Deep Learning para Business Analytics

Día 17 · Tema 6 — Sistemas multi-agente en producción

Coste, latencia, fallbacks y monitor.

Eduardo C. Garrido-Merchán

Profesor Propio Adjunto, Métodos Cuantitativos, ICADE

ecgarrido@comillas.edu

Madrid, Día 17

BBVA Santander Telefónica Iberdrola MERCADONA endesa

ejemplos del día

COMILLAS UNIVERSIDAD PONTIFICIA
IGAI ICADE CIHS

Lo que vamos a cubrir hoy.

1. Arquitectura completa

2. Coste y latencia desglosados

3. Fallbacks y robustez

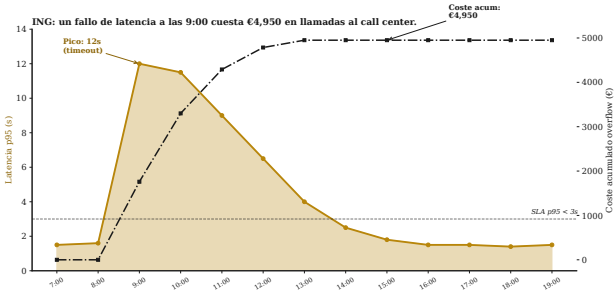
4. Observabilidad y monitor

Al final del día: sabes diseñar la arquitectura completa de un asistente corporativo multi-agente, con fallbacks y monitor.

Parte 1 / 4

Arquitectura completa

El chatbot de ING procesa 2M consultas/mes: un fallo cuesta €50K.



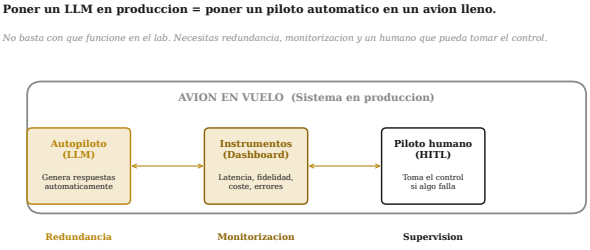
Incidente. ⚠️ A las 9:00, un pico de latencia a 12 s desborda el chatbot. Las consultas caen al call center humano.

Coste directo. 💰 Cada llamada overflow cuesta ~€5,50. En 3 horas se acumulan 900 llamadas extra = €4 950 solo en ese episodio.

SLA. En producción, la ley es: p95 latencia <3 s, disponibilidad >99,5 %. Cada minuto caído tiene coste directo medible.

Idea clave. Producción no es “funciona en el lab”. Es latencia, coste por minuto y SLA contractual.

Poner un LLM en producción = poner un piloto automático en un avión lleno.



Autopiloto = LLM. 🛡️ Genera respuestas automáticamente, pero no puede volar solo.

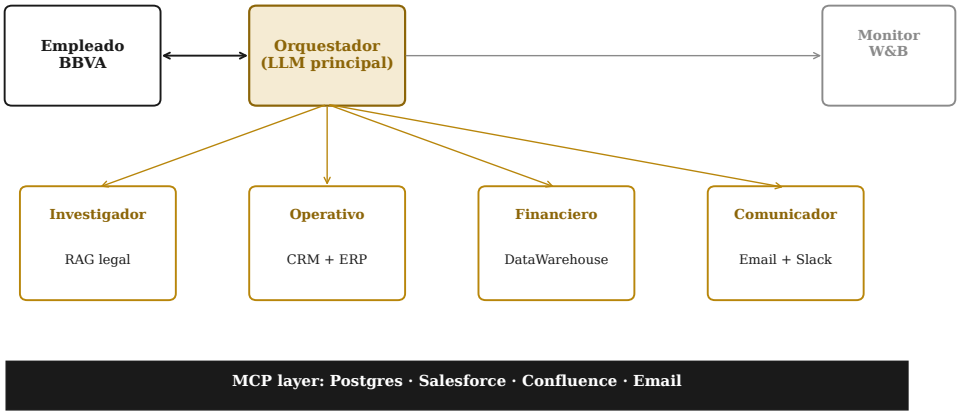
Instrumentos = Dashboard. 👁️ Latencia, fidelidad, coste por sesión, tasa de fallback. Si un indicador se sale de rango, salta alerta.

Piloto humano = HITL. Un humano que puede tomar el control si la IA alucina o si un edge case escapa al sistema.

Idea clave. Tres ingredientes no negociables: redundancia, monitorización y supervisión humana.

Asistente corporativo BBVA 🏭: orquestador + 4 subagentes + MCP.

Arquitectura multi-agente: 1 orquestador, 4 subagentes, capa MCP, monitor.



Concepto → aplicación: cinco casos canónicos.

