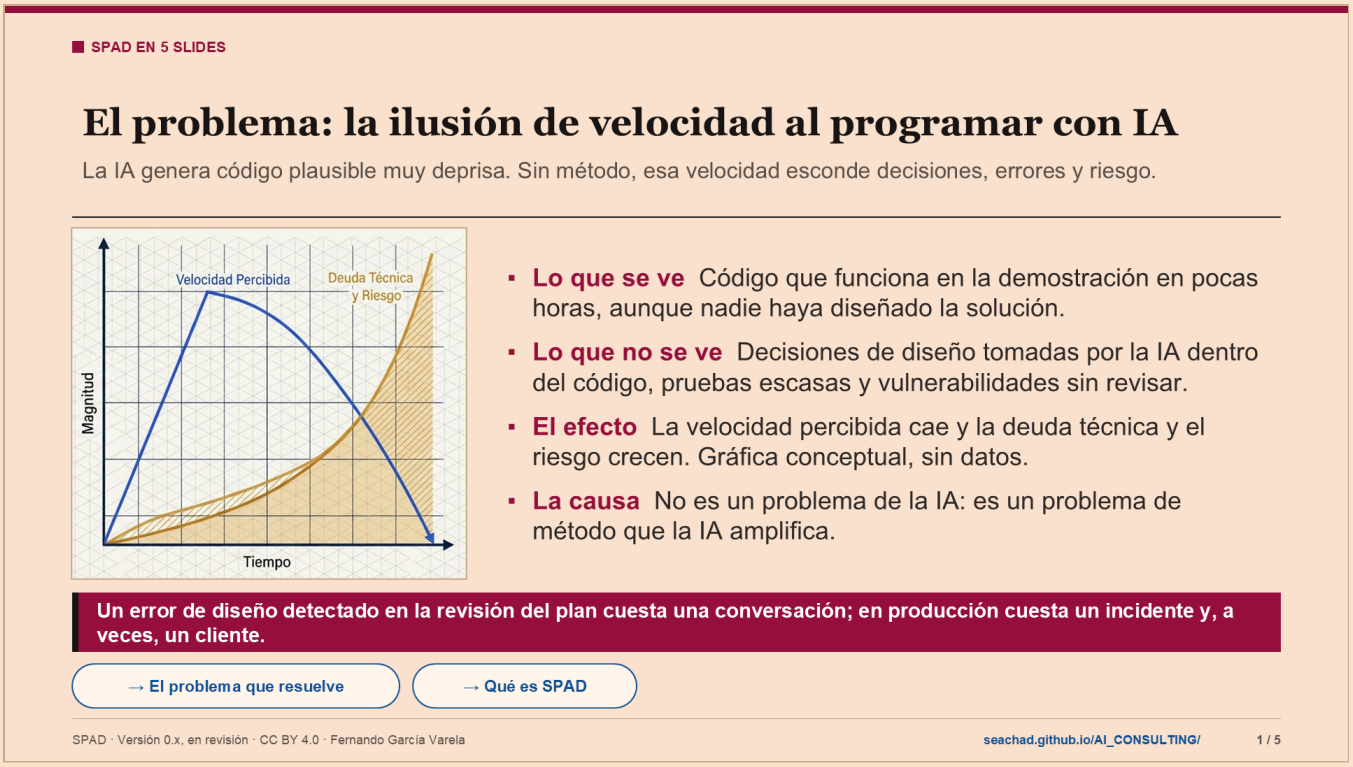


Notas

SPAD significa Structured Prompt-Driven Engineering: ingeniería estructurada guiada por instrucciones. Es una forma de trabajar para construir software con ayuda de la IA de manera controlada, de modo que el trabajo pueda revisarse, reproducirse y auditarse.

La gráfica es conceptual y no tiene datos. Muestra una intuición: al principio, programar con IA parece mucho más rápido. Con el tiempo, si nadie diseña, nadie revisa y nada queda escrito, la velocidad percibida cae y la deuda técnica y el riesgo crecen.

Síntomas habituales que recoge SPAD:

- El código funciona en la demostración y falla en producción, porque se fue pidiendo código hasta que «funcionó».
- Nadie sabe por qué el sistema está construido así, porque las decisiones las tomó la IA dentro del código.
- Aparecen vulnerabilidades, porque revisó la misma IA que escribió el código o no revisó nadie.
- Una auditoría pide evidencias del desarrollo y no existen: el trabajo quedó en conversaciones con la IA.

