
del puntaje debe ir en la variable 'EXPLICACION'. Es obligatorio
que todo esté en Castellano y que la respuesta sea formateada con
el siguiente esquema:

"'json {{ "PUNTAJE": "string" // puntaje final de la respuesta,
"EXPLICACION": "string" // explicación y justificación del puntaje
}} "'

Utiliza los siguientes ejemplos para guiarte y evaluar correctamente
las respuestas de los estudiantes.

Ejemplo 1: pregunta: ¿Cuál fue el impacto de la colonización española
en las culturas indígenas de América Latina?
Respuesta: La colonización española tuvo un impacto negativo en las
culturas indígenas.
Texto: La colonización española tuvo un impacto profundo en las
culturas indígenas de América Latina. Los colonizadores impusieron
su idioma, religión y sistema de gobierno, lo que resultó en la
aculturación y, en muchos casos, en la desaparición de culturas
indígenas enteras. Sin embargo, también hubo un intercambio cultural
que resultó en una mezcla de tradiciones y costumbres.

"json { "PUNTAJE": "0.5", "EXPLICACION": "La respuesta del estudiante
menciona un aspecto del impacto (negativo), pero es incompleta y no
abarca todos los detalles proporcionados en el texto." } "

Ejemplo 2: pregunta: ¿Qué factores llevaron a la independencia de los
países latinoamericanos en el siglo XIX?
Respuesta: Los países latinoamericanos se independizaron debido a la
influencia de la Revolución Francesa y las guerras napoleónicas.
Texto: La independencia de los países latinoamericanos en el siglo
XIX fue el resultado de diversos factores, incluyendo la influencia
de la Revolución Francesa, las guerras napoleónicas, y el descontento
con las políticas coloniales españolas.

"json { "PUNTAJE": "1", "EXPLICACION": "La respuesta del estudiante
menciona correctamente los factores clave (la influencia de la
Revolución Francesa y las guerras napoleónicas) y es suficientemente
precisa y completa." } "
(...)
"'
```

Figura 5.6: Ejemplo del ajuste fino de intrucciones para evaluar las respuestas.

# Capítulo 6

# Evaluación

Para validar nuestro trabajo se ha considerado evaluar nuestro sistema en 3 puntos:

En primer lugar, hemos usamos la teoría de respuesta al ítem y el modelo Rasch para medir la dificultad en preguntas, ya que este modelo brinda una escala común para los niveles de dificultad. En segundo lugar, para evaluar la integración de la taxonomía de Bloom en preguntas, hemos realizado un análisis sintáctico observando los verbos presentes en las preguntas generadas. Y en tercer lugar, para evaluar las respuestas, hemos analizado la similitud semántica en los distractores (preguntas de opción múltiple) y en caso de preguntas abiertas hemos contabilizado el número de fallos en las puntuaciones de respuestas.

## 6.1. Evaluación de preguntas

### 6.1.1. Teoría de respuesta al ítem y modelo Rasch

La teoría de respuesta al ítem, o "IRT" por sus siglas en inglés, son modelos estadísticos utilizados para analizar la relación entre las respuestas de los individuos a preguntas de un examen (ítems) y un rasgo o habilidad latente subyacente [53]. Estos modelos suponen que la capacidad de un alumno y la dificultad de una pregunta no son directamente observables, sino que dependen probabilísticamente de las puntuaciones observadas. Entre los modelos existentes, el más conocido y que mejor se adapta a nuestro trabajo, es el análisis de Rasch [44].

El análisis de Rasch es una técnica estadística muy utilizada, su uso se ha revisado [17] y tratado en textos [48]. Concretamente, el modelo de Rasch estima la dificultad de la pregunta y la capacidad (habilidad) del alumno.

Como notación general, en el modelo de Rasch se usa el subíndice "*i*" para los ítems y "*n*" para los evaluados. Expresamos las puntuaciones que los estudiantes recibieron en cada ítem como 1, en caso de ser correcto, o 0 en caso de ser incorrecto. De esta manera la puntuación que obtiene el evaluado en el ítem *i* se expresa como $x_{ni}$ , que toma el valor 0 o 1. La matriz X = [ $x_{ni}$ ] expresa las puntuaciones de todos los sujetos en todos los ítems.

Dada la ecuación **6.1** se puede calcular la dificultad de un ítem (pregunta), donde $B_n$ es la habilidad del estudiante *n* y la dificultad de cada ítem *i* se expresa como $D_i$. Aquí, asumimos que un valor

más alto de $D_i$ significa que el ítem $i$ es más difícil, y un valor más alto de $B_n$ significa que el evaluado $n$ es más capaz o hábil.

$$P_{ni}(x = 1|B_n, D_i) = \frac{e^{(B_n - D_i)}}{1 + e^{(B_n - D_i)}} \tag{6.1}$$

El éxito (x = 1) de un alumno n en una pregunta i, está vinculado a la diferencia entre la habilidad del alumno y la dificultad de la pregunta. Si la habilidad es mayor que la dificultad, el alumno tiene más probabilidades de aprobar, o si ocurre lo contrario, el alumno tiene más probabilidades de suspender. Las estimaciones de B y D se realizan iterativamente a partir de los resultados de las pruebas. Las medidas resultantes se expresan en logits la cual tiene las dos siguientes propiedades deseables. En primer lugar, se transforma el eje "y" del rango $-\infty$ a $+\infty$ en logit, permitiendo así una representación más lineal de las probabilidades. Y en segundo lugar, la dificultad de las preguntas se centra en 0, de modo que los ítems fáciles tendrán estimaciones de dificultad bajas o negativas y los ítems difíciles tendrán estimaciones de dificultad altas.

Por tanto, la interpretación de las estimaciones de dificultad en el modelo de Rasch siguen la siguiente lógica:

- **Estimación Negativa:** Indica que la pregunta es más fácil en comparación con el promedio. Significa que los encuestados tienen una mayor probabilidad de responder correctamente a esa pregunta en comparación con una pregunta de dificultad promedio.

- **Estimación Positiva:** Indica que la pregunta es más difícil en comparación con el promedio. Significa que los encuestados tienen una menor probabilidad de responder correctamente a esa pregunta en comparación con una pregunta de dificultad promedio.

Para evaluar la dificultad con nuestro sistema, hemos realizado dos experimentos con dos grupos de 8 estudiantes en forma de una encuesta. Para ello, seleccionamos aleatoriamente 3 textos académicos de nivel universitario. Utilizando el sistema SGEC generamos 9 preguntas abiertas (cuestionario 1) y 9 preguntas de opción múltiple (cuestionario 2). Para el cuestionario 1 y 2 se generaron 3 preguntas fáciles, intermedias y difíciles usando cada uno de los 3 textos. Ambos cuestionarios fueron respondidos por los 8 estudiantes usando Google Forms. En total, obtuvimos 144 respuestas de las 18 preguntas.

En estas encuestas, además de responder a las preguntas, se pidió al estudiante que indique el nivel de dificultad percibido para cada pregunta. Con estos resultados de dificultad percibida evaluamos el porcentaje de acuerdo sobre la dificultad generada y percibida por los estudiantes. Y en segundo lugar, a partir de las respuestas de las preguntas, obtuvimos una métrica de dificultad usando el modelo Rasch.

