



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

**GENERACIÓN DE REACTIVOS CON LA AYUDA DE GPT**

TESINA PARA OBTENER EL TÍTULO DE LICENCIATURA EN INGENIERÍA EN CIENCIAS  
DE LA COMPUTACIÓN

PRESENTA:

SAID CONSTANTINO CARCAMO GARCIA

ASESORA: DRA. ETELVINA ARCHUNDIA SIERRA

PUEBLA PUE., MÉXICO 2024

## ÍNDICE

Capítulo 1. Introducción	5
1.1. Planteamiento del problema	6
1.2. Objetivo general	6
1.2.1. Objetivo específico	6
1.3. Justificación	7
Capítulo 2. Estado del arte y teorías	8
2.1 La Inteligencia Artificial	9
2.2 La Inteligencia Artificial en la Educación	13
2.2.1 Plataformas de Aprendizaje mediante la IA	14
2.3 Evaluación del aprendizaje	16
2.3.1 Tipos de reactivos	17
2.3.2 Reactivos de opción múltiple	17
2.4 Ingeniería de software	18
2.5 Casos de Uso	19
2.5.1 Relaciones de los casos de uso	19
2.5.2 Notación de Casos de Uso	20
Capítulo 3. Análisis y diseño	22
3.1. Requerimientos casos de uso	23
3.2. Análisis	23
3.3. Diseño	28
Capítulo 4. Implementación y pruebas del sistema	41
4.1. Pruebas	42
Conclusiones y trabajos futuros	51

## ÍNDICE FIGURAS

Figura 1. Elementos que integran un Caso de Uso.	21
Figura 2. Casos de uso de generación de reactivos.	26
Figura 3. Interacción de frontend con servicio backend y base de datos.	27
Figura 4. Resultado del primer intento.	34
Figura 5. Resultado del segundo intento.	35
Figura 6. Resultado del tercer intento.	36
Figura 7. Resultado del cuarto intento.	37
Figura 8. Resultado del Quinto intento.	38
Figura 9. Resultado del sexto intento.	39
Figura 10. Resultado del séptimo intento el cual fue exitoso.	40
Figura 11. Se realiza una consulta http de los reactivos que se tienen almacenados.	43
Figura 12. Se realiza una consulta http para generar un nuevo reactivo.	44
Figura 13. Interfaz principal sin reactivos previos.	44
Figura 14. Formulario para solicitar un nuevo reactivo.	45
Figura 15. Formulario de nuevo reactivo, con datos.	46
Figura 16. Resultado del reactivo.	46
Figura 17. Interfaz principal con el listado de reactivos disponibles o existentes.	48
Figura 18. Interfaz principal con el listado de reactivos disponibles o existentes, se muestra los reactivos que ya fueron revisados.	49

## ÍNDICE TABLAS

Tabla 1. Unesco (2021, p12). Inteligencia artificial y educación. Guía para personas para a cargo de formular políticas. <a href="https://unesdoc.unesco.org/ark:/48223/pf0000379376">https://unesdoc.unesco.org/ark:/48223/pf0000379376</a>	13
Tabla 2. Estructura de la opción múltiple.	28
Tabla 3. Estructura del Reactivo.	30
Tabla 4. Estructura del Usuario.	31
Tabla 5 Estructura del examen.	31
Tabla 6. Estructura de una pregunta de examen.	32

# Capítulo 1. Introducción

## **1.1. Planteamiento del problema**

La evaluación del aprendizaje aunado con el potencial de la Inteligencia Artificial (IA) para generar reactivos de distintos temas en una base de datos dinámica para almacenar preguntas para medir el avance del aprendizaje; la utilización de modelos de IA avanzados como GPT-3 Y GPT4 ofrece una solución prometedora para el desarrollo de reactivos.

La investigación está basada en la colaboración de académicos y expertos en distintas áreas para que la gama de reactivos sea amplia y de una alta calidad. Los modelos GPT-3 y GPT-4 actúan como herramientas para crear y generar las preguntas, así ayudando a agilizar el proceso.

Estos modelos permiten la adaptabilidad a la hora de crear los reactivos, ya que pueden incrementar su dificultad, cambiar tema de interés o el enfoque e incluso el estilo de las preguntas, claro que, aunque estemos realizando los reactivos con ayuda de IA necesitamos la parte humana, de expertos y académicos para la validación y la retro alimentación de los reactivos generados, ya que esto garantiza la precisión y valor en su mejora continua.

Estos reactivos, cada vez que sean aprobados por expertos y académicos en colaboración serán almacenados en una base de datos que se ira enriqueciendo con cada reactivo aprobado y generándose una fuente valiosa de reactivo para exámenes de distintos temas o áreas.

## **1.2. Objetivo general**

Diseño de reactivos mediante la inteligencia artificial como herramienta de evaluación para docentes para diversos temas y áreas.

### **1.2.1. Objetivo específico**

- Requerimientos sobre reactivos para la evaluación e IA.
- Análisis y diseño de los reactivos en IA.
- Desarrollo de los reactivos aplicando IA.
- Resultados y conclusiones

### **1.3. Justificación**

El planteamiento del problema en el proyecto destaca la necesidad de aprovechar el potencial de la inteligencia artificial (IA) para generar reactivos de distintos temas y crear una base de datos dinámica de preguntas de evaluación. A continuación, se justifican los aspectos clave del proyecto:

**Aprovechamiento de la Inteligencia Artificial (IA):**

Se reconoce la capacidad de modelos avanzados de inteligencia artificial, como GPT-3 y GPT-4, para generar preguntas de evaluación de manera eficiente y efectiva.

La implementación de IA permite la automatización del proceso de creación de reactivos, optimizando tiempos y recursos.

**Colaboración Interdisciplinaria:**

El proyecto se basa en la colaboración de académicos y expertos en diversas áreas, asegurando una amplia gama de reactivos de alta calidad. La diversidad de conocimientos aportados por expertos de diferentes campos enriquece la variedad y profundidad de las preguntas generadas.

**Adaptabilidad de los Reactivos:**

Los modelos GPT-3 y GPT-4 permiten la adaptabilidad en la creación de reactivos, facilitando ajustes en la dificultad, temas de interés, enfoque y estilo de las preguntas. Esta flexibilidad garantiza la capacidad de generar preguntas adecuadas para diferentes niveles de dificultad y áreas de conocimiento.

**Validación Humana:**

A pesar de la generación automática, se reconoce la importancia de la validación humana para garantizar la precisión y calidad de los reactivos generados.

La retroalimentación humana se incorpora como un proceso constante, asegurando mejoras continuas en la generación de preguntas y adaptándose a las necesidades específicas de los usuarios.

**Generación de una Base de Datos Dinámica:**

Los reactivos aprobados por la validación humana se almacenan en una base de datos dinámica, enriqueciéndose con cada nuevo reactivo aprobado.

La base de datos resultante se convierte en una fuente valiosa de reactivos para exámenes y evaluaciones en distintos temas y áreas, brindando una solución integral y sostenible.

## Capítulo 2. Estado del arte y teorías



## 2.1 La Inteligencia Artificial

La definición más sencilla y generalizada de Inteligencia Artificial (IA) la concibe como máquinas capaces de emular determinadas funcionalidades de la inteligencia humana, incluyendo percepción, aprendizaje, razonamiento, resolución de problemas, interacción lingüística y producción creativa [1]. La IA comprende los conocimientos recibidos por las máquinas a través de la experiencia, su capacidad de adaptarse a nuevas aportaciones y su habilidad para realizar tareas de manera similar a las personas [2]. A una IA se la entrena a base de texto, se le hacen preguntas y se le añade información, de manera que este sistema, a base de correcciones a lo largo del tiempo, va entrenándose para realizar de forma automática la tarea para la que ha sido diseñada. Se ha convertido en una realidad presente en todos los ámbitos de la sociedad, en este caso en particular nos enfocamos en su aporte en la educación.

La UNESCO sostiene que la IA proporciona el potencial necesario para abordar algunos de los desafíos mayores de la educación actual, innovar las prácticas de enseñanza y aprendizaje y acelerar el progreso para la consecución del ODS 4 [3]. Según Rouhiainen la IA es la capacidad de las máquinas para usar algoritmos, aprender de los datos y utilizar lo aprendido en la toma de decisiones tal y como lo haría un ser humano [4]. Lo anterior, supone la creación de una herramienta con capacidad de realizar las mismas tareas que un ser humano, pero reduciendo significativamente los errores, imprecisiones y mejorando la productividad.

Las tecnologías de IA se utilizan cada vez más en los servicios que se detallan de la siguiente manera (véase tabla 1) [5]:

- Procesamiento del lenguaje natural (PLN) El uso de la IA para interpretar textos de forma automática incluido el análisis semántico (como el que se utiliza en los servicios jurídicos y la traducción), y la generación de textos (como en el periodismo automatizado).
- Reconocimiento del habla. La aplicación del PLN a la palabra hablada, incluyendo a los teléfonos inteligentes, los asistentes personales de IA y los chatbots de servicios bancario
- Reconocimiento y procesamiento de imágenes. El uso de la IA para el reconocimiento facial (por ejemplo, para los pasaportes electrónicos); el reconocimiento de la escritura

a mano (por ejemplo, para la clasificación automatizada de correo postal); la manipulación de imágenes (por ejemplo, para Deep fakes); y los vehículos autónomos.

- Agentes automatizados. El uso de la IA en los avatares de los juegos de computadora, los bots de software malicioso, los acompañantes virtuales, los robots inteligentes y la guerra automatizada
- Detección de afecto. El uso de la IA para analizar sentimientos en textos, comportamientos y rostros.
- Minería de datos para predicciones. El uso de la IA para diagnósticos médicos, previsiones meteorológicas, proyecciones empresariales, ciudades inteligentes, predicciones financieras y detección del fraude.
- Creatividad artificial. El uso de la IA en sistemas que pueden crear nuevas fotografías, música, obras de arte o historia.

	<b>DETALLES</b>	<b>PRINCIPALES TÉCNICAS DE IA</b>	<b>DESARROLLO</b>	<b>EJEMPLOS</b>
<b>Procesamiento del lenguaje natural (PLN)</b>	El uso de la IA para generar textos de forma automática (como en el periodismo automatizado), incluido el análisis semántico (como el que se utiliza en los servicios jurídicos y la traducción).	Aprendizaje automático (especialmente el aprendizaje profundo), regresión y K-medias	La PLN, el reconocimiento del habla y el reconocimiento de imágenes han logrado una precisión superior al 90%. Sin embargo, algunos investigadores sostienen que, incluso con más datos y procesadores más rápidos, esto no mejorará mucho hasta que se desarrolle un nuevo paradigma de IA.	ChatGPT <a href="https://openai.com/chatgpt/">https://openai.com/chatgpt/</a>

Reconocimiento del habla	La aplicación de la PLN a la palabra hablada, incluidos los teléfonos inteligentes, los asistentes personales y los chatbots de servicios bancarios.	Aprendizaje automático, especialmente un enfoque de redes neuronales recurrentes de aprendizaje profundo llamado memoria a corto plazo (GMCP).		Alibaba Cloud <a href="https://www.alibabacloud.com">https://www.alibabacloud.com</a>
<b>Reconocimiento y procesamiento de imágenes</b>	Incluye el reconocimiento facial (por ejemplo, para los pasaportes electrónicos); el reconocimiento de la escritura a mano (por ejemplo, para la clasificación automatizada de correo postal); la manipulación de imágenes (por ejemplo, para deepfakes); y los vehículos autónomos.	Aprendizaje automático, especialmente redes neuronales convolucionales de aprendizaje profundo.		Google Lens <a href="https://lens.google/intl/es-419/">https://lens.google/intl/es-419/</a>
<b>Agentes automatizados</b>	Incluye los avatares de los juegos de computadora, los bots de software malicioso, los acompañantes virtuales, los robots inteligentes y	GOFAI y aprendizaje automático (por ejemplo, redes neuronales autoorganizadas de aprendizaje profundo,	Los esfuerzos de investigación se centran en la inteligencia emergente, la actividad coordinada, la ubicación y la incorporación física, inspirados en formas	Woebot <a href="https://woebothealth.com/">https://woebothealth.com/</a>

	la guerra automatizada .	aprendizaje evolutivo y aprendizaje por refuerzo).	biológicas más simples.	
<b>Detección de afecto</b>	Incluye análisis de sentimientos en textos, comportamientos y rostros.	Redes bayesianas y aprendizaje automático, especialmente el aprendizaje profundo.	Se están desarrollando múltiples productos en todo el mundo; sin embargo, su uso normalmente es controversial.	Afectiva <a href="https://www.affective.com/">https://www.affective.com/</a>
<b>Minería de datos para predicciones</b>	Incluye predicciones financieras, detección de fraudes, diagnóstico médico, previsiones meteorológicas, procesos empresariales y ciudades inteligentes.	Aprendizaje automático (especialmente aprendizaje supervisado y profundo), redes de Bayes y máquinas de vectores de soporte.	Las aplicaciones de la minería de datos están creciendo exponencialmente, desde la predicción de compras hasta la interpretación de señales ruidosas en electroencefalograma (EEG).	Proyecto de investigación Proyecto de investigación: NeurIPS (Neural Information Processing Systems) Sitio web: <a href="https://nips.cc/">https://nips.cc/</a> ICML (International Conference on Machine Learning) Sitio web: <a href="http://icml.cc/">http://icml.cc/</a>

<b>Creatividad artificial</b>	Incluye sistemas que pueden crear nuevas fotografías, música, obras de arte o historias.	Las redes generativas antagónicas (RGA), un tipo de aprendizaje profundo que implica dos redes neuronales enfrentadas entre sí. Modelos lingüísticos autorregresivos que utilizan el aprendizaje profundo para producir textos similares a los humanos.	Las RGA se encuentran a la vanguardia de la IA, por lo que sus aplicaciones futuras recién comienzan a hacerse evidentes. Un modelo lingüístico autorregresivo conocido como GPT-3 puede producir un texto impresionantemente parecido al humano. Sin embargo, contrariamente a lo que podría parecer, el sistema no entiende el texto que produce.	GPT-3 (Brown et al., 2020) <a href="https://openai.com/chatgpt/">https://openai.com/chatgpt/</a>
-------------------------------	--	---	---	---

Tabla 1. Unesco (2021, p12). Inteligencia artificial y educación. Guía para personas para a cargo de formular políticas. <https://unesdoc.unesco.org/ark:/48223/pf0000379376>

## 2.2 La Inteligencia Artificial en la Educación

La educación es un fenómeno multidimensional, conformada por diversas acciones que promueven el aprendizaje integral del ser humano; testigos de diversos procesos, desde la relación pedagógica entre aprendientes donde se produce el proceso de enseñanza/aprendizaje, hasta las relaciones de carácter político-administrativo que dan orden y gerencia al proceso educativo. La IA en entornos educativos facilita y promueve la creación de procesos de enseñanza/aprendizaje flexibles y adaptados a las necesidades de los involucrados, permite la aplicación de ambientes inteligentes de aprendizaje, los cuáles se caracterizan por su flexibilidad, adaptación y autonomía [6]. Su impacto en el mundo de la educación está siendo cada vez más relevante en una multiplicidad de desarrollos, desde el campo de la analítica de datos hasta el reconocimiento facial de las expresiones de los aprendientes [7]. Esto ha llevado a la comunidad científica a centrar esfuerzos en un nuevo campo del conocimiento denominado Inteligencia Artificial en la

Educación (IAEd). La IAEd está afectando de manera directa la transformación educativa revolucionando los métodos de enseñanza, las formas de aprender, de acceder al conocimiento y de capacitar a los docentes [8]. La tecnología de IA ofrece posibilidades prácticamente ilimitadas a la educación, entre las aplicaciones de la IAEd más populares encontramos: chatbots, modelos de toma de decisiones y tutores inteligentes [9]. Estas aplicaciones involucran el uso de diversas tecnologías, como aprendizaje automático, mapeo del conocimiento, procesamiento del lenguaje natural [10], web semántica, minería de datos, aprendizaje profundo [11], inteligencia de negocio, Big Data y analíticas [12]. Existen múltiples investigaciones en tecnología educativa que han utilizado IA en materias de educación, con resultados satisfactorios para ayudar a los aprendientes a mejorar su rendimiento académico [13].

