

Notas

Este curso presenta SPAD (Structured Prompt-Driven Engineering, ingeniería estructurada guiada por instrucciones), la metodología de Fernando García Varela para construir software con ayuda de la inteligencia artificial de forma controlada. No es una colección de instrucciones para pedir código: es una forma de trabajar.

El curso sirve para una sesión completa de aula y también como material de difusión. Tiene diez módulos (0 a 9) y un cierre con casos para discutir. Cada módulo abre con su objetivo y cierra con un ejercicio o una autocomprobación. Las notas de cada slide son el guion del profesor.

No hace falta saber programar. Para un directivo, la pregunta clave es sencilla: si su compañía ya construye software con IA, ¿quién toma las decisiones de diseño, quién revisa y dónde queda la evidencia?

- SPAD no promete cifras de mejora: cada organización mide con sus propios datos.
- SPAD no certifica: la conformidad se comprueba con una autoevaluación interna.
- Los ejemplos del curso son ficticios o ilustrativos.

SPAD se publica con licencia CC BY 4.0 para los contenidos y MIT para el código. Hay que citar la autoría: SPAD · Fernando García Varela.

Pregunta para abrir la sesión: ¿cuántos de ustedes usan ya la IA para escribir código, informes o fórmulas en su trabajo?

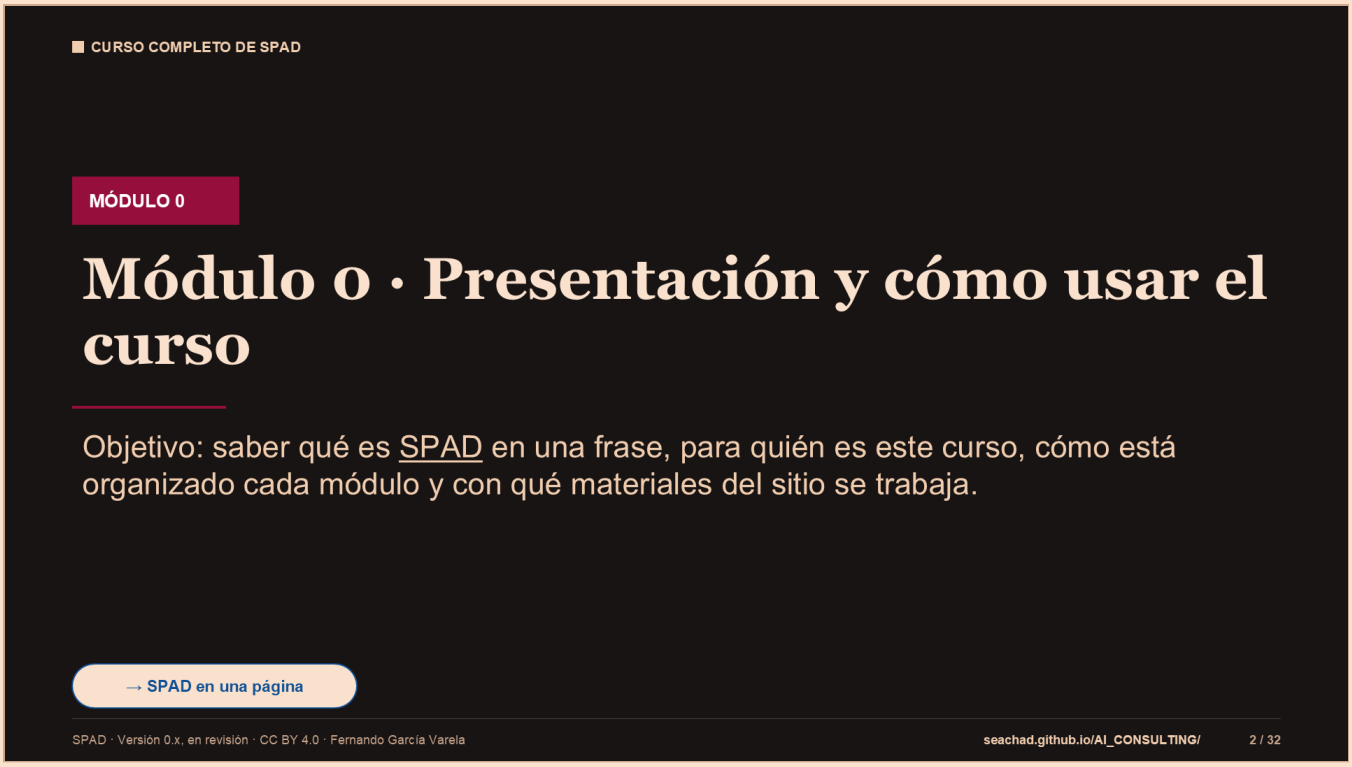
TÉRMINOS DE ESTA SLIDE

- **Principio de ejecución bloqueante**: Regla central de SPAD: ninguna fase empieza hasta que la anterior ha entregado su artefacto y ha recibido un veredicto favorable. Evita que la IA construya sobre un plan que nadie ha revisado. El enlace abre el principio en la guía operativa. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#el-principio-de-ejecucion-bloqueante
- **SPAD**: SPAD (Structured Prompt-Driven Engineering) es una metodología para construir software con ayuda de la IA de forma ordenada: fases secuenciales, cada una con su artefacto, revisiones con veredicto y personas que

validan. Es secuencial, bloqueante y auditable. La usan equipos de desarrollo. El enlace abre su documento de presentación. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html

FUENTES Y ENLACES

- **SPAD 00 · Estado, licencia y próximos pasos** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#estado-licencia-y-proximos-pasos
- **Biblioteca SPAD** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/index.html



Notas

Objetivo del módulo 0: que el alumno sepa qué va a aprender y cómo usar el material.

SPAD responde a una pregunta muy concreta: cómo construir software con IA sin perder el control. La IA escribe código plausible a gran velocidad. Sin estructura, esa velocidad produce decisiones de diseño que nadie ha tomado de forma consciente, pruebas insuficientes y vulnerabilidades que entran en silencio.

SPAD tiene tres propiedades que veremos en el módulo 2: es secuencial (las fases van en un orden fijo), bloqueante (ninguna fase empieza sin revisar la anterior) y auditable (cada fase deja un documento guardado).

Por qué importa a un directivo: no necesita saber cómo se revisa un plan de arquitectura, pero sí necesita saber que existe un método que lo hace y que deja evidencias. Eso es lo que pedirán el consejo, el cliente o el auditor.

Sugerencia para el profesor: dedique pocos minutos a este módulo y pase pronto al problema (módulo 1), que es lo que más interesa a un público no técnico.

TÉRMINOS DE ESTA SLIDE

- **SPAD**: SPAD (Structured Prompt-Driven Engineering) es una metodología para construir software con ayuda de la IA de forma ordenada: fases secuenciales, cada una con su artefacto, revisiones con veredicto y personas que validan. Es secuencial, bloqueante y auditable. La usan equipos de desarrollo. El enlace abre su documento de presentación. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html

FUENTES Y ENLACES

- **SPAD 00 · Qué es SPAD** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#que-es-spad
- **SPAD 00 · Cuándo usar SPAD** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#cuando-usar-spad
- **SPAD en una página** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#que-es-spad



Notas

Esta slide explica cómo está hecho el curso y propone el primer ejercicio.

SPAD en una frase: cada fase deja un documento (en SPAD se llama artefacto), ninguna fase empieza sin revisar la anterior y ninguna IA se aprueba a sí misma. Siempre valida una persona.

Cada módulo sigue la misma estructura: objetivo, conceptos con un ejemplo sencillo y un ejercicio o autocomprobación. Las respuestas de los ejercicios están en las notas.

Materiales: la biblioteca SPAD tiene diez documentos. El 00 presenta el método; el 01 es la guía operativa con las fases, los roles y los veredictos; el 02 explica contextos, temas y registros; el 03 es la política de validación; el 04, los ciclos complementarios; el 05, los sistemas que incluyen IA; el 06 y el 07, las instrucciones y los contratos de cada fase; el 08, la autoevaluación y las métricas. El documento 00 explica también para qué le sirve SPAD a una compañía: construir con método y exigir a sus proveedores (módulo 9). Las láminas están en español.

Ejercicio de partida (cinco minutos, por parejas): piense en un sistema de su compañía hecho con ayuda de IA, aunque sea una hoja de cálculo con fórmulas generadas o un programa pequeño. Responda: ¿quién diseñó la solución?, ¿quién revisó el resultado?, ¿dónde está la evidencia de esa revisión? Guarde las respuestas: las retomaremos en el cierre.

Por qué importa: muchos participantes descubrirán que no saben responder a la tercera pregunta. Ese es el hueco que cubre SPAD.

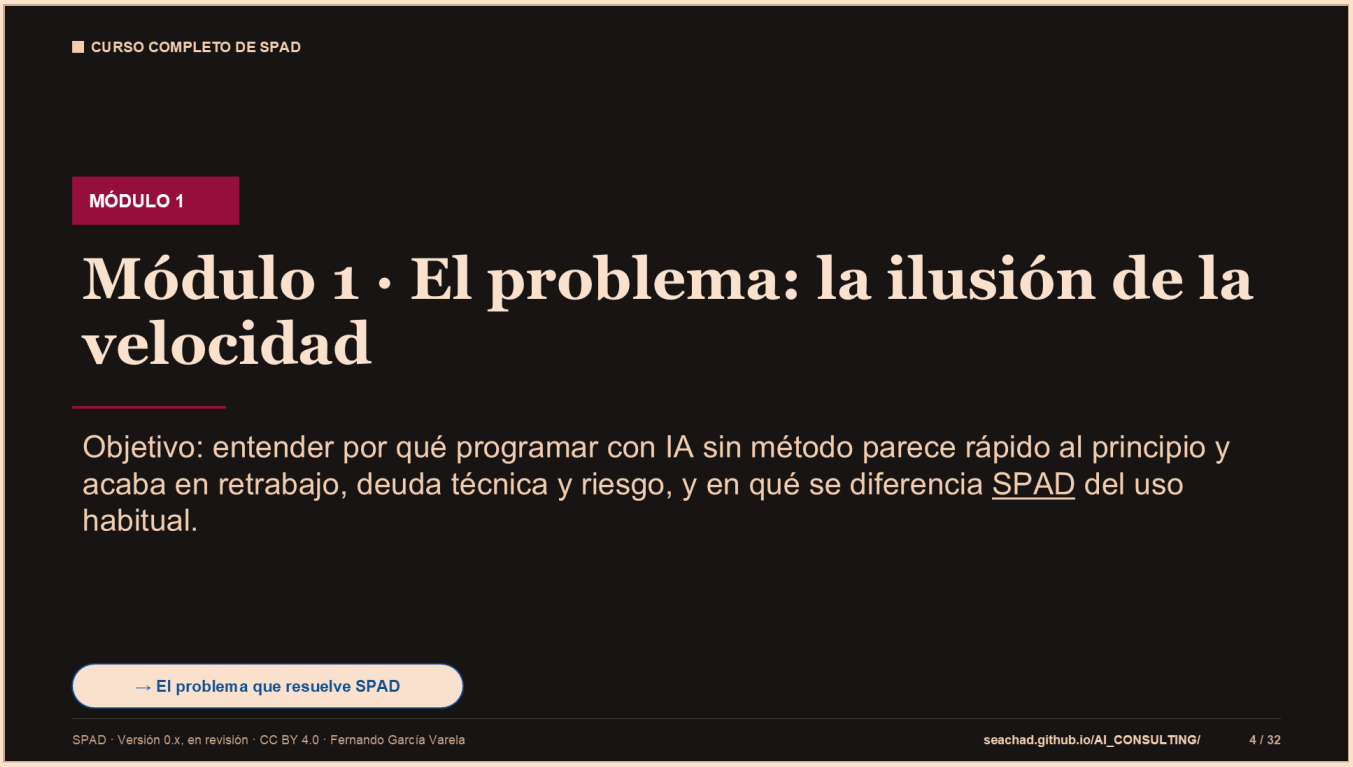
TÉRMINOS DE ESTA SLIDE

- **SPAD**: SPAD (Structured Prompt-Driven Engineering) es una metodología para construir software con ayuda de la IA de forma ordenada: fases secuenciales, cada una con su artefacto, revisiones con veredicto y personas que validan. Es secuencial, bloqueante y auditable. La usan equipos de desarrollo. El enlace abre su documento de presentación. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html

- **Métricas del proceso**: Indicadores que mide un equipo que usa SPAD, como cuántas revisiones terminan en NO-GO, cuántos defectos llegan a producción o cuánto tarda un tema. SPAD no promete cifras: propone medirlas y fijar objetivos propios. El enlace abre las métricas. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/08_SPAD_Autoevaluacion_y_metricas.html#metricas-del-proceso

FUENTES Y ENLACES

- **SPAD 00 · Estado, licencia y próximos pasos (mapa de la biblioteca)** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#estado-licencia-y-proximos-pasos
- **SPAD 00 · Para qué le sirve a una compañía** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#para-que-le-sirve-a-una-compania
- **Biblioteca SPAD** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/index.html
- **Para qué le sirve a una compañía** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#para-que-le-sirve-a-una-compania



Notas

Objetivo del módulo 1: entender el problema antes de la solución.

La IA ha cambiado la economía de escribir código: lo que antes llevaba días ahora puede llevar minutos. Eso es real y valioso. El problema es otro: la velocidad de escribir no es la velocidad de entregar un sistema que funcione en producción, que se pueda mantener y que alguien pueda explicar.

El documento 00 de SPAD resume los síntomas típicos:

- El código funciona en la demostración y falla en producción, porque nadie diseñó la solución.
- Nadie sabe por qué el sistema está construido así, porque las decisiones las tomó la IA dentro del código.
- Las pruebas son escasas, porque se escriben al final, si sobra tiempo.
- Aparecen vulnerabilidades, porque revisó la misma IA que escribió o no revisó nadie.
- Una auditoría pide evidencias y no existen, porque todo quedó en conversaciones con la IA.

Idea clave para el aula: ninguno de estos problemas es de la IA. Son problemas de método que la IA amplifica.

Pregunta para el aula: ¿alguien ha vivido un proyecto que iba muy rápido al principio y se atascó al final? ¿Qué faltaba?

TÉRMINOS DE ESTA SLIDE

- **SPAD**: SPAD (Structured Prompt-Driven Engineering) es una metodología para construir software con ayuda de la IA de forma ordenada: fases secuenciales, cada una con su artefacto, revisiones con veredicto y personas que validan. Es secuencial, bloqueante y auditable. La usan equipos de desarrollo. El enlace abre su documento de presentación. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html

FUENTES Y ENLACES

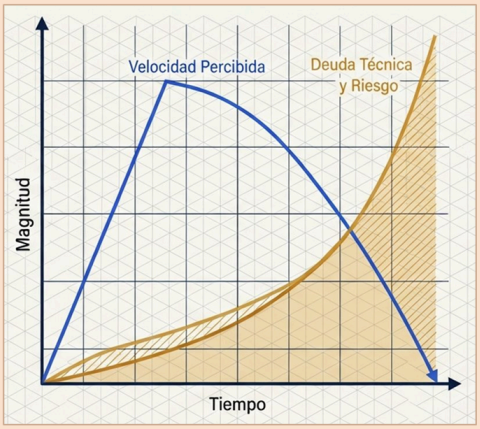
CURSO COMPLETO DE SPAD

- **SPAD 00 · El problema que resuelve** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#el-problema-que-resuelve
- **El problema que resuelve SPAD** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#el-problema-que-resuelve

CURSO COMPLETO DE SPAD

La ilusión de la velocidad

Gráfica conceptual: la velocidad percibida sube y después cae; la deuda técnica y el riesgo crecen.



- **Lo que se ve** Al principio, la IA genera código plausible muy deprisa. La sensación de avance es enorme.
- **Lo que no se ve** Decisiones de diseño que nadie tomó, pruebas que faltan, vulnerabilidades y deuda técnica que crecen en silencio.
- **Por qué importa** No es un fallo de la IA: es un problema de método que la IA amplifica. Se paga después, en incidentes y retrabajo.
- **Lámina** Lámina de SPAD en español. Es conceptual: no contiene datos medidos.

→ El problema que resuelve SPAD

SPAD · Versión 0.x, en revisión · CC BY 4.0 · Fernando García Varela seachad.github.io/AI_CONSULTING/ 5 / 32

Notas

La lámina es una gráfica conceptual, sin datos medidos. Muestra dos curvas a lo largo del tiempo.