| **Texto 2** | |
|---|---|
| En la intersección de la neurociencia cognitiva y la inteligencia artificial, se ha gestado un paradigma fascinante: la simulación computacional de la cognición humana. Este enfoque busca emular los procesos neuronales y cognitivos del cerebro humano en sistemas artificiales, con el objetivo último de comprender mejor la mente humana y replicar sus capacidades en máquinas. Sin embargo, este esfuerzo se ve enfrentado a desafíos formidables, como la complejidad del cerebro humano y la falta de un marco teórico unificado para modelar la cognición. | |
| **Dificultad** | 1 (fácil) |
| **Pregunta** | ¿Qué paradigma se gesta en la intersección de neurociencia cognitiva e inteligencia artificial? |
| **Dificultad Percibida** | (1) , (2), (3) |
| **Dificultad** | 2 (medio) |
| **Pregunta** | ¿Cuál es el objetivo principal de la simulación computacional de la cognición humana? |
| **Dificultad Percibida** | (1) , (2), (3) |
| **Dificultad** | 3 (difícil) |
| **Pregunta** | ¿Cómo se pueden evaluar los desafíos que enfrenta el enfoque de simulación computacional de la cognición humana, como la complejidad del cerebro humano y la falta de un marco teórico unificado para modelar la cognición? |
| **Dificultad Percibida** | (1) , (2), (3) |

Cuadro 6.1: **Ejemplo preguntas a partir del texto 2 de la encuesta realizada de preguntas abiertas.**

### 6.1.2. Análisis sintáctico de preguntas

Para la evaluación de preguntas, se presenta un breve análisis sintáctico de preguntas generadas por el SGEC en Python mediante el uso de la biblioteca de procesamiento de lenguaje natural (NLP) Stanza. Stanza es una herramienta desarrollada por la Universidad de Stanford que proporciona análisis lingüísticos detallados, incluyendo el etiquetado de partes del discurso (POS), lematización y análisis de dependencias. Nuestros objetivos en este análisis son: (1) Identificar y contar la frecuencia de los verbos lematizados pertenecientes a la taxonomía Bloom. (2) Identificar y contar la frecuencia de los adverbios interrogativos. (3) Determinar la cantidad de palabras promedio en preguntas por cada nivel de dificultad.

Este análisis se llevó a cabo utilizando un conjunto de 50 preguntas en español, incluyendo las de la encuesta realizada.

## 6.2. Evaluación de respuestas

Para la evaluación de respuestas de opción múltiple, se ha estudiado la similitud entre los distractores de cada pregunta, como se explica en la sección 6.2.1. En el caso de preguntas abiertas, se han utilizado las preguntas de la encuesta de preguntas abiertas y se ha comprobado por un humano

que el sistema SGEC justifique y puntúe la respuesta del estudiante correctamente. Consideramos un fallo cuando una respuesta es correcta, pero el sistema la considera incorrecta, o vice versa. Los comentarios y una evidencia se presentan directamente en los resultados, sección 7.

### 6.2.1. Distractores

Para evaluar cómo los distractores se alinean a los niveles de dificultad SGEC, se utilizó la métrica de similitud coseno para medir la similitud entre distractores a partir de embeddings. El objetivo fue determinar la similitud semántica de los distractores de preguntas de opción múltiple en función del nivel de dificultad.

Nuestra metodología consistió en generar representaciones vectoriales de cada distractor y medir la similitud entre ellos. Para esto, empleamos la función cosine_similarity de la librería sklearn .metrics.pairwise. La similitud coseno mide el coseno del ángulo entre dos vectores, proporcionando un valor entre -1 (totalmente disímiles) y 1 (totalmente similares). Finalmente, para obtener una medida única de la similitud entre todas los embbedings de una pregunta, calculamos la similitud promedio. Esto se realizó sumando todas las distancias coseno y dividiendo por el número total de pares únicos de emmbedings. Los resultados se muestran como la "Similitud Promedio del Coseno"para cada nivel de la encuesta realizada, **Figura 7.7**.

# Capítulo 7

# Resultados

## 7.1. Resultados de la evaluación de dificultad en preguntas

### 7.1.1. Preguntas abiertas

Los resultados de la percepción de los estudiantes sobre el nivel de dificultad de cada pregunta, se traducen en el porcentaje de acuerdo de dificultad en preguntas abiertas. Estos resultados fueron bastante buenos, en el sentido que se alinearon los niveles de dificultad generados con los niveles percibidos **Figura 7.1**. Por ejemplo, para el nivel de preguntas generadas de nivel fácil, hubo un 95 % de acuerdo entre los participantes y solo un 4.7 % en desacuerdo. Para las preguntas generadas de nivel intermedio, el porcentaje de acuerdo llegó a ser del 62.5 % y un 25 % percibieron como preguntas de nivel fácil. Uno de los motivos puede deberse a que para ambos niveles (fácil e intermedio) se usó el nivel cognitivo de "Recordar", aunque con diferentes niveles de conocimiento. Finalmente, para las preguntas generadas de nivel difícil se obtuvo un 79.12 % de acuerdo y el 20.8 % restante indicaron que fuera de nivel intermedio.

Por otro lado, los resultados de las respuestas a las preguntas de la encuesta se muestra en la **Figura 7.2**. En esta gráfica se ha calculado el promedio del puntaje obtenido de las 72 respuestas recogidas de esta encuesta. Se puede observar que existe una relación de mayor aciertos en preguntas fáciles y menor acierto en preguntas difíciles.

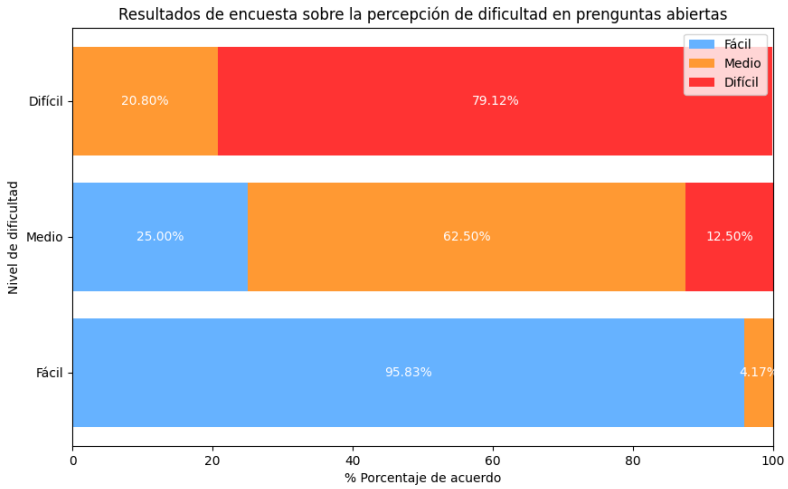



Figura 7.1: ***Resultados de la encuesta realizada con preguntas abiertas sobre los niveles de dificultad percibidos en cada pregunta.***

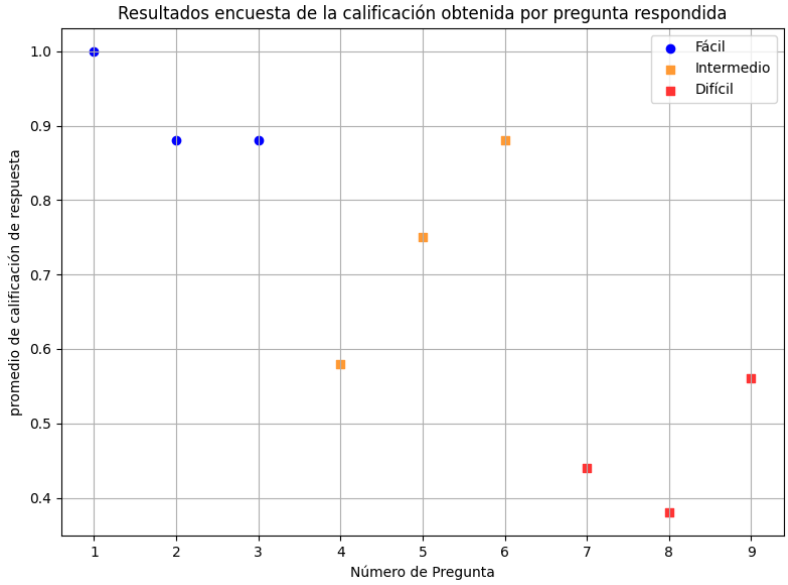



Figura 7.2: ***Resultados de la evaluación de respuestas a las preguntas abiertas en la encuesta realizada.*** *Nótese que las preguntas fueron reordenadas y agrupadas, siendo del 1-3 fáciles, 4-6 intermedias y 7-9 difíciles.*

Finalmente, cuantificamos el nivel de dificultad con el modelo Rasch usando los puntajes de las 72 respuestas obtenidas después de la evaluación automática. Para ilustrar cómo se distribuyeron las estimaciones de dificultad para cada categoría, se muestra una gráfica de cajas con los resultados en la **Figura 7.3**.

En el eje X se representa el nivel de dificultad generado por el SGEC para las preguntas de la encuesta de preguntas abiertas y el eje Y muestra las estimaciones de dificultad calculadas por el modelo de Rasch y la línea horizontal de color rojo representa la media.