### **2.2.1 Plataformas de Aprendizaje mediante la IA**

Entre estas herramientas encontramos plataformas de aprendizaje, asistentes virtuales, creadores de contenido, plataformas de enseñanza personalizada, diseño de planeación y gestión educativa, creación de rúbricas, chats. Cada una ofrece diferentes características y enfoques por lo que es necesario investigar las características a detalle antes de elegir la que mejor se adapte a las necesidades de la clase.

Una clasificación simple de la variedad de chatbots que existen [14], podría ser:

- **Chatbots Genéricos:** Modelan conocimiento genérico desde un punto de vista holístico. Entre los más populares destacan: Youchat, Jasperchat, Google Bard, Chatsonic, Perplexity y ChatGPT. Herramientas potentes que con la gestión adecuada pueden sumar de forma positiva en el contexto educativo.
- **Chatbots Especifico:** Modelan conocimiento de un dominio en particular, para aprender idiomas, para la enseñanza de las matemáticas, para aprender lenguajes de programación y para aprender base de datos.

Para aplicar estas herramientas de IA es importante comenzar con habilidades que permitan a los estudiantes adoptar e incorporar gradualmente estrategias de aprendizaje y enseñarles a ser conscientes de cómo aprenden para que enfrenten diversas situaciones de aprendizaje. Es en este punto donde el docente debe acompañarlos, guiarlos y orientarlos para fortalecer su autoestima y aumentar la confianza en sí mismos.

Según OpenAI (2024) [15], algunas herramientas de IA tipo chat que pueden ser útiles para estudiantes como complemento en su aprendizaje son: Brainly, Eva Chatbot de la UNAM, Chatterbot Eliza, Soy Tu Profe, Aprendiz, Tia Tutor Interactivo de Aprendizaje, Mia Educación, y UVirtual.

Chatbot. ChatGPT 3.5, también conocido como Chat Generative Pre-trained Transformer, es un modelo de IA y procesamiento de lenguaje natural (PLN) cuya función es procesar texto, entender lo que se le está preguntando y proporcionar respuestas en lenguaje natural, utilizando algoritmos de aprendizaje entrenados con grandes volúmenes de datos, es importante su capacidad de utilizar el lenguaje para responder lo que el usuario pide, desde la creación de contenido, la atención al cliente e inclusive la programación [16].

La plataforma ChatGPT 3.5 informa que su conocimiento está limitado a eventos anteriores a enero de 2022. Su alcance abarca una amplia gama de tareas relacionadas con el procesamiento del lenguaje natural, como responder preguntas, generar texto, traducir entre idiomas, completar texto y más. Sin embargo, notifica que tiene limitaciones en situaciones que requieren un conocimiento muy específico o contexto especializado.

Esta herramienta se ha establecido como una tecnología disruptiva que está transformando la manera en que se enseña, promueve el aprendizaje y brinda apoyo a los estudiantes en entornos académicos [17].

Otras aplicaciones educativas con IA se mencionan a continuación:

- Thinkster Math. Es una plataforma la cual da seguimiento al proceso que realizan los estudiantes para resolver problemas matemáticos y los compara con los profesores o tutores que personalizan los programas de aprendizaje para los estudiantes con base en los fortalezas y debilidades de estos, para esto utiliza la ayuda de la IA y el aprendizaje automático [18].
- Brainly. Es una plataforma que permite que las personas se puedan educar a través de la transmisión de enseñanzas esta plataforma tiene una gran relevancia por el beneficio que da a los padres estudiantes y maestros, la compañía a la cual pertenece Brainly desarrolla distintos algoritmos de aprendizaje automático y de IA con los cuales filtra el contenido de dicha plataforma [19].
- CTI. Content Technologies INC empresa que se dedica al desarrollo e investigación de la IA que se especializa en la automatización de la producción, de procesos comerciales, el diseño instruccional y las soluciones de aplicaciones de

contenido. Cuenta con la herramienta Cram 101 que con la implementación de IA puede generar una guía de estudio inteligente completa con resúmenes de capítulos, pruebas de falso o verdadero ilimitadas, pruebas de opción múltiple, tarjetas didácticas y todo esto a partir de un libro de texto [20].

- Gradescope. Es una plataforma que utiliza la IA para agrupar y ordenar las respuestas de exámenes cargadas por estudiantes y así verificarlas, esto conlleva que los profesores ahorren hasta un 70% del tiempo dedicado a la calificación de exámenes [21].

### **2.3 Evaluación del aprendizaje**

La evaluación es un proceso fundamental que implica la recopilación, análisis e interpretación de información para respaldar la toma de decisiones en diversos ámbitos escolares. La cual proporciona información a los agentes educativos para identificar y evaluar los resultados alcanzados frente a los objetivos iniciales.

La evaluación se manifiesta a través de la calificación numérica (evaluación sumativa) y el seguimiento del proceso (evaluación formativa), permitiendo una comprensión más clara de las áreas que necesitan mejoras. El desarrollo del currículo oficial se evalúa mediante tres dimensiones: evaluación diagnóstica para analizar conocimientos previos, evaluación formativa para retroalimentar durante el proceso y evaluación sumativa al final del proceso.

Las pruebas son instrumentos esenciales para evaluar habilidades, competencias y aprendizajes. Generalmente, proporcionan evidencia sobre cómo los alumnos comprenden, procesan y aplican conocimientos. Se clasifican en pruebas no estandarizadas, diseñadas para un solo grupo de alumnos en un salón de clases, y pruebas estandarizadas, destinadas a comparaciones objetivas entre alumnos de diferentes escuelas y estados.

Una prueba objetiva, ya sea estandarizada o no, debe cumplir con tres características: validez, confiabilidad y facilidad en la aplicación y calificación. La validez asegura que la prueba mida lo que pretende medir, mientras que la confiabilidad garantiza que los resultados reflejen realmente el conocimiento del alumno. Además, se destacan características como la univocidad, la inequívoca interpretación, la adaptación al currículo y la economía en tiempo de aplicación [22].



### **2.3.1 Tipos de reactivos**

Existen dos tipos principales de reactivos utilizados en evaluaciones: la opción múltiple o simple y la respuesta construida. Los reactivos de opción múltiple consisten en enunciados interrogativos con varias opciones de respuesta, donde solo una es completamente correcta. Normas específicas para su diseño incluyen la claridad en la base del reactivo, evitar información innecesaria, redacción sencilla, y la inclusión de una respuesta correcta junto con distractores que no confundan al estudiante.

Las ventajas de los reactivos de opción múltiple incluyen su aplicación eficiente en poco tiempo y la evaluación de objetivos de bajo nivel, mientras que las desventajas se centran en el tiempo necesario para su redacción y su limitada utilidad para demostrar el trabajo realizado [23].

En contraste, los reactivos de respuesta construida solicitan al estudiante producir respuestas libres, siendo utilizados para evaluar habilidades como la escritura de ensayos o la resolución de problemas complejos. Estos reactivos pueden ser de respuesta breve o larga, cada uno con sus ventajas y desventajas. Mientras permiten evaluar aprendizajes complejos y representan la vía ideal para evaluar la expresión escrita, presentan desafíos en términos de tiempo de respuesta y confiabilidad en la calificación [24]. Howitt, L.C. (1968). Prueba práctica en el aula. México: UTEHA.

### **2.3.2 Reactivos de opción múltiple**

Un reactivo, o pregunta de evaluación, puede adoptar diversas formas, siendo el de opción múltiple uno de ellos. Este tipo de reactivo, ampliamente utilizado en pruebas escritas estructuradas, presenta un enunciado seguido de varias alternativas de respuesta, de las cuales solo una es correcta. Algunas ventajas de este formato incluyen la facilidad para obtener respuestas únicas y una calificación rápida y objetiva.

La estructura general de un reactivo de opción múltiple consta de dos partes principales: la base y las opciones de respuesta. La base puede ser una proposición interrogativa, afirmativa o imperativa, o bien, un planteamiento de situación o problema. Se destaca la importancia de utilizar un lenguaje apropiado, claro y preciso en la redacción, evitando errores gramaticales y manteniendo independencia entre reactivos. Se pueden incluir elementos visuales como gráficos o imágenes si son esenciales para la resolución.

La base se complementa con las opciones de respuesta, donde se destaca la importancia de utilizar cuatro opciones (A, B, C, D), verificar la correspondencia gramatical con la

base y mantener una extensión y nivel de generalidad similar. La respuesta correcta debe ser única y resolver el reactivo de manera incuestionable.

Los distractores, opciones incorrectas, deben ser plausibles y reflejar errores comunes, evitando confundir al estudiante. Además, se proporcionan pautas para el uso de imágenes, incluyendo claridad y referencia si no son de creación propia. La síntesis enfatiza la necesidad de justificar cada opción de respuesta de forma clara e independiente, con argumentos válidos para respaldar la corrección o incorrección de cada opción [25].

## **2.4 Ingeniería de software**

La Ingeniería de Software se le considera como el proceso de construcción del software por las siguientes etapas: la obtención de los requisitos, el diseño del sistema, la codificación y las pruebas del sistema; desde la perspectiva del producto se parte de una necesidad, se especifican los requisitos, se obtiene el diseño del mismo, el código respectivo y por último el sistema de software, por lo tanto, se puede decir que se denomina ciclo de vida a toda las etapas o procesos, iniciando con su concepción y finalizando en el momento de la desinstalación del mismo. El ciclo de desarrollo se entiende como al subconjunto de las etapas que empieza en el análisis y finaliza en la entrega del producto [26]. Los avances de la Ingeniería de Software se derivan de una serie de métodos de análisis y diseño, orientada a objetos siendo una herramienta el Lenguaje Unificado de Modelado (Unified Modeling Language UML), se originó a fines de los ochenta y principios de los noventa, y no fue concebido como un método en sí mismo, sino como la notación básicamente gráfica que cualquier metodología o proceso de software puede utilizar para expresar sus productos de análisis y diseño. Sin embargo, esta notación nació con el famoso proceso de software denominado proceso unificado, resultado del trabajo de Grady Booch, Jim Rumbaugh e Ivar Jacobson. Booch [27] describe cómo este proceso de software con la notación de UML permite el análisis y diseño orientado a objetos de cualquier software. UML como base de Ingeniería ya que es un estándar en la industria del Software y en específico a los diagramas de Casos de Usos, ya que los mismos nos ayudan a visualizar la interacción de los artefactos de software con los actores.

Los casos de uso son una técnica para la especificación de requisitos funcionales propuesta inicialmente por Ivar Jacobson [28] e incorporada a UML, para modelar la funcionalidad

del sistema tal como la perciben los agentes externos, denominados actores, que interactúan con el sistema desde un punto de vista particular

Sus componentes principales son:

- Sujeto: sistema que se modela
- Casos de Uso: unidades funcionales completas
- Actores: entidades externas que interactúan con el sistema

El sujeto se muestra como una caja negra que proporciona los casos de uso. El modelo de casos de uso se representa mediante los diagramas de casos de uso

## **2.5 Casos de Uso**

Un Caso de Uso se define como un conjunto de acciones realizadas por el sistema que dan lugar a un resultado observable. El caso de uso especifica un comportamiento que el sujeto puede realizar en colaboración con uno o más actores, pero sin hacer referencia a su estructura interna. El Caso de Uso puede contener posibles variaciones de su comportamiento básico incluyendo manejo de errores y excepciones. Una instancia de un caso de uso es un escenario que representa un uso particular del sistema (un camino).

Características de los casos de uso:

- Un Caso de Uso se inicia por un actor
- Los Casos de Uso proporcionan valores a los actores
- La funcionalidad de un Caso de Uso debe ser completa

El comportamiento de un Caso de Uso se puede describir mediante interacciones, actividades y máquinas de estado.

### **2.5.1 Relaciones de los casos de uso**

Los procesos de desarrollo de software UML son iterativos e incrementales, repitiendo una serie de iteraciones sobre el ciclo de vida de un sistema. Cada iteración consiste de un paso a través de las etapas de requerimientos, análisis, diseño, implementación y prueba. El resultado de cada iteración representa un incremento sobre cada uno de los modelos construidos en las etapas anteriores. Los distintos Casos de Uso que se definen a lo largo del proceso de desarrollo no son independientes sino que es posible establecer relaciones de dependencia entre ellos. Las principales relaciones consideradas por UML son [29]:

- Generalización (generalization): es una relación que amplía la funcionalidad de un Caso de Uso o refina su funcionalidad original mediante el agregado de nuevas operaciones y/o atributos y/o secuencias de acciones.
- Inclusión(include): es una relación mediante la cual se re-usa un Caso de Uso encapsulado en distintos contextos a través de su invocación desde otros Casos de Uso.
- Extensión(extend): es una relación que amplía la funcionalidad de un Caso de Uso mediante la extensión de sus secuencias de acciones.

### 2.5.2 Notación de Casos de Uso

El diagrama de Casos de Uso utiliza elementos estandarizados para elaborarlo. En primer lugar, hay tres elementos principales (Véase la **Figura 1**):

Actor: tanto si es una persona, como un sistema, se representa con el dibujo de una figura esquemática humana.

Sistema: el sistema al que se refiere el caso de uso tiene forma de rectángulo.

Caso de uso: se muestra como una elipse que suele incluir un texto describiendo brevemente el proceso.