Ejemplo sencillo: un equipo pide a un asistente de IA una pantalla de pagos. En una tarde funciona. Nadie ha decidido cómo se gestionan los reintentos ni quién puede ver los datos. El fallo aparece meses después, ante un cliente.

Por qué importa a un directivo: la velocidad sin control traslada el coste al futuro, en forma de incidentes, retrabajo y riesgo reputacional o regulatorio. SPAD no promete cifras de mejora; cada organización mide su efecto frente a su propia situación de partida.

Pregunta para el aula: ¿quién en su organización podría explicar hoy por qué una aplicación generada con IA está construida como está?

FUENTES Y ENLACES

SPAD EN 5 SLIDES

- **SPAD 00 · Qué es SPAD y para qué sirve, sección 2** 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** 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** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html#el-problema-que-resuelve
- **Qué es SPAD** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/00_SPAD_Que_es_y_para_que_sirve.html



Notas

SPAD tiene tres propiedades. Es secuencial: el trabajo avanza por fases en un orden fijo. Es bloqueante: ninguna fase empieza hasta que la anterior ha producido su resultado y una persona lo ha validado. Es auditable: cada fase deja un artefacto escrito (un plan, un informe de revisión, una estrategia de pruebas, un registro de versión) que se guarda.

Las tres primeras tarjetas son los tres pilares: pensar, validar y ejecutar. Las otras tres son reglas no negociables. La guía operativa recoge doce, entre ellas:

- Toda decisión es explícita: no hay decisiones de diseño escondidas en el código.
- Correcciones mínimas: quirúrgicas y justificadas, nunca reescrituras encubiertas.
- Pruebas obligatorias, con cobertura mínima fijada antes de construir.
- Seguridad por diseño: el código sensible pasa una revisión de seguridad y los hallazgos críticos y altos bloquean.
- Modelo registrado: cada artefacto dice qué modelo y qué instrucciones lo produjeron.

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 se descarta entera y la fase se repite. Ejemplo: se pide un plan y la IA añade «un ejemplo de implementación» de quinientas líneas. Aunque el código sea bueno, la respuesta es inválida.

Por qué importa a un directivo: si se acepta «solo esta vez» una respuesta que rompe las reglas porque el código funciona, el proceso deja de existir. La disciplina es lo que hace fiables las evidencias.

Pregunta para el aula: ¿qué regla de esta slide le costaría más cumplir a un equipo con prisa, y por qué?