La línea azul es la velocidad percibida. Sube muy deprisa al principio: la IA genera pantallas, funciones y pruebas aparentes en minutos. Después cae, porque cada cambio nuevo choca con decisiones que nadie documentó y hay que rehacer trabajo.

La curva naranja es la deuda técnica y el riesgo. Crece despacio al principio y se dispara después. La deuda técnica es el coste futuro de los atajos de hoy: código difícil de cambiar, sin pruebas o sin explicación.

Ejemplo ficticio: un equipo construye con IA un portal de pedidos en dos semanas. En la demostración todo funciona. Al tercer mes, cada cambio rompe otra cosa, nadie sabe por qué una tabla se llama como se llama y una revisión de seguridad encuentra una contraseña escrita en el código.

Por qué importa para decidir: el coste no aparece en el presupuesto del proyecto, sino después, en incidentes, retrabajo y pérdida de confianza. SPAD mantiene la velocidad donde aporta y pone control en las decisiones, las revisiones y las evidencias.

Advertencia honesta: SPAD no afirma ninguna cifra de mejora. Al principio hace el trabajo más lento mientras el equipo aprende las fases.

FUENTES Y ENLACES

- **SPAD 00 · El problema que resuelve** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#el-problema-que-resuelve
- **SPAD 00 · Cuándo usar SPAD (sin cifras de mejora)** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#cuando-usar-spad
- **El problema que resuelve SPAD** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#el-problema-que-resuelve



Notas

La lámina compara seis aspectos del uso de la IA sin método con SPAD.

- Planificación: sin método es opcional e informal; en SPAD es obligatoria y explícita, antes de escribir código.
- Revisión del diseño: sin método es irregular; en SPAD la hace una IA distinta de la que diseñó y la valida una persona.
- Pruebas: sin método se omiten a menudo; en SPAD la cobertura mínima se fija antes de construir.
- Seguridad: sin método es reactiva; en SPAD el código sensible pasa una revisión que puede bloquear el despliegue.
- Trazabilidad: sin método es pobre; en SPAD cada fase deja un artefacto con nombre, tema y versión.
- Ejecución: sin método hay atajos constantes; en SPAD las fases son bloqueantes.

Aclaración: la lámina habla de revisión «automatizada» y de «Gates». En SPAD la revisión técnica la hace una IA revisora, pero el veredicto lo acepta o rechaza siempre una persona. Los «Gates» de la lámina son esos puntos de bloqueo entre fases, no decisiones de inversión de la compañía.

Respuestas orientativas: si el diseño no es obligatorio, si revisa la misma IA o si no puede enseñar la evidencia, su equipo está en la columna de la izquierda en ese aspecto. No es un juicio: es el punto de partida.

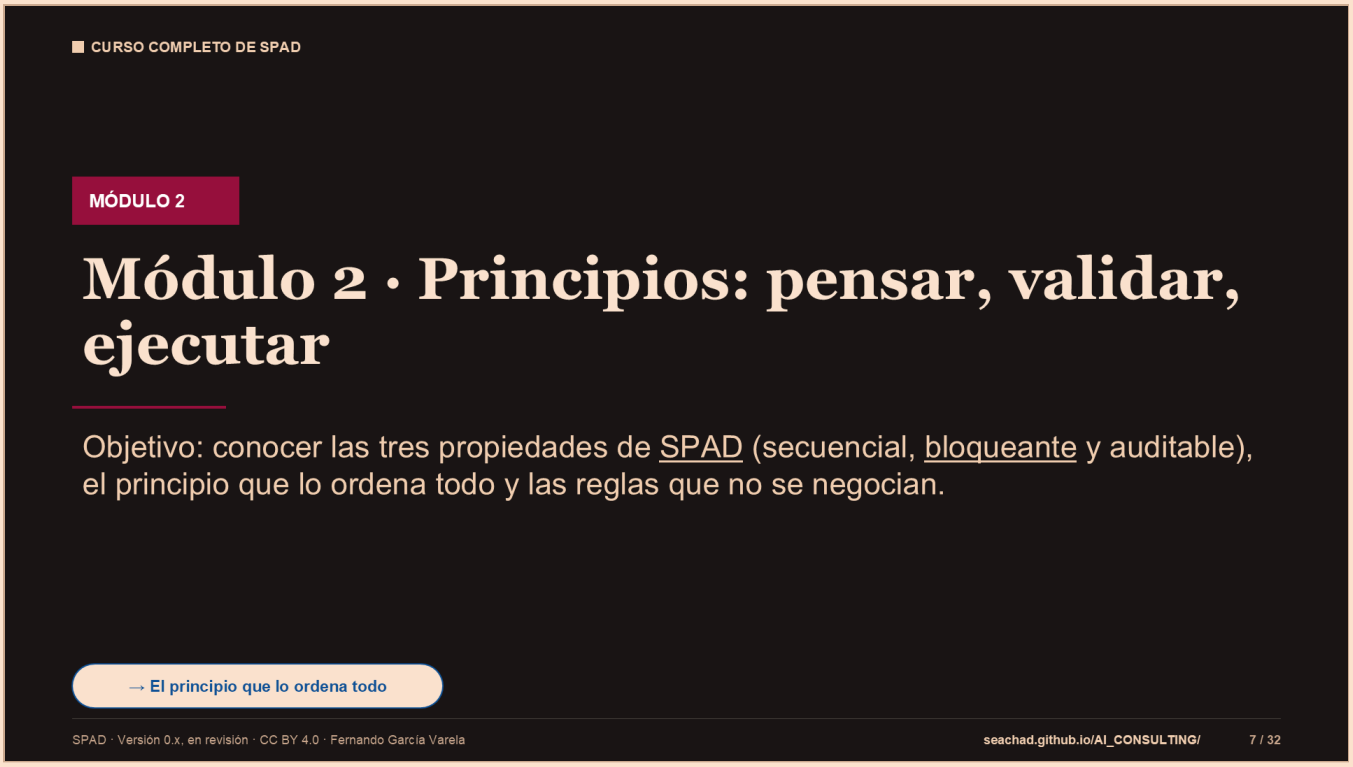
Por qué importa: con SPAD, la calidad no depende de la destreza de quien escribe las instrucciones ni del modelo del momento, sino de un proceso que cualquier equipo puede repetir.

TÉRMINOS DE ESTA SLIDE

- **SPAD**: SPAD (Structured Prompt-Driven Engineering) es una metodología para construir software con ayuda de la IA de forma ordenada: fases secuenciales, cada una con su artefacto, revisiones con veredicto y personas que validan. Es secuencial, bloqueante y auditable. La usan equipos de desarrollo. El enlace abre su documento de presentación. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html

FUENTES Y ENLACES

- **SPAD 00 · El problema que resuelve** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#el-problema-que-resuelve
- **SPAD 01 · Veredictos** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#veredictos
- **Síntomas y respuesta de SPAD** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#el-problema-que-resuelve



Notas

Objetivo del módulo 2: conocer las ideas que sostienen todo el método.

SPAD convierte la IA de un asistente informal de programación en una herramienta de ingeniería gobernada, cuyo trabajo se puede revisar, reproducir y auditar.

Las tres propiedades:

- Secuencial: el trabajo avanza por fases en un orden fijo.
- Bloqueante: ninguna fase empieza hasta que la salida de la anterior se ha producido y evaluado.
- Auditable: cada fase produce un artefacto explícito que se guarda.

El principio que lo ordena todo: el cumplimiento del flujo es más importante que la calidad aparente del resultado. La prioridad es, por este orden: cumplir el proceso, la corrección técnica y la eficiencia.

Ejemplo para el aula: un alumno entrega un examen perfecto, pero copiado. El resultado parece bueno, pero no vale, porque no se cumplió el proceso. En SPAD pasa lo mismo con una respuesta de la IA que se salta su fase: se descarta entera y la fase se repite.

Por qué importa: si una respuesta que rompe las reglas se acepta «porque el código funciona», el proceso deja de existir a la segunda ocasión.

TÉRMINOS DE ESTA SLIDE

- **SPAD**: SPAD (Structured Prompt-Driven Engineering) es una metodología para construir software con ayuda de la IA de forma ordenada: fases secuenciales, cada una con su artefacto, revisiones con veredicto y personas que validan. Es secuencial, bloqueante y auditable. La usan equipos de desarrollo. El enlace abre su documento de presentación. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html
- **Principio de ejecución bloqueante**: Regla central de SPAD: ninguna fase empieza hasta que la anterior ha entregado su artefacto y ha recibido un veredicto favorable. Evita que la IA construya sobre un plan que nadie ha

revisado. El enlace abre el principio en la guía operativa. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#el-principio-de-ejecucion-bloqueante

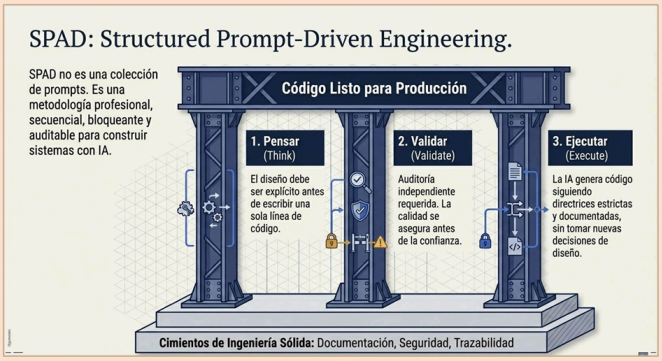
FUENTES Y ENLACES

- **SPAD 00 · Qué es SPAD** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#que-es-spad
- **SPAD 03 · Principio fundamental** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/03_SPAD_Politica_de_validacion.html#principio-fundamental
- **El principio que lo ordena todo** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#el-principio-que-lo-ordena-todo

CURSO COMPLETO DE SPAD

Pensar, validar, ejecutar

Tres pilares sobre unos cimientos de documentación, seguridad y trazabilidad.



- **Secuencial** Primero se entiende el problema; después se diseña, se revisa, se definen las pruebas, se construye, se revisa y se versiona.
- **Bloqueante** Por el principio de ejecución bloqueante, ninguna fase empieza hasta que una persona valida la anterior.
- **Auditable** Cada fase deja un documento explícito que es la entrada obligatoria de la siguiente y queda guardado.
- **Lámina** Lámina de SPAD en español: pensar, validar y ejecutar.

En SPAD, cumplir el flujo es más importante que la calidad aparente del resultado.

→ Qué es SPAD | → Principio de ejecución bloqueante

SPAD · Versión 0.x, en revisión · CC BY 4.0 · Fernando García Varela seachad.github.io/AI_CONSULTING/ 8 / 32

Notas

La lámina muestra tres pilares que sostienen el código listo para producción, sobre unos cimientos de documentación, seguridad y trazabilidad.

Pensar: el diseño debe ser explícito antes de escribir una sola línea de código. Lo hace la IA planificadora en la fase PLAN, y lo revisa otra IA.

Validar: una revisión independiente asegura la calidad antes de confiar en el resultado. La IA que revisa nunca es la que construye, y una persona acepta o rechaza su veredicto.

Ejecutar: la IA genera código siguiendo reglas estrictas y documentadas, sin tomar decisiones de diseño nuevas.

Las tres propiedades de SPAD:

- Secuencial: primero entender; después diseñar y revisar, definir las pruebas, construir, revisar y versionar.
- Bloqueante: no se empieza el código «mientras se termina el plan». En ese momento el plan dejaría de gobernar nada.
- Auditable: plan, informes de revisión, estrategia de pruebas y registro de versión quedan guardados.

Ejemplo: en una obra no se levanta una pared antes de tener los planos aprobados. SPAD aplica la misma lógica al software hecho con IA.

Pregunta para el aula: ¿en qué otros procesos de su compañía es normal que quien hace algo no se lo apruebe a sí mismo? Por ejemplo, en contabilidad, compras o riesgos.

TÉRMINOS DE ESTA SLIDE

- **Principio de ejecución bloqueante**: Regla central de SPAD: ninguna fase empieza hasta que la anterior ha entregado su artefacto y ha recibido un veredicto favorable. Evita que la IA construya sobre un plan que nadie ha

revisado. El enlace abre el principio en la guía operativa. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#el-principio-de-ejecucion-bloqueante

- **SPAD**: SPAD (Structured Prompt-Driven Engineering) es una metodología para construir software con ayuda de la IA de forma ordenada: fases secuenciales, cada una con su artefacto, revisiones con veredicto y personas que validan. Es secuencial, bloqueante y auditable. La usan equipos de desarrollo. El enlace abre su documento de presentación. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html

FUENTES Y ENLACES

- **SPAD 00 · Qué es SPAD** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#que-es-spad
- **Qué es SPAD** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#que-es-spad



Notas

La guía operativa de SPAD (documento 01, sección 7) fija doce reglas operativas que no se negocian. La slide muestra seis; las otras seis son: toda salida es auditable, correcciones mínimas (quirúrgicas, nunca refactorizaciones encubiertas), pruebas obligatorias con cobertura mínima fijada antes de construir, contextos cargados antes de cualquier fase, un tema por trabajo y modelo registrado en cada artefacto.

Cada regla protege algo concreto:

- Ninguna fase se salta: protege el bloqueo.
- Toda decisión es explícita: protege la explicabilidad. Si dentro de un año alguien pregunta por qué el sistema hace algo, la respuesta está en el plan, no en la memoria de una persona.
- El código ejecuta, no decide: protege el plan.
- Revisiones independientes: protegen la calidad. Nadie corrige bien su propio examen.
- Seguridad por diseño: protege los datos y las personas.
- Ninguna IA aprueba: protege la responsabilidad. Detrás de cada aceptación hay un nombre.

Ejercicio del módulo 2 (diez minutos, en grupos): elija una regla que hoy no se cumpla en su compañía y describa un fallo plausible que esa ausencia permitiría. Ejemplo de respuesta: «Ninguna IA aprueba no se cumple: el equipo acepta lo que el asistente genera. Riesgo: un cambio en el cálculo de comisiones entra en producción sin que nadie lo haya leído».

Por qué importa para un directivo: estas reglas son las preguntas que puede hacer en un comité sin saber programar.

TÉRMINOS DE ESTA SLIDE

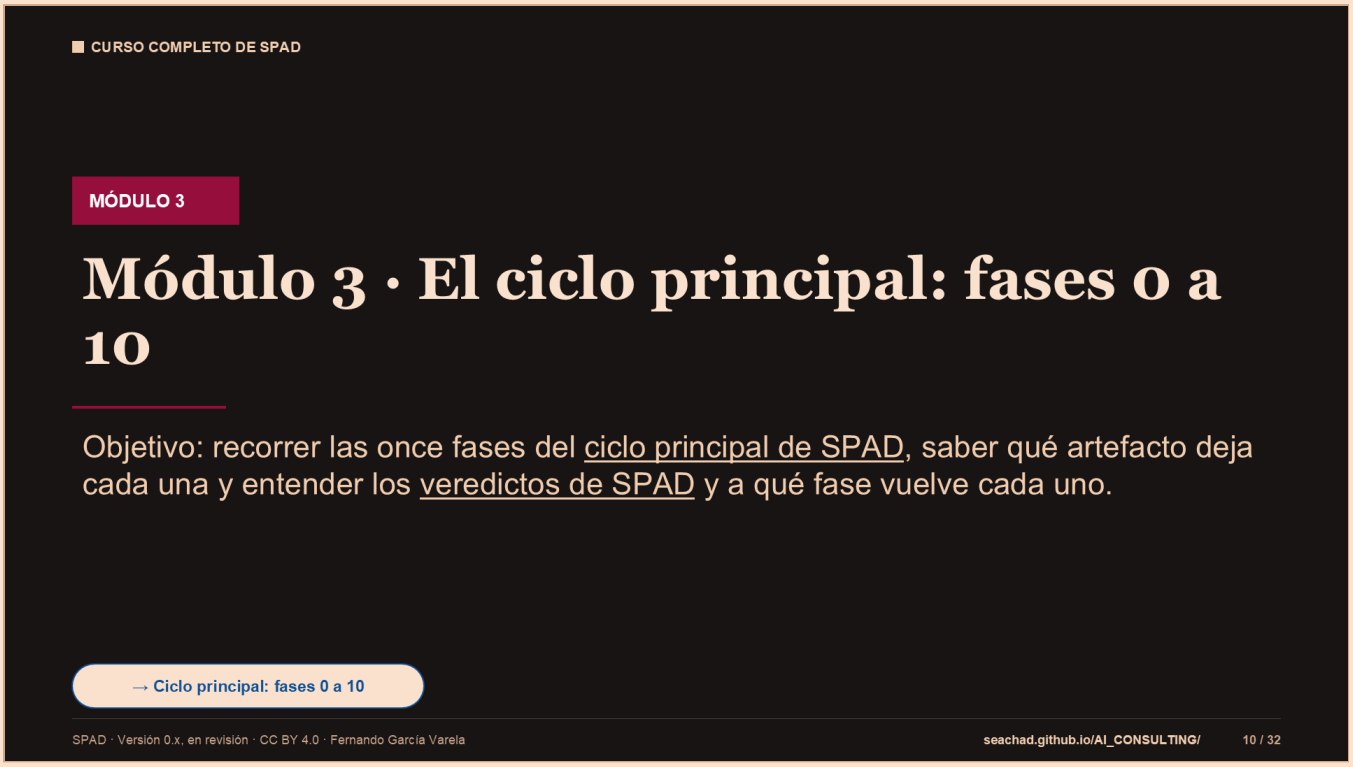
- **Roles de SPAD**: Los papeles que intervienen en SPAD: el orquestador (una persona que dirige el ciclo y valida) y varias IA con funciones separadas: planificadora, revisora, constructora, correctora, analista y de diagnóstico.