La relación entre estos elementos se representa con unas líneas de conexión llamadas asociaciones. Una línea recta entre el actor y el caso de uso evidencia que el actor y el caso de uso descrito en la elipse están relacionados. Una línea discontinua establece una relación entre diferentes casos de uso. Como hay dos tipos diferentes de asociación entre casos de uso, a las líneas se les añade una palabra clave, denominada estereotipo en UML, que se pone entre dos pares de paréntesis angulares. La relación de dependencia entre los casos de uso se representa con la punta de una flecha. Se distingue entre estos dos estereotipos:

Asociación: el caso de uso en el cual comienza la línea discontinua se relaciona con un segundo caso de uso señalado por la punta de la flecha.

Asociación: el caso de uso en el cual comienza la línea discontinua puede extenderse al caso de uso señalado por la punta de la flecha bajo ciertas condiciones, que no han de cumplirse necesariamente en todos los casos.

# Elementos UML

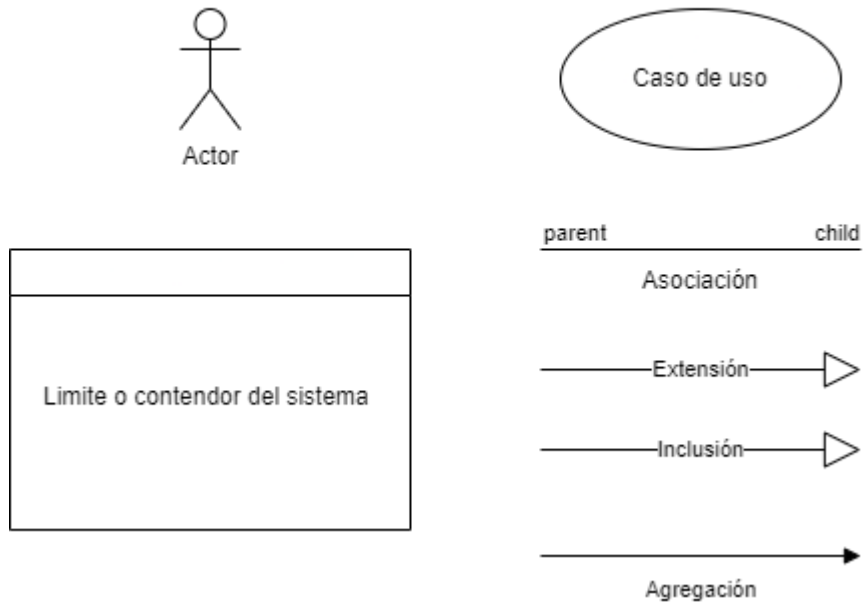


Figura 1. Elementos que integran un Caso de Uso.

## Capítulo 3. Análisis y diseño

### 3.1. Requerimientos casos de uso

Los requerimientos para la generación, colaboración, validación y almacenamiento de la base de datos se describen a continuación:

1. Generación de reactivos: se pretende crear preguntas de evaluación de distintas áreas de conocimiento y dificultad utilizando modelos de IA como herramientas para agilizar el proceso de creación de las preguntas a la vez que estos permiten la adaptabilidad en dificultad enfoque y estilo de estas.
2. Colaboración multidisciplinaria: se busca la colaboración entre académicos y expertos de distintas áreas para garantizar la variedad y calidad de los reactivos para enriquecer la base de datos.
3. Validación y retroalimentación humana: a pesar de la utilizar modelos avanzados de IA es importante la validación y retroalimentación de expertos y académicos para garantizar la calidad relevancia y precisión de las preguntas la retroalimentación permitirá mejoras continuas en el proceso
4. Base de datos: los reactivos aprobados serán almacenados en una base de datos. Esta base de datos se convertirá en una fuente valiosa para las evaluaciones y exámenes

### 3.2. Análisis

A continuación, se muestran 3 casos de usos y su diagrama

CU-001		Generar reactivos	
<b>Versión.</b>	1 (20/08/2023)		
<b>Dependencias.</b>			
<b>Precondición.</b>	El docente o experto previamente debe haber iniciado sesión en la plataforma.		
<b>Descripción.</b>	Generación de preguntas de evaluación mediante modelos de IA y validación de expertos.		
<b>Secuencia</b>	Paso	acción	
<b>Normal</b>	1	El docente o experto solicita al sistema generar preguntas	
	2	El docente o experto proporciona la información necesaria para generar las preguntas	

	3	El sistema prepara la información para consumir el api del modelo GPT
	4	El sistema consume el api del modelo GPT
	5	El sistema captura las preguntas a partir de la respuesta dada por el modelo GPT
	6	El sistema muestra las preguntas al docente o experto y se le da a elegir si guardarlas para revisión o descartarlas
	6.1	Se descartan y se envía al paso 1
	6.2	Se guardan las preguntas y respuestas.
<b>Postcondición.</b>	Preguntas y respuestas generadas se almacenan en lista para revisión.	
<b>Excepción.</b>	Paso	acción
	5	En caso de algún error al esperar la respuesta del modelo GPT
	E.1	El sistema Informa de que a ocurrido un error
	E.2	Se cancela el proceso
<b>Comentarios</b>		

<b>CU-002 Revisión y aprobación</b>		
<b>Versión.</b>	1 (20/08/2023)	
<b>Dependencias.</b>	CU-001	
<b>Precondición.</b>	Preguntas generadas almacenadas para revisión	
<b>Descripción.</b>	Revisión y aprobación por expertos y académicos de las preguntas almacenadas	
<b>Secuencia Normal</b>	Paso	acción
	1	Experto accede a lista de preguntas y respuestas generadas.
	2	Experto revisa y aprueba o descarta cada pregunta.
	3	Preguntas y respuestas aprobadas se almacenan en la base de datos.
<b>Postcondición.</b>	Preguntas y respuestas generadas se almacenan en lista para revisión.	
<b>Excepción.</b>	Paso	Acción
<b>Comentarios</b>		

<b>CU-003 Consulta de Preguntas Aprobadas</b>		
---	--	--



<b>Versión.</b>	1 (20/08/2023)	
<b>Dependencias.</b>	CU-001 CU-002	
<b>Precondición.</b>	Preguntas aprobadas almacenadas en la base de datos	
<b>Descripción.</b>	Búsqueda y selección de preguntas aprobadas en la base de datos para su utilización	
<b>Secuencia Normal</b>	Paso	Acción
	1	Docente explora la base de datos de preguntas aprobadas por área temática
	2	Docente selecciona preguntas específicas para utilizar en la creación de exámenes.
<b>Postcondición.</b>	Preguntas seleccionadas quedan en lista para su uso en la creación de exámenes.	
<b>Excepción.</b>	Paso	Acción
<b>Comentarios</b>		

La **Figura 2** muestra un diagrama de casos de uso relacionado con la Generación de Reactivos con la Ayuda de GPT. En este diagrama, se identifica al actor principal como un Docente o Experto, quien interactúa con el sistema a través de varias funcionalidades clave. Las principales actividades que el docente o experto puede realizar incluyen la Generación Inicial de Reactivos, Revisión y Aprobación, Consulta de Preguntas Aprobadas, Generación de Exámenes Personalizados, Registrarse y Cerrar Sesión.

El caso de uso de Iniciar Sesión es un componente central que se incluye en las demás funcionalidades, asegurando que el usuario esté autenticado antes de realizar cualquier operación. Este flujo de trabajo asegura que solo los usuarios registrados y autenticados puedan interactuar con las funciones críticas del sistema.

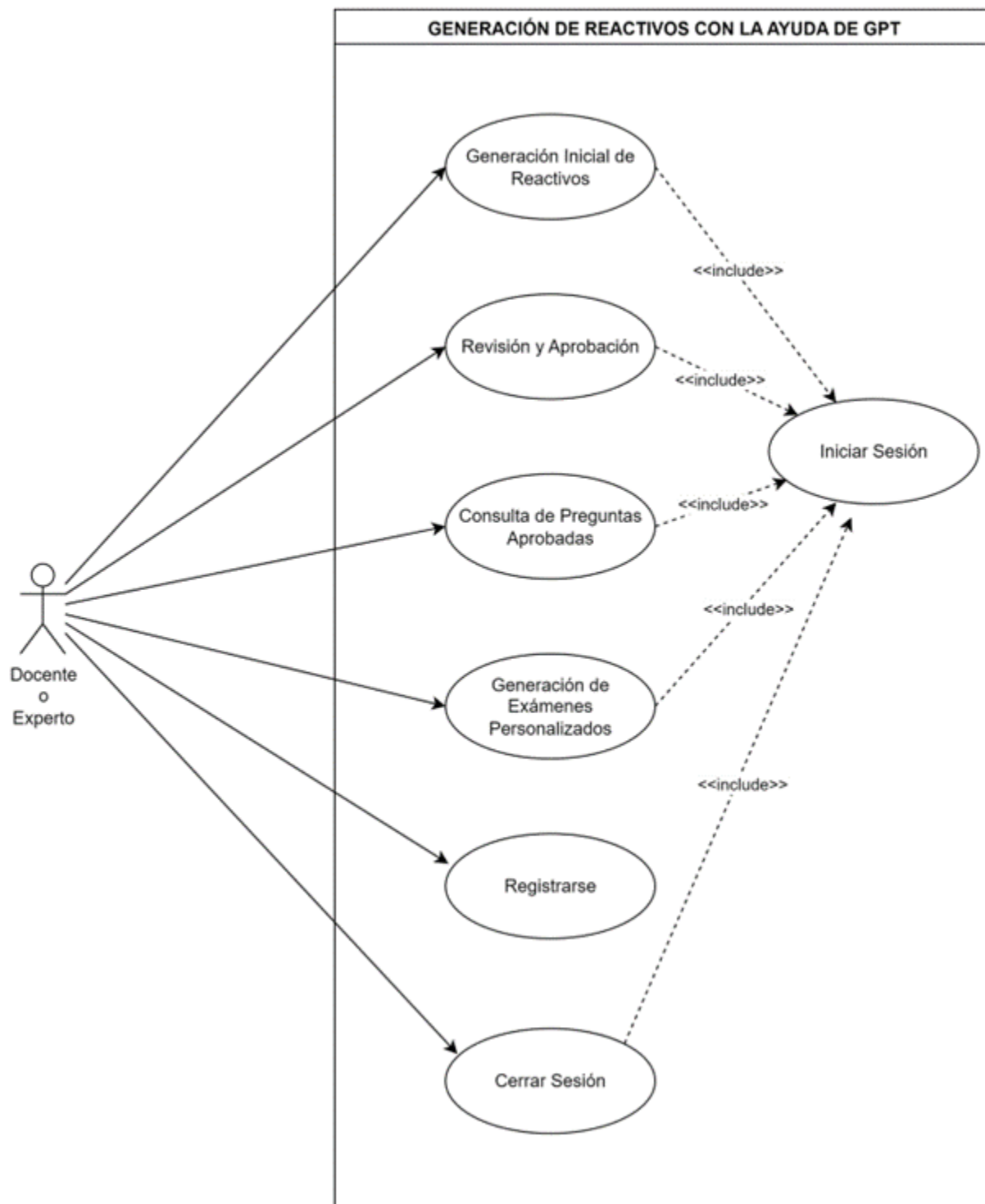


Figura 2. Casos de uso de generación de reactivos.

Modelo de interacción de servicios. véase el diagrama de la **Figura 3**.

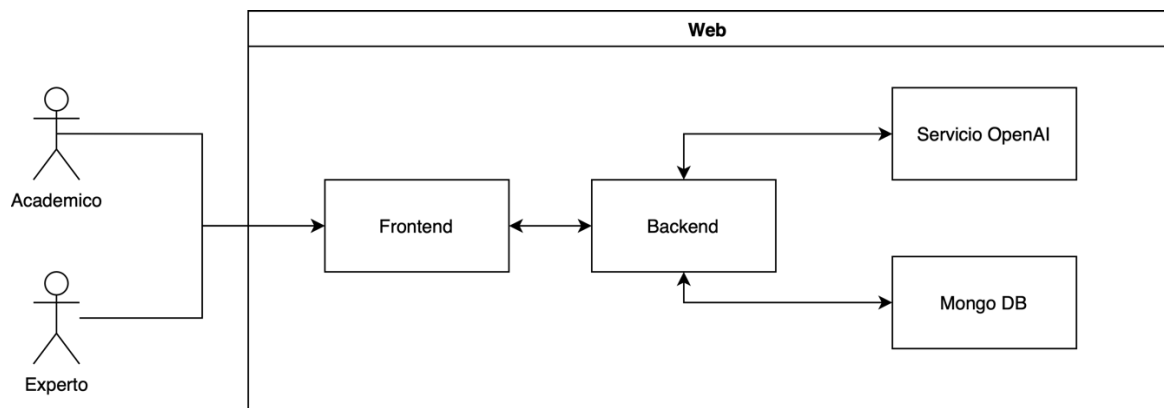


Figura 3. Interacción de frontend con servicio backend y base de datos.

En el Diagrama muestra la interacción de los usuarios con el componente y la comunicación entre componentes.

- **Generar Nuevo reactivo:** Para la generación de un nuevo reactivo el usuario (Académico, experto) interactúa con el frontend insertando el tema sobre el que decía el nuevo reactivo y da clic en un botón el cual ejecuta una petición que accede al backend el cual interpreta y manda a traer el servicio de OpenAI para generar dicho reactivo una vez que el servicio concluye el backend guarda el reactivo en base de datos con un estatus de pendiente y lo regresa al front para que se le muestre al usuario en una ventana emergente la cual tiene un botón para cerrarse al cerrar dicha ventana se actualiza la lista de reactivos que están almacenados en base de datos.
- **Lista reactivo de bases de datos:** esta petición se ejecuta cuando al usuario se le muestra la interfaz principal del sistema (Frontend) el cual contiene una tarjeta para generar un nuevo reactivo y una lista de tarjetas que contienen la información de los reactivos que están almacenados en base de datos, los reactivos que están con estatus de pendiente muestran dos botones con la opción de aceptar y rechazar, los reactivos que están con estatus rechazado o aceptado no muestran estas opciones al igual que con generar reactivo el flujo de interacciones es a través del frontend hacia el backend y posteriormente se accesa a la base de datos para extraer la lista de reactivos almacenados, esta es retornada al frontend y mostrada al usuario.
- **Aceptar o rechazar un reactivo:** Los reactivos con estatus de pendiente como se menciona contienen dos botones uno para aceptar el reactivo y otro para rechazarlo estos a su vez ejecutan una función que es actualizar en base de datos su estatus del reactivo el flujo de interacción es a través del frontend ejecutando una petición hacia

el backend para actualizar esta información en base de datos con la información requerida en este caso el estatus de un reactivo en específico, una vez terminada la actualización de los datos se actualiza en el listado general de los reactivos que se están mostrando al usuario.