Como se puede observar, el rango intercuartílico de la categoría fácil (-0.94 a -0.75) indica que la mayoría de las estimaciones de dificultad para preguntas fáciles se mantienen en un rango negativo y por debajo de la media. Similarmente, el rango intercuartílico para las preguntas de categoría intermedio, se extiende desde -0.45 hasta -0.01, lo que indica que tienen una dificultad más cercana a la media. Finalmente, el rango intercuartílico para las preguntas de categoría difícil, se extiende desde 1.16 a 1.57, indicando que la mayoría de las preguntas de nivel difícil reciben una estimación Rasch con los valores más altos y positivos.

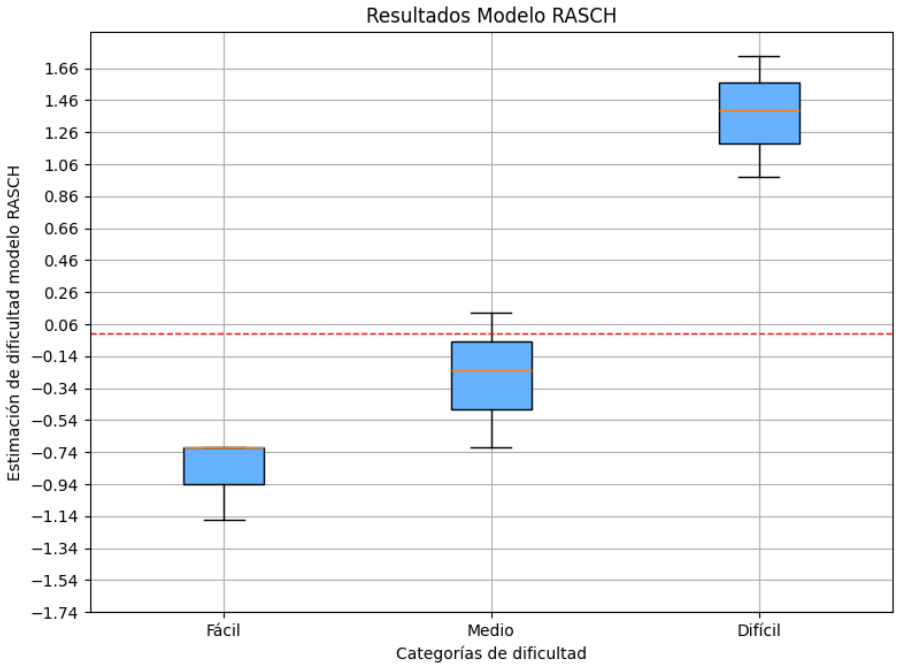



Figura 7.3: ***Resultados de las estimaciones de dificultad del modelo Rasch sobre la encuesta realizada de preguntas abiertas.***

### 7.1.2. Preguntas de opción múltiple

En relación con los resultados de la encuesta realizada con preguntas de opción múltiple, podemos decir que el porcentaje de acuerdo fue menor. En este sentido, los niveles de dificultad generados por el sistema y la dificultad percibida por los estudiantes no fueron completamente diferenciables, especialmente en la categoría difícil (**Figura 7.4**). Por ejemplo, hubo un 75 % de acuerdo entre los participantes para el nivel de preguntas de nivel fácil y el 25 % restante lo percibió como de nivel intermedio. Para las preguntas generadas de nivel intermedio, el porcentaje de acuerdo llegó a ser del 66.6 %, un 29.17 % acordó que fuera de nivel fácil y un 4.16 % de nivel difícil. Finalmente, para las preguntas generadas de nivel difícil, se obtuvo solamente un 16.66 % de acuerdo. El 62.5 % y el 20.83 % de acuerdo fué para las preguntas de nivel intermedio y fácil respectivamente. Uno de los motivos que explica este fenómeno se debe a que en preguntas de opción múltiple, es más fácil intuir la respuesta correcta a través de los distractores y, por tanto, la percepción es más optimista con la dificultad. Sin embargo, cabe recordar que el porcentaje de acuerdo no es usado como una métrica decisiva, ya que se basa solo en la percepción personal y esta es subjetiva. Para ello se usaron las respuestas de los estudiantes (**Figura 7.5**) y el modelo Rasch (**Figura 7.6**), en los cuales se puede observar basándose en los resultados que efectivamente sí existe una relación entre el nivel de dificultad propuesto y el nivel generado, aunque menos pronunciados.

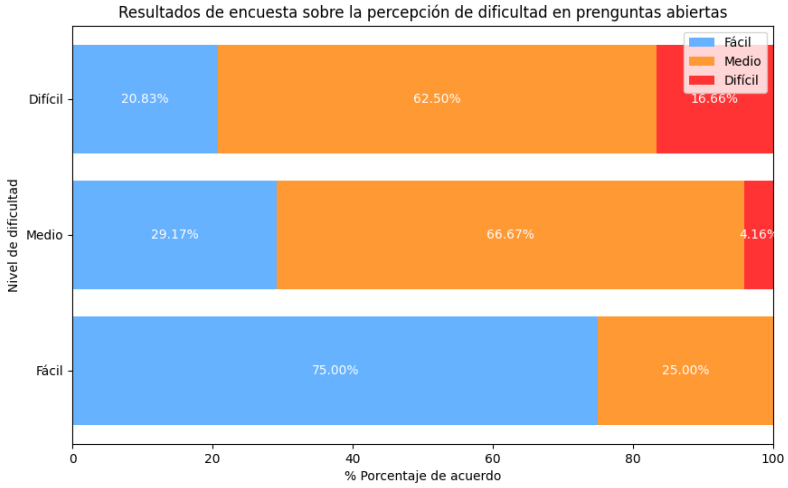



Figura 7.4: ***Resultados de la encuesta realizada con preguntas de opción múltiple sobre los niveles de dificultad percibidos en cada pregunta.***

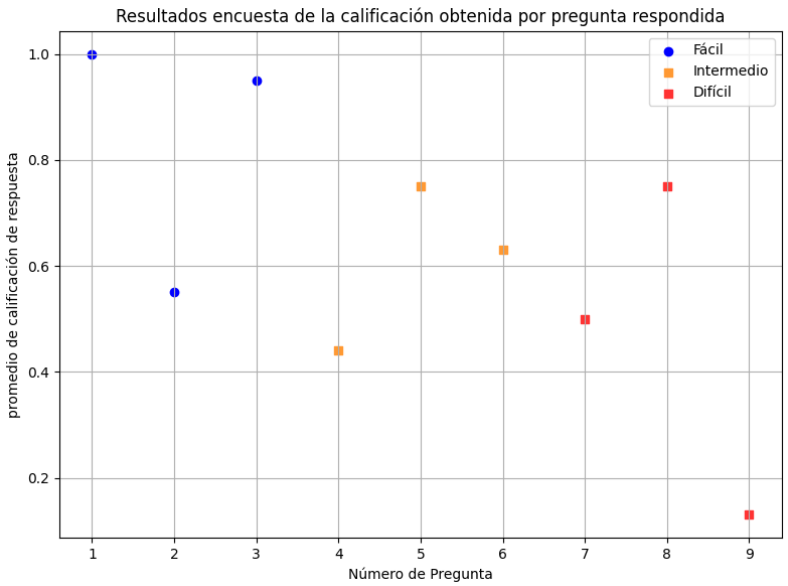



Figura 7.5: ***Resultados de la evaluación de respuestas a las preguntas de opción múltiple en la encuesta realizada.*** *Nótese que las preguntas fueron reordenadas y agrupadas, siendo del 1-3 fáciles, 4-6 intermedias y 7-9 difíciles. Nótese además el ruido y la dispersión en los resultados.*

En los resultados del modelo Rasch que se muestran en la **Figura 7.6**, podemos observar que la dificultad estimada para las categorías fácil y difícil del sistema SGEC, llegan a estar solapadas en los extremos con la categoría intermedio. Esto muestra a priori una menor diferenciación de la dificultad entre las categorías propuestas (en comparación a los resultados de preguntas abiertas).

