

Notas

SPAD (Structured Prompt-Driven Engineering, ingeniería estructurada guiada por instrucciones) es el marco de trabajo de Fernando García Varela para gestionar el desarrollo de software con inteligencia artificial de forma controlada. Convierte la IA de un asistente informal de programación en una herramienta de ingeniería gobernada: su trabajo puede revisarse, reproducirse y auditarse.

SPAD no es una colección de instrucciones para pedir código. Es una forma de trabajar con tres propiedades:

- Secuencial: el trabajo avanza por fases en un orden fijo.
- Bloqueante: ninguna fase empieza hasta que la anterior ha producido su resultado y una persona lo ha validado.
- Auditable: cada fase deja un artefacto explícito (plan, informe de revisión, estrategia de pruebas, registro de versión).

Por qué importa: la IA ya escribe código en muchas compañías. La pregunta para la dirección no es si se usa, sino si alguien controla qué decide, qué se revisa y qué evidencias quedan.

Esta presentación recorre el problema, el principio, el ciclo de fases, los roles, la validación, los registros, los ciclos complementarios, los sistemas que incluyen IA, la medición, la versión reducida y para qué le sirve a una compañía: construir con método y exigir a sus proveedores.

SPAD se publica con licencia CC BY 4.0 para los contenidos y MIT para el código. No es una certificación y no promete cifras de mejora.

Pregunta para el aula: en su compañía, ¿quién decide hoy cómo se diseña el código que escribe una IA?

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

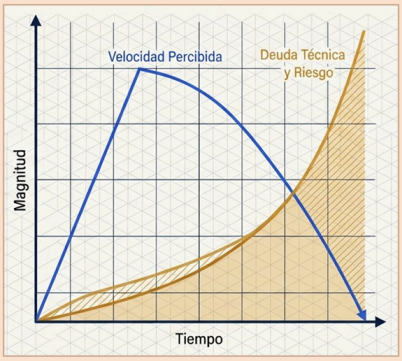
FUENTES Y ENLACES

- **SPAD 00 · Qué es SPAD y para qué sirve** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#que-es-spad
- **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
- **Qué es SPAD** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html
- **Biblioteca SPAD** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/index.html

SPAD EN 20 SLIDES

El problema: la ilusión de velocidad

La IA genera código plausible muy rápido. Sin método, la deuda técnica y el riesgo crecen sin que nadie lo vea.



- Funciona en la demostración y falla en producción: nadie diseñó la solución.
- Nadie sabe por qué el sistema es así: las decisiones las tomó la IA dentro del código.
- La auditoría pide evidencias y no existen: todo quedó en conversaciones con la IA.
- Gráfica conceptual de SPAD, sin datos. Lámina en español.

No son problemas de la IA: son problemas de método que la IA amplifica.

→ El problema que resuelve

SPAD · Versión 0.x, en revisión · CC BY 4.0 · Fernando García Varela seachad.github.io/AI_CONSULTING/ 2 / 20

Notas

La gráfica es conceptual y no contiene datos. Muestra una idea: al principio, la velocidad percibida sube mucho porque la IA produce código en minutos. Con el tiempo, esa velocidad cae, porque cada cambio choca con decisiones que nadie tomó de forma consciente. Mientras tanto, la deuda técnica y el riesgo crecen.

SPAD identifica seis síntomas habituales:

- El código funciona en la demostración y falla en producción.
- Nadie sabe por qué el sistema está construido así.
- Las pruebas son escasas o solo prueban el caso fácil.
- Aparecen vulnerabilidades en código que nadie revisó.
- Un cambio pequeño rompe otra parte del sistema.
- Una auditoría pide evidencias del desarrollo y no existen.

Ejemplo sencillo (ficticio): un equipo construye en una semana un asistente interno pidiendo código a la IA hasta que «funciona». Con datos reales falla. Nadie sabe qué decisiones tomó la IA, las pruebas solo cubren el caso de la demostración y no queda rastro de lo que se pidió.

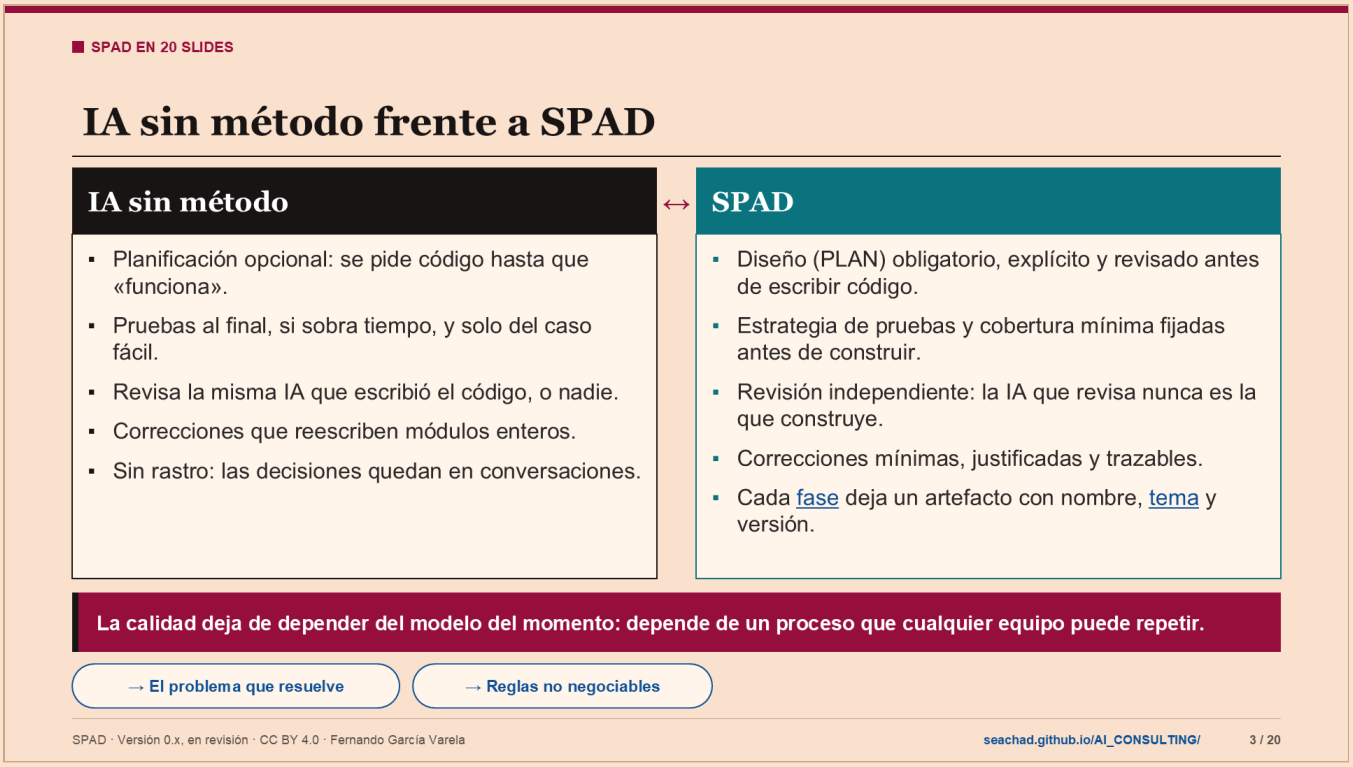
Por qué importa: ninguno de estos problemas es de la IA. Son problemas de método que la IA amplifica. Con un método, la calidad no depende de la destreza de quien escribe las instrucciones ni del modelo del momento.

Error frecuente: medir el avance por la cantidad de código generado, y no por lo que está diseñado, probado y revisado.

Pregunta para el aula: ¿qué síntoma de la lista ha visto en un proyecto real? ¿Cuánto costó corregirlo después?

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
- **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



Notas

Esta comparación resume la tabla de síntomas del documento 00 de SPAD. Cada fila de la izquierda tiene una respuesta concreta a la derecha.

- Diseño: sin método, el diseño aparece solo, a fuerza de pedir código. Con SPAD, el plan es obligatorio, se escribe sin código y otra IA lo revisa antes de construir. Toda decisión queda en el plan, no escondida en el código.
- Pruebas: sin método se escriben al final. Con SPAD, la estrategia de pruebas y su cobertura mínima se fijan antes de construir y una revisión comprueba que se cumplen.
- Revisión: sin método revisa la misma IA o nadie. Con SPAD, la IA que revisa nunca es la que construye y una persona revisa además el código.
- Correcciones: sin método se reescriben módulos enteros y se rompen otras partes. Con SPAD, cada corrección es mínima, justificada y trazable.
- Rastro: sin método todo queda en conversaciones. Con SPAD, cada fase deja un artefacto guardado en una estructura común.