### 3.3. Diseño

Para la parte de implementación se penso en un sistema de base de datos no SQL específicamente en MongoDB a continuación se muestra el diseño de las estructuras en TypeScript (**interface / interfaz**).

En la interfaz **OpcionMultipleReactivo**, se definen los atributos esenciales para manejar las opciones en un reactivo de opción múltiple. Estos atributos permiten describir tanto el texto de la opción como si esta es la respuesta correcta. La **Tabla 2** detalla los atributos de esta interfaz.

OpcionMultipleReactivo	
textoOpcion	Propiedad de la estructura de tipo cadena de texto, se utiliza para mostrar la descripción de una posible respuesta en reactivos de opción multiple.
esCorrecta	Propiedad de la estructura de tipo boolean, se utiliza para contener si la respuesta es correcta o no.

```
1 export interface OpcionMultipleReactivo {
2     textoOpcion: String,
3     esCorrecta: Boolean
4 }
```

Tabla 2. Estructura de la opción múltiple.

En la interfaz **Reactivo**, se definen los atributos necesarios para representar una pregunta en el sistema de reactivos. Esta interfaz incluye atributos que permiten describir la pregunta, sus posibles respuestas, el estado de aprobación, entre otros detalles esenciales. La **Tabla 3** detalla los atributos de esta interfaz.

Reactivo	
textoPregunta	Propiedad de la estructura de tipo cadena de texto, se utiliza para mostrar la descripción o enunciado de la pregunta.
textoRespuesta	Propiedad de la estructura de tipo cadena de texto, se utiliza para mostrar la idea principal que se espera en la respuesta si esta no es de opción multiple.
opciones	Propiedad de la estructura que contiene múltiples instancias o referencias a la estructura OpcionMultipleReactivo.
estadoAprobacion	Propiedad de la estructura de tipo cadena de texto, se utiliza para mostrar el estado de revisión de una pregunta, se cuenta con 3 posibles estados Pendiente indica que aún no se ha aprobado, Aprobado indica que este reactivo fue aprobado y se puede usar para exámenes o evaluaciones y Rechazado indica que no puede ser utilizado de ninguna forma.
comentarios	Propiedad de la estructura de tipo cadena de texto, se utiliza para almacenar comentarios u observaciones para el reactivo.
fechaRevision	Propiedad de la estructura de tipo cadena de texto, se utiliza para mostrar la fecha en la que se aprobó o rechazo el reactivo.
uidUser	Propiedad de la estructura de tipo cadena de texto, se utiliza para referenciar al usuario que aprobó o rechazo el reactivo.

```

1  export interface Reactivo {
2      textoPregunta: String,
3      textoRespuesta: String,
4      opciones: [OpcionMultipleReactivo]
5      estadoAprobacion: 'Pendiente' | 'Aprobado' | 'Rechazado',
6      comentarios: String,
7      fechaRevision: String,
8      uidUser: String
9  }

```

Tabla 3. Estructura del Reactivo.

En la interfaz **Usuario**, se definen los atributos necesarios para gestionar la información de los usuarios dentro del sistema. Esta interfaz incluye atributos para el nombre del usuario, su correo electrónico, contraseña, y el rol asignado, lo que permite controlar el acceso y las funciones disponibles para cada tipo de usuario. La **Tabla 4** detalla los atributos de esta interfaz.

Usuario	
Nombre	Propiedad de la estructura de tipo cadena de texto, se utiliza mostrar el nombre del usuario.
correoElectronico	Propiedad de la estructura de tipo cadena de texto, se utiliza para iniciar sesión, se aplicará filtro para correos autorizados.
contrasena	Propiedad de la estructura de tipo cadena de texto, se utiliza para almacenar la contraseña debe estar encriptada.
rol	Propiedad de la estructura de tipo cadena de texto, se utiliza para almacenar el rol del usuario de los disponibles esto para tener control del acceso de los usuarios.

```

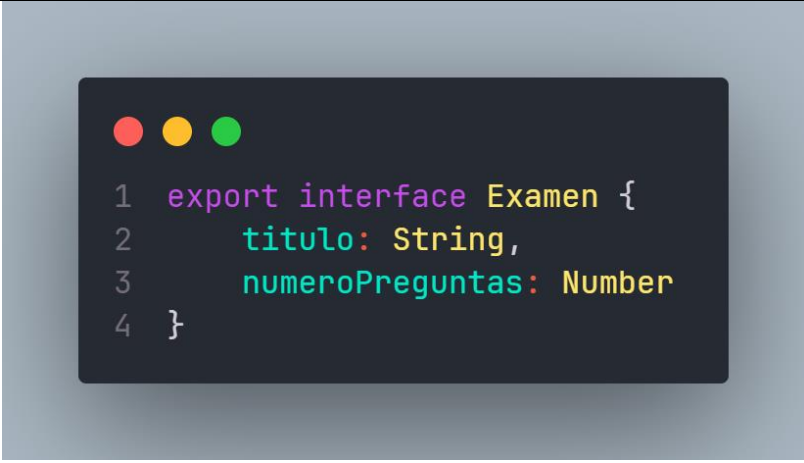
1 export interface Usuario {
2     nombre: String,
3     correoElectronico: String,
4     contraseña: String,
5     rol: 'Academico' | 'Colaborador Experto' | 'Estudiante' | 'Revisor'
6 }

```

Tabla 4. Estructura del Usuario.

En la interfaz **Examen**, se definen los atributos esenciales para representar un examen dentro del sistema. Esta interfaz incluye atributos que permiten identificar el título del examen y la cantidad de preguntas que lo componen. La **Tabla 5** detalla los atributos de esta interfaz.

Examen	
titulo	Propiedad de la estructura de tipo cadena de texto, se utiliza para identificar el examen.
numeroPreguntas	Propiedad de la estructura de tipo numérico que indica la cantidad de preguntas que tiene ese examen



```

1 export interface Examen {
2     titulo: String,
3     numeroPreguntas: Number
4 }

```

Tabla 5 Estructura del examen.

En la interfaz **PreguntaExamen**, se definen los atributos necesarios para gestionar las preguntas que forman parte de un examen. Esta interfaz incluye atributos que permiten

referenciar el reactivo asociado, el puntaje asignado a la pregunta, su posición dentro del examen, y el examen al que pertenece. La **Tabla 6** detalla los atributos de esta interfaz.

PreguntaExamen	
ReactivoID	Propiedad de la estructura de tipo cadena de texto, se utiliza para referenciar al reactivo que se está agregando al examen.
puntaje	Propiedad de la estructura numérico, se utiliza para referenciar el valor de la pregunta sobre la calificación del examen.
posicion	Propiedad de la estructura de tipo numérico se utiliza para saber la posición donde se encuentra en el examen.
examenID	Propiedad de la estructura de tipo cadena de texto, se utiliza para referenciar del examen al cual pertenece.

```
1 export interface PreguntaExamen {
2   reactivoID: String,
3   puntaje: Number,
4   posicion: Number,
5   examenID: String
6 }
```

Tabla 6. Estructura de una pregunta de examen.

### Diseño del Mensaje para generar reactivos con OpenAI

Para obtener una respuesta concisa del servicio de OpenAI y poder recuperar la información de manera eficiente, se hicieron pruebas con el servicio de chat gpt de OpenAI, el sistema se basa en la utilización de la biblioteca de OpenAI para interactuar con el modelo de lenguaje GPT-3.5. Este modelo es responsable de generar los reactivos que utilizamos en nuestro servicio. La interacción con GPT-3.5 se lleva a cabo mediante mensajes que enviamos al modelo. Para que esta interacción sea efectiva, es crucial enviar cierta información de una



manera específica. Esta información permite que el servicio de OpenAI nos responda con un objeto de tipo JSON, o que la respuesta del servicio contenga un objeto JSON que podemos extraer para obtener la información que necesitamos.

Este proceso estableció a través de un método de prueba y error, trabajando con el servicio público de OpenAI conocido como ChatGPT. Experimentando con diferentes formas de enviar mensajes y estructurar la información para obtener respuestas coherentes y útiles del modelo. Este enfoque nos ha permitido perfeccionar la interacción con GPT-3.5 y aprovechar al máximo sus capacidades para generar los reactivos por medio de una respuesta que utiliza el sistema.

Se llegó a la conclusión de utilizar el siguiente mensaje para obtener de manera efectiva la información que necesitamos de GPT-3.5:

'Me puedes generar un reactivo de 3 opciones, \$reemplazar, quiero que me devuelvas un JSON con el reactivo (textoPregunta), que las opciones vengan en un arreglo de JSON (opciones) con texto de la respuesta (textoOpcion) y un booleano que indique si es la correcta o no (esCorrecta).'

La razón detrás de este mensaje específico radica en que identifique que este formato es el más adecuado para comunicar nuestras necesidades al modelo de manera clara y precisa. Al utilizar este mensaje, GPT-3.5 entiende que queremos que genere un reactivo con tres opciones, donde '\$reemplazar' es el marcador de posición que indicará el tema del cual deseamos los reactivos.

Además, especifique claramente que se desea recibir la respuesta en formato JSON, lo que facilita la extracción de información estructurada. En este JSON, se definieron tres elementos clave:

'textoPregunta': Aquí se espera que GPT-3.5 coloque el texto del reactivo que desea generar.

'opciones': Este será un arreglo de JSON que contendrá las opciones de respuesta. Cada elemento del arreglo tendrá dos campos: 'textoOpcion', que contendrá el texto de la respuesta generada, y 'esCorrecta', un valor booleano que nos indicará si la opción es la correcta o no.

Este mensaje bien estructurado permite obtener respuestas coherentes y en el formato deseado, lo que hace que el proceso de extracción de información sea más sencillo y efectivo.

A continuación, se muestran las evidencias del proceso para obtener dicho mensaje.

En la **Figura 4** se muestra el resultado del primer intento al solicitar a ChatGPT la generación de un reactivo de opción múltiple sobre el tema de las constantes en programación. El mensaje utilizado fue: Genera un reactivo de opción múltiple sobre las constantes en programación.

En la respuesta proporcionada por ChatGPT, se presenta una pregunta con cuatro opciones de respuesta:

- **A)** Puede cambiar su valor durante la ejecución del programa.
- **B)** Debe inicializarse sin un valor específico.
- **C)** Su valor no puede cambiar una vez asignado.
- **D)** Solo se puede usar en lenguajes de programación antiguos.

El sistema indica que la respuesta correcta es la opción **C)**: "Su valor no puede cambiar una vez asignado".

Este primer intento genera un reactivo que cumple con la estructura básica solicitada, pero aún no está en el formato JSON deseado para el proyecto.

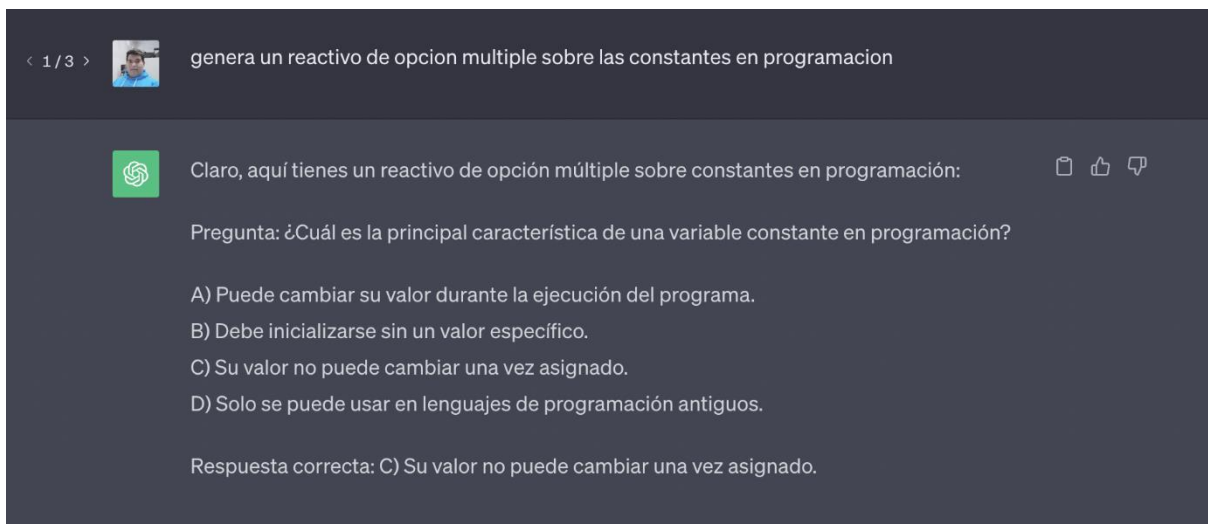


Figura 4. Resultado del primer intento.

En la **Figura 5** se muestra el resultado del segundo intento al solicitar a ChatGPT la generación de un reactivo de opción múltiple sobre las constantes en programación, especificando que incluya tres opciones de respuesta. El mensaje utilizado fue: "Genera un reactivo de opción múltiple sobre las constantes en programación de 3 respuestas".

En la respuesta proporcionada, se presentan cuatro opciones de respuesta, aunque se había solicitado solo tres:

- **A)** Una variable que puede cambiar su valor durante la ejecución del programa.

- **B)** Un valor que se utiliza solo una vez en un programa.
- **C)** Una variable cuyo valor no puede cambiar una vez que se le asigna un valor inicial.
- **D)** Un tipo de dato utilizado para almacenar números enteros.

El sistema indica que la respuesta correcta es la opción **C)**: "Una variable cuyo valor no puede cambiar una vez que se le asigna un valor inicial." Aunque se generó un reactivo correcto, se observa que no se cumplió con la restricción de tres opciones.