TÉRMINOS DE ESTA SLIDE

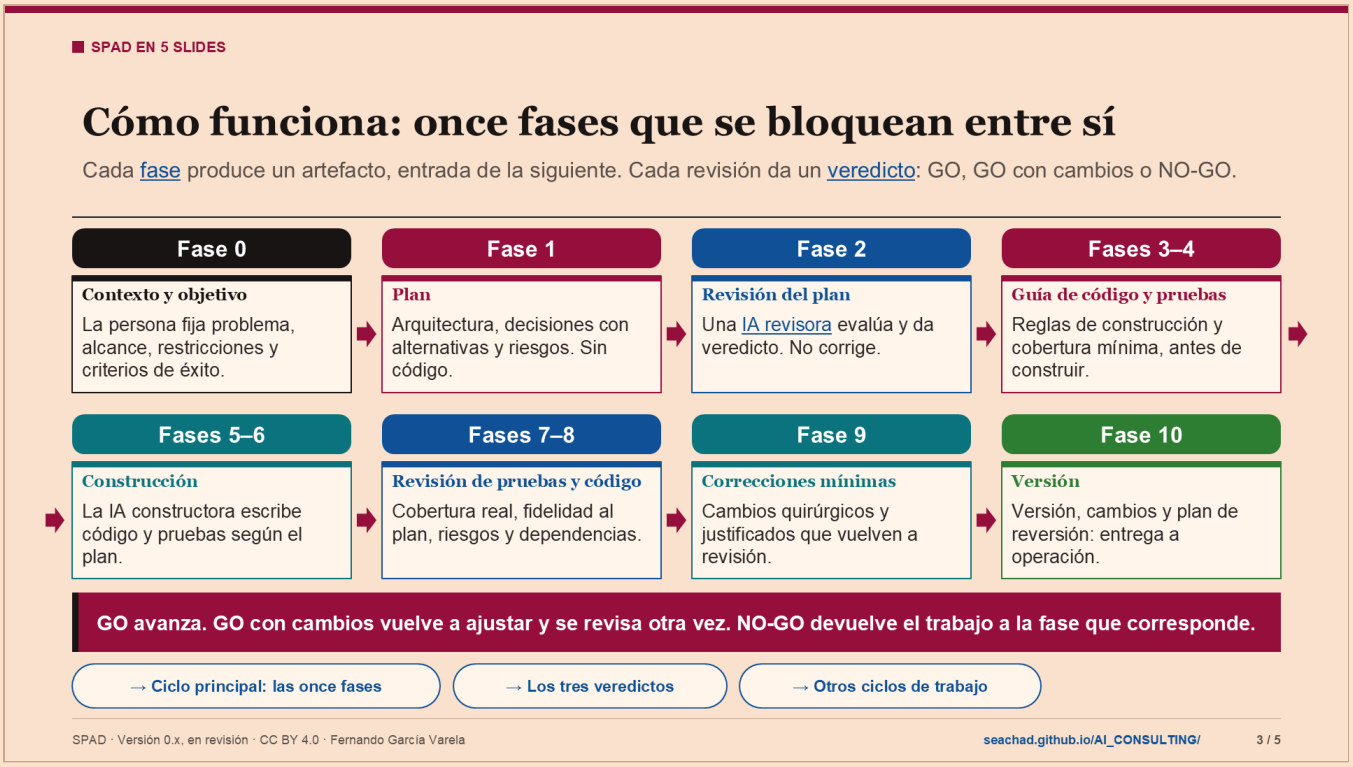
- **Principio de ejecución bloqueante**: Regla central de SPAD: ninguna fase empieza hasta que la anterior ha entregado su artefacto y ha recibido un veredicto favorable. Evita que la IA construya sobre un plan que nadie ha

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

- **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
- **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
- **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 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 01 · Reglas operativas no negociables** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#reglas-operativas-no-negociables
- **SPAD 03 · Política de validación, ejemplos de decisión** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/03_SPAD_Politica_de_validacion.html#ejemplos-de-decision
- **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 no negociables** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#reglas-operativas-no-negociables



Notas

El ciclo principal se usa para construir una funcionalidad nueva y tiene once fases, de la 0 a la 10. La slide agrupa algunas para que se lean de un vistazo.

- Fase 0, contexto y objetivo: sin criterios de éxito no hay fase 1.
- Fase 1, plan: la IA planificadora diseña sin escribir código y anota las alternativas que descartó.
- Fase 2, revisión del plan: una IA distinta evalúa acoplamiento, escalabilidad, seguridad y otros aspectos. Evalúa; no corrige.
- Fases 3 y 4, guía de código y estrategia de pruebas: son las que más se saltan y las que más valor protegen. La guía impide que quien construye decida; la estrategia evita que las pruebas se escriban al final solo para confirmar lo que ya existe.
- Fases 5 y 6, construcción: si el plan no prevé una situación, la IA constructora no decide; la señala y una persona resuelve.
- Fases 7 y 8, revisiones: incluyen las dependencias añadidas y su licencia.
- Fase 9, correcciones: cada corrección vuelve a la revisión del código.
- Fase 10, versión: el sistema queda listo para operar, con plan de despliegue y de reversión.

Los veredictos son siempre los mismos: GO (se avanza), GO con cambios (se ajusta y se revisa de nuevo) y NO-GO (se rehace). Un NO-GO en la revisión del plan devuelve el trabajo a la fase 0, porque el problema está mal planteado. Un NO-GO bien fundado no es un fallo: es el proceso funcionando.

Hay además ciclos para código heredado, depuración, corrección urgente y seguridad. En una urgencia las fases se aceleran, pero la revisión y las pruebas siguen siendo obligatorias.

Por qué importa a un directivo: el ciclo lleva las decisiones caras al momento en que todavía son baratas.

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 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

SPAD distingue roles lógicos, no personas ni herramientas concretas. Una misma IA puede asumir varios roles en un trabajo, pero nunca dos en la misma fase, y la IA que revisa un artefacto nunca es la que lo produjo.