Quien construye no revisa lo que construye. El enlace abre el documento de presentación de SPAD, que describe cada rol. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html

- **Revisión humana del código**: En SPAD, el código que genera la IA siempre lo revisa una persona antes de integrarlo, y hay una segunda revisión en código sensible, agentes con autonomía y producción crítica. La IA ayuda a construir, pero no se aprueba a sí misma. El enlace abre la regla en la guía operativa. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#3-1-revision-humana-del-codigo
- **Política de validación**: Documento de SPAD que fija qué debe validar una persona, cuándo y con qué rigor, según el riesgo del cambio, y qué evidencia queda de esa validación. Es la garantía de que la IA no decide sola qué entra en el código. El enlace abre el documento. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/03_SPAD_Politica_de_validacion.html

FUENTES Y ENLACES

- **SPAD 01 · Reglas operativas no negociables** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#reglas-operativas-no-negociables
- **Las doce reglas operativas** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#reglas-operativas-no-negociables



Notas

Objetivo del módulo 3: conocer el recorrido completo de un trabajo en SPAD.

El ciclo principal tiene once fases, de la 0 a la 10. Se pueden agrupar en dos bloques:

- Pensar antes de construir (fases 0 a 4): contexto y objetivo, plan, revisión del plan, guía de código y estrategia de pruebas.
- Construir y revisar (fases 5 a 10): implementación, pruebas, revisión de las pruebas, revisión del código, correcciones y gestión de versiones.

Una idea que sorprende a un público de negocio: casi la mitad de las fases ocurren antes de escribir código. Es deliberado. Un error de diseño encontrado en la revisión del plan cuesta una conversación; el mismo error encontrado en producción cuesta un incidente.

Ejemplo que acompañará todo el módulo (ficticio): un equipo debe añadir a la web de una tienda el pago con una nueva pasarela. El tema de trabajo se llamará payment_gateway.

Pregunta para el aula: en sus proyectos, ¿qué porcentaje del tiempo se dedica a pensar antes de construir? ¿Y a rehacer después?

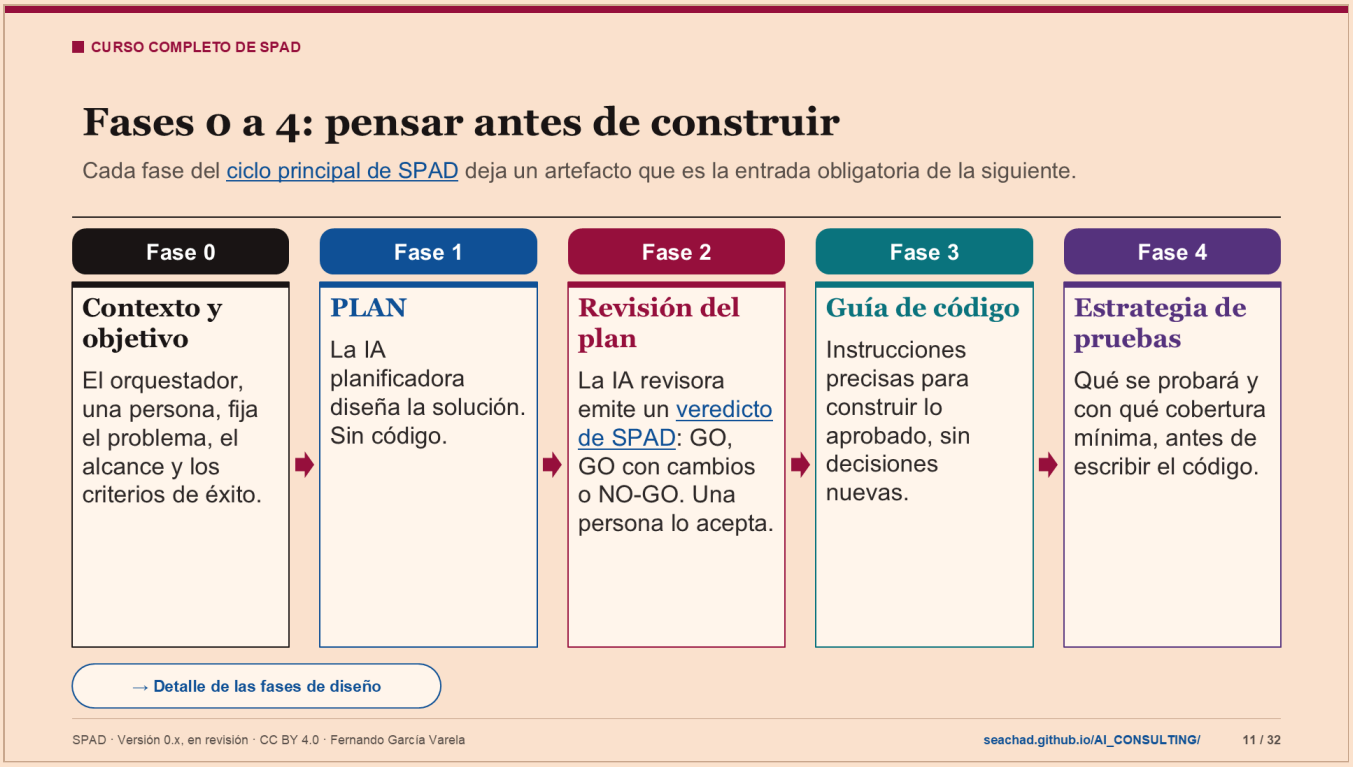
TÉRMINOS DE ESTA SLIDE

- **Ciclo principal de SPAD** (fases 0–10): Las once fases de SPAD: contexto, plan, revisión del plan, guía de código, estrategia de pruebas, implementación, pruebas, revisión de pruebas, revisión del código, correcciones y gestión de versiones. Cada una deja un artefacto que se revisa. El enlace abre el ciclo en la guía operativa. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#ciclo-principal-fases-0-a-10
- **Veredictos de SPAD** (GO / NO-GO): Toda revisión de SPAD termina con uno de tres veredictos: GO (se continúa), GO con cambios (se continúa tras corregir lo indicado) o NO-GO (vuelve a la fase que corresponda). Un solo vocabulario en todas las revisiones evita interpretaciones. El enlace abre los veredictos en la guía operativa. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#veredictos

FUENTES Y ENLACES

CURSO COMPLETO DE SPAD

- **SPAD 00 · El ciclo principal** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#el-ciclo-principal



Notas

Las fases 0 a 4 son el bloque de pensar.

- Fase 0 · Contexto y objetivo: la hace una persona, el orquestador. Define qué problema se resuelve, qué queda fuera y cómo se sabrá que el trabajo ha tenido éxito. En el ejemplo de payment_gateway: «el cliente puede pagar con la pasarela nueva; los pagos fallidos no generan pedidos; nada cambia en los métodos de pago existentes».
- Fase 1 · PLAN: la IA planificadora propone la arquitectura, los componentes, los riesgos y las decisiones de diseño. No escribe código.
- Fase 2 · Revisión del plan: una IA distinta revisa el plan y emite un veredicto justificado. Una persona lo acepta o lo rechaza. Si el plan no maneja el caso de una pasarela que no responde, aquí se detecta.
- Fase 3 · Guía de código: traduce el plan aprobado en instrucciones para construir. No añade decisiones.
- Fase 4 · Estrategia de pruebas: fija qué se probará y la cobertura mínima antes de construir. Así las pruebas no dependen de que sobre tiempo al final.

Por qué importa: cada fase produce un artefacto con nombre, número y tema. Si la fase 2 rechaza el plan, el plan rechazado no se borra: se guarda junto al nuevo. Eso permite explicar después por qué se cambió el diseño.

Pregunta para el aula: ¿quién debería ser el orquestador en un proyecto de su compañía: el jefe de producto, el responsable técnico o el dueño de negocio?

TÉRMINOS DE ESTA SLIDE

- **Ciclo principal de SPAD** (fases 0–10): Las once fases de SPAD: contexto, plan, revisión del plan, guía de código, estrategia de pruebas, implementación, pruebas, revisión de pruebas, revisión del código, correcciones y gestión de versiones. Cada una deja un artefacto que se revisa. El enlace abre el ciclo en la guía operativa. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#ciclo-principal-fases-0-a-10

- **Veredictos de SPAD** (GO / NO-GO): Toda revisión de SPAD termina con uno de tres veredictos: GO (se continúa), GO con cambios (se continúa tras corregir lo indicado) o NO-GO (vuelve a la fase que corresponda). Un solo vocabulario en todas las revisiones evita interpretaciones. El enlace abre los veredictos en la guía operativa. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#veredictos

FUENTES Y ENLACES

- **SPAD 01 · Detalle de las fases de diseño** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#4-1-detalle-de-las-fases-de-diseno
- **SPAD 06 · Instrucciones por fase** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/06_SPAD_Instrucciones_por_fase.html#como-usar-este-documento
- **Detalle de las fases de diseño** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#4-1-detalle-de-las-fases-de-diseno



Notas

Las fases 5 a 10 son el bloque de construir y revisar.

- Fase 5 · Implementación: la IA constructora escribe el código siguiendo la guía de la fase 3. Si encuentra algo que el plan no resuelve, no improvisa: lo señala.
- Fase 6 · Implementación de pruebas: se escriben las pruebas definidas en la fase 4.
- Fase 7 · Revisión de pruebas: una IA revisora comprueba que las pruebas cubren lo pactado y emite su veredicto.
- Fase 8 · Revisión del código: la IA revisora emite su veredicto técnico. Además, siempre revisa una persona y queda registrado. En código sensible (autenticación, autorización, pagos, datos personales, límites de agentes, decisiones sobre personas o producción crítica) hace falta una segunda revisión humana.
- Fase 9 · Correcciones: cambios mínimos y justificados para cada hallazgo. Después se vuelve a la fase 8. Una corrección nunca va directa a producción.
- Fase 10 · Gestión de versiones: se versiona el cambio y se entrega a operación con su registro.

En el ejemplo de payment_gateway, el pago toca dinero: el código pasa además el ciclo de seguridad (módulo 6) y la segunda revisión humana.

Por qué importa: el bucle entre las fases 8 y 9 evita la trampa habitual de «arreglo rápido sin revisar», que es donde más errores entran en producción.

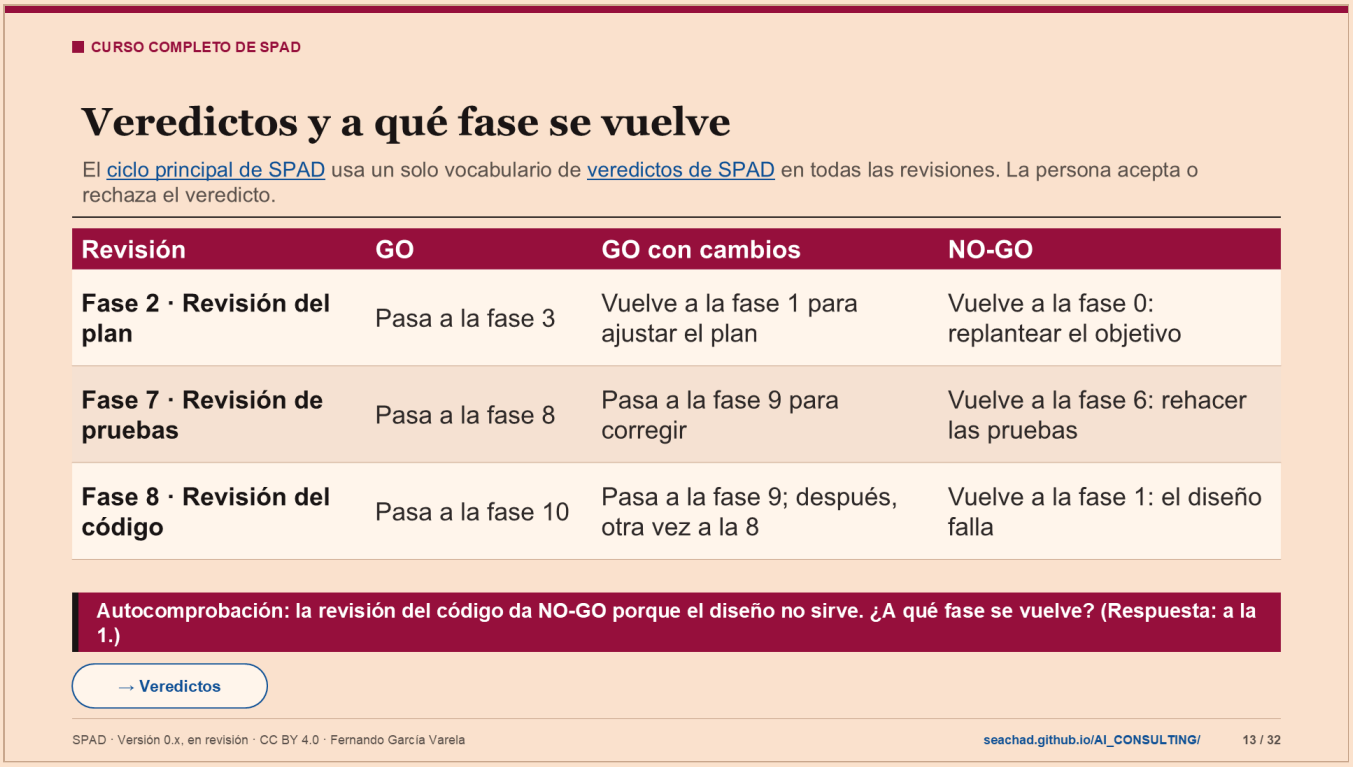
TÉRMINOS DE ESTA SLIDE

- **Ciclo principal de SPAD** (fases 0–10): Las once fases de SPAD: contexto, plan, revisión del plan, guía de código, estrategia de pruebas, implementación, pruebas, revisión de pruebas, revisión del código, correcciones y gestión de versiones. Cada una deja un artefacto que se revisa. El enlace abre el ciclo en la guía operativa. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#ciclo-principal-fases-0-a-10

- **Veredictos de SPAD** (GO / NO-GO): Toda revisión de SPAD termina con uno de tres veredictos: GO (se continúa), GO con cambios (se continúa tras corregir lo indicado) o NO-GO (vuelve a la fase que corresponda). Un solo vocabulario en todas las revisiones evita interpretaciones. El enlace abre los veredictos en la guía operativa. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#veredictos
- **Revisión humana del código**: En SPAD, el código que genera la IA siempre lo revisa una persona antes de integrarlo, y hay una segunda revisión en código sensible, agentes con autonomía y producción crítica. La IA ayuda a construir, pero no se aprueba a sí misma. El enlace abre la regla en la guía operativa. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#3-1-revision-humana-del-codigo

FUENTES Y ENLACES

- **SPAD 01 · Fases de construcción y revisión** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#4-2-detalle-de-las-fases-de-construccion-y-revision
- **SPAD 04 · Entrega a operación** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/04_SPAD_Ciclos_complementarios.html#entrega-a-operacion
- **Fases de construcción y revisión** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#4-2-detalle-de-las-fases-de-construccion-y-revision



CURSO COMPLETO DE SPAD

Veredictos y a qué fase se vuelve

El ciclo principal de SPAD usa un solo vocabulario de veredictos de SPAD en todas las revisiones. La persona acepta o rechaza el veredicto.