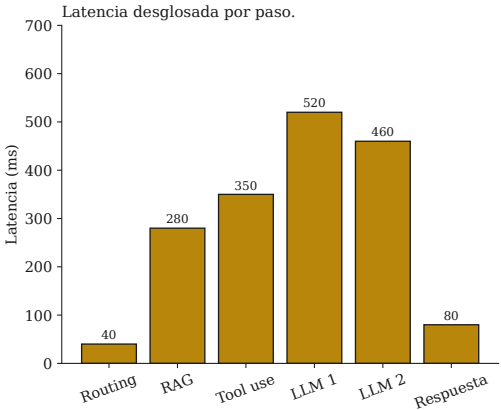
Caso	Subagentes	MCP requeridos
Asistente interno empleados	Buscador + redactor.	RAG corp., Confluence, mail.
Agente comercial CRM	Investigador + escritor + analista.	Salesforce, LinkedIn, mail.
Soporte sistema tickets	Triage + buscador + redactor.	Zendesk/Jira, base FAQ.
Agente analista financiero	SQL + visualizador + redactor.	BigQuery, Tableau, mail.
Auditor regulatorio	Investigador + revisor + escritor.	Confluence, AI Act DB.

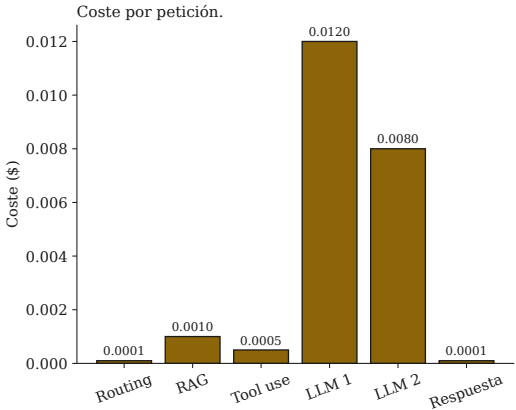
Idea clave. Cualquier caso real es *este patrón* con subagentes y MCPs específicos.

Parte 2 / 4

Coste y latencia desglosados

Latencia ⏱️ y coste 💰 por paso del pipeline.





Ejemplo a mano: cuánto cuesta una petición compleja.

Asistente comercial BBVA 💰, una petición.

$$\underbrace{0{,}0001}_{\textit{routing}} + \underbrace{0{,}001}_{\textit{RAG}} + \underbrace{0{,}0005}_{\textit{tool}} + \underbrace{0{,}012}_{\textit{LLM1}} + \underbrace{0{,}008}_{\textit{LLM2}} + \underbrace{0{,}0001}_{\textit{respuesta}} = \mathbf{\$\,0{,}022}.$$

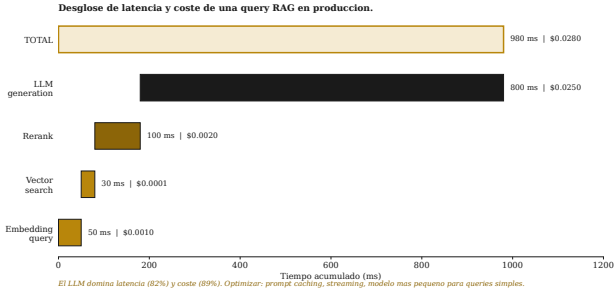
Volumen. 5 000 peticiones/día × 22 días laborables = 110 000 peticiones/mes.

Coste mensual. 110 000 × 0,022 = **$2 420** en LLMs.

Latencia. 40 + 280 + 350 + 520 + 460 + 80 = **1 730** ms end-to-end.

Idea clave. Tu pitch al CFO de BBVA: $30K/año por un asistente para 1.000 empleados con respuesta en 2 segundos.

Desglose de latencia y coste de una query RAG en producción.



Pipeline. ⏱️ Embedding (50 ms, $0,001) → vector search (30 ms) → rerank (100 ms, $0,002) → LLM (800 ms, $0,025).

Total. 980 ms y $0,028 por query. El LLM domina el 82 % de la latencia y el 89 % del coste.

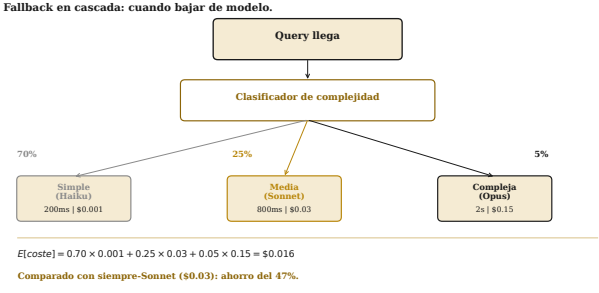
Optimizar. 💰 Prompt caching, streaming para percepción, modelo más pequeño para queries simples (routing).

Idea clave. Conocer el desglose es imprescindible para saber dónde cortar latencia y coste.

Parte 3 / 4

Fallbacks y robustez

Fallback en cascada: cuándo bajar de modelo.



Idea. 🤖 Un clasificador de complejidad enruta cada query al modelo adecuado.

Modelos. Simple → Haiku (200 ms, $0,001). Media → Sonnet (800 ms, $0,03). Compleja → Opus (2 s, $0,15).

Cálculo. 💰 $E[\text{coste}] = 0{,}70 \times 0{,}001 + 0{,}25 \times 0{,}03 + 0{,}05 \times 0{,}15 = \$0{,}016$.

Comparado con siempre-Sonnet ($0,03): ahorro del 47 %.