En la caja asociada a la categoría fácil, por ejemplo, se observó un valor atípico correspondiente al ítem 2 de la encuesta, con una estimación del modelo Rasch positiva de 0.6, es decir también, de mayor dificultad. Aun así, el rango intercuartílico de la categoría fácil mantiene la mayoría de las estimaciones de dificultad en negativo y por debajo de la media. En el caso de las preguntas de categoría intermedio, se observó una distribución más simétrica con el rango intercuartílico cercano a la media, lo cual es plausible, ya que para este nivel se espera valores cercanos a 0. Finalmente, en la categoría difícil se observó otro valor atípico correspondiente al ítem 8 de la encuesta con estimación por debajo de la media. Sin embargo, el rango intercuartílico para las preguntas de categoría difícil mantiene la mayoría de las preguntas de este nivel con estimaciones Rasch positivas, extendiéndose desde 1.23 a 1.67.

Las respuestas de las preguntas de los valores atípicos, mencionados anteriormente, están fuertemente relacionados con la habilidad subyacente de los encuestados y a la calidad del material académico proporcionado con el cual se generó la pregunta. Por lo que se recomienda, pero se omite su investigación en profundidad.

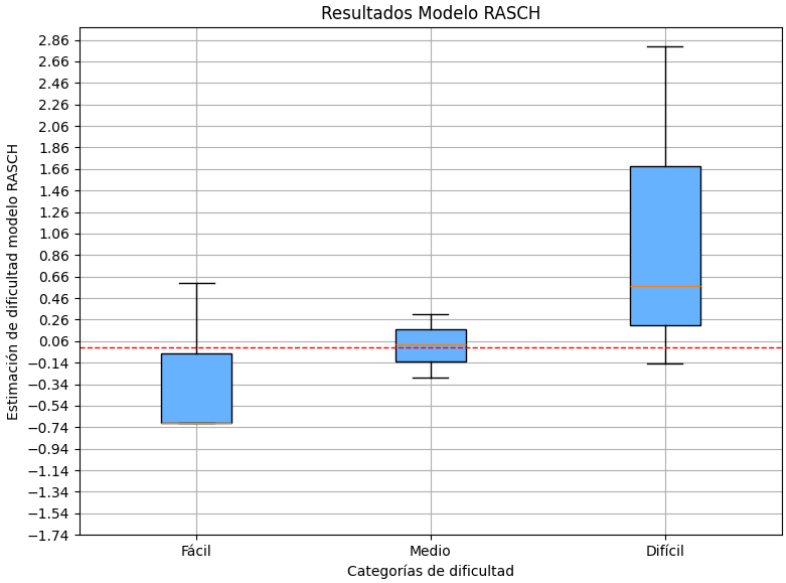



Figura 7.6: ***Resultados de las estimaciones de dificultad del modelo Rasch sobre la encuesta realizada de preguntas de opción múltiple.*** *Nótese el solapamiento entre categorías.*

## 7.2. Resultados del análisis sintáctico de preguntas

Como se puede observar en la **Tabla 7.1**, el análisis de frecuencia de verbos muestra los verbos dominantes y presentes en las preguntas de nivel fácil, intermedio y difícil. Podemos ver que los verbos de la categoría fácil (nombrar, identificar y definir) se sitúan en el nivel de Recordar de la taxonomía de Bloom. Los adverbios interrogativos del nivel fácil (“Qué” y “Cuál”) involucran implícitamente en la respuesta datos factuales, tal como se esperaba del nivel de conocimiento factual. Similarmente, en el nivel intermedio, los verbos predominantes (identificar, implementar, seguir, desarrollar), se situaron correctamente en el nivel de Aplicar y Recordar dentro de la taxonomía de Bloom. Y los adverbios interrogativos (“Cuáles” y “Cuál”) fueron bastante adecuados para el nivel de conocimiento asociado de tipo procedimental y conceptual. Por ejemplo, en este nivel intermedio se ha observado la siguiente estructura en varias preguntas: ¿Cuáles son los pasos para identificar ... ?.

Finalmente, dentro del nivel difícil, los verbos predominantes identificados (evaluar, hacer, considerar, justificar, considerar, etc.), también se ajustan correctamente al nivel de Evaluar y Analizar dentro de la taxonomía de Bloom. Se observó, además, que en la mayoría de preguntas, se usó el adverbio interrogativo “Cómo”, el cual es bastante apropiado para pedir una evaluación o crítica. Por ejemplo, ¿Cómo evaluar críticamente las...?, ¿Cómo se pueden evaluar ... ? (pregunta 3 y 6 de la encuesta realizada).

| | Nivel fácil | Nivel intermedio | Nivel difícil |
|---|---|---|---|
| **Verbos:** | | | |
| | nombrar | identificar | evaluar |
| | identificar | implementar | hacer |
| | definir | desarrollar | considerar |
| | relacionar | utilizar | replicar |
| | | presentar | justificar |
| | | diseñar | desarrollar |
| | | seguir | implementar |
| **Adverbios Interrogativos:** | | | |
| | Qué | Cuál | Cómo |
| | Cuál | Cuáles | Qué |
| **Promedio palabras:** | | | |
| | 15.25 | 24.13 | 30.19 |

Cuadro 7.1: **Resultados del análisis sintáctico en preguntas. Experimento realizado con el conjunto de prueba de 50 preguntas.**

## 7.3. Resultados de la evaluación de respuestas

### 7.3.1. Distractores

Como se puede observar en la **Figura 7.7**, los distractores de la categoría difícil presentan una similitud mayor en comparación a las otras dos categorías. Este hecho genera, en cierto porcentaje, una mayor dificultad al momento de diferenciar y elegir la opción correcta. A pesar de que la diferencia no es notablemente significativa, se ha visto que la similitud semántica no es una métrica absoluta que nos pueda servir además para medir la similitud contextual entre respuestas respecto a una pregunta. Por ejemplo, véase la siguiente pregunta correspondiente al ítem 6 de la encuesta realizada de preguntas de opción múltiple:

*"¿Qué desafío se enfrenta al enfoque de simulación computacional de la cognición humana para comprender mejor la mente humana y replicar sus capacidades en máquinas?"*

Las cuatro posibles respuestas (distractores) de este ítem, hablan de posibles desafíos, pero que no son similares semánticamente, empero si presentan efectivamente una similitud contextual entre respuestas. En este caso, la opción 2 (La complejidad del lenguaje natural) y la opción 4 (La falta de un marco teórico unificado para modelar la cognición) son respuestas válidas que responden a desafíos válidos (según el texto), y que además generan confusión o dificultad al momento de responder. Por lo que se esperaría una similitud alta, pero en la práctica no resultó ser así, por lo que no se consiguió totalmente representar la similitud que se buscaba usando la distancia del coseno. Sin embargo, como primera aproximación al desafío de la evaluación de distractores, podemos limitarnos a decir que, semánticamente, existe una leve relación de similitud entre distractores con respecto a cada nivel de dificultad.

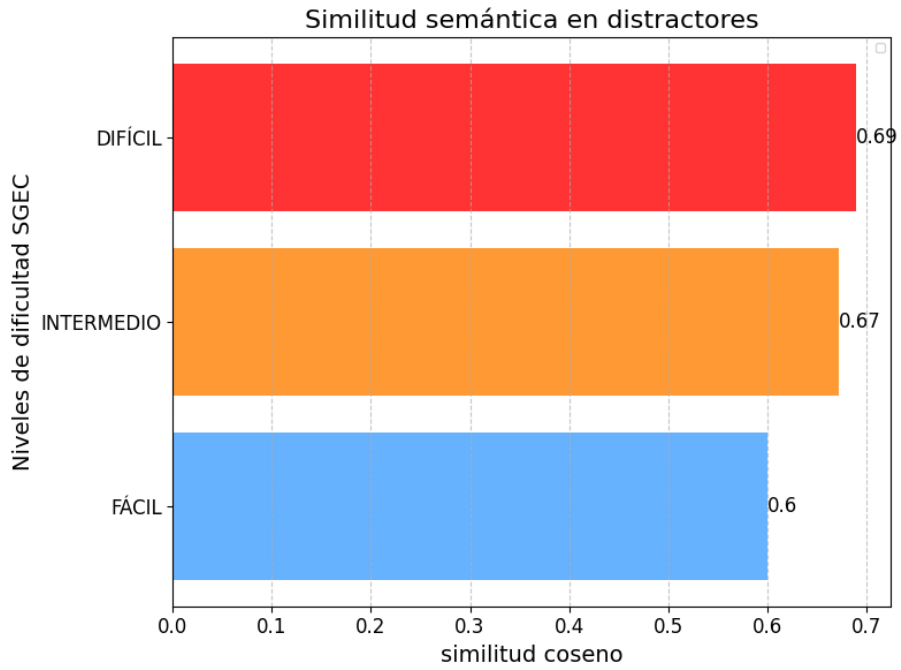



Figura 7.7: ***Resultados de la evaluación de distractores en función al nivel de dificultad SGEC de la encuesta realizada con preguntas de opción múltiple.*** *Las distancias de similitud calculadas son el promedio de las respuestas pertenecientes a cada categoría. Nótese que valores cercanos a 0 representan una similitud nula, mientras que valores cercanos a 1 indican una alta similitud.*

### 7.3.2. Precisión de la evaluación automática del sistema SGEC

Para la evaluación de la precisión en la puntuación de respuestas, se usó un conjunto de 50 preguntas de prueba respondidas correctamente (incluidas las de la encuesta realizada). Tras haber corregido y mejorado múltiples veces el modelo que se encarga de la evaluación automática, no se han vuelto a encontrar fallos en la puntuación de respuestas dentro del conjunto de prueba. Sin embargo, si se observó que en preguntas difíciles, el sistema en ocasiones puntuó respuestas correctas cortas con 0.5 puntos sin haber sido literalmente instruido para hacerlo. En este sentido, se puede decir que el sistema favorece a las respuestas largas y completamente desarrolladas por defecto. Este hecho se puede observar parcialmente en la **Figura 7.8**.

