Facsímil 11

IA multimodal

Capítulo 12

Recapitulación multimodal

686f6c61 · 2026

Capítulo 12: Recapitulación multimodal

Qué deberías poder hacer al terminar

Este capítulo cierra el facsímil. Si has llegado hasta aquí, ya no estamos preguntando “qué es una imagen para un modelo” o “cómo se evalúa un audio”. La pregunta ahora es más profesional:

> ¿Puedo defender un release multimodal con evidencias descargables, métricas, riesgos, controles y una decisión clara?

Un sistema multimodal serio no se valida con una captura bonita. Se valida conectando todo lo aprendido: contrato de entrada, representación, recuperación, grounding, audio, vídeo, acción, evaluación, privacidad, seguridad, coste, latencia y operación.[1]

Al terminar deberías poder hacer esto:

RESULTADO DE APRENDIZAJE	EVIDENCIA DE QUE LO SABES HACER
Recapitular el facsímil completo.	Conectas capítulos 01-11 con un caso real de release.
Diseñar un release gate multimodal.	Combinas calidad, riesgo, evidencias, coste, latencia y fallos.
Separar `pass`, `review` y `block`.	No publicas todo porque algunos casos funcionen bien.
Identificar evidencias faltantes.	Nombras contrato, golden set, retrieval manifest, slot eval, policy decision o lineage.
Proponer una remediación.	Dices qué archivo, control, métrica o prueba cambiarías.
Defender la decisión.	Entregas una decision card y un paquete de evidencias.
Explicar baseline frente a remediación.	Sabes por qué un caso pasa de bloqueado a publicable y qué evidencia lo justifica.
Convertir una revisión en candidato.	Preparas un change request con SLI/SLO, owner, aprobadores, rollback y versionado.

La frase central:

> La multimodalidad no termina cuando el modelo responde. Termina cuando puedes explicar por qué esa respuesta puede, o no puede, llegar a producción.

Lectura de ingeniería: cerrar es saber defender una decisión

Un cierre de facsímil no debería ser solo un resumen. Para un ingeniero de IA, cerrar significa poder mirar un sistema completo y decir qué publicaría, qué dejaría en revisión, qué bloquearía y qué evidencia pediría antes de cambiar de opinión. Esa habilidad es más valiosa que memorizar nombres de arquitecturas, porque los modelos cambian rápido y los criterios de ingeniería permanecen más tiempo.

El cierre del facsímil fuerza una incomodidad deliberada: el baseline bloquea, la remediación mejora y el release candidate solo pasa cuando se añaden evidencias, contract tests, SLI/SLO, owner, aprobadores y rollback. Esa secuencia enseña una idea importante: la calidad no es un estado emocional. Es una decisión que debe poder reproducirse, discutirse y auditarse.

Qué debería saber defender el lector

Al cerrar este facsímil, el lector debería poder defender cinco cosas. La primera: por qué una modalidad entra en el sistema y qué evidencia aporta. La segunda: qué representación se usa y qué se pierde al convertir la señal. La tercera: qué controles separan observación, recuperación, decisión y acción. La cuarta: qué métricas y slices determinan si el sistema se publica. La quinta: qué política protege privacidad, seguridad y operación.

Si alguna de esas piezas falta, el sistema puede parecer avanzado pero queda cojo. Un VLM sin contrato puede inventar campos. Un RAG sin permisos puede filtrar fuentes internas. Un sistema de voz sin confirmación puede ejecutar sobre una transcripción dudosa. Un agente de pantalla sin arnés puede hacer click en el sitio equivocado. Una eval sin slices puede ocultar errores críticos. Cerrar el facsímil obliga a juntar esas piezas para que no queden como capítulos aislados.

Cómo leer un cierre como una revisión técnica

Si esta revisión se hiciera en un equipo real, no bastaría con enseñar el informe final. Habría que abrir el diff, revisar los casos que cambian, mirar qué métrica sigue cerca del umbral, comprobar que el código ejecuta, entender qué política se aplicó y decidir qué pasaría si cambia el modelo visual, el OCR, el ASR, el índice, el prompt o una tool. Ese hábito es el que quiero que el lector se lleve.

El cierre sigue tres pasos. El primero, diagnóstico: no todo baseline bloqueado es un fracaso; a veces es la señal correcta de que faltan evidencias. El segundo, remediación: mejorar no es tocar un prompt, sino añadir campos, umbrales, fuentes y decisiones. El tercero, release: publicar exige manifest, tests, SLI/SLO, rollback, owner y límites conocidos. Esa secuencia se parece a una revisión de ingeniería, no a un ejercicio de completar huecos.