Revisión	GO	GO con cambios	NO-GO
Fase 2 · Revisión del plan	Pasa a la fase 3	Vuelve a la fase 1 para ajustar el plan	Vuelve a la fase 0: replantear el objetivo
Fase 7 · Revisión de pruebas	Pasa a la fase 8	Pasa a la fase 9 para corregir	Vuelve a la fase 6: rehacer las pruebas
Fase 8 · Revisión del código	Pasa a la fase 10	Pasa a la fase 9; después, otra vez a la 8	Vuelve a la fase 1: el diseño falla

Autocomprobación: la revisión del código da NO-GO porque el diseño no sirve. ¿A qué fase se vuelve? (Respuesta: a la 1.)

→ Veredictos

SPAD · Versión 0.x, en revisión · CC BY 4.0 · Fernando García Varela seachad.github.io/AI_CONSULTING/ 13 / 32

Notas

Todas las revisiones de SPAD usan el mismo vocabulario de tres veredictos:

- GO: se puede avanzar.
- GO con cambios: se puede avanzar tras corregir lo indicado.
- NO-GO: no se puede avanzar; hay que volver atrás.

La tabla dice a qué fase se vuelve en cada caso. Observe la lógica: cuanto más grave es el problema, más atrás se vuelve. Un NO-GO en la revisión del código no se arregla con parches: si el diseño falla, se vuelve al plan.

Tres reglas de los veredictos:

- Un veredicto sin justificación está incompleto.
- Un GO con hallazgos críticos o altos abiertos es autoaprobación, y la respuesta se invalida (módulo 4).
- Un NO-GO bien fundado no es un fracaso: es el proceso funcionando.

La IA revisora emite el veredicto, pero no decide: una persona lo acepta o lo rechaza y queda registrado.

Autocomprobación del módulo 3 (respuestas):

- La revisión del plan da GO con cambios. ¿Adónde? A la fase 1.
- La revisión de las pruebas da NO-GO. ¿Adónde? A la fase 6.
- Tras las correcciones de la fase 9, ¿adónde se va? A la fase 8, nunca a la 10.

Por qué importa: un comité de dirección también sabe parar un proyecto. SPAD hace lo mismo en cada revisión técnica, con reglas escritas y no a criterio del momento.

TÉRMINOS DE ESTA SLIDE

- **Ciclo principal de SPAD** (fases 0–10): Las once fases de SPAD: contexto, plan, revisión del plan, guía de código, estrategia de pruebas, implementación, pruebas, revisión de pruebas, revisión del código, correcciones y gestión de

versiones. Cada una deja un artefacto que se revisa. El enlace abre el ciclo en la guía operativa. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#ciclo-principal-fases-0-a-10

- **Veredictos de SPAD** (GO / NO-GO): Toda revisión de SPAD termina con uno de tres veredictos: GO (se continúa), GO con cambios (se continúa tras corregir lo indicado) o NO-GO (vuelve a la fase que corresponda). Un solo vocabulario en todas las revisiones evita interpretaciones. El enlace abre los veredictos en la guía operativa. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#veredictos

FUENTES Y ENLACES

- **SPAD 03 · Autoaprobación** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/03_SPAD_Politica_de_validacion.html#3-5-autoaprobacion



Notas

Objetivo del módulo 4: entender quién hace qué y cómo se controla.

SPAD reparte el trabajo en ocho roles. Uno es una persona: el orquestador, que dirige el proceso, elige qué IA cumple cada rol y valida cada artefacto. Los otros siete son roles de IA: planificadora, revisora, constructora, correctora, analista, de diagnóstico y revisora de seguridad.

Dos reglas ordenan los roles:

- Nunca hay dos roles en la misma fase.
- Quien revisa nunca es quien produjo lo que revisa.

La segunda parte del módulo trata de la política de validación: qué hace la persona cuando una respuesta de la IA rompe las reglas del proceso.

Ejemplo cotidiano: en una auditoría financiera, el auditor no prepara las cuentas que audita. SPAD aplica la misma separación a la IA. Por eso se usa el nombre «IA revisora» y no «auditor»: un auditor es una persona independiente, con otra función; la IA revisora es una herramienta y su veredicto lo acepta siempre una persona.

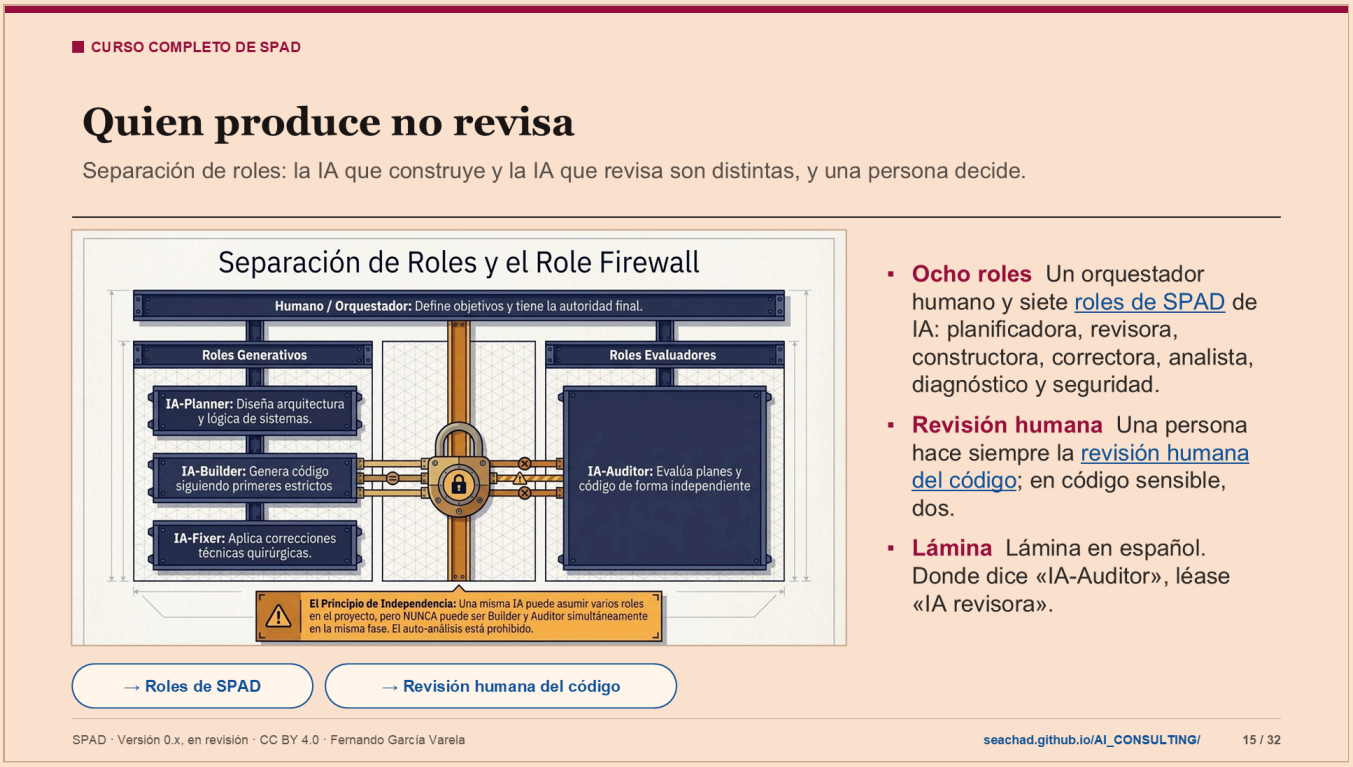
Pregunta para el aula: ¿por qué no basta con pedir a la misma IA que «revise bien su trabajo»?

TÉRMINOS DE ESTA SLIDE

- **Roles de SPAD**: Los papeles que intervienen en SPAD: el orquestador (una persona que dirige el ciclo y valida) y varias IA con funciones separadas: planificadora, revisora, constructora, correctora, analista y de diagnóstico. Quien construye no revisa lo que construye. El enlace abre el documento de presentación de SPAD, que describe cada rol. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html
- **Política de validación**: Documento de SPAD que fija qué debe validar una persona, cuándo y con qué rigor, según el riesgo del cambio, y qué evidencia queda de esa validación. Es la garantía de que la IA no decide sola qué entra en el código. El enlace abre el documento. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/03_SPAD_Politica_de_validacion.html

FUENTES Y ENLACES

- **SPAD 01 · Roles** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#roles
- **SPAD 03 · Autoaprobación** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/03_SPAD_Politica_de_validacion.html#3-5-autoaprobacion
- **Roles** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#roles



Notas

La lámina muestra la separación de roles con un candado: la IA que construye y la IA que revisa no se mezclan. La lámina usa un nombre antiguo, «IA-Auditor»; el nombre vigente es «IA revisora».

Los ocho roles:

- Orquestador (persona): dirige, elige los modelos y valida cada artefacto.
- IA planificadora: diseña el plan y la estrategia de pruebas.
- IA revisora: revisa planes, pruebas y código, y emite veredictos.
- IA constructora: implementa el código según la guía.
- IA correctora: aplica las correcciones mínimas.
- IA analista: documenta código existente (ciclo de legado).
- IA de diagnóstico: busca la causa raíz de un fallo; no escribe la solución.
- IA revisora de seguridad: revisa el código sensible.

Puede ser el mismo modelo de IA en roles distintos, pero nunca en la misma fase ni revisando lo que produjo.

Revisión humana: además de la IA revisora, una persona revisa siempre el código y queda registrado. En código sensible (autenticación, autorización, pagos, datos personales, límites de agentes, decisiones sobre personas o producción crítica) hacen falta dos revisiones humanas.

Ejemplo: en payment_gateway, una IA construye el cobro, otra lo revisa, una desarrolladora revisa el código y, por tocar pagos, un segundo revisor humano lo vuelve a mirar.

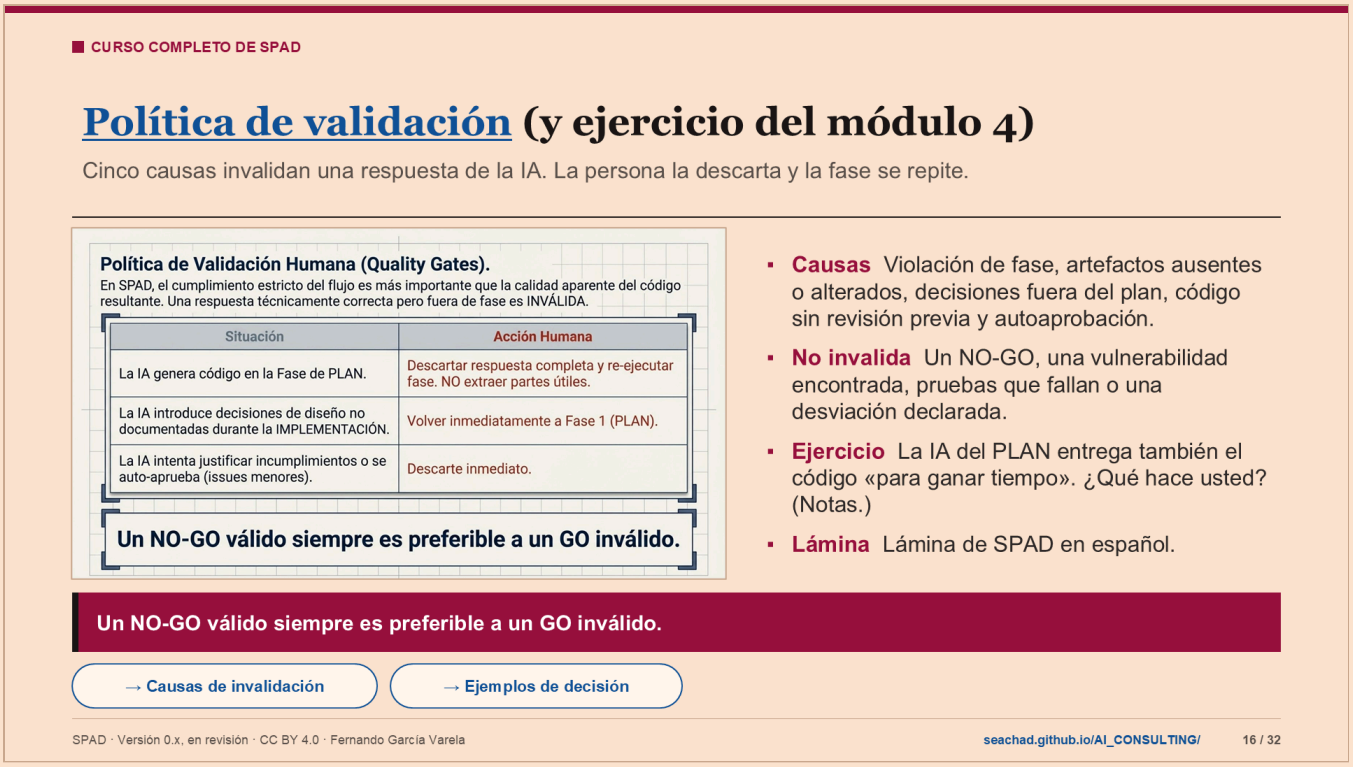
Por qué importa: la separación evita que los errores de un modelo se confirmen a sí mismos.

TÉRMINOS DE ESTA SLIDE

- **Roles de SPAD**: Los papeles que intervienen en SPAD: el orquestador (una persona que dirige el ciclo y valida) y varias IA con funciones separadas: planificadora, revisora, constructora, correctora, analista y de diagnóstico. Quien construye no revisa lo que construye. El enlace abre el documento de presentación de SPAD, que describe cada rol. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html
- **Revisión humana del código**: En SPAD, el código que genera la IA siempre lo revisa una persona antes de integrarlo, y hay una segunda revisión en código sensible, agentes con autonomía y producción crítica. La IA ayuda a construir, pero no se aprueba a sí misma. El enlace abre la regla en la guía operativa. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#3-1-revision-humana-del-codigo

FUENTES Y ENLACES

- **SPAD 01 · Roles** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#roles
- **Roles de SPAD** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#roles



Notas

La política de validación (documento 03) dice qué hace la persona cuando una respuesta de la IA rompe el proceso. La aplica siempre una persona.

Cinco causas invalidan una respuesta:

- Violación de fase: hace algo que no corresponde a su fase, por ejemplo escribir código en el PLAN.
- Artefactos ausentes o alterados: falta un documento obligatorio o se ha cambiado uno ya aprobado.
- Decisiones fuera del plan: la constructora cambia el diseño por su cuenta.
- Implementación sin revisión previa: se construye sobre un plan no revisado.
- Autoaprobación: la IA se aprueba a sí misma o da GO con hallazgos críticos o altos abiertos. Es la causa más grave.

Lo que no invalida: un NO-GO, encontrar vulnerabilidades, pruebas que fallan o una desviación declarada. Todo eso es información útil.

Procedimiento ante una respuesta inválida: reconocer la causa, no corregir la respuesta, descartarla entera, repetir la fase, reforzar las instrucciones y escalar si se repite. El máximo de intentos es un parámetro (valor de partida: 3).

Respuesta al ejercicio: la respuesta se descarta entera, aunque el código parezca bueno. Es una violación de fase. Se repite la fase 1 y se anota en el registro de infracciones. No se aprovecha «la parte buena».

Por qué importa: si se acepta una vez, la regla deja de existir.

TÉRMINOS DE ESTA SLIDE

- **Política de validación**: Documento de SPAD que fija qué debe validar una persona, cuándo y con qué rigor, según el riesgo del cambio, y qué evidencia queda de esa validación. Es la garantía de que la IA no decide sola qué entra en el código. El enlace abre el documento. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/03_SPAD_Politica_de_validacion.html

FUENTES Y ENLACES

- **SPAD 03 · Causas de invalidación** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/03_SPAD_Politica_de_validacion.html#causas-de-invalidacion
- **SPAD 03 · Lo que no invalida una respuesta** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/03_SPAD_Politica_de_validacion.html#lo-que-no-invalida-una-respuesta
- **SPAD 03 · Procedimiento ante una respuesta inválida** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/03_SPAD_Politica_de_validacion.html#procedimiento-ante-una-respuesta-invalida
- **SPAD 03 · Ejemplos de decisión** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/03_SPAD_Politica_de_validacion.html#ejemplos-de-decision
- **Causas de invalidación** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/03_SPAD_Politica_de_validacion.html#causas-de-invalidacion
- **Ejemplos de decisión** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/03_SPAD_Politica_de_validacion.html#ejemplos-de-decision



Notas

Objetivo del módulo 5: entender la «memoria» de SPAD.

Una IA no recuerda las normas de su compañía de un trabajo a otro. Si cada persona le explica las reglas a su manera, cada trabajo sale distinto. SPAD lo resuelve con tres piezas:

- Los contextos: documentos con las reglas que la IA recibe antes de cualquier fase. Hay uno global, común a toda la organización, y uno por proyecto.
- El tema: cada funcionalidad, corrección o incidente tiene un nombre corto y una carpeta donde se guardan todos sus artefactos.
- Los registros: qué modelo produjo cada artefacto, quién lo validó y qué infracciones hubo.