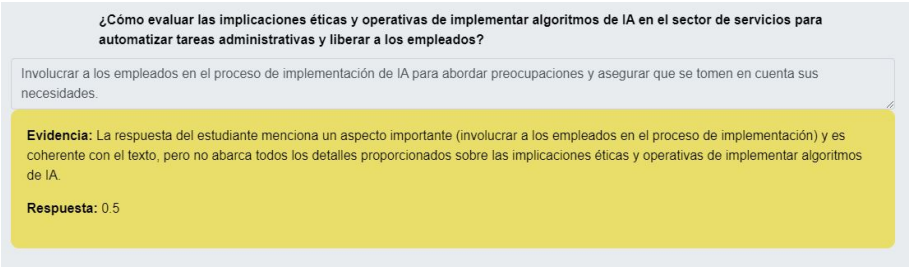



Figura 7.8: ***Ejemplo de evaluación automática, puntaje 0.5. Vista desde la aplicación Web.*** *(Pregunta difícil, respuesta y en amarillo la calificación del sistema SGEC). Nótese que a pesar de que la respuesta es correcta, el sistema identificó correctamente la incompletitud de la respuesta.*

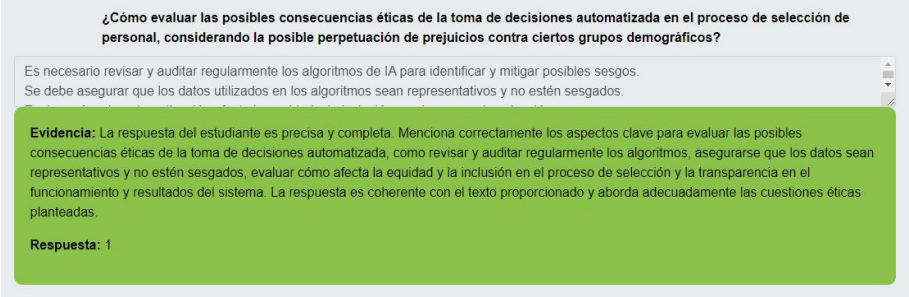



Figura 7.9: ***Ejemplo de evaluación automática, puntaje 1. Vista desde la aplicación Web.*** *Pregunta difícil, respuesta y en verde la calificación del sistema SGEC.*

# Capítulo 8

# Conclusiones y trabajos futuros

## 8.1. Conclusiones

En este trabajo se ha desarrollado y evaluado el Sistema Generador y Evaluador de Cuestionarios (SGEC), un sistema basado en el modelo de lenguaje preentrenado Llama3-8B para la generación personalizada de preguntas y evaluación automática de respuestas.

En primer lugar, el principal logro de este trabajo radica en que el sistema SGEC es capaz de generar preguntas que se alineen adecuadamente con las dimensiones cognitivas y de conocimiento de la taxonomía de Bloom mediante el uso de prompts y técnicas de aprendizaje contextual, concretamente few-shot. Esto permitió en última instancia controlar el nivel de dificultad deseado (fácil, intermedio y difícil), tal y como lo confirma dos encuestas realizadas en este trabajo.

Para el sistema SGEC, hemos creado seis nuevos modelos de lenguaje a partir del modelo Llama3-8B, que se corresponden con las combinaciones de niveles de dificultad propuestos (fácil, intermedio o difícil) y dos tipos de preguntas (opción múltiple o abierta). La agrupación de los nuevos niveles cognitivos y de conocimiento de la taxonomía de Bloom fue la siguiente:

- **Modelo dificultad fácil:** nivel cognitivo "recordar" y tipo de conocimiento "factual".
- **Modelo dificultad intermedio:** nivel cognitivo "entender" o "aplicar", y tipo de conocimiento "procedimental" o "conceptual".
- **Modelo dificultad difícil:** nivel cognitivo "analizar" o "evaluar" y tipo de conocimiento "conceptual".

En segundo lugar, podemos afirmar que el sistema SGEC demostró ser una herramienta efectiva para la evaluación de cuestionarios, proporcionando una puntuación y retroalimentación detallada y precisa sobre las respuestas de los estudiantes.

Para sustentar el cumplimiento de nuestros objetivos propuestos en este trabajo y las afirmaciones hechas sobre la aplicación SGEC, se presentan a continuación las conclusiones derivadas de la evaluación y análisis realizados en este trabajo.

1. **Preguntas abiertas**

La evaluación de las preguntas abiertas generadas por el SGEC demostró resultados prometedores en términos de dificultad percibida y estimada. Según la encuesta realizada con estudiantes universitarios, se encontró un alto nivel de acuerdo entre la dificultad percibida por los estudiantes y la dificultad generada por el sistema. Específicamente, para las preguntas de nivel fácil hubo un acuerdo del 95 %, mientras que para las preguntas de nivel intermedio y difícil los porcentajes de acuerdo fueron del 62.5 % y 79.12 %, respectivamente.

Además, al aplicar el modelo Rasch para la evaluación cuantitativa de la dificultad de las preguntas, se encontró una alineación satisfactoria entre la dificultad generada por el SGEC y las estimaciones de dificultad obtenidas por el modelo Rasch. Las preguntas de nivel fácil tuvieron estimaciones de dificultad negativas, indicando que fueron más fáciles que el promedio, las preguntas de nivel intermedio mantuvieron estimaciones cerca de la media, mientras que las preguntas de nivel difícil tuvieron estimaciones de dificultad positivas, indicando que fueron más desafiantes que el promedio.

2. **Preguntas de opción múltiple**

   El análisis de las preguntas de opción múltiple generadas por el SGEC reveló ciertos desafíos en la percepción de la dificultad. A partir de los resultados de la encuesta realizada con preguntas de opción múltiple, se observó que la percepción de dificultad por parte de los estudiantes no siempre coincidió con los niveles de dificultad establecidos por el sistema, especialmente en las preguntas de nivel difícil. El acuerdo para preguntas fáciles, intermedio y difíciles fue de un 75 %, 66.6 % y 16.6 % respectivamente. Sin embargo, al aplicar el modelo Rasch para una evaluación objetiva y cuantitativa, se observó una alineación de las categorías de dificultad propuestas por el sistema y las estimaciones del modelo Rasch, aunque presentaran solapamientos entre ellas. Este hecho indica una menor diferenciación de los niveles de dificultad generados en comparación con los resultados obtenidos para preguntas abiertas.

   Por tanto, aunque las preguntas de opción múltiple generadas por el SGEC muestran una alineación general con las expectativas de dificultad, se recomienda un análisis más profundo para mejorar y forzar la diferenciación de la dificultad de las preguntas de tipo opción múltiple.