Qué se lleva alguien a su trabajo o a clase

El artefacto final debería poder reutilizarse. Puede convertirse en plantilla de evaluación para un RAG multimodal, en checklist de privacidad para capturas, en gate de release para VLMs, en política de computer use o en rúbrica docente para una práctica. Esa es la vara de medir: si el lector no puede llevarse algo a un proyecto real, el cierre no ha hecho suficiente.

Por eso el capítulo no termina diciendo “ya sabes multimodalidad”. Termina con una pregunta más seria: ¿puedes defender una decisión técnica cuando hay varias modalidades, varios riesgos y varias evidencias incompletas? Si puedes, el facsímil ha hecho su trabajo. Si no puedes, no pasa nada: vuelve al capítulo donde se rompió la cadena. Esa es precisamente la utilidad del facsímil.

El mapa del facsímil

Este facsímil no era una colección de temas sueltos. Tenía una progresión:

CAPÍTULO	PREGUNTA QUE RESPONDE	QUÉ TE LLEVAS
01	¿Qué cambia cuando la IA ve, oye y actúa?	Modalidades, contratos y límites.
02	¿Cómo entra una imagen en un modelo?	Píxeles, patches, embeddings y coste de entrada.
03	¿Cómo se alinean texto e imagen?	CLIP, contraste, ranking y búsqueda.
04	¿Cómo habla un VLM con un LLM?	Encoder visual, conector, tokens visuales y contrato de llamada.
05	¿Cómo se leen documentos reales?	OCR, layout, tablas, páginas y evidencias.
06	¿Cómo se recupera evidencia multimodal?	Índices, RAG, ACL, grounding y manifest.
07	¿Qué implica usar voz?	ASR, turnos, latencia, slots y conversación.
08	¿Qué implica usar vídeo?	Frames, eventos, memoria temporal y evaluación temporal.
09	¿Qué pasa si el agente mira pantallas y actúa?	Computer use, permisos, approval cards y tool traces.
10	¿Cómo se evalúa todo eso?	Golden sets, slices, coste, latencia y gates.
11	¿Cómo se opera sin exponer datos?	Privacidad, threat model, policy-as-code, lineage y runbook.

La parte visual no aparece por estética. ViT ayuda a entender por qué una imagen puede convertirse en secuencia de patches, y CLIP ayuda a entender por qué texto e imagen pueden compartir un espacio de comparación.[2]

Si tuviera que resumir el facsímil en una sola idea:

> Multimodalidad es ingeniería de evidencias. La modalidad aporta señal; el sistema debe conservar trazabilidad.

Lo que no deberías olvidar

Hay varias trampas recurrentes:

TRAMPA	POR QUÉ SEDUCE	POR QUÉ ROMPE PRODUCCIÓN
“El modelo ve la imagen, ya está.”	Parece que el VLM resuelve todo.	No sabes qué región, página o frame sostuvo la respuesta.
“Hacemos OCR y lo metemos en RAG.”	Es rápido de prototipar.	Pierdes layout, tablas, bboxes, páginas y permisos.
“Si el promedio sube, publicamos.”	Simplifica la decisión.	Puede esconder un slice crítico: vídeo largo, audio ruidoso, PII o computer use.
“El agente decidirá si puede actuar.”	Suena autónomo.	Permisos, egress y aprobación deben vivir fuera del modelo.
“Guardemos todo para depurar.”	Ayuda durante la demo.	Crea un repositorio de capturas, audio, documentos y trazas sensibles.
“Ya tenemos un benchmark.”	Da una cifra externa.	No sustituye tus casos, tus fuentes, tus riesgos ni tu producto.

El cierre del facsímil existe para practicar justo lo contrario: no quedarse en sensación, sino producir evidencia.[3]