Idea clave. El routing inteligente es la palanca de coste más potente en producción.

Cinco patrones de fallback que un sistema serio implementa.

Cinco patrones de fallback que un sistema multi-agente serio implementa.

Patrón	Cuándo activarlo	Acción
Timeout subagente	Subagente tarda más de 30 s.	Cancelar y devolver respuesta parcial.
Score bajo en RAG	Top-1 chunk con score < 0,4.	Respuesta: «no encontré información».
Confianza baja LLM	Logprob promedio < -3.	Escalar a humano (HITL).
Loop detectado	Mismo tool 3 veces seguidas.	Detener y avisar al orquestador.
Coste explotado	Sesión supera $0,50.	Resumir y devolver lo mejor.

Tres errores frecuentes en sistemas multi-agente.

Error 1 — Loops entre subagentes ⚠️. El orquestador llama al investigador, el investigador devuelve nada, el orquestador insiste. Detector: contar llamadas a la misma tool con misma entrada.

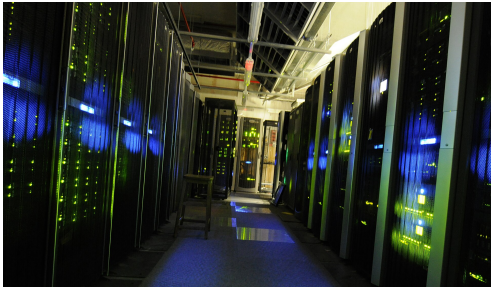
Error 2 — Sin presupuesto por sesión 💰. Una conversación puede llegar a 30 turnos y $1 de coste. Solución: límite por sesión, hard-fail si se supera.

Error 3 — Tools sin schema ⚠️. El subagente intenta llamar con JSON inválido. Solución: JSON Schema en cada tool y validación antes de ejecutar.

Idea clave. Estos tres errores son responsables del 80 % de las regresiones en producción.

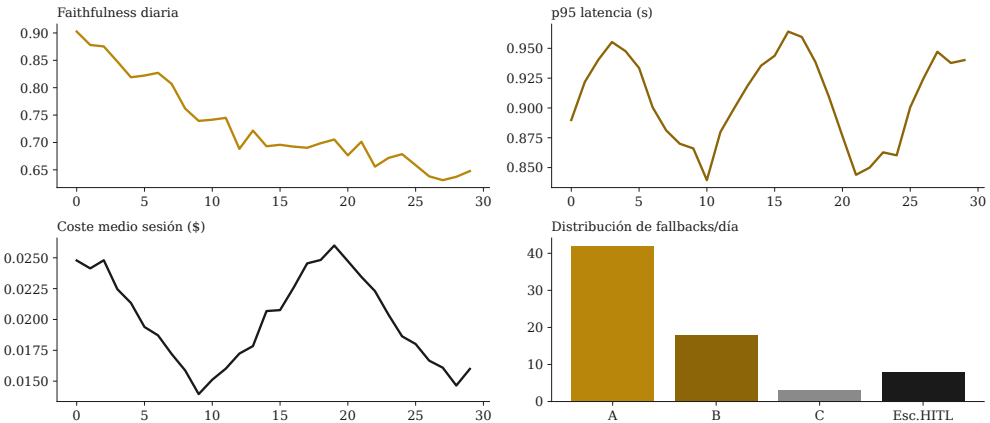
Parte 4 / 4

Observabilidad y monitor



Cuatro paneles del dashboard de un asistente en producción.

Cuatro paneles del dashboard de un asistente en producción.



Lo que el alumno se lleva a casa de este día.

Concepto 1. Un asistente corporativo serio 🏭 es un grafo de orquestador + subagentes + MCP + monitor.

Concepto 2. El coste 💰 se desglosa por paso; el cuello suele estar en LLMs múltiples.

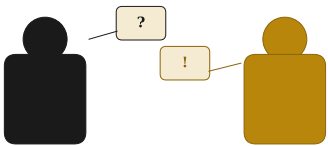
Concepto 3. Cinco patrones de fallback 🛡️ (timeout, low-score, low-confidence, loop, budget) son no-negociables.

Concepto 4. La observabilidad 👁️ (faithfulness, latencia p95, coste medio, fallbacks/día) determina si el asistente vive o muere.

Para la práctica. Diseñar la arquitectura completa del asistente del proyecto AI-first del grupo.

Próximo día. Día 18 (Tema 7): regulación y ética, cierre del Bloque III.

Pausa para debatir: ¿Cuándo es lícito apagar un asistente desplegado?



La pregunta. Tu sistema sirve a 100K usuarios y detectas un sesgo. ¿Lo apagas inmediatamente o lo corriges en caliente?

Posición A. Apagar es obligación regulatoria si el sesgo es discriminatorio.

Posición B. Corregir sin apagar es viable con A/B y rollback en horas.

Reglas. 5 minutos, dos turnos de palabra por bando, móviles boca abajo. Yo modero, no participo.

Hasta el Día 18.

AI Act, gobierno y auditoría.

Eduardo C. Garrido-Merchán

ecgarrido@comillas.edu