

RAG con Spring AI

Un curso de Academia para Cluster Canarias.

Por Alberto Basalo

| Noviembre 2025

CONEXIÓN

¿Haces preguntas sobre actualidad?

- > Los LLM tienen conocimiento limitado a su fecha de corte.
- > Cualquier respuesta posterior puede ser incorrecta o inventada.

¿Necesitas certeza?

- > Los LLM responden de forma probabilística, no determinista.
- > A veces responden con confianza a cosas que no saben.

¿Cuántos tokens consumes?

- > Consultar información externa puede ser costoso en tokens.
- > Una base de conocimiento propia reduce costes.

¿Tienes problemas legales/confidencialidad?

- > No envíes datos sensibles a terceros.
- > Usa tu propia base de conocimiento para cumplir normativas.

Por qué RAG?

- > Con **RAG** (*Retrieval-Augmented Generation*) añadimos contexto desde nuestras fuentes.

¿Qué ventajas ofrece?

- > Permite combinar IA generativa con bases de conocimiento propias para respuestas más precisas y relevantes sin enviar datos sensibles a terceros.

CONCEPTOS

¿Qué es RAG?

- > Es un patrón de arquitectura que **recupera** información relevante de una fuente para **aumentar** el contexto antes de **generar** la respuesta con un modelo de lenguaje.

Consulta del usuario → Recuperación de contexto → Generación aumentada

- > **Retriever**: busca información relevante en una base vectorial.
- > **Augmenter**: combina la información recuperada con la del usuario.
- > **Generator**: usa el modelo para producir la respuesta final.

Integración con herramientas

> RAG puede complementarse con **herramientas (tools)** específicas que amplían las capacidades del modelo, como búsqueda web o acceso a APIs internas.

Esto permite integrar RAG con servicios externos de forma controlada y auditable.

RAG y Vector Stores

- > Los datos se representan como vectores numéricos con valor semántico en un **Vector Store**.
- > Se compone de dos fases:
 - > **Ingestión:** convertir documentos en vectores y almacenarlos.
 - > **Consulta:** buscar vectores similares a la consulta del usuario.

Usuario - (Ingestión → Vector Store)

Usuario → (Retriever → Vector Store) → ChatClient → LLM → Respuesta

Almacenamiento vectorial

- > El `VectorStore` permite representar documentos en forma de vectores, capturando su significado semántico.
- > Estos `vectores` se utilizan para encontrar documentos similares a una consulta dada mediante técnicas de búsqueda vectorial.

Flujo típico de RAG

- > 1 **Ingestión**: se cargan documentos relevantes.
- > 2 **Troceado (chunking)**: se dividen en fragmentos semánticos.
- > 3 **Embeddings**: cada fragmento se transforma en vector numérico.
- > 4 **Indexado**: los vectores se almacenan en un `VectorStore`.
- > 5 **Recuperación**: se buscan los más similares a la consulta.
- > 6 **Generación aumentada**: se pasa el contexto recuperado al modelo.

Spring AI for Vector Stores

- > Ofrece integración con varios proveedores de almacenamiento vectorial.
- > Permite configurar fácilmente la ingestión, consulta y gestión de vectores.
- > Es independiente del motor de embeddings, del almacén de vectores... y del modelo LLM.
- > **ChatClient** para las llamadas al modelo.
- > **VectorStore** para almacenar e indexar documentos.
- > **RagClient** para combinar ambos procesos.

CONCRETANDO

Demo AstroBiblia

- > Rag Controller
- > Tool Controller
- > Vector Controller

Práctica: Extiende la aplicación de blogs

- > [] Busca información sobre el tema principal en Wikipedia.
- > [] Guarda el contenido en el Vector Store.
- > [] Busca información previa al generar el post del blog.

CONCLUSIONES

Próxima lección:

Prácticas en el mundo real.

No es magia, es tecnología.

Alberto Basalo