Por qué importa para un directivo: cada fila de la derecha es un control que se puede comprobar. Si alguien afirma que su equipo trabaja con SPAD, se puede pedir el plan, su revisión y el registro de quién lo validó.

Error frecuente: pensar que SPAD es «escribir mejores instrucciones». La calidad de una instrucción ayuda, pero lo que da control es el orden de las fases, la separación de roles y la validación humana.

Pregunta para el aula: ¿cuál de las cinco filas cree que más se incumple en los equipos que conoce? ¿Por qué?

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

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



Notas

La lámina resume la lógica de SPAD en tres verbos.

- Pensar: antes de escribir código, el problema se describe y la solución se diseña en un plan. El plan no contiene código.
- Validar: una IA distinta de la que produjo el artefacto lo revisa y emite un veredicto. Una persona, la que orquesta el trabajo, acepta o rechaza ese veredicto. Ninguna fase empieza sin esa validación: es el principio de ejecución bloqueante.
- Ejecutar: la IA que construye sigue el plan y las reglas de código. Si encuentra algo que el plan no prevé, no decide: lo señala y pide decisión.

El principio que lo ordena todo: el cumplimiento del flujo es más importante que la calidad aparente del resultado. Una respuesta técnicamente correcta que se salta las reglas de su fase no se aprovecha: se descarta entera y la fase se repite. La prioridad de validación es: primero, cumplimiento del proceso; segundo, corrección técnica; tercero, eficiencia.

Ejemplo sencillo: la IA entrega un plan excelente que incluye quinientas líneas de código «de ejemplo». Aunque el código sea bueno, la respuesta se descarta, porque el plan define y no implementa.

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

Pregunta para el aula: ¿le parece razonable descartar un buen resultado solo porque no siguió el proceso? ¿Qué se pierde y qué se gana?

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

FUENTES Y ENLACES

- **SPAD 00 · 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
- **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



Notas

La guía operativa de SPAD fija doce reglas. Cada una protege algo concreto:

- 1. Ninguna fase se salta (protege el bloqueo). En la versión reducida las fases se agrupan, pero no desaparecen.
- 2. Toda salida es auditable (la trazabilidad).
- 3. Toda decisión es explícita: no hay decisiones de diseño escondidas en el código (la explicabilidad).
- 4. El código ejecuta, no decide: la IA constructora no toma decisiones de diseño (el plan).
- 5. Revisiones independientes: la IA revisora nunca es la constructora y una persona revisa además el código (la calidad).
- 6. Correcciones mínimas, nunca refactorizaciones encubiertas (la estabilidad).
- 7. Pruebas obligatorias, con cobertura mínima fijada antes de construir.
- 8. Seguridad por diseño: los hallazgos críticos y altos bloquean.
- 9. Contextos global y de proyecto cargados antes de cualquier fase.
- 10. Un tema por trabajo (la memoria de la organización).
- 11. Modelo registrado: qué modelo, versión e instrucciones produjeron cada artefacto (la reproducibilidad).
- 12. Ninguna IA aprueba (la responsabilidad).

Nota sobre la lámina: donde dice «auditor de IA» se refiere a la IA revisora, que es una IA y no una persona.

Por qué importa: son reglas simples de comprobar. Un directivo no necesita leer código para preguntar si se cumplen.

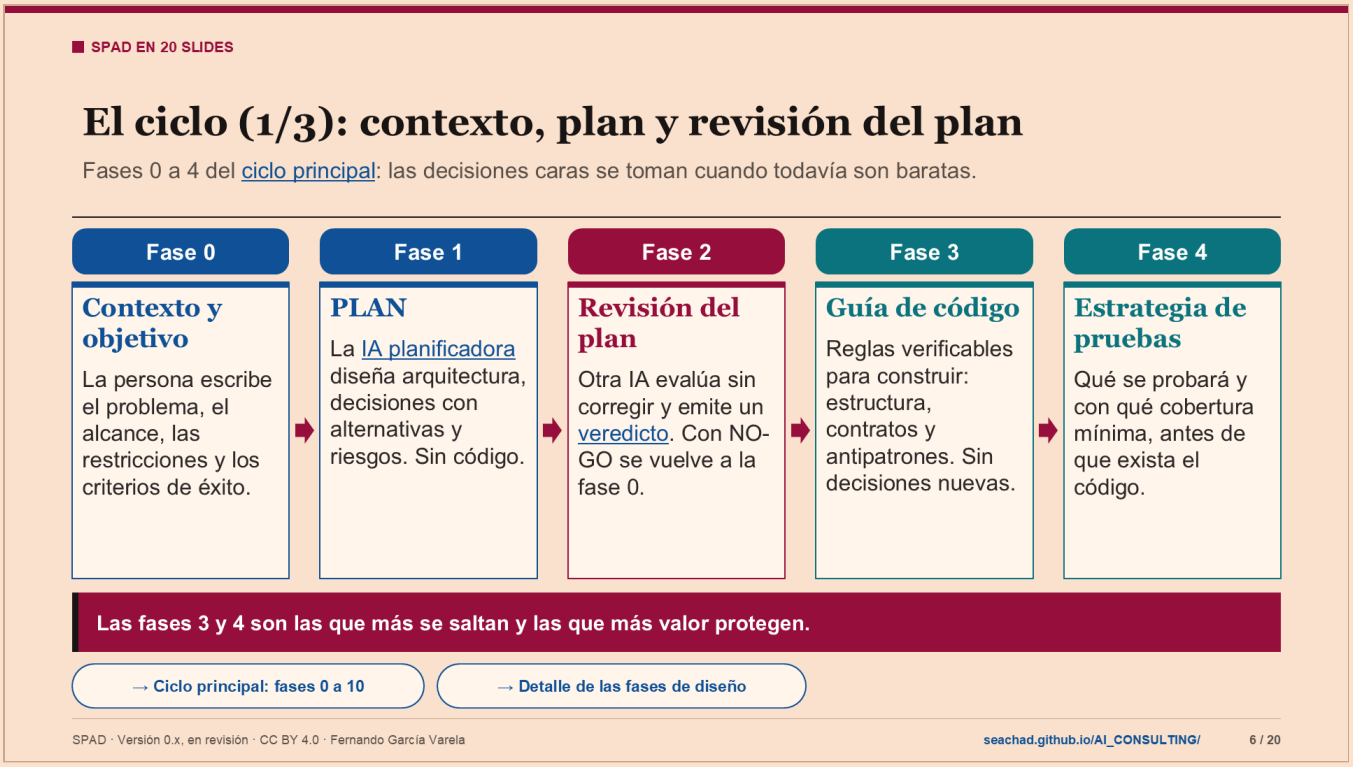
Pregunta para el aula: ¿cuál de las doce reglas le parece más difícil de mantener bajo presión de plazos?

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 01 · Reglas operativas no negociables** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#reglas-operativas-no-negociables
- **SPAD 00 · 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
- **Las doce reglas** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#reglas-operativas-no-negociables



Notas

El ciclo principal sirve para construir una funcionalidad nueva. Tiene once fases, numeradas de 0 a 10. Esta slide muestra las cinco de diseño.

- Fase 0, contexto y objetivo: la persona que orquesta escribe qué problema se resuelve, para quién, qué queda dentro y fuera del alcance, qué restricciones hay (tecnológicas, legales, de plazo) y cómo se sabrá que el resultado es correcto. Sin criterios de éxito no hay fase 1.
- Fase 1, PLAN: la IA planificadora diseña sin escribir código. Cada decisión se registra con sus alternativas y contrapartidas, y cada riesgo con su severidad y probabilidad.
- Fase 2, revisión del plan: una IA revisora, distinta de la planificadora, evalúa acoplamiento, cohesión, escalabilidad, concurrencia, seguridad y observabilidad. No corrige. Con GO con cambios, el plan vuelve a la fase 1; con NO-GO, se vuelve a la fase 0, porque el problema está mal planteado.
- Fase 3, guía de código: traduce el plan en reglas que quien construye puede cumplir y comprobar.
- Fase 4, estrategia de pruebas: fija antes de construir qué se probará y con qué cobertura mínima. Las rutas críticas exigen cobertura completa.

Ejemplo sencillo: para un módulo que exporta facturas a PDF, el plan decide dónde se generan los ficheros y qué pasa si falla la generación. La revisión detecta que el plan no prevé muchos usuarios a la vez y devuelve el plan con cambios.

Por qué importa: detectar un error de arquitectura en la revisión del plan cuesta una conversación; detectarlo en producción cuesta un incidente.

Pregunta para el aula: ¿por qué cree que las fases 3 y 4 son las que más se saltan?

