Rúbrica Tarea 3 – Chatbot RAG con Artículos de Wikipedia
Puntaje total: 7.0 puntos

Categoría	Criterio	Puntaje
Punto base	Entrega válida, accesible y dentro del plazo	1.0
Backend	**Scraping del artículo desde Wikipedia (automatizado, sin errores graves):** El sistema debe obtener correctamente el contenido del artículo ingresado. Para evaluarlo, se puede probar con un artículo distinto y ver si se responde con base en ese contenido.	0.6
Backend	**Preprocesamiento y limpieza del texto** Las respuestas deben estar limpias, sin etiquetas HTML ni fragmentos irrelevantes. Si el modelo responde con texto ordenado y coherente, se asume que hubo una limpieza adecuada.	0.4
Backend	**Generación correcta de embeddings usando nomic-embed-text** Debe usarse la API de embeddings con el modelo 'nomic-embed-text'. Esto se puede verificar viendo el código y confirmando que las respuestas varían según la parte del artículo consultada.	0.6
Backend	**Base de datos vectorial funcional y consultable** Se espera que el sistema almacene los fragmentos vectorizados y luego recupere los más relevantes según la consulta. Esto se nota si al hacer preguntas distintas se obtienen respuestas distintas.	0.6
Backend	**Recuperación de contexto relevante con RAG** Las respuestas del modelo deben estar basadas en el contenido real del artículo. Si entrega datos precisos y relevantes, es señal de que la recuperación está funcionando bien.	0.8
Backend	**Llamado a la API del LLM con contexto adecuado** El sistema debe enviar el contexto recuperado junto con la consulta	0.7

	al modelo 'integracion'. Si el modelo responde de forma coherente y detallada, se puede asumir que esto está bien implementado.	
Backend	**Manejo de errores y timeouts con feedback útil al usuario** Ante fallas como consultas vacías o errores de red, el sistema debe mostrar un mensaje claro al usuario. No debe quedarse pegado ni lanzar errores visibles.	0.3
Frontend	**Diseño tipo chatbot funcional e intuitivo**	0.5
Frontend	**Campos para ingresar link y consulta correctamente conectados**	0.4
Frontend	**Visualización clara de la respuesta generada**	0.4
Frontend	**Manejo de estados (carga, error, etc.)**	0.2
Integración	**Flujo completo funcional: desde scraping hasta respuesta en pantalla**	0.5
Bonus	A criterio del corrector	0.3