Ejemplo cotidiano: un despacho profesional tiene un manual de estilo común (contexto global), unas instrucciones por cliente (contexto de proyecto) y un expediente por asunto (tema). Nadie empieza un asunto sin abrir su expediente.

Por qué importa: sin contextos, la calidad depende de quién escribe las instrucciones. Sin tema, el trabajo se pierde en conversaciones sueltas. Sin registros, no hay forma de responder a una auditoría.

Pregunta para el aula: ¿dónde están hoy las normas técnicas de su compañía? ¿Las recibe la IA que usan sus equipos?

TÉRMINOS DE ESTA SLIDE

- **Contextos de SPAD**: Documentos de reglas que la IA debe respetar en cada fase: el contexto global (normas de la organización, datos, modelos aprobados) y el contexto de proyecto. Las excepciones las evalúa la IA revisora y las aprueba una persona. Sirven para que la IA no trabaje con supuestos propios. El enlace abre los contextos. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/02_SPAD_Contextos_temas_y_registro.html#contextos
- **Temas de SPAD**: Un tema es la unidad de trabajo de SPAD: una funcionalidad o un cambio que recorre el ciclo completo con su propia carpeta de artefactos. Permite seguir de principio a fin qué se pidió, qué se planificó, qué se

construyó y quién lo validó. El enlace abre su definición. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/02_SPAD_Contextos_temas_y_registro.html#temas

- **Registros por artefacto**: Cada artefacto de SPAD lleva tres registros: del modelo (proveedor, modelo, versión, instrucción y contextos usados), de validación humana (quién lo revisó y cuándo) y de infracciones (reglas que no se cumplieron). Hacen el proceso auditable. El enlace abre los registros. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/02_SPAD_Contextos_temas_y_registro.html#4-1-registro-del-modelo

FUENTES Y ENLACES

- **SPAD 00 · Contextos y temas** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#contextos
- **Contextos, temas y registro** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/02_SPAD_Contextos_temas_y_registro.html



Notas

La lámina muestra las tres piezas que la IA recibe o usa en cada trabajo.

Contexto global: lo fija la organización y vale para todos los proyectos. Incluye cómo se nombran los artefactos, la cobertura mínima de pruebas (el documento propone 80 % como valor de partida, que cada organización ajusta), el versionado, la seguridad base (con OWASP Top 10 como referencia), las reglas de datos y de proveedores de modelos, las prohibiciones, los parámetros de los ciclos complementarios y los modelos aprobados.

Contexto de proyecto: añade lo propio del proyecto: tecnología, arquitectura, reglas de negocio, cumplimiento, convenciones y despliegue. Si contradice al global, solo prevalece con una excepción escrita y justificada que aprueba una persona, nunca una IA. Ejemplo del documento: bajar la cobertura del 80 % al 70 % durante una migración, con motivo y fecha.

Tema: un nombre corto en minúsculas, con guiones bajos y hasta 20 caracteres, como user_auth o payment_gateway. Cada tema tiene su carpeta. Los intentos rechazados se guardan: si el primer plan fue rechazado, la carpeta tiene el plan 01 rechazado y el plan 02 aprobado.

Por qué importa: la excepción escrita convierte una decisión informal («esta vez no hacemos pruebas») en una decisión con responsable, motivo y fecha.

Pregunta para el aula: ¿quién debería aprobar en su compañía una excepción al contexto global?

TÉRMINOS DE ESTA SLIDE

- **Contextos de SPAD**: Documentos de reglas que la IA debe respetar en cada fase: el contexto global (normas de la organización, datos, modelos aprobados) y el contexto de proyecto. Las excepciones las evalúa la IA revisora y las aprueba una persona. Sirven para que la IA no trabaje con supuestos propios. El enlace abre los contextos. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/02_SPAD_Contextos_temas_y_registro.html#contextos
- **Temas de SPAD**: Un tema es la unidad de trabajo de SPAD: una funcionalidad o un cambio que recorre el ciclo completo con su propia carpeta de artefactos. Permite seguir de principio a fin qué se pidió, qué se planificó, qué se

construyó y quién lo validó. El enlace abre su definición. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/02_SPAD_Contextos_temas_y_registro.html#temas

FUENTES Y ENLACES

- **SPAD 02 · Contexto global** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/02_SPAD_Contextos_temas_y_registro.html#2-1-contexto-global
- **SPAD 02 · Contexto de proyecto** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/02_SPAD_Contextos_temas_y_registro.html#2-2-contexto-de-proyecto
- **SPAD 02 · Organización de artefactos** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/02_SPAD_Contextos_temas_y_registro.html#3-1-organizacion-de-artefactos
- **OWASP Top 10 (fuente oficial)** https://top10.owasp.org/2025/es/
- **Precedencia y excepciones** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/02_SPAD_Contextos_temas_y_registro.html#2-3-precedencia-y-excepciones



Notas

Cada tema guarda tres registros.

Registro del modelo: cada artefacto indica qué proveedor, modelo y versión lo produjeron, con qué instrucciones y con qué contextos. Los modelos cambian a menudo. Si dentro de seis meses aparece un defecto, este registro permite saber con qué se hizo y repetirlo.

Registro de validación humana: cada veredicto aceptado o rechazado lleva el nombre de la persona, la fecha y el motivo. Es la regla «ninguna IA aprueba» convertida en evidencia.

Registro de infracciones: cada respuesta descartada por la política de validación, con su causa. No es un registro de culpas: sirve para mejorar las instrucciones y para comparar qué modelos cumplen mejor el proceso, como veremos en las métricas del módulo 8.

Ejercicio del módulo 5 (cinco minutos, individual): elija un trabajo reciente hecho con IA en su equipo y responda: ¿qué modelo lo produjo?, ¿con qué instrucciones?, ¿quién lo validó?, ¿hubo intentos descartados? Después repase la lista de comprobación del documento 02, que resume lo que debe estar listo antes de empezar: contextos cargados, tema creado, roles asignados y modelos aprobados.

Respuesta esperada: casi nadie podrá responder a las cuatro preguntas. Ese es el valor de los registros.

Por qué importa para un responsable de tecnología: los registros cuestan poco si se llevan desde el principio y mucho si se reconstruyen para una auditoría.

TÉRMINOS DE ESTA SLIDE

- **Registros por artefacto**: Cada artefacto de SPAD lleva tres registros: del modelo (proveedor, modelo, versión, instrucción y contextos usados), de validación humana (quién lo revisó y cuándo) y de infracciones (reglas que no se cumplieron). Hacen el proceso auditable. El enlace abre los registros. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/02_SPAD_Contextos_temas_y_registro.html#4-1-registro-del-modelo
- **Veredictos de SPAD** (GO / NO-GO): Toda revisión de SPAD termina con uno de tres veredictos: GO (se continúa), GO con cambios (se continúa tras corregir lo indicado) o NO-GO (vuelve a la fase que corresponda). Un

solo vocabulario en todas las revisiones evita interpretaciones. El enlace abre los veredictos en la guía operativa. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#veredictos

- **Política de validación**: Documento de SPAD que fija qué debe validar una persona, cuándo y con qué rigor, según el riesgo del cambio, y qué evidencia queda de esa validación. Es la garantía de que la IA no decide sola qué entra en el código. El enlace abre el documento. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/03_SPAD_Politica_de_validacion.html

FUENTES Y ENLACES

- **SPAD 02 · Registros por artefacto** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/02_SPAD_Contextos_temas_y_registro.html#registros-por-artefacto
- **SPAD 03 · Registro de validación e infracciones** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/03_SPAD_Politica_de_validacion.html#registro-de-validacion-e-infracciones
- **SPAD 02 · Lista de comprobación** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/02_SPAD_Contextos_temas_y_registro.html#lista-de-comprobacion-antes-de-empezar
- **Registros por artefacto** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/02_SPAD_Contextos_temas_y_registro.html#registros-por-artefacto
- **Lista antes de empezar** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/02_SPAD_Contextos_temas_y_registro.html#lista-de-comprobacion-antes-de-empezar



Notas

Objetivo del módulo 6: conocer los ciclos que acompañan al principal.

No todo el trabajo empieza desde cero. A veces hay que tocar un sistema antiguo que nadie documentó, investigar por qué falla algo o arreglar un problema urgente en producción. SPAD no inventa un método distinto para cada caso: define ciclos complementarios que preparan o aceleran el ciclo principal, sin saltarse sus reglas.

El módulo también trata la seguridad. En SPAD la seguridad no es una revisión opcional al final: para el código sensible es una fase obligatoria, con reglas claras sobre qué hallazgos bloquean el despliegue y quién puede aceptar un riesgo.

Ejemplo cotidiano: un hospital tiene un protocolo de urgencias. Va más deprisa, pero no se salta la identificación del paciente ni el informe posterior. La corrección urgente de SPAD funciona igual.

Pregunta para el aula: cuando hay una urgencia en producción en su compañía, ¿qué controles se saltan? ¿Se revisan después?

TÉRMINOS DE ESTA SLIDE

- **Ciclos complementarios de SPAD**: Variantes del ciclo para situaciones especiales: código heredado (con pruebas de caracterización), depuración, correcciones urgentes (con análisis posterior obligatorio), seguridad y cadena de suministro. Mantienen el control cuando no se puede seguir el ciclo completo. El enlace abre el documento. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/04_SPAD_Ciclos_complementarios.html

FUENTES Y ENLACES

- **SPAD 00 · Otros ciclos de trabajo** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#otros-ciclos-de-trabajo
- **Otros ciclos de trabajo** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#otros-ciclos-de-trabajo



Notas

Los cuatro ciclos complementarios del documento 04:

- Legado: antes de cambiar un sistema que nadie documentó, la IA analista lo documenta y analiza el impacto del cambio. Después se escriben pruebas de caracterización: pruebas que fijan cómo se comporta hoy el sistema, aunque no sea perfecto. Así, cualquier cambio que altere ese comportamiento se detecta. Solo entonces empieza la fase 1.
- Depuración: la IA de diagnóstico investiga un fallo y entrega un informe de causa raíz con un veredicto: el problema está en el código, en la configuración o en la infraestructura. No escribe la solución; la solución sigue el ciclo principal.
- Corrección urgente: para incidentes en producción. Acelera, pero no quita fases. La revisión y las pruebas mínimas son obligatorias, el análisis posterior es obligatorio aunque la corrección haya funcionado, y una corrección provisional abre un tema nuevo para planificar la solución definitiva. Los plazos los fija el contexto global.
- Seguridad: revisión S1 obligatoria para autenticación, autorización, pagos, datos personales, entradas no confiables y componentes de IA en producción. Incluye la cadena de suministro: que las dependencias existan, sus vulnerabilidades conocidas, sus licencias y una lista de componentes del software.

Ejemplo: un cobro duplicado en payment_gateway. Primero, depuración para hallar la causa. Si urge, corrección urgente con revisión y análisis posterior. Después, tema nuevo con la solución definitiva por el ciclo principal.

Por qué importa: la urgencia es el momento en que más controles se saltan; SPAD la acelera sin dejarla sin control.

TÉRMINOS DE ESTA SLIDE

- **Ciclos complementarios de SPAD**: Variantes del ciclo para situaciones especiales: código heredado (con pruebas de caracterización), depuración, correcciones urgentes (con análisis posterior obligatorio), seguridad y cadena de suministro. Mantienen el control cuando no se puede seguir el ciclo completo. El enlace abre el documento. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/04_SPAD_Ciclos_complementarios.html
- **Ciclo principal de SPAD** (fases 0–10): Las once fases de SPAD: contexto, plan, revisión del plan, guía de código, estrategia de pruebas, implementación, pruebas, revisión de pruebas, revisión del código, correcciones y gestión de versiones. Cada una deja un artefacto que se revisa. El enlace abre el ciclo en la guía operativa. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#ciclo-principal-fases-0-a-10
- **Temas de SPAD**: Un tema es la unidad de trabajo de SPAD: una funcionalidad o un cambio que recorre el ciclo completo con su propia carpeta de artefactos. Permite seguir de principio a fin qué se pidió, qué se planificó, qué se construyó y quién lo validó. El enlace abre su definición. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/02_SPAD_Contextos_temas_y_registro.html#temas

FUENTES Y ENLACES

- **SPAD 04 · Ciclo de legado** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/04_SPAD_Ciclos_complementarios.html#ciclo-de-legado
- **SPAD 04 · Ciclo de depuración** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/04_SPAD_Ciclos_complementarios.html#ciclo-de-depuracion
- **SPAD 04 · Ciclo de corrección urgente** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/04_SPAD_Ciclos_complementarios.html#ciclo-de-correccion-urgente
- **SPAD 04 · Ciclo de seguridad** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/04_SPAD_Ciclos_complementarios.html#ciclo-de-seguridad
- **Cadena de suministro y licencias** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/04_SPAD_Ciclos_complementarios.html#5-4-cadena-de-suministro-y-licencias



Notas

La lámina tiene dos partes. A la izquierda, las pruebas obligatorias: la estrategia se define en la fase 4, antes de escribir código; las pruebas se construyen en la fase 6; y en la fase 7 se comprueba la cobertura real frente a la esperada. A la derecha, la revisión de seguridad del código sensible, con OWASP como lista de referencia (para componentes de IA, también la lista OWASP para aplicaciones con modelos de lenguaje).

Las severidades (documento 04, sección 5.3):

- Crítica: bloquea el despliegue, vuelve a correcciones y a una nueva revisión de seguridad. Nadie puede aceptarla como riesgo residual.
- Alta: bloquea el despliegue. Solo de forma excepcional la acepta la autoridad de riesgo definida en el contexto global, por escrito y con fecha de corrección.
- Media: no bloquea. Se corrige o la acepta la persona con autoridad según el contexto global.
- Baja: no bloquea. Se corrige o la acepta la persona que orquesta.

Un riesgo residual nunca lo acepta una IA: siempre una persona con autoridad, por escrito.

Respuesta al ejercicio: la vulnerabilidad alta bloquea el lanzamiento. La IA no decide. Solo la autoridad de riesgo del contexto global puede aceptarla, de forma excepcional, por escrito y con una fecha de corrección. Si fuera crítica, no habría lanzamiento el lunes.

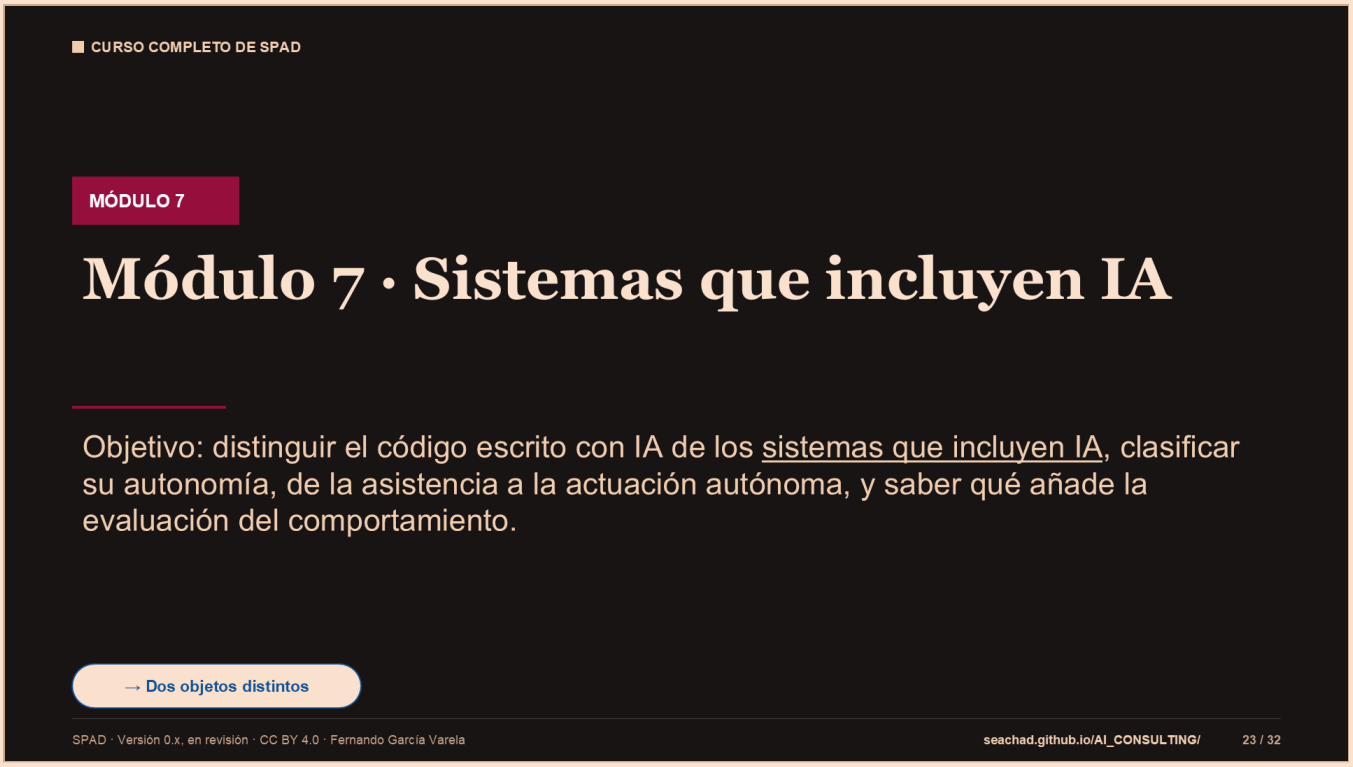
Por qué importa: la presión de la fecha es real; la regla escrita protege a quien tiene que decir que no.