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

- **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
- **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

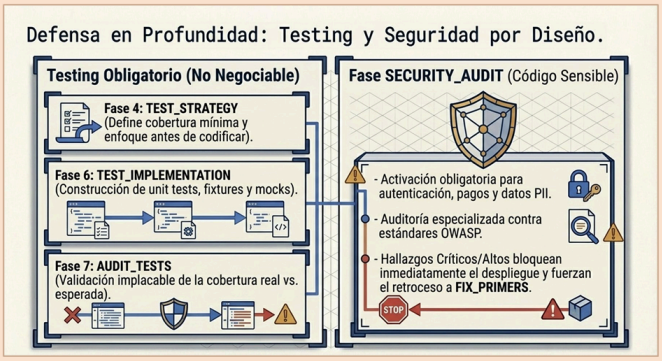
FUENTES Y ENLACES

- **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 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
- **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 EN 20 SLIDES

El ciclo (2/3): implementación, pruebas y revisiones

Fases 5 a 8: se construye lo que dice el plan y otra IA lo revisa antes de seguir.



- Fase 5: la IA constructora ejecuta el plan. Si algo no está previsto, no decide: lo señala.
- Fase 6: se implementan las pruebas de la estrategia. Las que solo copian el código no cuentan.
- Fases 7 y 8: otra IA revisa pruebas, código, dependencias y licencias.
- Lámina de SPAD, en español.

El código sensible pasa además una revisión de seguridad, uno de los ciclos complementarios: lo crítico y alto bloquea el despliegue.

→ Fases de construcción y revisión | → Ciclo de seguridad

SPAD · Versión 0.x, en revisión · CC BY 4.0 · Fernando García Varela seachad.github.io/AI_CONSULTING/ 7 / 20

Notas

Con el plan aprobado, empieza la construcción.

- Fase 5, implementación: la IA constructora escribe el código conforme al plan y a la guía de código. Entrega además la lista de ficheros creados y modificados, las dependencias añadidas y una declaración de conformidad con el plan. Si encuentra una situación que el plan no prevé, no decide: la señala como desviación. La persona que orquesta devuelve el trabajo al plan o acepta una desviación menor documentada.
- Fase 6, implementación de pruebas: se implementan las pruebas de la estrategia, no las que resulten cómodas. Una prueba que solo reproduce el código, sin verificar el comportamiento esperado, no cuenta.
- Fase 7, revisión de pruebas: una IA revisora compara la cobertura real con la esperada y revisa los casos límite y las rutas de error.
- Fase 8, revisión del código: comprueba la fidelidad al plan, el cumplimiento de la guía, los riesgos técnicos, la deuda que se asume y las dependencias: si existen, de dónde vienen, si tienen vulnerabilidades conocidas y su licencia.

Ejemplo sencillo: la IA constructora no sabe cómo reintentar un envío que falla. Lo correcto no es inventar una solución, sino escribir: «el plan no dice cómo manejar el reintento; pido decisión».

Nota sobre la lámina: «auditoría» se refiere a la revisión de la IA revisora y a la revisión de seguridad.

Por qué importa: las pruebas definidas antes de construir evitan que se escriban al final solo para confirmar lo que ya existe.

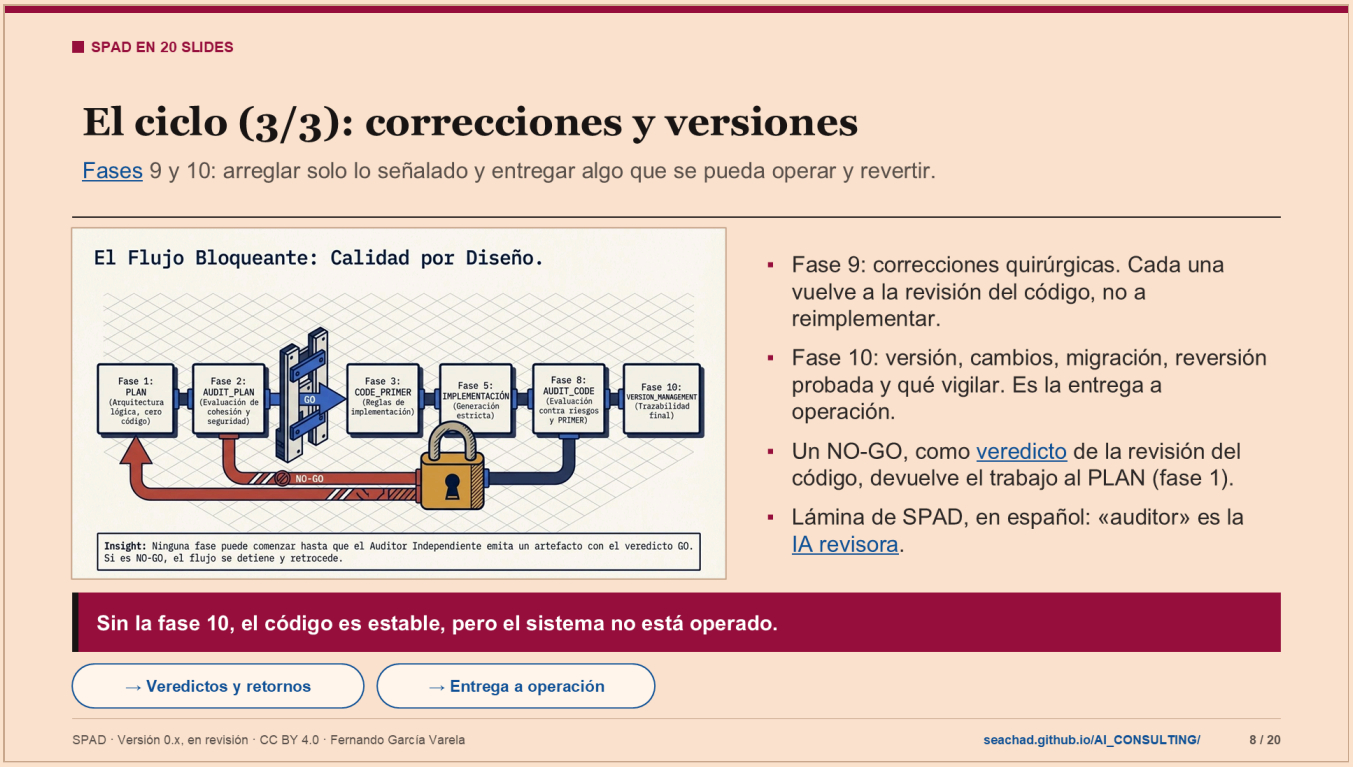
Pregunta para el aula: ¿cómo distinguiría una prueba que verifica el comportamiento de una que solo copia el código?

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
- **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 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 03 · Ejemplos de decisión** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/03_SPAD_Politica_de_validacion.html#ejemplos-de-decision
- **SPAD 04 · Ciclo de seguridad** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/04_SPAD_Ciclos_complementarios.html#ciclo-de-seguridad
- **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
- **Ciclo de seguridad** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/04_SPAD_Ciclos_complementarios.html#ciclo-de-seguridad



Notas

La lámina muestra el flujo bloqueante: un veredicto negativo detiene el flujo y lo devuelve a la fase que corresponde.

- Fase 9, correcciones: la IA correctora aplica cambios mínimos a los hallazgos de las revisiones. Por cada corrección escribe el problema, el cambio, la justificación, el impacto y el riesgo de regresión. No puede rediseñar, refactorizar módulos enteros ni ampliar funcionalidad. Cada corrección vuelve a la revisión del código, y también a la de pruebas si las pruebas cambian.
- Fase 10, gestión de versiones: número de versión, registro de cambios, cambios incompatibles, instrucciones de migración, plan de despliegue, plan de reversión y requisitos de vigilancia.

Los retornos de cada veredicto son fijos:

- Revisión del plan (fase 2): GO lleva a la fase 3; GO con cambios, a la 1; NO-GO, a la 0.
- Revisión de pruebas (fase 7): GO lleva a la 8; GO con cambios, a la 9; NO-GO, a la 6.
- Revisión del código (fase 8): GO lleva a la 10; GO con cambios, a la 9; NO-GO, a la 1.

Ejemplo sencillo: la revisión señala que falta validar una entrada. La corrección añade esa validación y nada más. Si además «aprovecha» para reorganizar el módulo, es una refactorización encubierta y se descarta.

Por qué importa: las correcciones pequeñas y trazables evitan que un arreglo rompa otra parte del sistema. La fase 10 asegura que alguien sabe cómo volver atrás si algo sale mal.

Pregunta para el aula: ¿por qué un NO-GO en la revisión del código vuelve al plan y no a la implementación?