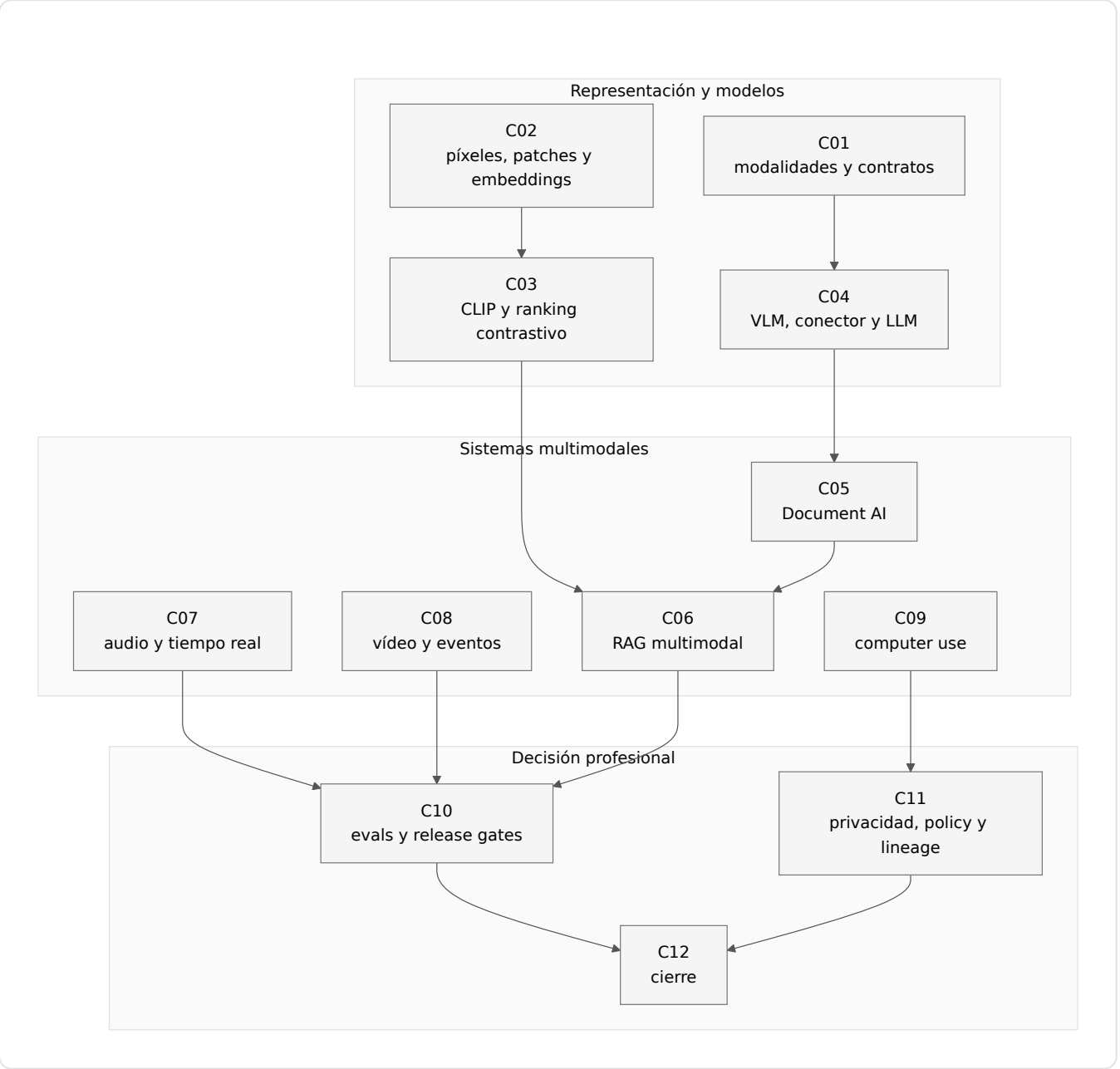
Un gate de release multimodal

Una decisión de publicación no puede depender solo de la calidad. Un score orientativo de preparación combina varios factores, que el cuaderno pondera y calibra con evaluación:

SÍMBOLO	LECTURA
G	Score orientativo de preparación para release.
Q	Calidad: groundedness, extracción, temporalidad y utilidad.
E	Evidencia: contratos, golden set, manifest, traces, lineage.
O	Operación: latencia, coste, tasa de fallo y degradación.
M	Mitigación: controles presentes frente a riesgos activos.
R	Riesgo residual: PII, secreto, contenido no confiable, acción externa.

No lo uses como una verdad universal. Úsalo como recordatorio: una decisión de publicación no puede depender solo de calidad. Un sistema puede responder muy bien y aun así no poder publicarse porque conserva PII, ejecuta una acción externa sin aprobación o no deja evidencia.

Cómo encaja todo



Este mapa importa porque el cierre mezcla todo. Un caso de helpdesk multimodal no es “un capítulo 07 con voz”. También toca Document AI, RAG, evaluación, privacidad y operación. La vida real hace eso: mezcla capítulos.

