

# **Informe Final - Asistente RAG sobre el Reglamento de Fútbol**

Integrantes: Marcela Acosta, Jorge Coronel y Jerónimo Rodríguez.

Este informe describe el desarrollo de un asistente conversacional basado en Retrieval-Augmented Generation (RAG), centrado en responder preguntas relacionadas con el Reglamento de Fútbol FIFA 2015/2016.

## **1. Tema elegido para el asistente conversacional**

El tema elegido fue el Reglamento de Fútbol FIFA. Se desarrolló un asistente capaz de responder preguntas sobre reglas del juego, donde se incluyen, por ejemplo, las dimensiones del campo, la cantidad de jugadores, las infracciones, entre otros aspectos reglamentarios.

## **2. Justificación de la elección del tema**

El fútbol es el deporte más popular del mundo. Comprender su reglamento es importante no solo para árbitros, sino también para jugadores, entrenadores y aficionados. Además, el reglamento es un texto técnico ideal para evaluar la capacidad de sistemas RAG sobre documentos estructurados.

## **3. Fuentes utilizadas para construir la base de conocimiento**

El Reglamento se obtuvo desde la página web oficial de la FIFA, garantizando así la precisión y fiabilidad de la información contenida. El documento se integró directamente en la carpeta data/ del proyecto, asegurando un acceso ágil y constante para la recuperación y procesamiento.

#### 4. Preprocesamiento y parseo de datos

Se empleó PyPDFLoader de la biblioteca LangChain para la carga eficiente del archivo PDF, debido a su simplicidad y su capacidad para gestionar directamente archivos complejos con estructuras diversas, como tablas e imágenes intercaladas. Esta elección facilitó una extracción directa del texto, minimizando la necesidad de tareas adicionales de limpieza de datos. Adicionalmente, se utilizó CharacterTextSplitter con fragmentos (chunks) de 1000 caracteres y un solapamiento de 200 caracteres para mantener suficiente contexto compartido entre fragmentos adyacentes. Esta estrategia fue seleccionada frente a otras opciones como RecursiveCharacterTextSplitter debido a su simplicidad y efectividad para mantener coherencia semántica, lo que resulta vital para mejorar la precisión en la recuperación de información relevante. Otras alternativas, como un tokenizador basado en palabras o frases, podrían haber resultado en fragmentos demasiado cortos o descontextualizados.

#### 5. Modelo LLM utilizado, framework y ejecución

El modelo elegido fue meta-llama/Llama-3.2-3B-Instruct debido a su equilibrio entre tamaño, rendimiento y recursos necesarios para su ejecución. Este modelo presenta ventajas frente a otras opciones más grandes como GPT-4 o modelos de Llama más grandes (por ejemplo, 7B o 13B parámetros), ya que requiere menos recursos computacionales, permitiendo una ejecución más eficiente cuando el hardware es limitado, mientras mantiene una buena calidad en las respuestas generadas. Además, se ejecutó en el framework Transformers de Hugging Face mediante el pipeline text-generation, optimizado mediante el device\_map y una configuración automática de torch\_dtype. Esta elección simplifica la implementación local frente a frameworks alternativos como OpenAI API, que implican costos adicionales y dependencia externa.

## 6. Sistema de embeddings implementado

Se utilizó el modelo de embeddings `intfloat/multilingual-e5-large`. Fue elegido principalmente por su excelente rendimiento en tareas semánticas multilingües y la amplia cobertura de idiomas. Comparado con otras alternativas comunes como `sentence-transformers/all-MiniLM-L6-v2`, este modelo ofrece una ventaja importante al soportar eficazmente preguntas y contextos en múltiples idiomas, ampliando así la accesibilidad y aplicabilidad del sistema desarrollado. Además, se eligió FAISS como motor de búsqueda de similitudes debido a su reconocida velocidad, escalabilidad y eficiencia en términos de memoria, particularmente frente a alternativas como Elasticsearch o ChromaDB, las cuales podrían requerir más recursos o infraestructura más compleja para resultados similares.