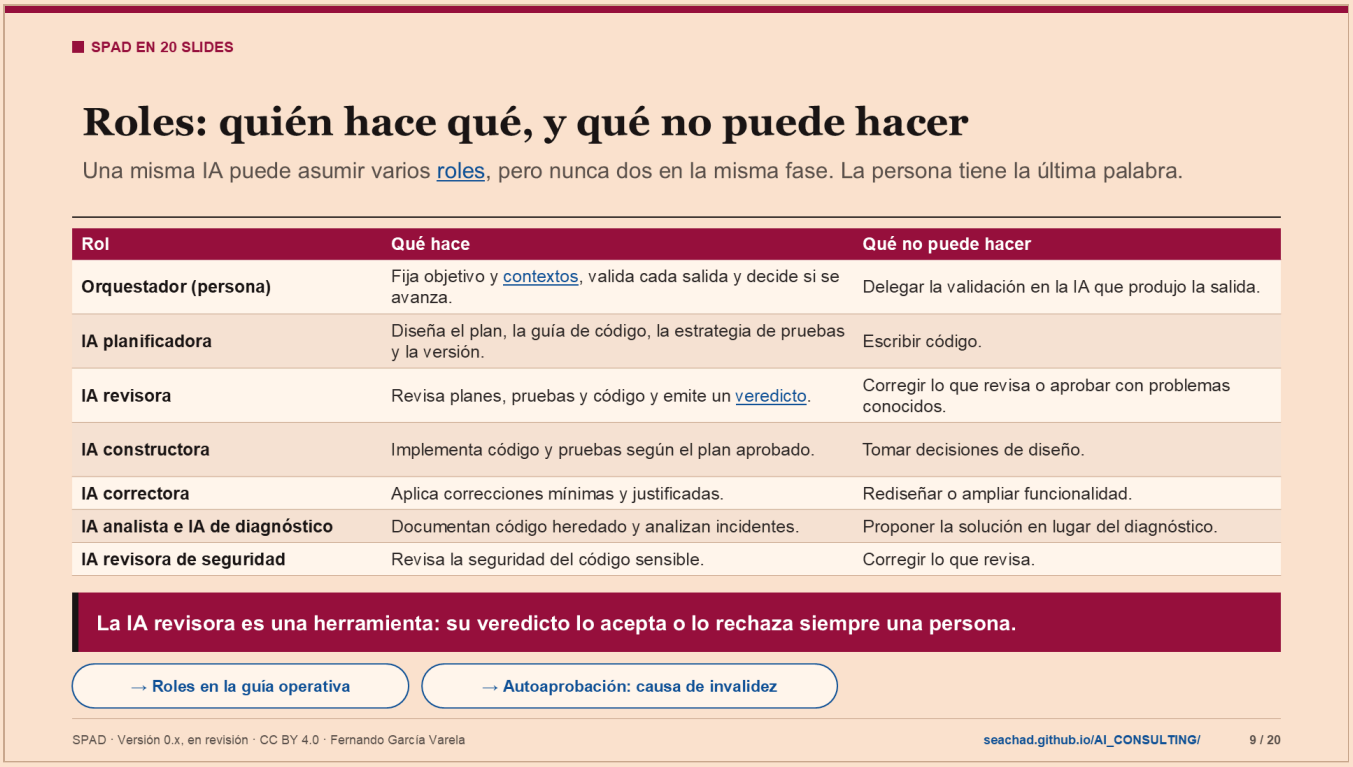
TÉRMINOS DE ESTA SLIDE

- **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
- **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

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
- **Entrega a operación** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/04_SPAD_Ciclos_complementarios.html#entrega-a-operacion



Notas

SPAD separa los roles lógicos de las personas y de las IA que intervienen. Hay un rol humano y siete roles de IA.

- El orquestador es una persona. Define el objetivo, carga los contextos, fija el tema, valida cada salida, aplica la política de validación y decide si se avanza. Tiene la última palabra en todo.
- Las IA tienen funciones separadas: una planifica, otra construye, otra corrige y otra revisa. La analista y la de diagnóstico trabajan en los ciclos de legado y de depuración. La revisora de seguridad revisa el código sensible.

Dos reglas de independencia:

- Una misma IA puede asumir varios roles a lo largo de un trabajo, pero nunca dos roles en la misma fase.
- La IA que revisa un artefacto nunca es la que lo produjo.

Por qué se llama «IA revisora» y no «auditor»: un auditor es una persona de la función que controla, independiente del equipo que construye. La IA revisora pertenece a la función que construye y su salida es solo un insumo: no puede aprobar nada.

Ejemplo sencillo: el mismo modelo puede actuar como planificador por la mañana y como constructor por la tarde, en fases distintas. Pero no puede revisar el plan que él mismo escribió.

Por qué importa: la separación de funciones es un control clásico de cualquier proceso serio: quien hace no se aprueba a sí mismo. SPAD la aplica también entre inteligencias artificiales y deja siempre a una persona como responsable.

Pregunta para el aula: ¿qué riesgo ve en que la IA que revisa sea del mismo proveedor que la que construye?

TÉRMINOS DE ESTA SLIDE

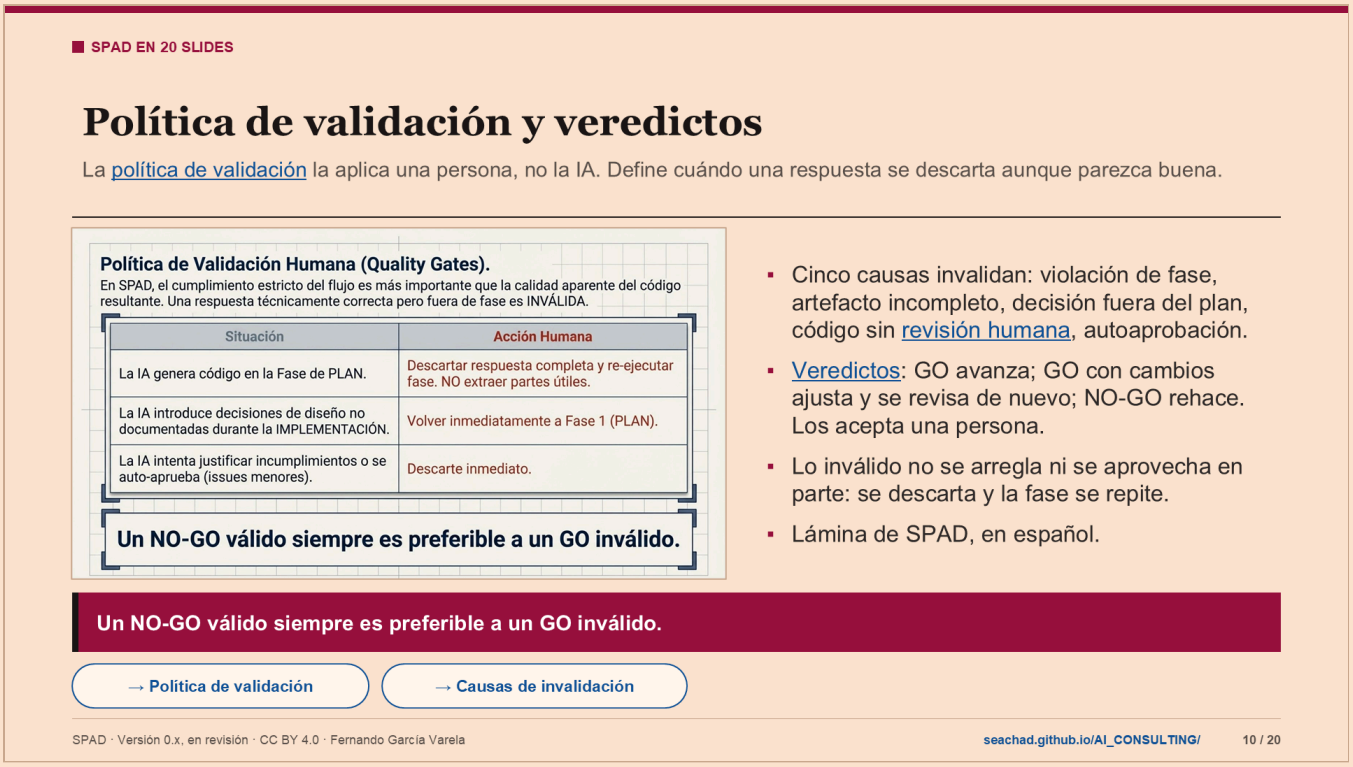
- **Roles de SPAD**: Los papeles que intervienen en SPAD: el orquestador (una persona que dirige el ciclo y valida) y varias IA con funciones separadas: planificadora, revisora, constructora, correctora, analista y de diagnóstico.

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

- **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
- **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 · Roles** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#roles
- **SPAD 00 · Roles** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#roles
- **SPAD 03 · Autoaprobación** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/03_SPAD_Politica_de_validacion.html#3-5-autoaprobacion
- **Roles en la guía operativa** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#roles
- **Autoaprobación: causa de invalidez** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/03_SPAD_Politica_de_validacion.html#3-5-autoaprobacion



Notas

La política de validación es una política humana de gobierno, no una instrucción para la IA. La aplica la persona que orquesta y es obligatoria en todo trabajo SPAD, completo o reducido.