Dónde solía tropezar yo

TROPIEZO	POR QUÉ ES UN PROBLEMA	ANTÍDOTO
Celebrar el caso que pasa y olvidar el que bloquea	Un release puede fallar por una única ruta crítica.	Decisión global: si hay `block`, el release completo no pasa.
Confundir remediación con maquillaje	Añadir una frase al informe no arregla un control ausente.	Cambiar datos, evidencias, policy, tests o arquitectura.
No conectar con capítulos anteriores	La práctica se vuelve un ejercicio aislado.	Matriz de trazabilidad capítulo → caso → evidencia.
Publicar sin decision card	Nadie sabe qué quedó pendiente.	Decisión final con condiciones y criterio de salida.
No dejar nada descargable	El alumno no puede repetir ni adaptar la práctica.	ZIP con datos, runner, tests, salidas y solución.

Cuadernos para practicar

Has operado un sistema multimodal a lo largo del facsímil; estos cuadernos te dejan abrir las dos piezas que lo sostienen: cómo entra una imagen en el modelo y cómo se busca por contenido. Son *notebooks* que se abren en Google Colab —gratis, en el navegador— o te puedes descargar. Cada uno, explicado paso a paso, con salidas reales, y anclado al capítulo del que sale.

De píxeles a patches: cómo una imagen entra en un Transformer

Qué practicas: trocear una imagen en parches y comprobar que, para el modelo, es una secuencia de vectores. **Dónde encaja:** capítulo 2 (de píxeles a patches: cómo una imagen se convierte en representación). **Qué necesitas:** un navegador. Corre en CPU; sin claves.

Ves el truco que permitió a los Transformers —nacidos para texto— entender imágenes. Una foto de 64×64 no entra píxel a píxel: se trocea en una rejilla de **16 parches** de 16×16, y cada parche se aplana en un vector de **768 números**. La imagen deja de ser una cuadrícula de píxeles y pasa a ser una secuencia de 16 vectores, igual que una frase es una secuencia de *tokens*. A partir de ahí se le aplica la misma atención del facsímil 3: cada parche mira a los demás. Por eso un Transformer de texto, casi sin tocarlo, sirve para imágenes: solo cambió la puerta de entrada.

«Imagen igual a secuencia de parches» es la llave de los *Vision Transformers* y, sobre ellos, de CLIP y los modelos visión-lenguaje.

► Abrir en Google Colab **↓ Descargar el cuaderno (.ipynb)**

Buscar imágenes por su contenido: el germen de la búsqueda multimodal

Qué practicas: convertir imágenes en vectores y buscar las más parecidas a una consulta. **Dónde encaja:** capítulos 3 y 4 (CLIP y aprendizaje contrastivo; modelos visión-lenguaje). **Qué necesitas:** un navegador. Corre en CPU; sin claves.

Montas un buscador que no mira nombres de archivo ni etiquetas, sino el contenido. Conviertes cada imagen de una pequeña colección en un vector descriptor (un resumen de sus colores) y, ante una consulta —un atardecer naranja que no está en la colección—, buscas las más parecidas midiendo la similitud entre vectores. El buscador no sabe nada de «atardeceres»: aun así, coloca arriba el atardecer (**0,97** de similitud) y manda al fondo el azul del mar y los verdes. Cambia «color» por «un vector aprendido que entiende escenas» y tienes CLIP: la misma mecánica, descriptores mucho más inteligentes.

La búsqueda multimodal es «todo a vectores y comparar distancias»: la misma idea de los *embeddings* de texto, ahora poniendo imágenes y palabras en el mismo espacio.

► Abrir en Google Colab **↓ Descargar el cuaderno (.ipynb)**

Vocabulario aprendido

TÉRMINO	DEFINICIÓN
Release multimodal	Decisión de publicar, revisar o bloquear un sistema con varias modalidades.
Evidence pack	Paquete de informes y matrices que justifica una decisión.
Remediación	Cambio concreto para cerrar una evidencia, control o métrica.
Traceability matrix	Matriz que conecta capítulos, casos y decisiones.
Release gate	Regla de publicación basada en calidad, riesgo, operación y evidencias.
Baseline	Estado inicial contra el que comparas una mejora.
Decision card	Resumen de decisión, condiciones y pendientes.
Release candidate	Versión candidata a publicarse porque ya pasó gates, pero aún debe revisarse con checklist, owner y rollback.
SLI	Indicador medido: latencia, coste, tasa de fallo, calidad o cobertura de evidencias.
SLO	Objetivo que debe cumplir un SLI para considerar sano el sistema.
Contract test	Prueba que valida que un caso tiene campos, evidencias y controles mínimos antes de evaluarlo con un modelo.
Change request	Documento de cambio: qué se modifica, qué mejora, quién aprueba y cómo se revierte.
Rollback	Plan para volver atrás si el candidato empeora una métrica, pierde evidencia o incumple policy.