TÉRMINOS DE ESTA SLIDE

- **Ciclo principal de SPAD** (fases 0–10): Las once fases de SPAD: contexto, plan, revisión del plan, guía de código, estrategia de pruebas, implementación, pruebas, revisión de pruebas, revisión del código, correcciones y gestión de versiones. Cada una deja un artefacto que se revisa. El enlace abre el ciclo en la guía operativa. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#ciclo-principal-fases-0-a-10

FUENTES Y ENLACES

- **SPAD 04 · Severidades y aceptación del riesgo** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/04_SPAD_Ciclos_complementarios.html#5-3-severidades-bloqueo-y-aceptacion-del-riesgo
- **OWASP Top 10 (fuente oficial)** https://top10.owasp.org/2025/es/
- **OWASP Top 10 para aplicaciones con LLM** https://genai.owasp.org/resource/owasp-top-10-for-llm-applications-2025/
- **Severidades y aceptación del riesgo** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/04_SPAD_Ciclos_complementarios.html#5-3-severidades-bloqueo-y-aceptacion-del-riesgo



Notas

Objetivo del módulo 7: entender el caso de mayor riesgo.

Hasta ahora hemos hablado de código escrito con ayuda de IA: la IA produce código, pruebas y documentación, que se revisan antes de desplegar. Una vez en producción, el programa hace siempre lo mismo.

Un sistema que incluye IA es otra cosa: su comportamiento en producción depende de un modelo, de unas instrucciones, de una base de conocimiento o de un agente que ejecuta acciones. Responde a entradas que nadie ha visto antes. Y puede cambiar de comportamiento sin que nadie toque el código: basta con que el proveedor actualice el modelo.

Ejemplo: un programa que calcula el IVA de una factura es código. Un asistente que responde a las preguntas de los clientes sobre su factura es un sistema que incluye IA. El primero se prueba una vez; el segundo hay que evaluarlo antes y vigilarlo durante el uso.

Riesgos propios del segundo: respuestas incorrectas dadas con seguridad, inyección de instrucciones, fuga de datos, acciones no autorizadas, sesgo y deriva.

Pregunta para el aula: ¿cuántos de los proyectos de IA de su compañía son de un tipo y cuántos del otro?

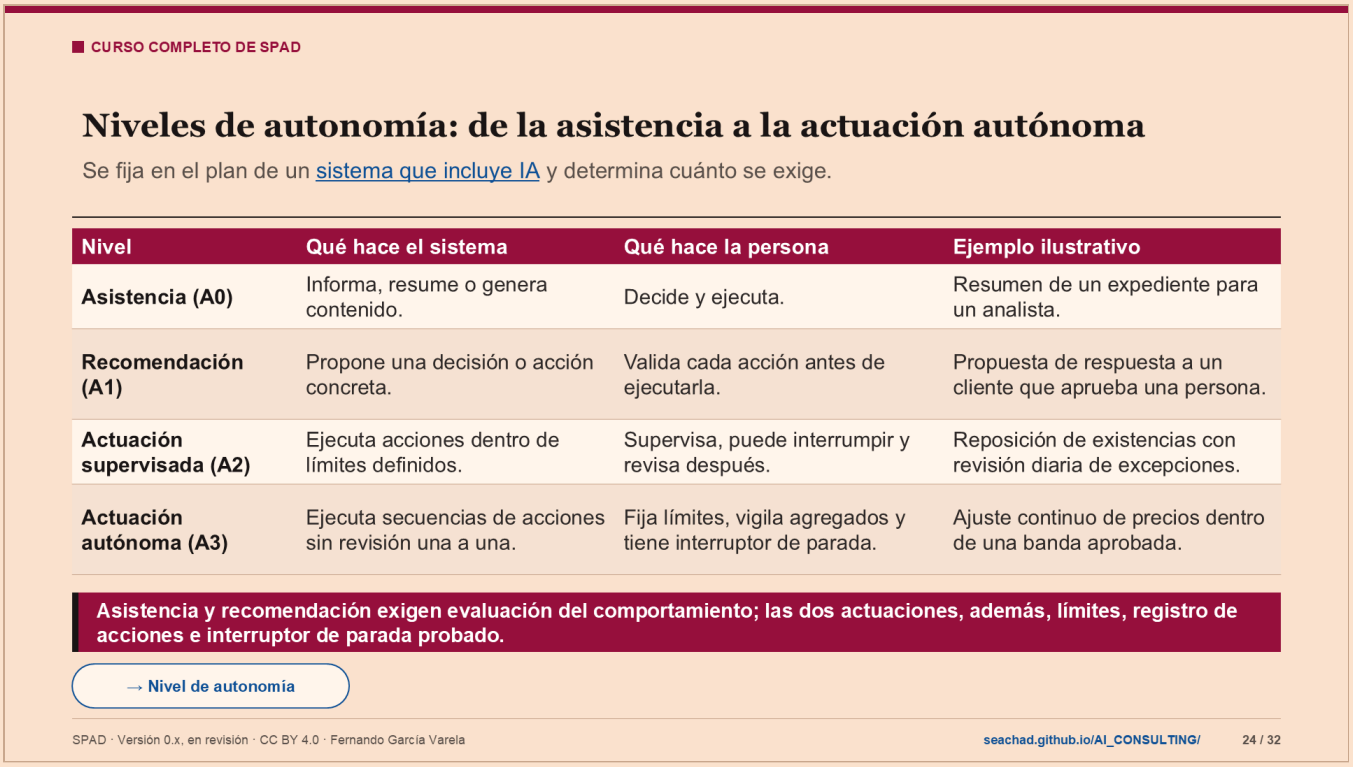
TÉRMINOS DE ESTA SLIDE

- **Sistemas que incluyen IA**: Qué añade SPAD cuando lo que se construye es un sistema que usa IA: niveles de autonomía, una fase de evaluación del comportamiento con conjuntos de prueba, pruebas adversarias y de sesgo, plan de agentes y control de cambios que no tocan el código (como cambiar el modelo). El enlace abre el documento. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/05_SPAD_Sistemas_que_incluyen_IA.html

FUENTES Y ENLACES

- **SPAD 05 · Dos objetos distintos** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/05_SPAD_Sistemas_que_incluyen_IA.html#dos-objetos-distintos

- **Dos objetos distintos** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/05_SPAD_Sistemas_que_incluyen_IA.html#dos-objetos-distintos



CURSO COMPLETO DE SPAD

Niveles de autonomía: de la asistencia a la actuación autónoma

Se fija en el plan de un sistema que incluye IA y determina cuánto se exige.

Nivel	Qué hace el sistema	Qué hace la persona	Ejemplo ilustrativo
Asistencia (A0)	Informa, resume o genera contenido.	Decide y ejecuta.	Resumen de un expediente para un analista.
Recomendación (A1)	Propone una decisión o acción concreta.	Valida cada acción antes de ejecutarla.	Propuesta de respuesta a un cliente que aprueba una persona.
Actuación supervisada (A2)	Ejecuta acciones dentro de límites definidos.	Supervisa, puede interrumpir y revisa después.	Reposición de existencias con revisión diaria de excepciones.
Actuación autónoma (A3)	Ejecuta secuencias de acciones sin revisión una a una.	Fija límites, vigila agregados y tiene interruptor de parada.	Ajuste continuo de precios dentro de una banda aprobada.

Asistencia y recomendación exigen evaluación del comportamiento; las dos actuaciones, además, límites, registro de acciones e interruptor de parada probado.

→ Nivel de autonomía

SPAD · Versión 0.x, en revisión · CC BY 4.0 · Fernando García Varela seachad.github.io/AI_CONSULTING/ 24 / 32

Notas

Todo componente de IA que decide o actúa se clasifica desde el plan con una escala de autonomía de cuatro niveles, de A0 a A3. La tabla resume el documento 05; los ejemplos son ilustrativos.

La clave para un público de negocio está en la columna de la persona. En A0 la persona decide todo. En A1 valida cada acción. En A2 ya no valida una a una: supervisa y revisa después. En A3 solo fija los límites y vigila el conjunto, con un interruptor de parada.

Cuanto más actúa el sistema, más se exige:

- A0 y A1: evaluación del comportamiento.
- A2 y A3: además, límites de actuación, registro de cada acción, interruptor de parada probado y un plan de agentes con sus herramientas y permisos.

Ejemplo de debate: un asistente que redacta respuestas a reclamaciones es A1 si una persona las aprueba antes de enviarlas. Si se decide enviarlas automáticamente para ganar tiempo, pasa a A2. Eso no es un ajuste técnico: es una decisión de negocio que cambia los requisitos.

Pregunta para el aula: piense en un caso de uso de IA de su sector. ¿En qué nivel está hoy? ¿A qué nivel quieren llevarlo? ¿Qué tendrían que demostrar antes?

TÉRMINOS DE ESTA SLIDE

- **Sistemas que incluyen IA**: Qué añade SPAD cuando lo que se construye es un sistema que usa IA: niveles de autonomía, una fase de evaluación del comportamiento con conjuntos de prueba, pruebas adversarias y de sesgo, plan de agentes y control de cambios que no tocan el código (como cambiar el modelo). El enlace abre el documento. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/05_SPAD_Sistemas_que_incluyen_IA.html

FUENTES Y ENLACES

- **SPAD 05 · Nivel de autonomía** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/05_SPAD_Sistemas_que_incluyen_IA.html#nivel-de-autonomia
- **SPAD 05 · Agentes** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/05_SPAD_Sistemas_que_incluyen_IA.html#agentes

- **Nivel de autonomía** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/05_SPAD_Sistemas_que_incluyen_IA.html#nivel-de-autonomia



Notas

Las pruebas de código verifican que el programa hace lo que dice el código. La evaluación del comportamiento verifica que el sistema hace lo que dice el objetivo.

La fase E ejecuta el sistema sobre un conjunto de evaluación: casos representativos del uso real, con la respuesta esperada o el criterio para juzgarla, incluidos casos que el sistema debe rechazar. Incluye:

- Criterios de aceptación con umbrales fijados de antemano (exactitud, rechazo correcto, latencia, coste).
- Pruebas adversarias: intentos de inyección de instrucciones, directa o escondida en un documento recuperado.
- Pruebas de sesgo con pares contrafactuales: la misma entrada cambiando solo un atributo protegido. Son obligatorias si el sistema decide, recomienda o se comunica con personas.
- Una muestra revisada por personas.
- El modelo de respaldo, si lo hay, con el mismo conjunto.

Un NO-GO devuelve el trabajo a la fase 1 si el fallo es de diseño y a la fase 5 si es de instrucciones.

Agentes: contratos de herramientas con permisos mínimos, límites, memoria controlada, registro de acciones, identidad propia e interruptor de parada probado. Una regla clave: el contenido recuperado es dato, no instrucción.

Respuesta al ejercicio: se repite la evaluación del comportamiento con el mismo conjunto versionado antes de seguir usando el sistema. En operación cambian cosas sin que cambie el código: el modelo, las instrucciones, la base de conocimiento, el uso y el coste.

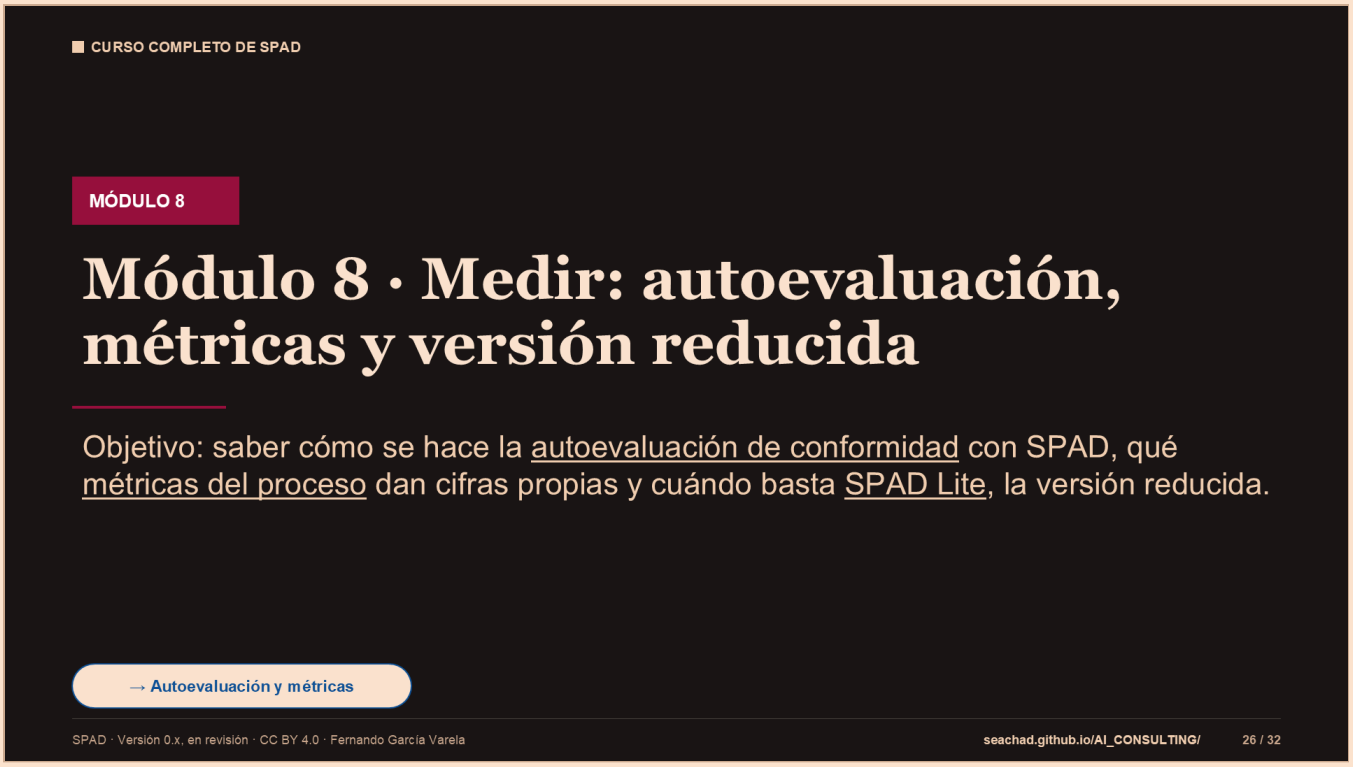
Por qué importa: sin conjunto de evaluación, la única prueba de que el asistente funciona es una demostración con las preguntas que se le ocurrieron a alguien.