- El orquestador es una persona. Define el objetivo, carga los contextos, fija el tema, valida cada salida y decide si se avanza. No puede delegar la validación en la IA que produjo la salida.
- La IA planificadora diseña y no escribe código. La IA constructora escribe código y no toma decisiones de diseño. La IA correctora aplica correcciones mínimas y no rediseña.
- La IA revisora evalúa y no corrige. En la ilustración aparece como «IA-Auditor»: es la IA revisora. Es una IA, no una persona: su evaluación no sustituye a la revisión humana.

Dos inteligencias artificiales pueden coincidir en el mismo error. Por eso una persona competente revisa también el código antes de incorporarlo. En autenticación, pagos, datos personales, agentes o sistemas críticos, revisa además una segunda persona que no orquestó la generación.

Contextos: antes de cualquier fase se cargan las reglas de la organización (por ejemplo, «ningún dato personal real en instrucciones a proveedores externos») y las del proyecto. Tema: un nombre corto que agrupa todos los artefactos de un trabajo en una carpeta. Registros: cada artefacto dice qué modelo lo produjo, quién lo validó y con qué resultado; y cada respuesta descartada queda anotada.

Por qué importa a un directivo: es la separación de funciones de cualquier proceso serio. Quien responde ante la organización de que el código es correcto es una persona, y esa responsabilidad no se delega en una herramienta.

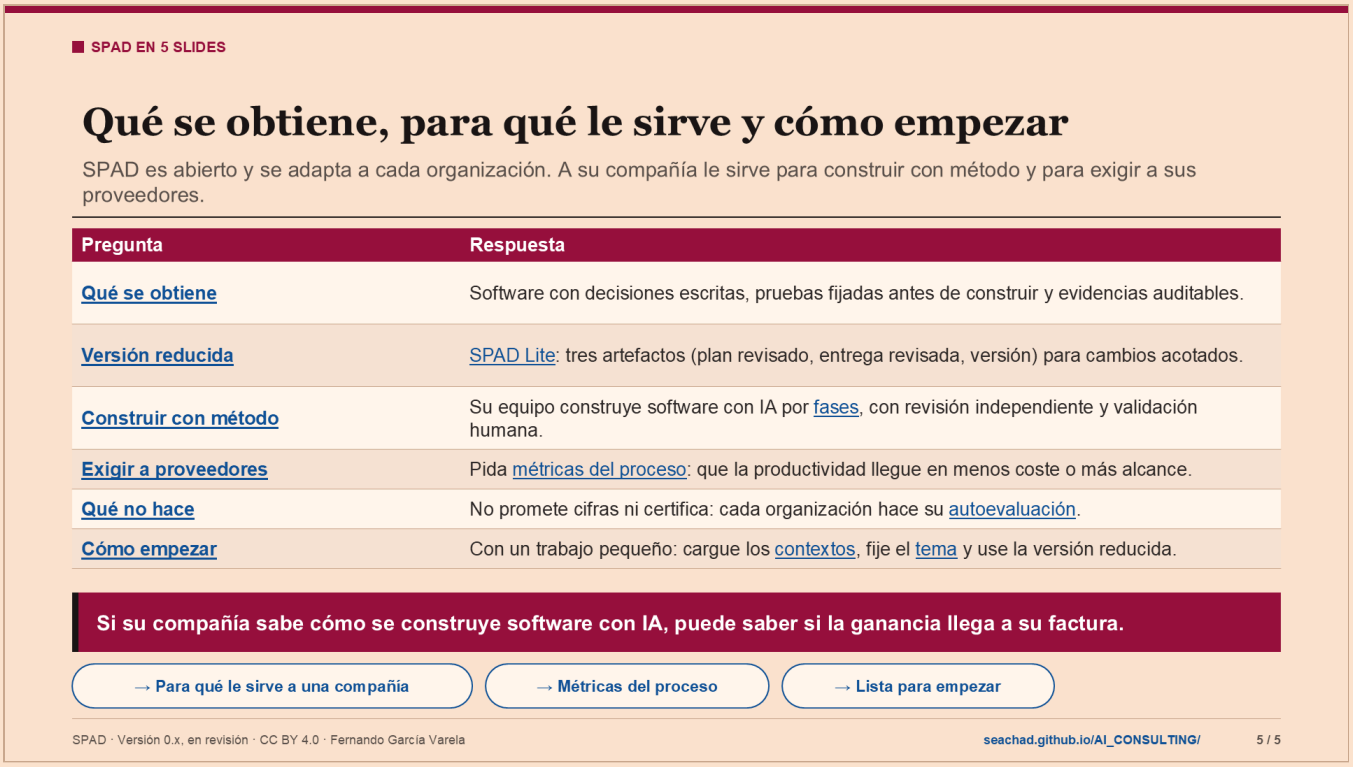
Pregunta para el aula: ¿qué datos de su compañía no deberían llegar nunca a una herramienta de IA externa?