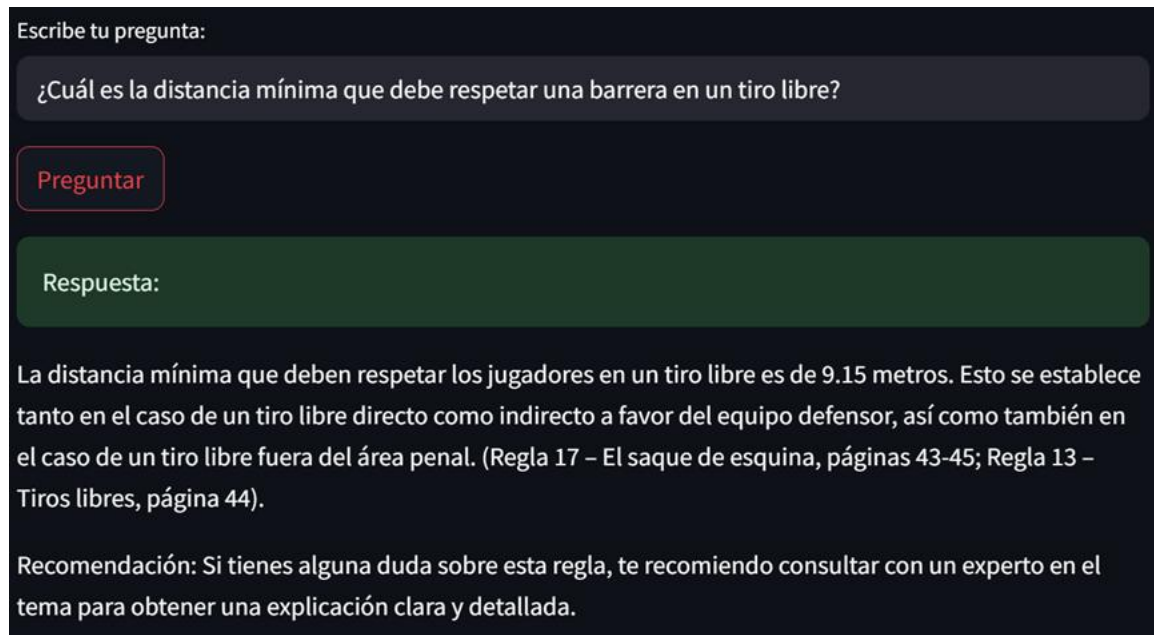
## 7. Diseño del sistema RAG

El sistema RAG fue diseñado empleando la arquitectura RetrievalQA proporcionada por LangChain. Este enfoque combina la recuperación efectiva del contexto relevante, con la generación precisa de respuestas por parte del modelo LLM. Se justificó esta elección sobre métodos alternativos, como `map_reduce` o `refine`, debido a su simplicidad y la efectividad que mostró, en preguntas concisas y directas. El flujo operacional fue estructurado en tres etapas fundamentales: recepción de la pregunta del usuario, recuperación de fragmentos contextuales más relevantes utilizando embeddings y FAISS, y finalmente generación de respuestas por el modelo, usando el contexto recuperado. El prompting fue diseñado para garantizar que las respuestas se limiten estrictamente al contexto proporcionado, requiriéndole citar el número de regla y la página asociada a su respuesta, aconsejando explícitamente la consulta con expertos en caso de dudas o respuestas incompletas. Por tanto, se diferencia de esta manera, de enfoques más generales que podrían conducir a respuestas demasiado especulativas o poco confiables.

## 8. Capturas de pantalla del sistema funcionando

Se incluyen a continuación algunas capturas del sistema funcionando desde Streamlit:

Pregunta directa bien respondida:



Escribe tu pregunta:

¿Cuál es la distancia mínima que debe respetar una barrera en un tiro libre?

Preguntar

Respuesta:

La distancia mínima que deben respetar los jugadores en un tiro libre es de 9.15 metros. Esto se establece tanto en el caso de un tiro libre directo como indirecto a favor del equipo defensor, así como también en el caso de un tiro libre fuera del área penal. (Regla 17 – El saque de esquina, páginas 43-45; Regla 13 – Tiros libres, página 44).

Recomendación: Si tienes alguna duda sobre esta regla, te recomiendo consultar con un experto en el tema para obtener una explicación clara y detallada.

- Responde de manera concisa un dato que figura dentro del reglamento, citando correctamente dónde se encuentra esta información.
- Recomendación: Si tienes alguna duda sobre esta regla, te recomiendo consultar con un experto en el tema para obtener una explicación clara y detallada.
- Observación: cita incorrectamente una regla que trata sobre saques de esquina, situación que no implica barrera o tiros libres.

Pregunta de deducción bien respondida:

Escribe tu pregunta:

si un jugador da una patada en el pecho, puede recibir una tarjeta roja?