TÉRMINOS DE ESTA SLIDE

- **Sistemas que incluyen IA**: Qué añade SPAD cuando lo que se construye es un sistema que usa IA: niveles de autonomía, una fase de evaluación del comportamiento con conjuntos de prueba, pruebas adversarias y de sesgo, plan de agentes y control de cambios que no tocan el código (como cambiar el modelo). El enlace abre el documento. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/05_SPAD_Sistemas_que_incluyen_IA.html
- **Ciclo principal de SPAD** (fases 0–10): Las once fases de SPAD: contexto, plan, revisión del plan, guía de código, estrategia de pruebas, implementación, pruebas, revisión de pruebas, revisión del código, correcciones y gestión de versiones. Cada una deja un artefacto que se revisa. El enlace abre el ciclo en la guía operativa. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#ciclo-principal-fases-0-a-10

FUENTES Y ENLACES

- **SPAD 05 · Fase E · Evaluación del comportamiento** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/05_SPAD_Sistemas_que_incluyen_IA.html#fase-e-evaluacion-del-comportamiento
- **SPAD 05 · Qué añade a cada fase** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/05_SPAD_Sistemas_que_incluyen_IA.html#que-anade-a-cada-fase
- **SPAD 05 · Lo que cambia sin cambiar el código** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/05_SPAD_Sistemas_que_incluyen_IA.html#operacion-lo-que-cambia-sin-que-cambie-el-codigo
- **OWASP Top 10 para aplicaciones con LLM** https://genai.owasp.org/resource/owasp-top-10-for-llm-applications-2025/
- **OWASP Top 10 para aplicaciones agénticas** https://genai.owasp.org/resource/owasp-top-10-for-agentic-applications-for-2026/
- **Fase E · Evaluación** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/05_SPAD_Sistemas_que_incluyen_IA.html#fase-e-evaluacion-del-comportamiento
- **Agentes** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/05_SPAD_Sistemas_que_incluyen_IA.html#agentes



Notas

Objetivo del módulo 8: medir sin prometer.

Una metodología que no se mide se convierte en un ritual. SPAD da dos instrumentos:

- La autoevaluación de conformidad: una revisión interna, hecha por una persona independiente de quienes construyeron, que comprueba si un trabajo se hizo conforme a SPAD.
- Las métricas del proceso: indicadores calculados con los registros del módulo 5.

Un mensaje importante para el aula: SPAD no afirma cifras de mejora. No dice «reduce los incidentes un tanto por ciento». Cada organización mide frente a su propia situación de partida: retrabajo después de la entrega, incidentes por trabajo entregado, horas para preparar una auditoría y tiempo hasta producción, antes y después de usar SPAD.

El módulo cierra con la versión reducida, porque las métricas también dicen cuándo el método completo es excesivo para un trabajo pequeño.

Ejemplo cotidiano: un plan de entrenamiento no promete cuántos kilos perderá usted; le dice qué medir cada semana para saber si funciona.

Pregunta para el aula: si un proveedor le promete una cifra exacta de mejora con su método, ¿qué le pediría antes de creerla?

TÉRMINOS DE ESTA SLIDE

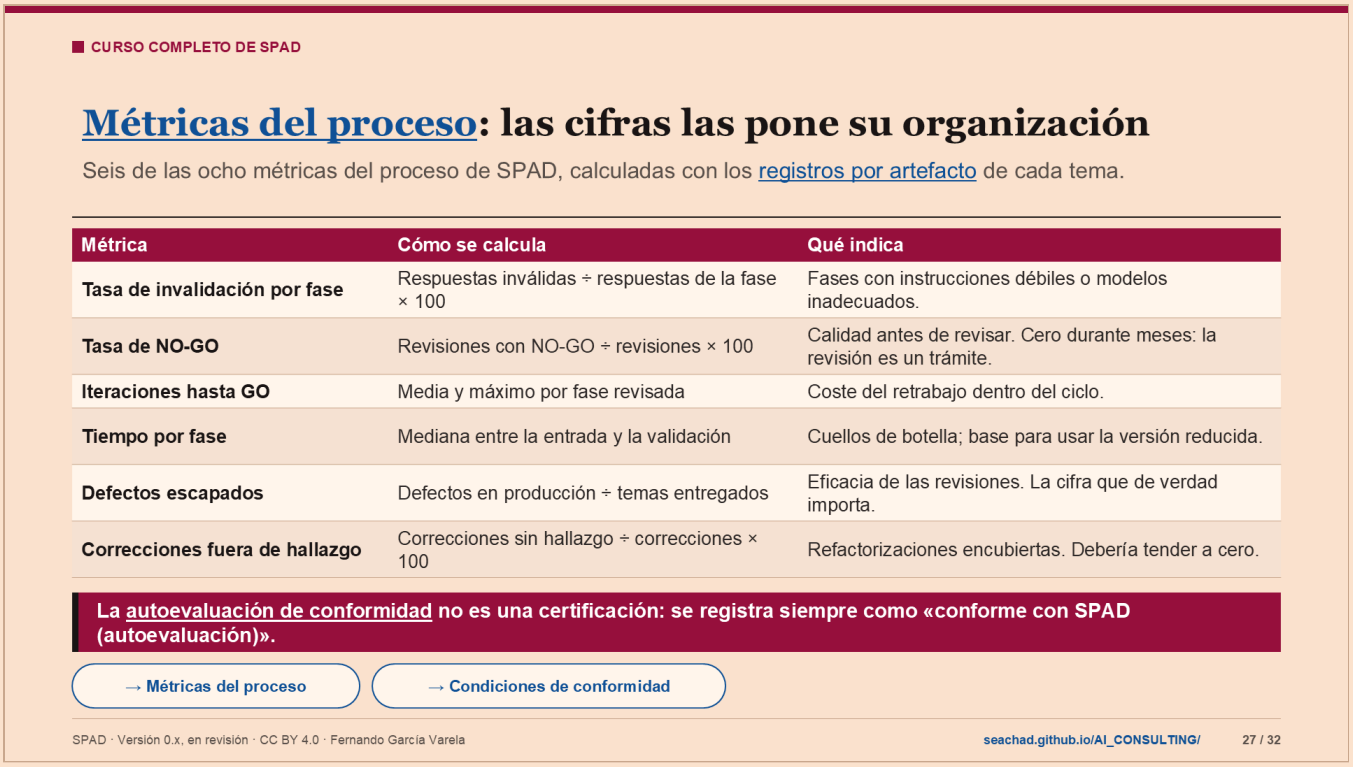
- **Autoevaluación de conformidad**: Revisión con la que un equipo comprueba si aplica SPAD como dice: qué fases sigue, qué registros guarda y qué falla. Vale doce meses y no la otorga nadie desde fuera. Sirve para mejorar el proceso con datos y no con impresiones. El enlace abre la autoevaluación. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/08_SPAD_Autoevaluacion_y_metricas.html#que-es-la-autoevaluacion
- **Métricas del proceso**: Indicadores que mide un equipo que usa SPAD, como cuántas revisiones terminan en NO-GO, cuántos defectos llegan a producción o cuánto tarda un tema. SPAD no promete cifras: propone medirlas y

fijar objetivos propios. El enlace abre las métricas. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/08_SPAD_Autoevaluacion_y_metricas.html#metricas-del-proceso

- **SPAD Lite**: Versión reducida de SPAD para cambios pequeños y de bajo riesgo, con solo tres artefactos. Mantiene la revisión y la validación humana, pero con menos pasos. Sirve para no cargar un cambio menor con todo el ciclo. El enlace abre la versión reducida en la guía operativa. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#version-reducida-spad-lite

FUENTES Y ENLACES

- **Autoevaluación y métricas** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/08_SPAD_Autoevaluacion_y_metricas.html



CURSO COMPLETO DE SPAD

Métricas del proceso: las cifras las pone su organización

Seis de las ocho métricas del proceso de SPAD, calculadas con los registros por artefacto de cada tema.

Métrica	Cómo se calcula	Qué indica
Tasa de invalidación por fase	Respuestas inválidas ÷ respuestas de la fase × 100	Fases con instrucciones débiles o modelos inadecuados.
Tasa de NO-GO	Revisiones con NO-GO ÷ revisiones × 100	Calidad antes de revisar. Cero durante meses: la revisión es un trámite.
Iteraciones hasta GO	Media y máximo por fase revisada	Coste del retrabajo dentro del ciclo.
Tiempo por fase	Mediana entre la entrada y la validación	Cuellos de botella; base para usar la versión reducida.
Defectos escapados	Defectos en producción ÷ temas entregados	Eficacia de las revisiones. La cifra que de verdad importa.
Correcciones fuera de hallazgo	Correcciones sin hallazgo ÷ correcciones × 100	Refactorizaciones encubiertas. Debería tender a cero.

La autoevaluación de conformidad no es una certificación: se registra siempre como «conforme con SPAD (autoevaluación)».

→ Métricas del proceso → Condiciones de conformidad

SPAD · Versión 0.x, en revisión · CC BY 4.0 · Fernando García Varela seachad.github.io/AI_CONSULTING/ 27 / 32

Notas

La tabla recoge seis de las ocho métricas del proceso. Las otras dos son la tasa de invalidación por modelo (qué modelos siguen mejor las reglas) y la cobertura real frente a la exigida. Para sistemas con IA se añaden las métricas de la evaluación del comportamiento y, en operación, la deriva y el coste por interacción.

Cómo leerlas, con ejemplos del documento 08:

- Una tasa de invalidación alta en una sola fase indica que su instrucción necesita reforzarse; no es un problema del método.
- Una tasa de NO-GO cercana a cero durante meses no es buena noticia: suele indicar que la revisión se ha vuelto un trámite.
- Si el tiempo por fase supera con mucho al de construir en trabajos pequeños, conviene la versión reducida.
- Los defectos escapados son la cifra que de verdad importa: miden si las revisiones funcionan.

La autoevaluación comprueba seis condiciones (siete si el sistema incluye IA): las decisiones se tomaron antes del código, cada fase produjo su artefacto, las revisiones fueron independientes, no se saltó ninguna fase, las correcciones fueron mínimas y ninguna IA aprobó nada. Si falta un artefacto obligatorio, el resultado es negativo. El resultado es conforme, conforme con condiciones o no conforme, y su vigencia la fija el contexto global (valor de partida: doce meses).

No es una certificación ni un sello: presentarla así incumple el uso permitido del nombre SPAD.

Pregunta para el aula: ¿qué métrica le pediría usted a su equipo en el primer trimestre?

TÉRMINOS DE ESTA SLIDE

- **Métricas del proceso**: Indicadores que mide un equipo que usa SPAD, como cuántas revisiones terminan en NO-GO, cuántos defectos llegan a producción o cuánto tarda un tema. SPAD no promete cifras: propone medirlas y

fijar objetivos propios. El enlace abre las métricas. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/08_SPAD_Autoevaluacion_y_metricas.html#metricas-del-proceso

- **Registros por artefacto**: Cada artefacto de SPAD lleva tres registros: del modelo (proveedor, modelo, versión, instrucción y contextos usados), de validación humana (quién lo revisó y cuándo) y de infracciones (reglas que no se cumplieron). Hacen el proceso auditable. El enlace abre los registros. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/02_SPAD_Contextos_temas_y_registro.html#4-1-registro-del-modelo
- **Autoevaluación de conformidad**: Revisión con la que un equipo comprueba si aplica SPAD como dice: qué fases sigue, qué registros guarda y qué falla. Vale doce meses y no la otorga nadie desde fuera. Sirve para mejorar el proceso con datos y no con impresiones. El enlace abre la autoevaluación. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/08_SPAD_Autoevaluacion_y_metricas.html#que-es-la-autoevaluacion

FUENTES Y ENLACES

- **SPAD 08 · Procedimiento de autoevaluación** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/08_SPAD_Autoevaluacion_y_metricas.html#procedimiento
- **SPAD 08 · Cuándo conviene usar SPAD** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/08_SPAD_Autoevaluacion_y_metricas.html#cuando-conviene-usar-spad-segun-las-metricas
- **Condiciones de conformidad** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/08_SPAD_Autoevaluacion_y_metricas.html#condiciones-de-conformidad



Notas

SPAD completo no tiene sentido para cada cambio pequeño. Por eso la guía operativa define una versión reducida, SPAD Lite, con tres artefactos:

- Plan revisado: agrupa las fases 0 a 4.
- Entrega revisada: agrupa las fases 5 a 9.
- Versión: la fase 10.

Las fases se agrupan, pero no desaparecen (regla 1). Y hay cuatro cosas que nunca se recortan: la separación entre quien construye y quien revisa, la validación humana, la política de validación y el registro del tema.

Se usa la versión completa si se cumple cualquiera de estos criterios: código sensible (autenticación, pagos, datos personales…), agentes A2 o A3, producción crítica, varios equipos o proveedores, o sector regulado.

El documento 00 añade cuándo usar SPAD: es muy recomendable en sistemas que van a producción, que tocan datos sensibles, dinero o decisiones sobre personas, o en sectores regulados. No es necesario para prototipos desechables sin datos reales. Y al principio hace el trabajo más lento, mientras el equipo aprende.

Autocomprobación del módulo 8 (respuestas):

- Cambio en la nómina con datos personales: versión completa, por código sensible.
- Cambio de texto en una pantalla interna, sin datos personales ni dinero: puede ir en versión reducida.
- Prototipo para una demostración interna, sin datos reales: SPAD no es necesario.

Por qué importa: la proporcionalidad evita que el método se abandone por pesado.

TÉRMINOS DE ESTA SLIDE

- **Ciclo principal de SPAD** (fases 0–10): Las once fases de SPAD: contexto, plan, revisión del plan, guía de código, estrategia de pruebas, implementación, pruebas, revisión de pruebas, revisión del código, correcciones y gestión de

versiones. Cada una deja un artefacto que se revisa. El enlace abre el ciclo en la guía operativa. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#ciclo-principal-fases-0-a-10

- **SPAD Lite**: Versión reducida de SPAD para cambios pequeños y de bajo riesgo, con solo tres artefactos. Mantiene la revisión y la validación humana, pero con menos pasos. Sirve para no cargar un cambio menor con todo el ciclo. El enlace abre la versión reducida en la guía operativa. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#version-reducida-spad-lite
- **Política de validación**: Documento de SPAD que fija qué debe validar una persona, cuándo y con qué rigor, según el riesgo del cambio, y qué evidencia queda de esa validación. Es la garantía de que la IA no decide sola qué entra en el código. El enlace abre el documento. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/03_SPAD_Politica_de_validacion.html

FUENTES Y ENLACES

- **SPAD 00 · Cuándo usar SPAD** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#cuando-usar-spad
- **Artefactos obligatorios** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/08_SPAD_Autoevaluacion_y_metricas.html#artefactos-obligatorios



Notas

Objetivo del módulo 9: llevar SPAD a la conversación de la dirección. Es el módulo más importante para un público directivo.

SPAD trata de cómo se construye software con IA. Para una compañía tiene dos usos.

Uso 1 · Construir su propio software con IA, con método. Si sus equipos ya usan asistentes de código o agentes, SPAD les da un ciclo con fases bloqueantes, revisiones independientes, validación humana y artefactos que se pueden auditar. El resultado es software que se puede mantener y explicar, no solo código que funciona en una demostración.

Uso 2 · Exigir como cliente. Sus proveedores e integradores también construyen con IA y son más productivos. Es posición del autor que esa ganancia debe llegar al cliente: como menor coste o como más alcance por el mismo precio. Para negociarlo con datos, la compañía puede pedir lo que SPAD ya define: los artefactos de cada fase, el registro del modelo, el registro de validación humana, los veredictos de cada revisión y las métricas del proceso.

Los dos usos se apoyan en lo mismo: un trabajo con IA que deja rastro. Sin rastro, la compañía no sabe qué hizo la IA, quién lo revisó ni cuánto esfuerzo costó de verdad.

Pregunta para el aula: ¿sabe su compañía si sus proveedores construyen con IA, y si esa ganancia aparece en sus facturas?