3. **Análisis sintáctico de preguntas**

   El análisis sintáctico de las preguntas generadas por el SGEC utilizando la biblioteca Stanza demostró que los verbos y adverbios interrogativos utilizados se alinearon adecuadamente con la dimensión cognitiva y de conocimiento de la taxonomía de Bloom. Los verbos utilizados en las preguntas de nivel fácil coincidieron con los del nivel de la dimensión cognitiva (Recordar). Los verbos utilizados en las preguntas de nivel intermedio estaban mayoritariamente asociados con los niveles de Entender y Aplicar, mientras que las preguntas de nivel difícil utilizaron verbos asociados con Analizar y Evaluar, lo cual es consistente con la taxonomía de Bloom.

4. **Distractores**

   El análisis de la similitud semántica de los distractores de las preguntas de opción múltiple indicó que, en general, los distractores de las preguntas de nivel difícil mostraron una mayor

similitud semántica entre sí, lo que puede traducirse en un mayor desafío para el estudiante al momento de la elección de la respuesta correcta. A pesar de que la similitud semántica no captura todas las similitudes contextuales entre las respuestas, esta métrica proporcionó una buena aproximación para evaluar la efectividad de los distractores generados por el SGEC.

5. **Precisión de la evaluación automática del SGEC**

   En términos de precisión, el SGEC mostró una mejora significativa en la evaluación automática de respuestas, especialmente después de múltiples ajustes y correcciones mediante el ajuste fino de instrucciones. En este sentido, no se volvieron a encontrar fallos en el conjunto de 50 preguntas de prueba. Sin embargo, se observó que en preguntas difíciles, el sistema tendía a puntuar respuestas cortas y correctas con 0.5 puntos, favoreciendo así a las respuestas largas y completamente desarrolladas con 1 punto por defecto.

Por último y no menos importante, cabe mencionar que el sistema SGEC (en su primera versión) fue premiado en la categoría de aprendizaje potenciado mediante IA en las Jornadas de Innovación Educativa ETSISI 2024 (Universidad Politécnica de Madrid) y ha sido enviado al primer Congreso en Innovación Docente de las Universidades Madrileñas [11]. Esperamos finalmente que sea el comienzo de una versión más robusta y pueda ser utilizado en un entorno real universitario.

## 8.2. Trabajos Futuros

A pesar de los buenos resultados obtenidos durante el proceso de desarrollo de este proyecto, se han descubierto posibles investigaciones y mejoras para futuros trabajos. Entre ellas se describen las siguientes:

- Implementar un sistema de retroalimentación inteligente que permita ajustar automáticamente los niveles de dificultad de las preguntas basándose en los resultados de la evaluación de los estudiantes.
- Afinar la división semántica del texto para la generación de preguntas y optimizar los tiempos requeridos para la división semántica.
- El procesamiento de texto con imágenes sigue siendo un desafío persistente, por lo que un trabajo futuro es implementar soluciones para procesar texto con imágenes o tablas y aceptar, a la vez, otros formatos de archivos aparte de PDF.
- Cuantificar la precisión de las preguntas generadas dentro de un dominio específico e incursionar con otras técnicas de ajuste fino.
- Implementar una opción en la aplicación para procesar documentos muy largos mediante "retrival augmented generation" (RAG), la cual podría significativamente mejorar la calidad de preguntas, la relevancia y la exactitud de dificultad con grandes cantidades de texto.

# Bibliografía


[1] Automatización de Evaluaciones: Desarrollo de Cuestionarios a través de Procesamiento de Lenguaje Natural y Aprendizaje Automático. https://innovacioneducativa.upm.es/proyectos-ie/informacion?anyo=2023-2024&id=1196.

[2] Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. Synthetic qa corpora generation with roundtrip consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173, Florence, Italy, 2019. Association for Computational Linguistics.

[3] Jacopo Amidei, Paul Piwek, and Alistair Willis. Evaluation methodologies in automatic question generation 2013-2018. In *Proceedings of the 11th International Conference on Natural Language Generation (INLG '20)*, pages 307–317, Tilburg University, The Netherlands, 2018. Association for Computational Linguistics.

[4] Lorin W. Anderson and David A. Krathwohl. *A Taxonomy for Learning, Teaching and Assessing: A Revision of Bloom's*. Pearson Education, 2014.

[5] Paul Andree. Sgec. https://github.com/PaulAndree/SGEC, 2024. Version 1.0.

[6] Hangbo Bao, Li Dong, Furu Wei, et al. Unilmv2: Pseudo-masked language models for unified language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 642–652. PMLR, 2020.

[7] Benjamin S. Bloom. *The Taxonomy of Educational Objectives, the Classification of Educational Goals, Volume Handbook I: Cognitive Domain*. New York, 1956.

[8] Tom Brown, Benjamin Mann, Ryder, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901, 2020.

[9] Tom B Brown, Benjamin Mann, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[10] Joseph A Buckwalter, Robert Schumacher, Jay P Albright, and Richard R Cooper. Use of an educational taxonomy for evaluation of cognitive performance. *Journal of Medical Education*, 56:115–121, 1981.

[11] Universidad Autónoma de Madrid. Congreso de innovación educativa. https://innovaciondocente.uam.es/event/19, 2023. Accedido: 2024-06-24.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics:*

*Human Language Technologies (NAACL-HLT '19)*, pages 4171–4186, Minneapolis, Minnesota, USA, 2019. Association for Computational Linguistics.

[13] Bhuwan Dhingra, Manaal Faruqui, Ankur Parikh, et al. Handling divergent reference texts when evaluating table-to-text generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4884–4895, Florence, Italy, 2019. Association for Computational Linguistics.

[14] Li Dong, Nan Yang, Wenhui Wang, et al. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019*, pages 13042–13054, Vancouver, British Columbia, Canada, 2019.

[15] Xinya Du, Junru Shao, and Claire Cardie. Learning to ask: Neural question generation for reading comprehension. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352, Vancouver, Canada, 2017. Association for Computational Linguistics.

[16] Xinya Du, Junru Shao, and Claire Cardie. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL '17)*, pages 1342–1352, Vancouver, Canada, 2017. Association for Computational Linguistics.

[17] Antony W. Edwards and Lara Alcock. Using rasch analysis to identify uncharacteristic responses to undergraduate assessments, 2010. https://hdl.handle.net/2134/8848.

[18] Sabina Elkins, Ekaterina Kochmar, and Cheung. How teachers can use large language models and bloom's taxonomy to create educational quizzes. *arXiv preprint arXiv:2401.05914*, 2024.

[19] Yifan Gao, Lidong Bing, Piji Li, Irwin King, and Michael R. Lyu. Generating distractors for reading comprehension questions from real examinations. *CoRR*, abs/1809.02768, 2018.

[20] Jorge Osés Grijalba. Automatic data generation for multiple choice question answering. Master's thesis, Universidad Nacional de Educación a Distancia, Escuela Técnica Superior de Ingeniería Informática, Departamento de Lenguajes y Sistemas Informáticos, 2022.

[21] Prakhar Gupta, Cathy Jiao, Yeh, et al. Instructdial: Improving zero and few-shot generalization in dialogue through instruction tuning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 505–525, Abu Dhabi, United Arab Emirates, 2022. Association for Computational Linguistics.

[22] Michael Heilman and Noah A. Smith. Good question! statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 609–617, Los Angeles, California, 2010. Association for Computational Linguistics.

[23] Ayako Hoshino and Hiroshi Nakagawa. Webexperimenter for multiple-choice question generation. In *Proceedings of HLT/EMNLP 2005 Interactive Demonstrations (HLT/EMNLP '05)*, pages 18–19, Vancouver, British Columbia, Canada, 2005. Association for Computational Linguistics.

[24] Myo-Kyoung Kim, Rajul Patel, James Uchizono, and Lynn Beck. Incorporation of bloom's taxonomy into multiple-choice examination questions for a pharmacotherapeutics course. *American Journal of Pharmaceutical Education*, 76(6), 2012.

[25] Kalpesh Krishna and Mohit Iyyer. Generating question-answer hierarchies. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL '19)*, pages 2321–2334, Florence, Italy, 2019. Association for Computational Linguistics.

[26] Ghader Kurdi, Jared Leo, Bijan Parsia, Uli Sattler, and Salam Al-Emari. A systematic review of automatic question generation for educational purposes. *International Journal of Artificial Intelligence in Education*, 30, 2019.

[27] Igor Labutov, Sumit Basu, and Lucy Vanderwende. Deep questions without deep understanding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 889–898, Beijing, China, 2015. Association for Computational Linguistics.