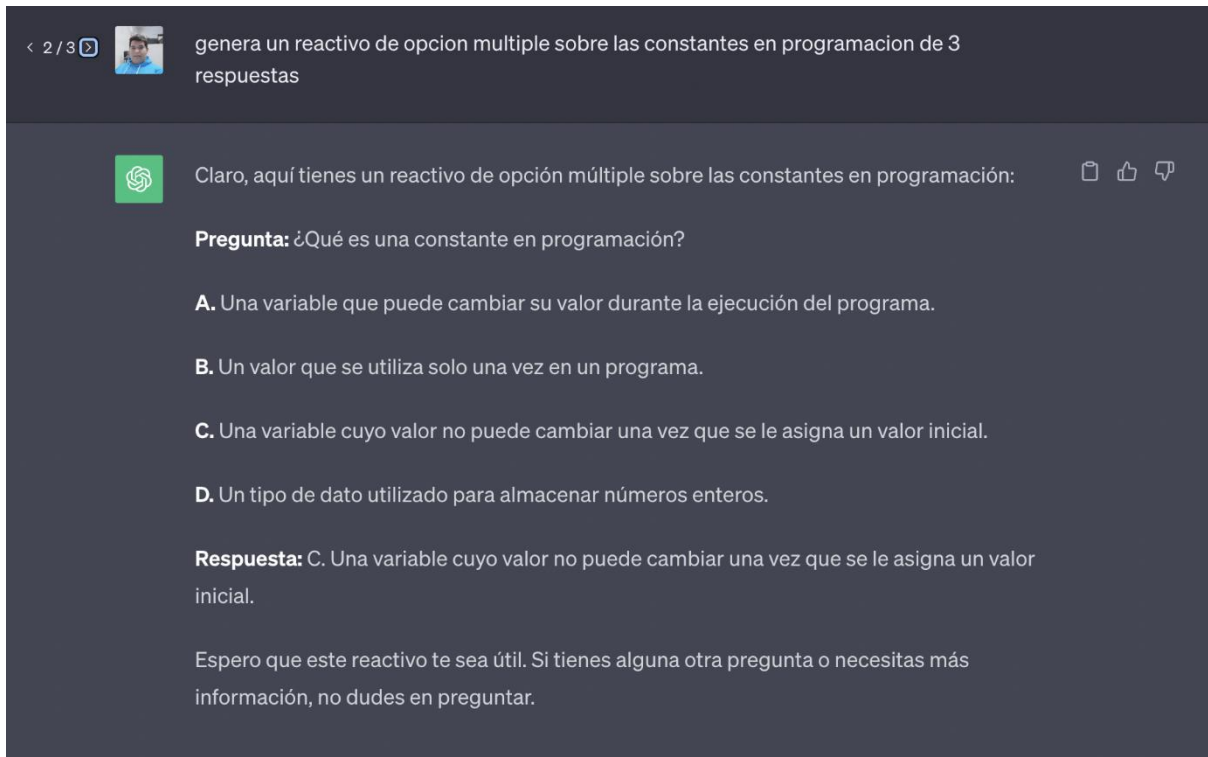


Figura 5. Resultado del segundo intento.

En la **Figura 6** se muestra el resultado del tercer intento, donde se especificó que el reactivo debía tener tres opciones de respuesta. El mensaje utilizado fue: "Genera un reactivo de opción múltiple sobre las constantes en programación de 3 opciones".

En la respuesta, se presentan tres opciones de respuesta:

- **A)** Una variable que puede cambiar su valor durante la ejecución del programa.
- **B)** Una variable que almacena valores que son conocidos y fijos, y no puede cambiar su valor una vez que se le asigna.
- **C)** Una variable que solo puede utilizarse dentro de una función específica.

El sistema indica que la respuesta correcta es la opción **B)**: "Una variable que almacena valores que son conocidos y fijos, y no puede cambiar su valor una vez que se le asigna." Además, se proporciona una breve explicación sobre las constantes en programación.

Este tercer intento se acerca más a lo solicitado, cumpliendo con el requisito de tres opciones, pero aún falta la estructura en formato JSON.

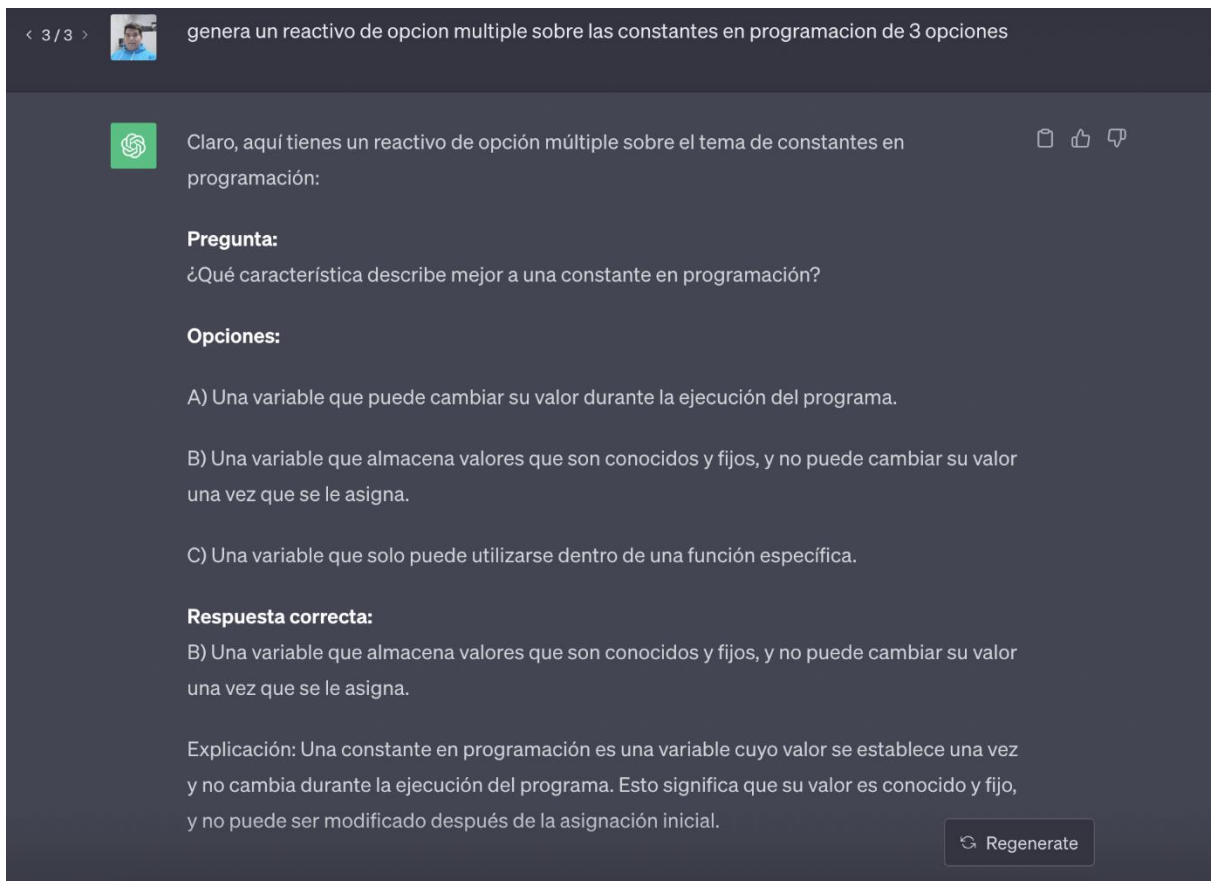


Figura 6. Resultado del tercer intento.

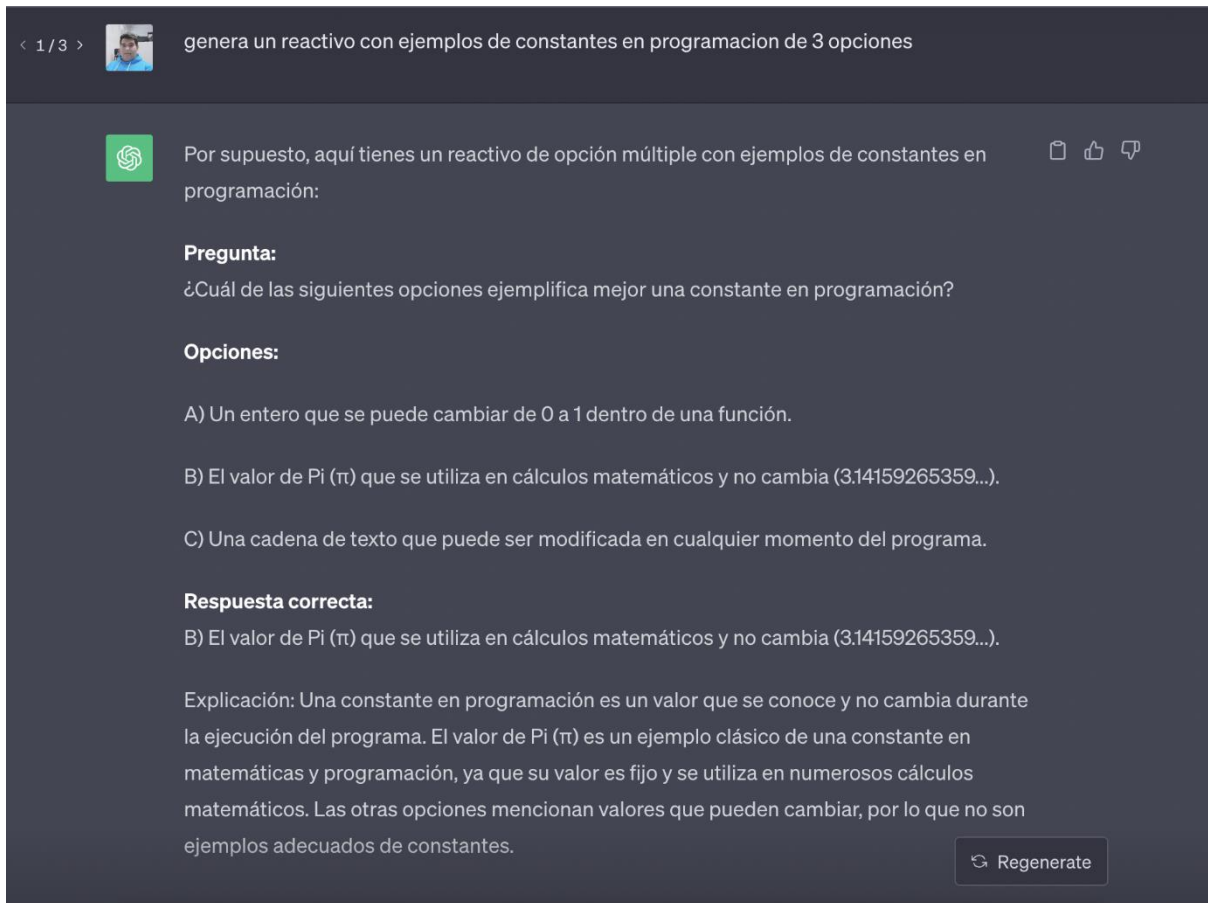
En la **Figura 7** se muestra el resultado del cuarto intento al solicitar a ChatGPT la generación de un reactivo de opción múltiple con ejemplos de constantes en programación, especificando que incluya tres opciones de respuesta. El mensaje utilizado fue: "Genera un reactivo con ejemplos de constantes en programación de 3 opciones".

En la figura se observa que se presentan tres opciones de respuesta:

- **A)** Un entero que se puede cambiar de 0 a 1 dentro de una función.
- **B)** El valor de Pi ( $\pi$ ) que se utiliza en cálculos matemáticos y no cambia (3.14159265359...).
- **C)** Una cadena de texto que puede ser modificada en cualquier momento del programa.

El sistema indica que la respuesta correcta es la opción **B)**: "El valor de Pi ( $\pi$ ) que se utiliza en cálculos matemáticos y no cambia (3.14159265359...)." Además, se proporciona una

breve explicación sobre la naturaleza de las constantes en programación, con un enfoque en Pi como un ejemplo clásico de una constante.



The screenshot shows a chat window with a dark background. At the top, a user profile icon and the text "genera un reactivo con ejemplos de constantes en programacion de 3 opciones" are visible. Below this, a green circular icon with a white swirl is followed by the text "Por supuesto, aquí tienes un reactivo de opción múltiple con ejemplos de constantes en programación:". To the right of this text are icons for copy, like, and share. The question is titled "Pregunta:" and asks "¿Cuál de las siguientes opciones ejemplifica mejor una constante en programación?". The options are listed under "Opciones:" and are: A) Un entero que se puede cambiar de 0 a 1 dentro de una función. B) El valor de Pi ( $\pi$ ) que se utiliza en cálculos matemáticos y no cambia (3.14159265359...). C) Una cadena de texto que puede ser modificada en cualquier momento del programa. The correct answer is indicated as "Respuesta correcta:" and is B) El valor de Pi ( $\pi$ ) que se utiliza en cálculos matemáticos y no cambia (3.14159265359...). Below the answer is an "Explicación:" which states that a constant in programming is a value that is known and does not change during execution, and that Pi is a classic example of a constant in mathematics and programming because its value is fixed and used in many mathematical calculations. The other options mention values that can change, so they are not appropriate examples of constants. At the bottom right of the chat window is a "Regenerate" button.

Figura 7. Resultado del cuarto intento.

En la **Figura 8** se muestra el resultado del quinto intento, donde se especificó que el reactivo debía contener ejemplos de constantes en código de programación. El mensaje utilizado fue: "Genera un reactivo con ejemplos de constantes en código de programación de 3 opciones".

En la figura se observa que se presentan tres fragmentos de código como opciones de respuesta:

- **Opción A:** Un ejemplo donde  $PI = 3.14159265359$  se utiliza en un cálculo.
- **Opción B:** Un ejemplo donde se definen variables como nombre y edad.
- **Opción C:** Un ejemplo que incluye una lista de temperaturas y una constante `TEMPERATURA_MAXIMA`.

El sistema indica que la opción correcta es **A**), donde PI se utiliza como una constante en un cálculo matemático. Este intento utiliza código realista para ilustrar las opciones,

acercándose aún más a lo solicitado en términos de relevancia y contexto, aunque sigue faltando el formato JSON deseado.