Las cinco causas de invalidación:

- Violación de fase: el plan incluye código o la revisión corrige en lugar de evaluar.
- Artefacto incompleto: por ejemplo, una revisión sin veredicto o una estrategia de pruebas sin cobertura mínima.
- Decisión fuera del plan: la implementación añade una capa que el plan no preveía, aunque esté bien razonada.
- Implementación sin revisión previa: código sin plan aprobado o tras un NO-GO. Se descarta todo ese código.
- Autoaprobación: «hay tres problemas, pero son menores: GO». Es la más grave y se considera una infracción crítica.

Lo que no invalida: un NO-GO bien fundado, una vulnerabilidad detectada o una prueba que falla. Es el proceso funcionando. Tampoco invalida que la IA declare una desviación y pida decisión.

Los veredictos son tres y se usan en todas las revisiones: GO, GO con cambios y NO-GO. Un veredicto sin justificación escrita es un artefacto incompleto. Un GO con hallazgos críticos o altos abiertos es una autoaprobación. El veredicto lo emite la IA revisora y lo acepta o rechaza la persona, que lo registra.

Si la misma infracción se repite, la instrucción se refuerza. Tras el número máximo de intentos que fije el contexto global (valor de partida: tres), se cambia de modelo o interviene una persona.

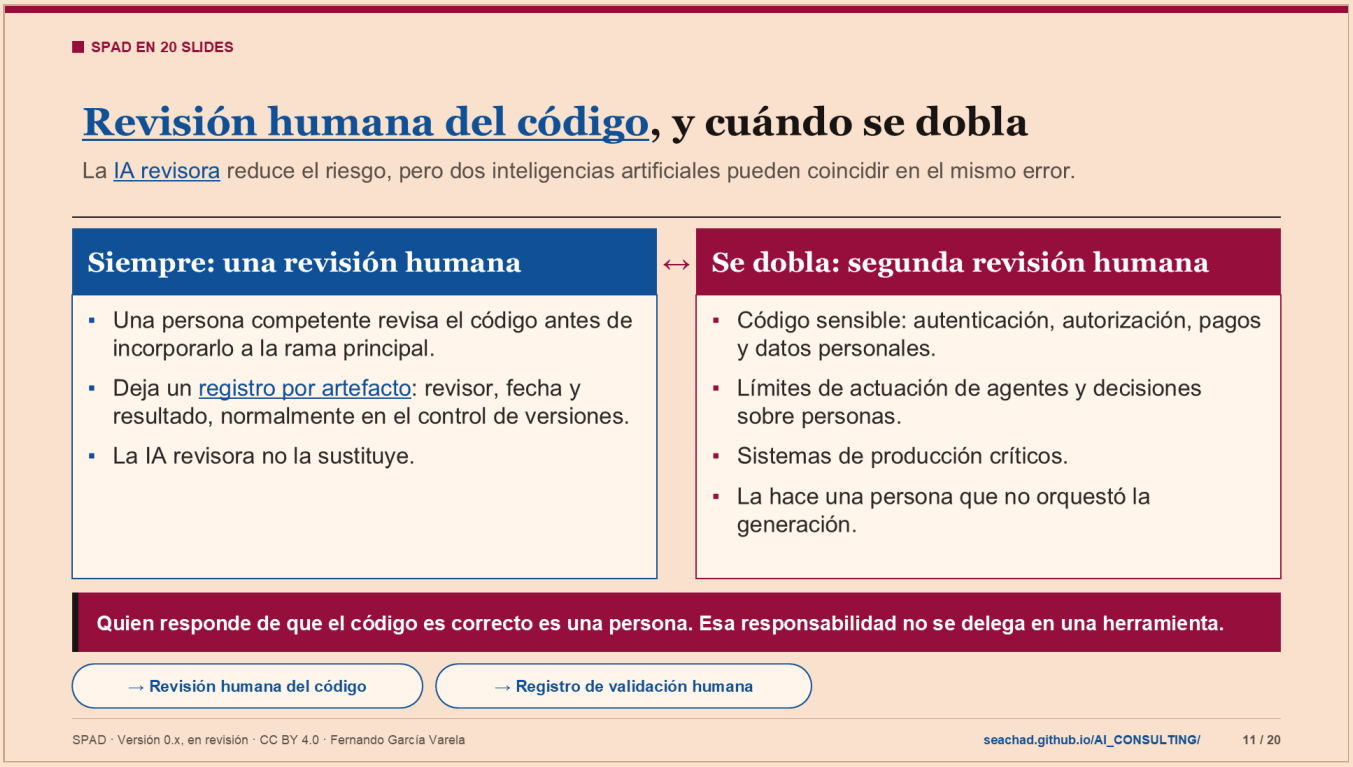
Pregunta para el aula: ¿por qué la autoaprobación es más grave que un plan con código?

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
- **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
- **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 · 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
- **Causas de invalidación** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/03_SPAD_Politica_de_validacion.html#causas-de-invalidacion



Notas

SPAD pone dos capas de revisión sobre el código: la IA revisora, que trabaja en las fases 7 y 8, y la revisión humana.

La revisión humana es obligatoria en todo trabajo SPAD. Al menos una persona competente revisa el código antes de incorporarlo a la rama principal y deja registro de quién revisó, cuándo y con qué resultado. Lo normal es dejarlo en la herramienta de control de versiones.

La revisión se dobla cuando el error sería caro: código de autenticación, autorización, pagos o datos personales; límites de actuación de agentes; decisiones sobre personas; y sistemas de producción críticos. La segunda revisión la hace una persona que no orquestó la generación.

La revisión automatizada nunca sustituye a la humana: la IA revisora reduce el riesgo, pero la responsabilidad de incorporar el código es de una persona, que queda anotada en el registro de validación humana.

Ejemplo sencillo: un cambio en la pantalla de inicio de sesión lo genera la IA, lo revisa la IA revisora y lo revisa una persona del equipo. Como toca la autenticación, una segunda persona, que no pidió el cambio a la IA, lo revisa también.

Por qué importa: la separación entre IA es necesaria, pero no suficiente. La responsabilidad final ante la organización es de una persona.

Error frecuente: dar por revisado el código porque «ya lo revisó otra IA».

Pregunta para el aula: ¿qué perfil debería tener la persona que hace la segunda revisión en su organización?

TÉRMINOS DE ESTA SLIDE

- **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

- **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
- **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 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
- **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



Notas

Tres mecanismos hacen que el trabajo con SPAD sea coherente entre proyectos y reconstruible después.

Contextos. Antes de cualquier fase se cargan dos:

- Contexto global: reglas de toda la organización. Dónde se guardan los artefactos, cobertura mínima de pruebas, versionado, seguridad base, datos y proveedores de modelos, prohibiciones y modelos aprobados.
- Contexto de proyecto: tecnología, arquitectura, reglas de negocio, cumplimiento y excepciones justificadas.

Si chocan, prevalece el de proyecto solo con la excepción escrita, con su riesgo. La IA revisora la evalúa y la aprueba una persona. Ninguna IA aprueba una excepción.

Temas. Cada trabajo tiene un tema: un nombre corto en formato snake_case, de hasta 20 caracteres, como user_auth o payment_gateway. Agrupa todos los artefactos en una carpeta. Un plan rechazado no se borra: se conserva con su revisión y la siguiente iteración toma el número siguiente.

Registros. Cada artefacto lleva:

- Registro del modelo: proveedor, modelo, versión, fecha, rol, instrucción y versión de los contextos.
- Registro de validación humana: quién validó, cuándo, con qué resultado y si aceptó el veredicto de la IA.
- Registro de infracciones: cada respuesta descartada, con su causa.

Ejemplo ilustrativo del documento 02: el contexto global exige un 80 % de cobertura unitaria; un proyecto de migración fija un 70 % para el código nuevo, con plan gradual. El responsable técnico lo aprueba y queda registrado. Las cifras son valores de partida ilustrativos.

Pregunta para el aula: si mañana un auditor le pide por qué un componente se construyó así, ¿dónde lo buscaría hoy?

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 02 · Registros por artefacto** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/02_SPAD_Contextos_temas_y_registro.html#registros-por-artefacto
- **Contextos, temas y registro** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/02_SPAD_Contextos_temas_y_registro.html
- **Registros por artefacto** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/02_SPAD_Contextos_temas_y_registro.html#registros-por-artefacto



Notas

Cuando un equipo trabaja con IA, envía a la herramienta código, documentos y a veces datos. Sin una regla común, cada persona decide por su cuenta qué pega en una herramienta externa. SPAD lo resuelve en el contexto global, con dos secciones: datos y proveedores de modelos, y modelos aprobados.