Antes de pasar página

Hazte estas preguntas:

1. ¿Sé explicar qué aporta cada modalidad?
2. ¿Cada caso tiene evidencias descargables?
3. ¿Puedo separar calidad de riesgo?
4. ¿Sé qué caso bloquea release y por qué?

5. ¿La remediación cambia datos, controles o tests, no solo texto?
6. ¿La decisión usa capítulos concretos del facsímil?
7. ¿La práctica deja una decision card?
8. ¿El cuaderno se puede ejecutar sin depender de APIs externas?
9. ¿Puedo adaptar el patrón a un proyecto real?
10. ¿Sé transformar un caso en revisión en release candidate?
11. ¿Sé qué SLI/SLO bloquearía el release?
12. ¿Sé qué cambio de modelo, OCR, ASR, retrieval, prompt, tool o policy obliga a repetir evals?

Si respondes sí, el facsímil ha cumplido su función: darte un criterio técnico para construir, evaluar y operar sistemas que perciben.

En resumen

IDEA	QUÉ DEBERÍAS LLEVARTE
La multimodalidad es ingeniería de evidencias.	No basta con que el modelo “vea”. Hay que conservar fuente, región, página, timestamp o traza.
Cada modalidad cambia el contrato.	Imagen, PDF, audio, vídeo y pantalla fallan de formas distintas.
RAG multimodal exige trazabilidad.	Recuperar no es copiar contexto: es construir evidencia con permisos.
Computer use exige permisos externos al modelo.	El agente puede pedir, pero la policy decide.
Evaluar es decidir.	Las métricas sirven si bloquean, revisan o permiten una acción.
Privacidad y seguridad viven en la arquitectura.	Redacción, egress, retention, lineage y runbook no son apéndices.
El cierre junta todo.	El resultado profesional es una decisión de release defendible.
Un candidato exige ingeniería.	No basta con subir una métrica: necesitas diff, contract tests, SLI/SLO, owner, aprobadores, rollback y versionado.

Recursos para seguir: leer, construir y experimentar

Este facsículo se apoya en muchos artículos y mucha documentación, y eso puede dar la sensación de que la multimodalidad es algo solo para leer. No lo es. Aquí tienes esos recursos reorganizados por lo que quieras hacer con ellos, porque ser «gente curiosa» significa, sobre todo, tocar las cosas.

Para experimentar sin escribir apenas código. La forma más rápida de que todo lo anterior deje de ser abstracto es abrir un modelo multimodal y darle una imagen, un audio o una captura tuyas. Los entornos de prueba oficiales permiten justo eso desde el navegador: el playground de OpenAI (`platform.openai.com`), Google AI Studio (`aistudio.google.com`) y la consola de Anthropic (`console.anthropic.com`). Para ver y comparar modelos abiertos, conjuntos de datos y demos de la comunidad, Hugging Face (`huggingface.co`) es el sitio donde casi todo lo que cita este facsículo tiene una página con la que se puede jugar. Y si quieres explorar datos reales, los mismos que usan los benchmarks: LibriSpeech y Common Voice para voz (`openslr.org` , `commonvoice.mozilla.org`), o Kinetics, ActivityNet y Ego4D para vídeo, todos con web propia. Sube una de tus imágenes con texto pequeño, una nota de voz con ruido de fondo o una captura de pantalla, y comprueba en directo eso que el capítulo correspondiente te ha avisado que iba a fallar: la resolución que no llega, el slot crítico que se confunde, la alucinación sobre el silencio.

Para construir. Cuando quieras pasar de jugar a montar algo, las piezas reales por tema están repartidas por el facsículo: para visión y recuperación, encoders y modelos tipo CLIP o SigLIP y un índice vectorial (Qdrant); para documentos, modelos como LayoutLM o ColPali; para voz, las APIs en tiempo real de OpenAI y Gemini, Whisper para transcribir y pyannote para diarización; para vídeo, ffmpeg, OpenCV y PyAV como base de ingesta; para agentes de pantalla, Playwright y las herramientas de computer use de OpenAI y Anthropic; para evaluar, OpenAI Evals, Promptfoo y Ragas; y para privacidad y seguridad, Presidio y un motor de políticas como OPA o Cedar. Ninguna es «la solución»: cada una es una pieza con su sitio en la arquitectura, y el facsículo te ha dado el criterio para colocarla.