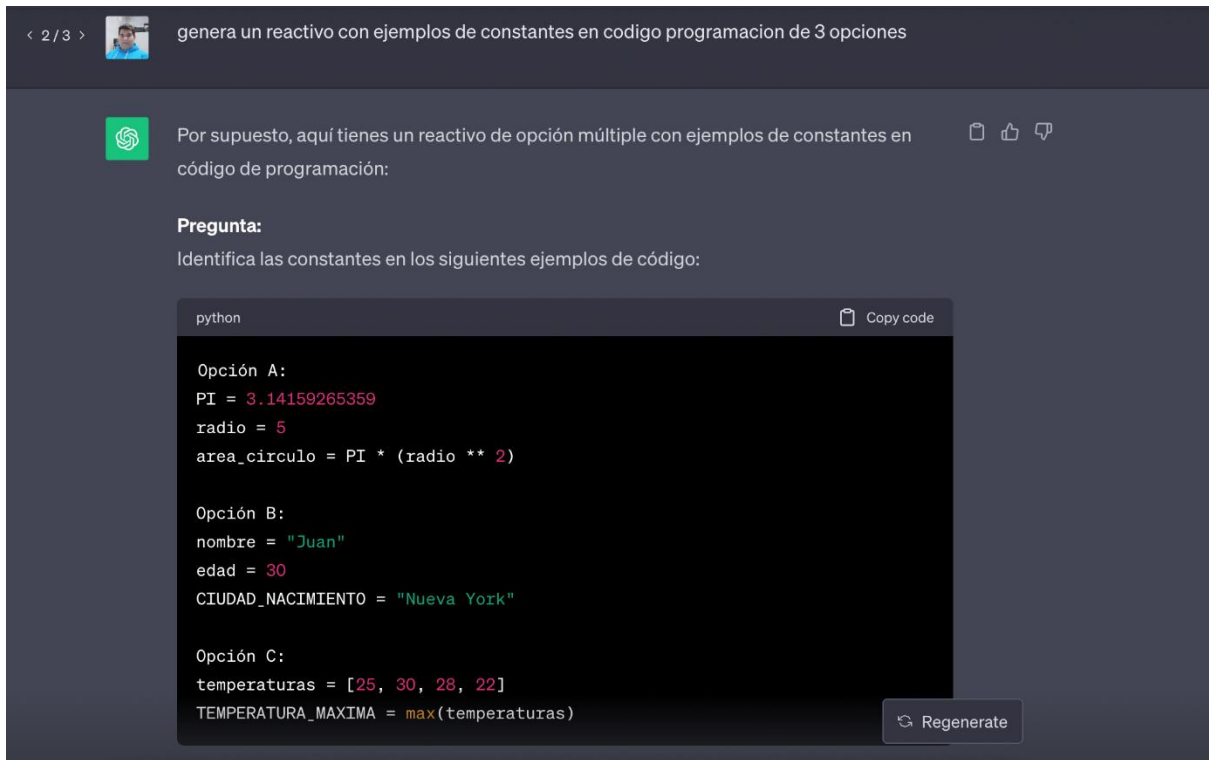


Figura 8. Resultado del Quinto intento.

En la **Figura 9** se muestra el resultado del sexto intento al solicitar a ChatGPT la generación de un reactivo de opción múltiple con ejemplos de constantes en código de programación TypeScript, especificando que incluya tres opciones de respuesta. El mensaje utilizado fue: "Genera un reactivo con ejemplos de constantes en código programación TypeScript de 3 opciones."

En la figura se observa que se presentan tres opciones de respuesta en forma de fragmentos de código:

**Opción A:** Un fragmento donde se define una constante `miVariable = 42;`, seguida de un intento de reasignación con `miVariable = 10;`.

**Opción B:** Un fragmento donde se define una variable `let nombre = "Juan";`.

**Opción C:** Otro fragmento de código, que no se muestra completamente en la imagen.

El sistema debería indicar cuál de estas opciones representa correctamente una constante en TypeScript, basándose en la inmutabilidad de las constantes definidas con `const`. Sin embargo, en esta figura se observa un intento de reasignar un valor a una constante, lo que no es permitido en TypeScript. Este sexto intento ilustra la necesidad de refinar aún más el mensaje para obtener un reactivo que refleje con precisión las características de las constantes en este lenguaje de programación.

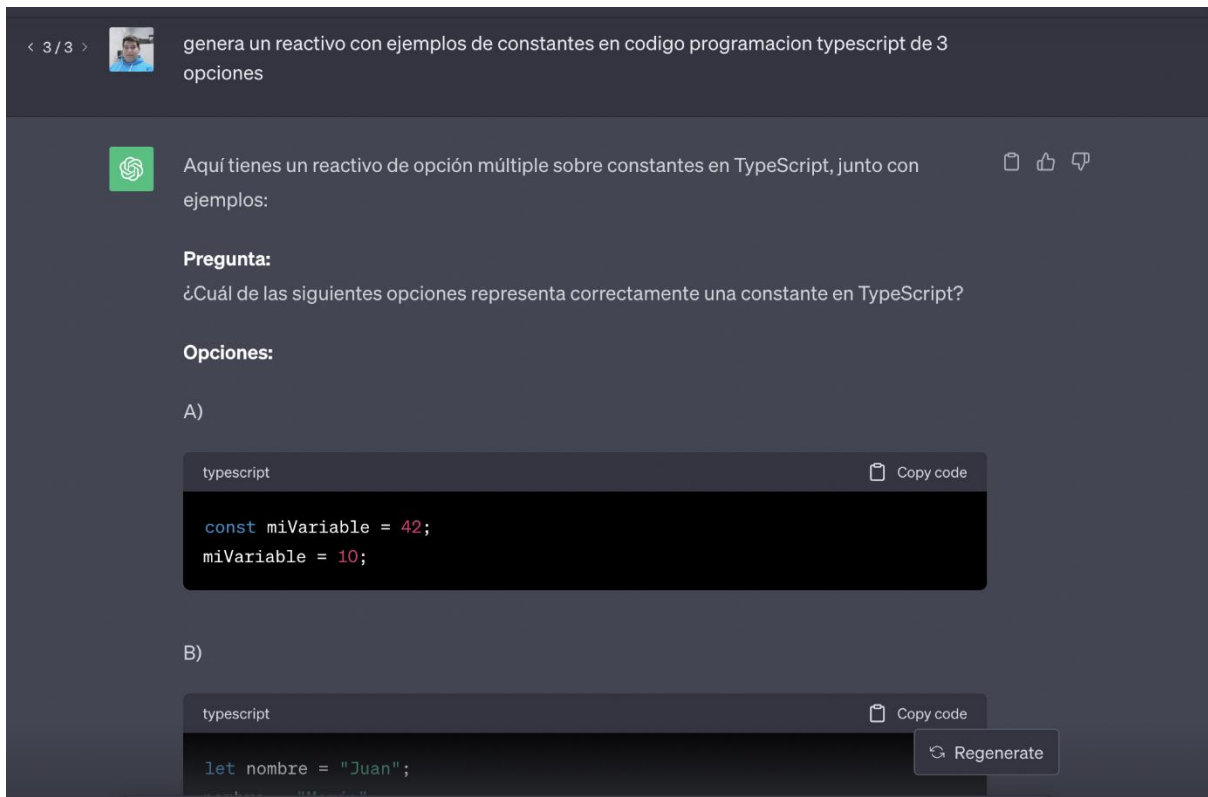
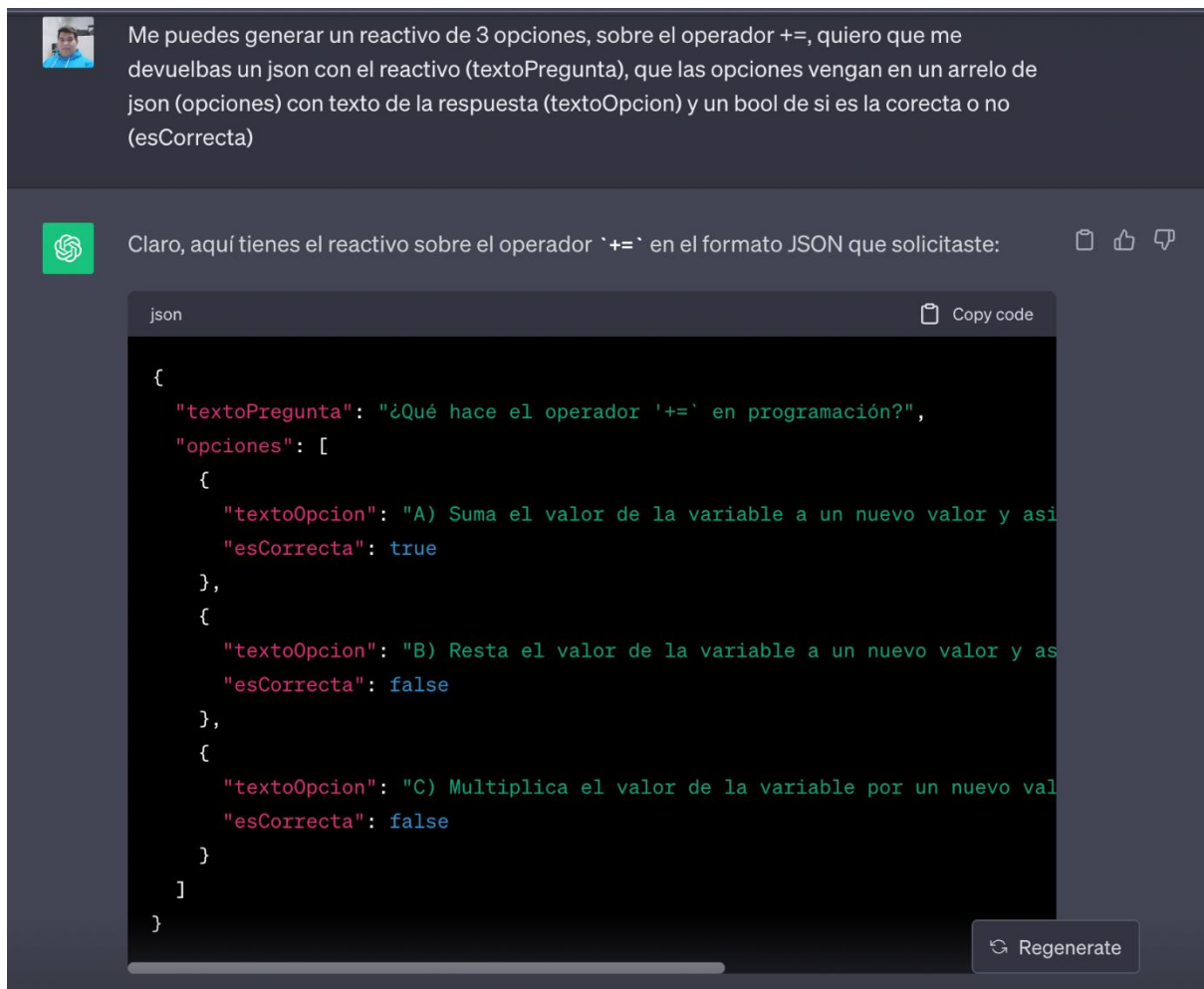


Figura 9. Resultado del sexto intento.

En la **Figura 10** se muestra el resultado del séptimo intento, en el cual se obtuvo finalmente el resultado esperado. En este intento, ChatGPT generó correctamente un reactivo de opción múltiple sobre el operador `+=` en programación, devolviendo el formato JSON solicitado, que incluye la pregunta, las opciones de respuesta, y un indicador booleano que señala la

opción correcta.



The screenshot shows a chat conversation. At the top, a user asks for a reactive with 3 options about the += operator, requesting a JSON response with fields for the question, options, and correctness. Below, the AI responds with the requested JSON. The JSON is displayed in a dark-themed code editor with a 'Copy code' button and a 'Regenerate' button at the bottom right.

```
json Copy code

{
  "textoPregunta": "¿Qué hace el operador `+=` en programación?",
  "opciones": [
    {
      "textoOpcion": "A) Suma el valor de la variable a un nuevo valor y asi",
      "esCorrecta": true
    },
    {
      "textoOpcion": "B) Resta el valor de la variable a un nuevo valor y as",
      "esCorrecta": false
    },
    {
      "textoOpcion": "C) Multiplica el valor de la variable por un nuevo val",
      "esCorrecta": false
    }
  ]
}
```

Regenerate

Figura 10. Resultado del séptimo intento el cual fue exitoso.

Este séptimo intento marca la conclusión exitosa del proceso, demostrando que, con un mensaje adecuadamente estructurado, es posible obtener la salida en el formato específico requerido. La serie de intentos documentados en las figuras anteriores refleja el proceso iterativo de refinamiento del mensaje hasta alcanzar el resultado deseado, subrayando la importancia de la precisión en la comunicación con herramientas de IA para obtener resultados específicos y útiles.



## Capítulo 4. Implementación y pruebas del sistema

## 4.1. Pruebas

En esta sección, se detallan las pruebas funcionales realizadas tanto en el backend como en el frontend del sistema. Estas pruebas se llevaron a cabo con el objetivo de verificar que cada parte del sistema cumple con los requisitos establecidos y funciona de manera coherente dentro del flujo general del sistema.

En esta sección, se detallan las pruebas funcionales realizadas tanto en el backend como en el frontend del sistema, con el fin de verificar que cada componente opere conforme a los requisitos especificados y funcione de manera coherente dentro del flujo general de la aplicación.

Para el backend, se centraron las pruebas en dos endpoints principales relacionados con la generación de reactivos. El primero es el endpoint encargado de **listar los reactivos**, donde se verifica que el sistema pueda recuperar y mostrar correctamente todos los reactivos almacenados, tanto los aprobados como los rechazados. El segundo endpoint probado es el de **generar reactivos**, el cual permite la creación de nuevos reactivos utilizando el modelo de lenguaje GPT-3.5-turbo. Estos endpoints fueron evaluados utilizando la extensión Thunder Client de Visual Studio Code, como se muestra en las **Figuras 11 y 12**.

Listar todas las preguntas almacenadas.

En la **Figura 11**, se muestra el listado de los reactivos que previamente fueron generados y almacenados mediante el sistema.

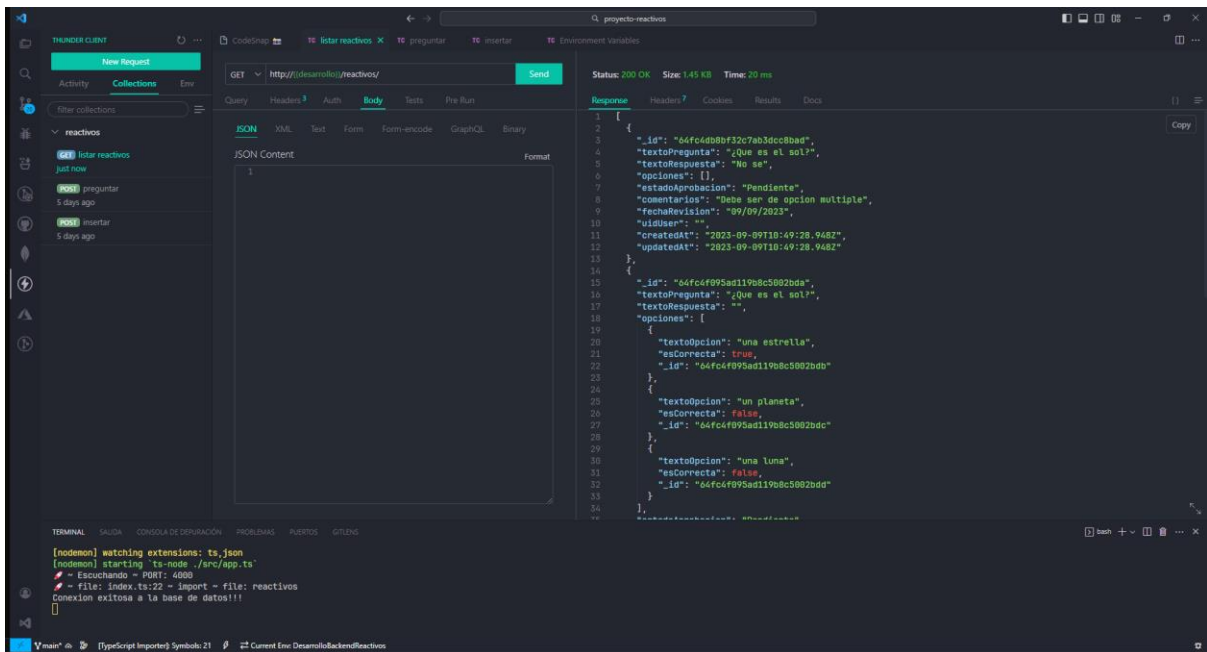


Figura 11. Se realiza una consulta http de los reactivos que se tienen almacenados.