- Qué se puede enviar: el contexto global dice qué clases de información pueden enviarse a cada proveedor y entorno aprobado, y cuáles nunca: por ejemplo, datos personales, secretos o código propietario, según su clasificación.
- Modelos aprobados: qué proveedores, modelos y versiones se autorizan para cada rol. En código sensible, la IA revisora usa un modelo o proveedor distinto del de la constructora.
- Datos personales: nunca en instrucciones ni en conjuntos de evaluación sin base jurídica documentada y medidas. Por defecto, datos sintéticos.
- Secretos: nunca en instrucciones ni en código.
- Retención y uso por el proveedor: se verifican por contrato antes de usar la herramienta.
- Separación: las instrucciones del sistema y lo que llega del usuario o de documentos recuperados se mantienen separados. Lo recuperado se trata como datos, no como órdenes.

Ejemplo ilustrativo: una regla como «ningún dato personal real en instrucciones a proveedores externos; datos de prueba sintéticos».

Por qué importa: es el punto de unión con la política corporativa de uso de la IA. Y es una de las primeras cosas que una compañía puede preguntar a un proveedor que construye con IA para ella.

Pregunta para el aula: ¿sabe su equipo qué datos puede pegar hoy en una herramienta de IA, y quién lo decidió?

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
- **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

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 05 · Datos y proveedor del modelo** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/05_SPAD_Sistemas_que_incluyen_IA.html#datos-y-proveedor-del-modelo
- **Contexto global** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/02_SPAD_Contextos_temas_y_registro.html#2-1-contexto-global
- **Datos y proveedor del modelo** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/05_SPAD_Sistemas_que_incluyen_IA.html#datos-y-proveedor-del-modelo



Notas

El ciclo principal construye funcionalidad nueva. El documento 04 de SPAD define cuatro ciclos para el resto del trabajo real, más la revisión de la cadena de suministro.

- Legado: hay que cambiar código existente sin documentación suficiente. La IA analista documenta lo que hace hoy y analiza el impacto del cambio. Describe, no propone soluciones. Si no hay pruebas del comportamiento actual, la estrategia de pruebas incluye pruebas de caracterización antes de modificarlo.
- Depuración: aparece un problema. La IA de diagnóstico produce un informe de causa raíz con veredicto. Si la causa es el código, se entra en el ciclo principal; si es configuración o infraestructura, se hace un cambio registrado con reversión.
- Corrección urgente: para incidentes críticos. Diagnóstico rápido, plan simplificado, revisión acelerada pero obligatoria, implementación con pruebas mínimas, despliegue con aprobación de emergencia de una persona, vigilancia y análisis posterior. Una corrección provisional abre siempre un tema para el plan definitivo.
- Seguridad: obligatoria en código sensible. Los hallazgos críticos bloquean y nadie puede aceptarlos como riesgo residual. Los altos bloquean salvo aceptación excepcional por escrito de la autoridad de riesgo.
- Cadena de suministro: la IA puede sugerir paquetes inexistentes o maliciosos. Se comprueba cada dependencia y su licencia.

Ejemplo sencillo: un fallo en producción por un fichero de configuración se corrige como cambio de configuración. El informe de causa raíz se guarda igual: es aprendizaje de la organización.

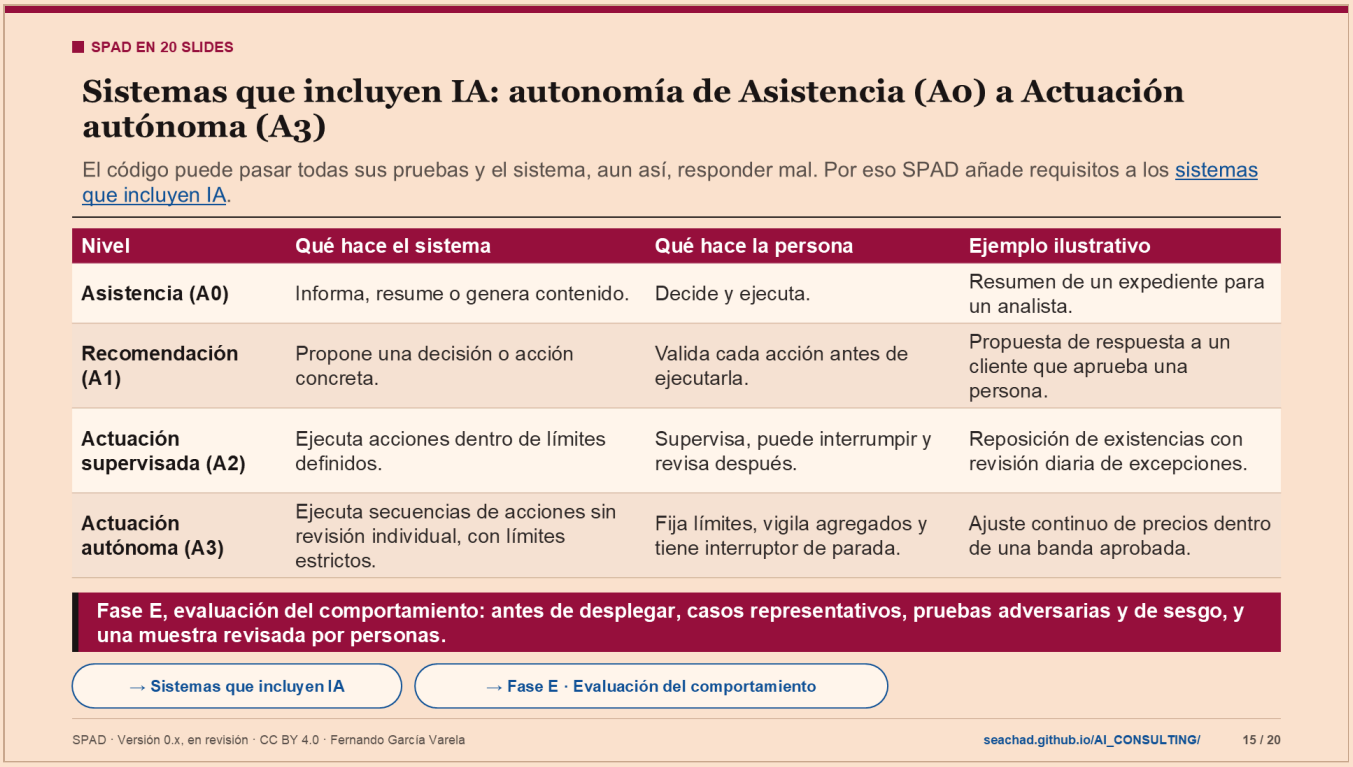
Pregunta para el aula: en una urgencia, ¿qué control tendería a saltarse su equipo primero?

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
- **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 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 · Severidades, bloqueo 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 para aplicaciones con modelos de lenguaje (edición 2025)** https://genai.owasp.org/resource/owasp-top-10-for-llm-applications-2025/
- **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



SPAD EN 20 SLIDES

Sistemas que incluyen IA: autonomía de Asistencia (A0) a Actuación autónoma (A3)

El código puede pasar todas sus pruebas y el sistema, aun así, responder mal. Por eso SPAD añade requisitos a los sistemas que incluyen IA.

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 individual, con límites estrictos.	Fija límites, vigila agregados y tiene interruptor de parada.	Ajuste continuo de precios dentro de una banda aprobada.

Fase E, evaluación del comportamiento: antes de desplegar, casos representativos, pruebas adversarias y de sesgo, y una muestra revisada por personas.

→ Sistemas que incluyen IA → Fase E · Evaluación del comportamiento

SPAD · Versión 0.x, en revisión · CC BY 4.0 · Fernando García Varela seachad.github.io/AI_CONSULTING/ 15 / 20

Notas

SPAD nació para gobernar el código escrito con ayuda de IA. Cada vez más, lo que se construye es un sistema que incluye IA: una aplicación cuyo comportamiento depende, en producción, de un modelo, de unas instrucciones, de una base de conocimiento o de un agente que ejecuta acciones. Ese sistema puede cambiar sin que cambie el código: cuando el proveedor actualiza el modelo, cuando cambian las instrucciones o los datos.

Todo componente de IA que decide o actúa se clasifica desde el plan con una escala de autonomía de cuatro niveles (A0 a A3). El nivel fija la exigencia: A0 y A1 requieren evaluación del comportamiento; A2 y A3 requieren además límites de actuación, registro de acciones e interruptor de parada probado.

La fase E, evaluación del comportamiento, va entre la revisión de pruebas y la revisión del código. El sistema se ejecuta sobre un conjunto versionado de casos representativos, incluidos los que debe rechazar, y se compara con umbrales fijados de antemano. Incluye pruebas adversarias (por ejemplo, intentos de inyectar instrucciones a través de un documento) y pruebas de sesgo con pares contrafactuales cuando el sistema decide sobre personas o se comunica con ellas. Una muestra la revisan personas. Termina con veredicto.