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
- **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
- **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 · Contexto global** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/02_SPAD_Contextos_temas_y_registro.html#2-1-contexto-global
- **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 · Política de validación: autoaprobación** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/03_SPAD_Politica_de_validacion.html#3-5-autoaprobacion
- **Roles y revisión humana** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/01_SPAD_Guia_operativa.html#roles
- **Contextos, temas y registros** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/02_SPAD_Contextos_temas_y_registro.html
- **Autoaprobación: causa de invalidez** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/03_SPAD_Politica_de_validacion.html#3-5-autoaprobacion



SPAD EN 5 SLIDES

Qué se obtiene, para qué le sirve y cómo empezar

SPAD es abierto y se adapta a cada organización. A su compañía le sirve para construir con método y para exigir a sus proveedores.

Pregunta	Respuesta
Qué se obtiene	Software con decisiones escritas, pruebas fijadas antes de construir y evidencias auditables.
Versión reducida	SPAD Lite: tres artefactos (plan revisado, entrega revisada, versión) para cambios acotados.
Construir con método	Su equipo construye software con IA por fases, con revisión independiente y validación humana.
Exigir a proveedores	Pida métricas del proceso: que la productividad llegue en menos coste o más alcance.
Qué no hace	No promete cifras ni certifica: cada organización hace su autoevaluación.
Cómo empezar	Con un trabajo pequeño: cargue los contextos, fije el tema y use la versión reducida.

Si su compañía sabe cómo se construye software con IA, puede saber si la ganancia llega a su factura.

→ Para qué le sirve a una compañía | → Métricas del proceso | → Lista para empezar

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

Notas

Qué se obtiene: decisiones tomadas antes de escribir código, un artefacto por fase, revisiones independientes y correcciones trazables. Todo queda en la carpeta del tema, que es la unidad de auditoría. La organización puede hacer una autoevaluación de conformidad, con una persona independiente de quienes construyeron. No es una certificación: nadie la emite ni la respalda desde fuera.

Versión reducida (SPAD Lite): agrupa las fases en tres artefactos. Sirve para cambios acotados, sin datos personales ni dinero, con un equipo pequeño. Nunca se recorta la separación entre la IA que construye y la que revisa, la validación humana, la política de validación ni el registro del tema. El código sensible, los agentes con autonomía alta y los sistemas críticos exigen la versión completa.

Para qué le sirve a su compañía (posición del autor, sin cifras). Primer uso: construir su propio software con IA con método, con fases, artefactos revisados, revisión independiente y validación humana. Segundo uso, como cliente: una compañía que sabe cómo se construye software con IA puede exigir a sus proveedores e integradores que la ganancia de productividad llegue a ella, en menos coste o en más alcance por el mismo precio. Puede pedirles lo que SPAD ya define, sin inventar nada más: el artefacto de cada fase, el registro del modelo y de la validación humana, los veredictos de cada revisión y las métricas del proceso (tiempo por fase, iteraciones hasta GO, defectos escapados, retrabajo). Por qué importa: la IA cambia lo que cuesta construir software; sin ese conocimiento, la compañía no sabe si el cambio llega a su factura o se queda en el proveedor. SPAD no fija un porcentaje: permite negociar con datos.

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.

Cómo empezar: elija un trabajo pequeño; cargue el contexto global y el del proyecto; fije el tema; asigne una IA revisora distinta de la constructora; abra el registro de validación; y compruebe qué información puede enviar a cada proveedor. Al principio irá más lento; mida retrabajo e incidentes frente a su punto de partida.

SPAD se publica con licencia CC BY 4.0 (contenidos) y MIT (código): puede usarse y adaptarse citando la autoría.

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
- **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
- **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
- **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

- **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 · 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
- **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
- **Lista para empezar** https://seachad.github.io/AI_CONSULTING/SPAD/html/es/02_SPAD_Contextos_temas_y_registro.html#lista-de-comprobacion-antes-de-empezar

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/