# Día 15 · Tema 6 — RAG: recuperación aumentada con generación

Eduardo C. Garrido-Merchán · ecgarrido@comillas.edu Universidad Pontificia Comillas · ICADE

## 1. El problema del conocimiento privado

Un LLM se entrena con datos hasta una fecha de corte. No conoce tu CRM, tus circulares ni el calendario de la oficina de Burgos. Hay dos vías para incorporar conocimiento privado:

- **Fine-tuning**: reentrenar (o LoRA) sobre tus datos. Caro, lento, y cada actualización requiere reentrenar.
- **RAG** (*retrieval-augmented generation*): mantener los documentos en una base vectorial, recuperar los relevantes en cada consulta y pasarlos como contexto.

RAG gana el 90 % de las veces porque las actualizaciones son incrementales (indexar el doc nuevo) y el coste por consulta es bajo.

## 2. Pipeline RAG canónico

Cuatro etapas:

1. **Indexación** (offline): trocear el corpus en *chunks* $\{d_j\}$ y calcular embeddings $\phi(d_j) \in \mathbb{R}^D$. Guardar en una base vectorial (FAISS, Qdrant, Pinecone, pgvector).
2. **Recuperación**: para una query $q$, recuperar los top-$k$ chunks

$$\text{top}_k(q) \;=\; \arg\operatorname*{máx}^{(k)}_{j \in [J]} \; \cos\big(\phi(q), \phi(d_j)\big). \tag{1}$$

3. **Generación**: el LLM responde condicionado al contexto recuperado,

$$\hat{y} \;\sim\; p_\theta\big(y \,\big|\, q,\, d_{j_1}, \ldots, d_{j_k}\big). \tag{2}$$

4. **Atribución**: devolver al usuario los $d_{j_i}$ usados como referencias verificables.

| | |
|---|---|
| $q$ | query del usuario (texto) |
| $d_j$ | chunk del corpus; típicamente 200–800 tokens |
| $\phi$ | modelo de embeddings (Día 08) |
| $k$ | número de chunks recuperados; típicamente 3–10 |
| $J$ | tamaño total del corpus indexado |

## 3. Chunking: el detalle que importa

Trocear un documento mal es *la* causa principal de RAG mediocre. La ventana óptima es la unidad coherente más pequeña: *cláusula* en contratos, *sección* en manuales, *turno* en conversaciones. Reglas defensivas:

- Chunks de 300–600 tokens; *overlap* de 50–100 tokens entre consecutivos para preservar contexto.
- Conservar metadatos: título de documento, sección, fecha. Estos atributos se incluyen junto al texto en el prompt.
- No partir tablas ni listas en mitad de fila.

## 4. Patrones de retrieval

- **Dense retrieval**: $\cos(\phi(q), \phi(d_j))$ con un único modelo de embeddings.
- **Hybrid**: combinar dense con BM25 (palabras clave), score $= \alpha \cos + (1-\alpha)$ BM25. Resuelve casos donde el dense pierde términos raros (códigos de producto, nombres propios).
- **Cross-encoder rerank**: tras un retrieval inicial de $k_0$ candidatos ($k_0 = 50$), aplicar un modelo más caro $\text{score}_{\text{ce}}(q, d_j)$ para reordenar y quedarse con los top-$k$ ($k = 5$).
- **Multi-query**: el LLM reescribe $q$ en $m$ variantes; se recuperan top-$k$ por variante y se unen.

## 5. Caso BBVA: dimensiones reales

- Corpus: 12 GB de normativa interna, contratos y respuestas FAQ.
- Chunks: $\sim$ 350 000, cada uno $\sim$ 400 tokens.
- Embedding: `intfloat/multilingual-e5-large`, $D = 1024$.
- Almacenamiento: $\sim$ 1,5 GB en pgvector (con cuantización a 8-bit).
- Consulta: top-10 + rerank a top-3 + generación con Sonnet 4.7.
- Latencia: ~1.8 s por consulta; coste $\sim$ 0,02 $/consulta.

### 6. Métricas de RAG

**Retrieval quality**: Recall@$k$ (¿el chunk gold está en los top-$k$?), MRR (Mean Reciprocal Rank). **Generation quality**: *faithfulness* (¿la respuesta está soportada por los chunks?), *relevance* (¿responde a la pregunta?). **End-to-end**: *answer correctness* (la única que ve el cliente).

**Idea clave.** Si *Recall@5* en tu evaluador interno es $\geq 0{,}85$, el cuello de botella ya es la generación, no el retrieval. Si es $< 0{,}7$, mejorar el embedding y el chunking antes que el LLM.

### 7. Cuándo RAG no basta

Tres casos donde fine-tuning es preferible: **(a)** estilo muy específico (legal corporativo, jerga técnica particular) — RAG no enseña al modelo *cómo* escribir, sólo *qué*; **(b)** latencia ajustada ($< 200$ ms) sin tiempo para retrieval + generation; **(c)** el conocimiento es vasto y se consulta entero (toda la normativa relevante para cada caso) — entonces el contexto se desborda incluso con $k = 20$.