Ejemplo sencillo: un asistente de atención pasa todas las pruebas de código, pero la evaluación muestra que responde con seguridad a preguntas fuera de su alcance. Es un NO-GO de instrucciones: vuelve a la implementación.

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

Pregunta para el aula: ¿qué debería ocurrir cuando el proveedor actualiza el modelo que usa su sistema?

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
- **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 · 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 · Agentes** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/05_SPAD_Sistemas_que_incluyen_IA.html#agentes
- **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



Notas

Esta slide da la idea general de dos documentos técnicos de SPAD: el 06, instrucciones por fase, y el 07, contratos de entrada y salida.

Instrucciones por fase. Cada fase se abre con una instrucción (prompt) de referencia con tres bloques:

- Rol y fase: qué rol asume la IA y en qué fase está.
- Salida obligatoria: las secciones que debe contener la respuesta, en orden.
- Prohibiciones: lo que la fase no admite, por ejemplo, escribir código en el plan.

Todas empiezan con un preámbulo común que recuerda la fase, el rol, los contextos y el tema, y que pide a la IA que no decida cuando algo no está cubierto. Cada fase se ejecuta en su propia interacción: no se fusionan fases. Las organizaciones adaptan el texto, pero conservan los tres bloques.

Contratos. Cada artefacto tiene una estructura con secciones y campos obligatorios. La forma canónica es JSON, aunque puede presentarse en Markdown. Todos comparten un bloque común: tema, fase, iteración, fecha y registro del modelo, y usan las mismas escalas de veredicto y severidad.

Validación automática: un validador comprueba que no falta nada y detecta, por ejemplo, un GO con problemas críticos abiertos. No juzga el contenido: eso es de la IA revisora y de la persona. La herramienta está propuesta como siguiente paso de SPAD; mientras tanto, la comprobación puede hacerse a mano.

Por qué importa: «artefacto incompleto» es la causa de invalidación más fácil de automatizar. El contrato la convierte en un paso mecánico y deja a la persona el juicio que sí requiere una persona.

Pregunta para el aula: ¿qué ventajas tiene que dos equipos usen la misma instrucción para la misma fase?

TÉRMINOS DE ESTA SLIDE

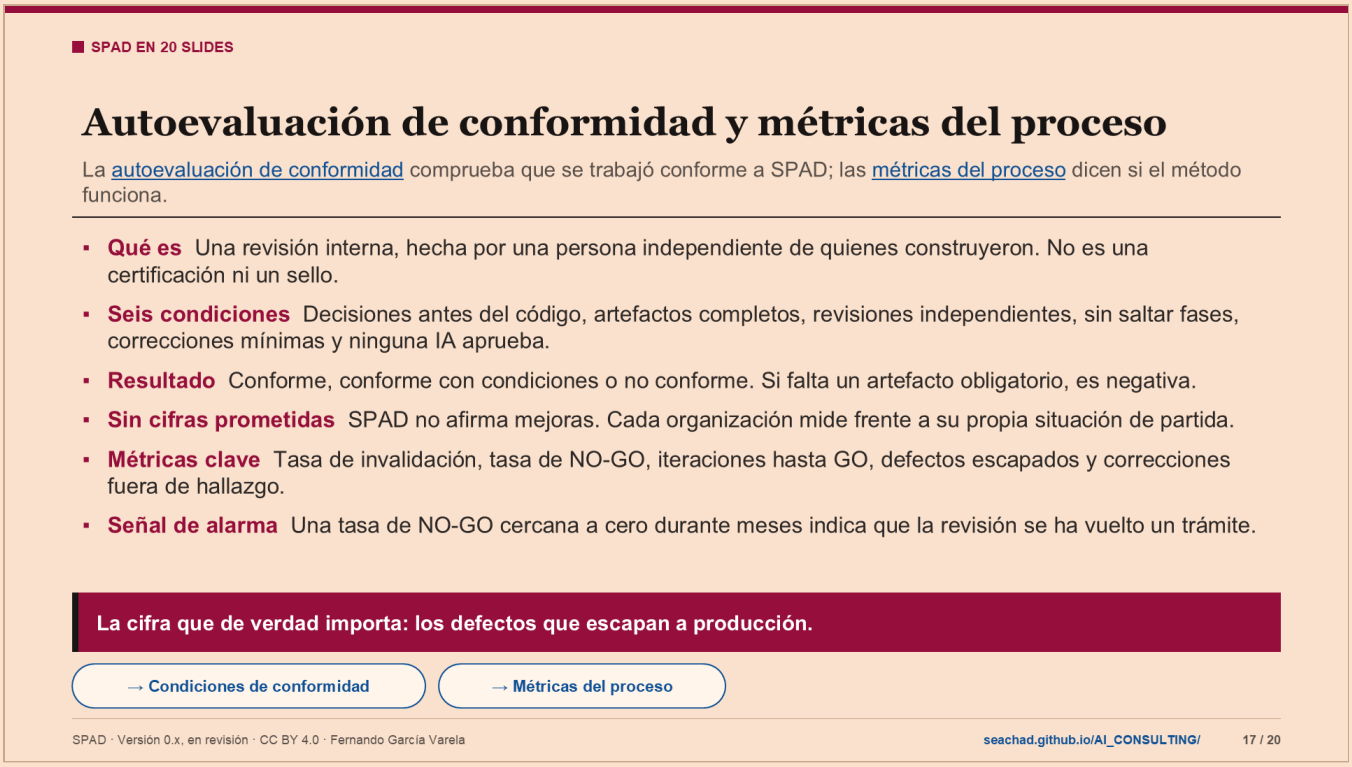
- **Instrucciones por fase**: Las instrucciones que se dan a la IA en cada fase de SPAD, con un preámbulo común y lo propio de cada fase: qué debe leer, qué debe producir y qué no puede hacer. Dan resultados repetibles y

comparables entre equipos. El enlace abre el documento. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/06_SPAD_Instrucciones_por_fase.html

- **Contratos de entrada y salida**: Estructura que debe tener lo que entra y lo que sale de cada fase de SPAD, con un bloque común, para que una herramienta pueda validarlo de forma automática. Así un artefacto incompleto no pasa a la fase siguiente. El enlace abre el documento. https://seachad.github.io/AI_CONSULTING/SPAD/html/es/07_SPAD_Contratos_de_entrada_y_salida.html
- **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 06 · Cómo usar este documento** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/06_SPAD_Instrucciones_por_fase.html#como-usar-este-documento
- **SPAD 06 · Preámbulo común** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/06_SPAD_Instrucciones_por_fase.html#preambulo-comun
- **SPAD 07 · Convenciones comunes** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/07_SPAD_Contratos_de_entrada_y_salida.html#convenciones-comunes
- **SPAD 07 · Validación automática** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/07_SPAD_Contratos_de_entrada_y_salida.html#validacion-automatica
- **Validación automática** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/07_SPAD_Contratos_de_entrada_y_salida.html#validacion-automatica



Notas

La autoevaluación de conformidad la hace la propia organización. Una persona designada, independiente de quienes construyeron y orquestaron, comprueba seis condiciones sobre las evidencias del tema:

- Las decisiones se tomaron antes de escribir código.
- Cada fase produjo su artefacto completo.
- Las revisiones fueron independientes.
- No se saltó ni se fusionó ninguna fase.
- Las correcciones fueron mínimas, justificadas y trazables.
- Ninguna IA aprobó nada.

Si el sistema incluye IA, hay una séptima: la evaluación del comportamiento se hizo antes del despliegue y se repite con cada cambio de modelo, instrucciones o datos.

No es una certificación. No la emite ni la respalda su autor ni ningún tercero, y no acredita el cumplimiento de ninguna norma. Se registra siempre como «conforme con SPAD (autoevaluación)». Su vigencia la fija el contexto global (valor de partida: doce meses).

Métricas del proceso. SPAD no afirma cifras de mejora. Cada organización mide frente a su situación de partida con los registros: tasa de invalidación por fase y por modelo, tasa de NO-GO, iteraciones hasta GO, tiempo por fase, defectos escapados a producción, cobertura real frente a exigida y correcciones sin hallazgo asociado. Como resultado se comparan, antes y después, el retrabajo, los incidentes, el tiempo de preparar una auditoría y el tiempo hasta producción.

Error frecuente: celebrar una tasa de NO-GO nula. Sostenida durante meses, indica una revisión nominal.