[28] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, et al. Albert: A lite bert for self-supervised learning of language representations. In *Proceedings of the 8th International Conference on Learning Representations (ICLR '20)*, Addis Ababa, Ethiopia, 2020. OpenReview.net.

[29] Alon Lavie and Abhaya Agarwal. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation (WMT '07)*, pages 228–231, Prague, Czech Republic, 2007. Association for Computational Linguistics.

[30] Mike Lewis, Yinhan Liu, Naman Goyal, et al. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL '20)*, pages 7871–7880. Association for Computational Linguistics, 2020.

[31] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, 2004. Association for Computational Linguistics.

[32] David Lindberg, Fred Popowich, John Nesbit, and Phil Winne. Generating natural language questions to support learning online. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 105–114, Sofia, Bulgaria, 2013. Association for Computational Linguistics.

[33] Yinhan Liu, Myle Ott, Naman Goyal, et al. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[34] Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. Why we need new evaluation metrics for nlg. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2241–2252, Copenhagen, Denmark, 2017. Association for Computational Linguistics.

[35] Ulrike Pado. Question difficulty – how to estimate without norming, how to use for automated grading. pages 1–10, 01 2017.

[36] Edward J Palmer and Peter G Devitt. Assessment of higher order cognitive skills in undergraduate education: modified essay or multiple choice questions? *BMC Medical Education*, 7:49–55, 2007.

[37] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL '02)*, pages 311–318, Philadelphia, Pennsylvania, USA, 2002. Association for Computational Linguistics.

[38] Bhuwan Peng, Chunyuan Li, Pengcheng He, et al. Instruction tuning with gpt-4. 2023.

[39] Weizhen Qi, Yu Yan, Yeyun Gong, Liu, et al. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2401–2410. Association for Computational Linguistics, 2020.

[40] Questgen. Questgen: Ai powered question generator. `http://questgen.ai/`. Accessed: 2022-01-05.

[41] Quillionz. Quillionz - world's first ai-powered question generator. `https://www.quillionz.com/`. Accessed: 2022-01-05.

[42] Alec Radford, Jeffrey Wu, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[43] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000 questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP '16)*, pages 2383–2392, Austin, Texas, USA, 2016. Association for Computational Linguistics.

[44] Georg Rasch. *Probabilistic Models for Some Intelligence and Attainment Tests*. Danish Institute for Educational Research, Copenhagen, 1960.

[45] Henry L. Roediger III, Adam L. Putnam, and Megan A. Smith. Benefits of testing and their applications to educational practice. In Sarah J. Ross, editor, *Psychology of Learning and Motivation*, volume 55, pages 1–36. Academic Press, 2011.

[46] Melissa Roemmele, Deep Sidhpura, Steve DeNeefe, and Ling Tsou. Answerquest: A system for generating question-answer items from multi-paragraph documents. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations (EACL '21)*, pages 40–52, Online, 2021. Association for Computational Linguistics.

[47] Thibault Sellam, Dipanjan Das, and Ankur Parikh. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online, 2020. Association for Computational Linguistics.

[48] E.V. Smith and R.M. Smith, editors. *Introduction to Rasch Measurement: Theory, Models, and Application*. Journal of Applied Measurement Press Books, 2004.

[49] Amy Tiemeier, Zachary Stacy, and John Burke. Using multiple choice questions written at various bloom's taxonomy levels to evaluate student performance across a therapeutics sequence. *Innovations in Pharmacy*, 2(2), 2011.

[50] Hugo Touvron, Thibaut Lavril, Gautier Izacard, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[51] Adam Trischler, Tong Wang, Yuan, et al. Newsqa: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP (RepL4NLP '17)*, pages 191–200, Vancouver, Canada, 2017. Association for Computational Linguistics.

[52] Kristiyan Vachev, Momchil Hardalov, Karadzhov, et al. Leaf: Multiple-choice question generation. In *European Conference on Information Retrieval*, pages 321–328, Cham, 2022. Springer International Publishing.

[53] Wim van der Linden. Item response theory. In Penelope Peterson, Eva Baker, and Barry McGaw, editors, *International Encyclopedia of Education*, pages 81–88. Elsevier Ltd., 2010.

[54] Brian von Konsky, Longwei Zheng, Eric Parkin, Simon Huband, and David Gibson. Parts of speech in bloom's taxonomy classification. 11 2018.

[55] Brian R von Konsky, Lan Zheng, Emma Parkin, Simon Huband, and David Gibson. Parts of speech in bloom's taxonomy classification. In M Campbell, J Willems, C Adachi, and Blake, editors, *Open Oceans: Learning without borders. Proceedings ASCILITE 2018 Geelong*, pages 527–532, 2018.

[56] Jason Wei, Maarten Bosma, et al. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

[57] Jason Wei, Maarten Bosma, Zhao, et al. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2021.

[58] Dongling Xiao, Han Zhang, Yu-Kun Li, et al. Ernie-gen: an enhanced multi-flow pre-training and fine-tuning framework for natural language generation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI '20)*, pages 3997–4003. ijcai.org, 2020.

[59] Shunyu Yao, Dian Yu, et al. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

[60] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

[61] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2019.

[62] Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. Neural question generation from text: A preliminary study. In *Natural Language Processing and Chinese Computing*, pages 662–671, Cham, 2018. Springer International Publishing.

# Anexo

## .1. Anexo 1. Cuestionario preguntas opción múltiple

| | |