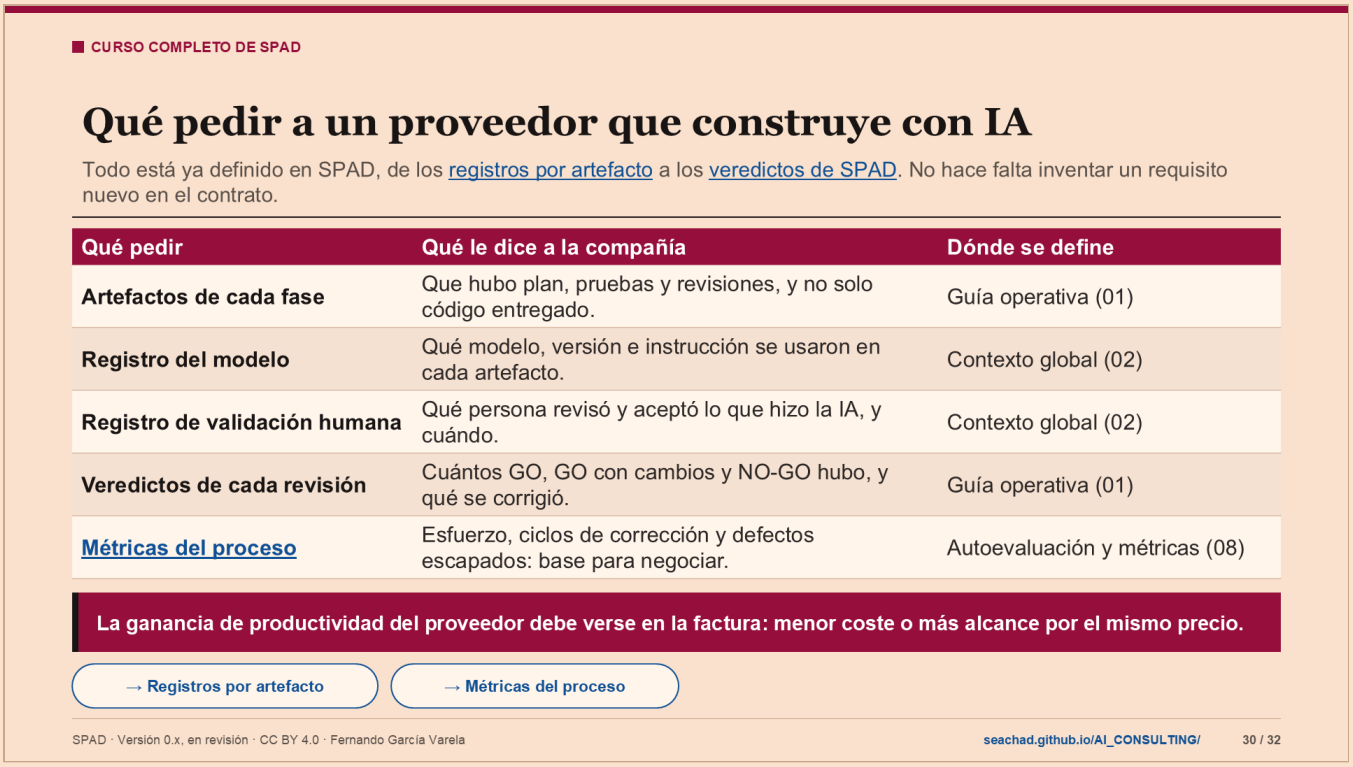
TÉRMINOS DE ESTA SLIDE

- **SPAD**: SPAD (Structured Prompt-Driven Engineering) es una metodología para construir software con ayuda de la IA de forma ordenada: fases secuenciales, cada una con su artefacto, revisiones con veredicto y personas que validan. Es secuencial, bloqueante y auditable. La usan equipos de desarrollo. El enlace abre su documento de presentación. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html

FUENTES Y ENLACES

CURSO COMPLETO DE SPAD

- **SPAD 00 · Para qué le sirve a una compañía** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#para-que-le-sirve-a-una-compania
- **SPAD 02 · Registros por artefacto** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/02_SPAD_Contextos_temas_y_registro.html#registros-por-artefacto
- **Para qué le sirve a una compañía** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#para-que-le-sirve-a-una-compania



Notas

Esta tabla es la lista práctica del uso 2: qué puede pedir la compañía, como cliente, a un proveedor o a un integrador que construye con IA.

Nada de la tabla es un requisito nuevo. Son artefactos y registros que SPAD ya define:

- Los artefactos de cada fase (plan, guía de código, estrategia de pruebas, informes de revisión, versión) muestran que hubo método.
- El registro del modelo dice qué proveedor, modelo, versión e instrucción produjeron cada artefacto.
- El registro de validación humana dice qué persona revisó y aceptó cada resultado. Un veredicto de una IA revisora nunca sustituye a esa persona.
- Los veredictos dicen cuántas revisiones pasaron a la primera y cuántas volvieron atrás.
- Las métricas del proceso dan el esfuerzo real, los ciclos de corrección y los defectos que llegaron a producción.

Con estos datos la compañía puede negociar. Es posición del autor que, si el proveedor tarda menos gracias a la IA, esa ganancia debe trasladarse al cliente: como menor coste o como más alcance por el mismo precio. SPAD no fija ninguna cifra: la cifra sale de las métricas de cada proveedor.

Un aviso: pedir los registros no basta. La compañía tiene que leerlos. Una tasa de NO-GO cercana a cero durante meses suele indicar que la revisión se ha vuelto un trámite.

Por qué importa: convierte una promesa comercial («usamos IA») en datos que se pueden comprobar en un contrato.

TÉRMINOS DE ESTA SLIDE

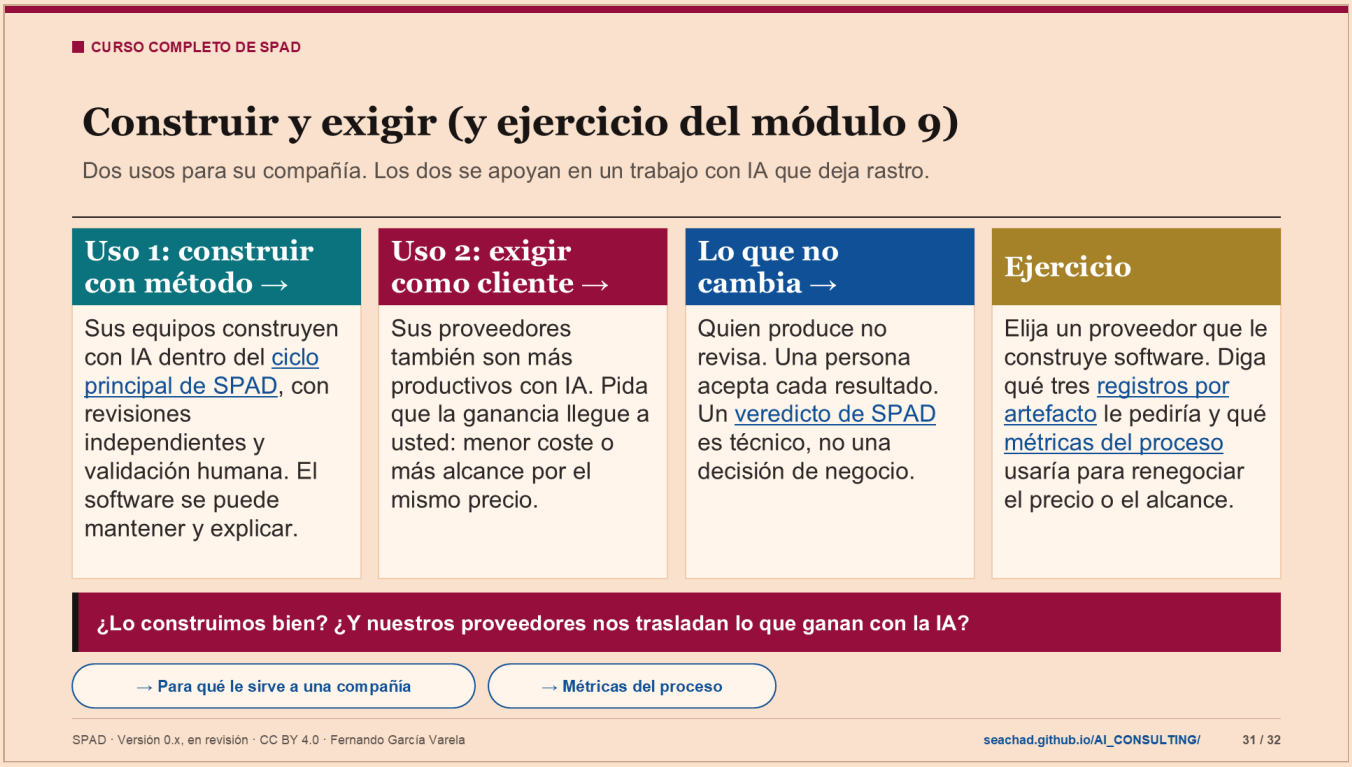
- **Registros por artefacto**: Cada artefacto de SPAD lleva tres registros: del modelo (proveedor, modelo, versión, instrucción y contextos usados), de validación humana (quién lo revisó y cuándo) y de infracciones (reglas que no

se cumplieron). Hacen el proceso auditable. El enlace abre los registros. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/02_SPAD_Contextos_temas_y_registro.html#4-1-registro-del-modelo

- **Veredictos de SPAD** (GO / NO-GO): Toda revisión de SPAD termina con uno de tres veredictos: GO (se continúa), GO con cambios (se continúa tras corregir lo indicado) o NO-GO (vuelve a la fase que corresponda). Un solo vocabulario en todas las revisiones evita interpretaciones. El enlace abre los veredictos en la guía operativa. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#veredictos
- **Métricas del proceso**: Indicadores que mide un equipo que usa SPAD, como cuántas revisiones terminan en NO-GO, cuántos defectos llegan a producción o cuánto tarda un tema. SPAD no promete cifras: propone medirlas y fijar objetivos propios. El enlace abre las métricas. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/08_SPAD_Autoevaluacion_y_metricas.html#metricas-del-proceso

FUENTES Y ENLACES

- **SPAD 02 · Registros por artefacto** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/02_SPAD_Contextos_temas_y_registro.html#registros-por-artefacto
- **SPAD 02 · Registro de validación humana** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/02_SPAD_Contextos_temas_y_registro.html#4-2-registro-de-validacion-humana
- **Registros por artefacto** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/02_SPAD_Contextos_temas_y_registro.html#registros-por-artefacto



Notas

Esta slide resume el módulo 9 en dos usos y una regla que no cambia.

Uso 1: la compañía construye su propio software con IA, con método. Gana trazabilidad, calidad y la posibilidad de explicar qué hizo la IA y quién lo aceptó.

Uso 2: la compañía exige como cliente. Es posición del autor que la ganancia de productividad de sus proveedores debe trasladarse: como menor coste o como más alcance por el mismo precio. Los artefactos, los registros, los veredictos y las métricas de SPAD dan los datos para esa conversación.

Lo que no cambia en ninguno de los dos usos: quien produce no revisa su propia salida; una persona acepta cada resultado; y un GO técnico solo dice que el trabajo está bien construido, no que la compañía deba ponerlo en producción.

Ejercicio del módulo 9 (diez minutos, en grupos). Ejemplo de respuesta ficticio: un integrador mantiene la web de comercio electrónico de la compañía. Le pediría el registro del modelo, el registro de validación humana y las métricas del proceso de los últimos seis meses. Usaría el esfuerzo por cambio y los ciclos de corrección: si el esfuerzo baja y la tarifa no, propondría bajar el precio por cambio o incluir más cambios por la misma cuota. Si el integrador no puede dar esos registros, eso también es una respuesta.

Por qué importa: la IA cambia el coste de construir software, dentro y fuera de la compañía. Sin datos, ese cambio se lo queda otro.

TÉRMINOS DE ESTA SLIDE

- **Ciclo principal de SPAD** (fases 0–10): Las once fases de SPAD: contexto, plan, revisión del plan, guía de código, estrategia de pruebas, implementación, pruebas, revisión de pruebas, revisión del código, correcciones y gestión de versiones. Cada una deja un artefacto que se revisa. El enlace abre el ciclo en la guía operativa. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#ciclo-principal-fases-0-a-10
- **Veredictos de SPAD** (GO / NO-GO): Toda revisión de SPAD termina con uno de tres veredictos: GO (se continúa), GO con cambios (se continúa tras corregir lo indicado) o NO-GO (vuelve a la fase que corresponda). Un

solo vocabulario en todas las revisiones evita interpretaciones. El enlace abre los veredictos en la guía operativa. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#veredictos

- **Registros por artefacto**: Cada artefacto de SPAD lleva tres registros: del modelo (proveedor, modelo, versión, instrucción y contextos usados), de validación humana (quién lo revisó y cuándo) y de infracciones (reglas que no se cumplieron). Hacen el proceso auditable. El enlace abre los registros. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/02_SPAD_Contextos_temas_y_registro.html#4-1-registro-del-modelo
- **Métricas del proceso**: Indicadores que mide un equipo que usa SPAD, como cuántas revisiones terminan en NO-GO, cuántos defectos llegan a producción o cuánto tarda un tema. SPAD no promete cifras: propone medirlas y fijar objetivos propios. El enlace abre las métricas. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/08_SPAD_Autoevaluacion_y_metricas.html#metricas-del-proceso

FUENTES Y ENLACES

- **SPAD 00 · Para qué le sirve a una compañía** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#para-que-le-sirve-a-una-compania
- **SPAD 03 · Autoaprobación** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/03_SPAD_Politica_de_validacion.html#3-5-autoaprobacion
- **Para qué le sirve a una compañía** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#para-que-le-sirve-a-una-compania



Notas

Cuatro casos ficticios para debatir en grupos (quince a veinte minutos). Lectura de SPAD para el profesor:

- «El prototipo ya funciona»: un prototipo desechable sin datos reales no necesita SPAD. Pero en cuanto va a producción, con datos o clientes reales, deja de ser un prototipo. Lo razonable es abrir un tema y recorrer el ciclo: como mínimo, la versión reducida; si toca datos personales, dinero o decisiones sobre personas, la completa, con revisión humana, pruebas y seguridad.
- De A2 a A3: quitar la revisión diaria convierte un agente supervisado en autónomo. No es un ajuste técnico: es una decisión con nuevos requisitos (límites estrictos, vigilancia de agregados, interruptor de parada probado) y la versión completa de SPAD. Debe decidirla una persona con autoridad, no el equipo para ganar tiempo.
- La revisión siempre dice GO: según el documento 08, una tasa de NO-GO cercana a cero durante meses suele indicar que la revisión se ha vuelto un trámite. Hay que revisar su independencia y mirar los defectos escapados.
- «Certificados en SPAD»: SPAD no certifica. La autoevaluación se registra siempre como «conforme con SPAD (autoevaluación)». Anunciar una certificación incumple el uso permitido del nombre y expone a la compañía.

Cierre: SPAD se publica con licencia CC BY 4.0 (contenidos) y MIT (código), citando la autoría: SPAD · Fernando García Varela. Los documentos 06 y 07 contienen las instrucciones y los contratos de cada fase para quien quiera aplicarlo.

Un marco aparte. SPAD trata solo de cómo se construye el software; no es gobierno de la IA de la compañía. Quien quiera gobernar qué se construye, con qué riesgo y cómo se mide su valor tiene un marco aparte, SEVEN-G. El único puente es su documento 53, y los veredictos de SPAD no sustituyen a sus puertas de decisión.

TÉRMINOS DE ESTA SLIDE

- **Sistemas que incluyen IA**: Qué añade SPAD cuando lo que se construye es un sistema que usa IA: niveles de autonomía, una fase de evaluación del comportamiento con conjuntos de prueba, pruebas adversarias y de sesgo, plan de agentes y control de cambios que no tocan el código (como cambiar el modelo). El enlace abre el documento. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/05_SPAD_Sistemas_que_incluyen_IA.html
- **Veredictos de SPAD** (GO / NO-GO): Toda revisión de SPAD termina con uno de tres veredictos: GO (se continúa), GO con cambios (se continúa tras corregir lo indicado) o NO-GO (vuelve a la fase que corresponda). Un solo vocabulario en todas las revisiones evita interpretaciones. El enlace abre los veredictos en la guía operativa. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#veredictos
- **Autoevaluación de conformidad**: Revisión con la que un equipo comprueba si aplica SPAD como dice: qué fases sigue, qué registros guarda y qué falla. Vale doce meses y no la otorga nadie desde fuera. Sirve para mejorar el proceso con datos y no con impresiones. El enlace abre la autoevaluación. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/08_SPAD_Autoevaluacion_y_metricas.html#que-es-la-autoevaluacion

FUENTES Y ENLACES

- **SPAD 00 · Cuándo usar SPAD** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#cuando-usar-spad
- **SPAD 07 · Contratos de entrada y salida** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/07_SPAD_Contratos_de_entrada_y_salida.html#objeto
- **SEVEN-G 53 · Construcción de soluciones con IA (único puente)** https://seachad.github.io/AI_CONSULTING/SEVEN-G/html/es/53_SEVEN-G_Construccion_de_soluciones_con_IA.html
- **SPAD 00 · Estado y licencia** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#estado-licencia-y-proximos-pasos
- **Biblioteca SPAD** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/index.html
- **Ejemplos de decisión** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/03_SPAD_Politica_de_validacion.html#ejemplos-de-decision

AVISO LEGAL

Aviso legal: ayuda metodológica genérica y gratuita, «tal cual» y con fines informativos; no es asesoramiento jurídico ni está revisada jurídicamente para ningún caso. Cada organización la adapta y es responsable de su revisión legal y de su cumplimiento regulatorio; puede no recoger cambios normativos recientes. En la máxima medida permitida por la ley, el autor no asume responsabilidad alguna por los efectos de su aplicación. No otorga certificación de ningún tipo. Detalle en el documento 93.

CC BY 4.0 (contenidos) · MIT (código) · Fernando García Varela · seachad.github.io/AI_CONSULTING/