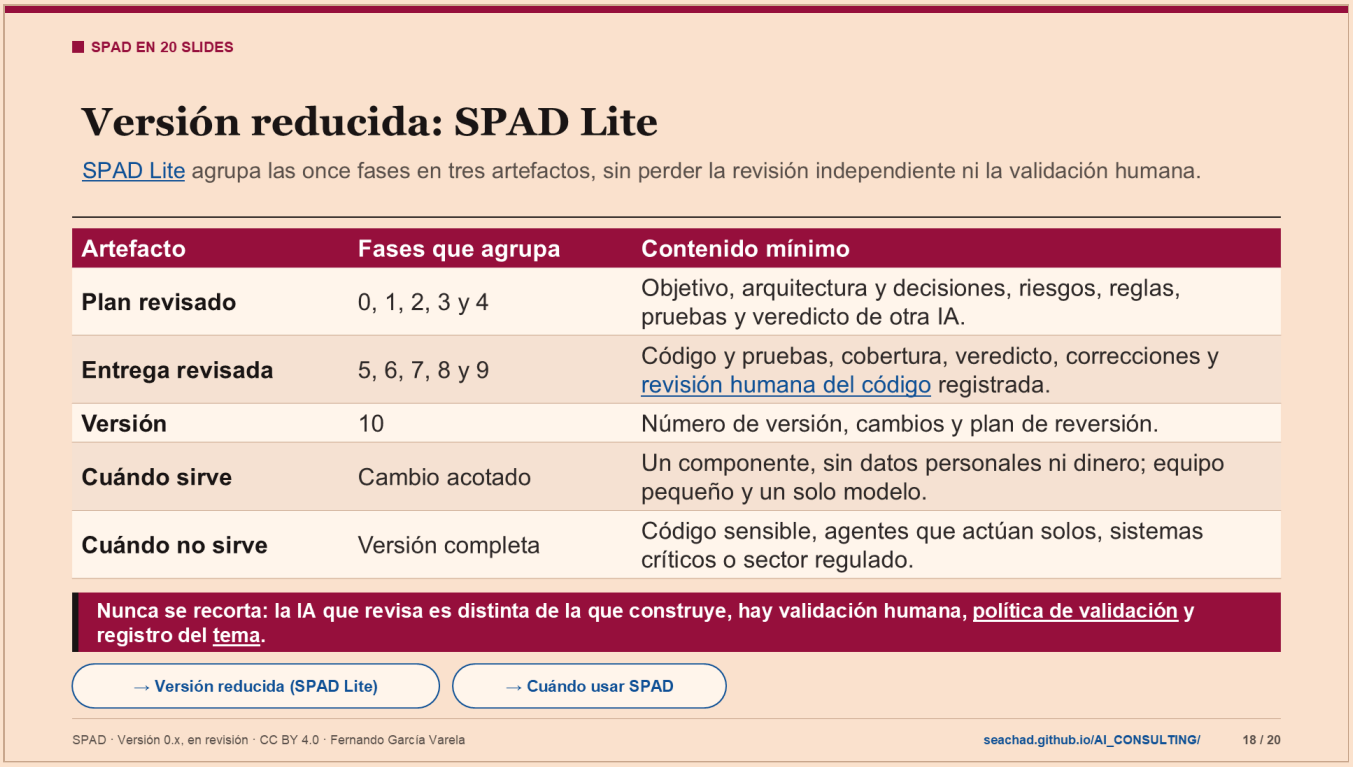
Pregunta para el aula: ¿qué métrica presentaría a un comité de dirección para decidir si se sigue usando SPAD?

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

FUENTES Y ENLACES

- **SPAD 08 · Condiciones de conformidad** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/08_SPAD_Autoevaluacion_y_metricas.html#condiciones-de-conformidad
- **SPAD 08 · Cuándo conviene usar SPAD, según las métricas** 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

No todo trabajo necesita once artefactos separados. La versión reducida agrupa las fases en tres artefactos: plan revisado, entrega revisada y versión.

Criterios para usarla: un cambio acotado en un componente, sin datos personales ni dinero y sin efecto sobre decisiones sobre personas; un equipo de una o dos personas con un solo modelo; y sin obligaciones de auditoría externa sobre el desarrollo.

Criterios que obligan a la versión completa: código sensible, agentes con autonomía de Actuación supervisada (A2) o de Actuación autónoma (A3), sistemas de producción críticos, varios equipos o proveedores de modelos, o un sector regulado con exigencias de trazabilidad del desarrollo.

Lo que no se recorta en ninguna versión: la separación entre la IA que construye y la que revisa, la validación humana de cada artefacto, la política de validación y el registro del tema.

La agrupación solo vale si se mantienen los contenidos mínimos de cada artefacto y la revisión humana queda registrada.

Ejemplo sencillo: añadir un filtro por fecha a un informe interno, sin datos personales. Un solo plan revisado por otra IA, una entrega revisada con su revisión humana y una versión con su plan de reversión bastan.

Por qué importa: sin versión reducida, los equipos aplican SPAD a los proyectos grandes y lo evitan en los pequeños, que son la mayoría y donde se acumula la deuda silenciosa.

Pregunta para el aula: ¿qué trabajos de su organización encajarían en la versión reducida?

TÉRMINOS DE ESTA SLIDE

- **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

- **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
- **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 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
- **Cuándo usar SPAD** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#cuando-usar-spad



Notas

SPAD resuelve un problema concreto: cómo se construye software con IA. Según la posición del autor, a una compañía le sirve de dos maneras. SPAD no promete cifras de ahorro ni de productividad: cada organización mide las suyas.

- Uso 1, construir su propio software con IA. Su equipo trabaja por fases, con un artefacto revisado en cada una, una IA revisora distinta de la que construye y la validación humana de cada salida.
- Uso 2, exigir como cliente. La IA cambia lo que cuesta construir software. Una compañía que sabe cómo se construye con IA puede pedir a sus proveedores e integradores que esa ganancia de productividad llegue a ella: menos coste, o más alcance por el mismo precio.

Qué puede pedir, sin inventar nada que SPAD no defina ya:

- El artefacto de cada fase: plan, revisión del plan, estrategia de pruebas, revisión del código y registro de versión. Así ve que el diseño se decidió antes del código.
- El registro del modelo de cada artefacto: qué parte hizo la IA y con qué herramienta.
- El registro de validación humana y el de infracciones: que una persona aprobó lo que se entrega y qué respuestas se descartaron.
- Los veredictos de cada revisión: dónde se volvió atrás y por qué.
- Las métricas del proceso (tiempo por fase, iteraciones hasta GO, defectos escapados, retrabajo) y la autoevaluación de conformidad: una definición compartida de «terminado».

Por qué importa: si la compañía no sabe cómo se construye con IA, no puede saber si el cambio de coste llega a su factura o se queda en el proveedor. SPAD no fija un porcentaje: permite negociar con datos del propio proveedor.

Ejemplo ilustrativo: ante una oferta de desarrollo, la compañía pide al integrador el tiempo por fase y el registro del modelo de un trabajo anterior, y con esos datos habla de precio y de alcance.

Pregunta para el aula: ¿qué le pediría hoy a su principal proveedor de software para saber si usa IA y quién se queda la ganancia?

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
- **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
- **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 · 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
- **Registros por artefacto** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/02_SPAD_Contextos_temas_y_registro.html#registros-por-artefacto



Notas

Para empezar no hace falta pedir permiso: SPAD se publica bajo CC BY 4.0 (contenidos) y MIT (código), con la única condición de citar la autoría: SPAD · Fernando García Varela.

- Paso 1, elegir dónde aplicarlo. El documento 00 lo dice: muy recomendable en sistemas en producción con clientes, datos sensibles, dinero o decisiones sobre personas, y en sectores con exigencias de auditoría; recomendable con varios desarrolladores o mucha deuda técnica; útil, en versión reducida, en herramientas internas; innecesario en prototipos desechables, programas personales o investigación pura.
- Paso 2, escribir los contextos global y de proyecto. La sección de datos y proveedores y la de modelos aprobados no deben faltar.
- Paso 3, fijar el tema en la fase 0 y asignar los roles. La IA revisora no coincide con la constructora y una persona orquesta.
- Paso 4, abrir el registro de validación y completarlo en el momento, no al final.
- Paso 5, medir. Con los registros se calculan las métricas del proceso y se comparan con la situación de partida.

La lista de comprobación del documento 02 resume lo mínimo antes de empezar: contextos con versión, excepciones aprobadas por una persona, tema y carpeta, modelos asignados por rol, registro de validación abierto y clase de información autorizada para cada proveedor.

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.

Error frecuente: empezar por el proyecto más grande y crítico. Al principio el método hace el trabajo más lento mientras el equipo aprende.

Pregunta para el aula: ¿qué trabajo de su organización elegiría para aplicar SPAD por primera vez, y por qué?

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
- **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
- **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
- **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

- **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 02 · Lista de comprobación 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
- **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
- **Cuándo usar SPAD** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#cuando-usar-spad
- **Lista de comprobación 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
- **Biblioteca SPAD** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/index.html

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/