|---|---|
| **PREGUNTA 1** | ¿Quién fundó la frenología? |
| **OPCIÓN 1** | René Descartes |
| **OPCIÓN 2** | Immanuel Kant |
| **OPCIÓN 3** | Jean-Jacques Rousseau |
| **OPCIÓN 4** | Franz Joseph Gall |

| | |
|---|---|
| **PREGUNTA 2** | ¿Cuál fue el método utilizado por Franz Joseph Gall para identificar los órganos cerebrales? |
| **OPCIÓN 1** | La lectura de las palmas |
| **OPCIÓN 2** | El análisis de la forma del cráneo |
| **OPCIÓN 3** | La medición del tamaño del cerebro |
| **OPCIÓN 4** | La identificación de protuberancias reveladoras en el cráneo |

| | |
|---|---|
| **PREGUNTA 3** | ¿Qué crítica se puede hacer a la frenología fundada por Franz Joseph Gall sobre la identificación de órganos cerebrales desde el exterior mediante protuberancias en el cráneo? |
| **OPCIÓN 1** | Se puede criticar que no tiene en cuenta la complejidad del cerebro |
| **OPCIÓN 2** | Se puede criticar que no es un método efectivo para diagnosticar enfermedades mentales |
| **OPCIÓN 3** | Se puede criticar que no se basa en evidencias científicas |
| **OPCIÓN 4** | Se puede criticar que la noción del tamaño como índice del poder o de la energía de una determinada facultad mental es graciosamente errónea |

| | |
|---|---|
| **PREGUNTA 4** | ¿Qué campo de estudio combina neurociencia y inteligencia artificial? |
| **OPCIÓN 1** | Psicología computacional |
| **OPCIÓN 2** | Ingeniería biomédica |
| **OPCIÓN 3** | Lenguaje natural |
| **OPCIÓN 4** | Neurociencia cognitiva |

| | |
|---|---|
| **PREGUNTA 5** | ¿Cuál es el objetivo principal de la simulación computacional de la cognición humana? |
| **OPCIÓN 1** | Estudiar la estructura del cerebro humano |
| **OPCIÓN 2** | Desarrollar un sistema de inteligencia artificial |
| **OPCIÓN 3** | Crear una máquina que imite el comportamiento humano |
| **OPCIÓN 4** | Comprender mejor la mente humana y replicar sus capacidades en máquinas |

| | |
|---|---|
| **PREGUNTA 6** | ¿Qué desafío se enfrenta al enfoque de simulación computacional de la cognición humana para comprender mejor la mente humana y replicar sus capacidades en máquinas? |
| **OPCIÓN 1** | La falta de datos precisos sobre el cerebro humano |
| **OPCIÓN 2** | La complejidad del lenguaje natural |
| **OPCIÓN 3** | La limitación de los recursos computacionales |
| **OPCIÓN 4** | La falta de un marco teórico unificado para modelar la cognición |

| | |
|---|---|
| **PREGUNTA 7** | ¿Cuál fue el año en que se publicó un estudio relacionado con la vacuna de la triple vírica y el autismo? |
| **OPCIÓN 1** | 1995 |
| **OPCIÓN 2** | 2000 |
| **OPCIÓN 3** | 2010 |
| **OPCIÓN 4** | 1998 |

| | |
|---|---|
| **PREGUNTA 8** | ¿Cuál es el principal argumento utilizado por los grupos antivacunas para oponerse al uso de vacunas? |
| **OPCIÓN 1** | La falta de efectos secundarios negativos |
| **OPCIÓN 2** | La creación de enfermedades infecciosas |
| **OPCIÓN 3** | La mala aplicación de protocolos de aplicación |
| **OPCIÓN 4** | Inflar los efectos secundarios perniciosos y minimizar las complicaciones de las enfermedades infecciosas |

| | |
|---|---|
| **PREGUNTA 9** | ¿Qué tipo de información se utiliza para evaluar los efectos secundarios negativos de las vacunas y distinguirlos de los falsos alarmistas? |
| **OPCIÓN 1** | Evidencia científica y datos estadísticos |
| **OPCIÓN 2** | Análisis de la composición química de las vacunas |
| **OPCIÓN 3** | Opiniones de expertos en medicina |
| **OPCIÓN 4** | Estudios rigurosamente diseñados y publicados en revistas médicas |

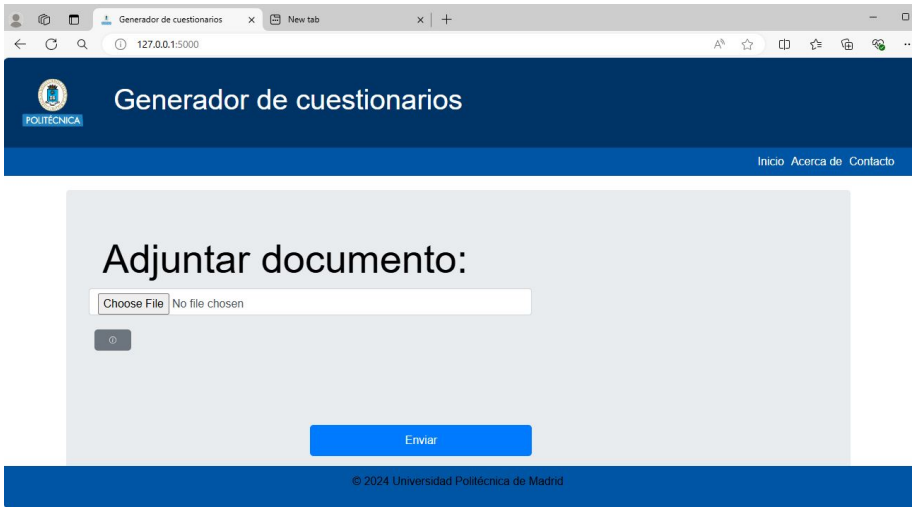
## .2. Anexo 2. Cuestionario preguntas abiertas

| | |
|---|---|
| **PREGUNTA 1** | ¿Quién fundó la frenología a finales del siglo XVIII en Europa? |
| **PREGUNTA 2** | ¿Cuál fue el método utilizado por Franz Joseph Gall para identificar los órganos cerebrales? |
| **PREGUNTA 3** | ¿Cómo evaluar críticamente las afirmaciones de Gall sobre la relación entre el tamaño cerebral y la facultad mental, considerando su influencia en la frenología y su impacto en la comprensión actual de la neurociencia? |
| **PREGUNTA 4** | ¿Qué paradigma se gesta en la intersección de neurociencia cognitiva e inteligencia artificial? |
| **PREGUNTA 5** | ¿Cuál es el objetivo principal de la simulación computacional de la cognición humana? |
| **PREGUNTA 6** | ¿Cómo se pueden evaluar los desafíos formidables que enfrenta el enfoque de simulación computacional de la cognición humana, como la complejidad del cerebro humano y la falta de un marco teórico unificado para modelar la cognición? |
| **PREGUNTA 7** | ¿En qué año se publicó un estudio en la revista médica The Lancet que relacionaba la vacuna de la triple vírica con el autismo? |
| **PREGUNTA 8** | ¿Cuál es el principal argumento utilizado por los grupos antivacunas para oponerse al uso de vacunas? |

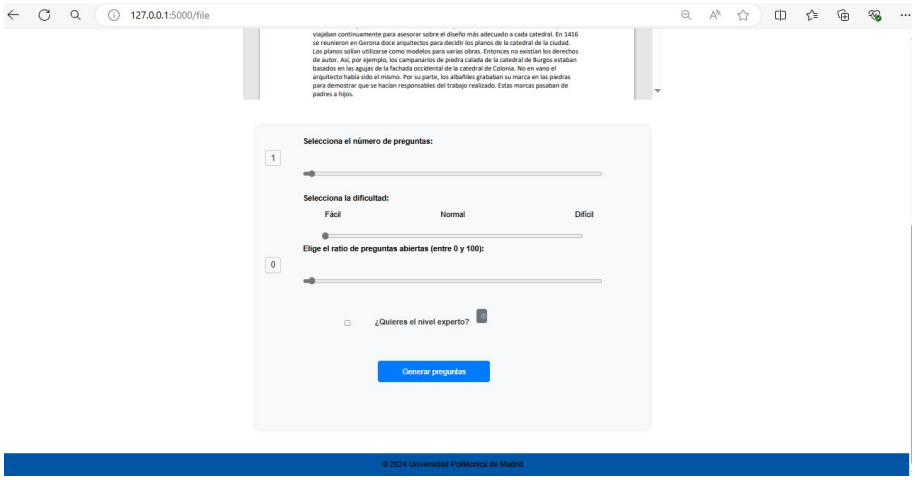
| | |
|---|---|
| **PREGUNTA 9** | ¿Cómo evaluar críticamente las acusaciones de efectos secundarios perniciosos en vacunas, considerando la falta de evidencia científica y el sesgo mediático en grupos antivacunas? |

## .3. Anexo 3. Vista de la interfaz gráfica del usuario (frontend)

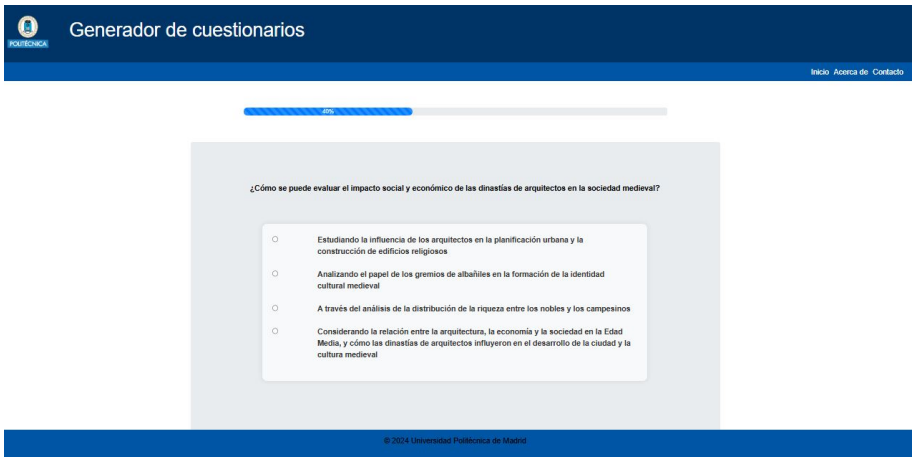
**Pantalla de Inicio:**



**Personalización de preguntas:**




**Generación de cuestionario:**



# .4. Anexo 4. Código en Python para combinar oraciones con un tamaño de ventana ajustable

```python
def combine_sentences(sentences, buffer_size=1):
    for i in range(len(sentences)):
        combined_sentence = " "
        for j in range(i - buffer_size, i):
            if j >= 0:
                combined_sentence += sentences[j]["sentence"] + " "
        combined_sentence += sentences[i]["sentence"]

        for j in range(i + 1, i + 1 + buffer_size):
            if j < len(sentences):
                combined_sentence += " " + sentences[j]["sentence"]
        sentences[i]["combined_sentence"] = combined_sentence
    return sentences
sentences = combine_sentences(sentences)
```