Para leer. Cada capítulo cierra con su «Para saber más». Si tuvieras que quedarte con una ruta mínima: ViT y CLIP para entender cómo una imagen se vuelve representación, Whisper para el audio, LayoutLM y ColPali para documentos, y el OWASP LLM Top 10 junto al AI RMF del NIST para seguridad y gobernanza.

Del concepto a la herramienta real. Cada idea del facsímil se trabaja primero de forma deliberadamente determinista, para que entiendas la capa de decisión sin el ruido de un modelo real; cuando la domines, cada una tiene su herramienta de verdad al lado, y el salto deja de dar vértigo:

LO QUE PRACTICAS	LA HERRAMIENTA REAL A LA QUE TE GRADÚAS
C2 · tokenizar una imagen en patches	processors y encoders de visión en Hugging Face
C3 · ranking contrastivo y margen	CLIP/SigLIP en Hugging Face + un índice vectorial (Qdrant)
C5 · campos, tablas y evidencia en documentos	LayoutLM, ColPali o el Document AI de un proveedor
C6 · RAG multimodal con citas	índice vectorial + Ragas para medir faithfulness
C7 · contrato de turno de voz	OpenAI Realtime o Gemini Live, Whisper y pyannote
C8 · eventos y orden en vídeo	ffmpeg/OpenCV/PyAV + un modelo de vídeo
C9 · arnés con policy gate	Playwright y el computer use de OpenAI o Anthropic
C10 · score, slices y gate	OpenAI Evals, Promptfoo o Ragas
C11 · redacción, egress y lineage	Presidio y un motor de políticas (OPA, Cedar)

La idea no es que memorices esta tabla, sino que sepas que ninguno de estos kits es un callejón sin salida: cada uno te deja a un paso de la herramienta que usarías en un trabajo real, con la diferencia de que ahora entiendes qué pregunta le estás haciendo.

Para saber más

- Baltrušaitis, T., Ahuja, C. y Morency, L. P. (2019). *Multimodal Machine Learning: A Survey and Taxonomy*. IEEE TPAMI.
- Dosovitskiy, A. y otros (2021). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. ICLR.
- Radford, A. y otros (2021). *Learning Transferable Visual Models From Natural Language Supervision*. ICML.
- Xu, Y. y otros (2020). *LayoutLM: Pre-training of Text and Layout for Document Image Understanding*. KDD.
- Faysse, M. y otros (2024). *ColPali: Efficient Document Retrieval with Vision Language Models*. arXiv.
- Radford, A. y otros (2023). *Robust Speech Recognition via Large-Scale Weak Supervision*. ICML.
- OpenAI. (2026). *Working with Evals*. https://developers.openai.com/api/docs/guides/evals
- OWASP Foundation. (2025). *OWASP Top 10 for LLM and Generative AI Applications 2025*. https://genai.owasp.org/resource/owasp-top-10-for-llm-applications-2025/

- NIST. (2024). *AI RMF Generative AI Profile*. https://www.nist.gov/publications/artificial-intelligence-risk-management-framework-generative-artificial-intelligence

---

Notas

1. La idea de tratar la multimodalidad como integración de señales, representación, alineamiento, traducción, fusión y coaprendizaje está muy bien ordenada en Baltrušaitis, Ahuja y Morency (2019). En este capítulo la bajo a una decisión de ingeniería: qué evidencia conserva el sistema cuando mezcla modalidades. ↩

2. Dosovitskiy y otros (2021) popularizan el tratamiento de la imagen como secuencia de patches en transformers; Radford y otros (2021) muestran cómo alinear texto e imagen mediante entrenamiento contrastivo a gran escala. No son recetas mágicas de producto, pero sí dos piezas conceptuales para no tratar un VLM como una caja negra. ↩

3. LayoutLM (Xu y otros, 2020) y ColPali (Faysse y otros, 2024) son buenos recordatorios de que documento no significa solo texto: la página tiene coordenadas, estructura visual y señales que pueden cambiar la respuesta. En evaluación y riesgo, uso como marco práctico la idea de evidencias, slices y controles que también aparece en guías de Evals, NIST AI RMF y OWASP Top 10 para aplicaciones LLM. ↩