La **Figura 12** ilustra cómo se utiliza una extensión de Visual Studio Code, Thunder Client, para acceder al servicio web del proyecto y probar el endpoint /reactivos/preguntar. En esta captura de pantalla, se realiza una solicitud HTTP tipo POST al servicio backend, con el propósito de generar un nuevo reactivo.

El cuerpo de la solicitud incluye detalles como el modelo de lenguaje utilizado, el mensaje que indica cómo se deben generar las opciones de respuesta, y parámetros adicionales como max\_tokens. La respuesta del backend se visualiza en el panel derecho, mostrando un JSON

que contiene la pregunta generada, las opciones de respuesta, la identificación de la opción correcta, y el estado actual de aprobación del reactivo.

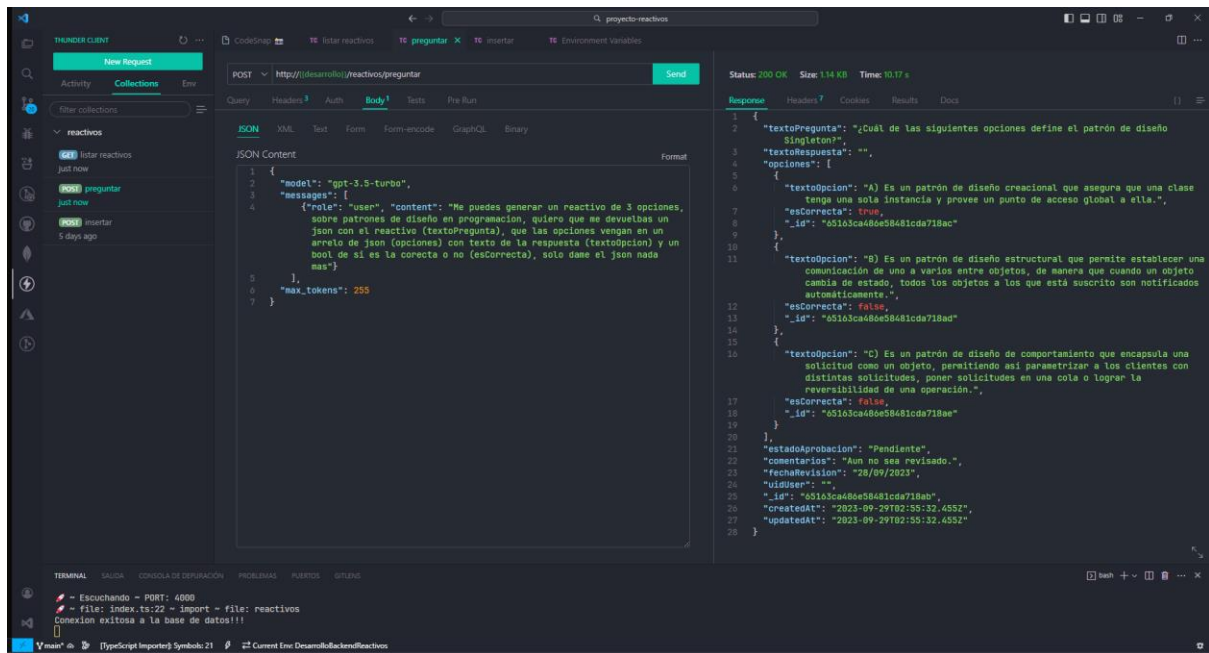


Figura 12. Se realiza una consulta http para generar un nuevo reactivo.

A continuación, se muestra la interfaz gráfica del proyecto (frontend), en la cual se va a mostrar el resultado de las distintas operaciones solicitadas al sistema.

Al acceder a la interfaz principal del sistema de reactivos, el usuario se encuentra con un diseño limpio y minimalista, especialmente cuando no hay reactivos creados previamente. En este estado inicial, la pantalla muestra un mensaje que invita al usuario a generar un nuevo reactivo. Para comenzar este proceso, el sistema ofrece un botón claramente etiquetado como "Nuevo", que es el punto de partida para la creación de reactivos. Esta disposición asegura que el usuario tenga una guía clara sobre cómo proceder, minimizando cualquier confusión o duda sobre los pasos a seguir. La **Figura 13** muestra cómo se presenta esta interfaz en ausencia de reactivos.

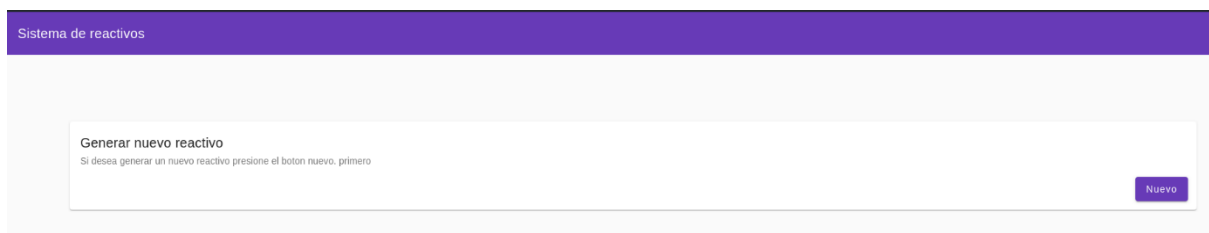
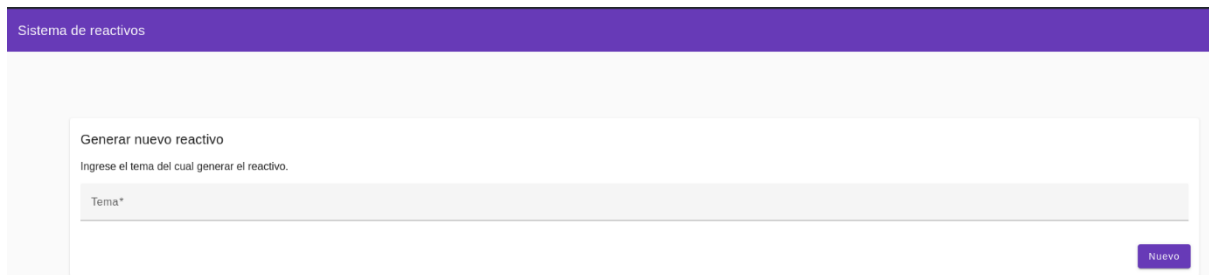


Figura 13. Interfaz principal sin reactivos previos.

Una vez que el usuario ha decidido generar un nuevo reactivo, el sistema presenta un formulario sencillo pero efectivo para ingresar la información necesaria. En esta etapa, la interfaz solicita

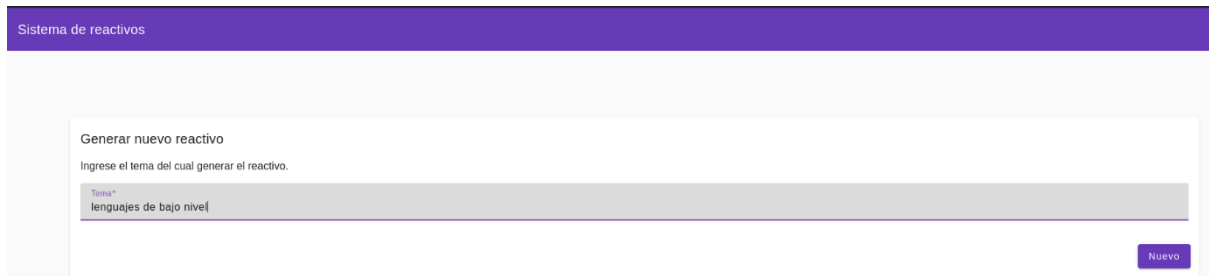
que el usuario proporcione el tema del reactivo, lo cual es un paso crucial para garantizar que el contenido generado sea relevante y específico. El campo de entrada de texto está claramente etiquetado como "Tema\*", lo que indica que es un campo obligatorio. Este enfoque guía al usuario a través del proceso de creación, asegurando que los datos esenciales sean ingresados correctamente antes de proceder. La **Figura 14** muestra cómo se ve esta parte del proceso en la interfaz del sistema.



The image shows a screenshot of a web application interface. At the top, there is a purple header bar with the text "Sistema de reactivos". Below this, the main content area is white. It features a form titled "Generar nuevo reactivo" with a subtitle "Ingrese el tema del cual generar el reactivo.". The form contains a single text input field with the label "Tema\*" and a purple button labeled "Nuevo" located at the bottom right of the form area.

Figura 14. Formulario para solicitar un nuevo reactivo.

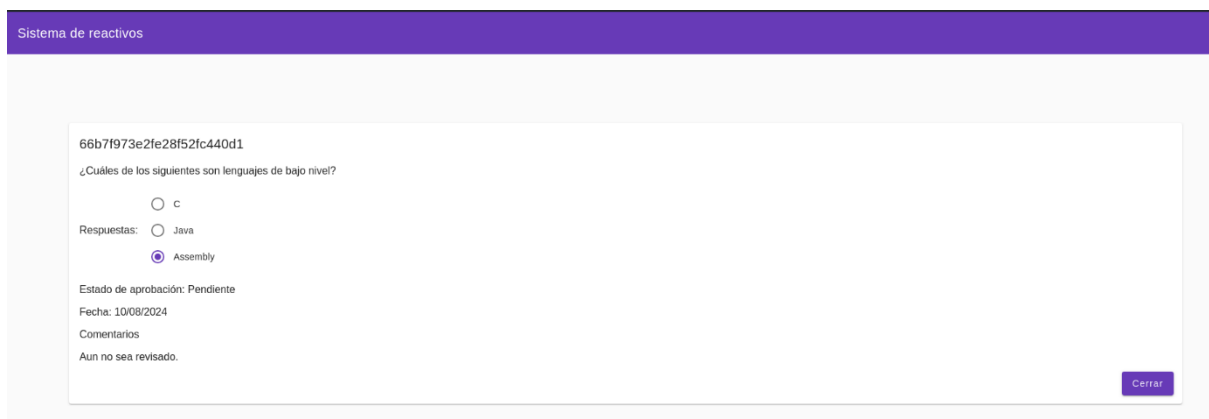
Después de abrir el formulario para la creación de un nuevo reactivo, el usuario tiene la opción de ingresar la información requerida. En esta etapa, el campo de texto obligatorio, etiquetado como "Tema\*", ya contiene datos específicos introducidos por el usuario. En este caso, se ha ingresado "lenguajes de bajo nivel" como el tema del reactivo. Este paso confirma que el usuario ha proporcionado la información necesaria para que el sistema pueda proceder con la generación del reactivo correspondiente. La **Figura 15** muestra cómo se ve el formulario una vez que el campo de tema ha sido completado con datos relevantes.



The screenshot shows a web interface titled "Sistema de reactivos". Inside, there is a form titled "Generar nuevo reactivo" with the instruction "Ingrese el tema del cual generar el reactivo." Below this is a text input field labeled "Tema\*" containing the text "lenguajes de bajo nivel". A purple button labeled "Nuevo" is located at the bottom right of the form.

Figura 15. Formulario de nuevo reactivo, con datos.

La **Figura 16** muestra el resultado de un reactivo generado. En esta interfaz se presentan tres opciones de respuesta, y se debe seleccionar la correcta. Además, incluye un estado de aprobación en "Pendiente", junto con la fecha de creación y un comentario que indica que el reactivo aún no ha sido revisado. En la parte inferior derecha, se encuentra un botón de "Cerrar" que, al presionarlo, retorna a la interfaz de la **Figura 7** y muestra la lista de reactivos.



The screenshot shows the result of a generated question. The question ID is "66b7f973e2fe28f52fc440d1". The question text is "¿Cuáles de los siguientes son lenguajes de bajo nivel?". There are three radio button options: "C", "Java", and "Assembly". The "Assembly" option is selected. Below the options, it shows "Estado de aprobación: Pendiente", "Fecha: 10/08/2024", and "Comentarios: Aun no sea revisado." A purple button labeled "Cerrar" is at the bottom right.

Figura 16. Resultado del reactivo.

La **Figura 17** muestra una interfaz que, aunque ya había sido presentada en la **Figura 13**, ha sido actualizada para incluir un nuevo reactivo que se ha generado recientemente. En esta versión más completa de la interfaz, se puede observar que ahora se presenta una lista detallada de todos los reactivos disponibles. Cada uno de estos reactivos, tanto los recién agregados como los anteriores, viene acompañado de opciones específicas que permiten al usuario decidir si desea aprobarlos o rechazarlos.

Además, es importante destacar que la interfaz no solo muestra los reactivos que están pendientes de revisión, sino que también incluye aquellos que ya han sido revisados y clasificados. Los reactivos que han sido rechazados permanecen visibles en la lista, permitiendo a los usuarios ver el comentario asociado a su rechazo. De la misma manera, los reactivos que han sido aprobados también se muestran, acompañados de los comentarios pertinentes que justifican su aceptación. Esta capacidad de mostrar tanto los reactivos aprobados como los rechazados, junto con sus respectivos comentarios, proporciona una visión completa y clara del estado actual de todos los reactivos disponibles.

### Generar nuevo reactivo

Si desea generar un nuevo reactivo presione el boton nuevo, primero

Nuevo

66b7f968e2fe28f52fc440cf

¿Cuáles de los siguientes lenguajes son considerados de bajo nivel?

Assembler

Respuestas:  Python

Java

Estado de aprobación: Pendiente

Fecha: 10/08/2024

Comentarios

Aun no sea revisado.

Aceptar

Rechazar

66b7f973e2fe28f52fc440d1

¿Cuáles de los siguientes son lenguajes de bajo nivel?

c

Respuestas:  Java

Assembly

Estado de aprobación: Pendiente

Fecha: 10/08/2024

Comentarios

Aun no sea revisado.

Aceptar

Rechazar

Figura 17. Interfaz principal con el listado de reactivos disponibles o existentes.



La **Figura 18** presenta una interfaz detallada que muestra el listado completo de los reactivos disponibles y existentes en el sistema. En esta versión actualizada de la interfaz, se evidencia que todos los reactivos han sido ya procesados, lo que significa que algunos han sido aprobados mientras que otros han sido rechazados. Para cada uno de estos reactivos, se ha agregado un comentario que indica claramente que ya han sido revisados, proporcionando una trazabilidad completa sobre el estado de cada uno de ellos.