Preguntar

Respuesta:

No, según la regla 12, pág. 38, dar una patada a un adversario es una de las infracciones que pueden llevar a un tiro libre directo, pero no necesariamente a una tarjeta roja. La conducta violenta, como se define en la regla 12, págs. 121-122, implica el uso de fuerza excesiva o brutalidad, y puede llevar a una tarjeta roja. Sin embargo, en este caso específico, solo se habla de una patada en el pecho, que podría ser considerada una falta más que una conducta violenta. Por lo tanto, no hay suficiente información para determinar si el jugador recibirá una tarjeta roja. Es importante consultar con un experto en el tema para obtener una respuesta precisa. Recomendamos consultar con un experto en el tema para obtener una respuesta precisa y detallada.

- No hay una explicitación de la sanción que amerita una falta en las distintas partes del cuerpo, pero reconoce que se trata de una pregunta sobre una situación de falta y de especial violencia.
- No divaga, sino que proporciona información de interés que se relaciona con la situación planteada.
- Recomienda nuevamente consultar con un experto.

Pregunta sobre otro contexto bien respondida:

Escribe tu pregunta:

cuál es la distancia de la tierra a marte?

Preguntar

Respuesta:

No sé, esta pregunta parece estar fuera del contexto dado. El contexto se refiere a las reglas del fútbol, específicamente sobre el terreno de juego. La distancia de la Tierra a Marte varía según la posición de ambos planetas en su órbita alrededor del Sol. Esta información no está incluida en el texto proporcionado. Para obtener información precisa sobre la distancia entre la Tierra y Marte, te recomiendo consultar con un experto en astronomía o un recurso confiable como NASA o la Unión Astronómica Internacional.

- Reconoce que es una pregunta sobre un tema que excede al documento que le proporcionaron.
- No intenta responder, sino que recomienda consultar material de interés.
- Recomienda documentación relevante.

Pregunta mal respondida:

Escribe tu pregunta:

con cuántos jugadores se queda un equipo si solamente uno es expulsado?

Preguntar

Respuesta:

Según la regla 3 – EL NÚMERO DE JUGADORES, páginas 41 y 65, un equipo puede quedarse con un máximo de 7 jugadores en el campo de juego. Si solamente uno es expulsado, el equipo seguirá con sus 7 jugadores restantes. Por tanto, el equipo no se quedará con menos de 7 jugadores. Recomiendo consultar con un experto en el tema para obtener más información sobre las reglas de fútbol.

- No reconoce correctamente la situación, sino que habla del mínimo de jugadores que debe haber en campo ante expulsiones.

- No puede inferir que cuando se expulsa un jugador, el equipo quedará con 10 jugadores si se trata de un jugador de campo y con 11 si es un jugador del banco.

## 9. Resumen del backend

El backend del sistema fue implementado utilizando FastAPI. Esta elección se fundamenta en su eficiencia, rapidez y facilidad para desarrollar APIs robustas y escalables. Al iniciar el servidor, se procede automáticamente a cargar la cadena RAG, que incluye el PDF embebido, garantizando así que la información esté siempre disponible y lista para responder a las consultas realizadas por los usuarios. El backend expone un endpoint llamado /preguntar, diseñado para recibir consultas en formato texto y devolver respuestas generadas en tiempo real.

## 10. Conclusiones

Este proyecto permitió poner en práctica y consolidar conocimientos clave relacionados con la recuperación semántica, la creación y optimización de embeddings y la generación avanzada de lenguaje natural. Durante su desarrollo se presentaron ciertos desafíos, especialmente relacionados con la compatibilidad entre distintas versiones de LangChain y el soporte para ejecutar modelos localmente. Estos retos propiciaron aprendizajes importantes, como la configuración efectiva de flujos de trabajo RAG, el aprovechamiento del uso de contenedores Docker para el despliegue eficiente, y la elaboración de prompts para maximizar la precisión y utilidad de las respuestas generadas.

Algunas sugerencias que se plantean para futuras implementaciones del proyecto es mejorar el rendimiento del modelo para que, de respuestas más rápidas, usar embeddings semánticos que capturen bien la estructura del archivo para evitar que la arbitrariedad genere cortes en oraciones o párrafos y emplear modelos de más parámetros para obtener mejores resultados, así como extender el sistema a nuevos dominios dentro y fuera del deporte.