Adicionalmente, es importante señalar que, una vez que los reactivos han sido aprobados o rechazados, los botones que anteriormente permitían realizar estas acciones desaparecen de la interfaz. Esto refuerza visualmente el hecho de que esos reactivos ya han pasado por el proceso de revisión y no requieren más acciones por parte del usuario. Esta funcionalidad asegura que la lista presente solo reactivos en su estado final, eliminando cualquier posibilidad de confusión o de cambios adicionales. De esta manera, la interfaz ofrece una representación clara, ordenada y final de todos los reactivos existentes en el sistema.

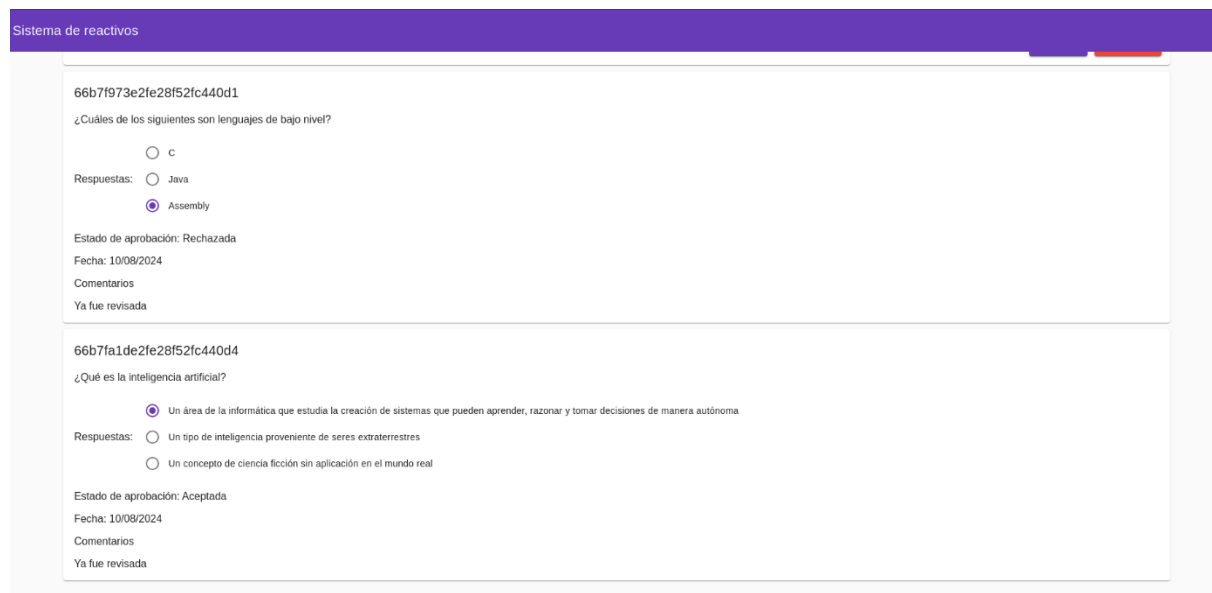


Figura 18. Interfaz principal con el listado de reactivos disponibles o existentes, se muestra los reactivos que ya fueron revisados.

Dado lo que se ha observado en las diferentes interfaces, este diseño ha sido cuidadosamente elaborado para proporcionar una experiencia de usuario amigable y fluida. El objetivo principal es facilitar la interacción del usuario con el sistema, asegurando que cada acción que realice se vea reflejada de manera clara y directa en la interfaz. Por ejemplo, cuando un nuevo reactivo es generado, este se agrega automáticamente a la lista de reactivos existentes, lo que permite al usuario ver de inmediato el progreso en la generación de contenidos. Además, si un reactivo es rechazado o aprobado, el sistema actualiza su estatus en tiempo real, permitiendo al usuario

distinguir fácilmente entre los reactivos aprobados y aquellos que han sido rechazados. Este manejo eficiente de eventos no solo mejora la claridad y orden en la presentación de los reactivos, sino que también refuerza la percepción de que el sistema está respondiendo de manera proactiva y eficiente a las acciones del usuario. De esta forma, se asegura que el usuario tenga siempre una visión precisa y actualizada del estado de los reactivos, lo cual es fundamental para un manejo eficiente y transparente del proceso de generación y revisión de contenidos.

## Conclusiones y trabajos futuros

La implementación de un sistema basado en IA para la generación de reactivos de opción múltiple ha demostrado ser una solución eficiente y efectiva en múltiples aspectos del proceso educativo. A lo largo del desarrollo de este proyecto, se han identificado los siguientes puntos clave:

1. **Eficiencia en la Generación de Reactivos:** La utilización de IA ha permitido automatizar la creación de preguntas y respuestas, reduciendo considerablemente el tiempo y esfuerzo necesario para diseñar reactivos de calidad. Esto ha facilitado la producción rápida y en gran escala, sin comprometer la precisión y relevancia del contenido.
2. **Calidad y Consistencia:** La IA garantiza un alto nivel de consistencia en la dificultad y estructura de los reactivos, lo que es esencial para mantener la equidad en las evaluaciones. La generación automática de preguntas sigue patrones predefinidos que minimizan el sesgo humano y aseguran un formato uniforme.
3. **Escalabilidad del Sistema:** La combinación de tecnologías como Node.js y TypeScript para el backend, junto con MongoDB para la gestión de datos, ha resultado en una arquitectura escalable. Esta configuración permite manejar grandes volúmenes de datos y usuarios, lo cual es crucial para adaptarse a las crecientes necesidades de las instituciones educativas.
4. **Optimización Continua del Aprendizaje:** Mediante el análisis continuo del funcionamiento de los reactivos y la intervención de la IA, se han identificado oportunidades de mejora en la formulación de preguntas y la selección de distractores. Este proceso iterativo contribuye a la optimización de la calidad educativa y la precisión de las evaluaciones.

## Trabajos Futuros

Actualmente, el frontend del sistema incluye funcionalidades como la generación, aprobación, rechazo y listado de reactivos, tanto aprobados como rechazados. Sin embargo, hay varias áreas en las que se puede expandir y mejorar el sistema:

1. **Implementación de Roles y Permisos:** Con el crecimiento previsto en el número de usuarios, es fundamental desarrollar un sistema de roles y permisos. Esto permitirá una administración más precisa y segura del acceso a las diferentes funcionalidades, estableciendo roles diferenciados para administradores, docentes y estudiantes, cada uno con sus respectivos privilegios.
2. **Desarrollo de un Servicio de Exámenes:** Una extensión clave para el futuro es la creación de un servicio de exámenes que permita a los usuarios contestar los reactivos generados por la IA. Este servicio no solo incluiría la administración de exámenes, sino también la calificación automática y la generación de reportes, proporcionando un ecosistema completo y funcional para la evaluación educativa.
3. **Autenticación y Seguridad para Roles:** Con la adición de roles, también será necesario implementar un sistema robusto de autenticación, utilizando tecnologías como OAuth o JWT para asegurar que solo los usuarios autorizados puedan acceder a recursos específicos. Esto mejorará la seguridad del sistema y protegerá la información sensible de los usuarios.
4. **Innovación y Adaptación al Alumno:** Una de las áreas más prometedoras para trabajos futuros es la adaptación del sistema al progreso del estudiante. Aunque actualmente no se cuenta con esta capacidad, implementar mecanismos de aprendizaje adaptativo permitirá que los reactivos se ajusten automáticamente según el rendimiento del alumno, ofreciendo una experiencia educativa más personalizada y efectiva. Este enfoque innovador tiene el potencial de transformar la manera en que los estudiantes interactúan con el contenido, maximizando su aprendizaje y retención.

Estas mejoras no solo fortalecerán la estructura y funcionalidad del sistema actual, sino que también sentarán las bases para futuras expansiones, facilitando una plataforma educativa más completa, segura y adaptable a diversas necesidades.

## Referencias.

1. Pedreño, A.; González, R.; Mora, T.; Pérez, E.; Ruiz, J.; Torres, A. La inteligencia artificial en las universidades: retos y oportunidades. Informe anual sobre IA y educación superior. Grupo 1MillonBot. (2024).
2. Baltazar, C. Herramientas de IA aplicables a la Educación. *Technology Rain Journal*, Vol.2 No.4, e15, (2023).
3. Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (UNESCO). <https://www.unesco.org/es/digital-education/artificial-intelligence>
4. Rouhiainen, L.: *Inteligencia artificial: 101 cosas que debes saber hoy sobre nuestro futuro*. Editorial Planeta, España (2018).
5. Miao, F.; Holmes, W.; Huang, R.; Zhang, H. *Inteligencia artificial y educación. Guía para las personas a cargo de formular políticas*. UNESCO. ISBN 978-92-3-300165-7. (2021)
6. Diaz Tito, L. P., Tito Cárdenas, J. V., Garcia Curo, G., & Boy Barreto, A. M. (2021). Inteligencia artificial aplicada al sector educativo. *Revista Venezolana de Gerencia*, 26(96), 1189-1200. <https://doi.org/10.52080/rvgluz.26.96.12>
7. Gomez Mont, C., May del Pozo, C., Martinez Pinto, C., & Martin del Campo, A. V. (2020). Artificial Intelligence for Social Good in Latin America and the Caribbean. 135. <https://www.iadb.org/>
8. Moreno Padilla, R. D. (2019). La llegada de la inteligencia artificial a la educación. *Revista de Investigación en Tecnologías de la Información*, 7(14), 260-270. <https://doi.org/10.36825/RITI.07.14.022>
9. Zhang, K., & Aslan, A. B. (2021). AI technologies for education: Recent research & future directions. *Computers and Education: Artificial Intelligence*, 2, 100025. <https://doi.org/10.1016/j.caeai.2021.100025>
10. Lai, T., Zeng, X., Xu, B., Xie, C., Liu, Y., Wang, Z., Lu, H., & Fu, S. (2023). The application of artificial intelligence technology in education influences Chinese adolescent's emotional perception. *Current Psychology*, 0123456789. <https://doi.org/10.1007/s12144-023-04727-6>
11. Park, D. M., Jeong, S. S., & Seo, Y. S. (2022). Systematic Review on Chatbot Techniques and Applications. *Journal of Information Processing Systems*, 18(1), 26-47. <https://doi.org/10.3745/JIPS.04.0232>

12. Ordoñez Cuthbert, D. K., & Sambola, D.-M. (2023b). Herramienta basada en Inteligencia de Negocios y Analíticas para la toma de decisiones académicas. Caso de Bluefields Indian & Caribbean University. *Revista Científica de FAREM-Estelí*, 46, 247-261. <https://doi.org/10.5377/farem.v12i46.16489>
13. Balderas, A., Baena-Pérez, R., Person, T., Mota, J. M., & Ruiz-Rube, I. (2022). Chatbot-Based Learning Platform for SQL Training. *International Journal of Interactive Multimedia and Artificial Intelligence*, In Press(In Press), 1. <https://doi.org/10.9781/ijimai.2022.05.003>
14. Dexon-Mckensy, S. Inteligencia Artificial en la Educación: Estado del Arte. *Revista del Caribe Nicaragüense*, N.79. doi 10.5377/wani.v39i79.16806. (2023).
15. OpenAI. ChatGPT 3.5. En <https://chat.openai.com/>.
16. Reyes, J. ChatGPT. Alcances, limitaciones y dilemas de la inteligencia artificial. *Ciencia UNAM-DGDC*. En <https://ciencia.unam.mx/leer/1415/chat-gpt-alcances-limitaciones-y-dilemas-de-la-inteligencia-artificial>. Accedido el 6 de septiembre de 2023.
17. Baltazar, C. Herramientas de IA aplicables a la Educación. *Technology Rain Journal*, Vol.2 No.4, e15, (2023).
18. Thinkster Math. (s.f.). Maths Tutoring Product Tour. Obtenido de <https://www.hellothinkster.co.in/get/how-our-math-tutor-app-works/>
19. Baron, A. (s.f.). De la IA a los expertos: Cómo Brainly mantiene las noticias falsas fuera de su plataforma. Obtenido de <https://elcanero.blogspot.com/2019/08/de-la-ia-los-expertos-como-brainly.html>
20. CTI - Content Technologies INC. (s.f.). Obtenido de [https://contenttechnologiesinc.com/Estrada, L. \(s.f.\). Usa la inteligencia artificial con estos 10 tips para estudiantes. Tecnológico de Monterrey. Obtenido de <https://conecta.tec.mx/es/noticias/nacional/educacion/usa-la-inteligencia-artificial-con-estos-10-tips-para-estudiantes#:~:text=Brainly%3A%20la%20red%20social%20del,y%20%C3%BAtiles%20a%20tus%20preguntas.>](https://contenttechnologiesinc.com/Estrada, L. (s.f.). Usa la inteligencia artificial con estos 10 tips para estudiantes. Tecnológico de Monterrey. Obtenido de https://conecta.tec.mx/es/noticias/nacional/educacion/usa-la-inteligencia-artificial-con-estos-10-tips-para-estudiantes#:~:text=Brainly%3A%20la%20red%20social%20del,y%20%C3%BAtiles%20a%20tus%20preguntas.)
21. Gradescope. (s.f.). Save Time grading. Obtenido de <https://www.gradescope.com/>
22. Tristán, L. A. (2009a). Guía de elaboración de reactivos con base en competencias educativas. México: IEIA.
23. Padilla, M. (2002). Técnicas e instrumentos para el diagnóstico y la evaluación educativa. Madrid. Ed. CCS.



24. Howitt, L.C. (1968). Prueba práctica en el aula. México: UTEHA.
25. Santillán G. (2023). Manual de elaboración de reactivos. [https://gaceta.cch.unam.mx/sites/default/files/libros/2023-08/manual-elaboracionreactivos\\_2ed-4.pdf](https://gaceta.cch.unam.mx/sites/default/files/libros/2023-08/manual-elaboracionreactivos_2ed-4.pdf)
26. Sigwart C. y col. (1990): Software Engineering: a project oriented approach. Franklin, Beedle y Associates, Inc., Irvine, California, citado en Piattini (1996)
27. G. Booch. Object-oriented analysis and design with applications. Segunda Edición. Ciudad de México: AddisonWesley, 1994, pp. 320-326
28. I. Jacobson. Object-Oriented Development in an Industrial Environment. En: ProceedingsOOPSLA' 87, special issue of SIGPLAN Notices. Vol 22, N°12, pp.183-191, 1987
29. I. Jacobson; I. Booch and G. Rumbaugh J.. The Unified Software Development Process, AddisonWesley. ISBN 0-201-57169-2, 1999
30. ChatGPT. <https://openai.com/chatgpt/>